Abstract Interpretation

- **Semantics-based** approach to program analysis
- Framework to develop **provably correct** and **terminating** analyses

**Ingredients:**

- **Concrete semantics**: Formalizes meaning of a program
- **Abstract semantics**
- Both semantics defined as **fixpoints of monotone functions** over some **domain**
- Relation between the two semantics establishing correctness
Concrete Semantics

Different **semantics** are required for different properties:

- “Is there an execution in which the value of x alternates between 3 and 5?” ➔ Trace Semantics
- “Is the final value of x always the same as the initial value of x?” ➔ “Input/Output” Semantics
- “May x ever assume the value 45 at program point 7?” ➔ Reachability Semantics
Concrete Semantics

- **Trace Semantics**: Captures set of traces of states that the program may execute.
- **Input/Output Semantics**: Captures the pairs of initial and final states of execution traces.
- **Reachability Semantics**: Captures the set of reachable states at each program point
Reachability Semantics

Captures the set of reachable states at each program point. Formally: \( \text{Reach} : V \rightarrow \mathcal{P}(\text{States}) \)

Example:

\( x \in \{ \ldots, -1, 0, 1, 2, \ldots \} \)

\( x \in \{ 0, \ldots, 100 \} \)

\( x \in \{ 101 \} \)
Reachability Semantics

Can be captured as the least solution of:

\[
\begin{align*}
\text{Reach}(\text{start}) &= \text{States} \\
\forall v' \in V \setminus \{\text{start}\} : \text{Reach}(v') &= \bigcup_{v \in V, (v, v') \in E} [\text{labeling}(v, v')] (\text{Reach}(v))
\end{align*}
\]

\[
\begin{align*}
\text{Reach}(1) &= [\text{labeling}(\text{start}, 1)] (\text{Reach}(\text{start})) \cup [\text{labeling}(2, 1)] (\text{Reach}(2)) \\
\text{Reach}(2) &= [\text{labeling}(1, 2)] (\text{Reach}(1)) \\
\text{Reach}(3) &= [\text{labeling}(1, 1)] (\text{Reach}(1)) \\
\text{Reach}(1) &= [x = 0] (\text{Reach}(\text{start})) \cup [x = x + 1] (\text{Reach}(2)) \\
\text{Reach}(2) &= [\text{Pos}(x < 100)] (\text{Reach}(1)) \\
\text{Reach}(3) &= [\text{Neg}(x < 100)] (\text{Reach}(1)) \\
\text{Reach}(1) &= \{0\} \cup \{v + 1 \mid v \in \text{Reach}(2)\} \\
\text{Reach}(2) &= \text{Reach}(1) \cap \{\ldots, 98, 99\} \\
\text{Reach}(3) &= \text{Reach}(1) \cap \{100, 101, \ldots\}
\end{align*}
\]
Questions

- Why the least solution?
- Is there more than one solution?
- Is there a unique least solution?
- Can we systematically compute it?
Answers

- Is there more than one solution? Often
- Is there a unique least solution? Yes
- Can we systematically compute it? Yes and No
Why? Knaster-Tarski Fixpoint Theorem

**Theorem 1** (Knaster-Tarski, 1955).

Assume $(D, \leq)$ is a complete lattice. Then every monotonic function $f : D \to D$ has a least fixed point $d_0 \in D$.

Raises more questions:

- What is a **complete lattice**?
- What is a **monotonic function**?
- What is a **fixed point**?
Monotone Functions

Let \((D, \leq)\) be partially-ordered set.
For example: \(D = \mathbb{N}\) and \(\leq\) the order on natural numbers.

Function \(f : D \rightarrow D\) is monotone (order-preserving) iff for all \(d_1, d_2 \in D\): 
\[d_1 \leq d_2 \Rightarrow f(d_1) \leq f(d_2)\].

Examples:
\[
\begin{align*}
  f(x) &= x & \checkmark \\
  g(x) &= -x & \times \\
  h(x) &= x - 1 & \text{\red \underline{\text{Circle}}}
\end{align*}
\]

\(F(X) = \{f(x) \mid x \in X\}\) \(\times\ \leq\ \Y\)
\(G(X) = \{y \mid x \in X \land (x, y) \in R\}\)

Need to know what the order is.
A binary relation $\leq : D \times D$ is a partial order, iff for all $a, b, c \in D$, we have that:

- $a \leq a$ (reflexivity),

- if $a \leq b$ and $b \leq a$ then $a = b$ (antisymmetry),

- if $a \leq b$ and $b \leq c$ then $a \leq c$ (transitivity).

A set with a partial order is called a partially-ordered set.
The natural numbers ordered by the standard less-than-or-equal relation: \((\mathbb{N}, \leq)\).

The set of subsets of a given set (its powerset) ordered by the subset relation: \((\mathcal{P}(A), \subseteq)\).

The set of subsets of a given set (its powerset) ordered by the subset relation: \((\mathcal{P}(A), \supseteq)\).

The natural numbers ordered by divisibility: \((\mathbb{N}, |)\).
The vertex set $V$ of a directed acyclic graph $G = (V, E)$ ordered by reachability (reflexive, transitive closure of edge relation).

The vertex set $V$ of an arbitrary graph $G = (V, E)$ ordered by reachability.

For a set $X$ and a partially-ordered set $P$, the function space $F : X \rightarrow P$, where $f \leq g$ if and only if $f(x) \leq g(x)$ for all $x$ in $X$.

What about $\text{Reach} : V \rightarrow \mathcal{P}(\text{States})$?

$f \leq g : \iff \forall x \in V. \ f(x) \leq g(x)$
A partially-ordered set \((L, \leq)\) is a complete lattice if every subset \(A\) of \(L\) has both a least upper bound (denoted \(\bigvee A\)) and a greatest lower bound (denoted \(\bigwedge A\)).

**What is an upper bound of a set \(A\)?**

An element \(x\) is an upper bound of a set \(A\) if \(x\) if for every element \(a\) of \(A\), we have \(a \leq x\).

**What is the least upper bound (also: join, supremum) of a set \(A\)?**

\(x\) is the least upper bound of \(A\), denoted \(\bigvee A\), if

1. \(x\) is an upper bound of \(A\),

2. for every upper bound \(y\) of \(A\), we have \(x \leq y\).
Least Upper Bounds: Examples I

<table>
<thead>
<tr>
<th>Partially-ordered set ((D, \leq))</th>
<th>(A \subseteq D)</th>
<th>(\bigcup A)</th>
<th>(\bigcap A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>((\mathbb{N}, \leq))</td>
<td>({1, 2, 3})</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>((\mathbb{R}, \leq))</td>
<td>({x \in \mathbb{R} \mid x &lt; 1})</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>((\mathbb{R}, \leq))</td>
<td>({x \in \mathbb{R} \mid x \leq 1})</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>((\mathbb{Q}, \leq))</td>
<td>({x \in \mathbb{Q} \mid x^2 \leq 2})</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>((\mathbb{N}, \leq))</td>
<td>({x \in \mathbb{N} \mid x \text{ is odd}})</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

Which of these are complete lattices?
Least Upper Bounds: Examples II

<table>
<thead>
<tr>
<th>Partially-ordered set $(D, \leq)$</th>
<th>$A \subseteq D$</th>
<th>$\bigcup A$</th>
<th>$\bigcap A$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(\mathcal{P}(\mathbb{N}), \subseteq)$</td>
<td>${{1, 2}, {2, 4, 5}}$</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>$(\mathcal{P}(\mathbb{N}), \supseteq)$</td>
<td>${{1, 2}, {2, 4, 5}}$</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>$(\mathbb{N},</td>
<td>)$</td>
<td>${3, 4, 5}$</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>$(A \rightarrow \mathbb{N}, \leq)$</td>
<td>${f, g, h}$</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

Which of these are **complete lattices**?
Properties of Complete Lattices

Every complete lattice \((D, \leq)\) has

- a least element (bottom element): \(\bot = \bigcup \emptyset\), and
- a greatest element (top element): \(\top = \bigcup D\).
Generic Lattice Constructions: Power-set Lattice

For any set \( S \), its power set \((\mathcal{P}(S), \subseteq)\) with set inclusion is a lattice:

- "join": \( \bigcup A = \bigcup A \)
- "meet": \( \bigcap A = \bigcap A \)
- "top": \( \top = S \)
- "bottom": \( \bot = \emptyset \)

Graphical representation (Hasse diagram):
Generic Lattice Constructions: Total Function Space

For any set $S$ and lattice $(L, \leq_L)$, the total function space $(S \rightarrow L, \leq)$ is a lattice, with $f \leq g :\iff \forall s \in S : f(x) \leq g(x)$:

- "join": $\bigcup A = \lambda s. \bigcup_{f \in A} f(s)$
- "meet": $\bigwedge A = \lambda s. \bigwedge_{f \in A} f(s)$
- "top": $\top = \lambda s. \top_L$
- "bottom": $\bot = \lambda s. \bot_L$

What about $\text{Reach} : V \rightarrow \mathcal{P}(\text{States})$?
For any set $S$ the flat lattice $(S \cup \{\bot, \top\}, \leq)$ is a lattice, with $a \leq b : \iff a = b \lor a = \bot \lor b = \top$.

**Graphical representation (Hasse diagram) with $S = \mathbb{Z}$:**

... -3 -2 -1 0 1 2 3 ...

...
Fixed Points

A fixed point of a function $f : D \to D$ is an element $x \in D$ with $x = f(x)$.

Example:

$$f : \mathcal{P}([1, 2, 3, 4, 5]) \to \mathcal{P}([1, 2, 3, 4, 5])$$

$$f(X) = \{1, 2, 3\} \cup X$$

**Has multiple fixed points:**

$$\{1, 2, 3\}$$

$$\{1, 2, 3, 4\}$$

$$\{1, 2, 3, 4, 5\}$$

**But a unique least fixed point.**

$$\{1, 2, 3\}$$

The least fixed point $l$, denoted $\text{lfp } f$, of a function $f : D \to D$ over a lattice $(D, \leq)$, is a fixed point of $f$, such that for every fixed point $x$ of $f$: $l \leq x$.
Knaster-Tarski Fixpoint Theorem

Theorem 1 (Knaster-Tarski, 1955).
Assume $(D, \leq)$ is a complete lattice. Then every monotonic function $f : D \to D$ has a least fixed point $d_0 \in D$.

Raises more questions:
- What is a complete lattice? ✓
- What is a monotonic function? ✓
- What is a fixed point? ✓
Can be captured as the **least fixed point** of:

\[
\text{Reach}(\text{start}) = \text{States} \\
\forall v' \in V \setminus \{\text{start}\} : \text{Reach}(v') = \bigcup_{v \in V, (v, v') \in E} \llbracket \text{labeling}(v, v') \rrbracket(\text{Reach}(v))
\]

\[
\begin{align*}
\text{Reach}(1) &= [x = 0](\text{Reach}(\text{start})) \cup [x = x + 1](\text{Reach}(2)) \\
\text{Reach}(2) &= [\text{Pos}(x < 100)](\text{Reach}(1)) \\
\text{Reach}(3) &= [\text{Neg}(x < 100)](\text{Reach}(1))
\end{align*}
\]

\[
\text{Reach}(1) = \{0\} \cup \{v + 1 \mid v \in \text{Reach}(2)\} \\
\text{Reach}(2) = \text{Reach}(1) \cap \{\ldots, 98, 99\} \\
\text{Reach}(3) = \text{Reach}(1) \cap \{100, 101, \ldots\}
\]

\[\mathcal{T}(x) = \bigcap \{f(x) \mid x \in \mathbb{N}\}\]
How to Compute the Least Fixed Point

Kleene Iteration:
\[ \bot \leq f(\bot) \leq f^2(\bot) \leq f^3(\bot) \leq \ldots \]

Why is this increasing?
Will this reach the fixed point?
It will here:
But in general?

start
1 3
\( x = 0 \)
Pos(false) \( \rightarrow \) 2
\( x = x + 1 \)

start
1 3
\( x = 0 \)
Pos(true) \( \rightarrow \) 2
\( x = x + 1 \)

No!

Lattice has infinite ascending chains.
Ascending Chain Condition

A partially-ordered set $S$ satisfies the *ascending chain condition* if every strictly ascending sequence of elements is finite.

→ Length of longest ascending chain determines worst-case complexity of Kleene Iteration.

**Power set lattice**

**Flat lattice**

$(\mathcal{P}(S), \subseteq)$  $|S| + 1$  How about total function space lattice?
Recap: Abstract Interpretation

- **Semantics-based** approach to program analysis
- Framework to develop *provably correct* and *terminating* analyses

**Ingredients:**
- **Concrete semantics:** Formalizes meaning of a program
- **Abstract semantics**
- Both semantics defined as *fixpoints of monotone functions* over some domain
- Relation between the two semantics establishing correctness
Abstract Semantics

Similar to concrete semantics:

- A complete lattice \((L^#, \leq)\) as the domain for abstract elements
- A monotone function \(F^#\) corresponding to the concrete function \(F\)
- Then the abstract semantics is the least fixed point of \(F^#\), \(\text{lfp } F^#\)

If \(F^#\) “correctly approximates” \(F\),
then \(\text{lfp } F^#\) “correctly approximates” \(\text{lfp } F\).
An Example Abstract Domain for Values of Variables

How to relate the two?

- **Concretization function**, specifying “meaning” of abstract values.
  \[ \gamma : \mathbb{Z}^\perp \rightarrow \mathcal{P}(\mathbb{Z}) \]

- **Abstraction function**: determines best representation concrete values.
  \[ \alpha : \mathcal{P}(\mathbb{Z}) \rightarrow \mathbb{Z}^\perp \]
Relation between Abstract and Concrete

\[ \gamma(\top) := \mathbb{Z} \]
\[ \gamma(\bot) := \emptyset \]
\[ \gamma(x) := \{x\} \]

\[ \alpha(A) := \begin{cases} 
\top & : |A| \geq 2 \\
x & : A = \{x\} \\
\bot & : A = \emptyset 
\end{cases} \]

Are these functions monotone?
Why should they be?
What is the meaning of the partial order in the abstract domain?
What if we first abstract and then concretize?

\[ \gamma(\alpha(A)) \supseteq A \]
How to Compute in the Abstract Domain

Example: Multiplication on Flat Lattice

<table>
<thead>
<tr>
<th></th>
<th>$\top$</th>
<th>$a$</th>
<th>$0$</th>
<th>$\bot$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\top$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$a$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$0$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\bot$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Denotes abstract version of operator
How to Compute in the Abstract Domain? Formally

*Local Correctness Condition:*

Correct by construction (if concretization and abstraction have certain properties):
From Local to Global Correctness
Fixpoint Transfer Theorem

Let \((L, \leq)\) and \((L^\#, \leq^\#)\) be two lattices, \(\gamma : L^\# \to L\) a monotone function, and \(F : L \to L\) and \(F^\# : L^\# \to L^\#\) two monotone functions, with

\[
\forall l^\# \in L^\# : \gamma(F^#(l^#)) \geq F(\gamma(l^#)).
\]

Then:

\[
lfp F \leq \gamma(lfp F^#).
\]

\[
x^\# = \text{lfp } F^#
\]

\[
\gamma(x^\#) = \gamma(F^#(x^\#)) \geq F(\gamma(x^\#))
\]