

Introduction to Program Analysis

5. Abstract Interpretation

Kihong Heo

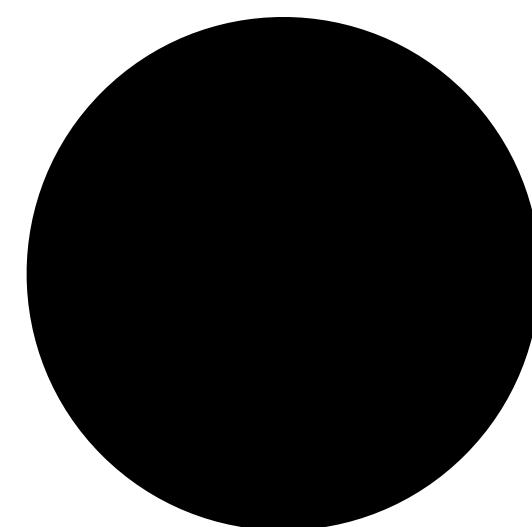


Abstract Interpretation

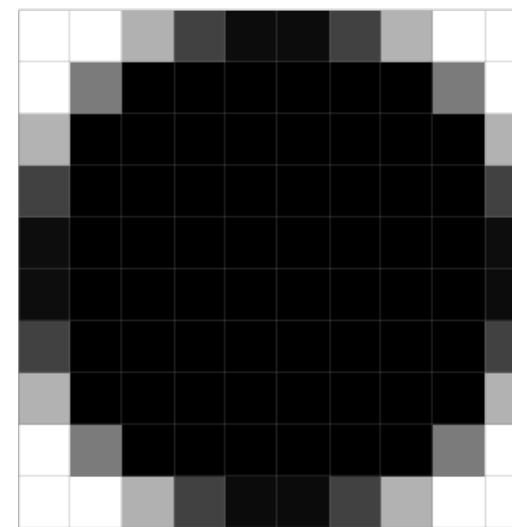
- A **powerful framework** for designing **correct** static analysis
 - Framework: given some inputs, a static analysis comes out
 - Powerful: all static analyses are understood in this framework (e.g., type systems, data-flow analysis, etc)
 - Correct: mathematically proven
- Established by Patrick and Radhia Cousot
 - *Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints*, 1977
 - *Systematic Design of Program Analysis Frameworks*, 1979

Abstract?

- Concrete (execution, dynamic) vs Abstract (analysis, static)
- Without abstraction, it is undecidable to subsume all possible behavior of SW
 - Recall the Rice's theorem and the Halting problem



Concrete



Abstract

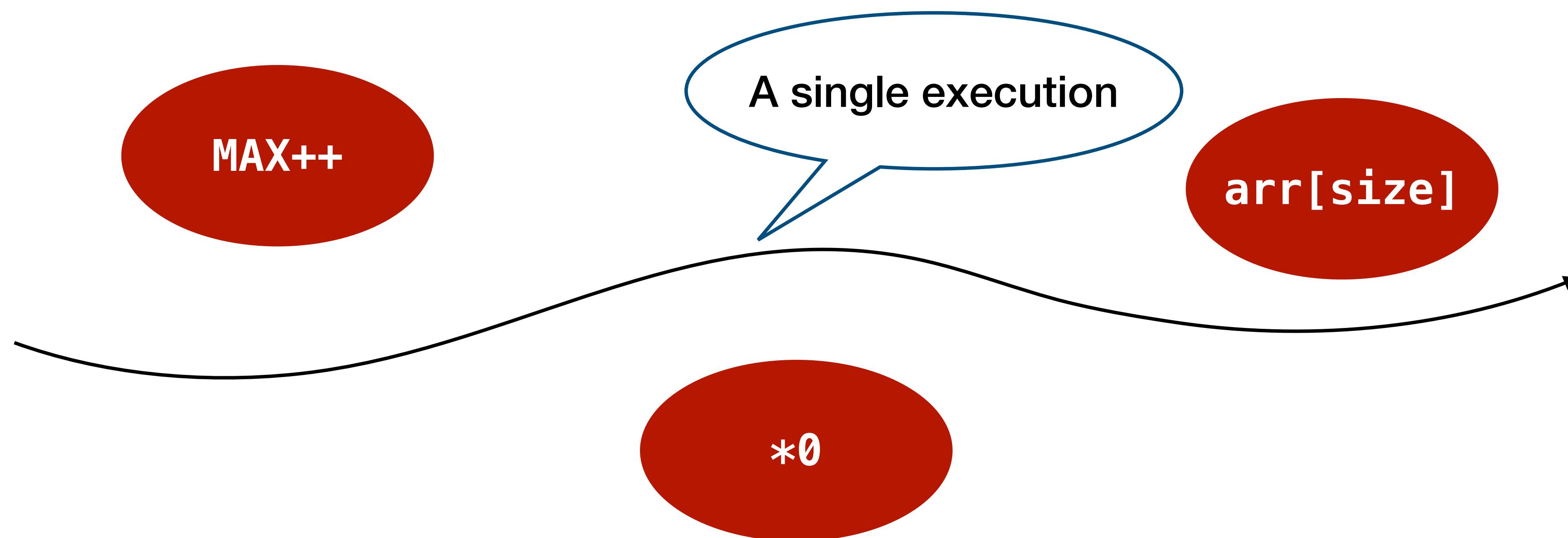
Example

```
x = 3;  
while (*) {  
    x += 2;  
}  
x -= 1;  
print(x);
```

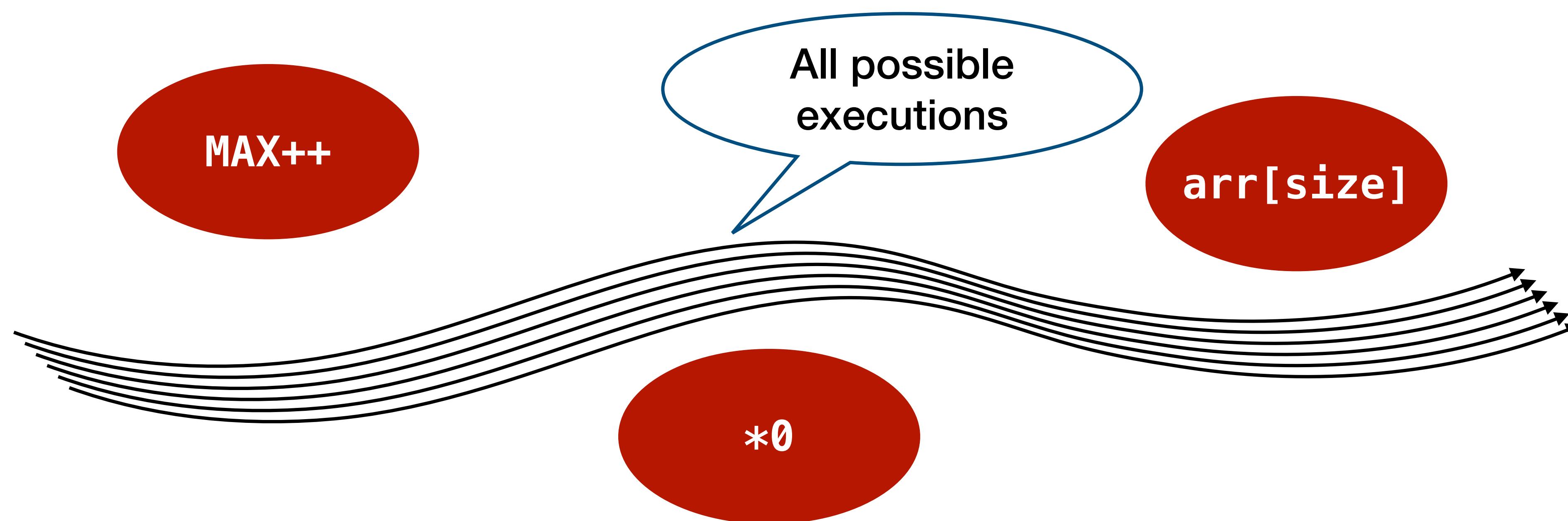
Q: What are the possible output values?

- Concrete interpretation : 2, 4, ..., uncomputable (infinitely many possibilities)
- Abstract interpretation 1 : “integers” (good)
- Abstract interpretation 2 : “positive integers” (better)
- Abstract interpretation 3 : “positive even integers” (best)

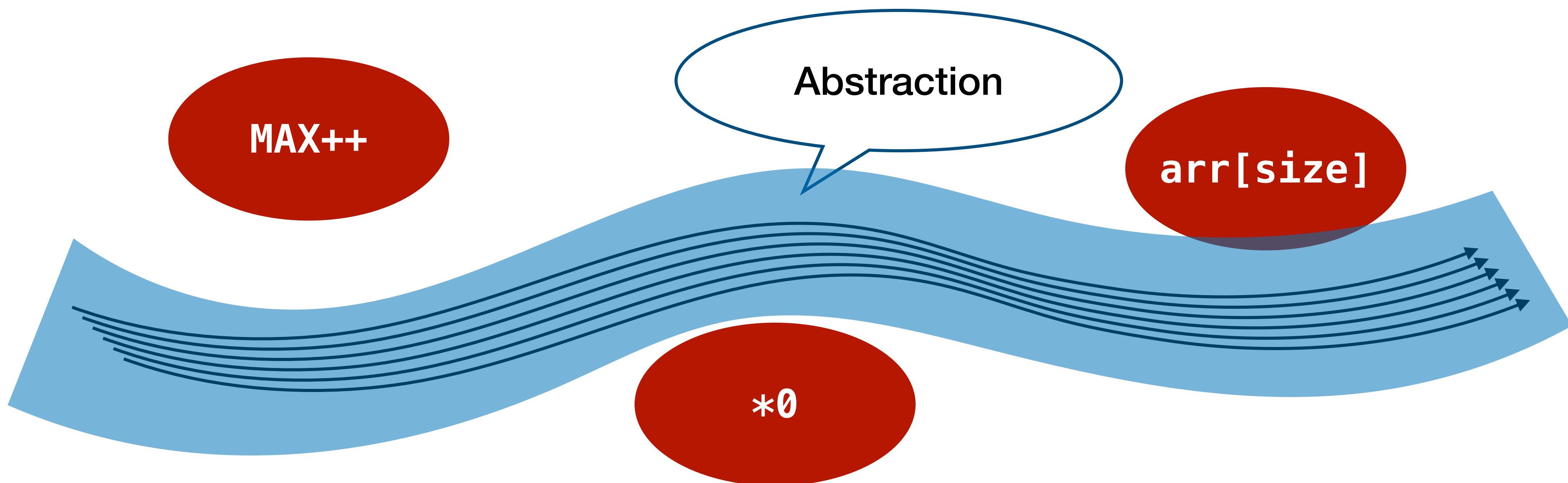
Abstraction of Executions



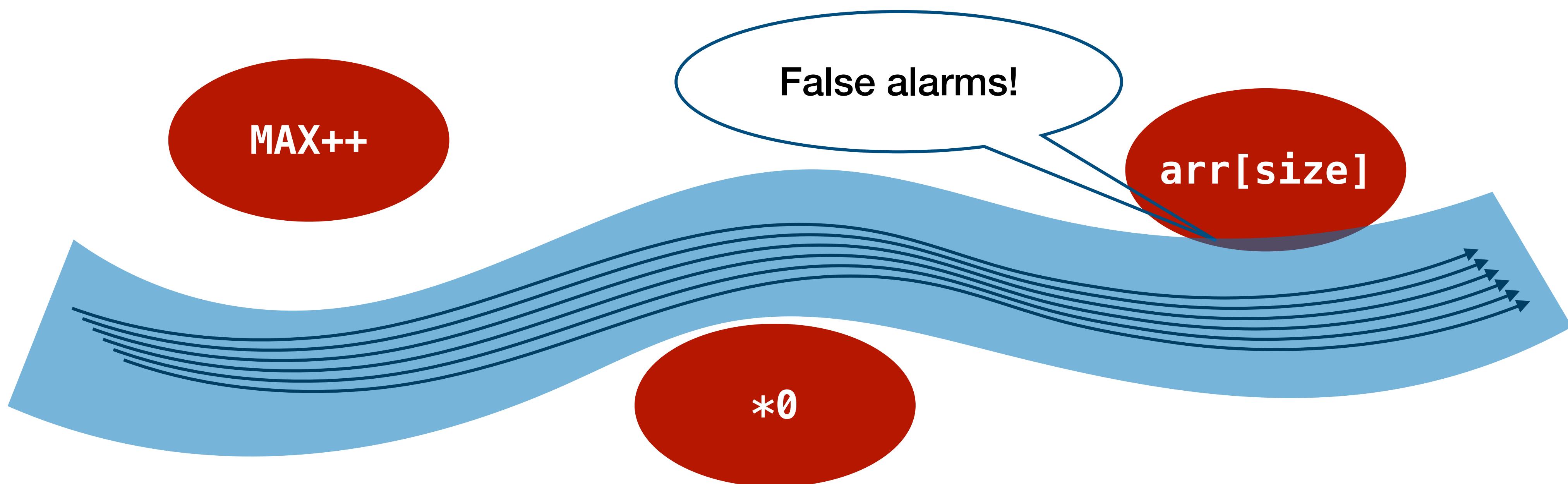
Abstraction of Executions



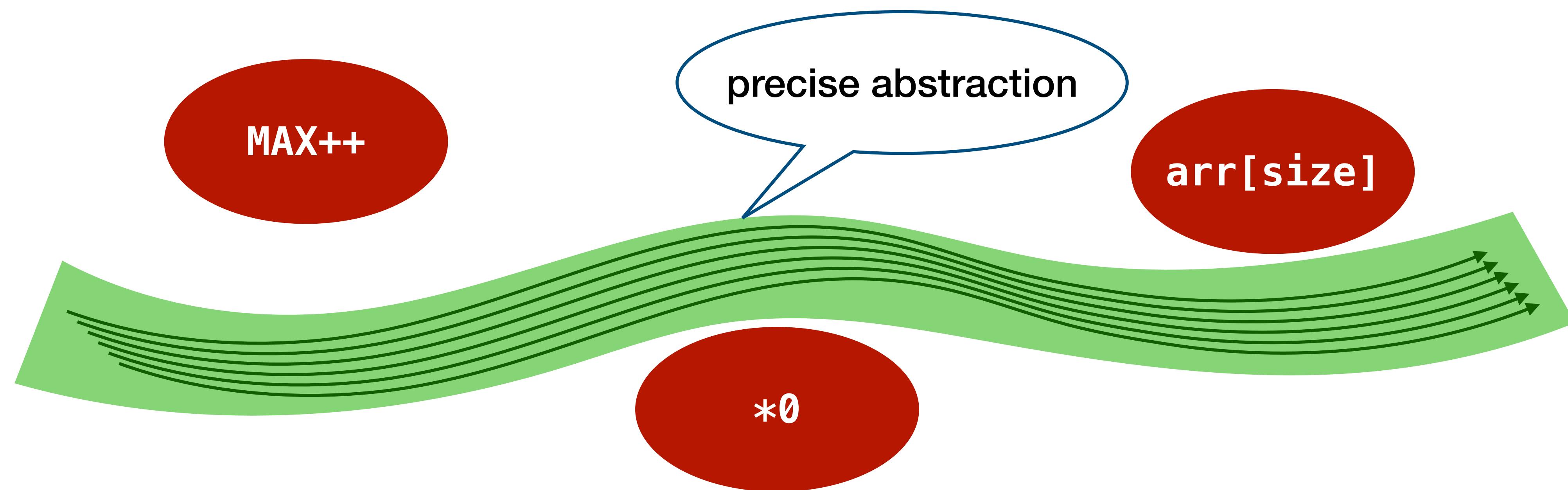
Abstraction of Executions



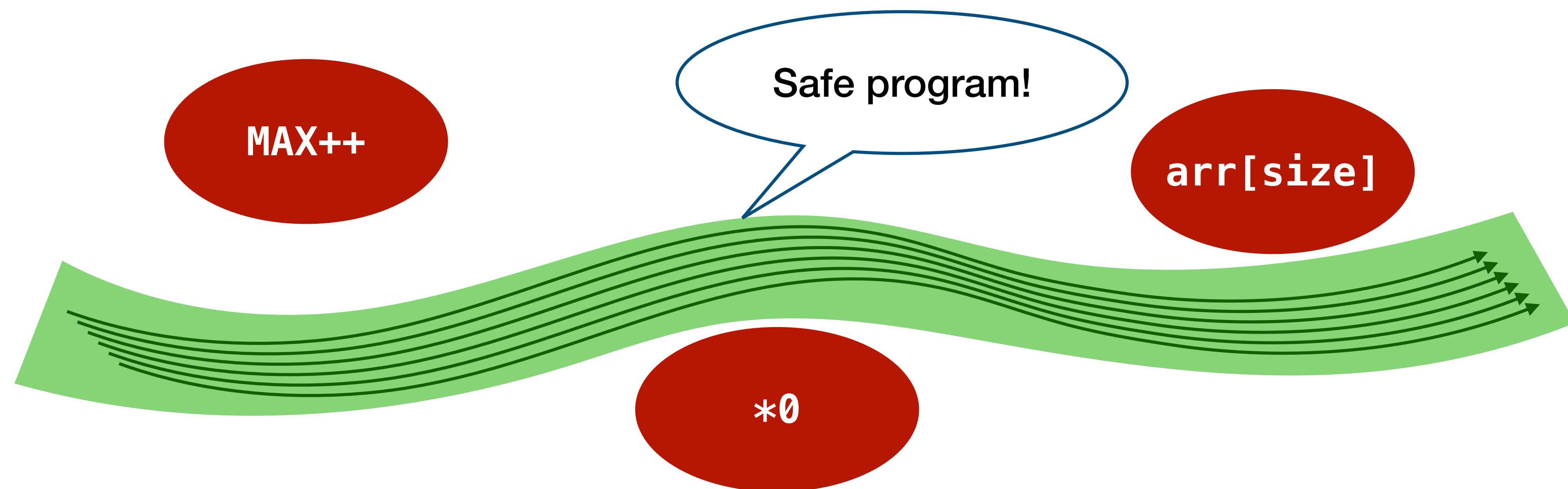
Abstraction of Executions



Abstraction of Executions



Abstraction of Executions



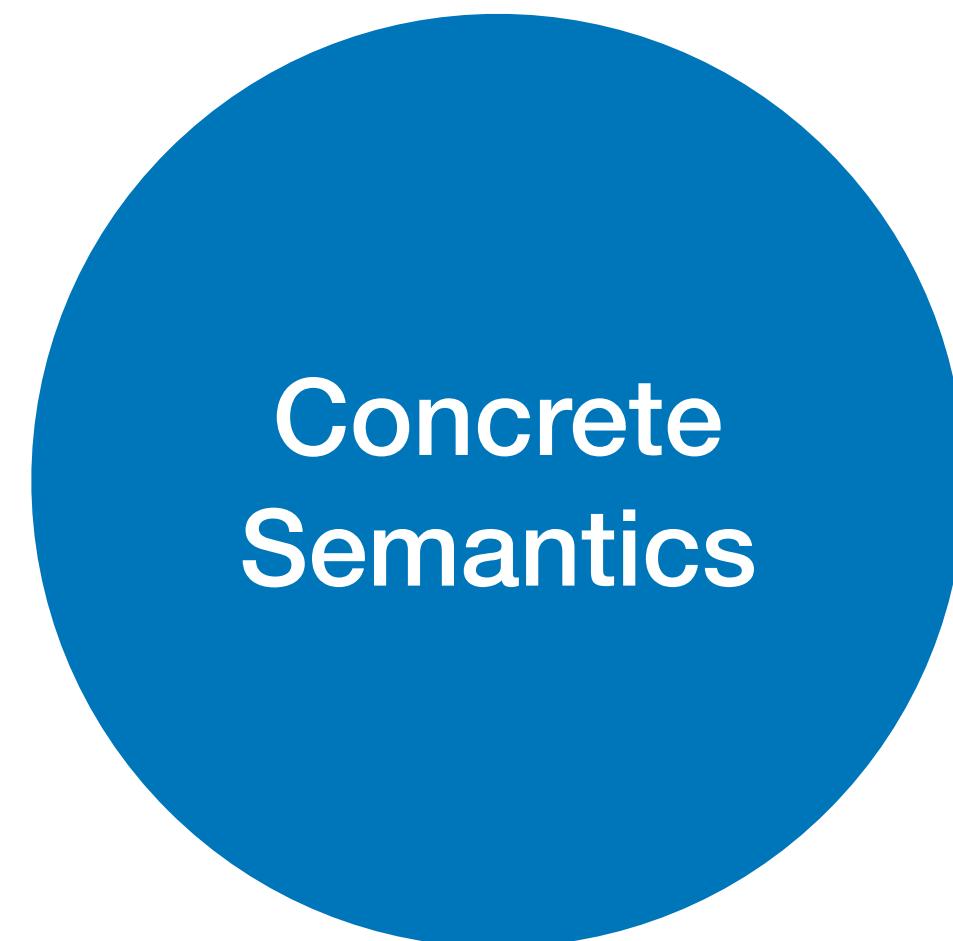
How to analyze?

- Interpret the target program
 - with abstract semantics (= analyzer's concern)
 - not concrete semantics (= interpreter's and compiler's concern)

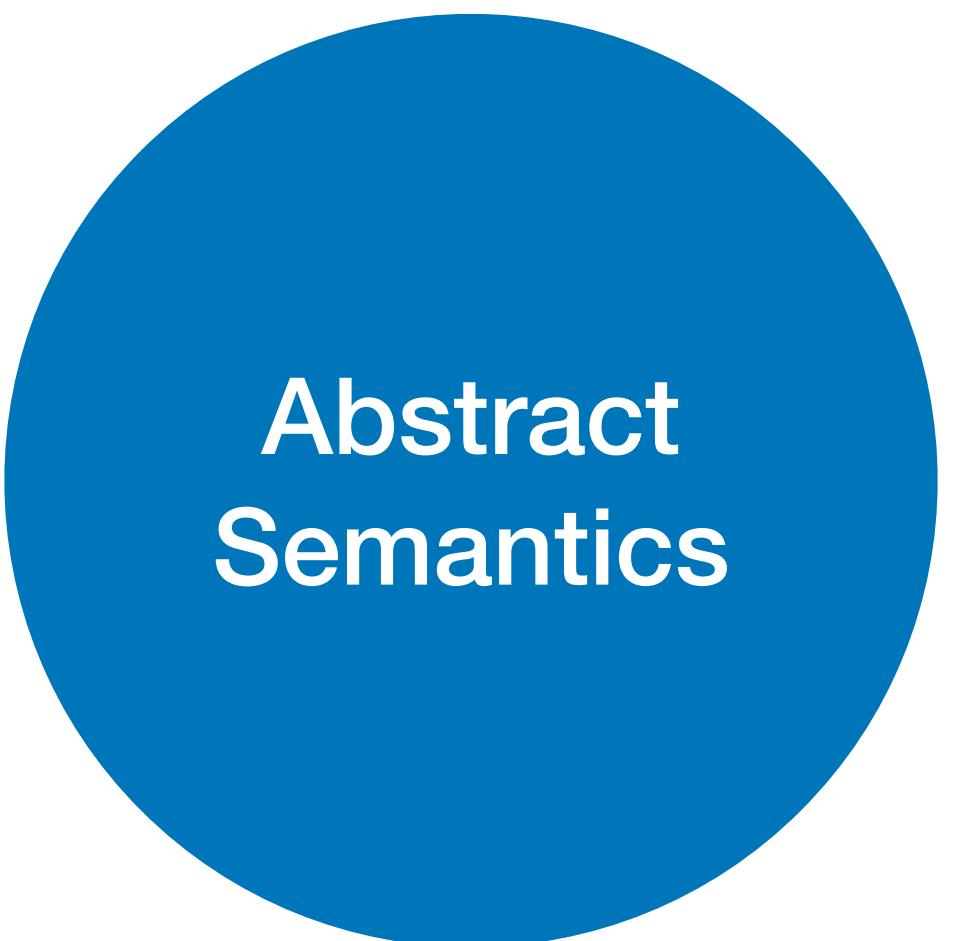
- Example

	Concrete	Abstract 1	Abstract 2	Abstract 3
x = 3;	{3}	Int	Pos	PosOdd
while (*) {				
x += 2;				
}	{3, 5, 7, ...}	Int	Pos	PosOdd
x -= 1;	{2, 4, 6, ...}	Int	Pos	PosEven
print(x);				

Principles



?
≈



- How to guarantee soundness?
- How to guarantee termination?
- How to design more precise abstraction?
- How to compute abstract semantics?

Practices



- Guidance for a lot of design choices in practice such as
 - Soundness vs Scalability vs Precision vs Usability vs ...
 - Characteristics of target programs and properties
 - Optimizations of program analyzers

Design of Static Analysis

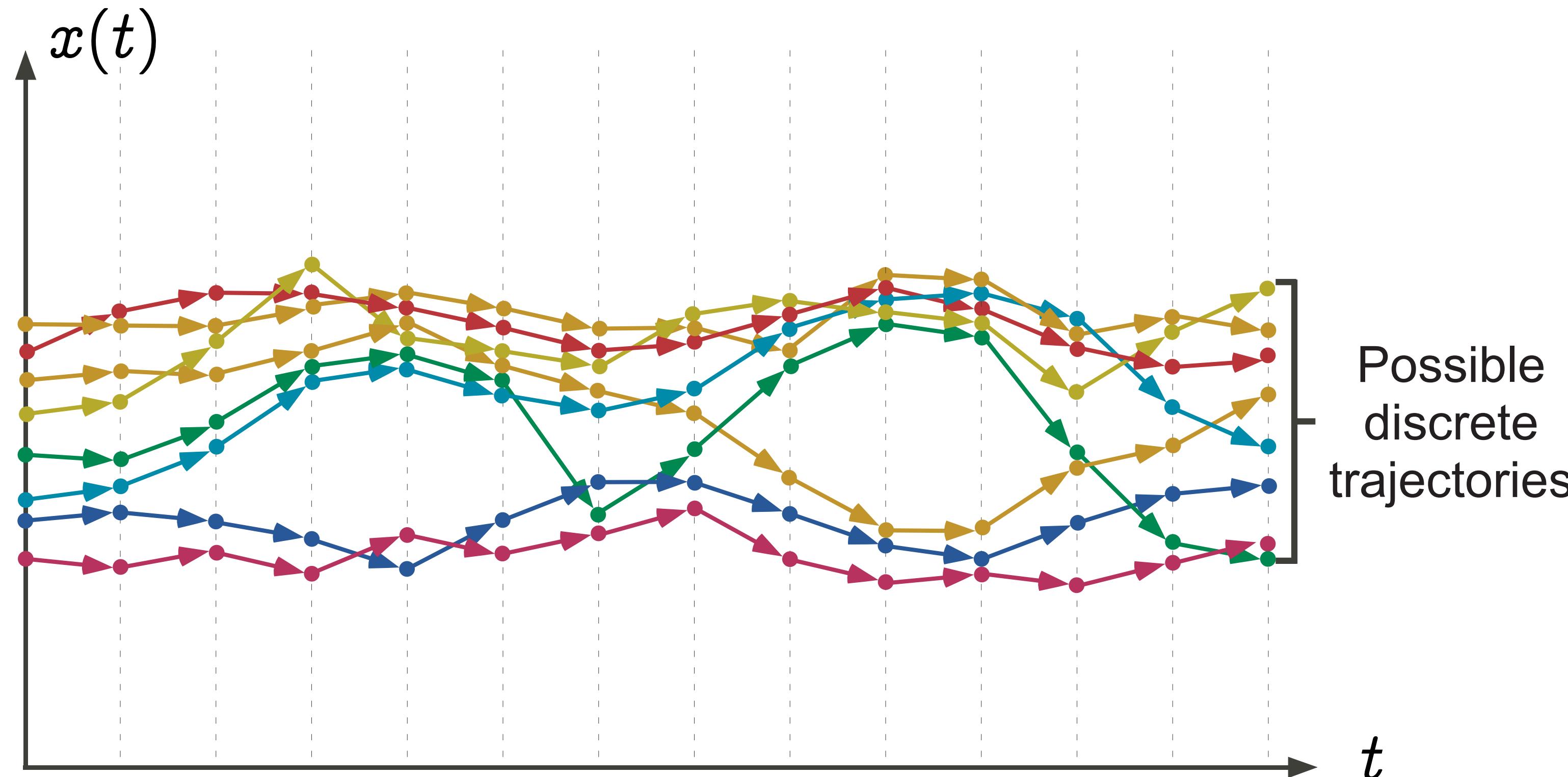
- Goal: **conservative** and **terminating** static analysis
- Design principles:
 - Define **concrete semantics**
 - Define **abstract semantics** (sound w.r.t the concrete semantics)
- Computation & implementation:
 - Abstract semantics of a program: **the least fixed point** of the semantic function
 - Static analyzer: **compute** the least fixed point within **finite time**

Define Standard Semantics

- Formalization of a **single program execution**
 - Recall Lecture 2 and 3 (operational and denotation semantics)
- What to describe: different choices depending on the purpose
 - E.g., denotational, operational, etc
- In this lecture, we will use denotational semantics
 - Recall the denotational semantics the simple imperative language

$$[C] : \mathbb{M} \rightarrow \mathbb{M}$$

Define Standard Semantics



*from Patrick Cousot's slides

Standard Semantics

- Define a semantic domain $\mathbb{D} = \mathbb{M} \rightarrow \mathbb{M}$ (CPO)
- Define a semantic function $F : \mathbb{D} \rightarrow \mathbb{D}$ (continuous)
- Semantics of a program: the least fixed point of F

$$\text{lfp } F = \bigsqcup_{i \geq 0} F^i(\perp)$$

Standard Semantics of Commands

$$\begin{aligned}\llbracket C \rrbracket &: \mathbb{M} \rightarrow \mathbb{M} \\ \llbracket \text{skip} \rrbracket &= \lambda m. m \\ \llbracket C_0 ; C_1 \rrbracket &= \lambda m. \llbracket C_1 \rrbracket(\llbracket C_0 \rrbracket(m)) \\ \llbracket x := E \rrbracket &= \lambda m. m\{x \mapsto \llbracket E \rrbracket(m)\} \\ \llbracket \text{input}(x) \rrbracket &= \lambda m. m\{x \mapsto n\} \\ \llbracket \text{if } B \text{ then } C_1 \text{ else } C_2 \rrbracket &= \lambda m. \begin{cases} \llbracket C_1 \rrbracket(m) & \text{if } \llbracket B \rrbracket(m) = \text{true} \\ \llbracket C_2 \rrbracket(m) & \text{if } \llbracket B \rrbracket(m) = \text{false} \end{cases} \\ \llbracket \text{while } B \text{ } C \rrbracket &= \text{lfp} \lambda X. \left(\lambda m. \begin{cases} X(\llbracket C \rrbracket(m)) & \text{if } \llbracket B \rrbracket(m) = \text{true} \\ m & \text{if } \llbracket B \rrbracket(m) = \text{false} \end{cases} \right)\end{aligned}$$

Standard Semantics of Programs

- What is the semantic function F ?
 - Straightforward from the definition of $\llbracket C \rrbracket$ (because of compositionality!)

$$F : (\mathbb{M} \rightarrow \mathbb{M}) \rightarrow (\mathbb{M} \rightarrow \mathbb{M})$$

$$F = \lambda X. \llbracket C \rrbracket$$

- Then, the semantics of a program C

$$\text{lfp } F : \mathbb{M} \rightarrow \mathbb{M}$$

$$\text{lfp } F = \llbracket C \rrbracket$$

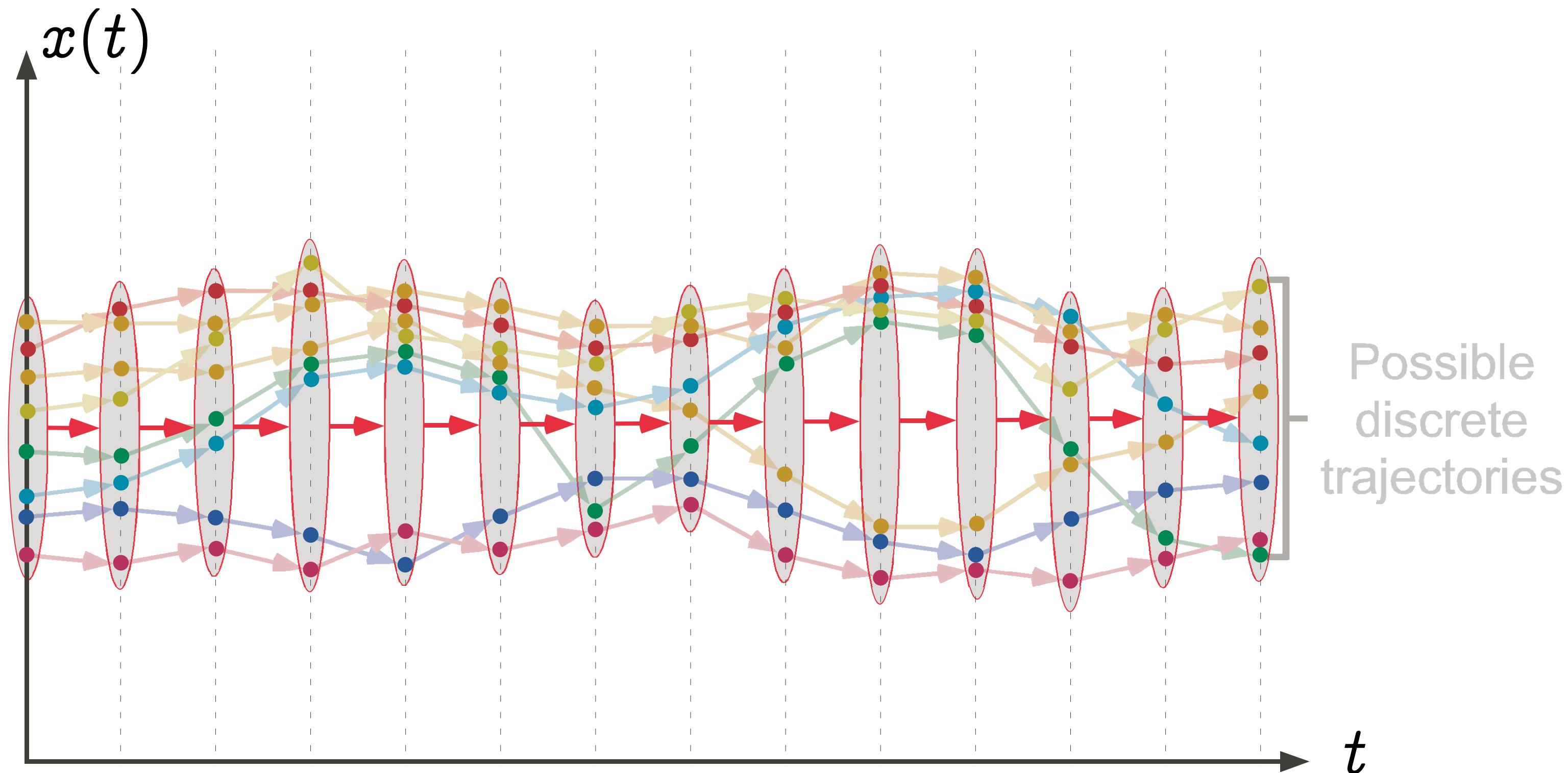
- In this lecture, we will consider only $\llbracket C \rrbracket$

Define Concrete Semantics

- Formalization of **all possible** program executions
 - So-called collecting semantics
 - Usually a simple extension of the standard semantics
- What to describe: different choices depending on the purposes (recall, property)
 - Some are more expressive than others
 - E.g., traces (sequence of states), reachable states (set of states), etc
- In this lecture, we will use reachable states for concrete semantics

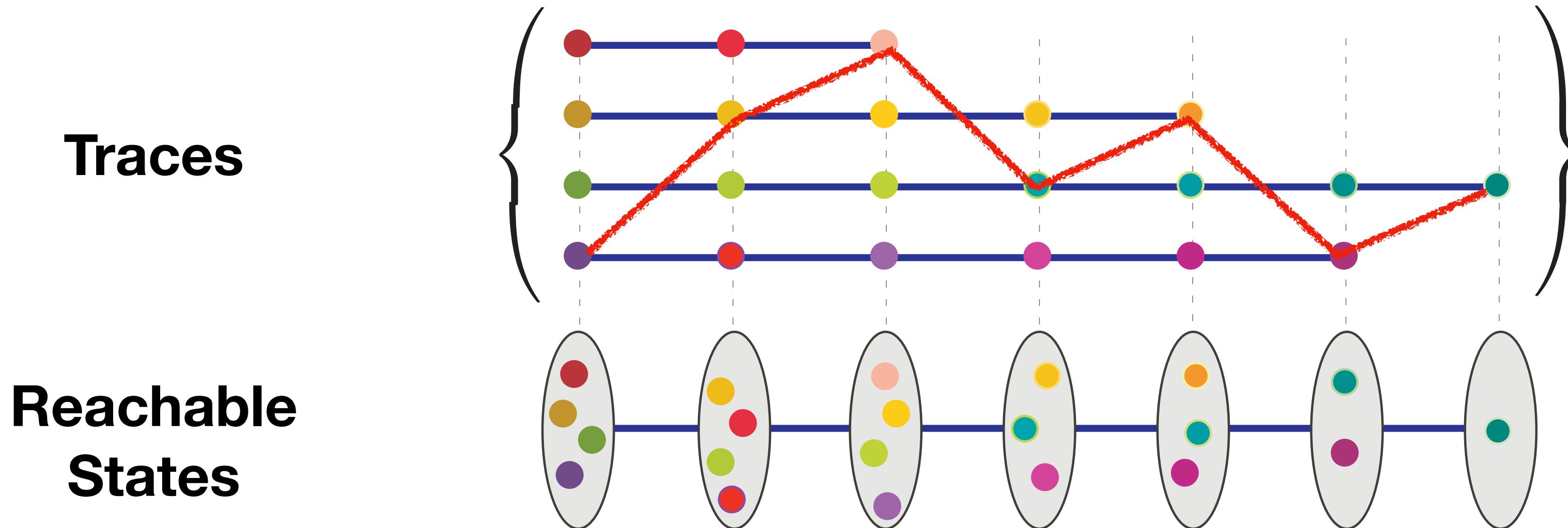
$$\llbracket C \rrbracket : \mathbb{M} \rightarrow \mathbb{M} \xrightarrow{\text{collecting}} \llbracket C \rrbracket_{\wp} : \wp(\mathbb{M}) \rightarrow \wp(\mathbb{M})$$

Transitions of Sets of States



*from Patrick Cousot's slides

Traces vs Reachable States



Can Answer:

- Can variable p be NULL at line 10?
- Can buffer index i be larger than size s?
- ...

Can't Answer:

- Does the red trace exist?
- ...

*from Patrick Cousot's slides

Concrete Semantics

- Define a concrete domain \mathbb{D} (CPO)
- Define a semantic function $F : \mathbb{D} \rightarrow \mathbb{D}$ (continuous)
- Then the concrete semantics is defined as the least fixed point of the semantic function F :

$$\text{lfp } F = \bigsqcup_{i \geq 0} F^i(\perp)$$

Example

- Define a concrete semantics of the simple language using **denotational semantics**
 - Concrete domain $\mathbb{D} = \wp(\mathbb{M}) \rightarrow \wp(\mathbb{M})$
 - Define a semantic function $F : \mathbb{D} \rightarrow \mathbb{D}$
 - Concrete semantics $\text{lfp } F \in \mathbb{D}$
- Q: How to define F ?

Concrete Semantics of Expressions

$$\llbracket E \rrbracket_{\wp} : \wp(\mathbb{M}) \rightarrow \wp(\mathbb{Z})$$

$$\llbracket n \rrbracket_{\wp} = \lambda M. \{n\}$$

$$\llbracket x \rrbracket_{\wp} = \lambda M. \{m(x) \mid m \in M\}$$

$$\llbracket E_1 \odot E_2 \rrbracket_{\wp} = \lambda M. \{v_1 \odot v_2 \mid v_1 \in \llbracket E_1 \rrbracket_{\wp}(M), v_2 \in \llbracket E_2 \rrbracket_{\wp}(M)\}$$

$$\llbracket B \rrbracket_{\wp} : \wp(\mathbb{M}) \rightarrow \wp(\mathbb{M})$$

$$\llbracket \text{true} \rrbracket_{\wp} = \lambda M. M$$

$$\llbracket \text{false} \rrbracket_{\wp} = \lambda M. \emptyset$$

$$\llbracket B_1 \oslash B_2 \rrbracket_{\wp} = \lambda M. \{m \in M \mid \llbracket B_1 \rrbracket(m) \oslash \llbracket B_2 \rrbracket(m) = \text{true}\}$$

Concrete Semantics of Commands

$$\llbracket C \rrbracket_{\wp} : \wp(\mathbb{M}) \rightarrow \wp(\mathbb{M})$$

$$\llbracket \text{skip} \rrbracket_{\wp} = \lambda M. M$$

$$\llbracket C_0 ; C_1 \rrbracket_{\wp} = \lambda M. \llbracket C_1 \rrbracket_{\wp} \circ \llbracket C_0 \rrbracket_{\wp}(M)$$

$$\llbracket x := E \rrbracket_{\wp} = \lambda M. \{m \{x \mapsto \llbracket E \rrbracket_{\wp}(m)\} \mid x \in M\}$$

$$\llbracket \text{input}(x) \rrbracket_{\wp} = \lambda M. \{m \{x \mapsto n\} \mid m \in M, n \in \mathbb{Z}\}$$

$$\llbracket \text{if } B \text{ then } C_1 \text{ else } C_2 \rrbracket_{\wp} = \lambda M. \llbracket C_1 \rrbracket_{\wp} \circ \llbracket B \rrbracket_{\wp}(M) \cup \llbracket C_2 \rrbracket_{\wp} \circ \llbracket \neg B \rrbracket_{\wp}(M)$$

$$\llbracket \text{while } B \text{ } C \rrbracket_{\wp} = \lambda M. \llbracket \neg B \rrbracket_{\wp} (\text{lfp} \lambda X. M \cup \llbracket C \rrbracket_{\wp} \circ \llbracket B \rrbracket_{\wp}(X))$$

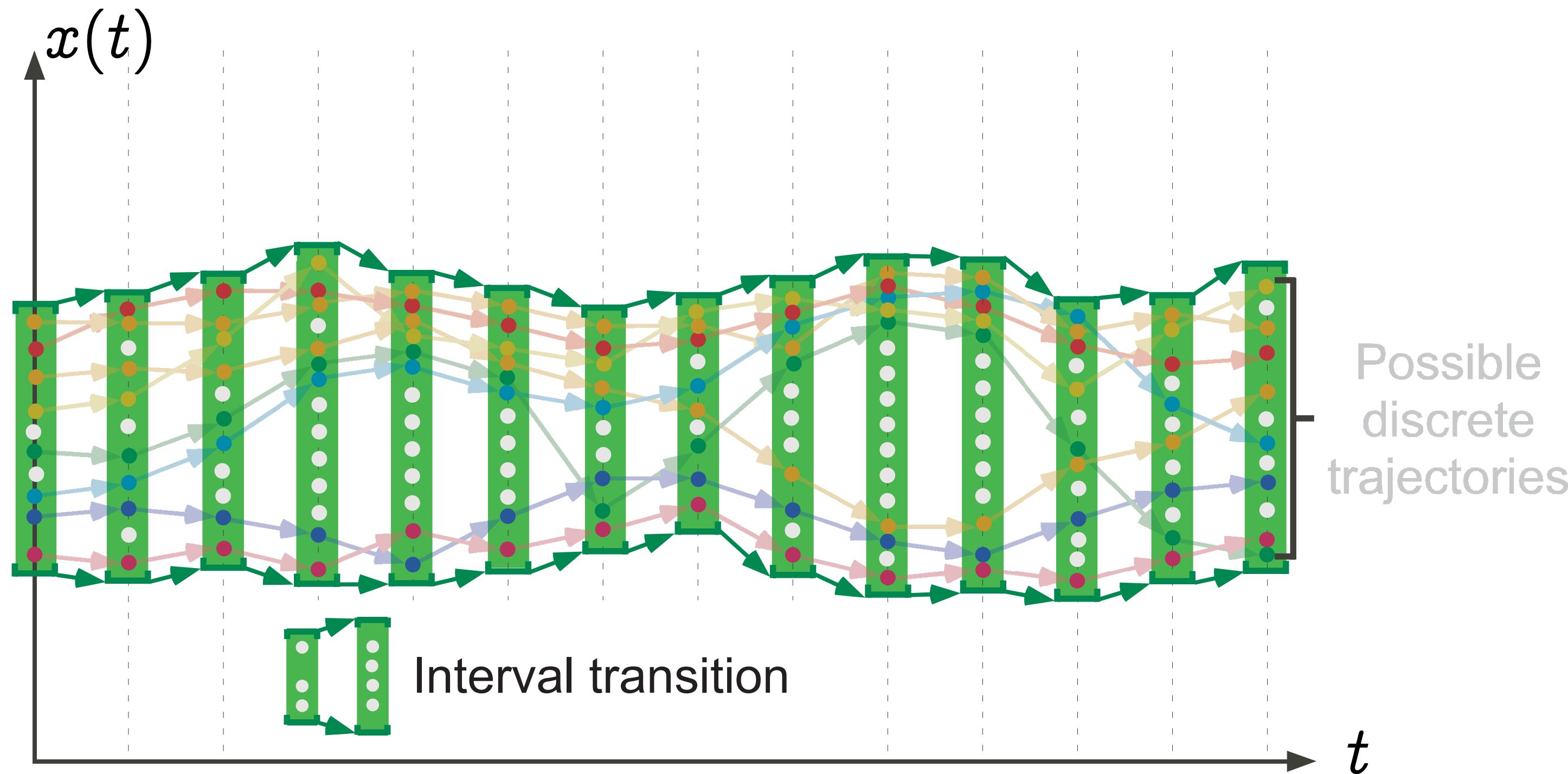
Design of Static Analysis

- Goal: **conservative** and **terminating** static analysis
- Design principles:
 - Define **concrete semantics**
 - Define **abstract semantics** (sound w.r.t the concrete semantics)
- Computation & implementation:
 - Abstract semantics of a program: **the least fixed point** of the semantic function
 - Static analyzer: **compute** the least fixed point within **finite time**

Step 2: Design Abstract Semantics

- Formalization of **abstract** program executions
 - Soundly subsume concrete executions
- How to subsume: different choices depending on the purposes
 - Some are more expressive than others
- Example: abstraction of {1, 3, 5, 7}
 - Integer, Positive, Odd, [1, 7], etc

Transitions of Abstract States



*from Patrick Cousot's slides

Abstract Semantics

- Define an abstract domain \mathbb{D}^\sharp (CPO)
- Define an abstract semantic function $F^\sharp : \mathbb{D}^\sharp \rightarrow \mathbb{D}^\sharp$ (monotone or extensive)

(Monotone) $\forall x^\sharp, y^\sharp \in \mathbb{D}^\sharp. x^\sharp \sqsubseteq y^\sharp \implies F^\sharp(x^\sharp) \sqsubseteq F^\sharp(y^\sharp)$

(Extensive) $\forall x^\sharp \in \mathbb{D}. x^\sharp \sqsubseteq F^\sharp(x^\sharp)$

- Static analysis is to compute an upper bound of the chain:

$$\bigsqcup_{i \geq 0} F^{\sharp i}(\perp^\sharp)$$

Q. How to ensure that the abstract semantics soundly subsume the concrete semantics?

Requirement 1: Galois Connection

$$\mathbb{D} \xrightleftharpoons[\alpha]{\gamma} \mathbb{D}^\sharp$$

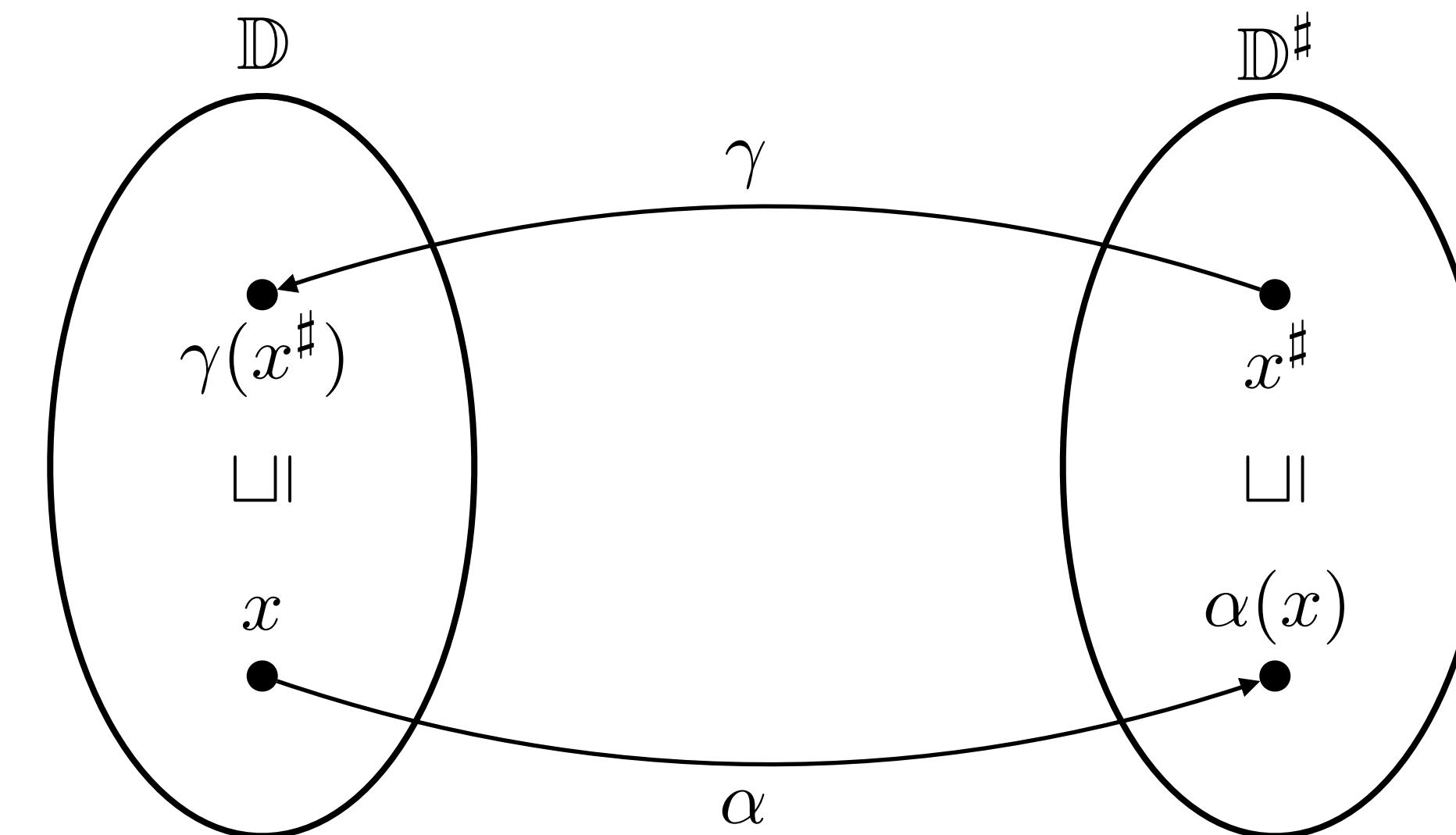
- \mathbb{D} and \mathbb{D}^\sharp must be related with a Galois connection where

- Abstraction function: $\alpha \in \mathbb{D} \rightarrow \mathbb{D}^\sharp$
- Concretization function: $\gamma \in \mathbb{D}^\sharp \rightarrow \mathbb{D}$

$$\forall x \in \mathbb{D}, x^\sharp \in \mathbb{D}^\sharp. \alpha(x) \sqsubseteq x^\sharp \iff x \sqsubseteq \gamma(x^\sharp)$$

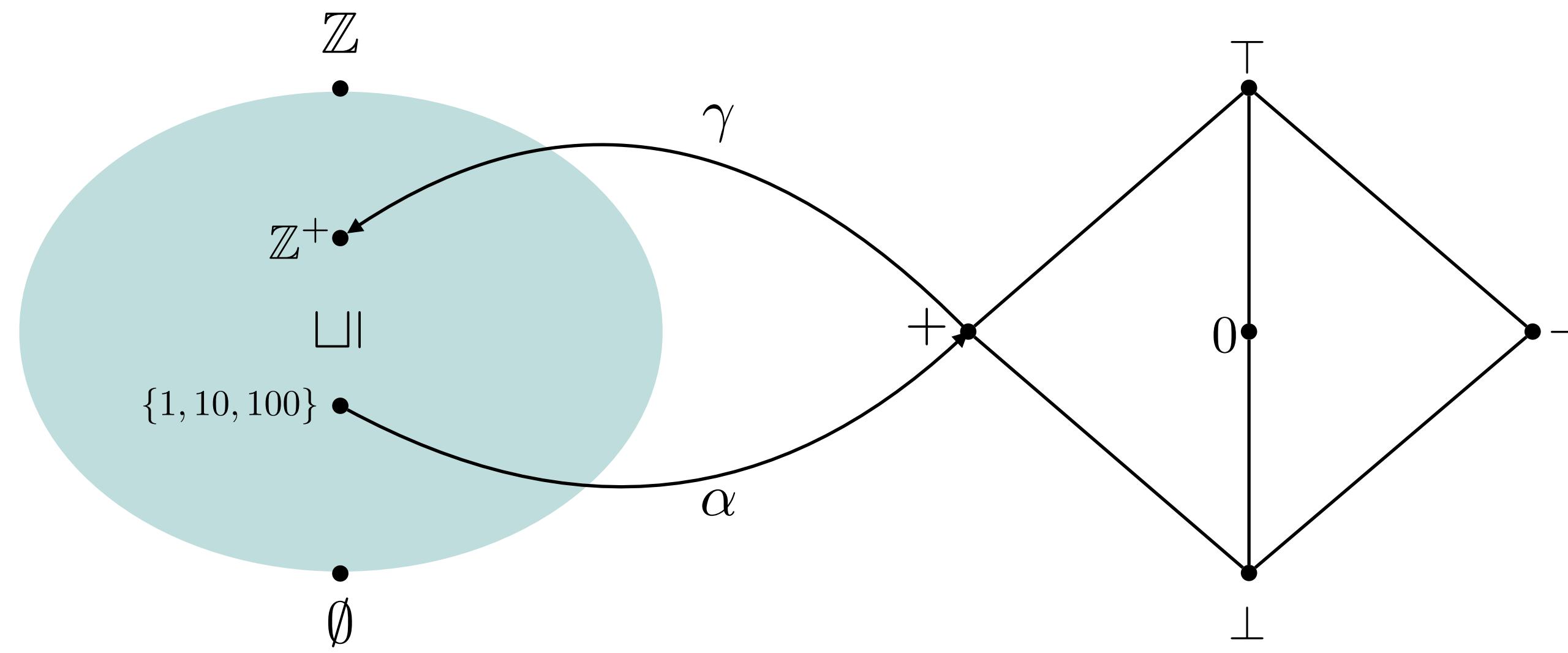
Requirement 1: Galois Connection

- Intuition: order preservation between two semantic domains



Example: Sign Abstraction

$$\wp(\mathbb{Z}) \xrightleftharpoons[\alpha]{\gamma} \{\perp, -, 0, +, \top\}$$



Example: Sign Abstraction

$$\wp(\mathbb{Z}) \xrightleftharpoons[\alpha]{\gamma} \{\perp, -, 0, +, \top\}$$

$$\alpha(Z) = \begin{cases} \perp & Z = \emptyset \\ + & \forall z \in Z. z > 0 \\ 0 & Z = \{0\} \\ - & \forall z \in Z. z < 0 \\ \top & \text{otherwise} \end{cases}$$

$$\gamma(\perp) = \emptyset$$

$$\gamma(+) = \{z \in \mathbb{Z} \mid z > 0\}$$

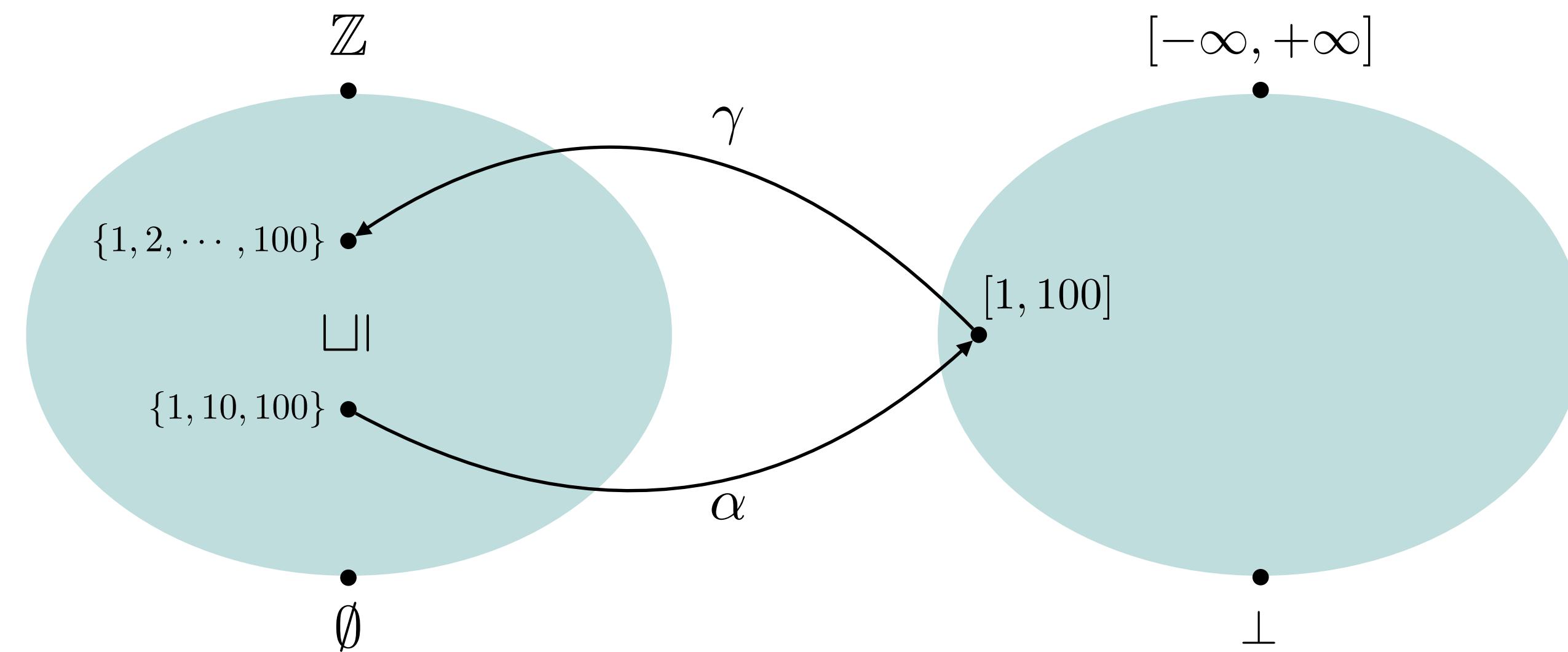
$$\gamma(0) = \{0\}$$

$$\gamma(-) = \{z \in \mathbb{Z} \mid z < 0\}$$

$$\gamma(\top) = \mathbb{Z}$$

Example: Interval Abstraction

$$\wp(\mathbb{Z}) \xrightleftharpoons[\alpha]{\gamma} \{\perp\} \cup \{[a, b] \mid a \in \mathbb{Z} \cup \{-\infty\}, b \in \mathbb{Z} \cup \{+\infty\}\}$$



Example: Interval Abstraction

$$\wp(\mathbb{Z}) \xrightleftharpoons[\alpha]{\gamma} \{\perp\} \cup \{[a, b] \mid a \in \mathbb{Z} \cup \{-\infty\}, b \in \mathbb{Z} \cup \{+\infty\}\}$$

$$\alpha(\emptyset) = \perp$$

$$\alpha(X) = [\min X, \max X]$$

$$\gamma(\perp) = \emptyset$$

$$\gamma([a, b]) = \{x \in \mathbb{Z} \mid a \leq x \leq b\}$$

Properties of Galois Connection

$$\forall x \in \mathbb{D}, x^\# \in \mathbb{D}^\#. \alpha(x) \sqsubseteq x^\# \iff x \sqsubseteq \gamma(x^\#)$$

- $id \sqsubseteq \gamma \circ \alpha$

$$\iff \alpha(x) \sqsubseteq \alpha(x) \\ \iff x \sqsubseteq \gamma(\alpha(x)) \quad (\text{by Galois connection})$$

- $\alpha \circ \gamma \sqsubseteq id$

$$\iff \gamma(x^\#) \sqsubseteq \gamma(x^\#) \\ \iff \alpha(\gamma(x^\#)) \sqsubseteq x^\# \quad (\text{by Galois connection})$$

- α is monotone

$$\implies x \sqsubseteq y \\ \implies x \sqsubseteq \gamma(\alpha(y)) \quad (id \sqsubseteq \gamma \circ \alpha) \\ \iff \alpha(x) \sqsubseteq \alpha(y) \quad (\text{by Galois connection})$$

- γ is monotone

$$\implies x^\# \sqsubseteq y^\# \\ \implies \alpha(\gamma(x^\#)) \sqsubseteq y^\# \quad (\alpha \circ \gamma \sqsubseteq id) \\ \iff \gamma(x^\#) \sqsubseteq \gamma(x^\#) \quad (\text{by Galois connection})$$

Deriving Galois Connections (1)

- Pointwise lifting:

Given a Galois connection $\mathbb{D} \xrightleftharpoons[\alpha]{\gamma} \mathbb{D}^\sharp$ and a set \mathbb{S}

$$\mathbb{S} \rightarrow \mathbb{D} \xrightleftharpoons[\alpha']{\gamma'} \mathbb{S} \rightarrow \mathbb{D}^\sharp$$

where $\alpha'(f) = \lambda x \in \mathbb{S}. \alpha(f(x))$ and $\gamma'(f^\sharp) = \lambda x \in \mathbb{S}. \gamma(f^\sharp(x))$

- Composition:

Given two galois connections $\mathbb{D}_1 \xrightleftharpoons[\alpha_1]{\gamma_1} \mathbb{D}_2 \xrightleftharpoons[\alpha_2]{\gamma_2} \mathbb{D}_3$

$$\mathbb{D}_1 \xrightleftharpoons[\alpha_2 \circ \alpha_1]{\gamma_1 \circ \gamma_2} \mathbb{D}_3$$

Deriving Galois Connections (2)

Given two Galois connections $\mathbb{D}_1 \xrightleftharpoons[\alpha_1]{\gamma_1} \mathbb{D}_1^\sharp$ and $\mathbb{D}_2 \xrightleftharpoons[\alpha_2]{\gamma_2} \mathbb{D}_2^\sharp$

- $\mathbb{D}_1 \times \mathbb{D}_2 \xrightleftharpoons[\alpha]{\gamma} \mathbb{D}_1^\sharp \times \mathbb{D}_2^\sharp$ where $\alpha = \lambda \langle x, y \rangle. \langle \alpha_1(x), \alpha_2(y) \rangle$
- $\mathbb{D}_1 + \mathbb{D}_2 \xrightleftharpoons[\alpha]{\gamma} \mathbb{D}_1^\sharp + \mathbb{D}_2^\sharp$ where $\alpha = \lambda x. \begin{cases} \alpha_1(x) & \text{if } x \in \mathbb{D}_1 \\ \alpha_2(x) & \text{if } x \in \mathbb{D}_2 \end{cases}$
- $\mathbb{D}_1 \rightarrow \mathbb{D}_2 \xrightleftharpoons[\alpha]{\gamma} \mathbb{D}_1^\sharp \rightarrow \mathbb{D}_2^\sharp$ where $\alpha = \lambda f. \alpha_2 \circ f \circ \gamma_1$

Deriving Galois Connections (3)

Given two Galois connections $\wp(\mathbb{A}) \xrightleftharpoons[\alpha_1]{\gamma_1} \mathbb{D}_1^\sharp$ and $\wp(\mathbb{B}) \xrightleftharpoons[\alpha_2]{\gamma_2} \mathbb{D}_2^\sharp$

- $\wp(\mathbb{A} \times \mathbb{B}) \xrightleftharpoons[\alpha]{\gamma} \mathbb{D}_1^\sharp \times \mathbb{D}_2^\sharp$ where $\alpha = \lambda X. \langle \alpha_1(\{a \mid \langle a, b \rangle \in X\}), \alpha_2(\{b \mid \langle a, b \rangle \in X\}) \rangle$
- $\wp(\mathbb{A} \times \mathbb{B}) \xrightleftharpoons[\alpha]{\gamma} \mathbb{A}' \rightarrow \mathbb{D}_2^\sharp$ where $\alpha = \lambda X. \{a \mapsto \alpha_2(S) \mid \langle a, b \rangle \in X, S = \{b \mid \langle a, b \rangle \in X\}\}$ and $\mathbb{A}' \subseteq \mathbb{A}$
- $\wp(\mathbb{A} + \mathbb{B}) \xrightleftharpoons[\alpha]{\gamma} \mathbb{D}_1^\sharp \times \mathbb{D}_2^\sharp$ where $\alpha = \lambda X. \langle \alpha_1(X \cap \mathbb{A}), \alpha_2(X \cap \mathbb{B}) \rangle$
- $\wp(\mathbb{A}) \rightarrow \wp(\mathbb{B}) \xrightleftharpoons[\alpha]{\gamma} \mathbb{D}_1^\sharp \rightarrow \mathbb{D}_2^\sharp$ where $\alpha = \lambda f. \alpha_2 \circ f \circ \gamma_1$

Example

- Concrete domain: $\mathbb{D} = \wp(\mathbb{M}) \rightarrow \wp(\mathbb{M})$ where $\mathbb{M} = \mathbb{X} \rightarrow \mathbb{Z}$
- Abstract domain: $\mathbb{D}^\sharp = \mathbb{M}^\sharp \rightarrow \mathbb{M}^\sharp$ where $\mathbb{M}^\sharp = \mathbb{X} \rightarrow \mathbb{Z}^\sharp$
- Galois connection: $\wp(\mathbb{D}) \xrightleftharpoons[\alpha]{\gamma} \mathbb{D}^\sharp$
 - Memory abstraction: $\wp(\mathbb{M}) \xrightleftharpoons[\alpha_{\mathbb{M}}]{\gamma_{\mathbb{M}}} \mathbb{M}^\sharp$ via $\wp(\mathbb{X} \rightarrow \mathbb{Z}) \xrightleftharpoons[\alpha_{\mathbb{M}}]{\gamma_{\mathbb{M}}} \mathbb{X} \rightarrow \mathbb{Z}^\sharp$
$$\gamma_{\mathbb{M}} : \mathbb{M}^\sharp \rightarrow \wp(\mathbb{M})$$
$$\gamma_{\mathbb{M}} = \lambda m^\sharp. \{m \mid \forall x. m(x) \in \gamma_{\mathbb{Z}}(m^\sharp(x))\}$$
 - Value abstraction: $\wp(\mathbb{Z}) \xrightleftharpoons[\alpha_{\mathbb{Z}}]{\gamma_{\mathbb{Z}}} \mathbb{Z}^\sharp$

Requirement 2: F and $F^\#$

- $F^\#$ is a sound abstraction of F (option 1)

$$F \circ \gamma \sqsubseteq \gamma \circ F^\#$$

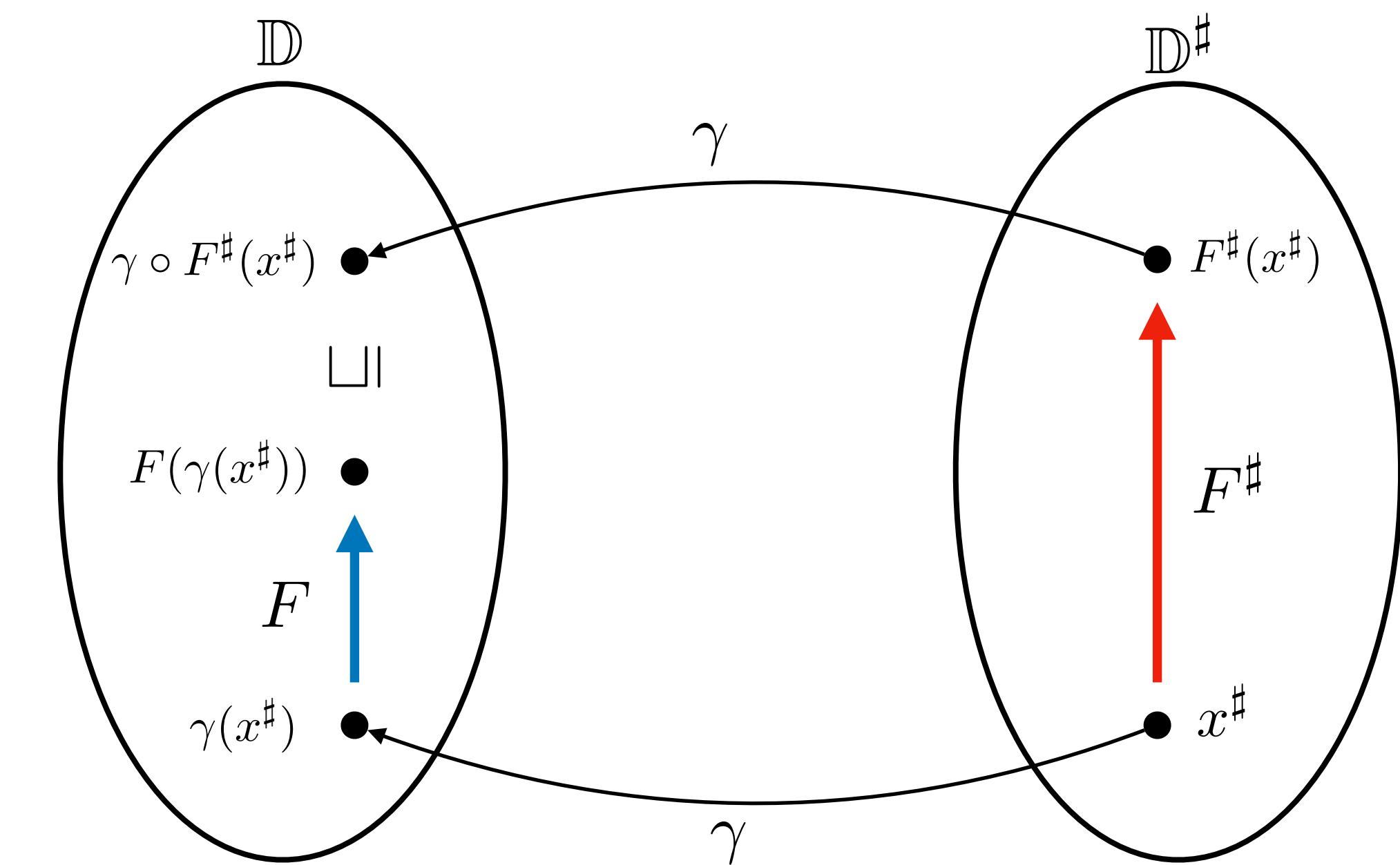
- $F^\#$ is a sound abstraction of F (option 2)

$$x \sqsubseteq \gamma(x^\#) \implies F(x) \sqsubseteq \gamma(F^\#(x^\#))$$

Intuition: the result of one-step abstract execution ($F^\#$)
subsumes that of one-step concrete execution (F)

Sound Abstract Semantics (1)

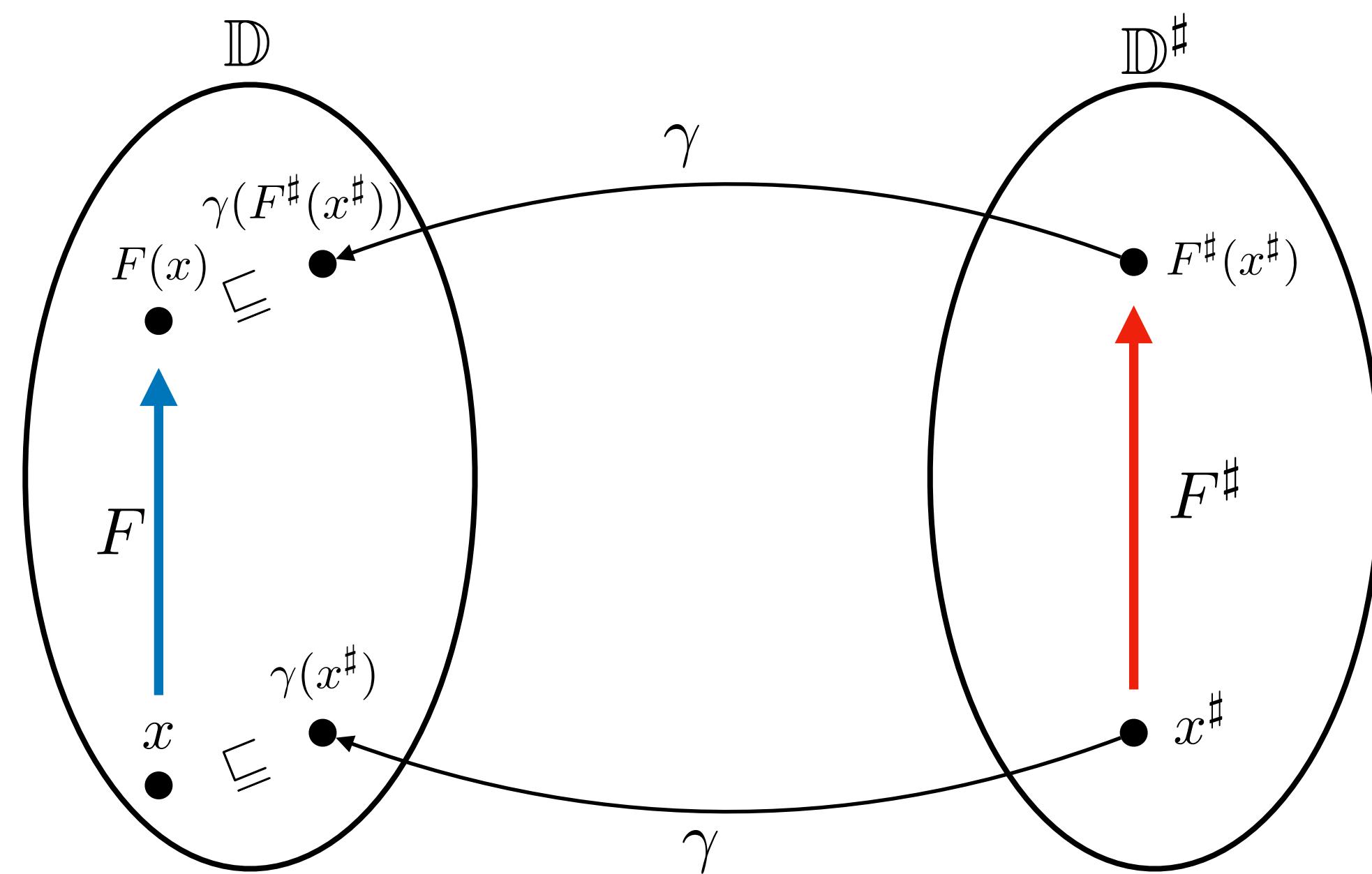
$$F \circ \gamma \sqsubseteq \gamma \circ F^\sharp$$



Intuition: the result of one-step abstract execution (F^\sharp)
subsumes that of one-step concrete execution (F)

Sound Abstract Semantics (2)

$$x \sqsubseteq \gamma(x^\sharp) \implies F(x) \sqsubseteq \gamma(F^\sharp(x^\sharp))$$



Intuition: the result of one-step abstract execution (F^\sharp)
subsumes that of one-step concrete execution (F)

Soundness

- Static analysis result $\bigsqcup_{i \geq 0} F^{\sharp i}(\perp)$ soundly subsumes all possible executions

$$\text{lfp } F \sqsubseteq \gamma\left(\bigsqcup_{i \geq 0} F^{\sharp i}(\perp)\right)$$

- How to guarantee the soundness?
- How to compute the sound result within finite time?

Fixpoint Transfer Theorems

- With option 1

Theorem (Fixpoint Transfer 1). *Let \mathbb{D} and \mathbb{D}^\sharp be related by Galois connection $\mathbb{D} \xrightleftharpoons[\alpha]{\gamma} \mathbb{D}^\sharp$. Let $F : \mathbb{D} \rightarrow \mathbb{D}$ be a continuous function and $F^\sharp : \mathbb{D}^\sharp \rightarrow \mathbb{D}^\sharp$ be a monotone or extensive function such that $F \circ \gamma \sqsubseteq \gamma \circ F^\sharp$. Then,*

$$\text{lfp } F \sqsubseteq \gamma \left(\bigsqcup_{i \geq 0} F^{\sharp i}(\perp^\sharp) \right).$$

- With option 2

Theorem (Fixpoint Transfer 2). *Let \mathbb{D} and \mathbb{D}^\sharp be related by Galois connection $\mathbb{D} \xrightleftharpoons[\alpha]{\gamma} \mathbb{D}^\sharp$. Let $F : \mathbb{D} \rightarrow \mathbb{D}$ be a continuous function and $F^\sharp : \mathbb{D}^\sharp \rightarrow \mathbb{D}^\sharp$ be a monotone or extensive function such that $x \sqsubseteq \gamma(x^\sharp) \implies F(x) \sqsubseteq \gamma(F^\sharp(x^\sharp))$. Then,*

$$\text{lfp } F \sqsubseteq \gamma \left(\bigsqcup_{i \geq 0} F^{\sharp i}(\perp^\sharp) \right).$$

Proof

Theorem (Fixpoint Transfer 1). Let \mathbb{D} and \mathbb{D}^\sharp be related by Galois connection $\mathbb{D} \xrightleftharpoons[\alpha]{\gamma} \mathbb{D}^\sharp$. Let $F : \mathbb{D} \rightarrow \mathbb{D}$ be a continuous function and $F^\sharp : \mathbb{D}^\sharp \rightarrow \mathbb{D}^\sharp$ be a monotone or extensive function such that $F \circ \gamma \sqsubseteq \gamma \circ F^\sharp$. Then,

$$\text{lfp } F \sqsubseteq \gamma\left(\bigcup_{i \geq 0} F^{\sharp i}(\perp^\sharp)\right).$$

Proof. First we prove $\forall n \in \mathbb{N}. F^n(\perp) \sqsubseteq \gamma(F^{\sharp n}(\perp^\sharp))$ by induction. The base case is trivial. The inductive case is as follows:

$$\begin{aligned} F^{n+1}(\perp) &= F \circ F^n(\perp) \\ &\sqsubseteq F \circ \gamma(F^{\sharp n}(\perp^\sharp)) \quad (\text{by induction hypothesis and monotonicity of } F) \\ &\sqsubseteq \gamma \circ F^\sharp \circ F^{\sharp n}(\perp^\sharp) \quad (\text{by assumption } F \circ \gamma \sqsubseteq \gamma \circ F^\sharp) \\ &= \gamma(F^{\sharp n+1}(\perp^\sharp)) \end{aligned}$$

$\{F^i(\perp)\}_i$ is a chain because F is continuous (so monotone). Then, the least upper bound of the chain $\bigcup_{i \geq 0} F^i(\perp)$ exists because \mathbb{D} is a CPO. $\{F^{\sharp i}(\perp^\sharp)\}_i$ is a chain because F^\sharp is monotone or extensive. Then, $\{\gamma(F^{\sharp i}(\perp^\sharp))\}_i$ is also a chain because γ is monotone. Therefore, the least upper bound of the chain $\bigcup_{i \geq 0} \{\gamma(F^{\sharp i}(\perp^\sharp))\}_i$ exists.

$$\begin{aligned} \text{lfp } F &= \bigcup_{i \geq 0} F^i(\perp) \sqsubseteq \bigcup_{i \geq 0} \gamma(F^{\sharp i}(\perp^\sharp)) \\ &\sqsubseteq \gamma\left(\bigcup_{i \geq 0} (F^{\sharp i}(\perp^\sharp))\right) \quad (\text{by monotonicity of } \gamma) \end{aligned}$$

Abstraction of Compositions

- Composition of sound semantic functions is also sound

$$\begin{array}{ccc} \mathbb{D}_1 & \xrightleftharpoons[\alpha_1]{\gamma_1} & \mathbb{D}_1^\# \\ f \downarrow & & \downarrow f^\# \\ \mathbb{D}_2 & \xrightleftharpoons[\alpha_2]{\gamma_2} & \mathbb{D}_2^\# \\ g \downarrow & & \downarrow g^\# \\ \mathbb{D}_3 & \xrightleftharpoons[\alpha_3]{\gamma_3} & \mathbb{D}_3^\# \end{array} \longrightarrow g \circ f \circ \gamma_1 \sqsubseteq \gamma_3 \circ g^\# \circ f^\#$$

- Implication: soundness proof can be compositional

Abstract Semantics of Expressions

$$\llbracket E \rrbracket^\sharp : \mathbb{M}^\sharp \rightarrow \mathbb{Z}^\sharp$$

$$\llbracket n \rrbracket^\sharp = \lambda m^\sharp. \alpha(\{n\})$$

$$\llbracket x \rrbracket^\sharp = \lambda m^\sharp. m^\sharp(x)$$

$$\llbracket E_1 \odot E_2 \rrbracket^\sharp = \lambda m^\sharp. \llbracket E_1 \rrbracket^\sharp(m^\sharp) \odot^\sharp \llbracket E_2 \rrbracket^\sharp(m^\sharp)$$

- Example
 - Sign domain $\mathbb{Z}^\sharp = \{\perp, -, 0, +, \top\}$
 - Interval domain $\mathbb{Z}^\sharp = \{\perp\} \cup \{[a, b] \mid a \in \mathbb{Z} \cup \{-\infty\}, b \in \mathbb{Z} \cup \{+\infty\}\}$
 - Soundness: $\llbracket E \rrbracket_\wp \circ \gamma_{\mathbb{M}} \subseteq \gamma_{\mathbb{Z}} \circ \llbracket E \rrbracket^\sharp$

Abstract Semantics of Conditions

$$\llbracket B \rrbracket^\# : \mathbb{M}^\# \rightarrow \mathbb{M}^\#$$

$$\llbracket \text{true} \rrbracket^\# = \lambda m^\#. m^\#$$

$$\llbracket \text{false} \rrbracket^\# = \lambda m^\#. \perp$$

- Example

- Sign domain

$$\llbracket x < 0 \rrbracket^\# = \lambda m^\#. \begin{cases} \perp & \text{if } m^\#(x) \in \{+, 0, \perp\} \\ m^\# \{x \mapsto -\} & \text{o.w.} \end{cases}$$

- Interval domain

$$\llbracket x < n \rrbracket^\# = \lambda m^\#. \begin{cases} \perp & \text{if } m^\#(x) = [a, b] \wedge a > n \\ m^\# \{x \mapsto [a, n-1]\} & \text{if } a \leq n \leq b \\ m^\# & \text{if } b < n \end{cases}$$

- Soundness: $\llbracket B \rrbracket_\wp \circ \gamma_{\mathbb{M}} \subseteq \gamma_{\mathbb{M}} \circ \llbracket B \rrbracket^\#$

Abstract Semantics of Commands

$$\llbracket C \rrbracket^\sharp : \mathbb{M}^\sharp \rightarrow \mathbb{M}^\sharp$$

$$\llbracket \text{skip} \rrbracket^\sharp = \lambda m^\sharp. m^\sharp$$

$$\llbracket C_0 ; C_1 \rrbracket^\sharp = \lambda m^\sharp. \llbracket C_1 \rrbracket^\sharp \circ \llbracket C_0 \rrbracket^\sharp(m^\sharp)$$

$$\llbracket x := E \rrbracket^\sharp = \lambda m^\sharp. m^\sharp \{x \mapsto \llbracket E \rrbracket^\sharp(m^\sharp)\}$$

$$\llbracket \text{input}(x) \rrbracket^\sharp = \lambda m^\sharp. m^\sharp \{x \mapsto \alpha(\mathbb{Z})\}$$

$$\llbracket \text{if } B \text{ then } C_1 \text{ else } C_2 \rrbracket^\sharp = \lambda m^\sharp. \llbracket C_1 \rrbracket^\sharp \circ \llbracket B \rrbracket^\sharp(m^\sharp) \sqcup \llbracket C_2 \rrbracket^\sharp \circ \llbracket \neg B \rrbracket^\sharp(m^\sharp)$$

$$\llbracket \text{while } B \text{ } C \rrbracket^\sharp = \lambda m^\sharp. \llbracket \neg B \rrbracket^\sharp (\text{lfp} \lambda X. M^\sharp \sqcup \llbracket C \rrbracket^\sharp \circ \llbracket B \rrbracket^\sharp(M^\sharp))$$

- Soundness: $\llbracket C \rrbracket_\wp \circ \gamma_{\mathbb{M}} \subseteq \gamma_{\mathbb{M}} \circ \llbracket C \rrbracket^\sharp$

Design of Static Analysis

- Goal: **conservative** and **terminating** static analysis
- Design principles:
 - Define **concrete semantics**
 - Define **abstract semantics** (sound w.r.t the concrete semantics)
- Computation & implementation:
 - Abstract semantics of a program: **the least fixed point** of the semantic function
 - Static analyzer: **compute** the least fixed point within **finite time**

Computing Abstract Semantics

- If the abstract domain \mathbb{D}^\sharp has **finite** height (i.e., all chains are finite)

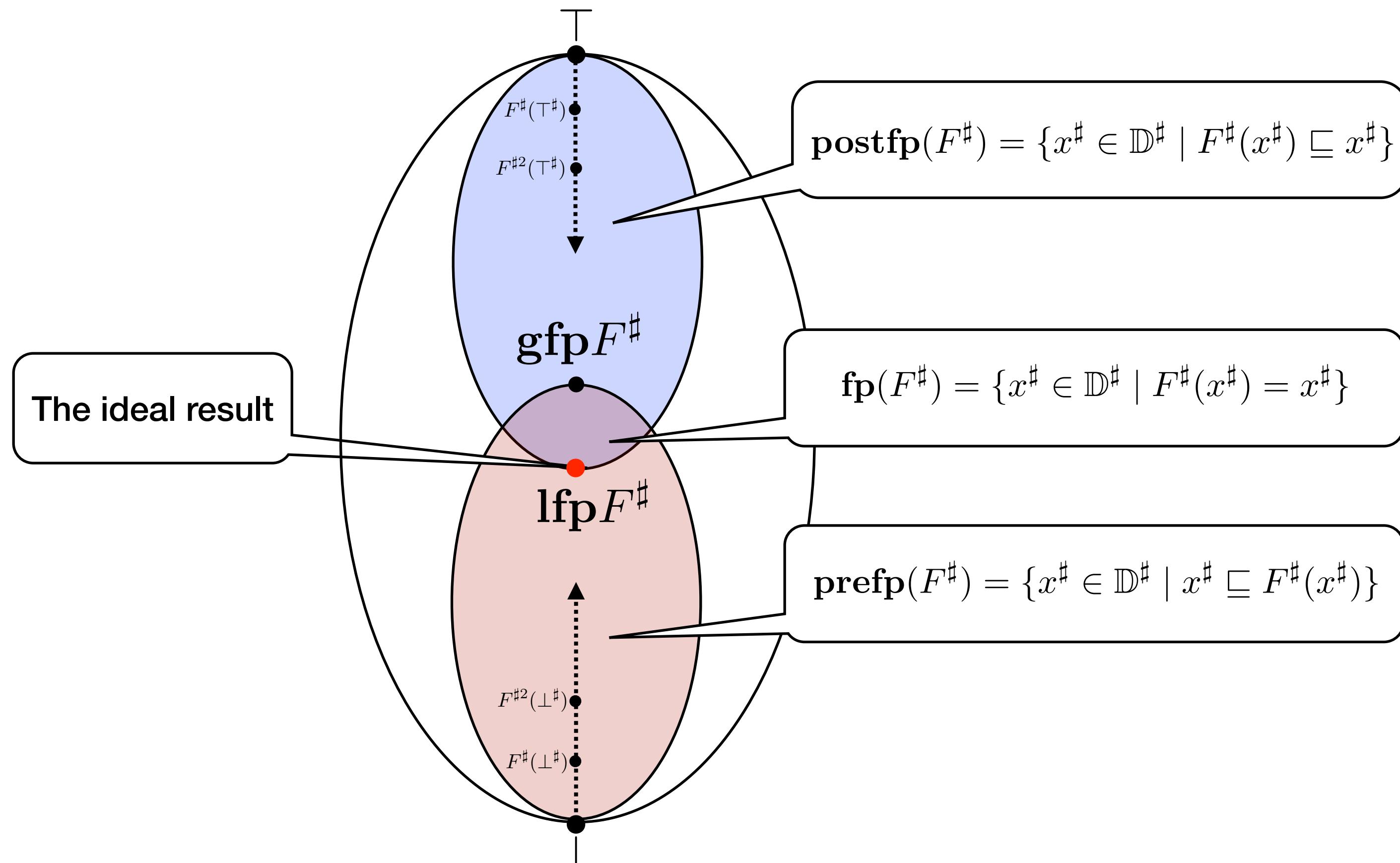
$$\bigsqcup_{i \geq 0} F^{\sharp i}(\perp^\sharp)$$

- If the abstract domain \mathbb{D}^\sharp has **infinite** height, we compute a finite chain

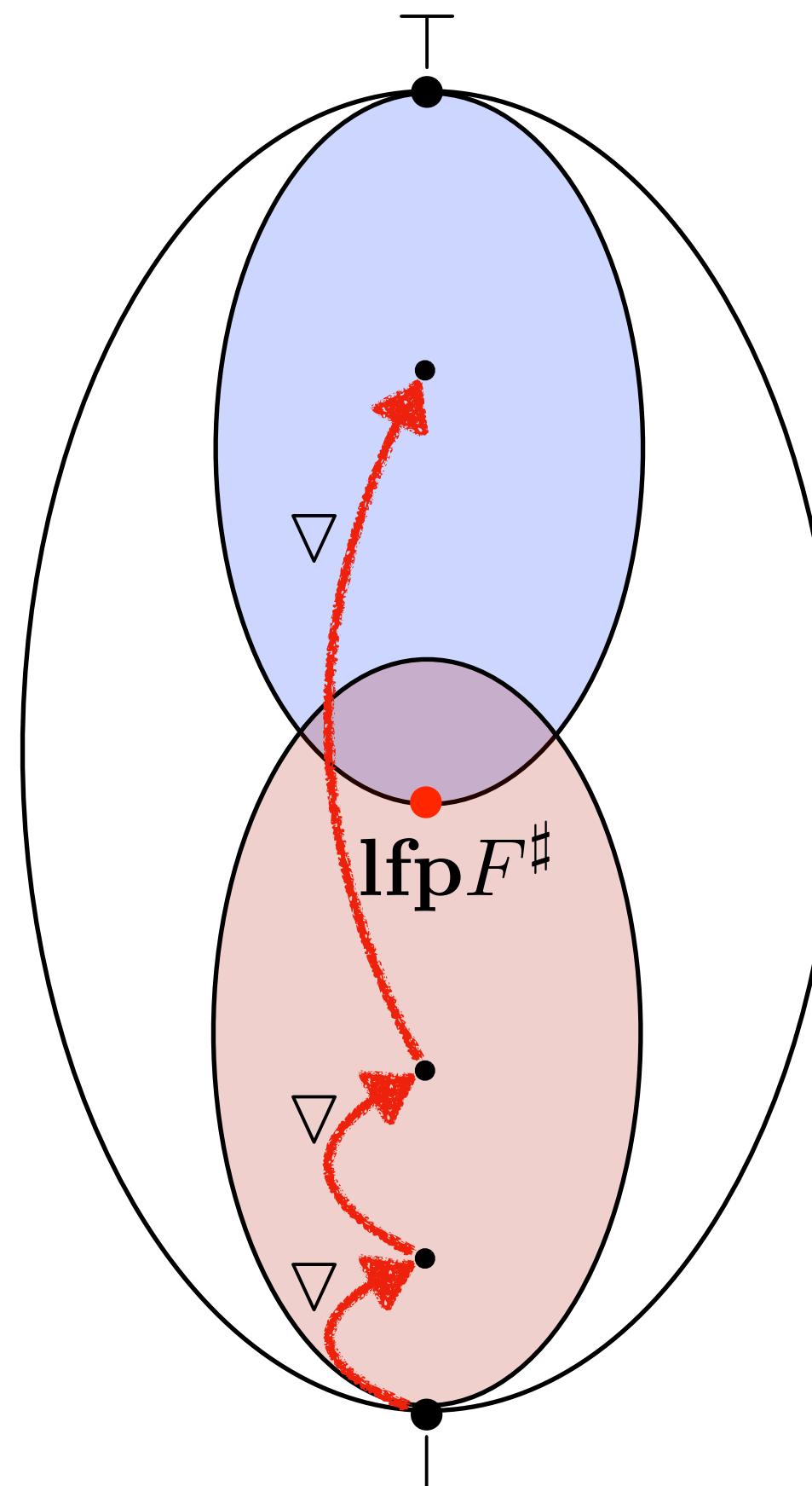
$X_0^\sharp \sqsubseteq X_1^\sharp \sqsubseteq X_2^\sharp \sqsubseteq \dots \sqsubseteq X_{\text{lim}}^\sharp$ such that

$$\bigsqcup_{i \geq 0} F^{\sharp i}(\perp^\sharp) \sqsubseteq X_{\text{lim}}^\sharp$$

Fixed Points



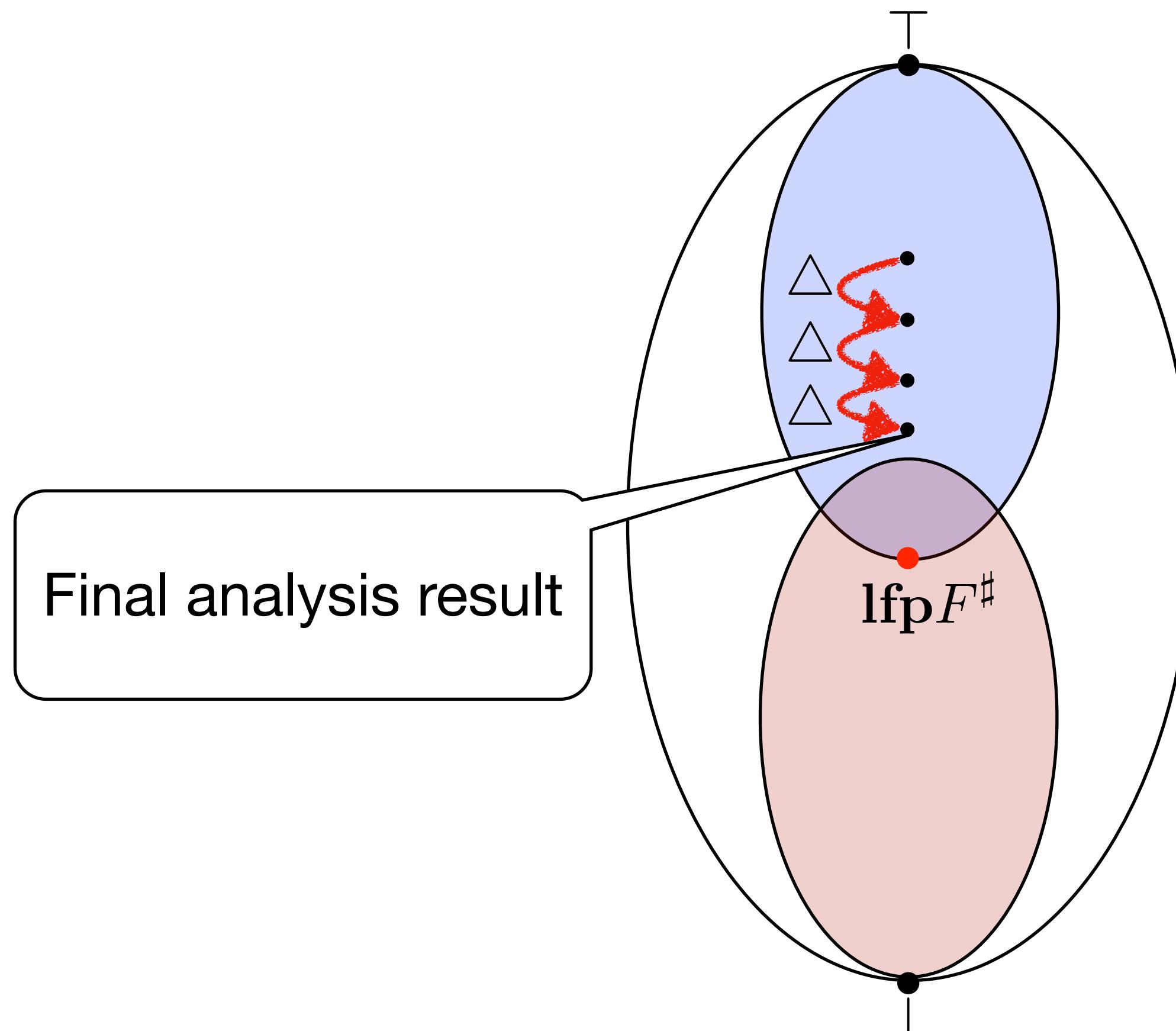
Widening



$$\nabla : \mathbb{D}^\# \times \mathbb{D}^\# \rightarrow \mathbb{D}^\#$$

Widening: enforcing the convergence of fix point iterations

Narrowing



$$\Delta : \mathbb{D}^\# \times \mathbb{D}^\# \rightarrow \mathbb{D}^\#$$

**Narrowing: refining the analysis results
with widening**

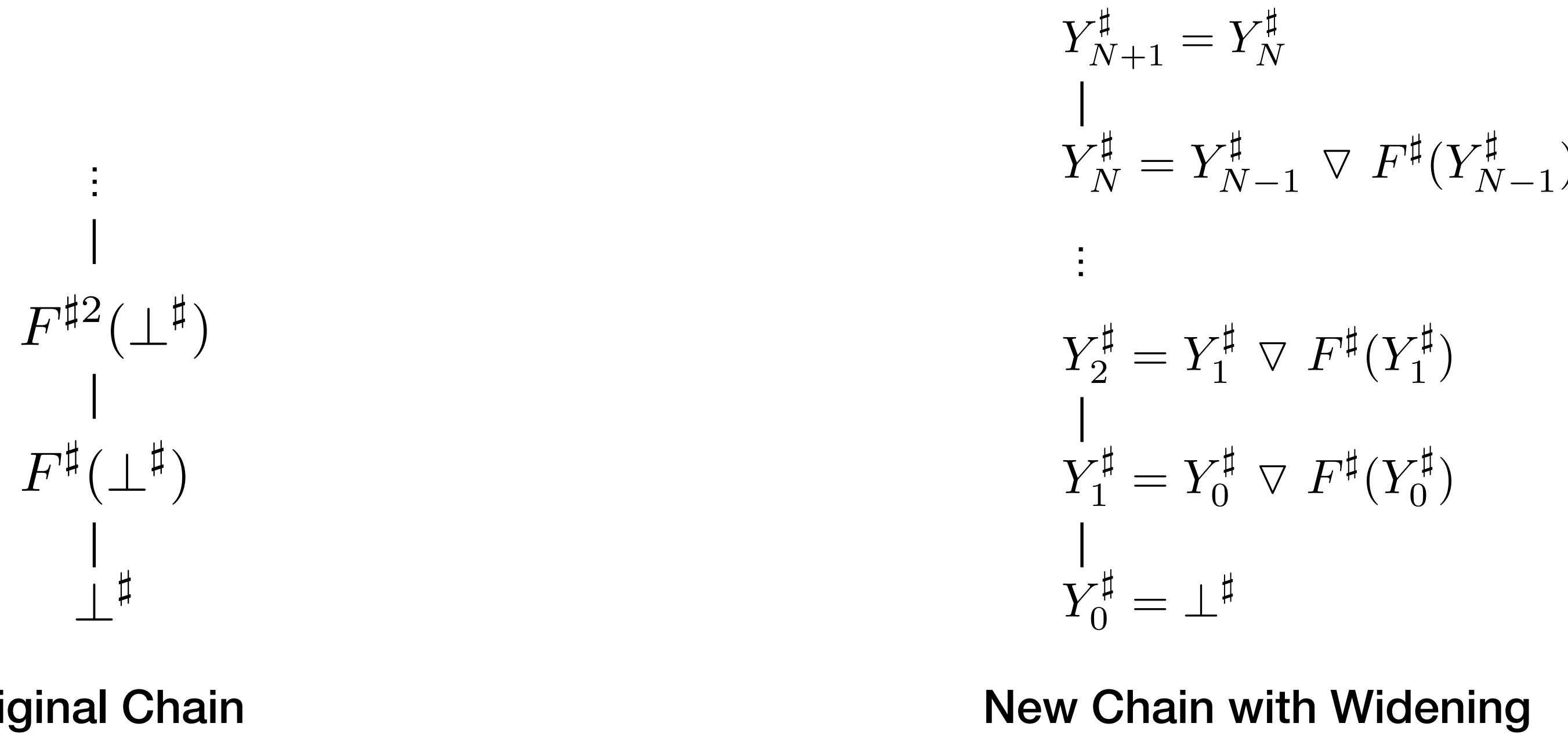
Overshooting by Widening

$$\bigsqcup_{i \geq 0} F^{\sharp i}(\perp^{\sharp}) \sqsubseteq Y_{\lim}^{\sharp}$$

- Define finite chain $\{Y_i\}_i$ by an widening operator $\nabla \in \mathbb{D}^{\sharp} \times \mathbb{D}^{\sharp} \rightarrow \mathbb{D}^{\sharp}$:

$$Y_0^{\sharp} = \perp^{\sharp}$$
$$Y_{i+1}^{\sharp} = \begin{cases} Y_i^{\sharp} & \text{if } F^{\sharp}(Y_i^{\sharp}) \sqsubseteq Y_i^{\sharp} \\ Y_i^{\sharp} \nabla F^{\sharp}(Y_i^{\sharp}) & \text{otherwise} \end{cases}$$

Finite Increasing Chain with Widening



Original Chain

New Chain with Widening

Q. What conditions are required to ensure

$$\bigsqcup_{i \geq 0} F^{\sharp i}(\perp^{\sharp}) \sqsubseteq Y_{\lim}^{\sharp}$$

Safety of Widening Operator

- Conditions on widening operator:
 - $\forall a, b \in \mathbb{D}^\sharp. (a \sqsubseteq a \vee b) \wedge (b \sqsubseteq a \vee b)$
 - \forall increasing chain $\{x_i\}_i$, the following increasing chain $\{y_i\}_i$ is finite:

$$y_0 = x_0$$

$$y_{i+1} = y_i \vee x_{i+1}$$

- Then,
 - Chain $\{Y_i^\sharp\}_i$ is finite
 - $\bigsqcup_{i \geq 0} F^{\sharp i}(\perp^\sharp) \sqsubseteq Y_{\lim}^\sharp$

Proof

Theorem (Widening's Safety). *Let \mathbb{D}^\sharp be a CPO, $F^\sharp : \mathbb{D}^\sharp \rightarrow \mathbb{D}^\sharp$ be a monotone function, and $\nabla : \mathbb{D}^\sharp \times \mathbb{D}^\sharp \rightarrow \mathbb{D}^\sharp$ be a widening operator. Then, chain $\{Y_i^\sharp\}_i$ eventually stabilizes and*

$$\bigsqcup_{i \geq 0} F^{\sharp i}(\perp^\sharp) \sqsubseteq Y_{\lim}^\sharp$$

where Y_{\lim}^\sharp is the greatest element of the chain.

Proof. First we prove chain $\{Y_i^\sharp\}_i$ is finite. According to the second condition on widening operator, it is enough to show that chain $\{F^\sharp(Y_i^\sharp)\}_i$ is increasing. The chain is increasing because 1) $F^\sharp(Y_{i+1}^\sharp)$ is either $F^\sharp(Y_i^\sharp)$ or $F^\sharp(Y_i^\sharp \nabla F^\sharp(Y_i^\sharp))$, 2) $Y_i^\sharp \sqsubseteq Y_i^\sharp \nabla F^\sharp(Y_i^\sharp)$ according to the first condition on widening, and 3) F^\sharp is monotone.

Second, we prove $\bigsqcup_{i \geq 0} F^{\sharp i}(\perp^\sharp) \sqsubseteq Y_{\lim}^\sharp$. It is enough to show that $\forall i \in \mathbb{N}. F^{\sharp i}(\perp^\sharp) \sqsubseteq Y_i^\sharp$ that can be proven by induction. The base case is trivial. The inductive case is as follows:

$$\begin{aligned} F^{\sharp i+1}(\perp^\sharp) &= F^\sharp(F^{\sharp i}(\perp^\sharp)) \\ &\sqsubseteq F^\sharp(Y_i^\sharp) \quad (\text{by induction hypothesis and monotonicity of } F^\sharp) \end{aligned}$$

If $F^\sharp(Y_i^\sharp) \sqsubseteq Y_i^\sharp$, then $Y_{i+1}^\sharp = Y_i^\sharp$ by definition. Therefore, $F^{\sharp i+1}(\perp^\sharp) \sqsubseteq Y_{i+1}^\sharp$.

If $F^\sharp(Y_i^\sharp) \sqsupset Y_i^\sharp$, then $Y_{i+1}^\sharp = Y_i^\sharp \nabla F^\sharp(Y_i^\sharp)$ by definition. According to the first condition on widening, $F^\sharp(Y_i^\sharp) \sqsubseteq Y_i^\sharp \nabla F^\sharp(Y_i^\sharp)$. Therefore, $F^{\sharp i+1}(\perp^\sharp) \sqsubseteq Y_{i+1}^\sharp$.

Proof (A Generalized Ver.)

Theorem (Widening's Safety). Let $\mathbb{D} \xrightleftharpoons[\alpha]{\gamma} \mathbb{D}^\sharp$ be a Galois connection. Let $F : \mathbb{D} \rightarrow \mathbb{D}$ and $F^\sharp : \mathbb{D}^\sharp \rightarrow \mathbb{D}^\sharp$ be a concrete and abstract semantic function. Let $\nabla : \mathbb{D}^\sharp \times \mathbb{D}^\sharp \rightarrow \mathbb{D}^\sharp$ be a widening operator. Then, chain $\{Y_i^\sharp\}_i$ eventually stabilizes and

$$\bigsqcup_{i \geq 0} F(\perp) \sqsubseteq \gamma(Y_{\lim}^\sharp)$$

where Y_{\lim}^\sharp is the greatest element of the chain.

Proof. First we prove chain $\{Y_i^\sharp\}_i$ is finite. According to the second condition on widening operator, it is enough to show that chain $\{F^\sharp(Y_i^\sharp)\}_i$ is increasing. The chain is increasing because 1) $F^\sharp(Y_{i+1}^\sharp)$ is either $F^\sharp(Y_i^\sharp)$ or $F^\sharp(Y_i^\sharp \nabla F^\sharp(Y_i^\sharp))$, 2) $Y_i^\sharp \sqsubseteq Y_i^\sharp \nabla F^\sharp(Y_i^\sharp)$ according to the first condition on widening, and 3) F^\sharp is monotone or extensive.

Second, we prove $\bigsqcup_{i \geq 0} F^i(\perp) \sqsubseteq \gamma(Y_{\lim}^\sharp)$. It is enough to show that $\forall i \in \mathbb{N}. F^i(\perp) \sqsubseteq \gamma(Y_i^\sharp)$ that can be proven by induction. The base case is trivial. The inductive case is as follows:

$$\begin{aligned} F^{i+1}(\perp) &= F \circ F^i(\perp) \\ &\sqsubseteq F \circ \gamma(Y_i^\sharp) \quad (\text{by induction hypothesis and monotonicity of } F) \\ &\sqsubseteq \gamma \circ F^\sharp(Y_i^\sharp) \quad (\text{by } F \circ \gamma \sqsubseteq \gamma \circ F^\sharp) \end{aligned}$$

If $F^\sharp(Y_i^\sharp) \sqsubseteq Y_i^\sharp$, then $Y_{i+1}^\sharp = Y_i^\sharp$ by definition. Therefore, $F^{i+1}(\perp) \sqsubseteq \gamma(Y_{i+1}^\sharp)$ by monotonicity of γ .

If $F^\sharp(Y_i^\sharp) \sqsupset Y_i^\sharp$, then $Y_{i+1}^\sharp = Y_i^\sharp \nabla F^\sharp(Y_i^\sharp)$ by definition. According to the first condition on widening, $F^\sharp(Y_i^\sharp) \sqsubseteq Y_i^\sharp \nabla F^\sharp(Y_i^\sharp)$. Therefore, $F^{\sharp(i+1)}(\perp^\sharp) \sqsubseteq Y_{i+1}^\sharp$ by monotonicity of γ .

Refinement by Narrowing

$$\bigsqcup_{i \geq 0} F^{\sharp i}(\perp^{\sharp}) \sqsubseteq Z_{\text{lim}}^{\sharp}$$

- Define finite chain $\{Z_i^{\sharp}\}_i$ by an narrowing operator $\Delta \in \mathbb{D}^{\sharp} \times \mathbb{D}^{\sharp} \rightarrow \mathbb{D}^{\sharp}$:

$$\begin{aligned} Z_0^{\sharp} &= Y_{\text{lim}}^{\sharp} \\ Z_{i+1}^{\sharp} &= Z_i^{\sharp} \triangle F^{\sharp}(Z_i^{\sharp}) \end{aligned}$$

Finite Decreasing Chain with Narrowing

$$\begin{array}{c} Y_{\lim}^{\sharp} \\ | \\ F^{\sharp}(Y_{\lim}^{\sharp}) \\ | \\ F^{\sharp 2}(Y_{\lim}^{\sharp}) \\ \vdots \end{array}$$

Original Chain

$$\begin{array}{c} Z_0^{\sharp} = Y_{\lim}^{\sharp} \\ | \\ Z_1^{\sharp} = Z_0^{\sharp} \triangle F^{\sharp}(Z_0^{\sharp}) \\ | \\ Z_2^{\sharp} = Z_1^{\sharp} \triangle F^{\sharp}(Z_1^{\sharp}) \\ \vdots \\ Z_N^{\sharp} = Z_{N-1}^{\sharp} \triangle F^{\sharp}(Z_{N-1}^{\sharp}) \\ | \\ Z_{N+1}^{\sharp} = Z_N^{\sharp} \end{array}$$

New Chain with Narrowing

Q. What conditions are required to ensure

$$\bigsqcup_{i \geq 0} F^{\sharp i}(\perp^{\sharp}) \sqsubseteq Z_{\lim}^{\sharp}$$

Safety of Narrowing Operator

- Conditions on narrowing operator:
 - $\forall a, b \in \mathbb{D}^\sharp. a \sqsupseteq b \implies a \sqsupseteq (a \triangle b) \sqsupseteq b$
 - For all decreasing chain $\{y_i\}_i$, the following decreasing chain $\{z_i\}_i$ is finite

$$z_0 = y_0$$

$$z_{i+1} = z_i \triangle y_{i+1}$$

- Then,
 - Decreasing chain $\{Z_i^\sharp\}_i$ is finite
 - $\bigsqcup_{i \geq 0} F^{\sharp i}(\perp^\sharp) \sqsubseteq Z_{\lim}$

Proof

Theorem (Narrowing's Safety). *Let \mathbb{D}^\sharp be a CPO, $F^\sharp : \mathbb{D}^\sharp \rightarrow \mathbb{D}^\sharp$ be a monotone function, and $\Delta : \mathbb{D}^\sharp \times \mathbb{D}^\sharp \rightarrow \mathbb{D}^\sharp$ be a narrowing operator. Then, chain $\{Z_i^\sharp\}_i$ eventually stabilizes and*

$$\bigsqcup_{i \geq 0} F^{\sharp i}(\perp^\sharp) \sqsubseteq Z_{\lim}^\sharp$$

where Z_{\lim}^\sharp is the least element of the chain.

Proof. First we prove chain $\{Z_i^\sharp\}_i$ is finite. According to the second condition on narrowing operator, it is enough to show that chain $\{F^\sharp(Z_i^\sharp)\}_i$ is decreasing. The chain is decreasing if $\forall i \in \mathbb{N}. Z_i^\sharp \sqsupseteq F^\sharp(Z_i^\sharp)$, because

$$\begin{aligned} & Z_i^\sharp \sqsupseteq F^\sharp(Z_i^\sharp) \\ \implies & Z_i^\sharp \sqsupseteq (Z_i^\sharp \Delta F^\sharp(Z_i^\sharp)) \sqsupseteq F^\sharp(Z_i^\sharp) \quad (\text{by the first condition on narrowing}) \\ \implies & F^\sharp(Z_i^\sharp) \sqsupseteq F^\sharp(Z_i^\sharp \Delta F^\sharp(Z_i^\sharp)) \quad (\text{by monotonicity of } F^\sharp) \\ \implies & F^\sharp(Z_i^\sharp) \sqsupseteq F^\sharp(Z_{i+1}^\sharp) \quad (\text{by definition of } Z_{i+1}^\sharp) \end{aligned}$$

We prove $\forall i \in \mathbb{N}. Z_i^\sharp \sqsupseteq F^\sharp(Z_i^\sharp)$ by induction. The base case is true by definition of Z_{\lim}^\sharp from the increasing chain by widening. The inductive case is as follows:

$$\begin{aligned} & Z_i^\sharp \sqsupseteq F^\sharp(Z_i^\sharp) \quad (\text{by induction hypothesis}) \\ \implies & Z_i^\sharp \sqsupseteq Z_i^\sharp \Delta F^\sharp(Z_i^\sharp) \sqsupseteq F^\sharp(Z_i^\sharp) \quad (\text{by the first condition on narrowing}) \\ \implies & Z_i^\sharp \sqsupseteq Z_{i+1}^\sharp \sqsupseteq F^\sharp(Z_i^\sharp) \quad (\text{by definition}) \\ \implies & Z_i^\sharp \sqsupseteq F^\sharp(Z_i^\sharp) \sqsupseteq F^\sharp(Z_{i+1}^\sharp) \quad (\text{by monotonicity of } F^\sharp) \\ \implies & Z_i^\sharp \sqsupseteq F^\sharp(Z_{i+1}^\sharp) \end{aligned}$$

Proof

Theorem (Narrowing's Safety). *Let \mathbb{D}^\sharp be a CPO, $F^\sharp : \mathbb{D}^\sharp \rightarrow \mathbb{D}^\sharp$ be a monotone function, and $\Delta : \mathbb{D}^\sharp \times \mathbb{D}^\sharp \rightarrow \mathbb{D}^\sharp$ be a narrowing operator. Then, chain $\{Z_i^\sharp\}_i$ eventually stabilizes and*

$$\bigsqcup_{i \geq 0} F^{\sharp i}(\perp^\sharp) \sqsubseteq Z_{\lim}^\sharp$$

where Z_{\lim}^\sharp is the least element of the chain.

(Cont'd) Second we prove chain $\bigsqcup_{i \geq 0} F^{\sharp i}(\perp^\sharp) \sqsubseteq Z_{\lim}^\sharp$. It is enough to show that $\forall i \in \mathbb{N}. F^{\sharp i}(\perp^\sharp) \sqsubseteq Z_i^\sharp$ that can be proven by induction. The base case is trivial. The inductive case is as follows:

$$\begin{aligned} F^{\sharp i+1}(\perp^\sharp) &= F^\sharp \circ F^{\sharp i}(\perp^\sharp) \\ &\sqsubseteq F^\sharp(Z_i^\sharp) && \text{(by induction hypothesis and monotonicity of } F^\sharp) \\ &\sqsubseteq Z_i^\sharp \triangle F^\sharp(Z_i^\sharp) && \text{(by condition } \forall i \in \mathbb{N}. Z_i^\sharp \sqsupseteq F^\sharp(Z_i^\sharp)) \\ &= Z_{i+1}^\sharp && \text{(by definition)} \end{aligned}$$

Summary

- Abstract interpretation: a **framework** for designing correct static analysis
- **Soundness** guarantee by Fixpoint transfer theorem
 - Galois connection, sound abstract semantic function
- Widening: **termination** guarantee
- Narrowing: **refinement** of widening results