Inferring Min and Max Invariants Using Max-Plus Polyhedra

Xavier Allamigeon[1,3], Stéphane Gaubert[2], and Éric Goubault[3]

[1] EADS Innovation Works, SE/CS – Suresnes, France
[2] INRIA Saclay and CMAP, École Polytechnique, France
[3] CEA, LIST MeASI – Gif-sur-Yvette, France
firstname.lastname@{eads.net,inria.fr,cea.fr}

Abstract. We introduce a new numerical abstract domain able to infer min and max invariants over the program variables, based on *max-plus polyhedra*. Our abstraction is more precise than octagons, and allows to express non-convex properties without any disjunctive representations. We have defined sound abstract operators, evaluated their complexity, and implemented them in a static analyzer. It is able to automatically compute precise properties on numerical and memory manipulating programs such as algorithms on strings and arrays.

1 Introduction

We present a new abstract domain that generalizes zones [1] and octagons [2] (*i.e.* invariants of the form $\mathsf{x}_i - \mathsf{x}_j \geq c_{ij}$ and $\pm\mathsf{x}_i \pm \mathsf{x}_j \geq c'_{ij}$ respectively), while expressing a certain amount of disjunctive properties. Abstract values are max-plus polyhedra, and allow to infer relations of the form $\max(\lambda_0, \mathsf{x}_1 + \lambda_1, \dots, \mathsf{x}_n + \lambda_n) \leq \max(\mu_0, \mathsf{x}_1 + \mu_1, \dots, \mathsf{x}_n + \mu_n)$ and $\min(\lambda'_0, \mathsf{x}_1 + \lambda'_1, \dots, \mathsf{x}_n + \lambda'_n) \leq \min(\mu'_0, \mathsf{x}_1 + \mu'_1, \dots, \mathsf{x}_n + \mu'_n)$ over program variables $\mathsf{x}_1, \dots, \mathsf{x}_n$, with constants λ_i, μ_i in $\mathbb{R} \cup \{-\infty\}$ and λ'_i, μ'_i in $\mathbb{R} \cup \{+\infty\}$. For instance, the constraint $\max(\mathsf{x}, \mathsf{y}) = \max(-\infty, \mathsf{z}) = \mathsf{z}$, which forms a particular max-plus polyhedron, encodes both $\mathsf{x} - \mathsf{z} \leq 0$ and $\mathsf{y} - \mathsf{z} \leq 0$ (zone information), and "either x or y is z" (disjunctive information). Intuitively, max-plus polyhedra are the analogues of "classical" closed convex polyhedra in the max-plus algebra, which is the set $\mathbb{R} \cup \{-\infty\}$ endowed with max as additive and + as multiplicative laws.

Max-plus polyhedra encode disjunctive information, the worst-case complexity of some abstract operators may be important, although relatively similar to the case of classical polyhedra (see Theorem 1), but this disjunctive information is treated entirely semantically, and is thus fairly efficient for notoriously difficult problems in static analysis, such as proving that sorting algorithms indeed do sort (see Sect. 4). It is in the best of our knowledge the first domain whose elements describe connected but non convex sets, without having to resort to complex heuristics for building (partial) disjunctive completions.

We will use a motivating example throughout this article, which is a possible implementation of the function `memcpy` which copies exactly the first `n` characters of the string buffer `src` to `dst`:

M. Alpuente and G. Vidal (Eds.): SAS 2008, LNCS 5079, pp. 189–204, 2008.

```
int i := 0;
unsigned int n, p, q;
string dst[p], src[q];
assert p >= n && q >= n;
while i <= n-1 do
  dst[i] := src[i];
  i := i+1;
done;
```

In string analysis, precise invariants over the length of the strings are needed to ensure the absence of string buffer overflows (see [3,4]). Without any information on n, no non-disjunctive string analysis is able to determine any precise invariant about the resulting length of the string len_dst (for instance, using classical polyhedra, we only get $\texttt{len\_dst} \geq 0$). Indeed, two cases have to be distinguished: (i) either n is strictly smaller than the source length len_src, so that only non-null characters are copied into dst, hence $\texttt{len\_dst} \geq \texttt{n}$, (ii) or $\texttt{n} \geq \texttt{len\_src}$ and the null terminal character of src will be copied into dst, thus $\texttt{len\_dst} = \texttt{len\_src}$. With our non-disjunctive analysis, we are able to infer automatically the invariant $\min(\texttt{len\_src}, \texttt{n}) = \min(\texttt{len\_dst}, \texttt{n})$, which exactly encodes the disjunction of the two cases. Besides, even with a disjunctive analysis (for example using trace partitioning [5]), it would be very complex to automatically determine the disjunction of the cases (i) and (ii), because it intrinsically relies on semantic information on strings.

Contents. Max-plus polyhedra are not new (see "Related Work" below) but have not been used yet in static analysis by abstract interpretation, and have been introduced for entirely different reasons: Section 2 is an introduction to the required results in the field. As for classical polyhedra, max-plus polyhedra can be presented both as a system of inequalities (constraints), or by a set of vertices and rays (generators), see Sections 2.2 and 2.3. But the underlying algorithms are quite different because of the structure of the max-plus algebra. For instance, unlike in classical algebra, systems of equality constraints and of inequality constraints are equivalent in $\mathbb{R}_{\max}$. In particular, the resulting abstract domains (max-plus analogues of Karr's [6] and Cousot-Halbwachs' [7]) have the same complexity. Theorem 1 is new: it gives an upper bound on the complexity of the resolution of a linear equation in the max-plus algebra, which is the cornerstone of the algorithm allowing to convert representations by (in)equalities to system of generators, and vice versa. We then prove in Sect. 2.4 that max-plus polyhedra subsume intervals and zones.

The abstraction and its semantics for a simple imperative language in terms of max-plus polyhedra are given in Section 3. It is able to infer both max and min invariants, and contains the abstract domain of octagons. We discuss linearization methods (in the sense of [2]) in Section 3.3 for abstracting in a precise manner non-linear max-plus expressions and assignments. We end up by describing some practical applications of this static analysis on strings and arrays, in Section 4. We also give a set of benchmarks based on the current implementation we made of the method.

Related Work. The max-plus analogues of convex sets were introduced by K. Zimmermann [8], who established an analogue of the separation theorem. Max-plus convex cones have been studied in idempotent analysis, after the work of

Maslov [9]. They also arise in the analysis of discrete event systems [10,11]. They have appeared recently in relation with tropical geometry and phylogenetic analysis [12]. See [13,12,14,15,16,17,18] for more background and recent developments.

Related work in abstract interpretation [19] include work on zones [1], octagons [2], (classical) polyhedra [7], disjunctive analysis [5,20,21]. An application of semirings (such as max-plus) has been made to static analysis by abstract interpretation in [22], although with different techniques and for different applications (timing behavior).

2 Max-Plus Polyhedra

2.1 The Max-Plus Semiring

The *max-plus semiring* $\mathbb{R}_{\max}$ is defined as the set $\mathbb{R} \cup \{-\infty\}$, equipped with the addition $x \oplus y := \max(x, y)$ and the multiplication $x \otimes y := x + y$. The additive law $\oplus$ is associative, commutative, and has a zero element $\mathbb{0} := -\infty$. The multiplicative law $\otimes$ is associative with a unit element $\mathbb{1} := 0$. Besides, the zero element $\mathbb{0}$ is absorbing, *i.e.* for any $x \in \mathbb{R}_{\max}$, $\mathbb{0} \otimes x = x \otimes \mathbb{0} = \mathbb{0}$. The semiring $\mathbb{R}_{\max}$ differs from a ring in that the elements are not necessarily invertible w.r.t the addition. An order $\preceq$ can be defined on $\mathbb{R}_{\max}$ by $x \preceq y \Leftrightarrow x \oplus y = y$. It coincides with the usual order on $\mathbb{R} \cup \{-\infty\}$.

The n-fold Cartesian product $\mathbb{R}^n_{\max}$ may be thought of as a space of vectors, or as an affine space of points. It can be endowed with the component-wise addition: if $\mathbf{u}, \mathbf{v}$ are two vectors in $\mathbb{R}^n_{\max}$, $\mathbf{u} \oplus \mathbf{v}$ denotes the vector whose ith component is the sum $\mathbf{u}_i \oplus \mathbf{v}_i$ of the ith components of $\mathbf{u}$ and $\mathbf{v}$. Similarly, the multiplication $\lambda\mathbf{u}$ of the vector $\mathbf{u}$ by a scalar $\lambda \in \mathbb{R}_{\max}$ is the vector of components $\lambda \otimes \mathbf{u}_i$.

Matrix operations are defined as well, by using max-plus addition and multiplication in the classical operations on matrices. Finally, given two subsets S_1 and S_2 of $\mathbb{R}^n_{\max}$, the max-plus *Minkowski sum* $S_1 \oplus S_2$ is defined as the set $\{\mathbf{x} \oplus \mathbf{y} \mid (\mathbf{x}, \mathbf{y}) \in S_1 \times S_2\}$.

2.2 Definition of Max-Plus Polyhedra Using Systems of Generators

Max-plus Convex Sets and Cones. A vector $\mathbf{x} \in \mathbb{R}^n_{\max}$ is a *max-plus linear combination* of the vectors $\mathbf{v}^1, \dots, \mathbf{v}^p \in \mathbb{R}^n_{\max}$ if $\mathbf{x} = \alpha_1\mathbf{v}^1 \oplus \dots \oplus \alpha_p\mathbf{v}^p$ for some scalars $\alpha_1, \dots, \alpha_p \in \mathbb{R}_{\max}$. The point $\mathbf{x}$ is a *max-plus convex combination* of the points $\mathbf{v}^1, \dots, \mathbf{v}^p$ if it can be written in the previous form, with the additional requirement that $\alpha_1 \oplus \dots \oplus \alpha_p = \mathbb{1}$. In the sequel, all max-plus convex or linear combinations will concern *finite* families.

A subset of $\mathbb{R}^n_{\max}$ is a *max-plus cone* if it contains all the max-plus linear combinations of its elements. Such cones may be thought of as the analogues of vector spaces or modules when the field or ring of scalars is replaced by the max-plus semiring. Hence, they have been studied under the names of *idempotent spaces* in [9] or *semimodules* in [13]. In the max-plus setting, positivity constraints

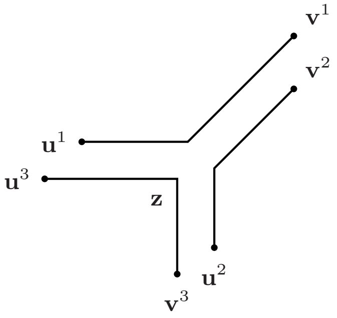

Fig. 1. The three kinds of generic max-plus segments in $\mathbb{R}^2_{\max}$

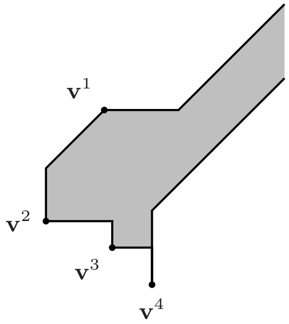

Fig. 2. A max-plus polyhedron in $\mathbb{R}^2_{\max}$ (the black border of the shape is included)

are implicit, since any scalar $\alpha \in \mathbb{R}_{\max}$ satisfies $\alpha \succeq \mathbb{0}$. For this reason, max-plus cones share many of the properties of classical convex cones.

If W is a subset of $\mathbb{R}^n_{\max}$, we denote by $\mathsf{cone}(W)$ the max-plus cone generated by W, which consists of the max-plus linear combinations of the elements of W. A max-plus cone is *finitely generated* if it can be written as $\mathsf{cone}(W)$ for some finite subset W of $\mathbb{R}^n_{\max}$. In particular, a max-plus cone generated by a (non-zero) vector is a *ray*. Such a vector is a *representative* of the ray that it generates.

Similarly, a subset of $\mathbb{R}^n_{\max}$ is a *max-plus convex* set if it contains all the max-plus convex combinations of its elements. In general, max-plus convex sets are not convex in the classical sense. We denote by $\mathsf{co}(V)$ the max-plus convex hull of V, which consists of the max-plus convex combinations of the elements of V. A set of the form $\mathsf{co}(V)$ for some finite set V is a *max-plus polytope.*

Max-plus Polyhedra. Polyhedra can be classically defined in two ways, either in terms of constraints (intersections of finitely many affine half-spaces), or in terms of vertices and rays (Minkowski sums of a polytope and of a finitely generated cone). The Minkowski-Weyl theorem proves both definitions equivalent. For the moment, let us adopt the second approach, and define a *max-plus polyhedron* to be a set of the form $P = \mathsf{co}(V) \oplus \mathsf{cone}(W)$, where V, W are finite subsets of $\mathbb{R}^n_{\max}$. The sets V and W constitute a *system of generators* of P.

Figure 2 depicts an unbounded max-plus polyhedron in $\mathbb{R}^2_{\max}$. The reader may check on the figure that this is a max-plus convex set, because it contains any segment joining two of its points (see Fig. 1 for the three kinds of max-plus segments in $\mathbb{R}^2_{\max}$).

Representing Max-plus Polyhedra by Max-plus Cones. We can represent max-plus polyhedra of $\mathbb{R}^n_{\max}$ as projections of finitely generated max-plus convex cones of $\mathbb{R}^{n+1}_{\max}$, by taking "homogeneous coordinates", as in the classical case.

Formally, if $P = \mathsf{co}(V) \oplus \mathsf{cone}(W)$, where V, W are finite subsets of $\mathbb{R}^n_{\max}$, let Z denote the subset of $\mathbb{R}^{n+1}_{\max}$ consisting of the vectors of the form $(\mathbf{v}, \mathbb{1})$ with $\mathbf{v} \in V$

or $(\mathbf{w}, \mathbb{0})$ with $\mathbf{w} \in W$, and consider the max-plus convex cone $C := \mathsf{cone}(Z)$. It is easily seen that $P = \{\mathbf{x} \in \mathbb{R}^n_{\max} \mid (\mathbf{x}, \mathbb{1}) \in C\}$. Conversely, let Z' denote any finite subset of $\mathbb{R}^{n+1}_{\max}$ such that $C = \mathsf{cone}(Z')$. After multiplying (in the max-plus sense) every element of Z' by a non-$\mathbb{0}$ scalar, we may assume that the last coordinate of every element of Z' is either $\mathbb{0}$ or $\mathbb{1}$. Then, it can be checked that $P = \mathsf{co}(V') \oplus \mathsf{cone}(W')$ where $V' = \{\mathbf{v} \mid (\mathbf{v}, \mathbb{1}) \in Z'\}$ and $W' = \{\mathbf{v} \mid (\mathbf{v}, \mathbb{0}) \in Z'\}$. This is a special case of the general correspondence between max-plus convex sets and max-plus cones which is discussed in [18].

Hence, representing max-plus polyhedra reduces to representing finitely generated max-plus cones. In the sequel, we will suppose that the representation of a max-plus polyhedron of $\mathbb{R}^n_{\max}$ by generators is a finite system G of generators of the associated max-plus cone of $\mathbb{R}^{n+1}_{\max}$. This avoids the distinction between vertices and ray representatives which may sometimes complicate the formalism, without loss of generality.

Membership to a Finitely Generated Max-plus Cone. Let G denote a set of p vectors of $\mathbb{R}^{n+1}_{\max}$. Testing whether a vector $\mathbf{x}$ is in $\mathsf{cone}(G)$ is equivalent to determine whether the system of equations $G\mathbf{y} = \mathbf{x}$ admits a solution. We use here the same notation, G for the matrix the columns of which are the elements of the generating set. The equation $G\mathbf{y} = \mathbf{x}$ may not have a solution, but the inequality $G\mathbf{y} \preceq \mathbf{x}$ always does. Besides, if it is interpreted in the completion of the max-plus semiring, it admits a maximal solution, denoted by $G \setminus \mathbf{x}$ and given by the following residuation formula: $(G \setminus \mathbf{x})_j := \min_{1 \leq i \leq n+1}(\mathbf{x}_i - G_{ij})$, with the convention $-\infty + \infty = +\infty$. If G has no column identically $-\infty$, $G \setminus \mathbf{x}$ belongs to $\mathbb{R}^p_{\max}$ for all $\mathbf{x} \in \mathbb{R}^{n+1}_{\max}$. It follows that $G\mathbf{y} = \mathbf{x}$ has a solution $\mathbf{y} \in \mathbb{R}^p_{\max}$ if and only if $G(G \setminus \mathbf{x}) = \mathbf{x}$, a test which can be done in $O(np)$ operations. More details can be found in [17].

Minimal Systems of Generators. The representation of P by a system of generators is not unique, but as for classical polyhedra, there is a minimal representation, involving sets of vectors having certain extremality properties in the associated cone C. A vector $\mathbf{w}$ is an *extreme generator* of a max-plus cone C if $\mathbf{w} \in C$, and $\mathbf{w}$ cannot be written as the max-plus sum of two vectors of C that are both different from it. Then, the set of scalars multiples of $\mathbf{w}$ is an *extreme ray* of C. It is known that a finitely generated max-plus cone is generated by its extreme rays. It follows that C has a non redundant generating family, which is unique up to a normalization of its elements, obtained by selecting one representative in each extreme ray of C. This family forms a minimal representation of P by generators. The reader can refer to [18,17] for recent accounts and refinements of this result.

To compute the extreme rays of C, we start from any generating system G, assuming that it contains no proportional vectors, and we eliminate any vector of the family which is a max-plus linear combination of the other ones (see the previous paragraph). If G consists of p generators, this can be done in $O(n \times p^2)$ operations. Some additional algorithmic information can be found in [17].

As in the classical case, we can find max-plus polyhedra of $\mathbb{R}^2_{\max}$ with an arbitrarily large number of extreme points.

2.3 Definition of Max-Plus Polyhedra by Systems of Constraints

An important similarity of max-plus polyhedra with classical ones is that they can be equivalently defined as the solutions of systems of constraints. Each constraint consists of an inequality of the form $\mathbf{a}\mathbf{x} \oplus b \succeq \mathbf{c}\mathbf{x} \oplus d$, where $\mathbf{x} \in \mathbb{R}_{\max}^n$, $\mathbf{a}, \mathbf{c} \in \mathbb{R}_{\max}^{1\times n}$ and $b, d \in \mathbb{R}_{\max}$.

However, in $\mathbb{R}_{\max}^n$, systems of equality and inequality constraints are equivalent. Indeed, any inequality can be written as an equality since $y \succeq z \Leftrightarrow y = y \oplus z$. As a consequence, systems of inequality and equality constraints have the same expressiveness, and in particular, inferring invariants involving equality constraints is as difficult as inferring inequality invariants. We chose to use here systems of equality constraints.

Solutions of a System of Constraints. The solutions of a homogeneous system of equations $A\mathbf{x} = B\mathbf{x}$, were first studied in [23]. In particular, the following proposition was proven (see also [10]):

Proposition 1. *The solutions of the homogeneous system* $A\mathbf{x} = B\mathbf{x}$ *of* $\mathbb{R}_{\max}^n$, *where* $A, B \in \mathbb{R}_{\max}^{s\times n}$, *form a finitely generated max-plus cone.*

More generally, a non-homogeneous system $A\mathbf{x} \oplus \mathbf{b} = C\mathbf{x} \oplus \mathbf{d}$ can be associated to the homogeneous system $\begin{pmatrix} A \; \mathbf{b} \end{pmatrix} \mathbf{z} = \begin{pmatrix} C \; \mathbf{d} \end{pmatrix} \mathbf{z}$ of $\mathbb{R}_{\max}^{n+1}$. Then, the solutions of the former are given by the first n coordinates of the solutions $\mathbf{z}$ verifying $\mathbf{z}_{n+1} = \mathbb{1}$ of the latter. Using the equivalence between max-plus polyhedra and cones established in Sect. 2.2, the following statement holds:

Corollary 1. *The solutions of the system of equations* $A\mathbf{x} \oplus \mathbf{b} = C\mathbf{x} \oplus \mathbf{d}$*, where* $A, B \in \mathbb{R}_{\max}^{s\times n}$ *and* $\mathbf{b}, \mathbf{d} \in \mathbb{R}_{\max}^s$*, form a max-plus polyhedron of* $\mathbb{R}_{\max}^n$.

In particular, a representation of the solution polyhedron can be obtained by computing a minimal system of generators of the cone of solutions of $\begin{pmatrix} A \; \mathbf{b} \end{pmatrix} \mathbf{z} = \begin{pmatrix} C \; \mathbf{d} \end{pmatrix} \mathbf{z}$. This is why we only consider homogeneous systems $E\mathbf{z} = F\mathbf{z}$ in $\mathbb{R}_{\max}^{n+1}$ for the rest of the section.

First, let us consider the case in which the system of constraints is reduced to one equation $\mathbf{e}\mathbf{z} = \mathbf{f}\mathbf{z}$, where $\mathbf{e}, \mathbf{f} \in \mathbb{R}_{\max}^{1\times(n+1)}$. We denote by $(\epsilon^i)_{1\leq i\leq n+1}$ the max-plus analogue of a canonical basis in $\mathbb{R}_{\max}^{n+1}$, *i.e.* $\epsilon_i^i = \mathbb{1}$ and $\epsilon_j^i = \mathbb{0}$ for $j \neq i$. It can be shown that the vectors $(\mathbf{f}_j \epsilon^i) \oplus (\mathbf{e}_i \epsilon^j)$, where $\mathbf{e}_i \succeq \mathbf{f}_i$ and $\mathbf{e}_j \preceq \mathbf{f}_j$, form a generating system of the solution cone.

The general case of a system of s equations can be solved by induction on s, following the method by *elimination* proposed in [23]:

- when $s = 0$, there is no constraint, so that the family ϵ^i form a generating system of the solution.
- if $s \geq 1$, let $E'\mathbf{z} = F'\mathbf{z}$ be the system of equations formed by the $(s-1)$ first equations, and $\mathbf{e}\mathbf{z} = \mathbf{f}\mathbf{z}$ the last equation of $E\mathbf{z} = F\mathbf{z}$. If $G' = (\mathbf{g}^1, \dots, \mathbf{g}^p)$ is a system of generators of $E'\mathbf{z} = F'\mathbf{z}$, then we have $E\mathbf{z} = F\mathbf{z}$ if and only if there exists $\mathbf{y} \in \mathbb{R}_{\max}^p$ such that $(\mathbf{e}G')\mathbf{y} = (\mathbf{f}G')\mathbf{y}$ and $\mathbf{z} = G'\mathbf{y}$ (G' being seen as a matrix whose columns are the $\mathbf{g}^i$). The equation $(\mathbf{e}G')\mathbf{y} = (\mathbf{f}G')\mathbf{y}$

of $\mathbb{R}^p_{\max}$ can be solved using the method given above. If $H = (\mathbf{h}^1, \dots, \mathbf{h}^q)$ is a generating system of its solutions, the vectors $G'\mathbf{h}^1, \dots, G'\mathbf{h}^q$ form a system of generators of the solutions of $E\mathbf{z} = F\mathbf{z}$.

Theorem 1. *A minimal system of generators of the solutions of the s equations $E\mathbf{z} = F\mathbf{z}$ in $\mathbb{R}^{n+1}_{\max}$ can be computed in $O(n \times s \times c^4_{n+1,s})$ operations, where $c_{n+1,s}$ is the maximal number of generators of the set of solutions of a system of s equations in $\mathbb{R}^{n+1}_{\max}$.*

As a consequence, the solving algorithm is polynomial in the maximal number of the generators which may arise. In comparison, the cost of Chernikova's algorithm [24], which allows to convert the representation of a classical convex polyhedron by inequalities into an equivalent representation by generators, is quadratic in the number of generators, when it is executed on hypercubes of $\mathbb{R}^n$. However, it seems that the problem of solving max-plus (in)equalities is intrinsically more complex, as even an inequality $\mathbf{ex} \succeq \mathbf{fx}$ of $\mathbb{R}^n_{\max}$ generates a quadratic number of generators in n, while this number is linear in the classical case. Moreover, most implementations of the domain of classical convex polyhedra now involve efficient tests based on the number of inequalities saturated by the computed generators, in order to eliminate redundant ones (see, for instance, [25]). In contrast, no such properties relative to the saturation of max-plus inequalities are yet proven.

Bounding the maximal number of generators $c_{n+1,s}$ is an interesting combinatorial problem. An exponential bound is given in [26], where it is shown that $c_{n+1,s} \leq s \times \left(n^2/3 + n + 1\right)^s$, but the optimal bound is not known. Future work could focus on the comparison of $c_{n+1,s}$ with its classical analogue for convex polyhedra, which is in $O(s^n)$. But for now, all we can say is that the solution of a system can be computed in $O(s^2 \times n^{8s+1})$ operations.

Example 1. With $n = 2$, let us consider the system of two equations $\mathbf{x}_1 \oplus 1 = \mathbf{x}_1$ and $\mathbf{x}_2 \oplus 1 = \mathbf{x}_2$. It corresponds to the system $\mathbf{x}_1 \geq 1$ and $\mathbf{x}_2 \geq 1$. The associated homogeneous system is $\mathbf{z}_1 \oplus 1\mathbf{z}_3 = \mathbf{z}_1$ and $\mathbf{z}_2 \oplus 1\mathbf{z}_3 = \mathbf{z}_2$. Solving the first equation yields a generating family G consisting of the vectors $\mathbf{g}^1 = [1; -\infty; 0]$, $\mathbf{g}^2 = [-\infty; 0; -\infty]$, $\mathbf{g}^3 = [0; -\infty; -\infty]$. Multiplying the left and the right members of the second equation by the matrix G yields the equation $1\mathbf{y}_1 \oplus \mathbf{y}_2 = \mathbf{y}_2$, whose generating family H consists of the vectors $\mathbf{h}^1 = [-\infty; 0; -\infty]$, $\mathbf{h}^2 = [-\infty; -\infty; 0]$, and $\mathbf{h}^3 = [0; 1; -\infty]$. The vectors $G\mathbf{h}^1$, $G\mathbf{h}^2$, and $G\mathbf{h}^3$ form the family $([-\infty; 0; -\infty], [0; -\infty; -\infty], [1; 1; 0])$, which is obviously minimal. It represents a max-plus polyhedron with one vertex $[1; 1]$, and two rays with representatives $[0; -\infty]$ and $[-\infty; 0]$.

From Systems of Generators to Systems of Constraints. A system of generators can be converted to a system of constraints describing the same max-plus polyhedron. Given a max-plus polyhedron P provided with a system of generators G, the set $P^\perp$ of constraints $\mathbf{ax} \oplus b = \mathbf{cx} \oplus d$ verified by the polyhedron P is a cone of $(\mathbb{R}^{1\times n}_{\max} \times \mathbb{R}_{\max})^2$ (each constraint $\mathbf{ax} \oplus b = \mathbf{cx} \oplus d$ being represented by the pair $((\mathbf{a}, b), (\mathbf{c}, d))$). Moreover, it can be shown that a constraint is verified by the polyhedron if and only if it is verified by all its generators. Hence, we have $P^{\perp} = \{((\mathbf{a}, b), (\mathbf{c}, d)) \in (\mathbb{R}_{\max}^{1\times n} \times \mathbb{R}_{\max})^2 \mid \forall i.\, \begin{pmatrix}\mathbf{a} & b\end{pmatrix} \mathbf{g}^i = \begin{pmatrix}\mathbf{c} & d\end{pmatrix} \mathbf{g}^i\}$, *i.e.*

$$P^{\perp} = \left\{((\mathbf{a}, b), (\mathbf{c}, d)) \in (\mathbb{R}_{\max}^{1\times n} \times \mathbb{R}_{\max})^2 \mid {}^tG \begin{pmatrix}{}^t\mathbf{a} \\ b\end{pmatrix} = {}^tG \begin{pmatrix}{}^t\mathbf{c} \\ d\end{pmatrix}\right\} \enspace ,$$

where ${}^t\cdot$ is the matrix transposition operator.

As a consequence, a minimal system of generators of the cone $P^{\perp}$ can be computed by using the algorithm presented in the previous paragraph, with a complexity in $O(p \times n \times c_{2n+2,p}^4)$. Then, the system of constraints formed by the generators of the cone $P^{\perp}$ can be shown to be a representation of the max-plus polyhedron P under the form of constraints. As for generators, we are interested in manipulating minimal representations by systems of constraints. Here, the computed constraints form a minimal generating family of the cone $P^{\perp}$, which is a good point. However, it may not be a minimal system of constraints, *i.e.* some constraints are possibly redundant. This is basically due to the fact that the cone $P^{\perp}$ represents a set of equations closed by symmetry, reflexivity, and transitivity. Extracting a minimal system of constraints would have a major drawback: it would be very costly since we would have to compare corresponding max-plus polyhedra, hence to convert many systems of constraints to generators (see the definition of the abstract partial order in Sect. 3.1). Moreover, in our experimentations (see Sect. 4), the size taken by systems of constraints is negligible, this is why minimal generating families of the cone $P^{\perp}$ are satisfactory.

2.4 Max-Plus Polyhedra and Zones

Interval and zone constraints are obviously particular forms of max-plus systems of constraints described in Sect. 2.3.

We next show that a representation by intervals and zones can be extracted from a system of generators of a max-plus polyhedron. Recall that if $A \in \overline{\mathbb{R}}_{\max}^{n\times p}$, the residuated matrix [27] A/A is given by $(A/A)_{ij} = \min_{1\leq k\leq p} A_{ik} - A_{jk}$ (with $-\infty + \infty = +\infty$). Observe that if G is a minimal system of generators of $\mathbb{R}_{\max}^{n+1}$, and if G (seen as a matrix) does not have a row consisting only of $-\infty$ entries, then $G/G \in \mathbb{R}_{\max}^{(n+1)\times(n+1)}$. Using the fact that $\mathsf{cone}(G/G)$ (*i.e.* the cone generated by the column of G/G) is the least sublattice containing $\mathsf{cone}(G)$ [28], we deduce the following theorem:

Theorem 2. *The cone* $\mathsf{cone}(G/G)$ *coincides with the zone of* $\mathbb{R}_{\max}^n$ *defined by:*

$$\begin{aligned} \forall i, j \in \{1, \dots, n\},\ \mathbf{x}_i - \mathbf{x}_j \geq (G/G)_{ij} \enspace , \\ \forall i \in \{1, \dots, n\},\ (G/G)_{i,n+1} \leq \mathbf{x}_i \leq -(G/G)_{n+1,i} \enspace . \end{aligned}$$

Moreover, if G has no row consisting only of $-\infty$ entries, then the smallest zone containing the $\mathsf{cone}(G)$ *is given by* $\mathsf{cone}(G/G)$*.*

During the reduction to zones, the rows of G consisting only of $-\infty$ entries can be simply not considered, since they correspond to coordinates $\mathbf{x}_i$ constant equal to $-\infty$. Hence, Th. 2 provides an effective algorithm to convert max-plus polyhedra to zones.

3 Abstract Semantics

Let us consider a set Var of n distinct variables x_i. Our abstraction consists in representing sets of environments $\sigma : \mathsf{Var} \rightarrow \mathbb{R}$ by max-plus polyhedra P of $\mathbb{R}^{2n}_{\max}$, *i.e.* either by minimal systems G of generators of $\mathbb{R}^{2n+1}_{\max}$, or by systems S of max-plus equality constraints $A\mathbf{x} \oplus \mathbf{b} = C\mathbf{x} \oplus \mathbf{d}$ of $\mathbb{R}^{2n}_{\max}$.[1] Intuitively, the n first dimensions of the polyhedra represent the variables x_i, while the n last ones represent their opposite $-\mathsf{x}_i$. The concretization of max-plus polyhedra is defined equivalently according to their representation:

$$\gamma(P) := \{\sigma \mid (\sigma(\mathsf{x}_1), \dots, \sigma(\mathsf{x}_n), -\sigma(\mathsf{x}_1), \dots, -\sigma(\mathsf{x}_n), \mathbb{1}) \in \mathsf{cone}(G)\} \enspace ,$$

$$\text{or } \gamma(P) := \left\{ \sigma \mid \begin{array}{l} A(\sigma(\mathsf{x}_1), \dots, \sigma(\mathsf{x}_n), -\sigma(\mathsf{x}_1), \dots, -\sigma(\mathsf{x}_n)) \oplus \mathbf{b} \\ \quad = C(\sigma(\mathsf{x}_1), \dots, \sigma(\mathsf{x}_n), -\sigma(\mathsf{x}_1), \dots, -\sigma(\mathsf{x}_n)) \oplus \mathbf{d} \end{array} \right\} \enspace .$$

As mentioned in Sect. 1, this allows to infer invariants of the form $\max(\lambda_0, \mathsf{x}_1 + \lambda_1, \dots, \mathsf{x}_n + \lambda_n) \leq \max(\mu_0, \mathsf{x}_1 + \mu_1, \dots, \mathsf{x}_n + \mu_n)$ and $\min(\lambda'_0, \mathsf{x}_1 + \lambda'_1, \dots, \mathsf{x}_n + \lambda'_n) \leq \min(\mu'_0, \mathsf{x}_1 + \mu'_1, \dots, \mathsf{x}_n + \mu'_n)$.[2]

3.1 Order-Theoretic Operators

Partial Order. An abstract partial order can be defined on max-plus polyhedra by comparing systems of generators. Given two max-plus polyhedra P and Q represented by the systems of generators G and H respectively, we have $P \sqsubseteq Q$ if and only if for any $\mathbf{g} \in G$, $\mathbf{g} \in \mathsf{cone}(H)$ (or equivalently, $H(H \setminus \mathbf{g}) = \mathbf{g}$). Hence, the concretization γ can be shown to be monotonic. The complexity of the evaluation of $G \sqsubseteq H$ is $O(n \times p \times q)$, p and q being the cardinality of the families G and H. Note that we can define equivalently $\sqsubseteq$ by using a representation of Q by a system of constraints $A\mathbf{x} \oplus \mathbf{b} = C\mathbf{x} \oplus \mathbf{d}$, and testing whether $\begin{pmatrix} A\ \mathbf{b} \end{pmatrix} \mathbf{g} = \begin{pmatrix} D\ \mathbf{d} \end{pmatrix} \mathbf{g}$ for any $\mathbf{g}$ in G. Then, the operation has a complexity in $O(n \times p \times t)$, where t is the number of constraints in the system of Q.

Joining Max-plus Polyhedra. Given two systems of generators G and H, an abstract join operator $\sqcup$ can be defined as the minimal system G of generators extracted from the family $G \cup H$. If p and q are the cardinality of the families G and H, the union can be performed in $O(n \times (p + q)^2)$. It can be shown to be a sound join operator, and even the best possible one.

Intersection. By duality, an abstract intersection operator can be naturally defined on two polyhedra by concatenating the systems of constraints representing the polyhedra. Equivalently, we can define the intersection operator when one polyhedron is represented by a minimal system G of generators while the other is

[1] Each system S is represented by a minimal generating family of the cone $P^{\perp}$.

[2] The latter are computed under the form $\max(-\lambda'_0, -\mathsf{x}_1 - \lambda'_1, \dots, -\mathsf{x}_n - \lambda'_n) \geq \max(-\mu'_0, -\mathsf{x}_1 - \mu'_1, \dots, -\mathsf{x}_n - \mu'_n)$. In fact, the abstract domain is able to infer more general invariants of the form $\max(\lambda_0, \mathsf{x}_1 + \lambda_1, \dots, \mathsf{x}_n + \lambda_n, -\mathsf{x}_1 - \lambda'_1, \dots, -\mathsf{x}_n - \lambda'_n) \leq \max(\mu_0, \mathsf{x}_1 + \mu_1, \dots, \mathsf{x}_n + \mu_n, -\mathsf{x}_1 - \mu'_1, \dots, -\mathsf{x}_n - \mu'_n)$.

represented by a system of constraints $A\mathbf{x}\oplus\mathbf{b} = C\mathbf{x}\oplus\mathbf{d}$. Indeed, it can be shown that solving the homogeneous system of constraints $\big(\big(A\ \mathbf{b}\big)\,G\big)\,\mathbf{z} = \big(\big(C\ \mathbf{d}\big)\,G\big)\,\mathbf{z}$ by replacing the family ϵ^i by the $\mathbf{g}^i$ in the initial step, exactly yields a minimal system of generators of the intersection. In that case, the complexity of the intersection is $O(t \times p \times c^4_{p,t})$, where p is the cardinality of G and t the number of constraints of the system $A\mathbf{x} \oplus \mathbf{b} = C\mathbf{x} \oplus \mathbf{d}$.

The intersection operator allows to handle conditional program statements of the form $\pm\mathsf{x} \leq k$ or $\pm\mathsf{x} \leq \pm\mathsf{y} + k$. Other conditions can be either ignored (which is sound), or handled using a linearization (see Sect. 3.3).

Widening. If $n \geq 2$, infinite ascending chains of max-plus polyhedra of $\mathbb{R}^n_{\max}$ can be built. As a result, a widening operator is defined to enforce convergence. It follows the initial definition of the widening over classical convex polyhedra [7]. Formally, if two max-plus polyhedra P and Q are respectively represented by a system of constraints and a minimal system G of generators, the max-plus polyhedron $P \,\nabla\, Q$ is defined as the system of constraints formed by the constraints $\mathbf{ax} \oplus b = \mathbf{cx} \oplus d$ of P which are also verified by Q, *i.e.* $\forall \mathbf{g} \in G.\ \big(\mathbf{a}\ b\big)\,\mathbf{g} = \big(\mathbf{c}\ d\big)\,\mathbf{g}$.

As for the widening defined in [7], the result of the widening depends on the system of constraints chosen to represent the max-plus polyhedra. In [29], the definition of the widening over classical convex polyhedra was improved to overcome this problem, by adding to the result the constraints of Q which are equivalent to some constraints of P. They can be discovered either by checking whether they can replace a constraint of P without changing the represented polyhedron, or by considerations on the saturation of some linear inequalities by generators [30]. In max-plus algebra, the former method is particularly costly since it requires to convert some systems of constraints to generators. Moreover, we do not have yet any proof that the latter approach could be applied, because the equivalence between inequalities and equalities in the max-plus algebra makes the problem harder. For that reason, the actual definition of the widening operator is not fully satisfactory. Nevertheless, the experimentations are very encouraging since the widening allows to exactly infer the expected invariant for each of our examples (see Sect. 4).

Reduction. A system of octagonal constraints (*i.e.* of the form $\pm\mathsf{x}_i \pm \mathsf{x}_j \geq c_{ij}$) can be extracted from any representation by generators using Sect. 2.4. These constraints can be then refined using the closure algorithm of octagons [2]. The resulting octagon can be seen as a max-plus polyhedron over the variables $\pm\mathsf{x}_i$. Intersecting it with the initial max-plus polyhedron yields a smaller abstract element w.r.t $\sqsubseteq$, but which represents the same set of concrete states. This defines a reduction operator, which allows our representation to be more precise than the abstract domain of octagons. Intuitively, the reduction operator enables a communication between the variables x_i and $-\mathsf{x}_i$. Note that, as for octagons, the convergence property of the widening operator may not hold if the reduction operator is applied to widened max-plus polyhedra.

3.2 Assignments

Max-plus Assignments. A *max-plus assignment* is an assignment of the form $\mathbf{x}_i \leftarrow \left(\oplus_{j=1}^{n} m_j \mathbf{x_j}\right) \oplus m_{n+1}$, for some $1 \leq i \leq n$, and $m_1, \ldots, m_n, m_{n+1} \in \mathbb{R}_{\max}$. In particular, max-plus assignments include operations $\mathbf{x}_i \leftarrow m_{n+1}$ and $\mathbf{x}_i \leftarrow \mathbf{x}_j + m_j$ (where + is the classical addition). The abstract operator for max-plus assignments consists in multiplying a minimal system of generators by a matrix M corresponding to the assignment: M coincides with the max-plus identity matrix, except that its ith row is replaced by $m_1 \ldots m_{n+1}$. It then remains to extract a minimal system of generators from the result. This operator can be shown to be sound. Its cost is $O(n \times (n^2 + p^2))$. It can be easily generalized to handle parallel max-plus assignments, without changing the complexity. Thus, program assignments of the form $\mathsf{x} \leftarrow k$ and $\mathsf{x} \leftarrow \pm\mathsf{y} + k$ are implemented as parallel max-plus assignments on the dimensions of variables x and $-\mathsf{x}$.

Non-deterministic Assignments. A non-deterministic assignment $\mathbf{x}_i \leftarrow ?$ can be handled by adding a representative $\mathbf{h}$ of the ray formed by the ith axis (*e.g.* $\mathbf{h}_i = \mathbb{1}$ and $\mathbf{h}_j = \mathbb{0}$) to a minimal system of generators, and then extracting a new minimal system from the resulting family. This defines a sound operator, whose cost is in $O(n \times p^2)$ (p being the size of the initial system of generators).

Some assignments which do not belong to the classes previously discussed can be linearized (see Sect. 3.3). Other can be soundly treated as non-deterministic.

3.3 Linearization

In this section, we indicate how to interpret general linear assignments (as in classical linear algebra), *i.e.* non-linear max-plus expressions. For sake of simplicity, the description is restricted to max-plus polyhedra of $\mathbb{R}^n_{\max}$, which infer information on the positive variables $\mathsf{x}_1, \ldots, \mathsf{x}_n$. A generalization to the full abstraction including the opposites $-\mathsf{x}_1, \ldots, -\mathsf{x}_n$ is straightforward.

Suppose the variables $\mathsf{x}_1, \ldots, \mathsf{x}_n$, at some control point of a program, belong to a max-plus polyhedron $P = \mathsf{co}(V) \oplus \mathsf{cone}(W)$. If $V = (\mathbf{v}^i)_i$ and $W = (\mathbf{w}^j)_j$, then for any k, $\mathsf{x}_k = \left(\bigoplus_{i=1}^{p} \alpha_i \mathbf{v}^i_k\right) \oplus \left(\bigoplus_{j=1}^{q} \beta_j \mathbf{w}^j_k\right)$, with $\bigoplus_{i=1}^{p} \alpha_i = \mathbb{1}$. In particular, $\alpha_i \preceq \mathbb{1}$ for $i \in \{1, \ldots, p\}$. More than that, we have $\alpha_i = \mathbb{1}$ for some $i \in \{1, \ldots, p\}$, hence $\bigoplus_{i=1}^{p} \alpha_i \mathbf{v}^i_k \succeq \min_{i=1}^{p} \mathbf{v}^i_k$. Hence, we can write equivalently: $\mathsf{x}_k = \mathbf{v}^0_k \oplus \bigoplus_{i=1}^{p} \alpha_i \mathbf{v}^i_k \oplus \bigoplus_{j=1}^{q} \beta_j \mathbf{w}^j_k$, with $\bigoplus_{i=1}^{p} \alpha_i = \mathbb{1}$ and $\mathbf{v}^0_j = \min_{i=1}^{p} \mathbf{v}^i_j$.

Sum. Consider now the assignment $\mathsf{x}_{n+1} \leftarrow \mathsf{x}_k + \mathsf{x}_l$ (+ is here the standard addition, *i.e.* the multiplication in the max-plus algebra), where x_{n+1} is a newly introduced variable (up to assigning x_{n+1} to a variable x_m later, and removing the $(n+1)$-th dimension). We then have:

$$\mathsf{x}_{n+1} = \mathbf{v}^0_k \mathbf{v}^0_l \oplus \bigoplus_{i=1}^{p} \alpha_i \left(\mathbf{v}^i_k \mathbf{v}^0_l \oplus \mathbf{v}^0_k \mathbf{v}^i_l\right) \oplus \bigoplus_{j=1}^{q} \beta_j \left(\mathbf{w}^j_k \mathbf{v}^0_l \oplus \mathbf{v}^0_k \mathbf{w}^j_l\right) \oplus N \ ,$$

which can be recognized as the sum of the first-order Taylor expansion (or linearization) of the function $(x, y) \mapsto xy$ (in the max-plus algebra), and a non-linear residual term N.

Let us define some new generators: $\mathbf{v}'^0 = \big(\mathbf{v}^0, \mathbf{v}^0_k\mathbf{v}^0_l\big)$, $\mathbf{v}'^i = (\mathbf{v}^i, \mathbf{v}^i_k\mathbf{v}^0_l \oplus \mathbf{v}^0_k\mathbf{v}^i_l)$ for $i = 1, \ldots, p$, and $\mathbf{w}'^i = (\mathbf{w}^i, \mathbf{w}^i_k\mathbf{v}^0_l \oplus \mathbf{v}^0_k\mathbf{w}^i_l)$ for $i = 1, \ldots, q$. Besides, we add up a vertex, abstracting the first part of the non-linear term N: $\mathbf{v}^{p+1} = [\mathbb{0}; \ldots; \mathbb{0}; \bigoplus_{i,j=1}^{p} \mathbf{v}^i_k\mathbf{v}^j_l]$, and if $q > 0$, a ray abstracting the second part of the non-linear term N: $\mathbf{w}^{q+1} = [\mathbb{0}; \ldots; \mathbb{0}; \mathbb{1}]$. The returned system of generators represents a sound approximation of the assignment on the initial polyhedron.[3]

Multiplication by a Constant. We interpret now the assignment $\mathsf{x}_{n+1} \leftarrow a \times \mathsf{x}_k$ where a is a constant. We suppose $a \geq 0$.[4] Then we have $\mathsf{x}_{n+1} = a \times \mathbf{v}^0_k \oplus \bigoplus_{i=1}^{p}(a \times \alpha_i)(a \times \mathbf{v}^i_k) \oplus \bigoplus_{j=1}^{q}(a \times \beta_j)(a \times \mathbf{w}^j_k),$. Except in the trivial case $a = 1$, we cannot abstract very precisely this expression. Our only choice is to introduce a new vertex $\mathbf{v}^{p+1} = [\mathbb{0}; \ldots; \mathbb{0}; \bigoplus_{i=1}^{p} a \times \mathbf{v}^i_k \oplus \bigoplus_{j=1}^{q} a \times \mathbf{w}^j_k]$.

Comparison with Linearization in Octagons. Ordinarily, only $\mathsf{x} \leftarrow \pm\mathsf{y} + [a, b]$, $\mathsf{x} \leftarrow [a, b]$, or $\mathsf{x} \leftarrow \pm\mathsf{x} + [a, b]$ are interpreted exactly on octagons [2]. We claim that given $\mathsf{z} \leftarrow [a, b]$ encoded as a max-plus polyhedron, $\mathsf{x} \leftarrow \pm\mathsf{y} + \mathsf{z}$, $\mathsf{x} \leftarrow \mathsf{z}$, and $\mathsf{x} \leftarrow \pm\mathsf{x}+\mathsf{z}$ are interpreted with our linearization, exactly as an octagon would do. We also claim that linearization of assignments in the style of [31] for octagons encoded as max-plus polyhedra is in general as precise or less precise than our linearization on these max-plus polyhedra. This will be developed elsewhere.

4 Examples and Benchmarks

The abstraction defined in Sect. 3 has been implemented in an analyzer of 3 500 lines of OCaml. Our prototype only manipulates systems of generators, except in the widening steps for which conversions to constraints are needed. It has been evaluated on various programs described below.[5] Table 1 indicates the number of lines and variables of each program, the time the analyzer needs to compute invariants, and the number of generators the resulting invariants have. For each example, the memory consumption is negligible (at worst 9 Mb for `oddeven9`).

String and Array Manipulation. Our analyzer is able to infer precise invariants on the advanced string manipulating functions `memcpy` and `strncpy`. The latter copies at most `n` characters from `src` into `dst`. In particular, if the length of `src` is smaller that `n`, the remainder of `dst` is filled with null characters. For both programs, the expected final invariant $\min(\mathtt{len\_src}, \mathtt{n}) = \min(\mathtt{len\_dst}, \mathtt{n})$ is successfully discovered by our analyzer.

The program `partd` is a decrementing initialization program which fills an array with a value `c`, from the index `q` to `p` + 1. Some array analyzes [32,33] allow to infer the loop invariant $\mathbf{c}(\mathtt{i} + 1, \mathtt{q})$, which means that the array `t` contains the value `c` between the indexes `i` + 1 and `q` (both included). However,

[3] Albeit not detailed here, the case $p = 1$ can be handled in a more precise manner.

[4] The case $\mathsf{x}_{n+1} \leftarrow a \times \mathsf{x}_k$ with $a \leq 0$ rewrites into $-\mathsf{x}_{n+1} = (-a) \times \mathsf{x}_k$ with $-a \geq 0$.

[5] Source codes are available at `http://www.lix.polytechnique.fr/~allamige`.

Table 1. Analysis benchmarks on a 3 GHz Pentium with 4 Gb RAM

Program	# lines	# var.	# time (s)	# gen.
memcpy	9	6	2.37	7
strncpy	19	7	9.82	8
parti	8	3	0.008	3
partd	7	3	0.008	3
partd2	10	4	0.084	4
partd3	14	5	0.272	5
partd4	17	6	0.73	6
partd5	20	7	2.15	7
partd6	22	8	6.98	8
partd7	25	9	10.98	9
partd8	28	10	28.35	10
partd9	31	11	40.95	11
partd10	34	12	68.94	12
partd11	37	13	129.86	13
partd12	40	14	196.32	14
partd13	43	15	335.62	15
partd14	46	16	420.03	16
partd15	49	17	672.49	17
bubble3	21	7	0.016	6
oddeven3	28	7	0.012	8
oddeven4	39	9	0.06	16
oddeven5	70	11	1.13	32
oddeven6	86	13	7.76	64
oddeven7	102	15	34.42	116
oddeven8	118	17	178.42	196
oddeven9	214	19	25939.76	512

without prior information on the order of p and q, the final invariant on i with classical convex polyhedra is only $\mathtt{i} \leq \mathtt{p} \wedge \mathtt{i} \leq \mathtt{q}$. Our analyzer is able to discover the relations $\mathtt{i} = \min(\mathtt{p}, \mathtt{q})$, which is the most precise invariant. Similarly, each program partd$k$ corresponds to a sequence of k partial initializations. For each, our analyzer infers the expected invariant expressing that i is the minimum of the $k + 1$ indexes. An incrementing version of partd, parti, is also successfully analyzed.

For these examples, the widening steps (which require conversions from generators to constraints) are by far the more time consuming steps.

Sorting Algorithms. Consider now an implementation of bubble sort. We completely unfold the loop and specialize it to an array of three elements $[\mathtt{x}, \mathtt{y}, \mathtt{z}]$. The resulting sorted array is supposed to be $[\mathtt{i}, \mathtt{j}, \mathtt{k}]$. Our analyzer proves in particular that i and k are respectively the smallest and biggest elements of the three initial ones, without resorting to a heavy disjunctive analysis. In order to prove more, *i.e.* about j (we "only" get $\mathtt{j} \geq \mathtt{i}$ and $\mathtt{j} \leq \mathtt{k}$, and j is less than the maximum of any pair of entries in the input array), we would need mixed constraints with min and max (see Section 5).

Last but not least, consider the odd-even sort [34] on 2^k elements. Here we start with an array of four elements $[\mathtt{i}, \mathtt{j}, \mathtt{k}, \mathtt{l}]$, the resulting sorted array should be $[\mathtt{x}, \mathtt{y}, \mathtt{z}, \mathtt{t}]$. We find automatically in particular that x and t are the minimum and the maximum of i, j, k, and l. In fact, the max-plus constraints that are generated are quite dense. This sorting algorithm is probably of worst-case complexity for our analysis. Programs oddeven$i$ (for $i = 3$ to 9) are odd-even sorting algorithms for i elements. The analyzer proves that the last (resp. first) element of the resulting array is the maximum (resp. minimum) of the inputs. The exponentially growing complexity of the analysis is mainly due to the intersection operations (corresponding to the conditional statements). Nevertheless, one should realize that the returned invariant could only be proven before our domain by a disjunctive version of at least a zone analyzer; but for oddeven9 for instance, which consists of a sequence of 40 independent if blocks, a full partition into the potential $2^{40} \sim 10^{12}$ paths would have to be used to discover the same invariant, which is intractable both in time and memory.

5 Conclusion and Future Work

In this article, we described the first few applications of max-plus algebra to static analysis. Many improvements are yet to be discovered. Among these are improvements of the widening operator, to be applicable directly on a representation with generators and not on a constraint form, which would allow to compute invariants only by using the generator form for the whole analysis.

Existing memory manipulation analyzes could take advantage of our abstraction. For instance, it could be directly integrated in non-disjunctive array predicate abstractions (*e.g.* [33]), and help to automatically discover preconditions on C library functions [35] without disjunction. Moreover, when analyzing sorting algorithms, in order to prove that the resulting array is correctly ordered (and not infer information over its first and last elements only), one would need min-max-plus [36] invariants, generalizing our max-plus and min-plus invariants.

Last but not least, in order to deal with the intrinsic complexity of the full max-plus polyhedra, it is natural to think of generalizations of templates [37] to max-plus algebra. This is left for future work.

References

1. Miné, A.: A new numerical abstract domain based on difference-bound matrices. In: Danvy, O., Filinski, A. (eds.) PADO 2001. LNCS, vol. 2053, Springer, Heidelberg (2001)
2. Miné, A.: The octagon abstract domain. In: AST 2001 in WCRE 2001, pp. 310–319. IEEE Computer Society Press, Los Alamitos (2001)
3. Dor, N., Rodeh, M., Sagiv, M.: Cssv: towards a realistic tool for statically detecting all buffer overflows in C. In: PLDI 2003, ACM Press, New York (2003)
4. Allamigeon, X., Godard, W., Hymans, C.: Static Analysis of String Manipulations in Critical Embedded C Programs. In: Yi, K. (ed.) SAS 2006. LNCS, vol. 4134, Springer, Heidelberg (2006)
5. Rival, X., Mauborgne, L.: The trace partitioning abstract domain. ACM TOPLAS 29(5) (2007)
6. Karr, M.: Affine relationships among variables of a program. Acta Inf. 6 (1976)
7. Cousot, P., Halbwachs, N.: Automatic discovery of linear restraints among variables of a program. In: POPL 1978, Tucson, Arizona, USA. ACM Press, New York (1978)
8. Zimmermann, K.: A general separation theorem in extremal algebras. Ekonom.-Mat. Obzor. 13(2), 179–201 (1977)
9. Litvinov, G., Maslov, V., Shpiz, G.: Idempotent functional analysis: an algebraical approach. Math. Notes 69(5), 696–729 (2001); Also eprint arXiv:math.FA/0009128
10. Gaubert, S., Plus, M.: Methods and applications of (max,+) linear algebra. In: Reischuk, R., Morvan, M. (eds.) STACS 1997. LNCS, vol. 1200, Springer, Heidelberg (1997)

11. Cohen, G., Gaubert, S., Quadrat, J.P.: Max-plus algebra and system theory: where we are and where to go now. Annual Reviews in Control 23, 207–219 (1999)
12. Develin, M., Sturmfels, B.: Tropical convexity. Doc. Math. 9, 1–27 (2004)
13. Cohen, G., Gaubert, S., Quadrat, J.P.: Duality and separation theorem in idempotent semimodules. Linear Algebra and Appl. 379, 395–422 (2004)
14. Joswig, M.: Tropical halfspaces. In: Combinatorial and computational geometry. Math. Sci. Res. Inst. Publ., vol. 52. Cambridge Univ. Press, Cambridge (2005)
15. Cohen, G., Gaubert, S., Quadrat, J.P., Singer, I.: Max-plus convex sets and functions. In: Litvinov, G.L., Maslov, V.P. (eds.) Idempotent Mathematics and Mathematical Physics. Contemporary Mathematics, vol. 377, pp. 105–129. AMS (2005)
16. Gaubert, S., Katz, R.: Max-plus convex geometry. In: Schmidt, R.A. (ed.) RelMiCS/AKA 2006. LNCS, vol. 4136, pp. 192–206. Springer, Heidelberg (2006)
17. Butkovič, P., Schneider, H., Sergeev, S.: Generators, extremals and bases of max cones. Linear Algebra Appl. 421, 394–406 (2007)
18. Gaubert, S., Katz, R.: The Minkowski theorem for max-plus convex sets. Linear Algebra and Appl. 421, 356–369 (2006)
19. Cousot, P., Cousot, R.: Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In: POPL 1977, Los Angeles, California. ACM Press, New York (1977)
20. Sankaranarayanan, S., Ivancic, F., Shlyakhter, I., Gupta, A.: Static analysis in disjunctive numerical domains. In: Yi, K. (ed.) SAS 2006. LNCS, vol. 4134, pp. 3–17. Springer, Heidelberg (2006)
21. Giacobazzi, R., Ranzato, F.: Compositional Optimization of Disjunctive Abstract Interpretations. In: Riis Nielson, H. (ed.) ESOP 1996. LNCS, vol. 1058, Springer, Heidelberg (1996)
22. Sotin, P., Cachera, D., Jensen, T.: Quantitative static analysis over semirings: analysing cache behaviour for java card. In: QAPL 2006. ENTCS, vol. 1380, Elsevier, Amsterdam (2006)
23. Butkovič, P., Hegedüs, G.: An elimination method for finding all solutions of the system of linear equations over an extremal algebra. Ekonomicko-matematicky Obzor. 20(2), 203–215 (1984)
24. Chernikova, N.V.: 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)
25. Le Verge, H.: A note on Chernikova's algorithm (1992)
26. Gaubert, S., Katz, R.: External and internal representation of max-plus polyhedra. Privately circuled draft (2008)
27. Baccelli, F., Cohen, G., Olsder, G.J., Quadrat, J.P.: Synchronization and Linearity. Wiley, Chichester (1992)
28. Cohen, G., Gaubert, S., Quadrat, J.P.: Regular matrices in max-plus algebra (preprint, 2008)
29. Halbwachs, N.: Détermination Automatique de Relations Linéaires Vérifiées par les Variables d'un Programme. Thèse de $3^{\text{ème}}$ cycle d'informatique, Université scientifique et médicale de Grenoble, Grenoble, France (March 1979)
30. Halbwachs, N., Proy, Y., Roumanoff, P.: Verification of real-time systems using linear relation analysis. Formal Methods in System Design 11(2) (August 1997)
31. Miné, A.: Symbolic methods to enhance the precision of numerical abstract domains. In: Emerson, E.A., Namjoshi, K.S. (eds.) VMCAI 2006. LNCS, vol. 3855, pp. 348–363. Springer, Heidelberg (2005)
32. Gopan, D., Reps, T., Sagiv, M.: A framework for numeric analysis of array operations. SIGPLAN Not. 40(1) (2005)

33. Allamigeon, X.: Non-disjunctive Numerical Domain for Array Predicate Abstraction. In: Drossopoulou, S. (ed.) ESOP 2008. LNCS, vol. 4960, pp. 163–177. Springer, Heidelberg (2008)
34. Batcher, K.: Sorting networks and their applications. In: Proceedings of the AFIPS Spring Joint Computer Conference 32, pp. 307–314 (1968)
35. Moy, Y.: Sufficient preconditions for modular assertion checking. In: Logozzo, F., Peled, D.A., Zuck, L.D. (eds.) VMCAI 2008. LNCS, vol. 4905. Springer, Heidelberg (2008)
36. Gaubert, S., Gunawardena, J.: The duality theorem for min-max functions. C. R. Acad. Sci. Paris. 326 (Série I), 43–48 (1998)
37. Sankaranarayanan, S., Sipma, H., Manna, Z.: Scalable analysis of linear systems using mathematical programming. In: Cousot, R. (ed.) VMCAI 2005. LNCS, vol. 3385, Springer, Heidelberg (2005)