Abstract Interpretation

Wei Le

February 23, 2021

Acknowledgement: this lecture used slides from Profs Alex Aiken, David Schmidt

Outline

- What is abstract intepretation?
- Abstract domain
- Galois connection
- Design an abstract interpretation system
- Certify neural networks: abstract interpretation for artificial intelligence (ai^2, ai4ai)

History: Patrick Cousot, Radhia Cousot 1977

- ▶ Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints, 1977
- ▶ Methods and Logics for Proving Programs, 1990
- ▶ Completeness in Abstract Interpretation, 1995
- ▶ Directions for Research in Approximate System Analysis, 1999
- ▶ Probabilistic Abstract Interpretation, 2012
- ▶ An abstract interpretation framework for termination, 2012
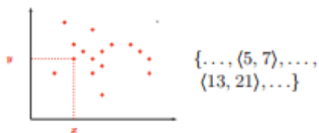- ▶ Abstract interpretation: past, present and future, 2014

What is abstract intepretation?

▶ Define an abstract domain and perform computation on abstract domain

▶ Soundess: the conclusions from abstract intepretation are correct comparing to the conclusions reached from concrete executions

▶ A theoretical framework to formalize *approximation*

▶ A sound approximation: the conclusion proved in the abstract domain will be held in the conrete domain

▶ Computing semantics on abstract domains: an applications of abstraction to the semantics of programming languages

▶ abstract intepretation can lose information, meaning some conclusions that can be reached by the concrete executions but cannot be reached by abstract intepretation
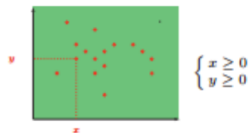
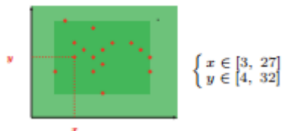# An Example

See Prof. Alex Aiken's slide
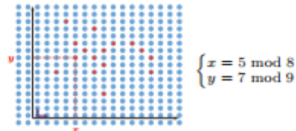
Partitioning and Abstract Domain

(a) [In]finite Set of Points

$$\{\ldots, \langle 5, 7 \rangle, \ldots, \langle 13, 21 \rangle, \ldots\}$$

(b) Sign Abstraction

$$\begin{cases} x \geq 0 \\ y \geq 0 \end{cases}$$

(c) Interval Abstraction

$$\begin{cases} x \in [3, 27] \\ y \in [4, 32] \end{cases}$$

(d) Simple Congruence Abstraction

$$\begin{cases} x = 5 \bmod 8 \\ y = 7 \bmod 9 \end{cases}$$
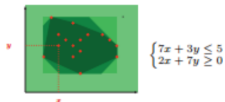
(b) Polyhedral Abstraction

$$\begin{cases} 7x + 3y \leq 5 \\ 2x + 7y \geq 0 \end{cases}$$

Partitioning and Abstract Domain

▶ Partition: abstract sets of environments/concrete inputs (D)

▶ Abstract domain construction: What are the properties about D that I wish to calculate?

    ▶ Interval domain: upper and lower bound

    ▶ Congruence domain: measure density of its values

- Intervals (nonrelational):

$$x \Rightarrow [a, b], \ y \Rightarrow [a', b'], \ ...$$

- Polyhedra (relational):

$$x + y - 2z \leq 10, \ ...$$

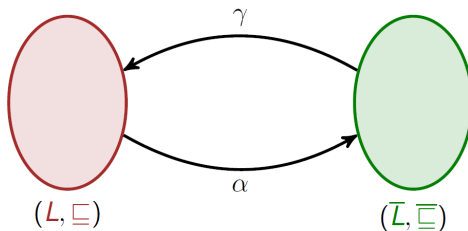- Difference-bound matrices (weakly relational):

$$y - x \leq 5, \ z - y \leq 10, \ ...$$

Abstract interpretation and Galois connection

An abstract interpretation consists of:

- An abstract domain A (+,-,0) and concrete domain D (Int)
- Concretization γ and abstraction functions σ, forming a *Galois connection*
- (sound) abstract semantic function (s)

Galois Connection: intuition

Concretization
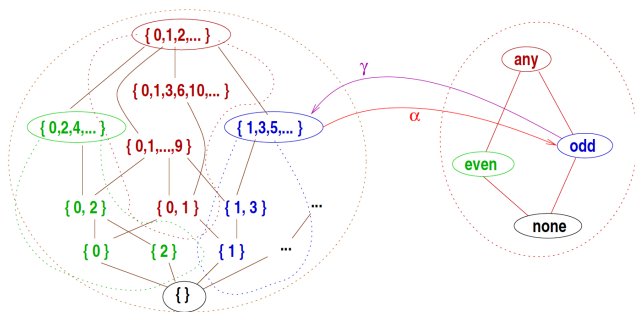
γ is the concretization function.

$\gamma(\overline{y})$ is the concrete value in L that is represented by $\overline{y}$.

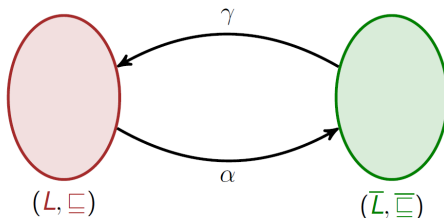### Abstraction

α is the abstraction function.

$\alpha(x)$ is the most precise abstract value in $\overline{L}$ whose concretization approximates x.

Galois connection: example

That is, for all $c \in \gamma[A], c = \gamma \circ \alpha(c)$; for all $a \in \alpha[C], a = \alpha \circ \gamma(a)$.

Galois Connection: Definition

Definition

A Galois connection between a lattice $(L, \sqsubseteq)$ and a lattice $(\overline{L}, \overline{\sqsubseteq})$ is a pair of functions (α, γ), with $\alpha : L \to \overline{L}$ and $\gamma : \overline{L} \to L$, satisfying:

$$\alpha(x) \mathrel{\overline{\sqsubseteq}} \overline{y} \quad \text{iff} \quad x \sqsubseteq \gamma(\overline{y}) \qquad \text{(for all } x \in L, \overline{y} \in \overline{L})$$

Notation for Galois connections: $(L, \sqsubseteq) \xleftrightarrow[\alpha]{\gamma} (\overline{L}, \overline{\sqsubseteq})$
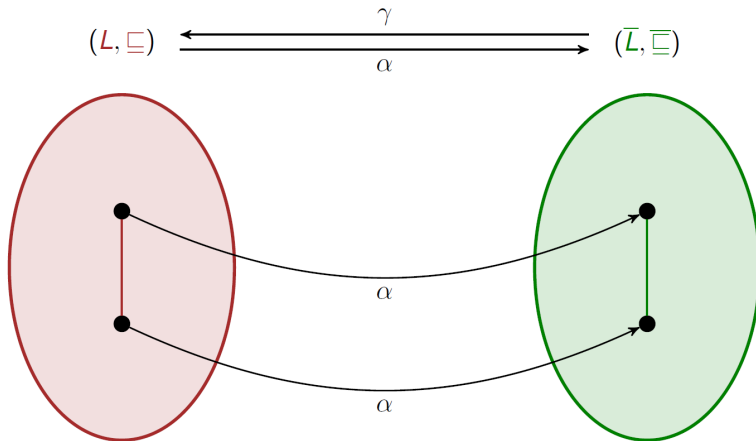
Galois Connection: monotonic function

Consider two lattices $(L, \sqsubseteq)$ and $(\overline{L}, \overline{\sqsubseteq})$.

For any two functions $\alpha : L \to \overline{L}$ et $\gamma : \overline{L} \to L$, we have

$$(L, \sqsubseteq) \xleftrightarrow[\alpha]{\gamma} (\overline{L}, \overline{\sqsubseteq}) \quad \text{iff} \quad \begin{cases} x \sqsubseteq \gamma \circ \alpha(x) & \text{(for all } x \in L) \\ \alpha \circ \gamma(\overline{y}) \overline{\sqsubseteq} \overline{y} & \text{(for all } \overline{y} \in \overline{L}) \\ \alpha \text{ is monotonic} \\ \gamma \text{ is monotonic} \end{cases}$$

Galois Connection: monotonic function

α is monotonic

Galois Connection: monotonic function
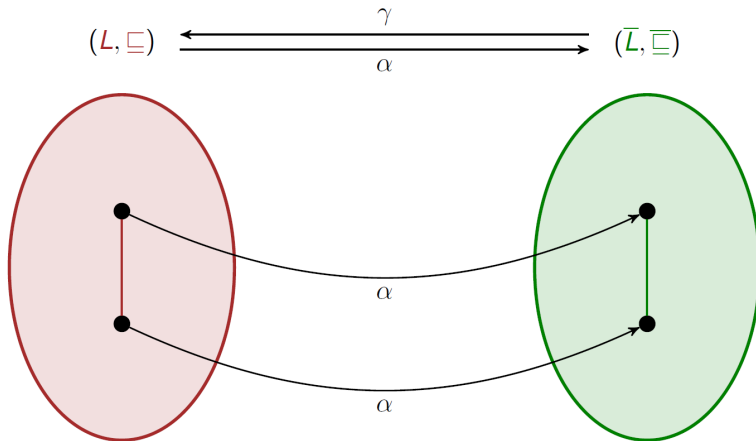
α is monotonic

Designing an abstract interpretation system: an example

Now that we know how to model $c \in C$ by $\alpha(c) \in A$, we must model concrete computation steps, $f : C \to C$, by abstract computation steps, $f^{\#} : A \to A$.

Example: We have concrete domain, Nat, and concrete operation, $succ : Nat \to Nat$, defined as $succ(n) = n + 1$.

We have abstract domain, $Parity$, and abstract operation, $succ^{\#} : Parity \to Parity$, defined as

$$succ^{\#}(even) = odd, \quad succ^{\#}(odd) = even$$
$$succ^{\#}(any) = any, \quad succ^{\#}(none) = none$$

$succ^{\#}$ must be consistent (sound) with respect to $succ$:

if $n \; \mathcal{R}_{Nat} \; a$, then $succ(n) \; \mathcal{R}_{Nat} \; succ^{\#}(a)$

where $\mathcal{R} \subseteq Nat \times Parity$ relates numbers to their parities (e.g., $2 \; \mathcal{R}_{Nat} \; even$, $5 \; \mathcal{R}_{Nat} \; odd$, etc.).

Designing an abstract interpretation system: an example

Example 1: $n \; \mathcal{R}_{\mathrm{Nat}} \; a$ implies $\mathrm{succ}(n) \; \mathcal{R}_{\mathrm{Nat}} \; \mathrm{succ}^{\#}(a)$

Galois connection: $\wp(\mathrm{Nat}) \langle \alpha, \gamma \rangle \mathrm{Parity}$

$\mathrm{succ}^{*} : \wp(\mathrm{Nat}) \rightarrow \wp(\mathrm{Nat})$

$\mathrm{succ}^{*}(S) = \{\mathrm{succ}(n) \mid n \in S\}$

where $\mathrm{succ}(n) = n + 1$

$\mathrm{succ}^{\#} : \mathrm{Parity} \rightarrow \mathrm{Parity}$

$\mathrm{succ}^{\#}(\mathrm{even}) = \mathrm{odd}, \quad \mathrm{succ}^{\#}(\mathrm{odd}) = \mathrm{even}$

$\mathrm{succ}^{\#}(\mathrm{any}) = \mathrm{any}, \quad \mathrm{succ}^{\#}(\mathrm{none}) = \mathrm{none}$

We have that $\alpha \circ \mathrm{succ}^{*} = \mathrm{succ}^{\#} \circ \alpha$:
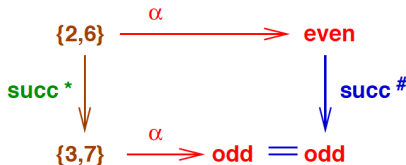
Abstract intepretation for robust neural networks

What does it mean to prove the robustness of a neural network?

"panda"
57.7% confidence

$+\epsilon$

$=$

"gibbon"
99.3% confidence
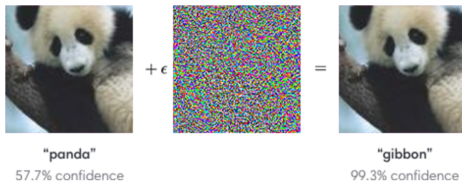
Attack	Original	Perturbed	Diff
FGSM [12], $\epsilon = 0.3$			
Brightening, $\delta = 0.085$			

Fig. 1: Attacks applied to MNIST images [25].

Abstract intepretation for robust neural networks

Why can we use abstract intepretation?

- ▶ Deep Neural Nets: Affine transforms + Restricted nonlinearity
- ▶ Abstract Interpretation: Scalable and Precise Numerical Domains
- ▶ Abstract semantics should be defined on: Affine transformation (multiplictaion and addition), ReLu

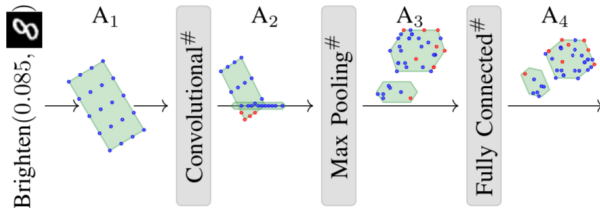# Abstract intepretation for robust neural networks

What is the abstract domain?

- ▶ Abstract domain: shapes expressible as a set of logical constraints
- ▶ Zonotope: a center-symmetric convex closed polyhedron [CAV09]

Abstract intepretation for robust neural networks

High level ideas:

- ▶ abstract element: here A1 is an abstract element that represent a group of inputs: a digit 8 and all the images generated throught he pertubation (red indicates that additional images are also included)

- ▶ abstract layer: process abstract element

- ▶ abstract transformer: design abstract semantics for each concrete transformation available in the neural network

- ▶ A4 is an overappoximation of input of interest

- ▶ verify robustness on A4

Abstract intepretation for robust neural networks

Further reading:
AI2: Safety and Robustness Certification of Neural Networks with
Abstract Interpretation