Widening Operators for Powerset Domains*

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

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

Abstract. The *finite powerset construction* upgrades an abstract domain by allowing for the representation of finite disjunctions of its elements. In this paper we define two generic widening operators for the finite powerset abstract domain. Both widenings are obtained by lifting any widening operator defined on the base-level abstract domain and are parametric with respect to the specification of a few additional operators. We illustrate the proposed techniques by instantiating our widenings on powersets of convex polyhedra, a domain for which no non-trivial widening operator was previously known.

1 Introduction

The design and implementation of effective, expressive and efficient abstract domains for data-flow analysis and model-checking is a very difficult task. For this reason, starting with [11], there continues to be strong interest in techniques that derive enhanced abstract domains by applying systematic constructions on simpler, existing domains. Disjunctive completion, direct product, reduced product and reduced power are the first and most famous constructions of this kind [11]; several variations of them as well as others constructions have been proposed in the literature.

Once the carrier of the enhanced abstract domain has been obtained by one of these systematic constructions, the abstract operations can be defined, as usual, as the optimal approximations of the concrete ones. While this completely solves the specification problem, it usually leaves the implementation problem with the designer and gives no guarantees about the efficiency (or even the computability) of the resulting operations. This motivates the importance of generic techniques whereby correct, even though not necessarily optimal, domain operations are derived automatically or semi-automatically from those of the domains the construction operates upon [8, 11, 18].

This paper focuses on the derivation of widening operators for a kind of disjunctive refinement we call *finite powerset construction*. As far as we know, this

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

is the first time that the problem of deriving non-trivial, provably correct widening operators in a domain refinement is tackled successfully. We also present its specialization to finite powersets of convex polyhedra. Not only is this included to help the reader gain a better intuition regarding the underlying approach but also to provide a definitely non-toy instance that is practically useful for applications such as data-flow analysis and model checking. Sets of polyhedra are implemented in Polylib [24, 28] and its successor *PolyLib* [25], even though no widenings are provided. Sets of polyhedra, represented with Presburger formulas made available by the Omega library [23, 26], are used in the verifier described in [7]; there, an extrapolation operator (i.e., a widening without convergence guarantee) on sets of polyhedra is described. Another extrapolation operator is implemented in the automated verification tool described in [16], where sets of polyhedra are represented using the clp(q, r) constraint library [22].

The rest of the paper is structured as follows: Section 2 recalls the basic concepts and notations; Section 3 defines the finite powerset construction as a disjunctive refinement for any abstract domain that is a join-semilattice; Section 4 presents two distinct strategies for upgrading any widening for the base-level domain into a widening for the finite powerset domain; Section 5 provides a technique for controlling the precision/efficiency trade-off of these widenings; Section 6 concludes. The proofs of all the stated results can be found in [5].

2 Preliminaries

For a set S, $\wp(S)$ is the powerset of S, whereas $\wp_{\mathrm{f}}(S)$ is the set of all the *finite* subsets of S; the cardinality of S is denoted by $\#S$. The first limit ordinal is denoted by ω. Let $\mathcal{O}$ be a set equipped with a well-founded ordering '$\succ$'. If M and N are finite multisets over $\mathcal{O}$, $\#(n, M)$ denotes the number of occurrences of $n \in \mathcal{O}$ in M and $M \gg N$ means that there exists $j \in \mathcal{O}$ such that $\#(j, M) > \#(j, N)$ and, for each $k \in \mathcal{O}$ with $k \succ j$, we have $\#(k, M) = \#(k, N)$. The relation '$\gg$' is well-founded [17].

In this paper we will adopt the abstract interpretation framework proposed in [13, Section 7], where the correspondence between the concrete and the abstract domains is induced from a concrete approximation relation and a concretization function. Since we are not aiming at maximum generality, for the sole purpose of simplifying the presentation, we will consider a particular instance of the framework by assuming a few additional but non-essential domain properties.

The concrete domain is modeled as a complete lattice of semantic properties $\langle C, \sqsubseteq, \bot, \top, \sqcup, \sqcap \rangle$; as usual, the concrete approximation relation $c_1 \sqsubseteq c_2$ holds if c_1 is a stronger property than c_2 (i.e., c_2 approximates c_1). The concrete semantics $c \in C$ of a program is formalized as the least fixpoint of a continuous (concrete) semantic function $\mathcal{F}\colon C \to C$, which is iteratively computed starting from the bottom element, so that $c = \mathcal{F}^{\omega}(\bot)$.

The abstract domain $\hat{D} = \langle D, \vdash, \mathbf{0}, \oplus \rangle$ is modeled as a join-semilattice (i.e., the least upper bound $d_1 \oplus d_2$ exists for all $d_1, d_2 \in D$). We will overload '$\oplus$'

so that, for each $S \in \wp_{\mathrm{f}}(D)$, $\bigoplus S$ denotes the least upper bound of S. The abstract domain $\hat{D}$ is related to the concrete domain by a monotonic and injective concretization function $\gamma\colon D \to C$. Monotonicity and injectivity mean that the abstract partial order '$\vdash$' is indeed the approximation relation induced on D by the concretization function γ. For all $d_1, d_2 \in D$, we will use the notation $d_1 \Vdash d_2$ to mean that $d_1 \vdash d_2$ and $d_1 \neq d_2$. We assume the existence of a monotonic abstract semantic function $\mathcal{F}^\sharp\colon D \to D$ that is sound with respect to $\mathcal{F}\colon C \to C$:

$$\forall c \in C : \forall d \in D : c \sqsubseteq \gamma(d) \implies \mathcal{F}(c) \sqsubseteq \gamma\big(\mathcal{F}^\sharp(d)\big). \tag{1}$$

This local correctness condition ensures that each concrete iterate can be safely approximated by computing the corresponding abstract iterate (starting from the bottom element $\mathbf{0} \in D$). However, due to the weaker algebraic properties satisfied by the abstract domain, the abstract upward iteration sequence may not converge. Even when it converges, it may fail to do so in a finite number of steps, therefore being useless for the purposes of static analysis.

Widening operators [9,10,13,14] provide a simple and general characterization for enforcing and accelerating convergence. We will adopt a minor variation of the classical definition of widening operator (see footnote 6 in [14, p. 275]).

Definition 1. (Widening.) *Let $\langle D, \vdash, \mathbf{0}, \oplus\rangle$ be a join-semilattice. The partial operator $\nabla\colon D \times D \rightarrowtail D$ is a* widening *if*

1. *$d_1 \vdash d_2$ implies $d_2 \vdash d_1 \nabla d_2$ for each $d_1, d_2 \in D$;*
2. *for each increasing chain $d_0 \vdash d_1 \vdash \cdots$, the increasing chain defined by $d'_0 := d_0$ and $d'_{i+1} := d'_i \nabla (d'_i \oplus d_{i+1})$, for $i \in \mathbb{N}$, is not strictly increasing.*

Any widening operator '∇' induces a corresponding partial ordering '$\vdash_\nabla$' on the domain D; this is defined as the reflexive and transitive closure of the relation $\big\{\, (d_1, d) \in D \times D \;\big|\; \exists d_2 \in D \,.\, d_1 \Vdash d_2 \wedge d = d_1 \nabla d_2 \,\big\}$. The relation '$\vdash_\nabla$' satisfies the ascending chain condition. We write $d_1 \Vdash_\nabla d_2$ to denote $d_1 \vdash_\nabla d_2 \wedge d_1 \neq d_2$.

It can be proved that the *upward iteration sequence with widenings* starting at the bottom element $d_0 := \mathbf{0}$ and defined by

$$d_{i+1} := \begin{cases} d_i, & \text{if } \mathcal{F}^\sharp(d_i) \vdash d_i; \\ d_i \nabla \big(d_i \oplus \mathcal{F}^\sharp(d_i)\big), & \text{otherwise,} \end{cases}$$

converges after a finite number $j \in \mathbb{N}$ of iterations [14]. Note that the widening is always applied to arguments $d = d_i$ and $d' = d_i \oplus \mathcal{F}^\sharp(d_i)$ satisfying $d \Vdash d'$. Also, when condition (1) holds, the post-fixpoint $d_j \in D$ of $\mathcal{F}^\sharp$ is a correct approximation of the concrete semantics, i.e., $\mathcal{F}^\omega(\bot) \sqsubseteq \gamma(d_j)$.

2.1 The Abstract Domain of Polyhedra

We now instantiate the abstract interpretation framework sketched above by presenting the well-known abstract domain of closed convex polyhedra. This

domain will be used throughout the paper to illustrate the generic widening techniques that will be defined.

Let $\mathbb{R}^n$, where $n > 0$, be the n-dimensional real vector space. The set $\mathcal{P} \subseteq \mathbb{R}^n$ is a *closed and convex polyhedron* (*polyhedron*, for short) if and only if $\mathcal{P}$ can be expressed as the intersection of a finite number of closed affine half-spaces of $\mathbb{R}^n$. The set $\mathbb{CP}_n$ of closed convex polyhedra on $\mathbb{R}^n$, when partially ordered by subset inclusion, is a lattice having the empty set and $\mathbb{R}^n$ as the bottom and top elements, respectively; the binary meet operation is set-intersection, whereas the binary join operation, denoted by '$\uplus$', is called *convex polyhedral hull* (*poly-hull*, for short). Therefore, we have the abstract domain

$$\widehat{\mathbb{CP}}_n := \langle \mathbb{CP}_n, \subseteq, \varnothing, \mathbb{R}^n, \uplus, \cap \rangle.$$

This domain can be related to several concrete domains, depending on the intended application. One example of a concrete domain is the complete lattice

$$\hat{\mathsf{A}}_n := \big\langle \wp(\mathbb{R}^n), \subseteq, \varnothing, \mathbb{R}^n, \cup, \cap \big\rangle.$$

Note that $\widehat{\mathbb{CP}}_n$ is a meet-sublattice of $\hat{\mathsf{A}}_n$, sharing the same bottom and top elements. Another example is the complete lattice $\hat{\mathsf{B}}_n := \big\langle \wp_c(\mathbb{R}^n), \subseteq, \varnothing, \mathbb{R}^n, \cup_c, \cap \big\rangle$, where $\wp_c(\mathbb{R}^n)$ is the set of all topologically closed and convex subsets of $\mathbb{R}^n$ and the join operation '$\cup_c$' returns the smallest topologically closed and convex set containing its arguments. As a final example of concrete domain for some analysis, consider the complete lattice $\hat{\mathsf{C}}_n := \big\langle \wp(\mathbb{CP}_n), \subseteq, \varnothing, \mathbb{CP}_n, \cup, \cap \big\rangle$.

The abstract domain $\widehat{\mathbb{CP}}_n$, which is a join-semilattice, is related to the concrete domains shown above by the concretization functions $\gamma^{\mathsf{A}} \colon \mathbb{CP}_n \to \wp(\mathbb{R}^n)$, $\gamma^{\mathsf{B}} \colon \mathbb{CP}_n \to \wp_c(\mathbb{R}^n)$ and $\gamma^{\mathsf{C}} \colon \mathbb{CP}_n \to \wp(\mathbb{CP}_n)$: for each $\mathcal{P} \in \mathbb{CP}_n$, we have both $\gamma^{\mathsf{A}}(\mathcal{P}) := \mathcal{P}$ and $\gamma^{\mathsf{B}}(\mathcal{P}) := \mathcal{P}$, and $\gamma^{\mathsf{C}}(\mathcal{P}) := {\downarrow}\mathcal{P} := \{\, \mathcal{Q} \in \mathbb{CP}_n \mid \mathcal{Q} \subseteq \mathcal{P} \,\}$. All these concretization functions are trivially monotonic and injective.

For each choice of the concrete domain $C \in \big\{ \wp(\mathbb{R}^n), \wp_c(\mathbb{R}^n), \wp(\mathbb{CP}_n) \big\}$, the continuous semantic function $\mathcal{F} \colon C \to C$ and the corresponding monotonic abstract semantic function $\mathcal{F}^\sharp \colon \mathbb{CP}_n \to \mathbb{CP}_n$, which is assumed to be correct, are deliberately left unspecified. The domain $\widehat{\mathbb{CP}}_n$ contains infinite ascending chains having no least upper bound in $\mathbb{CP}_n$. Thus, the convergence of the abstract iteration sequence has to be guaranteed by the adoption of widening operators.

The first widening on polyhedra was introduced in [15] and refined in [19]. This operator, denoted by '∇_s', has been termed *standard widening* and used almost universally. In [4], we presented a framework designed so that all its instances are widening operators on $\mathbb{CP}_n$. The standard widening '∇_s' is an instance of the framework and all the other instances, including the specific widening '$\hat{\nabla}$' defined and experimentally evaluated in [4], are at least as precise as '∇_s'. For a formal definition of both '∇_s' and '$\hat{\nabla}$', we refer the reader to [4].

3 A Disjunctive Refinement

In this section, we present the *finite powerset* operator, which is a domain refinement similar to disjunctive completion [11] and is obtained by a variant of

the *down-set completion* construction presented in [12]. The following notation and definitions are mainly borrowed from [2, Section 6].

Definition 2. (Non-redundancy.) *Let $\hat{D} = \langle D, \vdash, \mathbf{0}, \oplus \rangle$ be a join-semilattice. The set $S \in \wp(D)$ is called* non-redundant *with respect to '$\vdash$' if and only if $\mathbf{0} \notin S$ and $\forall d_1, d_2 \in S : d_1 \vdash d_2 \implies d_1 = d_2$. The set of finite non-redundant subsets of D (with respect to '$\vdash$') is denoted by $\wp_{\mathrm{fn}}(D, \vdash)$. The reduction function $\Omega^{\vdash}_{D} : \wp_{\mathrm{f}}(D) \to \wp_{\mathrm{fn}}(D, \vdash)$ mapping each finite set into its non-redundant counterpart is defined, for each $S \in \wp_{\mathrm{f}}(D)$, by*

$$\Omega^{\vdash}_{D}(S) := S \setminus \{\, d \in S \mid d = \mathbf{0} \vee \exists d' \in S \,.\, d \Vdash d' \,\}.$$

The restriction to the finite subsets reflects the fact that here we are mainly interested in an abstract domain where disjunctions are implemented by explicit collections of elements of the base-level abstract domain.

Definition 3. (Finite powerset domain.) *Let $\hat{D} := \langle D, \vdash, \mathbf{0}, \oplus \rangle$ be a join-semilattice. The* finite powerset domain *over $\hat{D}$ is the join-semilattice*

$$\hat{D}_{\mathrm{P}} := \big\langle \wp_{\mathrm{fn}}(D, \vdash), \vdash_{\mathrm{P}}, \mathbf{0}_{\mathrm{P}}, \oplus_{\mathrm{P}} \big\rangle,$$

where $\mathbf{0}_{\mathrm{P}} := \varnothing$ and, for all $S_1, S_2 \in \wp_{\mathrm{fn}}(D, \vdash)$, $S_1 \oplus_{\mathrm{P}} S_2 := \Omega^{\vdash}_{D}(S_1 \cup S_2)$.

The approximation ordering '$\vdash_{\mathrm{P}}$' induced by '$\oplus_{\mathrm{P}}$' is the Hoare powerdomain partial order [1], so that $S_1 \vdash_{\mathrm{P}} S_2$ if and only if $\forall d_1 \in S_1 : \exists d_2 \in S_2 \,.\, d_1 \vdash d_2$. A sort of Egli-Milner partial order relation [1] will also be useful: $S_1 \vdash_{\mathrm{EM}} S_2$ holds if and only if either $S_1 = \mathbf{0}_{\mathrm{P}}$ or $S_1 \vdash_{\mathrm{P}} S_2$ and $\forall d_2 \in S_2 : \exists d_1 \in S_1 \,.\, d_1 \vdash d_2$. An *(Egli-Milner) connector* for $\hat{D}_{\mathrm{P}}$, denoted by '$\boxplus_{\mathrm{EM}}$' is any upper bound operator for the Egli-Milner ordering on $\wp_{\mathrm{fn}}(D, \vdash)$. Note that although a *least* upper bound for '$\vdash_{\mathrm{EM}}$' may not exist, a connector can always be defined; for instance, we can let $S_1 \boxplus_{\mathrm{EM}} S_2 := \{ \bigoplus (S_1 \cup S_2) \}$.

Besides the requirement on finiteness, another difference with respect to the down-set completion of [12] is that we are dropping the assumption about the complete distributivity of the concrete domain. This is possible because our semantic domains are not necessarily related by Galois connections, so that this property does not have to be preserved.

The finite powerset domain is related to the concrete domain by means of the concretization function $\gamma_{\mathrm{P}} : \wp_{\mathrm{fn}}(D, \vdash) \to C$ defined by

$$\gamma_{\mathrm{P}}(S) := \bigsqcup \{\, \gamma(d) \;\big|\; d \in S \,\}.$$

Note that γ_{P} is monotonic but not necessarily injective. For $S_1, S_2 \in \wp_{\mathrm{fn}}(D, \vdash)$, we write $S_1 \equiv_{\gamma_{\mathrm{P}}} S_2$ to denote that the two abstract elements actually denote the same concrete element, i.e., when $\gamma_{\mathrm{P}}(S_1) = \gamma_{\mathrm{P}}(S_2)$. It is easy to see that '$\equiv_{\gamma_{\mathrm{P}}}$' is a congruence relation on $\hat{D}_{\mathrm{P}}$. As noted in [12], non-redundancy only provides a partial, syntactic form of reduction. On the other hand, requiring the full, semantic form of reduction for a finite powerset domain can be computationally very expensive.

A correct abstract semantic function $\mathcal{F}^\sharp_{\mathrm{P}}\colon\ \wp_{\mathrm{fn}}(D,\vdash) \to \wp_{\mathrm{fn}}(D,\vdash)$ on the finite powerset domain may be provided by an ad-hoc definition. More often, if the concrete semantic function $\mathcal{F}\colon C \to C$ satisfies suitable hypotheses, $\mathcal{F}^\sharp_{\mathrm{P}}$ can be safely induced from the abstract semantic function $\mathcal{F}^\sharp\colon D \to D$. For instance, if $\mathcal{F}$ is additive, we can define $\mathcal{F}^\sharp_{\mathrm{P}}$ as follows [11, 18]:

$$\mathcal{F}^\sharp_{\mathrm{P}}(S) := \Omega^{\vdash}_{D}\Big(\big\{\ \mathcal{F}^\sharp(d)\ \big|\ d \in S\ \big\}\Big).$$

3.1 The Finite Powerset Domain of Polyhedra

The polyhedral domain $(\widehat{\mathbb{CP}}_n)_{\mathrm{P}}$, having carrier $\wp_{\mathrm{fn}}(\mathbb{CP}_n, \subseteq)$, is the finite powerset domain over $\widehat{\mathbb{CP}}_n$. The approximation ordering is '$\subseteq_{\mathrm{P}}$' where, for each $\mathcal{S}_1, \mathcal{S}_2 \in \wp_{\mathrm{fn}}(\mathbb{CP}_n, \subseteq)$,

$$\mathcal{S}_1 \subseteq_{\mathrm{P}} \mathcal{S}_2 \iff \forall \mathcal{P}_1 \in \mathcal{S}_1 : \exists \mathcal{P}_2 \in \mathcal{S}_2\ .\ \mathcal{P}_1 \subseteq \mathcal{P}_2.$$

Let $\gamma^{\mathsf{A}}_{\mathrm{P}}$, $\gamma^{\mathsf{B}}_{\mathrm{P}}$ and $\gamma^{\mathsf{C}}_{\mathrm{P}}$ denote the (powerset) concretization functions induced by γ^{A}, γ^{B} and γ^{C}, respectively. Then, the relation '$\equiv_{\gamma^{\mathsf{A}}_{\mathrm{P}}}$' makes two finite sets of polyhedra equivalent if and only if they have the same set-union. The general problem of deciding the semantic equivalence with respect to $\gamma^{\mathsf{A}}_{\mathrm{P}}$ of two finite (non-redundant) collections of polyhedra is known to be computationally hard [27]. For $\gamma^{\mathsf{B}}_{\mathrm{P}}$, the relation '$\equiv_{\gamma^{\mathsf{B}}_{\mathrm{P}}}$' makes two finite sets of polyhedra equivalent if and only if they have the same poly-hull, so that the powerset construction provides no benefit at all. Finally, $\gamma^{\mathsf{C}}_{\mathrm{P}}$ is injective so that '$\equiv_{\gamma^{\mathsf{C}}_{\mathrm{P}}}$' coincides with the identity congruence relation.

Example 1. For the polyhedral domain $(\widehat{\mathbb{CP}}_1)_{\mathrm{P}}$, let[3]

$$\begin{aligned}
\mathcal{T}_0 &:= \big\{\{0 \le x \le 2\}, \{1 \le x \le 2\}, \{3 \le x \le 4\}, \{4 \le x \le 5\}\big\},\\
\mathcal{T}_1 &:= \big\{\{0 \le x \le 2\}, \{3 \le x \le 4\}, \{4 \le x \le 5\}\big\},\\
\mathcal{T}_2 &:= \big\{\{0 \le x \le 1\}, \{1 \le x \le 2\}, \{3 \le x \le 5\}\big\}.
\end{aligned}$$

Then $\mathcal{T}_0 \notin \wp_{\mathrm{fn}}(\mathbb{CP}_1, \subseteq)$, but $\mathcal{T}_1 = \Omega^{\subseteq}_{\mathbb{CP}_1}(\mathcal{T}_0) \in \wp_{\mathrm{fn}}(\mathbb{CP}_1, \subseteq)$. Also, $\mathcal{T}_1 \equiv_{\gamma^{\mathsf{A}}_{\mathrm{P}}} \mathcal{T}_2$.

4 Widening the Finite Powerset Domain

If the domain refinement of the previous section is meant to be used for static analysis, then a key ingredient that is still missing is a systematic way of ensuring the termination of the analysis. In this section, we describe two widening strategies that rely on the existence of a widening $\nabla\colon D \times D \rightarrowtail D$ on the base-level abstract domain.[4] We start by proposing a general specification of an extrapolation operator that lifts this '∇' operator to the powerset domain.

[3] In this and the following examples, a polyhedron $\mathcal{P} \in \mathbb{CP}_n$ will be denoted by a corresponding finite set of linear equality and non-strict inequality constraints.

[4] If the base-level abstract domain $\hat{D}$ is finite or Noetherian, so that it is not necessarily endowed with an explicit widening operator, then a dummy widening can be obtained by considering the least upper bound operator '$\oplus$'.

Definition 4. (The ∇-connected extrapolation heuristics.) *A partial operator* $h^{\nabla}_{\mathrm{P}} \colon \wp_{\mathrm{fn}}(D, \vdash)^2 \rightarrowtail \wp_{\mathrm{fn}}(D, \vdash)$ *is a* ∇-connected extrapolation heuristics for $\hat{D}_{\mathrm{P}}$ *if, for all* $S_1, S_2 \in \wp_{\mathrm{fn}}(D, \vdash)$ *such that* $S_1 \Vdash_{\mathrm{P}} S_2$, $h^{\nabla}_{\mathrm{P}}(S_1, S_2)$ *is defined and satisfies the following conditions:*

$$S_2 \vdash_{\mathrm{EM}} h^{\nabla}_{\mathrm{P}}(S_1, S_2); \tag{2}$$

$$\forall d \in h^{\nabla}_{\mathrm{P}}(S_1, S_2) \setminus S_2 : \exists d_1 \in S_1 \,.\, d_1 \Vdash_{\nabla} d; \tag{3}$$

$$\forall d \in h^{\nabla}_{\mathrm{P}}(S_1, S_2) \cap S_2 : \big((\exists d_1 \in S_1 \,.\, d_1 \Vdash d) \rightarrow (\exists d'_1 \in S_1 \,.\, d'_1 \Vdash_{\nabla} d)\big). \tag{4}$$

Informally, condition (2) ensures that the result is an upper approximation of S_2 in which every element covers at least one element of S_2 (i.e., the heuristics cannot add elements that are unrelated to S_2); conditions (3) and (4) ensure that in the resulting set, each element covering an element of S_1 originates from an application of '∇' to a (possibly different) element of S_1.

It is straightforward to construct an algorithm for computing a ∇-connected extrapolation heuristics for any given base-level widening '∇'.

Proposition 1. *For all* $S_1, S_2 \in \wp_{\mathrm{fn}}(D, \vdash)$ *such that* $S_1 \vdash_{\mathrm{P}} S_2$, *let*

$$h^{\nabla}_{\mathrm{P}}(S_1, S_2) := S_2 \oplus_{\mathrm{P}} \Omega^{\vdash}_{D}\big(\{\, d_1 \nabla d_2 \in D \mid d_1 \in S_1, d_2 \in S_2, d_1 \Vdash d_2 \,\}\big).$$

Then 'h^{∇}_{P}*' is a* ∇*-connected extrapolation heuristics for* $\hat{D}_{\mathrm{P}}$.

For the finite powerset domain over $\widehat{\mathbb{CP}}_n$, lines 10–15 of the algorithm specified in [7, Figure 8, page 773] provide an implementation of the heuristics 'h^{∇}_{P}' defined in Proposition 1, instantiated with the standard widening, '∇_s', on $\widehat{\mathbb{CP}}_n$.

Example 2. To see that the 'h^{∇}_{P}' defined in Proposition 1 is not a widening for $(\widehat{\mathbb{CP}}_n)_{\mathrm{P}}$, consider the strictly increasing sequence $\mathcal{T}_0 \subsetneq_{\mathrm{P}} \mathcal{T}_1 \subsetneq_{\mathrm{P}} \cdots$ in $\mathbb{CP}_1$ defined by $\mathcal{T}_j := \{\, \mathcal{P}_i \mid 0 \leq i \leq j \,\}$, where $\mathcal{P}_i := \{x = i\}$, for $i \in \mathbb{N}$. Then, no matter what the specification for '∇' is, we obtain $h^{\nabla}_{\mathrm{P}}(\mathcal{T}_j, \mathcal{T}_{j+1}) = \mathcal{T}_{j+1}$, for all $j \in \mathbb{N}$. Thus, the "widened" sequence is diverging.

4.1 Powerset Widenings Using Egli-Milner Connectors

Example 2 shows that, when computing $h^{\nabla}_{\mathrm{P}}(S_1, S_2)$, divergence is caused by those elements of S_2 that cover none of the elements occurring in S_1, i.e., when $S_1 \not\vdash_{\mathrm{EM}} S_2$. Thus, stabilization can be obtained by replacing S_2 with $S_1 \boxplus_{\mathrm{EM}} S_2$, where '$\boxplus_{\mathrm{EM}}$' is a connector for $\hat{D}_{\mathrm{P}}$. We therefore define a simple widening operator on the finite powerset domain that uses a connector to ensure termination.

Definition 5. (The '$_{\mathrm{EM}}\nabla_{\mathrm{P}}$' widening.) *Let '*$h^{\nabla}_{\mathrm{P}}$*' be a* ∇*-connected extrapolation heuristics and '*$\boxplus_{\mathrm{EM}}$*' be a connector for* $\hat{D}_{\mathrm{P}}$. *Let also* $S_1, S_2 \in \wp_{\mathrm{fn}}(D, \vdash)$, *where* $S_1 \Vdash_{\mathrm{P}} S_2$. *Then*

$$S_1 \,{}_{\mathrm{EM}}\nabla_{\mathrm{P}}\, S_2 := h^{\nabla}_{\mathrm{P}}(S_1, S'_2), \quad \textit{where } S'_2 := \begin{cases} S_2, & \textit{if } S_1 \vdash_{\mathrm{EM}} S_2; \\ S_1 \boxplus_{\mathrm{EM}} S_2, & \textit{otherwise.} \end{cases}$$

Theorem 1. *The '$_{\text{EM}}\nabla_{\text{P}}$' operator is a widening on $\hat{D}_{\text{P}}$.*

Example 3. To illustrate the widening operator '$_{\text{EM}}\nabla_{\text{P}}$' we consider the powerset domain $(\widehat{\mathbb{CP}}_1)_{\text{P}}$, with the standard widening '∇_s' on $\widehat{\mathbb{CP}}_1$ and the trivial connector '$\uplus_{\text{EM}}$' returning the singleton poly-hull of its arguments. Consider the sequence $\mathcal{T}_0 \subseteq_{\text{P}} \mathcal{T}_1 \subseteq_{\text{P}} \cdots$ of Example 2 and the widened sequence $\mathcal{U}_0 \subseteq_{\text{P}} \mathcal{U}_1 \subseteq_{\text{P}} \cdots$ where $\mathcal{U}_0 = \mathcal{T}_0$ and $\mathcal{U}_i = \mathcal{U}_{i-1} \,{}_{\text{EM}}\nabla_{\text{P}} (\mathcal{U}_{i-1} \uplus_{\text{P}} \mathcal{T}_i)$, for each $i > 0$. When computing $\mathcal{U}_1$, the second argument of the widening is $\mathcal{U}_0 \uplus_{\text{P}} \mathcal{T}_1 = \mathcal{T}_1$. Note that $\mathcal{U}_0 \vdash_{\text{EM}} \mathcal{T}_1$ does not hold so that the connector is needed. Thus, we obtain

$$\mathcal{U}_1 = h^{\nabla}_{\text{P}}(\mathcal{U}_0, \mathcal{U}_0 \uplus_{\text{EM}} \mathcal{T}_1) = h^{\nabla}_{\text{P}}\Big(\mathcal{U}_0, \{\{0 \leq x \leq 1\}\}\Big) = \{\{0 \leq x\}\}.$$

In the next iteration we obtain stabilization. Clearly, in general the precision of this widening will depend on the chosen connector operator.

For the polyhedral domain $\widehat{\mathbb{CP}}_n$, the powerset widening '$_{\text{EM}}\nabla_{\text{P}}$' using the '$h^{\nabla}_{\text{P}}$' heuristics defined in Proposition 1 is similar to but not quite the same as the operator sketched in [7]. As noted in that paper, the algorithm in [7, Figure 8, page 773] cannot ensure the termination of the analysis. To this end, instead of using a connector operator it is proposed that, when the cardinality of the abstract collection reaches a fixed threshold, a further poly-hull approximation be applied. However, there are examples indicating that such an approach cannot come with a termination guarantee when considering arbitrary increasing sequences [5].

4.2 Powerset Widenings Using Finite Convergence Certificates

We now present another widening operator (denoted here by '$_{\mu}\nabla_{\text{P}}$') for the finite powerset domain. This requires that the operator '∇' defined on the base-level domain is provided with a (finitely computable) finite convergence certificate. Formally, a *finite convergence certificate* for '∇' (on $\hat{D}$) is a triple $(\mathcal{O}, \succ, \mu)$ where $(\mathcal{O}, \succ)$ is a well-founded ordered set and $\mu\colon D \to \mathcal{O}$, which is called *level mapping*, is such that, for all $d_1 \Vdash d_2 \in D$, $\mu(d_1) \succ \mu(d_1 \nabla d_2)$. We will abuse notation by writing μ to denote the certificate $(\mathcal{O}, \succ, \mu)$.

Example 4. For the polyhedral domain $\widehat{\mathbb{CP}}_n$ and the standard widening '∇_s', we can define a certificate $(\mathcal{O}_s, \succ_s, \mu_s)$ where $\mathcal{O}_s$ is the pair $(\mathbb{N}, \mathbb{N})$, '$\succ_s$' the lexicographic ordering of the pair using $>$ for the individual ordering of the components and $\mu_s\colon \mathbb{CP}_n \to \mathcal{O}_s$ the level mapping $\mu_s(\mathcal{P}) = \big(n - \dim(\mathcal{P}), k\big)$, where $\dim(\mathcal{P})$ is the dimension of $\mathcal{P}$ and k the minimal number of half-spaces needed to define $\mathcal{P}$. Similarly, a certificate for the widening '$\hat{\nabla}$' on $\widehat{\mathbb{CP}}_n$ proposed in [4] can be obtained by considering the level mapping $\mu_b\colon \mathbb{CP}_n \to \mathcal{O}_b$ induced by the *limited growth ordering* relation '$\curvearrowright$' defined in [4], so that we have $\mu_b(\mathcal{P}_1) \succ_b \mu_b(\mathcal{P}_2)$ if and only if $\mathcal{P}_1 \curvearrowright \mathcal{P}_2$. For lack of space, we refer the reader to [4].

Given a certificate for '∇', we can define a suitable limited growth ordering relation '$\curvearrowright_{\text{P}}$' for the finite powerset domain $\hat{D}_{\text{P}}$ that satisfies the ascending chain condition.

Definition 6. (The '$\rightsquigarrow_{\mathrm{P}}$' relation.) *The relation $\rightsquigarrow_{\mathrm{P}} \subseteq \wp_{\mathrm{fn}}(D, \vdash) \times \wp_{\mathrm{fn}}(D, \vdash)$ induced by the certificate μ for '∇' is such that, for each $S_1, S_2 \in \wp_{\mathrm{fn}}(D, \vdash)$, $S_1 \rightsquigarrow_{\mathrm{P}} S_2$ if and only if either one of the following conditions holds:*

$$\mu(\bigoplus S_1) \succ \mu(\bigoplus S_2); \tag{5}$$

$$\mu(\bigoplus S_1) = \mu(\bigoplus S_2) \wedge \# S_1 > 1 \wedge \# S_2 = 1; \tag{6}$$

$$\mu(\bigoplus S_1) = \mu(\bigoplus S_2) \wedge \# S_1 > 1 \wedge \# S_2 > 1 \wedge \tilde{\mu}(S_1) \gg \tilde{\mu}(S_2) \tag{7}$$

where, for each $S \in \wp_{\mathrm{fn}}(D, \vdash)$, $\tilde{\mu}(S)$ denotes the multiset over $\mathcal{O}$ obtained by applying μ to each abstract element in S.

Proposition 2. *The '$\rightsquigarrow_{\mathrm{P}}$' relation satisfies the ascending chain condition.*

Intuitively, the relation '$\rightsquigarrow_{\mathrm{P}}$' will induce a certificate $\mu_{\mathrm{P}} \colon \hat{D}_{\mathrm{P}} \to \mathcal{O}_{\mathrm{P}}$ for the new widening. Namely, by defining $\mu_{\mathrm{P}}(S_1) \succ_{\mathrm{P}} \mu_{\mathrm{P}}(S_2)$ if and only if $S_1 \rightsquigarrow_{\mathrm{P}} S_2$, we will obtain $\mu_{\mathrm{P}}(S_1) \succ_{\mathrm{P}} \mu_{\mathrm{P}}(S_1 \, {}_{\mu}\nabla_{\mathrm{P}} \, S_2)$.

The specification of our "certificate-based widening" assumes the existence of a *subtract* operation for the base-level domain. It is expected that a specific subtraction would be provided for each domain; here we just indicate a minimal specification.

Definition 7. (Subtraction.) *The partial operator $\ominus \colon D \times D \rightarrowtail D$ is a* subtraction *for $\hat{D}$ if, for all $d_1, d_2 \in D$ such that $d_2 \vdash d_1$, we have $d_1 \ominus d_2 \vdash d_1$ and $d_1 = (d_1 \ominus d_2) \oplus d_2$.*

A trivial subtraction operator can always be defined as $d_1 \ominus d_2 := d_1$.

Example 5. In $\widehat{\mathbb{CP}}_n$, the function diff$\colon \mathbb{CP}_n \times \mathbb{CP}_n \to \mathbb{CP}_n$ is defined so that, for any $\mathcal{P}, \mathcal{Q} \in \mathbb{CP}_n$, $\mathrm{diff}(\mathcal{P}, \mathcal{Q})$ denotes the smallest closed and convex polyhedron containing the set difference $\mathcal{P} \setminus \mathcal{Q}$. Then, if $\mathcal{Q} \subseteq \mathcal{P}$, we have $\mathrm{diff}(\mathcal{P}, \mathcal{Q}) \subseteq \mathcal{P}$ and

$$\mathcal{P} = (\mathcal{P} \setminus \mathcal{Q}) \cup \mathcal{Q} = \mathrm{diff}(\mathcal{P}, \mathcal{Q}) \cup \mathcal{Q} = \mathrm{diff}(\mathcal{P}, \mathcal{Q}) \uplus \mathcal{Q},$$

so that 'diff' is a subtraction.

We can now define the *certificate-based widening* '${}_{\mu}\nabla_{\mathrm{P}}$'.

Definition 8. (The '${}_{\mu}\nabla_{\mathrm{P}}$' widening.) *Let '$\rightsquigarrow_{\mathrm{P}}$' be the limited growth ordering induced by the certificate μ for '∇' and let '$\boxplus_{\mathrm{P}}$' be any upper bound operator on $\hat{D}_{\mathrm{P}}$. Let $S_1, S_2 \in \wp_{\mathrm{fn}}(D, \vdash)$ be such that $S_1 \Vdash_{\mathrm{P}} S_2$. Also, if $\bigoplus S_1 \Vdash \bigoplus(S_1 \boxplus_{\mathrm{P}} S_2)$, let $d \in D$ be defined as $d := \big(\bigoplus S_1 \nabla \bigoplus(S_1 \boxplus_{\mathrm{P}} S_2)\big) \ominus \big(\bigoplus(S_1 \boxplus_{\mathrm{P}} S_2)\big)$. Then*

$$S_1 \, {}_{\mu}\nabla_{\mathrm{P}} \, S_2 := \begin{cases} S_1 \boxplus_{\mathrm{P}} S_2, & \text{if } S_1 \rightsquigarrow_{\mathrm{P}} S_1 \boxplus_{\mathrm{P}} S_2; \\ (S_1 \boxplus_{\mathrm{P}} S_2) \oplus_{\mathrm{P}} \{d\}, & \text{if } \bigoplus S_1 \Vdash \bigoplus(S_1 \boxplus_{\mathrm{P}} S_2); \\ \{\bigoplus S_2\}, & \text{otherwise.} \end{cases}$$

In the first case, we simply return the upper bound $S_1 \boxplus_{\mathrm{P}} S_2$, since this is enough to ensure a strict decrease in the level mapping. In the second case, the join of S_1 is strictly more precise than the join of $S_1 \boxplus_{\mathrm{P}} S_2$, so that we apply '$\nabla$' to them and then, using the subtraction operator, improve the obtained result, since $S_1 \frown_{\mathrm{P}} (S_1 \boxplus_{\mathrm{P}} S_2) \oplus_{\mathrm{P}} \{d\}$ holds. In the last case, since the join of $S_1 \boxplus_{\mathrm{P}} S_2$ is invariant, we return the singleton consisting of the join itself, as originally proposed in [11, Section 9].

Theorem 2. *The '$_{\mu}\nabla_{\mathrm{P}}$' operator is a widening on $\hat{D}_{\mathrm{P}}$.*

Example 6. To illustrate the last two cases of Definition 8, consider the domain $(\widehat{\mathbb{CP}}_1)_{\mathrm{P}}$ with the standard widening '∇_s' for $\widehat{\mathbb{CP}}_1$ certified by the level mapping μ_s defined in Example 4 and the upper bound operator '$\boxplus_{\mathrm{P}}$' defined as '$\oplus_{\mathrm{P}}$' so that $S_1 \boxplus_{\mathrm{P}} S_2 = S_2$ always holds.

Let $\mathcal{T}_1 = \{\{0 \leq x \leq 1\}\}$ and $\mathcal{T}_2 = \{\{0 \leq x \leq 1\}, \{2 \leq x \leq 3\}\}$. Then $\mathcal{T}_1 \not\frown_{\mathrm{P}} \mathcal{T}_2$, so that the condition for the first case in Definition 8 does not hold. The poly-hulls of $\mathcal{T}_1$ and $\mathcal{T}_2$ are $\{0 \leq x \leq 1\}$ and $\{0 \leq x \leq 3\}$, respectively, so that the condition for the second case holds. Since $\biguplus \mathcal{T}_1 \nabla_s \biguplus \mathcal{T}_2 = \{0 \leq x\}$, then by letting the polyhedron $\mathcal{P}$ be the element d as specified in Definition 8, we obtain $\mathcal{P} = \mathrm{diff}\big(\{0 \leq x\}, \{0 \leq x \leq 3\}\big) = \{3 \leq x\}$, so that

$$\mathcal{T}_1 \,{}_{\mu}\nabla_{\mathrm{P}}\, \mathcal{T}_2 = \mathcal{T}_2 \uplus_{\mathrm{P}} \{\mathcal{P}\} = \big\{\{0 \leq x \leq 1\}, \{2 \leq x \leq 3\}, \{3 \leq x\}\big\}.$$

Now let $\mathcal{T}_3 = \big\{\{x = 1\}, \{x = 3\}\big\}$ and $\mathcal{T}_4 = \big\{\{x = 1\}, \{x = 2\}, \{x = 3\}\big\}$. Then $\mathcal{T}_3 \not\frown_{\mathrm{P}} \mathcal{T}_4$, so that the condition for the first case in Definition 8 does not hold. Moreover, $\biguplus \mathcal{T}_3 = \biguplus \mathcal{T}_4 = \{1 \leq x \leq 3\}$, so that neither the second case applies. Thus, $\mathcal{T}_3 \,{}_{\mu}\nabla_{\mathrm{P}}\, \mathcal{T}_4 = \big\{\{1 \leq x \leq 3\}\big\}$.

As shown in the example above, Definition 8 does not require that the upper bound operator '$\boxplus_{\mathrm{P}}$' is based on the base-level widening '∇'. Moreover, the scheme of Definition 8 can be easily extended to any finite set of heuristically chosen upper bound operators on $\hat{D}_{\mathrm{P}}$, still obtaining a proper widening operator. The simplest heuristics, already used in the example above, is the one taking $\boxplus_{\mathrm{P}} := \oplus_{\mathrm{P}}$. If this fails to ensure a decrease in the level mapping, another possibility is the adoption of a ∇-connected extrapolation heuristics 'h^{∇}_{P}' for $\hat{D}_{\mathrm{P}}$. Anyway, many variations could be defined, depending on the required precision/efficiency trade-off. In the following section, we investigate one of these possibilities, which originates as a generalization of an idea proposed in [7].

5 Merging Elements According to a Congruence Relation

When computing a powerset widening $S_1 \nabla_{\mathrm{P}} S_2$, no matter if it is based on an Egli-Milner connector or a finite convergence certificate, some of the elements occurring in the second argument S_2 can be *merged together* (i.e., joined) without affecting the finite convergence guarantee. This merging operation can be guided by a congruence relation on the finite powerset domain $\hat{D}_{\mathrm{P}}$, the idea being that a well-chosen relation will benefit the precision/efficiency trade-off of the widening.

One option is to use semantics preserving congruence relations, i.e., refinements of the congruence relation '$\equiv_{\gamma_{\mathrm{P}}}$'. The availability of relatively efficient but incomplete tests for semantic equivalence can thus be exploited to improve the efficiency and/or the precision of the analysis. As the purpose of this paper is to provide generic widening procedures for powersets that are independent of the underlying domains and hence, of any intended concretizations, here we define these congruences in a way that is independent of the particular concrete domain adopted. Two such relations are the *identity congruence* relation, where no non-trivial equivalence is assumed, and the $\oplus$-*congruence* relation, where sets that have the same join are equivalent. However, the identity congruence will have no influence on the convergence of the iteration sequence, while the $\oplus$-congruence is usually the basis of the default, roughest heuristics for ensuring termination. We now define a new congruence relation that lies between these extremes.

Definition 9. ('$\lhd$' and '$\bowtie$'.) *The* content *relation* $\lhd \subseteq \wp_{\mathrm{fn}}(D,\vdash) \times \wp_{\mathrm{fn}}(D,\vdash)$ *is such that* $S_1 \lhd S_2$ *holds if and only if for all* $S_1' \in \wp_{\mathrm{fn}}(D,\vdash)$ *where* $S_1' \vdash_{\mathrm{P}} S_1$ *there exists* $S_1'' \in \wp_{\mathrm{fn}}(D,\vdash)$ *such that* $\bigoplus S_1' = \bigoplus S_1''$ *and* $S_1'' \vdash_{\mathrm{P}} S_2$. *The* same-content *relation* $\bowtie \subseteq \wp_{\mathrm{fn}}(D,\vdash) \times \wp_{\mathrm{fn}}(D,\vdash)$ *is such that* $S_1 \bowtie S_2$ *holds if and only if* $S_1 \lhd S_2$ *and* $S_2 \lhd S_1$.

Observe that the identity congruence relation can be obtained by strengthening the conditions in the definition of '$\lhd$', replacing $\bigoplus S_1' = \bigoplus S_1''$ with $S_1' = S_1''$; and the $\oplus$-congruence can be obtained by weakening the conditions, replacing $S_1'' \vdash_{\mathrm{P}} S_2$ with $\bigoplus S_1'' \vdash \bigoplus S_2$. Thus the same-content relation is a compromise between keeping all the information provided by the explicit set structure, as done by the identity congruence, and losing all of this information, as occurs with the $\oplus$-congruence.

For the finite powerset domain of polyhedra $(\widehat{\mathbb{CP}}_n)_{\mathrm{P}}$, the content relation '$\lhd$' corresponds to the condition that all the points in polyhedra in the first set are contained by polyhedra in the second set; and hence, the same-content congruence relation '$\bowtie$' coincides with the induced congruence relation '$\equiv_{\gamma_{\mathrm{P}}^{\mathrm{A}}}$'.

Proposition 3. *For all* $\mathcal{S}_1, \mathcal{S}_2 \in \wp_{\mathrm{fn}}(\mathbb{CP}_n, \subseteq)$, $\mathcal{S}_1 \bowtie \mathcal{S}_2$ *if and only if* $\mathcal{S}_1 \equiv_{\gamma_{\mathrm{P}}^{\mathrm{A}}} \mathcal{S}_2$.

Example 7. For $\mathcal{T}_1, \mathcal{T}_2 \in \wp_{\mathrm{fn}}(\mathbb{CP}_1, \subseteq)$ as defined in Example 1, we have $\mathcal{T}_1 \bowtie \mathcal{T}_2$. Consider also $\mathcal{T}_3, \mathcal{T}_4 \in \wp_{\mathrm{fn}}(\mathbb{CP}_1, \subseteq)$ where

$$\mathcal{T}_3 := \{\{0 \leq x \leq 3\}, \{1 \leq x \leq 5\}\}, \qquad \mathcal{T}_4 := \{\{0 \leq x \leq 5\}\}.$$

Then $\mathcal{T}_3 \bowtie \mathcal{T}_4$ and also $\mathcal{T}_2 \lhd \mathcal{T}_4$ although the converse does not hold. To see this, let S_1, S_2, and S_2' in Definition 9 be $\mathcal{T}_2$, $\mathcal{T}_4$, and $\mathcal{T}_4' := \{\{x = 2.5\}\}$, respectively. Then, if $\mathcal{T}_4''$ is such that $\biguplus \mathcal{T}_4'' = \biguplus \mathcal{T}_4'$ and $\mathcal{T}_4'' \subseteq_{\mathrm{P}} \mathcal{T}_4'$, we must have $\mathcal{T}_4'' = \mathcal{T}_4' \not\subseteq_{\mathrm{P}} \mathcal{T}_2$; hence, although $\mathcal{T}_4 = \{\biguplus \mathcal{T}_2\}$, we have $\mathcal{T}_4 \not\lhd \mathcal{T}_2$.

We now define an operation *merger* that is parametric with respect to the congruence relation and replaces selected subsets by congruent singleton sets.

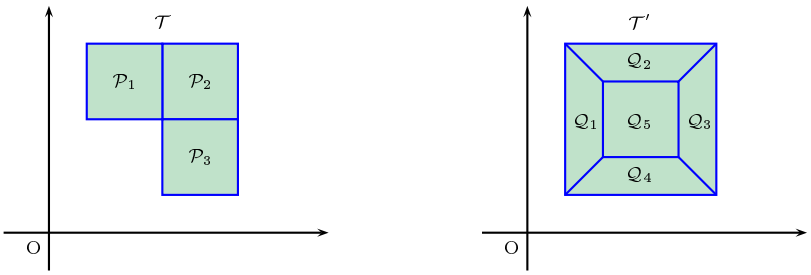

Fig. 1. Merging polyhedra according to '$\bowtie$'.

Definition 10. (Merge and mergers.) *Let R be a congruence relation on $\hat{D}_{\mathrm{P}}$. Then the* merge *relation* $\mathrm{merge}_R \subseteq \wp_{\mathrm{fn}}(D, \vdash) \times \wp_{\mathrm{fn}}(D, \vdash)$ *for R is such that* $\mathrm{merge}_R(S_1, S_2)$ *holds if and only if $S_1 \vdash_{\mathrm{P}} S_2$ and*

$$\forall d_2 \in S_2 : \exists S_1' \subseteq S_1 \,.\, d_2 = \bigoplus S_1' \wedge \{d_2\} \; R \; S_1'.$$

A set $S \in \wp_{\mathrm{fn}}(D, \vdash)$ is fully-merged for R, *if* $\mathrm{merge}_R(S, S')$ *implies $S = S'$; S is* pairwise-merged for R *if, for all $d_1, d_2 \in S$, we have that $\{d_1, d_2\}$ is fully-merged. An operator $\uparrow_R : \wp_{\mathrm{fn}}(D, \vdash) \to \wp_{\mathrm{fn}}(D, \vdash)$ is a* merger *for R if* $\mathrm{merge}_R(S, \uparrow_R S)$ *holds.*

Note that, for all $S \in \wp_{\mathrm{fn}}(D, \vdash)$ and congruence relations R, we have $S \vdash_{\mathrm{EM}} \uparrow_R S$.

For the finite powerset domain over $\widehat{\mathbb{CP}}_n$, lines 1–9 of the algorithm specified in [7, Figure 8, page 773] define a merger operator '$\uparrow_{\bowtie}$' such that, for each finite set $\mathcal{S}$ of polyhedra, $\uparrow_{\bowtie} \mathcal{S}$ is pairwise-merged.

Example 8. Figure 1 shows two examples of sets of polyhedra. In the left-hand diagram, the set $\mathcal{T} = \{\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3\}$ of three squares is not pairwise-merged for '$\bowtie$' since $\mathcal{P}_1 \cup \mathcal{P}_2$ and $\mathcal{P}_2 \cup \mathcal{P}_3$ are convex polyhedra. Both $\mathcal{T}_1 = \{\mathcal{P}_1 \cup \mathcal{P}_2, \mathcal{P}_3\}$ and $\mathcal{T}_2 = \{\mathcal{P}_1, \mathcal{P}_2 \cup \mathcal{P}_3\}$ are fully-merged and hence pairwise-merged for '$\bowtie$', and $\mathrm{merge}_{\bowtie}(\mathcal{T}, \mathcal{T}_i)$ holds for $i = 1, 2$. In the right-hand diagram, the set $\mathcal{T}' = \{\mathcal{Q}_1, \mathcal{Q}_2, \mathcal{Q}_3, \mathcal{Q}_4, \mathcal{Q}_5\}$ is pairwise-merged but not fully-merged for '$\bowtie$'. Since $\mathcal{Q}' := \bigcup \mathcal{T}'$ is a convex polyhedron, the singleton set $\{\mathcal{Q}'\}$ is fully-merged and hence pairwise-merged for '$\bowtie$' and $\mathrm{merge}_{\bowtie}(\mathcal{T}', \{\mathcal{Q}'\})$ holds.

6 Conclusion

We have studied the problem of endowing any abstract domain obtained by means of the finite powerset construction with a provably correct widening operator. We have proposed two generic widening operators and we have instantiated our techniques, which are completely general, on powersets of convex polyhedra, an abstract domain that is being used for static analysis and abstract model-checking and for which no non-trivial widening operator was previously known.

We have extended the *Parma Polyhedra Library* (PPL) [3, 6], a modern C++ library for the manipulation of convex polyhedra, with a prototype implementation of the widenings and their variants employing the 'widening up to' technique [20, 21]. The experimental work has just started, but the initial results obtained are very encouraging as our new widenings compare favorably, both in terms of precision and efficiency, with the extrapolation operator of [7].

References

1. S. Abramsky and A. Jung. Domain theory. In S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 3, chapter 1, pages 1–168. Clarendon Press, Oxford, UK, 1994.
2. R. Bagnara. A hierarchy of constraint systems for data-flow analysis of constraint logic-based languages. *Science of Computer Programming*, 30(1–2):119–155, 1998.
3. 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 0.5 edition, April 2003. Available at `http://www.cs.unipr.it/ppl/`.
4. R. Bagnara, P. M. Hill, E. Ricci, and E. Zaffanella. Precise widening operators for convex polyhedra. In R. Cousot, editor, *Static Analysis: Proceedings of the 10th International Symposium*, volume 2694 of *Lecture Notes in Computer Science*, pages 337–354, San Diego, California, USA, 2003. Springer-Verlag, Berlin.
5. R. Bagnara, P. M. Hill, and E. Zaffanella. Widening operators for powerset domains. Quaderno, Dipartimento di Matematica, Università di Parma, Italy, 2003. To appear.
6. R. Bagnara, E. Ricci, E. Zaffanella, and P. M. Hill. Possibly not closed convex polyhedra and the Parma Polyhedra Library. In M. V. Hermenegildo and G. Puebla, editors, *Static Analysis: Proceedings of the 9th International Symposium*, volume 2477 of *Lecture Notes in Computer Science*, pages 213–229, Madrid, Spain, 2002. Springer-Verlag, Berlin.
7. T. Bultan, R. Gerber, and W. Pugh. Model-checking concurrent systems with unbounded integer variables: Symbolic representations, approximations, and experimental results. *ACM Transactions on Programming Languages and Systems*, 21(4):747–789, 1999.
8. A. Cortesi, B. Le Charlier, and P. Van Hentenryck. Combinations of abstract domains for logic programming: Open product and generic pattern construction. *Science of Computer Programming*, 38(1–3):27–71, 2000.
9. P. Cousot and R. Cousot. Static determination of dynamic properties of programs. In B. Robinet, editor, *Proceedings of the Second International Symposium on Programming*, pages 106–130. Dunod, Paris, France, 1976.
10. P. Cousot and R. Cousot. Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Proceedings of the Fourth Annual ACM Symposium on Principles of Programming Languages*, pages 238–252, New York, 1977. ACM Press.
11. P. Cousot and R. Cousot. Systematic design of program analysis frameworks. In *Proceedings of the Sixth Annual ACM Symposium on Principles of Programming Languages*, pages 269–282, New York, 1979. ACM Press.
12. P. Cousot and R. Cousot. Abstract interpretation and applications to logic programs. *Journal of Logic Programming*, 13(2&3):103–179, 1992.

13. P. Cousot and R. Cousot. Abstract interpretation frameworks. *Journal of Logic and Computation*, 2(4):511–547, 1992.
14. 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.
15. 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.
16. G. Delzanno and A. Podelski. Model checking in CLP. In R. Cleaveland, editor, *Tools and Algorithms for Construction and Analysis of Systems, 5th International Conference, TACAS '99*, volume 1579 of *Lecture Notes in Computer Science*, pages 223–239, Amsterdam, The Netherlands, 1999. Springer-Verlag, Berlin.
17. N. Dershowitz and Z. Manna. Proving termination with multiset orderings. *Communications of the ACM*, 22(8):465–476, 1979.
18. G. Filé and F. Ranzato. The powerset operator on abstract interpretations. *Theoretical Computer Science*, 222:77–111, 1999.
19. N. Halbwachs. *Détermination Automatique de Relations Linéaires Vérifiées par les Variables d'un Programme*. Thèse de 3ème cycle d'informatique, Université scientifique et médicale de Grenoble, Grenoble, France, March 1979.
20. 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.
21. N. Halbwachs, Y.-E. Proy, and P. Roumanoff. Verification of real-time systems using linear relation analysis. *Formal Methods in System Design*, 11(2):157–185, 1997.
22. C. Holzbaur. OFAI clp(q,r) manual, edition 1.3.3. Technical Report TR-95-09, Austrian Research Institute for Artificial Intelligence, Vienna, 1995.
23. W. Kelly, V Maslov, W. Pugh, E. Rosser, T. Shpeisman, and D. Wonnacott. The Omega library interface guide. Technical Report CS-TR-3445, Department of Computer Science, University of Maryland, College Park, MD, USA, 1995.
24. H. Le Verge. A note on Chernikova's algorithm. *Publication interne* 635, IRISA, Campus de Beaulieu, Rennes, France, 1992.
25. 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 [28].
26. W. Pugh. A practical algorithm for exact array dependence analysis. *Communications of the ACM*, 35(8):102–114, 1992.
27. D. Srivastava. Subsumption and indexing in constraint query languages with linear arithmetic constraints. *Annals of Mathematics and Artificial Intelligence*, 8(3–4):315–343, 1993.
28. 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.