A New Encoding of Not Necessarily Closed Convex Polyhedra

Roberto Bagnara, Patricia M. Hill, Enea Zaffanella
University of Parma, Italy
University of Leeds, United Kingdom

http://www.cs.unipr.it/ppl/
**Convex Polyhedra: What and Why**

**What?**
- regions of \( \mathbb{R}^n \) bounded by a finite set of hyperplanes.

**Why? Solving Classical Data-Flow Analysis Problems!**
- array bound checking and compile-time overflow detection;
- loop invariant computations and loop induction variables.

**Why? Verification of Concurrent and Reactive Systems!**
- synchronous languages;
- linear hybrid automata (roughly, FSMs with time requirements);
- systems based on temporal specifications.

**And Again: Many Other Applications...**
- inferring argument size relationships in logic programs;
- termination inference for Prolog programs;
- string cleanness for C programs.
**Constraint Representation:** \( \text{con}(C) \)

- If \( a \in \mathbb{R}^n, a \neq 0 \), and \( b \in \mathbb{R} \), the linear non-strict (resp., strict) inequality constraint \( \langle a, x \rangle \geq b \) (resp., \( \langle a, x \rangle > b \)) defines a closed (resp., open) affine half-space.

- Mixed constraint systems \( \iff \) NNC polyhedra.
**NOT NECESSARILY CLOSED POLYHEDRA**

**Constraint Representation:** \( \text{con}(C) \)
- If \( \mathbf{a} \in \mathbb{R}^n, \mathbf{a} \neq \mathbf{0} \), and \( b \in \mathbb{R} \), the linear non-strict (resp., strict) inequality constraint \( \langle \mathbf{a}, \mathbf{x} \rangle \geq b \) (resp., \( \langle \mathbf{a}, \mathbf{x} \rangle > b \)) defines a closed (resp., open) affine half-space.
- Mixed constraint systems \( \iff \) NNC polyhedra.

**Generator Representation:** \( \text{gen}(\mathcal{G}) \), where \( \mathcal{G} = (\mathcal{R}, \mathcal{P}, C) \)
- \( \mathbf{r} \in \mathbb{R}^n \) is a ray of \( \mathcal{P} \subseteq \mathbb{R}^n \) iff it is a direction of infinity for \( \mathcal{P} \);
- \( \mathbf{p} \in \mathbb{R}^n \) is a point of \( \mathcal{P} \subseteq \mathbb{R}^n \) iff \( \mathbf{p} \in \mathcal{P} \).
- \( \mathbf{c} \in \mathbb{R}^n \) is a closure point of \( \mathcal{P} \subseteq \mathbb{R}^n \) iff \( \mathbf{c} \in \mathcal{C}(\mathcal{P}) \).
- All NNC polyhedra can be expressed as

\[
\left\{ \begin{array}{c}
R\rho + P\pi + C\gamma \\
\rho \in \mathbb{R}^r_+, \pi \in \mathbb{R}^p_+, \gamma \in \mathbb{R}^c_+,
\pi \neq \mathbf{0}, \sum_{i=1}^{p} \pi_i + \sum_{i=1}^{c} \gamma_i = 1
\end{array} \right. 
\]

- Extended generator systems \( \iff \) NNC polyhedra.
\( \mathcal{P} = \text{con}(\{2 \leq x, x < 5, 1 \leq y \leq 3, x + y > 3\}) \).
SAME EXAMPLE USING GENERATORS (I)

\[ \mathcal{P} = \text{gen}((R, P, C)) = \text{gen}((\emptyset, \emptyset, \emptyset)). \]
\[ \mathcal{P} = \text{gen}(\langle R, P, C \rangle) = \text{gen}\left(\langle \emptyset, \{A\}, \emptyset \rangle\right). \]
\[ \mathcal{P} = \text{gen}(\langle R, P, C \rangle) = \text{gen}\left(\emptyset, \{A\}, \{B\}\right). \]
\( \mathcal{P} = \text{gen}\left((R, P, C')\right) = \text{gen}\left((\emptyset, \{A\}, \{B, C\})\right) \).
SAME EXAMPLE USING GENERATORS (V)

\[ \mathcal{P} = \text{gen}((R, P, C)) = \text{gen}\left( (\emptyset, \{A\}, \{B, C, D\}) \right). \]
SAME EXAMPLE USING GENERATORS (VI)

\[ \mathcal{P} = \text{gen}((R, P, C)) = \text{gen}\left((\emptyset, \{A, E\}, \{B, C, D\})\right). \]
ENCODING NNC POLYHEDRA AS C POLYHEDRA

→ Let $\mathbb{P}_n$ and $\mathbb{CP}_n$ be the sets of all NNC and closed polyhedra, respectively: each $\mathcal{P} \in \mathbb{P}_n$ can be embedded into $\mathcal{R} \in \mathbb{CP}_{n+1}$.

→ A new dimension is added, the $\epsilon$ coordinate:
  - to distinguish between strict and non-strict constraints;
  - to distinguish between points and closure points.

→ (Will denote by $e$ the coefficient of the $\epsilon$ coordinate.)

→ The encoded NNC polyhedron:

$$\mathcal{P} = \llbracket \mathcal{R} \rrbracket \overset{\text{def}}{=} \{ \mathbf{v} \in \mathbb{R}^n \mid \exists e > 0 \cdot (\mathbf{v}^T, e)^T \in \mathcal{R} \}.$$
Example: Encoding $P_1$ into $CP_2$

$R_1$ encodes $P_1 = \text{con}\left(\{0 < x \leq 1\}\right)$,
$R_2$ encodes $P_2 = \text{con}\left(\{2 \leq x \leq 3\}\right)$.
THE APPROACH BY HALBWACHS ET AL.

If $\mathcal{P} \in \mathbb{P}_n$ and $\mathcal{P} = \text{con}(\mathcal{C})$, where

$$
\mathcal{C} = \{ \langle \mathbf{a}_i, \mathbf{x} \rangle \triangleright{i} b_i \mid i \in \{1, \ldots, m\}, \mathbf{a}_i \in \mathbb{R}^n, \triangleright{i} \in \{\geq, >\}, b_i \in \mathbb{R} \},
$$

then $\mathcal{R} \in \mathbb{C}\mathbb{P}_{n+1}$ is defined by $\mathcal{R} = \text{con}(\text{con\_repr}(\mathcal{C}))$, where

$$
\text{con\_repr}(\mathcal{C}) \overset{\text{def}}{=} \{0 \leq \epsilon \leq 1\}
\cup \{ \langle \mathbf{a}_i, \mathbf{x} \rangle - 1 \cdot \epsilon \geq b_i \mid i \in \{1, \ldots, m\}, \triangleright{i} \in \{>\} \}
\cup \{ \langle \mathbf{a}_i, \mathbf{x} \rangle + 0 \cdot \epsilon \geq b_i \mid i \in \{1, \ldots, m\}, \triangleright{i} \in \{\geq\} \}.
$$

If $\mathcal{P} \in \mathbb{P}_n$ and $\mathcal{P} = \text{gen}(\mathcal{G})$, where $\mathcal{G} = (R, P, C)$, then $\mathcal{R} \in \mathbb{C}\mathbb{P}_{n+1}$ is defined by $\mathcal{R} = \text{gen}(\text{gen\_repr}(\mathcal{G})) = \text{gen}(\,(R', P')\,$), where

$$
R' = \{ (\mathbf{r}^T, 0)^T \mid \mathbf{r} \in R \},
$$

$$
P' = \{ (\mathbf{p}^T, 1)^T, (\mathbf{p}^T, 0)^T \mid \mathbf{p} \in P \} \cup \{ (\mathbf{c}^T, 0)^T \mid \mathbf{c} \in C \}.$$

With a little precaution the operations on representations do (or can be slightly modified to do) what is expected:

- intersection;
- convex polyhedral hull;
- affine image and preimage;
- ...

This encoding is used in the New Polka library by B. Jeannet and in the Parma Polyhedra Library.

Is this approach the only possible one?

Can we generalize this construction so as to preserve its good qualities?
Suppose we do not add any $\epsilon$-upper-bound constraint:

$\mathcal{R}_1$ encodes $\mathcal{P}_1 = \text{con}(\{0 < x < 1\})$,

$\mathcal{R}_2$ encodes $\mathcal{P}_2 = \text{con}(\{2 \leq x \leq 3\})$. 
\textbf{... Because Otherwise the Poly-Hull is Not Correct}

The poly-hull $\mathcal{P}_1 \cup \mathcal{P}_2$ is \textbf{not} represented correctly by $\mathcal{R}_1 \cup \mathcal{R}_2$.

\[
\mathcal{P}_1 \cup \mathcal{P}_2 \overset{\text{def}}{=} \text{con}(\{0 < x \leq 3\}),
\]
\[
\mathcal{R}_1 \cup \mathcal{R}_2 \text{ encodes } \mathcal{P}' = \text{con}(\{0 \leq x \leq 3\}).
\]
Suppose we do not add the non-negativity constraint for $\epsilon$:

- $\mathcal{R}_1$ encodes $\mathcal{P}_1 = \text{con}(\{0 < x < 1\})$,
- $\mathcal{R}_2$ encodes $\mathcal{P}_2 = \text{con}(\{2 \leq x \leq 3\})$.
The poly-hull $\mathcal{P}_1 \cup \mathcal{P}_2$ is not represented correctly by $\mathcal{R}_1 \cup \mathcal{R}_2$.

$$\mathcal{P}_1 \cup \mathcal{P}_2 \overset{\text{def}}{=} \text{con}(\{0 < x \leq 3\}),$$

$\mathcal{R}_1 \cup \mathcal{R}_2$ encodes $\mathcal{P}'' = \text{con}(\{0 < x < 4\})$. 
In the encoding, for each strict inequality constraint, do also add the corresponding non-strict inequality.

\[ \mathcal{R}'_1 \overset{\text{def}}{=} \text{con}(\{ \epsilon \leq 1, x - \epsilon \geq 0, x \geq 0, -x - \epsilon \geq -1, -x \geq -1 \}). \]
... But This Time There is a Workaround!

In the encoding, for each strict inequality constraint, do also add the corresponding non-strict inequality.

\[ \mathcal{R}_1' \overset{\text{def}}{=} \text{con}(\{\epsilon \leq 1, x - \epsilon \geq 0, x \geq 0, -x - \epsilon \geq -1, -x \geq -1\}) \]
THE ALTERNATIVE ENCODING

→ If \( \mathcal{P} \in \mathbb{P}_n \) and \( \mathcal{P} = \text{con}(\mathcal{C}) \), where

\[
\mathcal{C} = \{ \langle \mathbf{a}_i, \mathbf{x} \rangle \triangleright_i b_i \mid i \in \{1, \ldots, m\}, \mathbf{a}_i \in \mathbb{R}^n, \triangleright_i \in \{\geq, >\}, b_i \in \mathbb{R} \},
\]

then \( \mathcal{R} \in \mathbb{C}\mathbb{P}_{n+1} \) is defined by \( \mathcal{R} = \text{con}(\text{con}_\text{repr}(\mathcal{C})) \), where

\[
\text{con}_\text{repr}(\mathcal{C}) \overset{\text{def}}{=} \{ \epsilon \leq 1 \}
\]

\[
\cup \{ \langle \mathbf{a}_i, \mathbf{x} \rangle - 1 \cdot \epsilon \geq b_i \mid i \in \{1, \ldots, m\}, \triangleright_i \in \{>\} \}
\]

\[
\cup \{ \langle \mathbf{a}_i, \mathbf{x} \rangle + 0 \cdot \epsilon \geq b_i \mid i \in \{1, \ldots, m\}, \triangleright_i \in \{\geq, >\} \}.
\]

→ If \( \mathcal{P} \in \mathbb{P}_n \) and \( \mathcal{P} = \text{gen}(\mathcal{G}) \), where \( \mathcal{G} = (R, P, C) \), then \( \mathcal{R} \in \mathbb{C}\mathbb{P}_{n+1} \) is defined by \( \mathcal{R} = \text{gen}(\text{gen}_\text{repr}(\mathcal{G})) = \text{gen}(\langle R', P' \rangle) \), where

\[
R' = \{(\mathbf{0}^T, -1)^T \} \cup \{ (\mathbf{r}^T, 0)^T \mid \mathbf{r} \in R \},
\]

\[
P' = \{(\mathbf{p}^T, 1)^T \mid \mathbf{p} \in P \} \cup \{ (\mathbf{q}^T, 0)^T \mid \mathbf{q} \in C \}.
\]
The alternative encoding has dual properties with respect to the original by Halbwachs et al.

With the original, the encoding of an NNC polyhedron may require a similar number of constraints but about twice the number of generators: it is constraint-biased.

With the alternative, it may require a similar number of generators but twice the number of constraints: this encoding is generator-biased.

Due to the use of exponential algorithms, their computational behavior can vary wildly depending on the operation and on the actual polyhedra being manipulated.

It seems likely that the performance of one encoding with respect to the other will heavily depend on the particular application.
**Future Work**

- An implementation of the proposed techniques is ongoing.
  - Interested? Go to [http://www.cs.unipr.it/ppl/](http://www.cs.unipr.it/ppl/), learn how to access the CVS repository anonymously, and check out the alt_nnc development branch!

- Can we devise efficient techniques so as to use both constraint- and generator-biased encodings, switching dynamically from one to the other in an attempt to maximize performance?

- A minimized encoding may represent a non-minimized NNC polyhedron:
  - this is true for both encodings;
  - in our SAS’02 paper we propose a stronger form of minimization;
  - we are working on a generalization of this idea that encompasses both the constraint- and the generator-biased encodings.