Verification of Real-Time Systems

Foundations of Abstract Interpretation

Jan Reineke

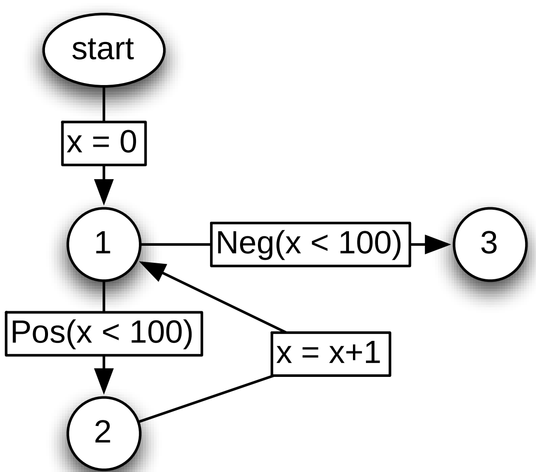
Advanced Lecture, Summer 2015

Recap: Reachability Semantics

Can be captured as the least solution of:

$$Reach(start) = States$$

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



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

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

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

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?

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 $\bigsqcap A$).

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

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

Which of these are complete lattices?

Least Upper Bounds: Examples II

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

Which of these are complete lattices?

Properties of Complete Lattices

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

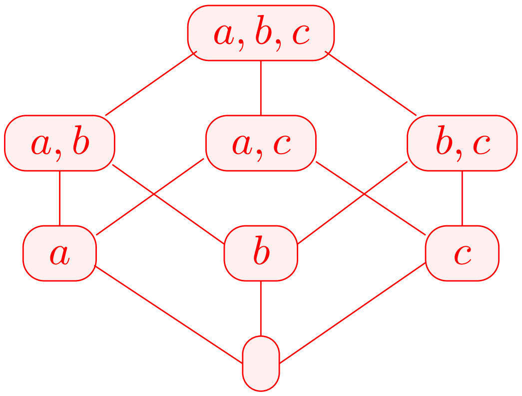
- a *least* element (*bottom* element): $\bot = \bigsqcup \emptyset$, and
- a *greatest* element (*top* element): $\top = \bigsqcup D$.

Generic Lattice Constructions: Power-set Lattice

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

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

Graphical representation (Hasse diagram):



Generic Lattice Constructions: Total Function Space

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

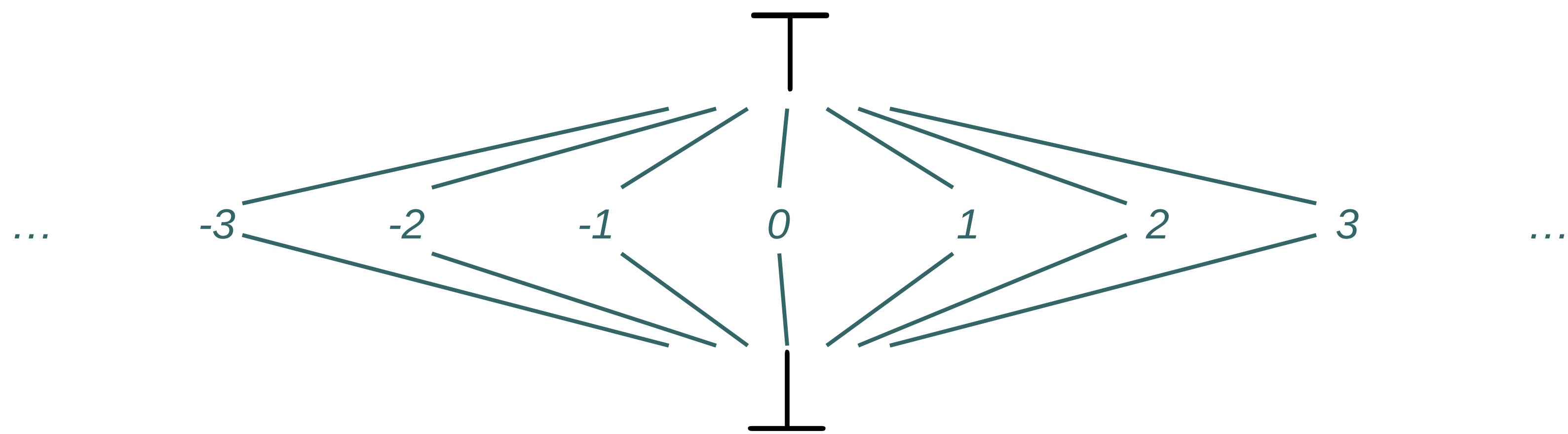
$$\begin{array}{rccc}
\text{“\textit{join}”:} & \bigsqcup A & = & \lambda s. \bigsqcup_{f \in A} f(s) \\
\text{“\textit{meet}”:} & \bigsqcap A & = & \lambda s. \bigsqcap_{f \in A} f(s) \\
\text{“\textit{top}”:} & \top & = & \lambda s. \top_L \\
\text{“\textit{bottom}”:} & \bot & = & \lambda s. \bot_L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

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

Generic Lattice Constructions: Flat Lattice

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

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:

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

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:

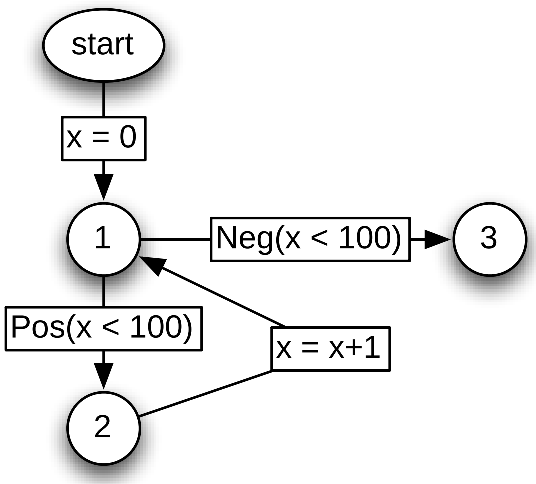
- What is a complete lattice? ✓
- What is a monotonic function? ✓
- What is a fixed point? ✓

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?

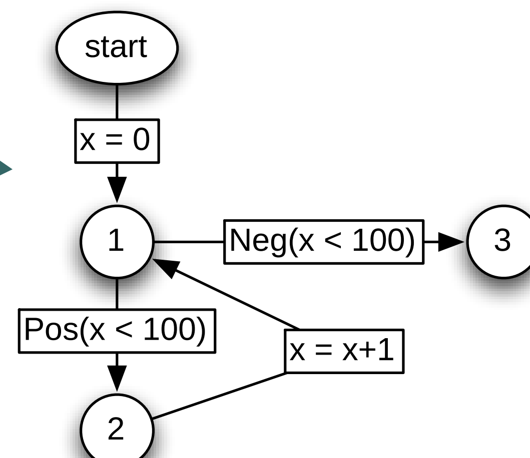
$$Reach(1) = \{0\} \cup \{v + 1 \mid v \in Reach(2)\}$$
$$Reach(2) = Reach(1) \cap \{\ldots, 98, 99\}$$
$$Reach(3) = Reach(1) \cap \{100, 101, \ldots\}$$

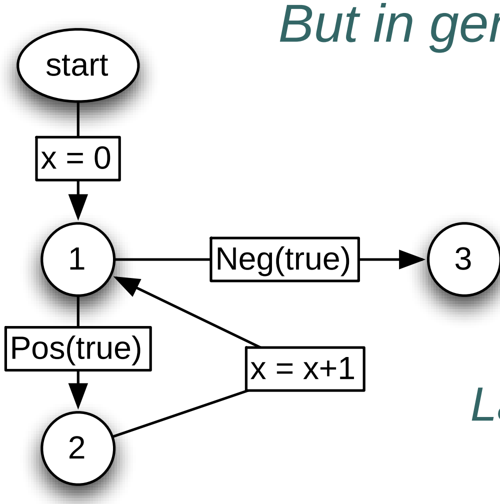
How to Compute the Least Fixed Point

Kleene Iteration:

$$\perp \leq f(\perp) \leq f^2(\perp) \leq f^3(\perp) \leq \ldots$$

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

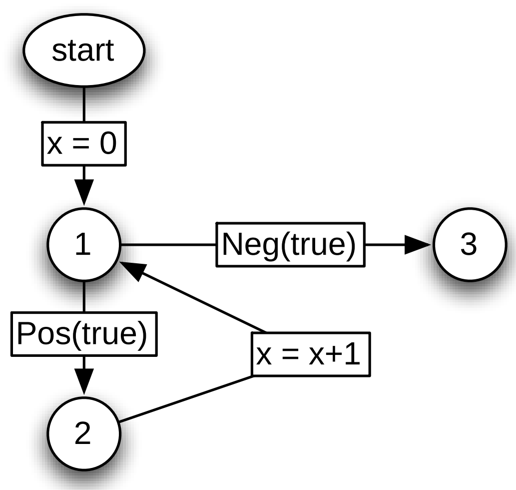




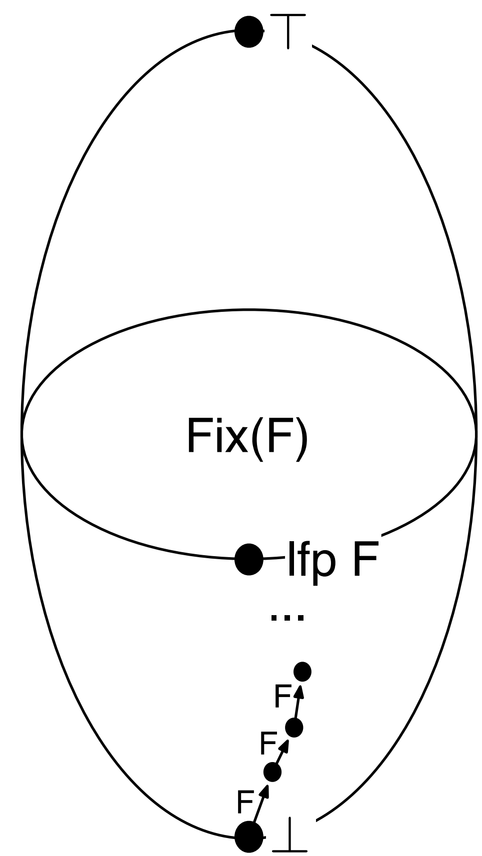
No!

Lattice has infinite ascending chains.

Infinite Ascending Chains



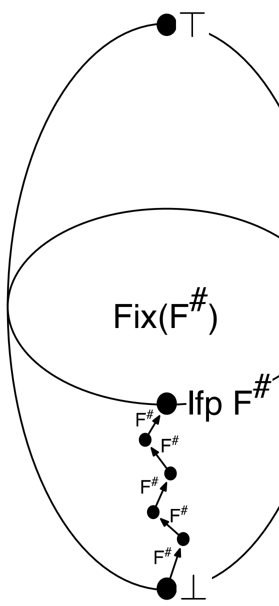
No!



Think of an example of an infinite ascending chain.

Ascending Chain Condition

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



Theorem (Ascending Chain Condition):

Let $(S, \leq)$ *be a* complete lattice *set that satisfies the ascending chain condition, and let* $f : S \to S$ *be a monotone function. Then, there is an* $n \in \mathbb{N}$*, such that*

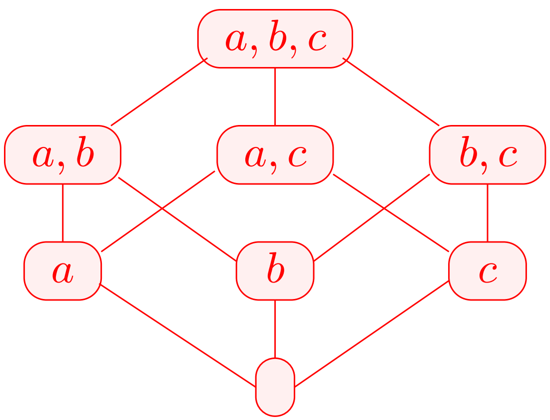
$$lfp\ f = f^n(\bot).$$

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

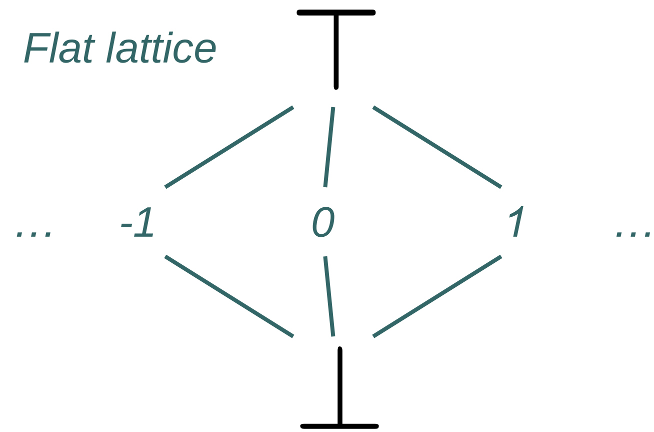
Ascending Chain Condition: Examples

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

Power set lattice



Flat lattice



→ *Ascending chain condition does not imply finite partially-ordered set!*

How about total function space lattice?
How about finite partially-ordered sets?

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:

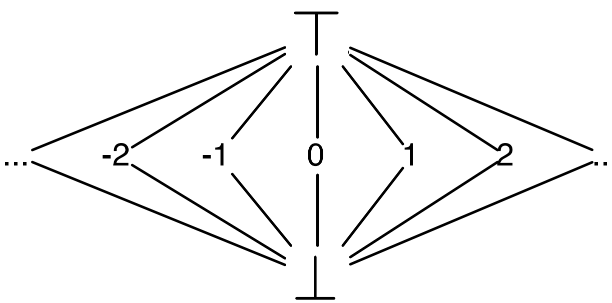
- A complete lattice $(L^{\#}, \leq)$ as the domain for abstract elements
- A monotone function $F^{\#}$ corresponding to the concrete function F
- Then the abstract semantics is the least fixed point of $F^{\#}$, lfp $F^{\#}$

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

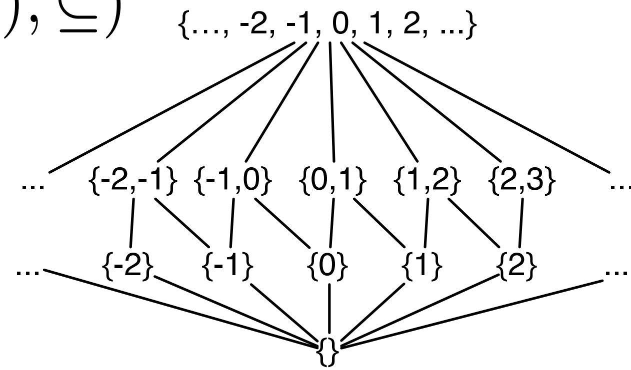
then lfp $F^{\#}$ "correctly approximates" lfp F.

An Example Abstract Domain for Values of Variables

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



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



How to relate the two?

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

$$\gamma : \mathbb{Z}_\bot^\top \to \mathcal{P}(\mathbb{Z})$$

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

$$\alpha : \mathcal{P}(\mathbb{Z}) \to \mathbb{Z}_\bot^\top$$

Relation between the Abstract and Concrete Domains

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

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

1. *Are these functions monotone?*
2. *Should they be?*
3. *What is the meaning of the partial order in the abstract domain?*
4. *What if we first abstract and the concretize?*

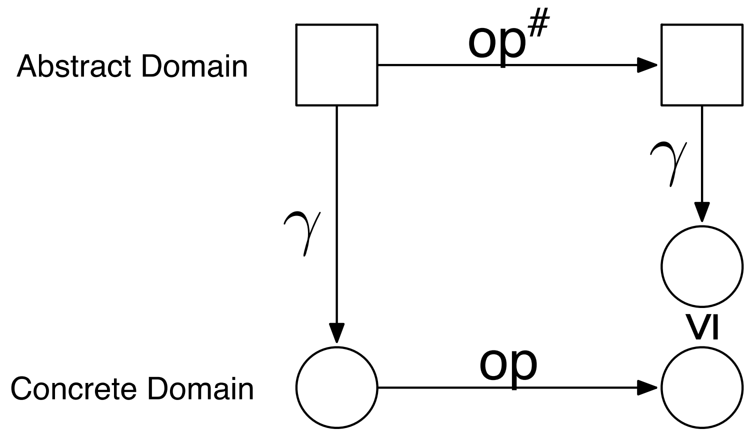
How to Compute in the Abstract Domain Example: Multiplication on Flat Lattice

Denotes abstract version of operator

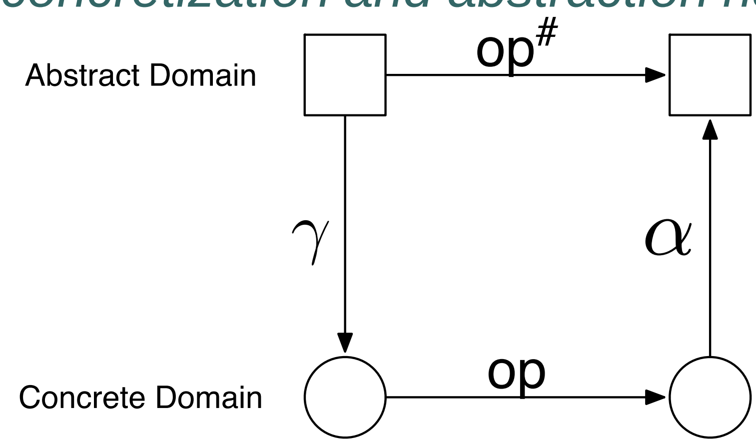
$*^{\#}$	$\top$	a	0	$\bot$
$\top$				
b				
0				
$\bot$				

How to Compute in the Abstract Domain: Correctness Conditions

Correctness Condition:



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



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

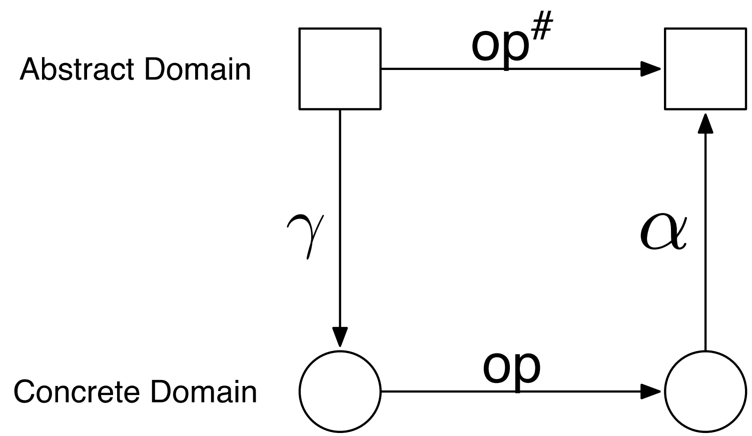
$$
\begin{tikzcd}
\top \arrow[d, "\gamma"'] & 0 \arrow[rr, "*^{\#}"] \arrow[d, "\gamma"'] & & 0 \\
\mathbb{Z} & \{0\} \arrow[rr, "*"] & & \{0\} \arrow[u, "\alpha"]
\end{tikzcd}
$$

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

$$
\begin{tikzcd}
a \arrow[d, "\gamma"'] & b \arrow[d, "\gamma"'] \arrow[r, "*^{\#}"] & a*b \\
\{a\} & \{b\} \arrow[r, "*"] & \{a * b\} \arrow[u, "\alpha"]
\end{tikzcd}
$$

How to Compute in the Abstract Domain: Correct by Construction

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



"Certain properties": Notion of Galois connections:

Let $(L, \leq)$ *and* $(M, \sqsubseteq)$ *be partially ordered sets and* $\alpha \in L \to M$, $\gamma \in M \to L$. *We call* $(L, \leq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (M, \sqsubseteq)$ *a* Galois connection *if* α *and* γ *are monotone functions and*

$$
\begin{aligned}
l &\leq \gamma(\alpha(l)) \\
\alpha(\gamma(m)) &\sqsubseteq m
\end{aligned}
$$

for all $l \in L$ *and* $m \in M$.

Galois connections

Notion of Galois connections:

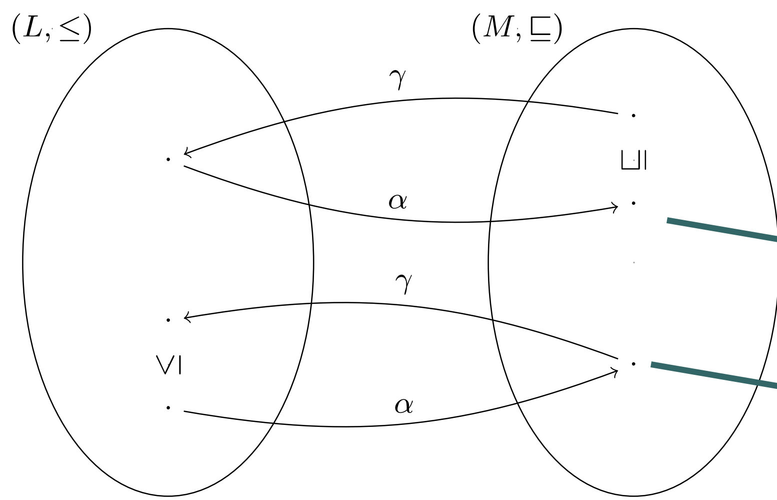
Let $(L, \leq)$ *and* $(M, \sqsubseteq)$ *be partially ordered sets and* $\alpha \in L \to M$, $\gamma \in M \to L$. *We call* $(L, \leq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (M, \sqsubseteq)$ *a* Galois connection *if* α *and* γ *are monotone functions and*

$$\begin{array}{ccc} l & \leq & \gamma(\alpha(l)) \\ \alpha(\gamma(m)) & \sqsubseteq & m \end{array}$$

for all $l \in L$ *and* $m \in M$.

Why monotone?

Graphically:

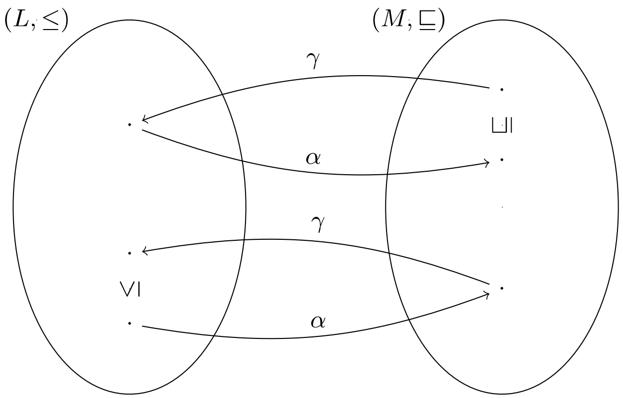


For precision.

For soundness.

Galois connections: Properties

Graphically:



Properties:

1) *Can be used to systematically construct correct (and in fact the most precise) abstract operations:* $op^{\#} = \alpha \circ op \circ \gamma$

2) *a) Abstraction function induces concretization function*

   b) Concretization function induces abstraction function

Why?

How?

Abstracting Sets of Concrete States
Recap: Concrete States

Concrete states are not just sets of values...

Concrete states consist of variables and memory:

$$s = (\rho, \mu) \in States$$

$$\rho : Vars \to int$$ Values of Variables

$$\mu : \mathbb{N} \to int$$ Contents of Memory

$$\cancel{States = (Vars \to int) \times (\mathbb{N} \to int)}$$

$$States = (Vars \to \mathbb{Z}) \times (\mathbb{N} \to \mathbb{Z})$$

Abstracting Sets of Concrete States
Recap: Concrete States

Reachability semantics is defined on sets of states:

$$[\![statement]\!] \subseteq States \times States$$

$$[\![statement]\!] : \mathcal{P}(States) \to \mathcal{P}(States)$$

$$[\![statement]\!](S) := \{s' \mid \exists s \in S : (s, s') \in [\![statement]\!]\}$$

$$\mathcal{P}(States) = \mathcal{P}((Vars \to \mathbb{Z}) \times (\mathbb{N} \to \mathbb{Z}))$$

Relation between Concrete Domain and Abstract Domain

Concrete domain!

$$\mathcal{P}(States) = \\ \quad \mathcal{P}((Vars \to \mathbb{Z}) \times (\mathbb{N} \to \mathbb{Z}))$$

Abstract domain?

$$\widehat{States} = Vars \to \mathbb{Z}_{\bot}^{\top}$$

Relation between the two?

→ For ease of understanding, introduce Intermediate domain:

$$Power\widehat{SetS}tates = Vars \to \mathcal{P}(\mathbb{Z})$$

Relation between Concrete Domain and Intermediate Domain

Concrete domain:

$$\mathcal{P}(States) = \mathcal{P}((Vars \to \mathbb{Z}) \times (\mathbb{N} \to \mathbb{Z}))$$

Intermediate domain:

$$\widehat{PowerSetStates} = Vars \to \mathcal{P}(\mathbb{Z})$$

Abstraction:

$$\alpha_{C,I} : \mathcal{P}(States) \to \widehat{PowerSetStates}$$

$$\alpha_{C,I}(C) := \lambda x \in Vars.\{v(x) \in \mathbb{Z} \mid (v,m) \in C\}$$

Concretization:

$$\gamma_{I,C} : \widehat{PowerSetStates} \to \mathcal{P}(States)$$

$$\gamma_{I,C}(\widehat{c}) := \{(v,m) \in States \mid \forall x \in Vars : v(x) \in \widehat{c}(x)\}$$

Relation between Intermediate Domain and Abstract Domain

Intermediate domain:

$$\widehat{PowerSetStates} = Vars \to \mathcal{P}(\mathbb{Z})$$

Abstract domain:

$$\widehat{States} = Vars \to \mathbb{Z}_{\bot}^{\top}$$

Abstraction:

$$\alpha_{I,A} : \widehat{PowerSetStates} \to \widehat{States}$$

$$\alpha(\widehat{c}) := \lambda x \in Vars.\alpha(c(x))$$

Concretization:

$$\gamma_{A,I} : \widehat{States} \to \widehat{PowerSetStates}$$

$$\gamma(\widehat{a}) := \lambda x \in Vars.\gamma(\widehat{a}(x))$$

Abstraction and Concretization functions from before!

Could plug in other abstractions for sets of values…

Relation between Concrete Domain and Abstract Domain

Concrete domain:

$$\mathcal{P}(States) = \mathcal{P}((Vars \to \mathbb{Z}) \times (\mathbb{N} \to \mathbb{Z}))$$

Abstract domain:

$$\widehat{States} = Vars \to \mathbb{Z}_{\perp}^{\top}$$

Abstraction:

$$\alpha_{C,A} : \mathcal{P}(States) \to \widehat{States}$$

$$\alpha_{C,A} := \alpha_{I,A} \circ \alpha_{C,I}$$

Concretization:

$$\gamma_{A,C} : \widehat{States} \to \mathcal{P}(States)$$

$$\gamma_{A,C} := \gamma_{I,C} \circ \gamma_{A,I}$$

Galois connections can be composed to obtain new Galois connections.

Meaning of Statements in the Abstract Domain

$$[\![R = e]\!]^\#(\widehat{a}) := \widehat{a}[R \mapsto [\![e]\!]^\#(\widehat{a})]$$

$$[\![R = M[e]]\!]^\#(\widehat{a}) := \widehat{a}[R \mapsto \top]$$

$$[\![M[e_1] = e_2]\!]^\#(\widehat{a}) := \widehat{a}$$

$$[\![Pos(e)]\!]^\#(\widehat{a}) := \widehat{a}$$

$$[\![Neg(e)]\!]^\#(\widehat{a}) := \widehat{a}$$

Can this be done better?

Again:

For Correctness:

$$\begin{tikzcd}
\text{Abstract Domain} & \square \arrow[r, "{[\![statement]\!]^\#}"] \arrow[dd, "\gamma"'] & \square \arrow[d, "\gamma"'] \\
 & & \bigcirc \arrow[d, phantom, "\mathrm{VI}"] \\
\text{Concrete Domain} & \bigcirc \arrow[r, "{[\![statement]\!]}"] & \bigcirc
\end{tikzcd}$$

For the best possible precision:

$$\begin{tikzcd}
\text{Abstract Domain} & \square \arrow[r, "{[\![statement]\!]^\#}"] \arrow[d, "\gamma"'] & \square \\
\text{Concrete Domain} & \bigcirc \arrow[r, "{[\![statement]\!]}"] & \bigcirc \arrow[u, "\alpha"']
\end{tikzcd}$$

Meaning of Expressions

Evaluation of expressions is as expected:

$$[\![x]\!]^{\#}(\widehat{a}) := \widehat{a}(x) \qquad\qquad if\ x \in Vars$$

$$[\![e_1\ \ op\ e_2]\!]^{\#}(\widehat{a}) := [\![e_1]\!]^{\#}(\widehat{a})\ \ op^{\#}\ \ [\![e_2]\!]^{\#}(\widehat{a})$$

As we have seen earlier!

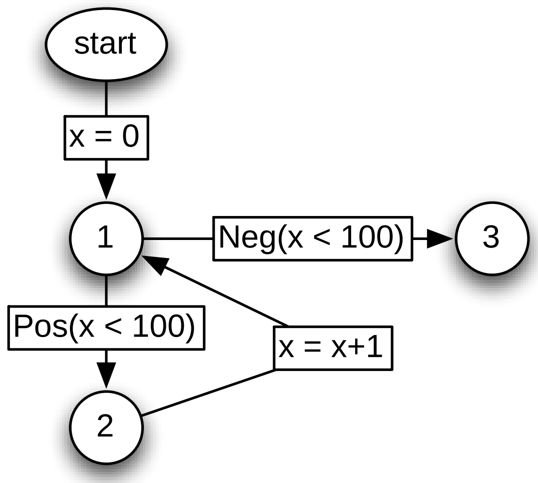
Putting it all together: The Abstract Reachability Semantics

Abstract Reachability Semantics captured as least fixed point of:

$$\widehat{Reach} : V \to \widehat{States}$$

$$\widehat{Reach}(start) = \top$$

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



$$\widehat{Reach}(1) = [\![labeling(start, 1)]\!]^{\#}(\widehat{Reach}(start)) \sqcup [\![labeling(2, 1)]\!](\widehat{Reach}(2))$$

$$\widehat{Reach}(2) = [\![labeling(1, 2)]\!]^{\#}(\widehat{Reach}(1))$$

$$\widehat{Reach}(3) = [\![labeling(1, 3)]\!]^{\#}(\widehat{Reach}(1))$$

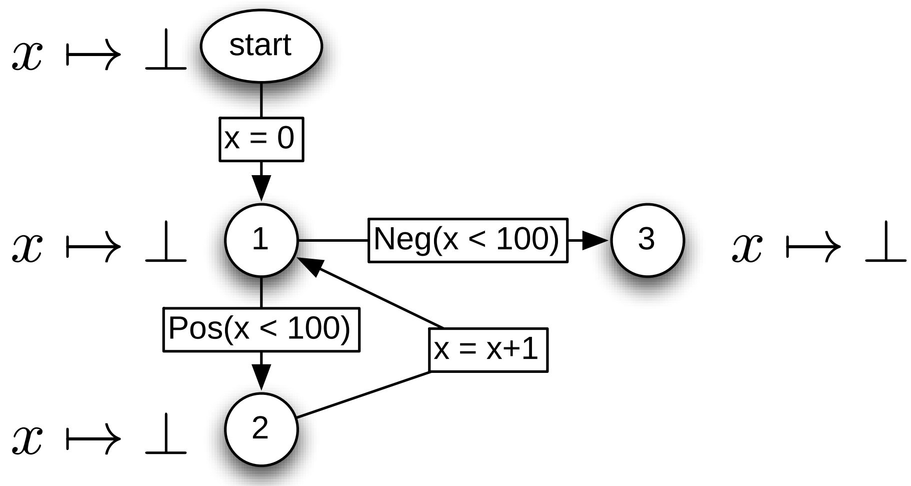
$$\Downarrow$$

$$\widehat{Reach}(1) = [\![x = 0]\!]^{\#}(\widehat{Reach}(start)) \sqcup [\![x = x + 1]\!]^{\#}(\widehat{Reach}(2))$$

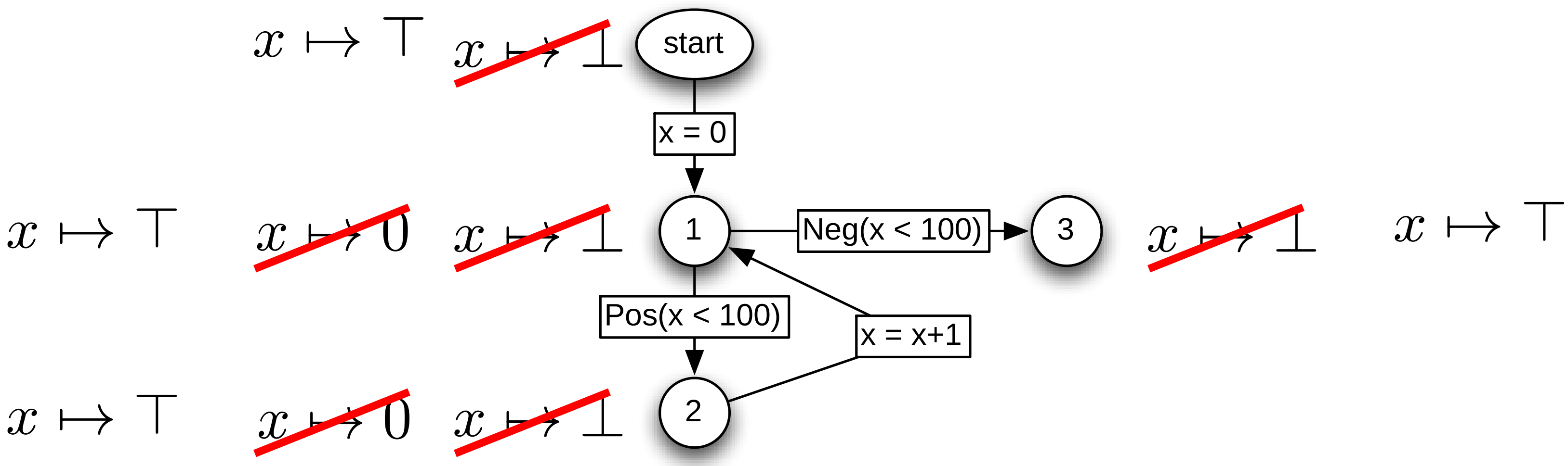
$$\widehat{Reach}(2) = [\![Pos(x < 100)]\!]^{\#}(\widehat{Reach}(1))$$

$$\widehat{Reach}(3) = [\![Neg(x < 100)]\!]^{\#}(\widehat{Reach}(1))$$

Example: Kleene Iteration to Compute Abstract Reachability Semantics

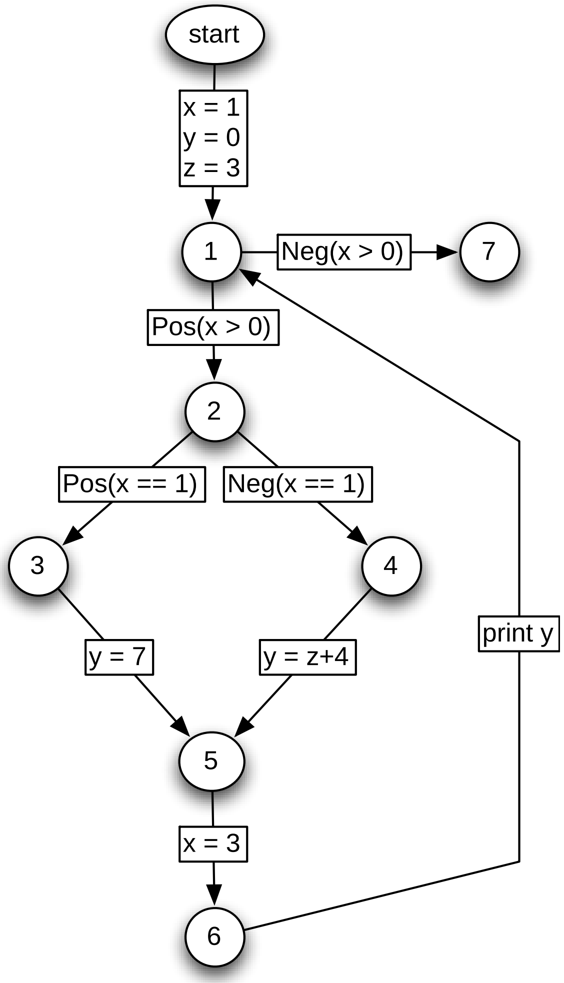


Example: Kleene Iteration to Compute Abstract Reachability Semantics



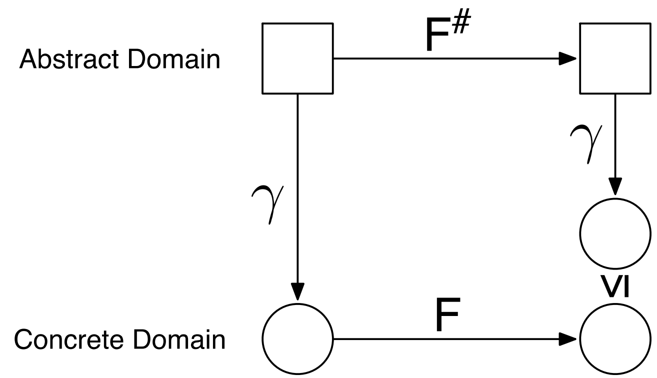
Example II: Kleene Iteration to Compute Abstract Reachability Semantics

```
y = 0;
x = 1;
z = 3;
while (x > 0) {
    if (x == 1) {
        y = 7;
    }
    else {
        y = z+4;
    }
    x = 3;
    print y;
}
```

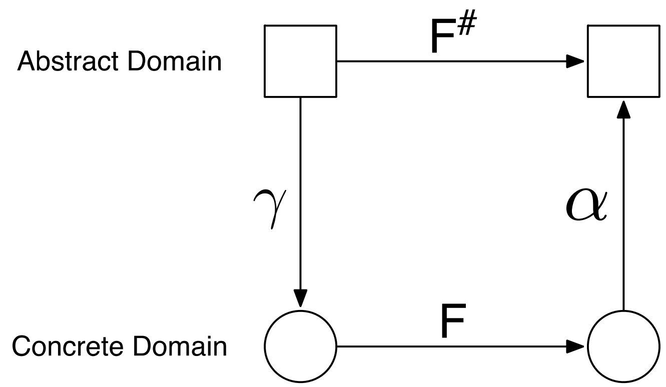


The Abstract Transformer $F^{\#}$

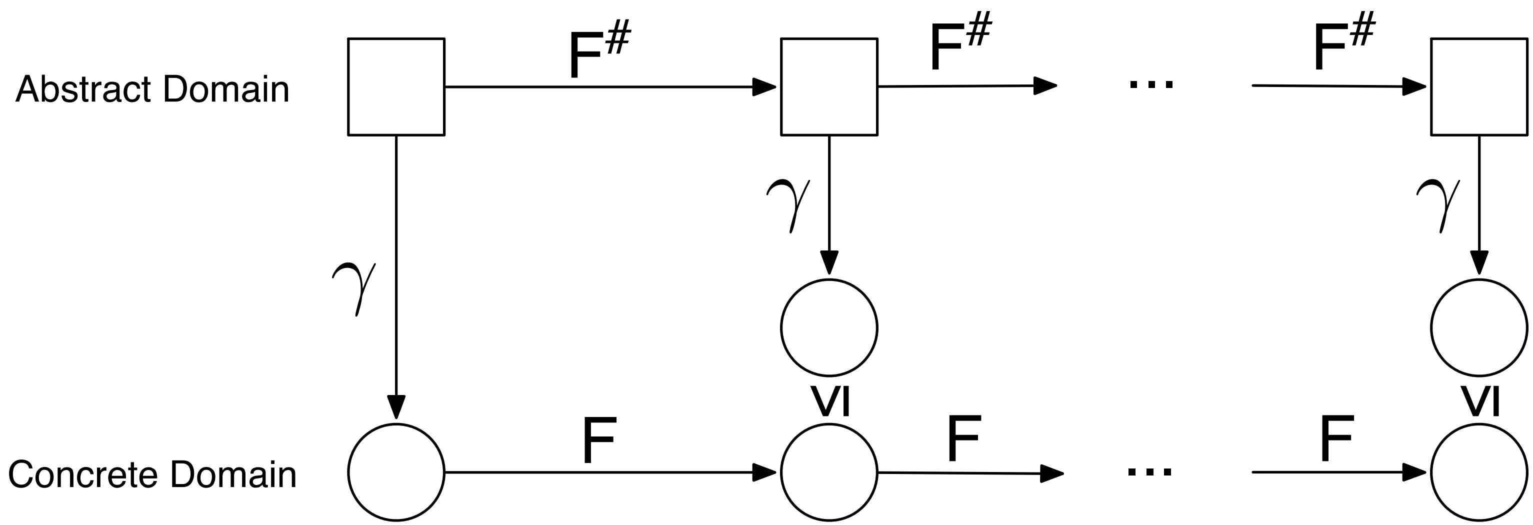
Local Correctness Condition:



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



From Local to Global Correctness: Kleene Iteration



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 $F^{\#} \to F^{\#}$ two monotone functions, with

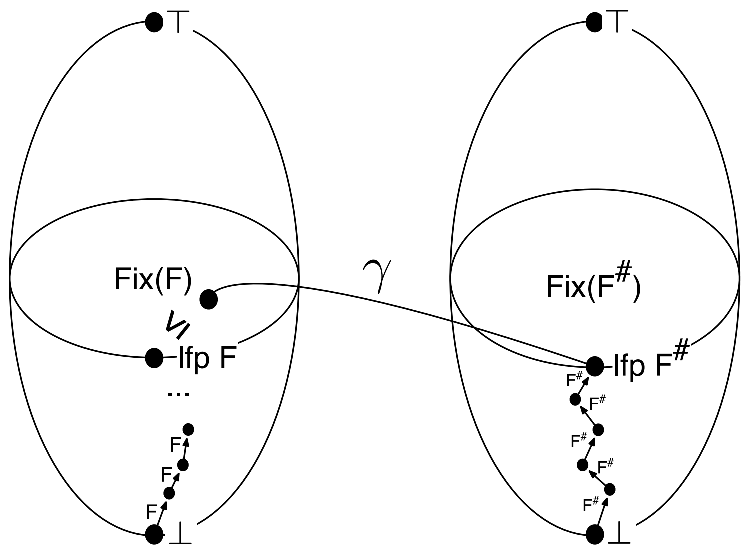
$$\forall l^{\#} \in L^{\#} : \gamma(F^{\#}(l^{\#})) \geq F(\gamma(l^{\#})).$$

Local Correctness

Then:

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

Global Correctness



Outlook: Other Abstractions

- Signs
- Parity
- Intervals
- Octagons
- Congruence