中国科学技术大学
University of Science and Technology of China

Abstract Interpretation

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

张昱

yuzhang@ustc.edu.cn

中国科学技术大学

计算机科学与技术学院

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

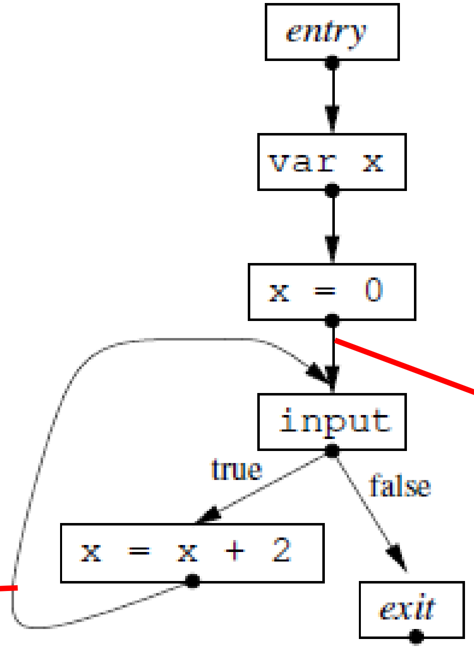
Program Semantics as Constraint Systems

- **Concrete state: program variables to integers**

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

- **Constraint variable for each CFG node v**
  - $[\![v]\!] \subseteq ConcreteStates$ *reachable states collecting semantics*
  - 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

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

- Evaluate E relative to a concrete state ρ, and result in a set of possible integer values

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

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

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

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

Successors and Joins

University of Science and Technology of China

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

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

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

$$CJOIN(v) =$$

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

Semantics of Statements

A flow-insensitive semantics analysis that tracks function values:

- **Assignment**

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

- **Variable declaration**

$$[\![\text{var } X_1, \ldots, X_n]\!] = \{\rho[X_1 \mapsto z_1, \ldots, X_n \mapsto z_n] \mid \rho \in CJOIN(v) \wedge z_1 \in \mathbb{Z} \wedge \cdots \wedge z_n \in \mathbb{Z}$$

- **Initial state** $[\![entry]\!] = \{[]\}$

- **Others** $[\![v]\!] = CJOIN(v)$

The Resulting Semantic Constraint System

University of Science and Technology of China

- 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

$$cf : (\mathcal{P}(ConcreteStates))^n \rightarrow (\mathcal{P}(ConcreteStates))^n$$

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

$$x = cf(x)$$

$$[\![P]\!] = lfp(cf)$$

the semantics of P

lfp:least fixed point

Example

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

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

the least solution

Tarski's 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**

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

(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;
}
```

the semantics of P

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

the analysis result of P

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

Agenda

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

Abstract Functions for Sign Analysis

中国科学技术大学 University of Science and Technology of China

Abstract functions

$$\alpha_a : \mathcal{P}(\mathbb{Z}) \to \mathit{Sign}$$

Sets of Concrete value → sets of abstract value

$$\alpha_b : \mathcal{P}(\mathit{ConcreteStates}) \to \mathit{States}$$

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

$$\alpha_c : (\mathcal{P}(\mathit{ConcreteStates}))^n \to \mathit{States}^n$$

$$\mathit{State} = \mathit{Vars} \to \mathit{Sign}$$

n-tuple of sets of concrete states → those of abstract states

$$\alpha_a(D) = \begin{cases} \bot & \text{if } D \text{ is empty} \\ + & \text{if } D \text{ is nonempty and contains only positive integers} \\ - & \text{if } D \text{ is nonempty and contains only negative integers} \\ 0 & \text{if } D \text{ is nonempty and contains only the integer } 0 \\ \top & \text{otherwise} \end{cases}$$

for any $D \in \mathcal{P}(\mathbb{Z})$

$$\alpha_b(R) = \sigma \text{ where } \sigma(X) = \alpha_a(\{\rho(X) \mid \rho \in R\})$$

for any $R \subseteq \mathit{ConcreteStates}$ and $X \in \mathit{Vars}$

$$\alpha_c(R_1, \ldots, R_n) = (\alpha_b(R_1), \ldots, \alpha_b(R_n))$$

for any $R_1, \ldots, R_n \subseteq \mathit{ConcreteStates}$

Concretization Functions for Sign Analysis

Concretization functions

$$\gamma_a : Sign \to \mathcal{P}(\mathbb{Z})$$
$$\gamma_b : States \to \mathcal{P}(ConcreteStates)$$
$$\gamma_c : States^n \to \big(\mathcal{P}(ConcreteStates)\big)^n$$

$$\gamma_a(s) = \begin{cases} \emptyset & \text{if } s = \bot \\ \{1, 2, 3, \dots\} & \text{if } s = + \\ \{-1, -2, -3, \dots\} & \text{if } s = - \\ \{0\} & \text{if } s = 0 \\ \mathbb{Z} & \text{if } s = \top \end{cases}$$
for any $s \in Sign$

$$\gamma_b(\sigma) = \{\rho \in ConcreteStates \mid \rho(X) \in \gamma_a(\sigma(X)) \text{ for all } X \in Vars\}$$
for any $\sigma \in States$

$$\gamma_c(\sigma_1, \dots, \sigma_n) = (\gamma_b(\sigma_1), \dots, \gamma_b(\sigma_n))$$
for any $(\sigma_1, \dots, \sigma_n) \in States^n$

Galois Connections

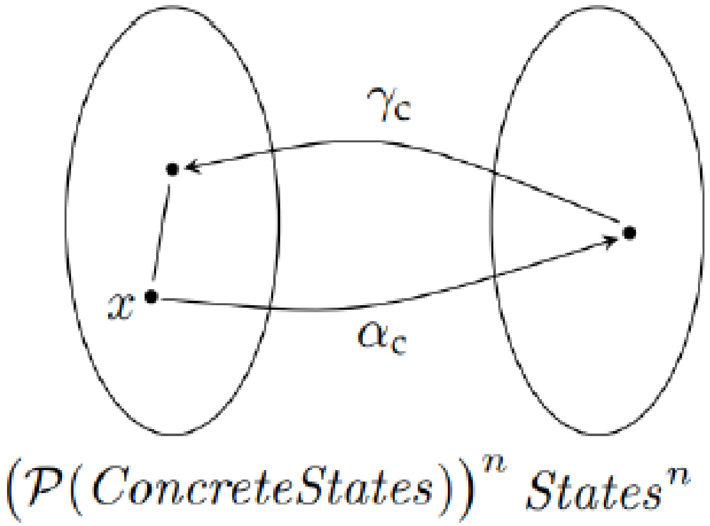
Galois Theory 伽罗瓦理论

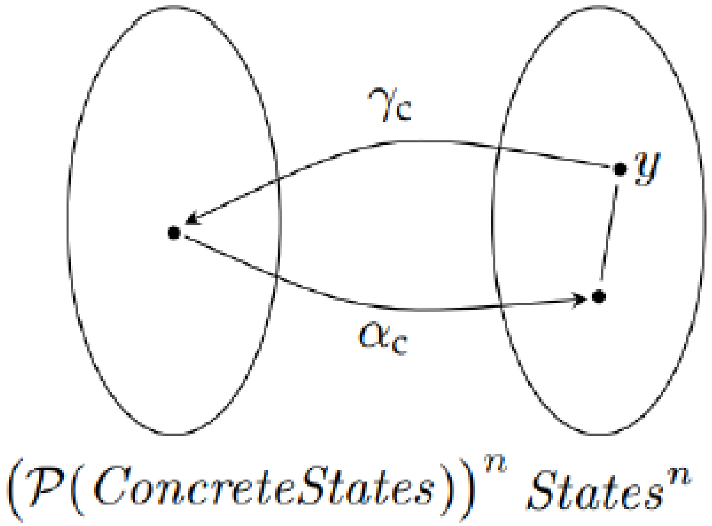
- 建立域论和群论之间的联系：自同构群（Galois 群）

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

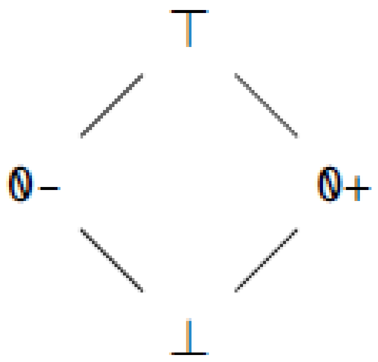
- The concretization function uniquely determines the abstraction function and vice versa:

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

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

Galois Connections

- For this lattice, given the “obvious” concretization function, is there an **abstraction** function such that the concretization function and the abstraction function form a Galois connection? **[Not exist]**



How should we define $\alpha_a(\{0\})$?
Exercise 11.22

the concretization function

$$\gamma_a(s) = \begin{cases} \emptyset & \text{if } s = \bot \\ \{0, 1, 2, 3, \ldots\} & \text{if } s = \mathbf{0}+ \\ \{0, -1, -2, -3, \ldots\} & \text{if } s = \mathbf{0}- \\ \mathbb{Z} & \text{if } s = \top \end{cases}$$

Problem about the Lack of a Galois Connection

University of Science and Technology of China

- Assume $\alpha_a(\{0\}) = \mathbf{0}_-$ and $\alpha_a(\{0,1\}) = \mathbf{0}_+$

```
x = input > 3;
if (!x) {
  output x;
}
```

Flow-insensitive sign analysis

$x \rightarrow \mathbf{0}_+$

$x \rightarrow \mathbf{0}_+$

```
x = input > 3;
if (!x) {
  output x;
}
```

$x \rightarrow \mathbf{0}_-$

Flow-sensitive sign analysis

$x \rightarrow \mathbf{0}_+$

assert(!x): $[\![v]\!] = \begin{cases} \sigma[\mathrm{x} \mapsto \alpha_{\mathrm{a}}(\{0\})] & \text{if } 0 \in \gamma_{\mathrm{a}}(\sigma(\mathrm{x})) \\ \bot & \text{otherwise} \end{cases}$

where $\sigma = JOIN(v)$

Representation Functions

University of Science and Technology of China

For ordinary Sign lattice

$$\beta : \mathbb{Z} \rightarrow Sign$$

Representation Function

$$\beta(d) = \begin{cases} + & \text{if } d > 0 \\ - & \text{if } d < 0 \\ 0 & \text{if } d = 0 \end{cases}$$

Abstraction Function

$$\alpha_{\mathrm{a}}(D) = \bigsqcup\{\beta(d) \mid d \in D\}$$

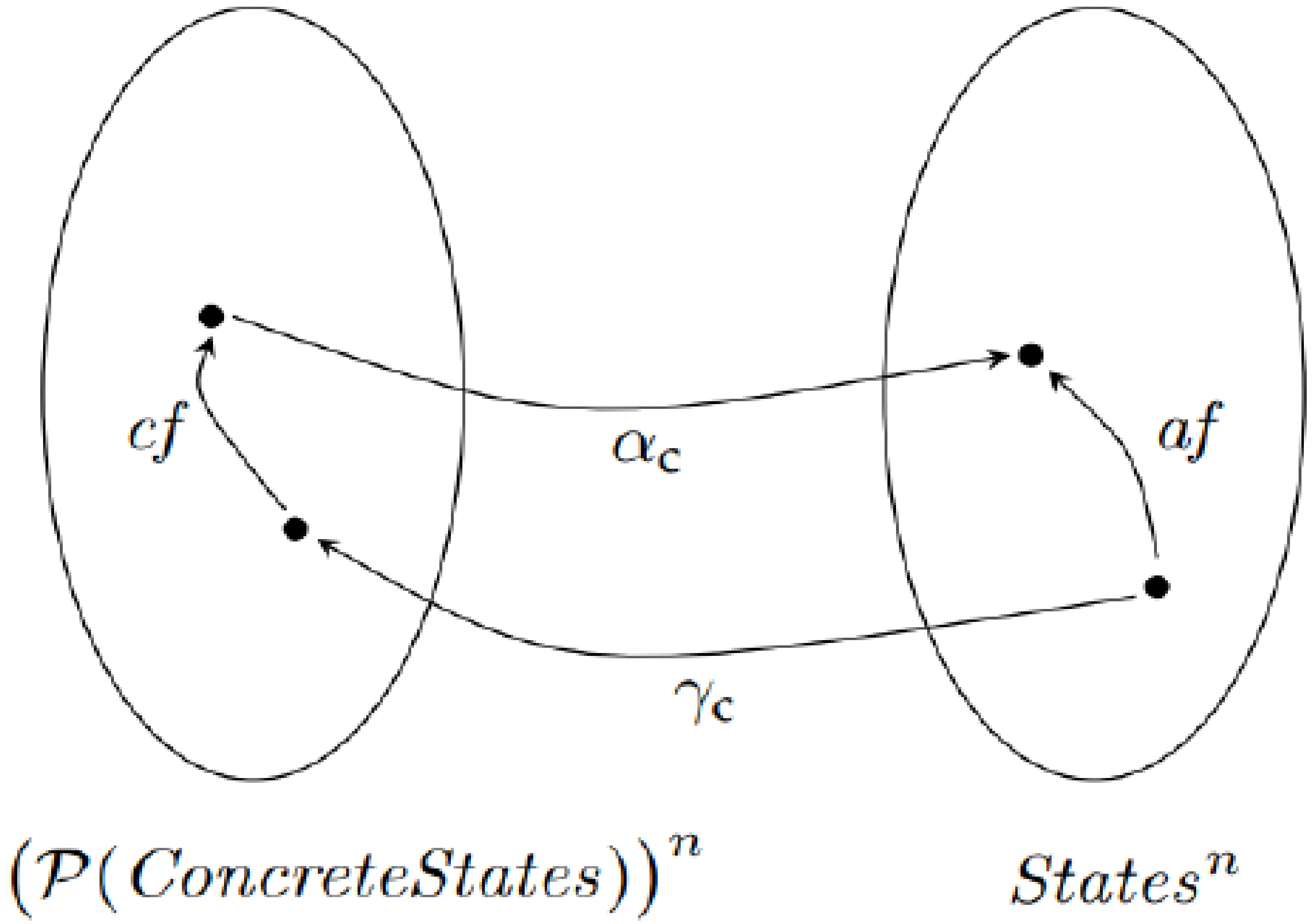
Agenda

- 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:

$$s_1 \widehat{*} s_2 = \alpha_a(\gamma_a(s_1) \cdot \gamma_a(s_2))$$

eval is *not* optimal:

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

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

$$\alpha_b(ceval(\gamma_b(\sigma), \mathbf{x}\text{-}\mathbf{x})) = \mathbf{0}$$

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

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

Conclusions

University of Science and Technology of China

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

Thanks