

Introduction to Abstract Interpretation

What is an Abstraction?

The Galois Insertion

Proving Correctness

Widening Operator

These notes are based on lecture notes made available
by Jeff Foster (CMSC 631, Fall 2003), David Schmidt, and Alex Aiken

What is an Abstraction?

A property from some domain



Blue (color)

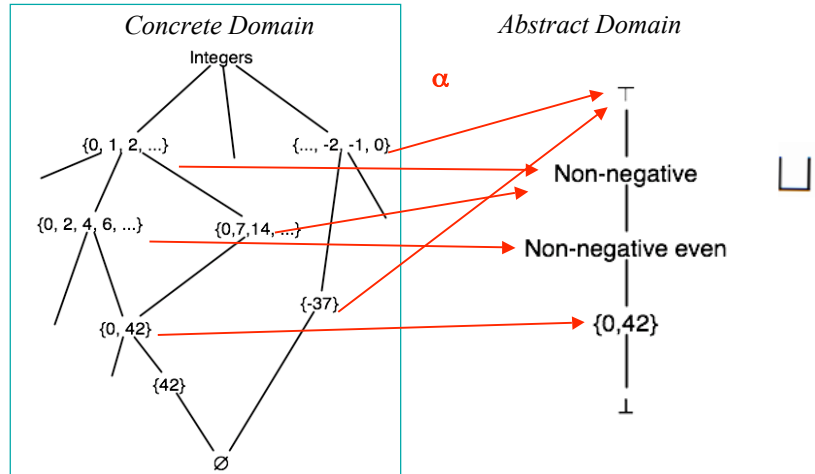
Planet (classification)



6000..7000km (radius)

Abstraction Function

The abstraction function α maps each concrete set within the lattice to the best abstract value.



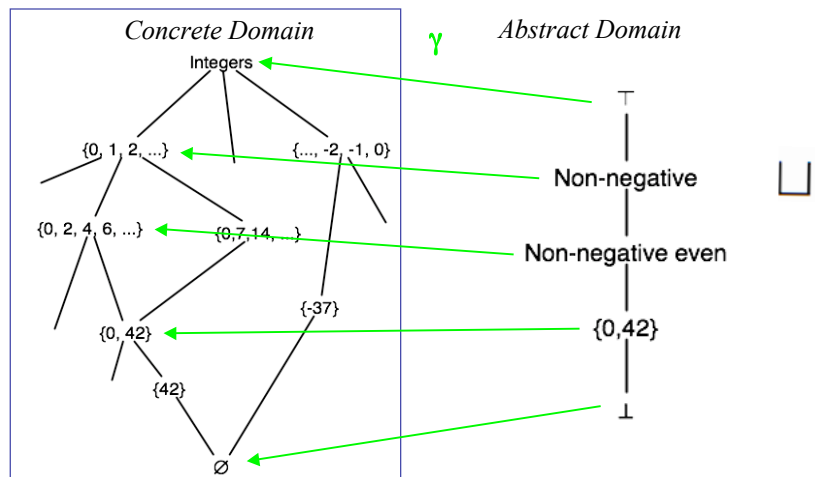
CS653 Lecture

Abstract Interpretation

3

Concretization Function

The concretization function γ maps each abstract value to concrete values it represents.



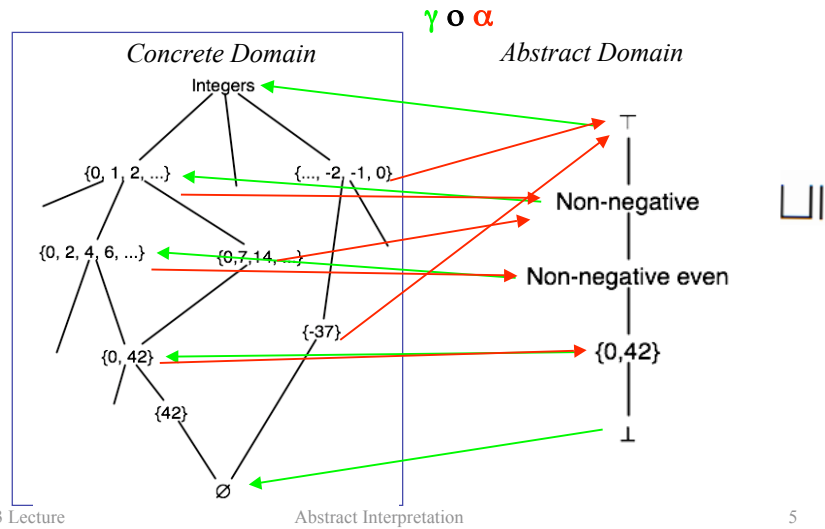
CS653 Lecture

Abstract Interpretation

4

Composing α and γ

Abstraction followed by concretization is sound but imprecise.



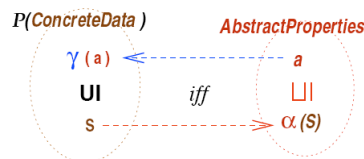
α and γ Form a Galois insertion

α and γ are monotonic

– recall: f is monotonic if $x \leq y$ implies $f(x) \leq f(y)$, order preserving

$S \subseteq \gamma(\alpha(S))$ for any concrete set S

$\alpha(\gamma(A)) = A$ for any abstract element A



Source Language

- Integers and multiplication
 - $e ::= i \mid e * e$
- Standard semantics of the program
 - $\text{Eval} : e \rightarrow \text{Int}$
 - $\text{Eval}(i) = i$
 - $\text{Eval}(e_1 * e_2) = \text{Eval}(e_1) \times \text{Eval}(e_2)$

Abstraction

- Define an abstract semantics that computes only the sign of the result

$\text{AEval} : e \rightarrow \{-, 0, +\}$

$$\text{AEval}(i) = \begin{cases} + & i > 0 \\ 0 & i = 0 \\ - & i < 0 \end{cases}$$

$\text{AEval}(e_1 * e_2) = \text{AEval}(e_1) \times \text{AEval}(e_2)$

\times	+	0	-
+	+	0	-
0	0	0	0
-	-	0	+

Soundness

- We can show our abstraction correctly predicts the sign of an expression
- Proof: by structural induction on e
 - $\text{Eval}(e) > 0$ iff $\text{AEval}(e) = +$
 - $\text{Eval}(e) = 0$ iff $\text{AEval}(e) = 0$
 - $\text{Eval}(e) < 0$ iff $\text{AEval}(e) = -$

Abstraction and Concretization

- Concretization function Υ

$$\Upsilon(\top) = \text{all integers}$$

$$\Upsilon(+)= \{i \mid i > 0\}$$

$$\Upsilon(0) = \{0\}$$

$$\Upsilon(-) = \{i \mid i < 0\}$$

$$\Upsilon(\perp) = \emptyset$$

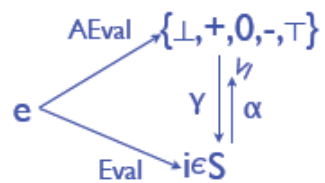
- Abstraction function maps concrete values (sets of integers) to smallest valid abstract element

$$\alpha(S) = \left\{ \begin{array}{l} - \exists i \in S . i < 0 \\ \perp \text{ otherwise} \end{array} \right\} \sqcup \left\{ \begin{array}{l} 0 \exists i \in S . i = 0 \\ \perp \text{ otherwise} \end{array} \right\} \sqcup \left\{ \begin{array}{l} + \exists i \in S . i > 0 \\ \perp \text{ otherwise} \end{array} \right\}$$

Definition

- An abstract interpretation consists of
 - A concrete domain S and an abstract domain A
 - Concretization and abstraction functions that form a Galois insertion [of A into S]
 - A (sound) abstract semantic function
 -
- Recall: α and γ form a Galois insertion if
 - α and γ are monotone
 - $S \subseteq \gamma(\alpha(S))$ or $\text{id} \leq \gamma \circ \alpha$ for any concrete set S
 - $A = \alpha(\gamma(A))$ or $\text{id} = \alpha \circ \gamma$ for any abstract element A

Soundness, Again



- Our abstraction is sound if
 - $\text{Eval}(e) \in \gamma(\text{AEval}(e))$
- Soundness proof: next

Conditions for Correctness

- We can show that if
 - α and γ form a Galois insertion
 - Abstract operations op are locally correct
 - $\gamma(\text{op}(a_1, \dots, a_n)) \supseteq \text{op}(\gamma(a_1), \dots, \gamma(a_n))$
 - Note: We've extended op pointwise to sets
 - I.e., if S and T are sets, $S+T = \{s+t \mid s \in S, t \in T\}$
-
- Then the abstract interpretation is sound

Proof: Show $\text{Eval}(e) \in \gamma(\text{AEval}(e))$

- By structural induction on expressions
 - Base cases: an integer i , so $\text{Eval}(i) = i$
 - if $i < 0$ then $\gamma(\text{AEval}(i)) = \gamma(-) = \{j \mid j < 0\}$
 - Other cases similar
 - Induction: for any operation

$$\begin{aligned} & \text{Eval}(e_1 \text{ op } e_2) \\ &= \text{Eval}(e_1) \text{ op } \text{Eval}(e_2) && \text{by definition of Eval} \\ &\in \gamma(\text{AEval}(e_1)) \text{ op } \gamma(\text{AEval}(e_2)) && \text{by induction} \\ &\subseteq \gamma(\text{AEval}(e_1) \text{ op } \text{AEval}(e_2)) && \text{by local correctness of op} \\ &= \gamma(\text{AEval}(e_1 \text{ op } e_2)) && \text{by definition of AEval} \end{aligned}$$

Widening (see perspectives paper for an example)
