Static Program Analysis

Foundations of Abstract Interpretation

Sebastian Hack, Christian Hammer, Jan Reineke

Advanced Lecture, Winter 2014/15

Abstract Interpretation

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

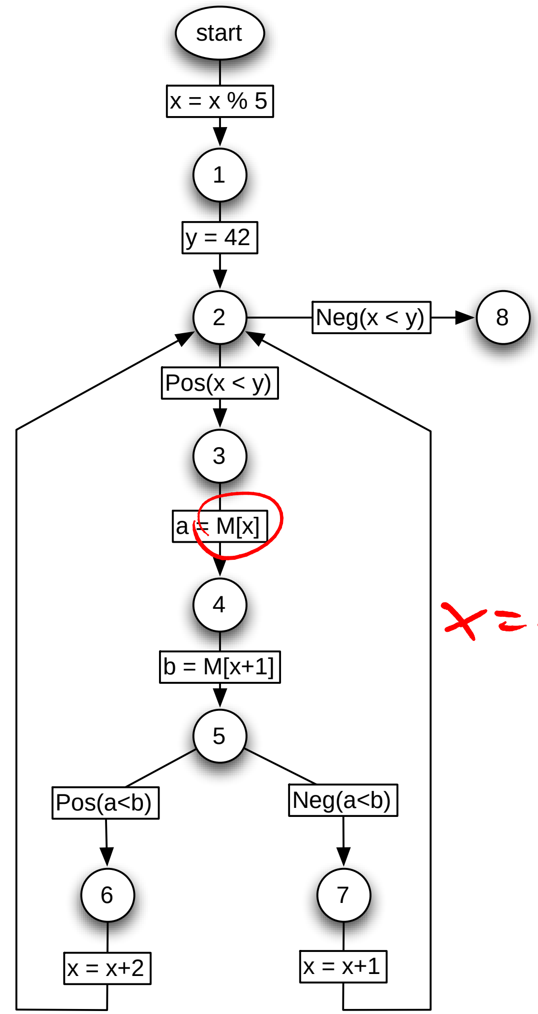
Ingredients:

- 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

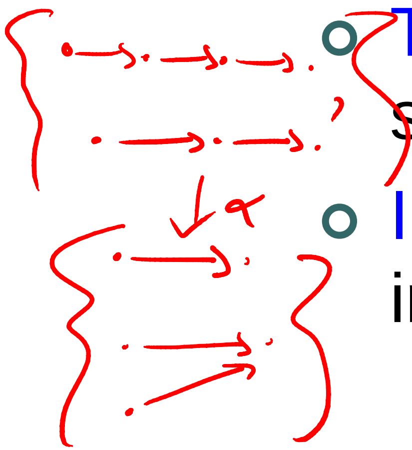
Different semantics are required for different properties:

- “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



$x = \frac{x}{7}$

Concrete Semantics



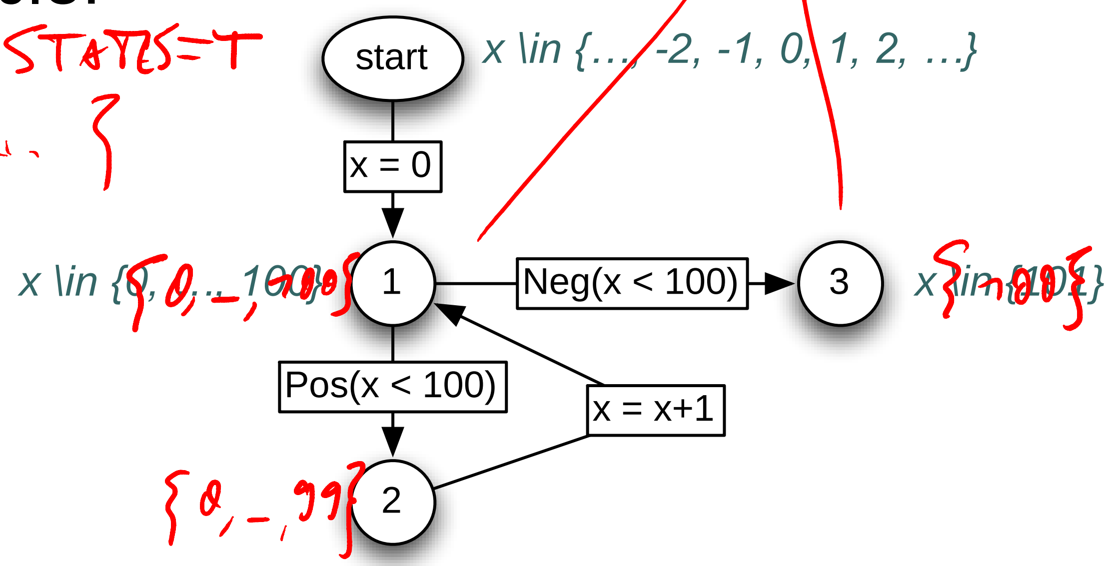
- **Trace Semantics**: Captures set of traces of states that the program may execute.
- **Input/Output Semantics**: Captures the pairs of initial and final states of execution traces.
  - Abstraction of Trace Semantics
- **Reachability Semantics**: Captures the set of reachable states at each program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

Reachability Semantics

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

Example:

{…-3, -2, -1, 0, STATES = T
…1, 2, 3, …}



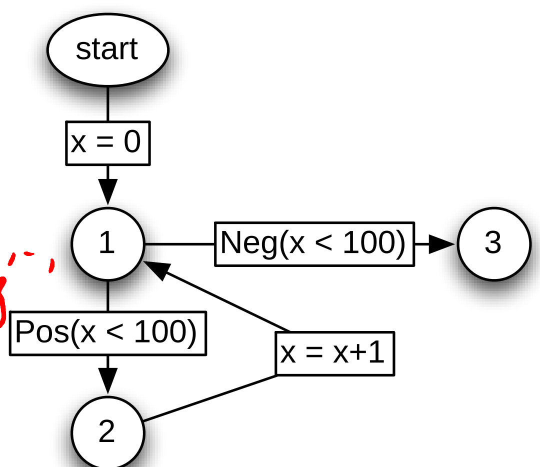
Reachability Semantics

Can be captured as the least solution of:

$$Reach(start) = States$$

$$\forall v' \in V \setminus \{start\} : Reach(v') = \bigcup_{v \in V, (v,v') \in E} [\![labeling(v, v')]\!](Reach(v))$$

{..., -1, ..., 100}



{..., -1, ..., 99}

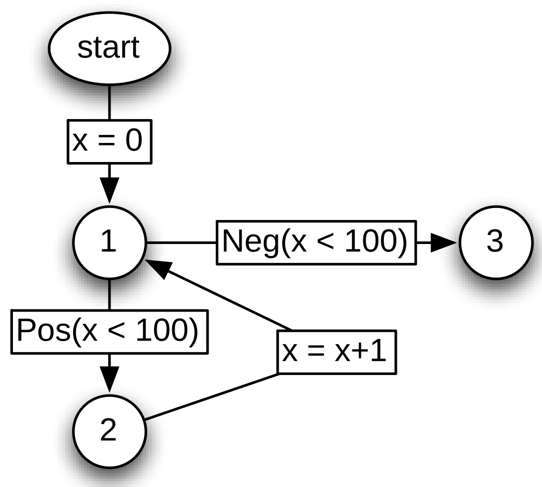
$$Reach(1) = [\![labeling(start, 1)]\!](Reach(start)) \cup [\![labeling(2, 1)]\!](Reach(2))$$
$$Reach(2) = [\![labeling(1, 2)]\!](Reach(1))$$
$$Reach(3) = [\![labeling(1, 3)]\!](Reach(1))$$

$$Reach(1) = [\![x = 0]\!](Reach(start)) \cup [\![x = x + 1]\!](Reach(2))$$
$$Reach(2) = [\![Pos(x < 100)]\!](Reach(1))$$
$$Reach(3) = [\![Neg(x < 100)]\!](Reach(1))$$

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

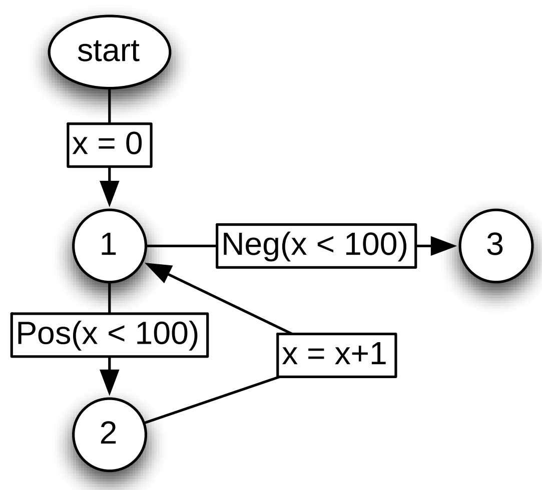
Questions

- 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 $f : D \to D$ 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:

$$f(x) = x$$ ✓

$$g(x) = -x$$ ✗

$$h(x) = x - 1$$

$$F(X) = \{f(x) \mid x \in X\}$$ $X \subseteq Y$

$$G(X) = \{y \mid x \in X \wedge (x, y) \in R\}$$

Which of these are monotone?

Need to know what the order is.

Partial Orders

A binary relation $\leq : D \times D$ is a *partial order*, iff for all $a, b, c \in D$, we have that:

rat

- $a \leq a$ (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*.

Partial Orders: Examples I

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*: $(\mathbb{N}, |)$.

3 | 6
3 | 3

Partial Orders: Examples II

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 $F : X \to P$, where $f \leq g$ if and only if $f(x) \leq g(x)$ for all x in X.

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

$f \leq g :\Leftrightarrow \forall x \in V.\ f(x) \leq g(x)$

Complete Lattices

A partially-ordered set $(L, \leq)$ is a *complete lattice* if every subset A of L has both a *least upper bound* (denoted $\bigsqcup A$) and a *greatest lower bound* (denoted $\bigsqcap A$).

$\vee A$

$\wedge A$

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 $\bigsqcup A$, 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

$Partially\text{-}ordered\ set\ (D, \leq)$	$A \subseteq D$	$\bigsqcup A$	$\bigsqcap A$
$(\mathbb{N}, \leq)$	$\{1, 2, 3\}$	?	?
$(\mathbb{R}, \leq)$	$\{x \in \mathbb{R} \mid x < 1\}$	? 1	$\cancel{?}$
$(\mathbb{R}, \leq)$	$\{x \in \mathbb{R} \mid x \leq 1\}$	? 1	$\cancel{?}$
$(\mathbb{Q}, \leq)$	$\{x \in \mathbb{Q} \mid x^2 \leq 2\}$	$\cancel{\sqrt{2}}$?	?
$(\mathbb{N}, \leq)$	$\{x \in \mathbb{N} \mid x\ is\ odd\}$	?	?

Which of these are complete lattices?

Least Upper Bounds: Examples II

$Partially\text{-}ordered\ set\ (D, \leq)$	$A \subseteq D$	$\bigsqcup A$	$\bigsqcap A$
$(\mathcal{P}(\mathbb{N}), \subseteq)$ ✓	$\{\{1,2\},\{2,4,5\}\}$	?	?
$(\mathcal{P}(\mathbb{N}), \supseteq)$ ✓	$\{\{1,2\},\{2,4,5\}\}$	?	?
$(\mathbb{N}, \mid)$	$\{3,4,5\}$	? 60	? 1
$(A \to \mathbb{N}, \leq)$	$\{f, g, h\}$	?	?

Which of these are complete lattices?

Properties of Complete Lattices

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

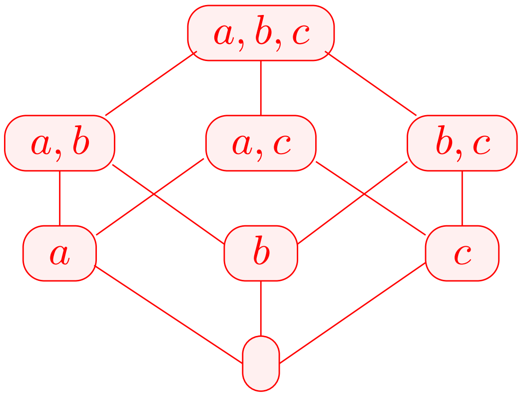
- a *least* element (*bottom* element): $\bot = \bigsqcup \emptyset$, and
- a *greatest* element (*top* element): $\top = \bigsqcup D$.

Generic Lattice Constructions: Power-set Lattice

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

$$\begin{array}{rccc} \textit{"join":} & \bigsqcup A & = & \bigcup A \\ \textit{"meet":} & \bigsqcap A & = & \bigcap A \\ \textit{"top":} & \top & = & S \\ \textit{"bottom":} & \bot & = & \emptyset \end{array}$$

Graphical representation (Hasse diagram):



Generic Lattice Constructions: Total Function Space

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

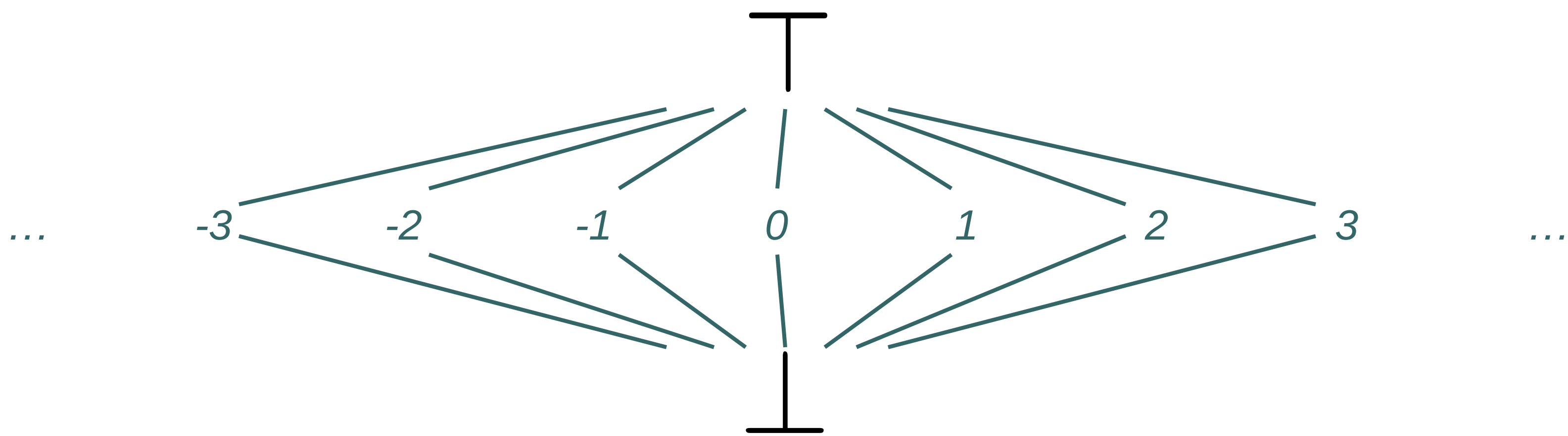
$$\begin{array}{rccc} \text{“}\textit{join}\text{”:} & \bigsqcup A & = & \lambda s. \bigsqcup_{f \in A} f(s) \\ \text{“}\textit{meet}\text{”:} & \bigsqcap A & = & \lambda s. \bigsqcap_{f \in A} f(s) \\ \text{“}\textit{top}\text{”:} & \top & = & \lambda s. \top_L \\ \text{“}\textit{bottom}\text{”:} & \bot & = & \lambda s. \bot_L \end{array}$$

What about $\mathit{Reach} : V \to \mathcal{P}(\mathit{States})$*?*

Generic Lattice Constructions: Flat Lattice

For any set S the flat lattice $(S \cup \{\perp, \top\}, \leq)$ is a lattice, with $a \leq b :\Leftrightarrow a = b \vee a = \perp \vee b = \top$.

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



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 *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? ✓

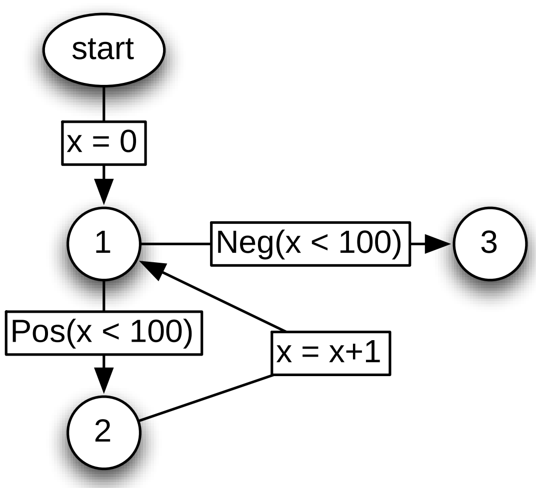
$F(X) = \{ f(x) \mid x \in d \}$

Back to the Reachability Semantics

Can be captured as the least fixed point of:

$$Reach(start) = States$$

$$\forall v' \in V \setminus \{start\} : Reach(v') = \bigcup_{v \in V, (v,v') \in E} [\![labeling(v,v')]\!](Reach(v))$$



$$Reach(1) = [\![x = 0]\!](Reach(start)) \cup [\![x = x + 1]\!](Reach(2))$$
$$Reach(2) = [\![Pos(x < 100)]\!](Reach(1))$$
$$Reach(3) = [\![Neg(x < 100)]\!](Reach(1))$$

Monotone?

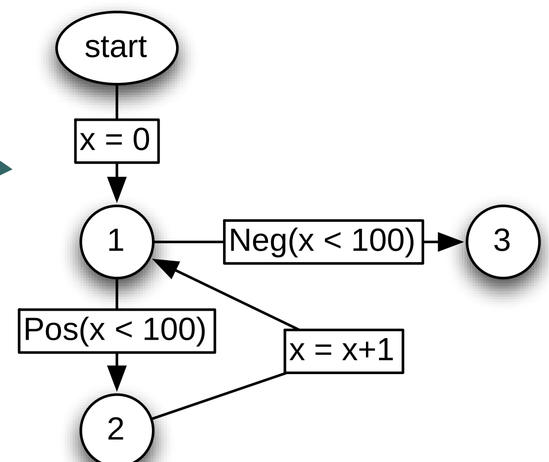
$$Reach(1) = \{0\} \cup \{v + 1 \mid v \in Reach(2)\}$$
$$Reach(2) = Reach(1) \cap \{\dots, 98, 99\}$$
$$Reach(3) = Reach(1) \cap \{100, 101, \dots\}$$

How to Compute the Least Fixed Point

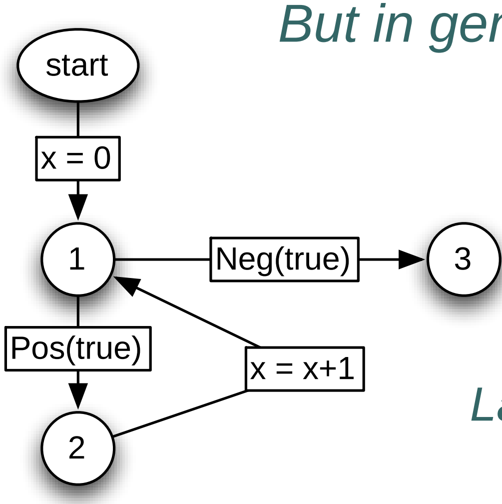
Kleene Iteration:

$$\bot \leq f(\bot) \leq f^2(\bot) \leq f^3(\bot) \leq \dots$$

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



But in general?



No!

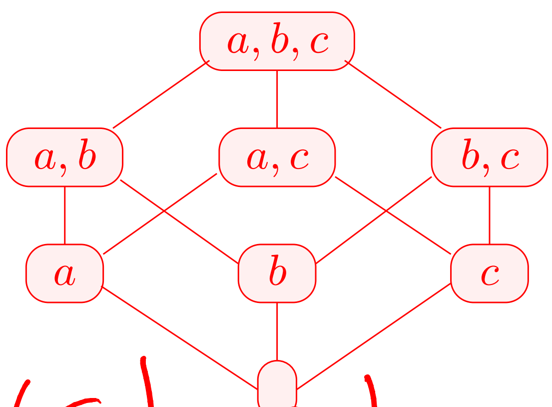
Lattice has infinite ascending chains.

Ascending Chain Condition

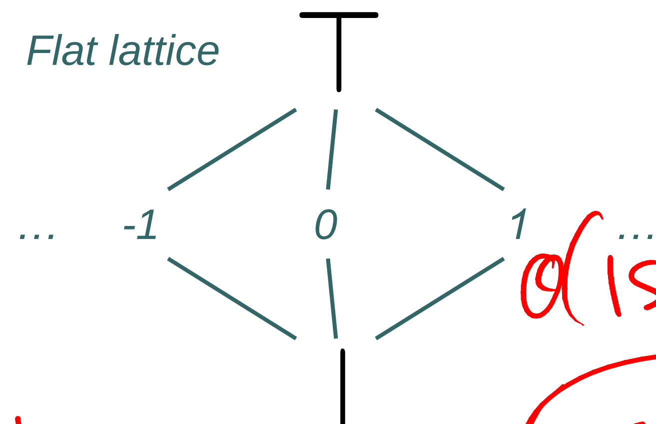
A partially-ordered set S satisfies the *ascending chain condition* if every strictly ascending sequence of elements is finite.

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

Power set lattice



Flat lattice



$\mathcal{O}(|S| \cdot \text{length}(L))$

$(P(S), \subseteq)$

$|S|+1$

$S \to L$

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 $F^\#$, lfp $F^\#$

If $F^\#$ "correctly approximates" F,

then lfp $F^\#$ "correctly approximates" lfp F.

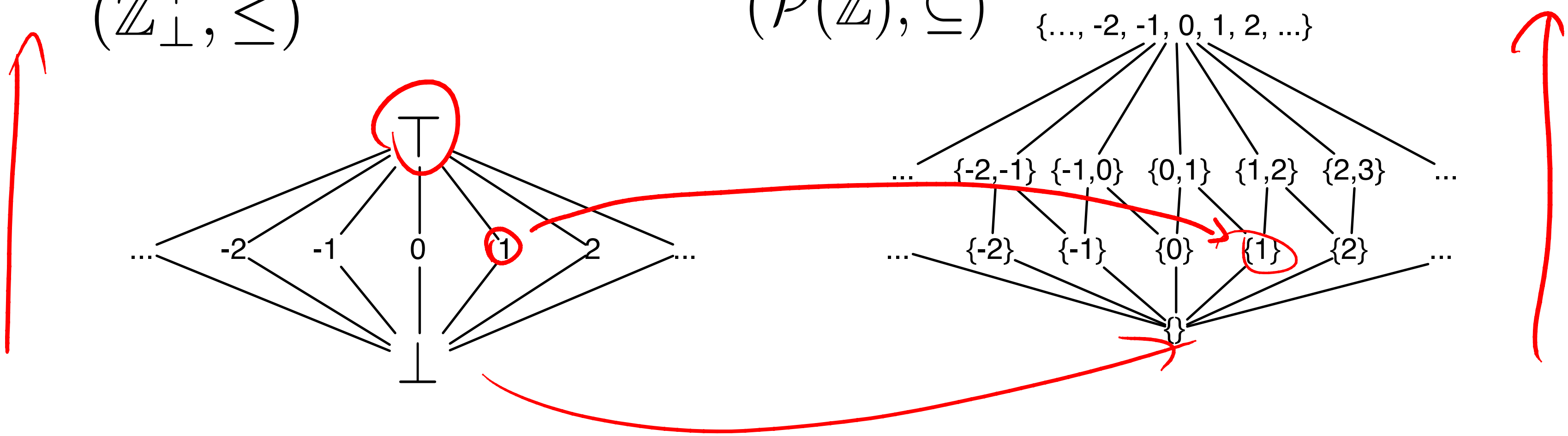
An Example Abstract Domain for Values of Variables

ABSTRACT

CONCRETE

$(\mathbb{Z}_{\perp}^{\top}, \leq)$

$(\mathcal{P}(\mathbb{Z}), \subseteq)$



How to relate the two?

➔ *Concretization function, specifying "meaning" of abstract values.*

$$\gamma : \mathbb{Z}_{\perp}^{\top} \rightarrow \mathcal{P}(\mathbb{Z})$$

➔ *Abstraction function: determines best representation concrete values.*

$$\alpha : \mathcal{P}(\mathbb{Z}) \rightarrow \mathbb{Z}_{\perp}^{\top}$$

Relation between Abstract and Concrete

$$
\begin{aligned}
\gamma(\top) &:= \mathbb{Z} \\
\gamma(\bot) &:= \emptyset \\
\gamma(x) &:= \{x\}
\end{aligned}
\qquad
\alpha(A) := \begin{cases} \top & : |A| \geq 2 \\ x & : A = \{x\} \\ \bot & : A = \emptyset \end{cases}
$$

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?

$\boxed{\gamma(\alpha(A)) \supseteq A}$

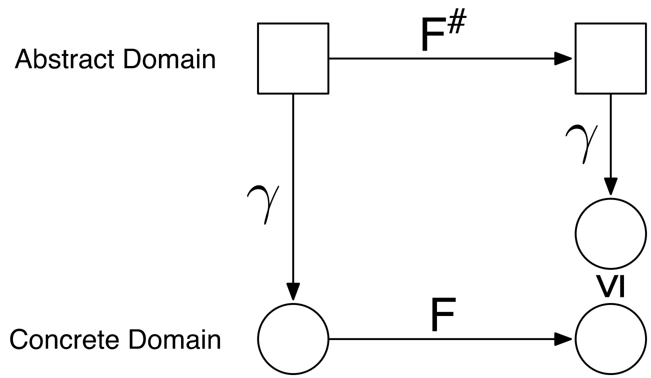
How to Compute in the Abstract Domain
Example: Multiplication on Flat Lattice

Denotes abstract version of operator

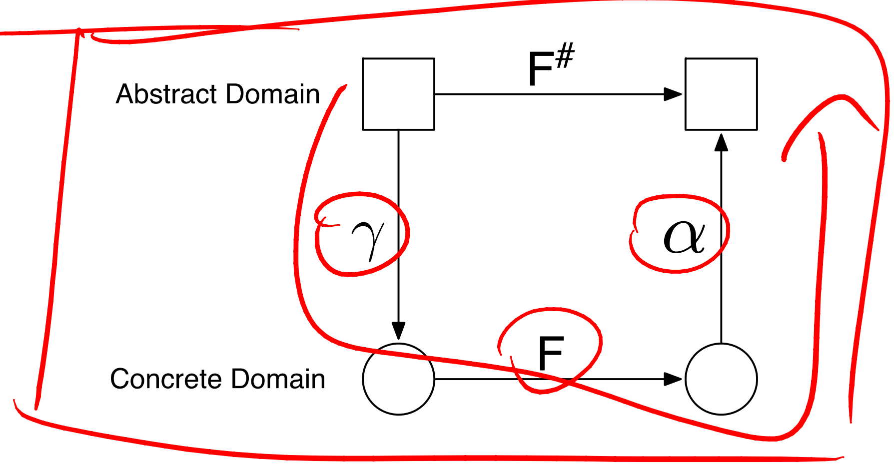
$*^{\#}$	$\top$	a	0	$\bot$
$\top$				
b				
0				
$\bot$				

How to Compute in the Abstract Domain? Formally

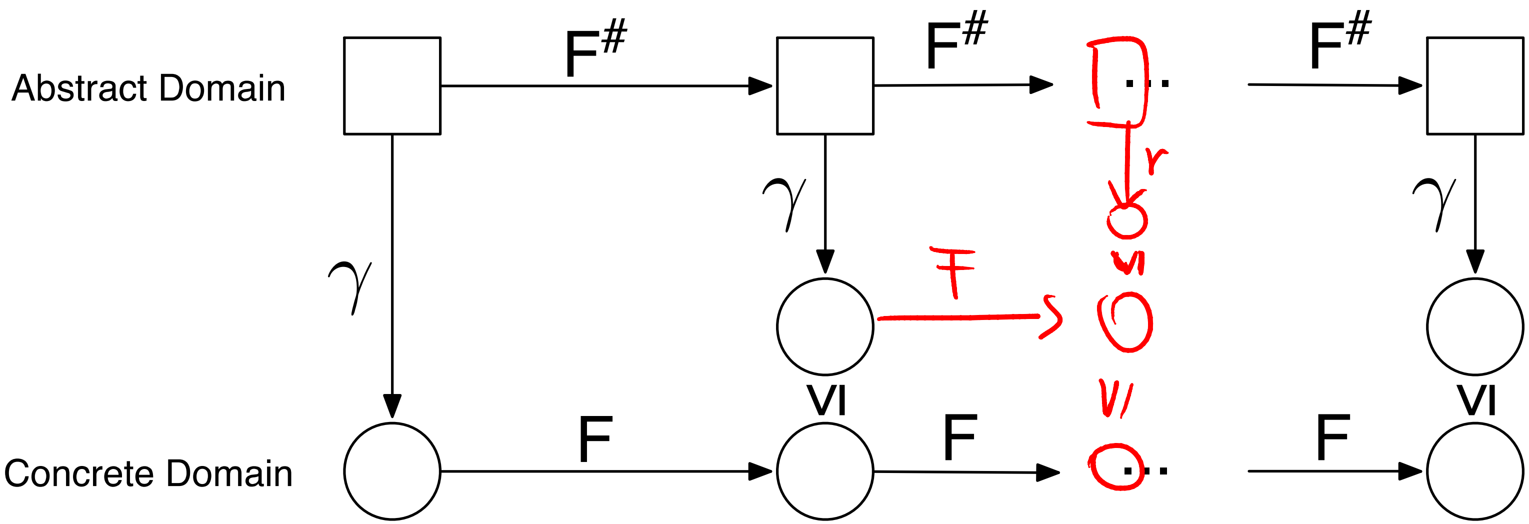
Local Correctness Condition:



Correct by construction
(if concretization and abstraction have certain properties):



From Local to Global Correctness



Fixpoint Transfer Theorem

COMPLETE

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

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

Then:

$$\boxed{lfp\ F \leq \gamma(lfp\ F^\#).}$$

$x^\# = lfp\ F^\#$

$F^\#(x^\#) = x^\#$

$\gamma(x^\#) = \gamma(F^\#(x^\#)) \geq F(\gamma(x^\#))$