Abstract Interpretation

Yu Zhang

Most content comes from http://cs.au.dk/~amoeller/spa/

http://staff.ustc.edu.cn/~yuzhang/pldpa

Abstract Interpretation

- Abstract Interpretation: a solid mathematical foundation for reasoning about static program analyses
  - Is the analysis sound?
    (Does it safely approximate the actual program behavior?)
  - Is it as precise as possible for the currently used analysis lattice?
    If not, where can precision losses arise?
    Which precision losses can be avoided (without sacrificing soundness)?

➔Require: a precise definition of the semantics of the PL and precise definitions of the analysis abstractions in terms of the semantics

Agenda

- **Collecting semantics**
- Abstraction and concretization
- Soundness
- Optimality
- Completeness

Program Semantics as Constraint Systems

- Concrete state: program variables to integers

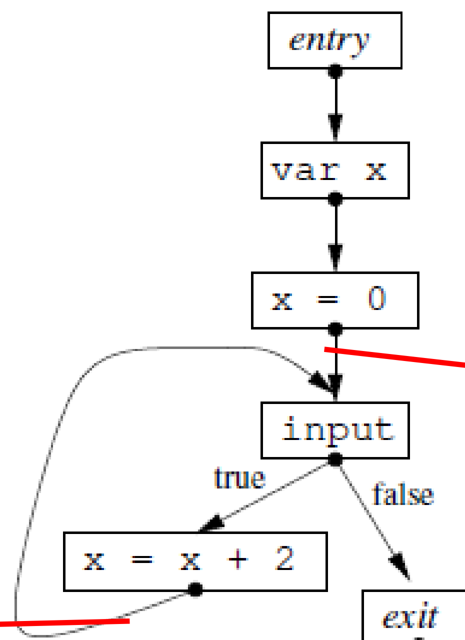
$$ConcreteStates = Vars \to \mathbb{Z}$$

- Constraint variable for each CFG node v

$$[\![v]\!] \subseteq ConcreteStates$$

  - Denote the state at the program point immediately after v

```
var x;
x = 0;
while (input) {
  x = x + 2;
}
```



x是0

状态不唯一
x是正偶数

The Semantics of Expressions

- Concrete execution ➔ Abstract execution

$$ceval: ConcreteStates \times Exp \rightarrow \mathcal{P}(\mathbb{Z})$$

  - A concrete state ρ results in a set of possible integer values

$$ceval(\rho, X) = \{\rho(X)\}$$
$$ceval(\rho, I) = \{I\}$$
$$ceval(\rho, \mathtt{input}) = \mathbb{Z}$$
$$ceval(\rho, E_1 \,\textsf{op}\, E_2) = \{v_1 \,\textsf{op}\, v_2 \mid v_1 \in ceval(\rho, E_1) \;\wedge\; v_2 \in ceval(\rho, E_2)\}$$

- Overload *ceval* to work on sets of concrete states

$$ceval(R, E) = \bigcup_{\rho \in R} ceval(\rho, E)$$

Successors and Joins

- Possible successors of a CFG node relative to a concrete state
  ➔ work on a set of concrete states

$$csucc\colon\ ConcreteStates \times Nodes \to \mathcal{P}(Nodes)$$

$$csucc(R, v) = \bigcup_{\rho \in R} csucc(\rho, v)$$

$$CJOIN(v) =$$

$$\{\rho \in ConcreteStates \mid \exists w \in Nodes\colon \rho \in \{\![w]\!\} \wedge v \in csucc(\rho, w)\}$$

Semantics of Statements

A flow-insensitive analysis that tracks function values:

$$\{\![X{=}E]\!\} = \Big\{ \rho[X \mapsto z] \;\Big|\; \rho \in CJOIN(v) \;\wedge\; z \in ceval(\rho, E) \Big\}$$

$$\begin{aligned} \{\![\mathbf{var}\ X_1\,,\ldots\,,X_n]\!\} = \\ &\big\{ \rho[X_1 \mapsto z_1, \ldots, X_n \mapsto z_n] \;\big|\; \rho \in CJOIN(v) \wedge z_1 \in \mathbb{Z} \wedge \cdots \wedge z_n \in \mathbb{Z} \end{aligned}$$

$$\{\![entry]\!\} = \{[\,]\}$$

$$\{\![v]\!\} = CJOIN(v)$$

The Resulting Constraint System

- A program with n CFG nodes, $v_1, \cdots, v_n$

$$\begin{aligned}
\{\![v_1]\!\} &= cf_{v_1}(\{\![v_1]\!\}, \ldots, \{\![v_n]\!\}) \\
\{\![v_2]\!\} &= cf_{v_2}(\{\![v_1]\!\}, \ldots, \{\![v_n]\!\}) \\
&\vdots \\
\{\![v_n]\!\} &= cf_{v_n}(\{\![v_1]\!\}, \ldots, \{\![v_n]\!\})
\end{aligned}$$

- Combine n functions into one

$$cf : \left(\mathcal{P}(ConcreteStates)\right)^n \to \left(\mathcal{P}(ConcreteStates)\right)^n$$

$$cf(x_1, \ldots, x_n) = \left(cf_{v_1}(x_1, \ldots, x_n), \ldots, cf_{v_n}(x_1, \ldots, x_n)\right)$$

$$x = cf(x)$$

Example

```
var x;
x = 0;
while (input) {
  x = x + 2;
}
```

	solution 1	solution 2
$[\![entry]\!]$	$\{[]\}$	$\{[]\}$
$[\![\texttt{var x}]\!]$	$\{[\mathtt{x} \mapsto z] \mid z \in \mathbb{Z}\}$	$\{[\mathtt{x} \mapsto z] \mid z \in \mathbb{Z}\}$
$[\![\texttt{x = 0}]\!]$	$\{[\mathtt{x} \mapsto 0]\}$	$\{[\mathtt{x} \mapsto 0]\}$
$[\![\texttt{input}]\!]$	$\{[\mathtt{x} \mapsto z] \mid z \in \{0, 2, 4, \dots\}\}$	$\{[\mathtt{x} \mapsto z] \mid z \in \mathbb{Z}\}$
$[\![\texttt{x = x + 2}]\!]$	$\{[\mathtt{x} \mapsto z] \mid z \in \{2, 4, \dots\}\}$	$\{[\mathtt{x} \mapsto z] \mid z \in \mathbb{Z}\}$
$[\![exit]\!]$	$\{[\mathtt{x} \mapsto z] \mid z \in \{0, 2, 4, \dots\}\}$	$\{[\mathtt{x} \mapsto z] \mid z \in \mathbb{Z}\}$

the least solution

A Fixed Point Theorem for Continuous Functions

- $f: L_1 \to L_2$ is continuous if

  $$f(\bigsqcup A) = \bigsqcup_{a \in A} f(a)$$ for every $A \subseteq L$

- If f is continuous

  $$fix(f) = \bigsqcup_{i \geq 0} f^i(\bot)$$

  (even when L has infinite height!)

- cf is continuous

Semantics vs. Analysis

```
var a,b,c;
a = 42;
b = 87;
if (input) {
  c = a + b;
} else {
  c = a - b;
}
```

$$\{\![\texttt{b = 87}]\!\} = \{[\mathsf{a} \mapsto 42, \mathsf{b} \mapsto 87, \mathsf{c} \mapsto z] \mid z \in \mathbb{Z}\}$$
$$\{\![\texttt{c = a - b}]\!\} = \{[\mathsf{a} \mapsto 42, \mathsf{b} \mapsto 87, \mathsf{c} \mapsto -45]\}$$
$$\{\![exit]\!\} = \{[\mathsf{a} \mapsto 42, \mathsf{b} \mapsto 87, \mathsf{c} \mapsto 129], [\mathsf{a} \mapsto 42, \mathsf{b} \mapsto 87, \mathsf{c} \mapsto -45]\}$$

$$[\![\texttt{b = 87}]\!] = [\mathsf{a} \mapsto +, \mathsf{b} \mapsto +, \mathsf{c} \mapsto \top]$$
$$[\![\texttt{c = a - b}]\!] = [\mathsf{a} \mapsto +, \mathsf{b} \mapsto +, \mathsf{c} \mapsto \top]$$
$$[\![exit]\!] = [\mathsf{a} \mapsto +, \mathsf{b} \mapsto +, \mathsf{c} \mapsto \top]$$

Agenda

- Collecting semantics
- **Abstraction and concretization**
- Soundness
- Optimality
- Completeness

Abstract Functions for Sign Analysis

- Abstract functions

$$\alpha_{\mathrm{a}}: \mathcal{P}(\mathbb{Z}) \to \mathit{Sign}$$
$$\alpha_{\mathrm{b}}: \mathcal{P}(\mathit{ConcreteStates}) \to \mathit{States}$$
$$\alpha_{\mathrm{c}}: \big(\mathcal{P}(\mathit{ConcreteStates})\big)^n \to \mathit{States}^n$$

$$\mathit{ConcreteStates} = \mathit{Vars} \to \mathbb{Z}$$
$$\mathit{State} = \mathit{Vars} \to \mathit{Sign}$$

$$\alpha_{\mathrm{a}}(D) = \begin{cases} \bot & \text{if } D \text{ is empty} \\ + & \text{if } D \text{ is nonempty and contains only positive integers} \\ - & \text{if } D \text{ is nonempty and contains only negative integers} \\ 0 & \text{if } D \text{ is nonempty and contains only the integer } 0 \\ \top & \text{otherwise} \end{cases}$$
for any $D \in 2^{\mathbb{Z}}$

$$\alpha_{\mathrm{b}}(R) = \sigma \text{ where } \sigma(X) = \alpha_{\mathrm{a}}(\{\rho(X) \mid \rho \in R\})$$
for any $R \subseteq \mathit{ConcreteStates}$ and $X \in \mathit{Vars}$

$$\alpha_{\mathrm{c}}(R_1, \ldots, R_n) = (\alpha_b(R_1), \ldots, \alpha_b(R_n))$$
for any $R_1, \ldots, R_n \subseteq \mathit{ConcreteStates}$

Concretization Functions for Sign Analysis

- Concretization functions

$$\gamma_{\mathrm{a}}: \mathit{Sign} \to \mathcal{P}(\mathbb{Z})$$
$$\gamma_{\mathrm{b}}: \mathit{States} \to \mathcal{P}(\mathit{ConcreteStates})$$
$$\gamma_{\mathrm{c}}: \mathit{States}^n \to \big(\mathcal{P}(\mathit{ConcreteStates})\big)^n$$

$$\gamma_{\mathrm{a}}(s) = \begin{cases} \emptyset & \text{if } s = \bot \\ \{1, 2, 3, \ldots\} & \text{if } s = + \\ \{-1, -2, -3, \ldots\} & \text{if } s = - \\ \{0\} & \text{if } s = \mathbf{0} \\ \mathbb{Z} & \text{if } s = \top \end{cases}$$
for any $s \in \mathit{Sign}$

$$\gamma_{\mathrm{b}}(\sigma) = \{\rho \in \mathit{ConcreteStates} \mid \rho(X) \in \gamma_{\mathrm{a}}(\sigma(X)) \text{ for all } X \in \mathit{Vars}\}$$
for any $\sigma \in \mathit{States}$

$$\gamma_{\mathrm{c}}(\sigma_1, \ldots, \sigma_n) = (\gamma_{\mathrm{b}}(\sigma_1), \ldots, \gamma_{\mathrm{b}}(\sigma_n))$$
for any $(\sigma_1, \ldots, \sigma_n) \in \mathit{States}^n$

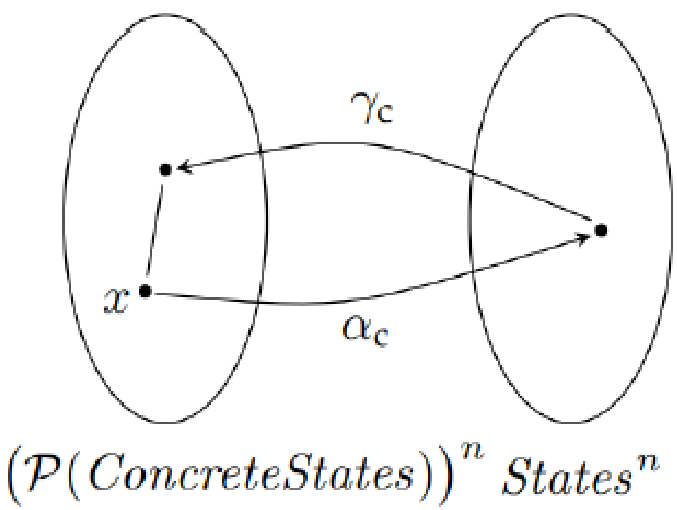
Galois Connections

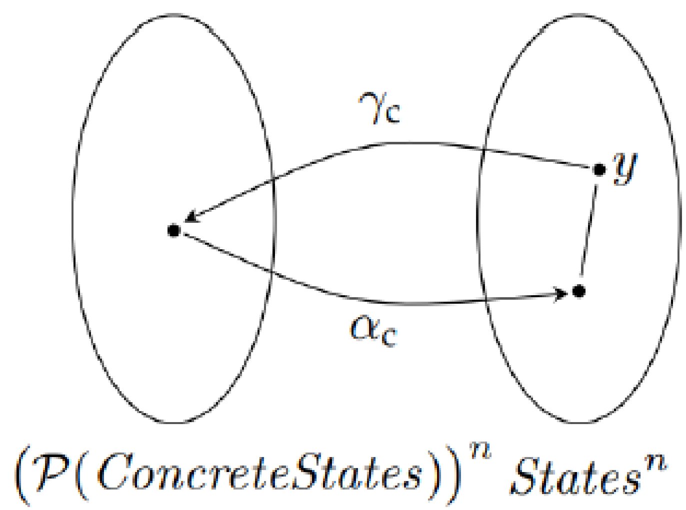
- Galois Theory 伽罗瓦理论
  - 建立域论和群论之间的联系

The pair of monotone functions, α and γ, is called a *Galois connection* if

$\gamma \circ \alpha$ is extensive $\quad x \sqsubseteq \gamma(\alpha(x))$ for all $x \in L_1$

$\alpha \circ \gamma$ is reductive $\quad \alpha(\gamma(y)) \sqsubseteq y$ for all $y \in L_2$





all three pairs of abstraction and concretization functions (α_a, γ_a), (α_b, γ_b), and (α_c, γ_c) from the sign analysis example are Galois connections

1

Galois Connections

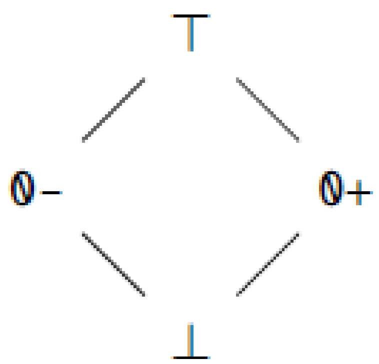
- The concretization function uniquely determines the abstraction function

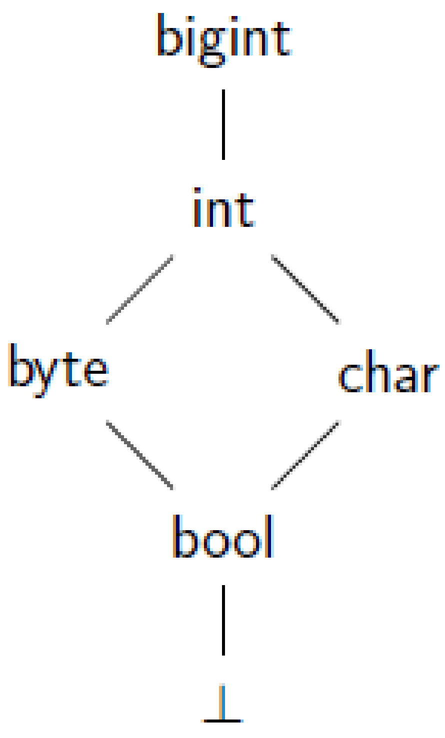
$$\gamma(y) = \bigsqcup_{x \in L_1 \text{ where } \alpha(x) \sqsubseteq y} x$$

$$\alpha(x) = \bigsqcap_{y \in L_2 \text{ where } x \sqsubseteq \gamma(y)} y$$

Galois Connections

- For each of these two lattices, given the “obvious” concretization function, is there an abstraction function such that the concretization function and the abstraction function form a Galois connection?





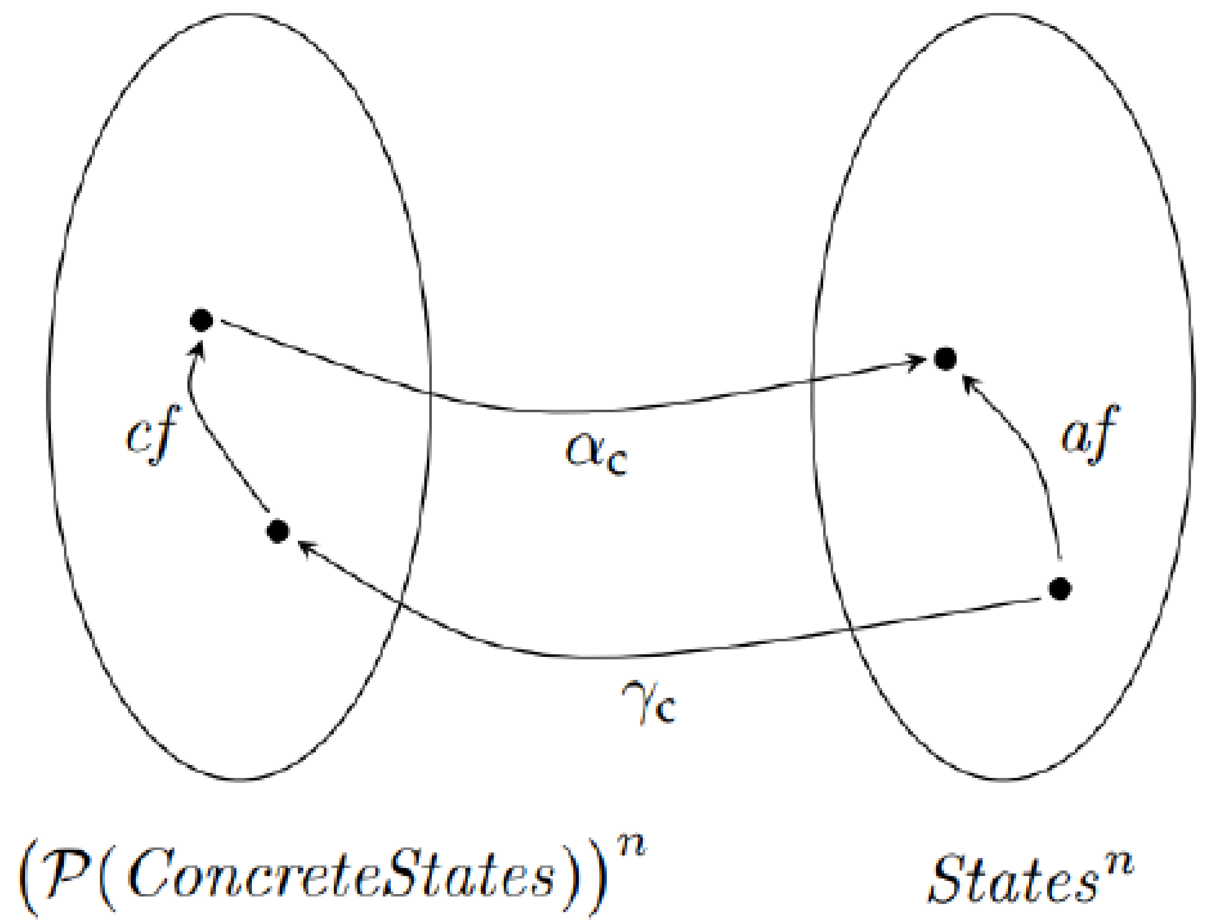
Agenda

- Collecting semantics
- Abstraction and concretization
- Soundness
- **Optimality**
- Completeness

Optimal Approximations

af is an *optimal* approximation of cf if

$$af = \alpha \circ cf \circ \gamma$$



Optimal Approximations in Sign Analysis?

$\widehat{*}$ is optimal:

$$s_1 \widehat{*} s_2 = \alpha_{\mathrm{a}}\big(\gamma_{\mathrm{a}}(s_1) \cdot \gamma_{\mathrm{a}}(s_2)\big)$$

eval is *not* optimal:

$$\sigma(\mathtt{x}) = \top$$

$$eval(\sigma, \mathtt{x\text{-}x}) = \top$$

$$\alpha_{\mathrm{b}}\big(ceval(\gamma_{\mathrm{b}}(\sigma), \mathtt{x\text{-}x})\big) = \mathbb{0}$$

Even if we could make *eval* optimal, the analysis result is not always optimal:

```
x = input;
y = x;
z = x - y;
```

Conclusions

We need

- Static analysis (the analysis lattices and constraint rules)
- Language semantics (suitable collecting semantics)
- Abstract/concretization functions that specify the meaning of the elements in the analysis lattice in terms of the semantic lattice

… and then

- If each constituent of the analysis is a sound abstraction of its semantic counterpart, then the analysis is sound
- If an abstraction is optimal, then it is as precise as possible, relative to the choice of analysis lattice
- If the analysis is sound and complete, then the analysis result is as precise as possible, relative to the choice of analysis lattice