Abstract interpretation

UW CSE P 504
Today

Abstract interpretation
● Lattices
● Abstraction function
● Concretization function
● Transfer function (vs. lub vs. glb)
● Galois connection
● Exercise: concrete examples
Abstract interpretation (intuition)
Abstract domain and abstraction function (intuition)

Program

```
x = 0;
y = read_even();
x = y + 1;
y = 2 * x;
x = y - 2;
y = x / 2;
```

Abstract domain (even, odd, unk)

```
{x=e}
{x=e; y=e}
```
Transfer function (intuition)

Program

```plaintext
x = 0;
y = read_even();
x = y + 1;
y = 2 * x;
x = y - 2;
y = x / 2;
```

Transfer function corresponds to the “abstract execution” of +
Abstract interpretation (a bit more formal)
Set, semilattice, lattice
Set, semilattice, lattice
Set
Set, semilattice, lattice

Set
- unordered collection of distinct objects
Set, semilattice, lattice

Set
- unordered collection of distinct objects

Partially ordered set
Set, semilattice, lattice

Set
- unordered collection of distinct objects

Partially ordered set
- Binary relationship $\leq$:
  - Reflexive: $x \leq x$
  - Anti-symmetric: $x \leq y \land y \leq x \Rightarrow x = y$
  - Transitive: $x \leq y \land y \leq z \Rightarrow x \leq z$
Set, semilattice, lattice

Set
- unordered collection of distinct objects

Partially ordered set
- Binary relationship ≤:
  - Reflexive: $x \leq x$
  - Anti-symmetric: $x \leq y \land y \leq x \Rightarrow x = y$
  - Transitive: $x \leq y \land y \leq z \Rightarrow x \leq z$

Join semilattice

Meet semilattice
Set, semilattice, lattice

Set
● unordered collection of distinct objects

Partially ordered set
● Binary relationship $\leq$:
  ○ Reflexive: $x \leq x$
  ○ Anti-symmetric: $x \leq y \land y \leq x \Rightarrow x = y$
  ○ Transitive: $x \leq y \land y \leq z \Rightarrow x \leq z$

Meet semilattice
● Partially ordered set with greatest lower bound (meet)
Set, semilattice, lattice

Set
- unordered collection of distinct objects

Partially ordered set
- Binary relationship ≤:
  - Reflexive: x ≤ x
  - Anti-symmetric: x ≤ y ∧ y ≤ x ⇒ x = y
  - Transitive: x ≤ y ∧ y ≤ z ⇒ x ≤ z

Join semilattice
- Partially ordered set with least upper bound (join)

Meet semilattice
- Partially ordered set with greatest lower bound (meet)

Lattice
- Both a join semilattice and a meet semilattice
Lattice: example

Abstract domain: even, odd, unknown, {}
Lattice: example

Abstract domain: even, odd, unknown ($\top$), $\emptyset$ ($\bot$)
Lattice: example

Abstract domain: -, 0, +, unknown, {}
Lattice: example

Abstract domain: -, 0, +, unknown, {}
Abstraction function

Concrete \((P(\mathbb{N}))\)

Abstract

What is the abstraction \((\alpha)\) of \(\{4\}\)?
What is the abstraction ($\alpha$) of \{8\}?
Abstraction function

Concrete \( (P(\mathbb{N})) \)  

Abstract

What is the abstraction (\( \alpha \)) of {}?
Abstraction function

Concrete \((P(\mathbb{N}))\)  

Abstract

Why do we need an abstraction function?
What is the concretization ($\gamma$) of $\bot$?
Concretization function

Concrete ($P(\mathbb{N})$)  Abstract

What is the concretization ($\gamma$) of $E$?
Concretization function

Concrete \((P(\mathbb{N}))\)  

Abstract

\[
\begin{align*}
\{\ldots, 4, 6, 8, \ldots\} & \quad \{1\} \quad \{4\} \quad \{8\} \\
\{\} & \quad \{1\} \quad \{4\} \quad \{8\}
\end{align*}
\]
Transfer function

Concrete state \rightarrow \text{Concrete exec} \rightarrow \text{Concrete state}

\rightarrow \text{Abstract state} \rightarrow \text{Abstract exec} \rightarrow \text{Abstract state}

\rightarrow \text{Concrete state}
Transfer function

Concrete state \rightarrow \text{Abstract state} \rightarrow \text{Concrete state} \rightarrow \text{Abstract state} \rightarrow \text{Concrete state}

Abstraction function \rightarrow \text{Transfer function} \rightarrow \text{Concretization function}
Abstract interpretation: approximation

Do both paths lead to the same abstract state?
Abstract interpretation: approximation

Do both paths lead to the same concrete state?
Abstract interpretation: soundness example

Abstract domain: \{odd, even_2, even_4, is2, unk\}

\[ x = 16 \quad \rightarrow \quad 16 / 2 \]
Abstract interpretation: soundness example

Abstract domain: \{odd, even\_2, even\_4, is2, unk\}

\[x = 16 \quad \xrightarrow{16 / 2} \quad x = 8 \quad \xrightarrow{8 / 2}\]
Abstract interpretation: soundness example

Abstract domain: \{\text{odd}, \text{even}_2, \text{even}_4, \text{is2}, \text{unk}\}
Abstract interpretation: soundness example

Abstract domain: \{odd, \textit{even}_2, \textit{even}_4, is2, unk\}

\[ x = 16 \quad \xrightarrow{16 / 2} \quad x = 8 \quad \xrightarrow{8 / 2} \quad x = 4 \]

\[ x = \text{even}_4 \]
Abstract interpretation: soundness example

Abstract domain: \{odd, even_2, even_4, is2, unk\}

\[
x = 16 \quad \xrightarrow{16 / 2} \quad x = 8 \quad \xrightarrow{8 / 2} \quad x = 4
\]

\[
x = \text{even}_4 \quad \xrightarrow{\text{even}_4 \ / \ ???} \quad \text{even}_4
\]
Abstract interpretation: soundness example

Abstract domain: \{odd, even_2, even_4, is2, unk\}

\[
x=16 \quad \rightarrow \quad 16 / 2 \quad \rightarrow \quad x=8 \quad \rightarrow \quad 8 / 2 \quad \rightarrow \quad x=4
\]
Abstract interpretation: soundness example

Abstract domain: \{odd, even_2, even_4, is2, unk\}

\[
x = 16 \quad 16 / 2 \quad x = 8 \quad 8 / 2 \quad x = 4
\]

\[
x = \text{even}_4 \quad \text{even}_4 / \text{is2} \quad x = \text{even}_2
\]
Abstract interpretation: soundness example

Abstract domain: \{odd, even_{2}, even_{4}, is2, unk\}

\[
x = 16 \quad \xrightarrow{16 \div 2} \quad x = 8 \quad \xrightarrow{8 \div 2} \quad x = 4
\]

\[
x = even_{4} \quad \xrightarrow{even_{4} \div is2} \quad x = even_{2} \quad \xrightarrow{even_{2} \div is2}
\]
Abstract interpretation: soundness example

Abstract domain: \{odd, even_2, even_4, is2, unk\}

\[
\begin{align*}
x &= 16 \\
\quad &\xrightarrow{16 / 2} x &= 8 \\
\quad &\xrightarrow{8 / 2} x &= 4 \\
\quad &\xrightarrow{\text{even}_4 / \text{is2}} x &= \text{even}_2 \\
\quad &\xrightarrow{\text{even}_2 / \text{is2}} x &= \text{unk}
\end{align*}
\]
Abstract interpretation: soundness example

Abstract domain: \{odd, even_2, even_4, is2, unk\}
Abstract interpretation: soundness example

Abstract domain: \{\text{odd, even}_2, \text{even}_4, \text{is2, unk}\}
Abstract interpretation: soundness

What properties must be satisfied by the abstraction, concretization, and transfer functions?
What properties must $\alpha$ and $\gamma$ satisfy?
Sound approximation: galois connection

Concrete ($P(\mathbb{N})$)  

Abstract

Galois connection

- $\alpha : C \rightarrow A$
- $\gamma : A \rightarrow C$
- $\forall c \in C : c \leq \gamma(\alpha(c))$
What properties must the transfer function(s) satisfy?
Sound approximation: consistency

Concrete \((P(\mathbb{N}))\)  
Abstract

Transfer function

- Consistent with concrete execution
  - \(c\): concrete state; \(c' = f_C(c)\)
  - \(a\): \(\alpha(c)\)
  - \(a' = f_A(a)\)
  - \(c'' = \gamma(a')\)
  - \(c' \leq c''\)
Sound approximation: properties

**Transfer function**
- \( f^+_A : A \times A \rightarrow A \)

**Lub**
- \( \text{lub} : A \times A \rightarrow A \)

What properties must the \( \text{lub} \) function satisfy?
Sound approximation: monotonicity

**Transfer function**
- $f^+_A : A \times A \rightarrow A$
- may not be monotone

**Lub**
- $\text{lub} : A \times A \rightarrow A$
- must be monotone

$lub(E, O) = T$
Sound approximation: join (lub) vs. meet (glb)

**Transfer function**
- $f^+_A: A \times A \rightarrow A$
- may not be monotone

**Lub**
- lub: $A \times A \rightarrow A$
- must be monotone

\[
\begin{array}{cccccc}
+ & E & O & T & \ldots \\
E & E & O & T & \\
O & O & E & T & \\
T & T & T & T & \\
\ldots & \\
\end{array}
\]

\[\text{lub}(E, O) = T\]

```java
int x = even();
if (x < 10) {
  x = x + 1;
} else {
  x = x + 2;
}
print(x);
```
Small-group exercise

• Work through two examples:
  ○ Join vs. meet operation ($f(\text{int } a, \text{ int } b, \text{ int } c): \text{ int}$)
    ```java
    if (cond) {
        x = a * b;
    } else {
        x = a * c;
    }
    return(x);
    ```
    Which parameters ($a$, $b$, $c$)
    • will definitely be used?
    • may be used?
    (cond is independent of the parameters)
  ○ Termination/fix point iteration
    ```java
    int x = 2;
    while (x < 10) {
        x = x + 2;
    }
    ```
    Is the value of $x$ after the loop an even number? Use an abstract domain with
    \{odd, 2, even$_2$, and even$_4$\}
Small-group exercise

- Work through two examples:
  - Join vs. meet operation (f(int a, int b, int c): int)
    ```java
    if (cond) {
      x = a * b;
    } else {
      x = a * c;
    }
    return(x);
    ```
    Which parameters (a, b, c) will definitely be used?
    Which parameters (a, b, c) may be used?
    (cond is independent of the parameters)
  - Termination/fix point iteration
    ```java
    int x = 2;
    while (x < 10) {
      x = x + 2;
    }
    ```
    Is the value of x after the loop an even number? Use an abstract domain with {odd, 2, even_2, and even_4}

See Q&A write-up:
https://docs.google.com/document/d/1VEWmFIJVtD2F9ZkXIZ9xeOXGAtkRZATIX13wc1NYmtw
Checker Framework live demo