Towards a Quantitative Estimation of Abstract Interpretations (Extended abstract)

Francesco Logozzo[1] Corneliu Popeea[2] Vincent Laviron[3]

[1] Microsoft Research, Redmond, WA (USA)
logozzo@microsoft.com
[2] Max Planck Institute for Software Systems, Saarbrücken (Germany)
cpopeea@mpi-sws.mpg.de
[3] École Normale Supérieure, 45, rue d'Ulm, Paris (France)
Vincent.Laviron@ens.fr

Abstract. We aim to extend the notion of distance of sets to partially ordered sets (posets). We discuss several possible definitions, and we propose a relaxed definition of distance between elements of a domain. We apply it in the abstract interpretation theory, and we show in some preliminary examples how it seems well suited to formally quantify the relative loss of precision induced by abstract domains.

1 Introduction

Abstract interpretation is a theory of semantic program approximation based on domain theory. It precisely capture the *qualitative* relative loss of precision induced by static analyses: the more abstract the domain the less the information it captures about program executions. However, the theory does not provide a *quantitative* estimation of the precision loss induces by the abstraction. In this paper we report some preliminary thoughts and results on providing a quantitative evaluation of the errors induced by abstractions.

We are interested in defining metric on the elements of a domain. Roughly, a metric allows to measure the distance between the elements of a given set. When applied to domains (*i.e.* sets whose elements are related by some order), we would like to have a distance $d(\cdot, \cdot)$ which is somehow *compatible* with the underlying order $\sqsubseteq$. For instance, if $x \sqsubseteq y \sqsubseteq z$, then one expects that $d(x, y) \leq d(x, z)$.

The classical definition of distance of measure theory do not work well with domain theory. Intuitively, this is because in a metric space, one wants to compare *any* two elements, and it is hard to define a generic distance which is also aware, *e.g.*, that some elements are not comparable. In this paper we aim at relaxing the classical notion of distance on sets, and to conjugate it with the underlying order on abstract elements. We argue that some natural extensions are not satisfactory for our purposes, and we introduce a pseudo-metric which seems to be satisfactory on some examples.

2 Background: Distance in unordered sets

The minimal requirements that one expect for a distance of two elements x and y of an *unordered* set are: (i) that the distance between x and y is always not negative (negative distances make no sense) and it is equal to zero iff $x = y$; (ii) the distance of x from y is the same as the distance of y from x; and (iii) the distance of x from z is minimal, in that the *indirect* distance from x to some y and that from y to z is always larger than the *direct* distance.

Definition 1 (Distance). *Let S be a set. We say that $d \in [S \times S \longrightarrow \mathbf{R}]$ is a distance for s if it satisfies the following axioms:*

$d(x,y) \geq 0$ *(non-negativity)*
$d(x,y) = 0 \Leftrightarrow x = y$ *(iff-identity)*
$d(x,y) = d(y,x)$ *(symmetry)*
$d(x,z) \leq d(x,y) + d(y,z)$ *(triangle inequality)*

We recall below the definition of the Hausdorff distance which is commonly used to measure the distance of sets by taking the maximum distance of a set to the nearest point in the other set. Hausdorff distance will be useful in the following (Sect. 5.2), as it may provide a way to measure the distance of convex polyhedra.

Lemma 1 (Hausdorff distance). *Let X and Y be two subsets of a metric space that is endowed with a distance d. Then the following defines a distance*

$$d_H(X,Y) = \sup_{x \in X} \{ \inf_{y \in Y} \{ d(x,y) \} \}.$$

We will also use two variations of d_H, the minimum Hausdorff distance d_{Hmin} and the maximum Hausdorff distance d_{Hmax}:

$$d_{Hmin}(X,Y) = \inf_{x \in X} \{ \inf_{y \in Y} \{ d(x,y) \} \} \qquad d_{Hmax}(X,Y) = \sup_{x \in X} \{ \sup_{y \in Y} \{ d(x,y) \} \}$$

3 Distance on a partial order

3.1 Pseudo distance

As briefly discussed in the introduction, in general the classical definition of distance for unordered sets does not take advantage of the partial order between elements of a domain. The goal of this section is to define a pseudo-distance between elements of a domain. Let us consider properties in Def.1 one by one, and consider how we can extend them to match the distance in a domain.

It seems obvious that the distance between two elements of a domain should always be non-negative, so we leave *(non-negativity)*.

Requiring that the distance between two elements is zero if and only if the two elements are the same seems to be a too strong requirement. For instance, in abstract interpretations it is often the case that two distinct abstract elements a_1 and a_2 represent the same concrete object. In particular, useful static analyses relies on the fact that the abstract domain is simply a pre-ordered set whose order relation $\sqsubseteq$ does not enjoy the the anti-symmetric property. Formally, it can be the case that both $a_1 \sqsubseteq a_2$ and $a_2 \sqsubseteq a_1$ hold but $a_1 = a_2$ does not hold. In that case, we'd like to have the freedom to define a distance function such that $d(a_1, a_2) = 0$ even if $a_1 \neq a_2$ but $\gamma(a_1) = \gamma(a_2)$. The axiom *(iff-identity)* does not allow us such a definition. A turn-around is to change the abstract domain, and quotient it with respect to the concretization function. However, such a turn-around goes against our goal which is the definition of a notion of distance on *arbitrary* domains.

A first variation of the *(iff-identity)* would be to relax it, by replacing the equality with the $\sqsubseteq$ operator. The consequent axiom

$d(x,y) = 0 \Leftrightarrow x \sqsubseteq y$ *(iff-identity')*

it is not affected by the problem above. However, the axiom *(iff-identity')* is not useful since it implies that all elements in an ascending chain have distance 0, so that if $x \sqsubseteq y \sqsubseteq z$ then $d(x,y) = d(y,z) = d(x,z) = 0$. Even worse, when applied to the semantics of a program, usually defined as a fixpoint, *(iff-identity')* implies that all the semantics have distance zero from the bottom: $d(\bot, \sqcup_{n<+\infty} f^n(\bot)) = 0$.

Our choice is to simply relax the double implication of *(iff-identity)*, and to require that each element has distance 0 from itself, but allowing a zero-distance also for some distinct elements (Def. 1 with *(if-identity)* is called a pseudo-metric [5]):

$d(x, y) = 0 \Leftarrow x = y$ *(if-identity)*

The rationale behind is that: (i) a distance function gives a quantitative evaluation on how far are two elements in a domain; and (ii) in some cases we want the freedom to say that the distance of two *distinct* elements is negligible (for instance when they represent the same information up to some abstraction).

In an unordered set, the triangle inequality states that the distance between *any* two points is the shortest path between those. It turns out that requiring the triangle property to hold for arbitrary elements of a domain, without considering the order relation is too restrictive. We decided to use a weaker axiom, which requires the triangle inequality to hold *only* for comparable elements:

if $x \sqsubseteq z \sqsubseteq y$ then $d(x, y) \leq d(x, z) + d(z, y)$ *(weak triangle inequality)*

The axiom formalizes the intuition that the distance between elements of a chain should be compatible with the order: If y is not an immediate successor of x, then the path between x and y should not be shorter than any path passing through some element in between x and y.

The next definition sums up what said so far, and it introduces the notion of pseudo-distance compatible with a domain D. When D is clear from the context, we will simply say pseudo-distance.

Definition 2 (Pseudo-distance D-compatible). *Let $\langle D, \sqsubseteq \rangle$ be a domain ordered according to the relation $\sqsubseteq$. Let $\delta \in [D \times D \to \mathbf{R} \cup \{+\infty\}]$. We say that δ is a pseudo-distance D-compatible iff it satisfies the following axioms:*

$\delta(x, y) \geq 0$	*(non-negativity)*
$x = y \Rightarrow \delta(x, y) = 0$	*(if-identity)*
$\delta(x, y) = \delta(y, x)$	*(symmetry)*
$x \sqsubseteq z \sqsubseteq y \Rightarrow \delta(x, z) \leq \delta(x, y) + \delta(y, z)$	*(weak triangle inequality)*

It is also worth noting that, unlike the classical definition, we allow the distance between two elements to be $+\infty$.

3.2 Some simple properties of pseudo distances

The easiest example of a pseudo-distance is the zero function:

Lemma 1 (Zero) *The function $\delta^0(x, y) = 0$ is a pseudo-distance.*

Lemma 2 (Additivity) *Let δ_1, δ_2 be pseudo-distances. Then δ^Σ defined as $\delta^\Sigma(x, y) = \delta_1(x, y) + \delta_2(x, y)$ is pseudo-distance.*

It is worth noting that: (i) $\delta^0(x, y)$ is not a distance in the sense of Def. 1; and (ii) as a consequence of the lemmas above, the set of all pseudo-distances over a domain D form an additive monoid (unlike classical distances).

Lemma 3 (Multiplication by a scalar) *Let δ by a pseudo-distance and $k \in \mathbf{R} \cup \{+\infty\}$. Then δ^* defined as $\delta^*(x, y) = k \cdot \delta(x, y)$ is a pseudo-distance.*

We call an operator $\sqcup$ a *gathering* operator if it satisfies $x \sqsubseteq x \sqcup y$, $y \sqsubseteq x \sqcup y$, and $x \sqsubseteq y \Rightarrow x \sqcup y = y$. A gathering operator is a weaker notion of least upper bound operator, and it is useful in static analyses as *e.g.* [4, 6].

Lemma 4 *Let D a domain endowed with a gathering operator $\sqcup$. If $x \sqsubseteq y$, then $\delta(x \sqcup y, y) = 0$ and $\delta(x \sqcup y, x) = \delta(x, y)$.*

A similar result can be proven for an *intersection* operator $\sqcap$ satisfying $x \sqcap y \sqsubseteq x$, $x \sqcap y \sqsubseteq y$, and $x \sqsubseteq y \Rightarrow x \sqcap y = x$:

Lemma 5 *Let D a domain endowed with an intersection operator $\sqcap$. If $x \sqsubseteq y$, then $\delta(x \sqcap y, y) = \delta(x, y)$ and $\delta(x \sqcap y, x) = 0$.*

4 Some pseudo-distances

4.1 Structure-based

A first thought for defining a more interesting pseudo-distance on a domain is to count the number of intermediate elements between two elements:

Definition 1 (Path length (plen)) *The path length for two elements $x \sqsubseteq y$ of a domain D is*

$$\mathrm{plen}(x, y) = \min\{n \mid \{x_0, x_1, \ldots x_n\} \in \wp(D), x_0 = x, x_n = y, \forall 0 \leq i < n . x_i \sqsubseteq x_{i+1}\}.$$

If x and y are not comparable, we let $\mathrm{plen}(x, y) = +\infty$.

The function plen is not a pseudo-distance as it does not satisfy *(symmetry)*. Fixing it requires little work:

Lemma 6 (δ_{plen}) *The function $\delta_{\mathrm{plen}} \in [D \times D \to \mathbf{R} \cup \{+\infty\}]$ defined as*

$$\delta_{\mathrm{plen}}(x, y) = x \sqsubseteq y ? \mathrm{plen}(x, y) : (y \sqsubseteq x ? \mathrm{plen}(x, y) : +\infty)$$

is a pseudo-distance.

The pseudo-distance δ_{plen} is not very interesting, as it relates only elements that belong to the *same* chain. One can think to refine it by taking the average distance between two elements and their least upper bound (or the result of the gathering if the least upper bound is not defined):

Lemma 7 ($\delta^{\sqcup}_{\mathrm{plen}}$) *The function $\delta^{\sqcup}_{\mathrm{plen}} \in [D \times D \to \mathbf{R} \cup \{+\infty\}]$ defined as*

$$\delta^{\sqcup}_{\mathrm{plen}}(x, y) = 1/2 \cdot (\delta_{\mathrm{plen}}(x, x \sqcup y) + \delta_{\mathrm{plen}}(y, x \sqcup y))$$

is a pseudo-distance.

It is immediate to check that if x and y are comparable, then $\delta^{\sqcup}_{\mathrm{plen}}(x, y) = \delta_{\mathrm{plen}}(x, y)$.

Example 1. Let x_0, y_0, x_1, y_1 be elements of a domain such that $\delta_{\mathrm{plen}}(x_0, x_0 \sqcup y_0) = \delta_{\mathrm{plen}}(y_0, x_0 \sqcup y_0) = 50$, $\delta_{\mathrm{plen}}(x_1, x_1 \sqcup y_1) = 1$ and $\delta_{\mathrm{plen}}(y_1, x_1 \sqcup y_1) = 99$. Then, $\delta^{\sqcup}_{\mathrm{plen}}(x_0, y_0) = \delta^{\sqcup}_{\mathrm{plen}}(x_1, y_1) = 50$. □

In the example above, from the view of an abstract interpretation one would have expected that $\delta(x_0, y_0) \neq \delta(x_1, y_1)$, because when approximating x_1 and y_1 with $x_1 \sqcup y_1$ one *may* perform a small or a large error, but when approximating x_0 and y_0 with $x_0 \sqcup y_0$ one *always* performs a medium error. A way to overcome this drawback is to consider the minimal distance to the gathering (instead of the average distance):

Lemma 8 ($\delta^{\sqcup,m}_{\mathrm{plen}}$) *The function $\delta^{\sqcup,m}_{\mathrm{plen}} \in [D \times D \to \mathbf{R} \cup \{+\infty\}]$ defined as*

$$\delta^{\sqcup,m}_{\mathrm{plen}}(x, y) = \min(\delta_{\mathrm{plen}}(x, x \sqcup y), \delta_{\mathrm{plen}}(y, x \sqcup y))$$

is a pseudo-distance.

The drawback of distances based on path length is that they do not work well for infinite height lattices.

4.2 Affinity

An alternative pseudo-distance considers the *affinity* between abstract elements. The affinity distance was originally used for bounded powerset construction of Polyhedra [11]. The intuition behind is to have an percentage estimation of the simple constraints that are preserved at the gathering. Before stating it formally, we need some auxiliary definitions. Let us assume that seq is a function which given an abstract element x returns a minimal and *finite* set of terms equivalents to x. For instance, in the domain of convex Polyhedra $seq(\{-u \leq 0, u+v \leq 2, 2 \cdot u + 2 \cdot v \leq 4\}) = \{-u \leq 0, u+v \leq 2\}$. The function seq can be defined for most abstract domains, but not for all (think of $seq(\{u = f(u)\})$, as it may appear in a domain for abstract unification). We let $|\cdot|$ denote set cardinality, *eg*, $|\{-u \leq 0, u+v \leq 2\}| = 2$.

Lemma 9 (Affinity pseudo-distance) *The affinity distance of two elements x and y wrt the operator $\sqcup$ gives rise to the affinity distance defined as:*

$$\delta^{\sqcup}_{aff}(x,y) = 1 - \frac{|seq(x \sqcup y)|}{|seq(x) \cup seq(y)|}$$

Proof. The proof that the affinity satisfies the *(weak triangle inequality)* property is quite tricky. We report it in the appendix. □

It is immediate to observe that when $x \sqsubseteq y$, then $\delta^{\sqcup}_{aff}(x,y) = 1 - \frac{|seq(y)|}{|seq(x) \cup seq(y)|}$. One possibility to lift the affinity distance from a base to a powerset domain is simply to ignore the powerset structure of the domain. However, we did not yet proved it to satisfy Def. 2, so we leave it as an hypothesis:

Hypothesis 1 (Affinity for powerset domain) *Given $X = \{x_1, \ldots x_n\}$ and $Y = \{y_1, \ldots y_m\}$ elements of a powerset domain their affinity distance is:*

$$\delta^{\sqcup}_{aff}(X,Y) = 1 - \frac{|seq(X \sqcup Y)|}{|(\cup_i seq(x_i)) \cup (\cup_j seq(y_j))|}$$

4.3 Examples

The notion of pseudo-distance on a domain is useful to *quantify* the relative precision of different inferred invariants.

4.4 Nullness Domains

Let us consider the code in Fig. 1 to be analyzed with four different abstract domains: NN (nullness), TNN (type+nullness), DNN (disjunctive nullness) and DTNN (disjunctive type+nullness).It is easy to prove that: NN is the less precise domain, DTNN is the most precise, and TNN and DNN stand between the two but they are not comparable. The notion of distance of abstract elements allow us to give a *quantitative* comparison of the result of TNN and DNN.

Let us consider the method *m* written in C#-like syntax (the expression `a as B` casts `a` to `B` if `a` is a subtype of `B`, otherwise it returns `null`). The abstract states at the exit point of *m* using different abstract domains are in Fig. 2. The domain TNN is in general more precise than NN, but in the example it does not provide a more precise abstract state. The domain DNN is not comparable with TNN, and in the example it infers a more precise abstract state. Using the affinity distance we can give a quantitative formal

characterization of that (with an abuse of notation we confuse the abstract element with the name of abstract domain used):

$$\begin{aligned}\delta^{\sqcup}_{aff}(\text{NN},\text{TNN}) &= 1-\tfrac{1}{1}=0\\ \delta^{\sqcup}_{aff}(\text{NN},\text{DNN}) &= 1-\tfrac{1}{3}=\tfrac{2}{3}\\ \delta^{\sqcup}_{aff}(\text{DNN},\text{DTNN}) &= 1-\tfrac{3}{7}=\tfrac{4}{7}\\ \delta^{\sqcup}_{aff}(\text{NN},\text{DTNN}) &= 1-\tfrac{1}{7}=\tfrac{6}{7}\end{aligned}$$

The distance between NN and TNN is zero, meaning that no gain of information is obtained in the example when refining the NN just with types. The refinement with explicit disjunction (also known as trace partitioning [8]) produces an improvement of the 66%. The refinement of DNN with types improves the result by a further 57%. Overall using disjunction and types one obtains a result which is 85% more precise than the abstract domain of NN alone.

```
m(A a, out A x) {
  requires a != null;
  B b = a as B;
  if (b != null)
    x = new B(b);
  else
    x = null;
}
```

Fig. 1. An example for nullness analysis

Abstract Domain	Result
NN :	$\langle \mathtt{a} \to \mathcal{NN}, \mathtt{x} \to \top \rangle$
DNN :	$\langle \mathtt{a} \to \mathcal{NN}, \mathtt{x} \to \mathcal{NN} \rangle \vee \langle \mathtt{a} \to \mathcal{NN}, \mathtt{x} \to \mathcal{N} \rangle$
TNN :	$\langle \mathtt{a} \to \mathcal{NN}, \mathtt{x} \to \top \rangle$
DTNN :	$\langle \langle \mathtt{a} \to \mathcal{NN}, \mathtt{x} \to \mathcal{NN} \rangle, typeof(\mathtt{a}) <: \mathtt{B}, typeof(\mathtt{x}) == \mathtt{B} \rangle \vee$ $\langle \langle \mathtt{a} \to \mathcal{NN}, \mathtt{x} \to \mathcal{N} \rangle, typeof(\mathtt{a}) == \mathtt{A}, typeof(\mathtt{x}) == \mathtt{A} \rangle$

Fig. 2. The different results of the analysis using four different abstract domains. $\mathcal{NN}$ denotes that the reference is not null, $\mathcal{N}$ that it is definitely null.

4.5 McCarthy function

```
int MC(int n) {
  int t1, t2, r;
  if (n>100)
     r = n-10;
  else {
     t1 = n+11;
     t2 = MC(t1);
     r  = MC(t2);}
  return r;
}
```

Fig. 3. The McCarthy function

Abstract Domain	Result
Sign :	$\{0 < \mathtt{r}\}$
Intervals :	$\{91 \leq \mathtt{r}\}$
Octagons :	$\{91 \leq \mathtt{r}, \mathtt{n} - 10 \leq \mathtt{r}\}$

Fig. 4. The different inferred postconditions for the McCarthy function using different numerical abstract domains.

Pseudo-distances apply also when the underlying abstract domain has infinite height. Let us consider the McCarthy function (recalled in Fig. 3). We can analyze it with three different abstract domains: Signs, Intervals and Octagons. The inferred invariants are summarized in Fig. 4.5. The distances between those are given below.

$$\begin{aligned}\delta^{\sqcup}_{aff}(\text{Signs},\text{Intervals}) &= 1-\tfrac{1}{2}=\tfrac{1}{2}\\ \delta^{\sqcup}_{aff}(\text{Intervals},\text{Octagons}) &= 1-\tfrac{1}{2}=\tfrac{1}{2}\\ \delta^{\sqcup}_{aff}(\text{Signs},\text{Octagons}) &= 1-\tfrac{1}{3}=\tfrac{2}{3}\end{aligned}$$

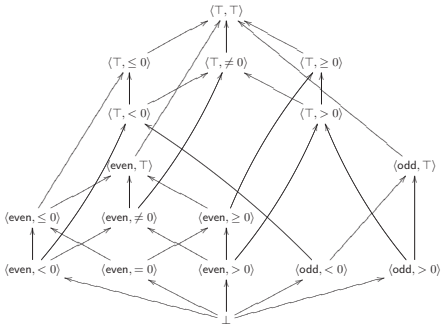

Fig. 5. The lattice Parity $\otimes$ Signs

Using Octagons, one obtains a postcondition which is 66% more precise than using Signs.

We can also consider to lift Octagons to disjunction of Octagons (DOctagons). In this case the inferred invariant is [10]:

$$\{101 \leq \mathtt{n}, \mathtt{n} - 10 \geq \mathtt{r}, \mathtt{n} - 10 \leq \mathtt{r}\} \vee \{\mathtt{n} < 101, \mathtt{r} \geq 91, \mathtt{r} \leq 91\}$$

and one can prove that the $\delta^{\sqcup}_{\mathit{aff}}(\text{Octagons}, \text{DOctagons}) = 1 - 2/6 = 2/3$.

5 Towards measuring the precision of Abstract Interpretations

The notion of distance is useful to formally quantify the error induced by using abstract elements. Given a concrete domain D and an abstract domains A, in the following we suppose that: (i) D and A are complete lattices; and (ii) they are related by a Galois connection $\langle \alpha, \gamma \rangle$.

5.1 Measuring abstract elements and domains

Definition 2 (δ-Error) *We define the error of approximating a concrete element c in A according to pseudo-distance δ as $\epsilon_\delta(c) = \delta(c, \gamma(\alpha(c))$.*

It is immediate to observe that as in a Galois connection $c \sqsubseteq \gamma(\alpha(c))$, then $\epsilon_{\delta^{\sqcup,m}_{\text{plen}}}$ boils down to $\epsilon_{\delta_{\text{plen}}}$.

Example 2. Let us consider the concrete domain to be the reduced product of Parity and Signs (Fig. 5) and the abstract domain to be Signs. Then $\gamma(\alpha(\langle \mathsf{even}, > 0\rangle)) = \gamma(> 0) = \langle \top, > 0\rangle$ implies that $\epsilon_{\delta_{\text{plen}}}(c) = \delta_{\text{plen}}(\langle \mathsf{even}, > 0\rangle, \langle \top, > 0\rangle) = 1$. If, on the other hand we chose Parity as abstract domain, then $\gamma(\alpha(\langle \mathsf{even}, > 0\rangle)) = \gamma(\mathsf{even}) = \langle \mathsf{even}, \top\rangle$ which implies that $\epsilon(c)_{\delta_{\text{plen}}} = \delta_{\text{plen}}(\langle \mathsf{even}, > 0\rangle, \langle \mathsf{even}, \top\rangle) = 2$, as $\langle \mathsf{even}, > 0\rangle \sqsubseteq \langle \mathsf{even}, \geq 0\rangle \sqsubseteq \langle \mathsf{even}, \top\rangle$ in Parity $\otimes$ Signs. $\square$

The example above suggests that one may be able to lift the Def. 2 to abstract domains, so that one can measure the loss of information induced by using an abstract domain.

Definition 3 (Error) *Let C be an abstract domain with a* finite *set of elements, and let A denote the abstraction of C. Then the average error of using A for C according to pseudo-distance δ is*

$$\epsilon_A = \frac{1}{|C|} \cdot \sum_{c \in C} \epsilon_\delta(c).$$

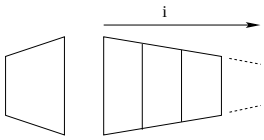

Fig. 6. The element x and the sequence of elements $\{y_1, .., y_i, ..\}$: $d_{Hmax}(x, y_i)$ grows arbitrary large with i, while $\epsilon_{\sqcup}(x, y_i)$ remains constant for all i.

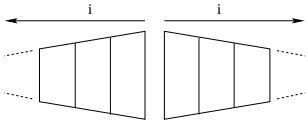

Fig. 7. Two sequences of elements $\{x_1, .., x_i, ..\}$ and $\{y_1, .., y_i, ..\}$: $d_{Hmin}(x_i, y_i)$ remains constant, while $\epsilon_{\sqcup}(x_i, y_i)$ grows arbitrary large with i.

Example 3. Let us consider two abstractions of Parity $\otimes$ Signs: Parity and SimpleSigns. Parity has four elements: $\bot$, $\top$, even and odd. The average error of using Parity as abstract domain is $18/17$. SimpleSigns has five elements: $\bot$, $\top$, < 0, > 0 and $\neq 0$. The average error of using SimpleSigns is $15/17$. □

As a consequence of the example, it turns out that even if from the point of the relative precision Parity and SimpleSigns are not comparable, on average, one may expect to have a smaller error when it uses SimpleSigns. It is worth noting that the quantitative errors we obtained are more relevant than those obtained using simple cardinality arguments. In fact $|5/17 - 4/17| = 1/17 < |18/17 - 15/17| = 3/17$.

5.2 Measuring operators

It is known that performing operations in the abstract may introduce a loss of precision. We can lift the previous results to formally evaluate the error induced by an abstract operator.

To estimate the error induced by the use of an abstract operator, we consider the average of the errors induced by applying the operator to each pair of abstract elements:

Definition 4 (op-Error) *Let op be the abstract counterpart for a concrete operator op_c. Then the average error of op with respect to δ is:*

$$\epsilon_{op} = \frac{1}{|A|^2} \sum_{a_1, a_2 \in A} \delta(\gamma(a_1 \ op \ a_2), \gamma(a_1) \ op_c \ \gamma(a_2)).$$

When $\epsilon_{op} = 0$ we say that op is δ-complete. Intuitively, a δ-complete operator does not introduce any error wrt the distance δ. A complete operator is one such that $\forall a_1, a_2.\ \gamma(a_1 \ op \ a_2) = \gamma(a_1) \ op_c \ \gamma(a_2)$. An immediate consequence of Def. 4 is that if op is a complete operator, then op is δ-complete for each δ.

The next logical step is to apply the definitions of this section to the most critical operator in a static analysis, that is the join. Our first approach was to use the Hausdorff distance, but it did not worked as one can have (i) $d_{Hmax}(x, y)$ arbitrary large, when err is constant (Fig. 6); or (ii) $d_{Hmin}(x, y)$ can be constant, when err can be arbitrary large (Fig. 7). Finding a good distance to evaluate the precision loss induced by the join that works for infinite abstract domains is still an open question for us.

6 Related Work

van Breugel [12] exploits the structure of a metric space to define the operational and the denotational semantics of a while language and he uses it to relate the two semantics, and to prove the existence of the fixpoints. His approach is a way different dual to ours, as we start from the domain structure, and we build a distance on the top of it.

Di Pierro and Wiklicky [2] propose a notion of probabilistic abstract interpretation, and they uses it to measure the measure the incompleteness of the abstract domain. With respect to our work, they change the

underlying framework (from standard abstract interpretation to linear spaces). An interesting future direction is to deepen the relation between our approach and theirs.

Distances and metric spaces have been object of wide investigation in other fields of computer science as machine learning or computer graphics. De Raedt and Ramon [1] propose to derive a distance from a partial order. They assume the existence of a weight function for the elements of the partial order, which is not clear how it works in the abstract interpretation setting, where abstract elements may approximate infinite elements. For instance we can use the affinity distance to also measure the distance between open convex polyhedra. Markov and Marinchev [7] define a semi-distance for Horn clauses. Eiter and Mannila [3] propose several distance measures for finite sets of points. Our affinity distance works also when the sets are infinite.

Monniaux [9] applies abstract interpretation-based techniques to bound the worst-case probability for some properties of interest. The affinity distance was originally used in [11] (where it was named "planar affinity measure") to construct a powerset extension of the polyhedron base domain. In general, such a powerset extension can be either expensive (the number of elements is exponential when compared to the base domain) or imprecise (when the number of disjuncts is syntactically bounded). In this context, the affinity distance was used to identify pairs of elements that are likely to be joined (using the least upper bound operator) with a small precision loss.

7 Conclusions

We presented the preliminary results on our investigations to *quantify* the loss of precision in static analyses. We show how the classical notion of distance on metric spaces is too strict, and we proposed a weaker notion, the pseudo-distance. We defined some pseudo-distances and we apply them to measure the relative precision of invariants inferred with (possibly non-comparable) abstract domains. We lifted the notion of pseudo-distance to the elements of the abstract domains (so to estimate the relative precision loss) and to operators on abstract domain. There are still some open issues, both technical and conceptual. For instance it is not clear if the affinity distance lifted to powerset is a pseudo-metric and we aim at extending the distance on abstract domain to cope with *infinite* abstract domains, which are often of more interest for static analyses.

References

1. L. De Raedt and J. Ramon. Deriving distance metrics from generality relations. *Pattern Recognition Letters*, 30(3):187–191, 2009.
2. A. Di Pierro and H. Wiklicky. Measuring the precision of abstract interpretations. In *LOPSTR (LNCS 2042: Selected Papers)*, pages 147–164, 2000.
3. T. Eiter and H. Mannila. Distance measures for point sets and their computation. *Acta Inf.*, 34(2):109–133, 1997.
4. J. Feret. The arithmetic-geometric progression abstract domain. In *VMCAI*, pages 42–58, 2005.
5. P. Giannopoulos and R. C. Veltkamp. A pseudo-metric for weighted point sets. In *Proceedings of the European Conference on Computer Vision*, 2002.
6. V. Laviron and F. Logozzo. Subpolyhedra: A (more) scalable approach to infer linear inequalities. In *VMCAI*, pages 229–244, 2009.
7. Z. Markov and I. Marinchev. Coverage-based semi-distance between horn clauses. In *AIMSA*, pages 331–339, 2000.
8. L. Mauborgne and X. Rival. Trace Partitioning in Abstract Interpretation Based Static Analyzers. In *ESOP*, 2005.
9. D. Monniaux. Abstract interpretation of programs as markov decision processes. In *SAS*, pages 237–254, 2003.
10. C. Popeea. *Disjunctive Invariants for Modular Static Analysis*. PhD thesis, School of Computing, National University of Singapore, 2008.
11. C. Popeea and W.N. Chin. Inferring disjunctive postconditions. In *ASIAN CS Conference*, 2006.
12. F. van Breugel. An introduction to metric semantics: operational and denotational models for programming and specification languages. *Theor. Comput. Sci.*, 258(1-2):1–98, 2001.

A Proof of Lemma 9

Proof: It is trivial to show that $\delta^{\sqcup}_{aff}$ satisfies the *non-negativity*, *identity* and *symmetry* properties. We then need to prove that $\delta^{\sqcup}_{aff}$ satisfies also the *weak triangle inequality*: $\delta^{\sqcup}_{aff}(x,y) \leq \delta^{\sqcup}_{aff}(x,z) + \delta^{\sqcup}_{aff}(z,y)$. Using the hypothesis $x \sqsubseteq z \sqsubseteq y$, the inequality reduces to:

$$1 - \frac{|seq(y)|}{|seq(x) \cup seq(y)|} \leq 1 - \frac{|seq(z)|}{|seq(x) \cup seq(z)|} + 1 - \frac{|seq(y)|}{|seq(y) \cup seq(z)|}$$

Subsequently:

$$\frac{|seq(x) \cup seq(y)| - |seq(y)|}{|seq(x) \cup seq(y)|} \leq \frac{|seq(x) \cup seq(z)| - |seq(z)|}{|seq(x) \cup seq(z)|} + \frac{|seq(y) \cup seq(z)| - |seq(y)|}{|seq(y) \cup seq(z)|}$$

We use P_0 to P_7 to represent cardinalities of sets where the subscript indicates the membership of edges to x,y,z:

	z	y	x	
P_1	0	0	1	$\|seq(x) \backslash seq(y) \backslash seq(z)\|$
P_2	0	1	0	$\|seq(y) \backslash seq(x) \backslash seq(z)\|$
P_3	0	1	1	$\|seq(x) \cap seq(y) \backslash seq(z)\|$
P_4	1	0	0	$\|seq(z) \backslash seq(x) \backslash seq(y)\|$
p_5	1	0	1	$\|seq(x) \cap seq(z) \backslash seq(y)\|$
P_6	1	1	0	$\|seq(y) \cap seq(z) \backslash seq(x)\|$
P_7	1	1	1	$\|seq(x) \cap seq(y) \cap seq(z)\|$

From $x \sqsubseteq z \sqsubseteq y$, we obtain that $seq(x) \cap seq(y) \subseteq seq(z)$. Subsequently, we have that $P_3 = 0$. We use the notation $N = \Sigma P_i$. The inequality to prove can then be simplified to:

$$\frac{P_1 + P_5}{N - P_4} \leq \frac{P_1}{N - P_2} + \frac{P_4 + P_5}{N - P_1}$$

This inequality can be proven as follows:

$$(P_1 + P_5)(N - P_2)(N - P_1) \leq P_1(N - P_4)(N - P_1) + (P_4 + P_5)(N - P_4)(N - P_2)$$

$$N^2P_1 + N^2P_5 + P_1^2P_2 + P_1P_2P_5 + NP_1^2 + NP_1P_4 + NP_2P_4 + NP_2P_5 + NP_4^2 + NP_4P_5 \leq$$
$$NP_1^2 + NP_1P_2 + NP_1P_5 + NP_2P_5 + N^2P_1 + P_1^2P_4 + N^2P_4 + N^2P_5 + P_2P_4^2 + P_2P_4P_5$$

$$P_1^2P_2 + P_1P_2P_5 + NP_1P_4 + NP_2P_4 + NP_4^2 + NP_4P_5 \leq$$
$$NP_1P_2 + NP_1P_5 + P_1^2P_4 + N^2P_4 + P_2P_4^2 + P_2P_4P_5$$

Since $NP_1P_4 + NP_2P_4 + NP_4^2 + NP_4P_5 \leq N^2P_4$, the inequality reduces to:
$P_1^2P_2 + P_1P_2P_5 \leq NP_1P_2 + NP_1P_5 + P_1^2P_4 + P_2P_4^2 + P_2P_4P_5$

Since $P_1^2P_2 \leq NP_1P_2$ and $P_1P_2P_5 \leq NP_1P_5$, the inequality is proven and thus$\delta^{\sqcup}_{aff}$ satisfies the weak triangle inequality. □