# Software Verification

Grégoire Sutre

LaBRI, University of Bordeaux, CNRS, France

Summer School on Verification Technology, Systems & Applications

September 2008

Part 2

Part IV

# Abstract Interpretation

# Outline — Abstract Interpretation

8 Some More Lattice Theory: Galois Connections

9 Abstract Interpretation-Based Data Flow Analysis

10 Convergence Acceleration with Widening and Narrowing

# Outline — Abstract Interpretation

8 Some More Lattice Theory: Galois Connections

9 Abstract Interpretation-Based Data Flow Analysis

10 Convergence Acceleration with Widening and Narrowing

# Concrete Lattice & Abstract Lattice: Notations

## Concrete lattice

$$(L, \sqsubseteq)$$

## Abstract lattice

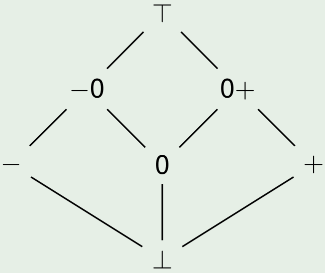
$$(\overline{L}, \overline{\sqsubseteq})$$

## Example (Sets of Values)

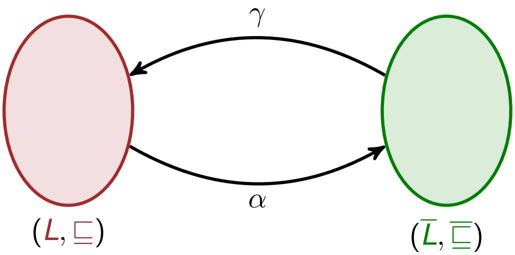
For a variable ranging over a domain $\mathbb{D}$:

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

## Example (Sign Lattice)



# Galois Connections: Definition



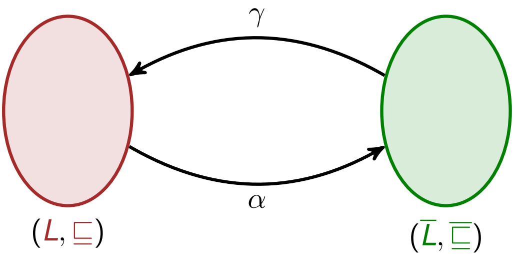
## Definition

A Galois connection between a lattice $(L, \sqsubseteq)$ and a lattice $(\overline{L}, \overline{\sqsubseteq})$ is a pair of functions $(\alpha, \gamma)$, with $\alpha : L \to \overline{L}$ and $\gamma : \overline{L} \to L$, satisfying:

$$\alpha(x) \overline{\sqsubseteq}\, \overline{y} \quad \text{iff} \quad x \sqsubseteq \gamma(\overline{y}) \qquad\qquad (\text{for all } x \in L, \overline{y} \in \overline{L})$$

Notation for Galois connections: $(L, \sqsubseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (\overline{L}, \overline{\sqsubseteq})$

# Galois Connections: Definition



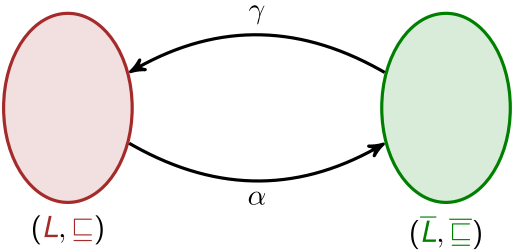
## Definition

A Galois connection between a lattice $(L, \sqsubseteq)$ and a lattice $(\overline{L}, \overline{\sqsubseteq})$ is a pair of functions $(\alpha, \gamma)$, with $\alpha : L \rightarrow \overline{L}$ and $\gamma : \overline{L} \rightarrow L$, satisfying:

$$\alpha(x) \mathrel{\overline{\sqsubseteq}} \overline{y} \quad \text{iff} \quad x \sqsubseteq \gamma(\overline{y}) \qquad\qquad (\text{for all } x \in L, \overline{y} \in \overline{L})$$

Notation for Galois connections: $(L, \sqsubseteq) \overset{\gamma}{\underset{\alpha}{\leftrightarrows}} (\overline{L}, \overline{\sqsubseteq})$

# Galois Connections: Intuition



## Concretization

$\gamma$ is the concretization function.

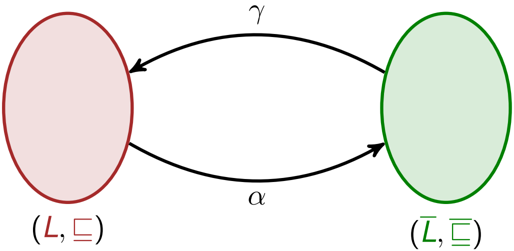
$\gamma(\overline{y})$ is the concrete value in $L$ that is represented by $\overline{y}$.

## Abstraction

$\alpha$ is the abstraction function.

$\alpha(x)$ is the most precise abstract value in $\overline{L}$ whose concretization approximates $x$.

# Galois Connections: Intuition



### Concretization

$\gamma$ is the concretization function.

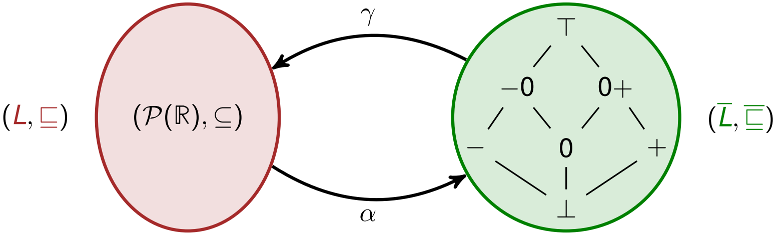
$\gamma(\overline{y})$ is the concrete value in $L$ that is represented by $\overline{y}$.

### Abstraction

$\alpha$ is the abstraction function.

$\alpha(x)$ is the most precise abstract value in $\overline{L}$ whose concretization approximates $x$.

# Galois Connections: Example

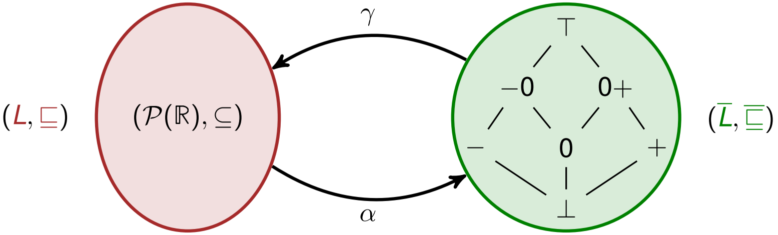


$(L, \sqsubseteq)$
$(\mathcal{P}(\mathbb{R}), \subseteq)$
$\gamma$
$\alpha$
$\top$
$-0$
$0+$
$-$
$0$
$+$
$\bot$
$(\overline{L}, \overline{\sqsubseteq})$

$$\alpha(x) = \begin{cases} \bot & \text{if } x = \emptyset \\ - & \text{if } x \subseteq \{r \in \mathbb{R} \mid r < 0\} \\ 0 & \text{if } x = \{0\} \\ + & \text{if } x \subseteq \{r \in \mathbb{R} \mid r > 0\} \\ -0 & \text{if } \{0\} \subset x \subseteq \{r \in \mathbb{R} \mid r \leq 0\} \\ 0+ & \text{if } \{0\} \subset x \subseteq \{r \in \mathbb{R} \mid r \geq 0\} \\ \top & \text{otherwise} \end{cases}$$

| $\overline{y}$ | $\gamma(\overline{y})$ |
|---|---|
| $\bot$ | $\emptyset$ |
| $-$ | $\{r \in \mathbb{R} \mid r < 0\}$ |
| $0$ | $\{0\}$ |
| $+$ | $\{r \in \mathbb{R} \mid r > 0\}$ |
| $-0$ | $\{r \in \mathbb{R} \mid r \leq 0\}$ |
| $0+$ | $\{r \in \mathbb{R} \mid r \geq 0\}$ |
| $\top$ | $\mathbb{R}$ |

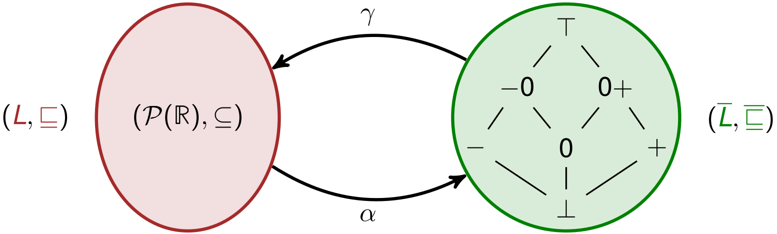
# Galois Connections: Example


$(L, \sqsubseteq)$
$(\mathcal{P}(\mathbb{R}), \subseteq)$
$\gamma$
$\alpha$
$\top$
$-0$
$0+$
$-$
$0$
$+$
$\bot$
$(\overline{L}, \overline{\sqsubseteq})$

$$\alpha(x) = \begin{cases} \bot & \text{if } x = \emptyset \\ - & \text{if } x \subseteq \{r \in \mathbb{R} \mid r < 0\} \\ 0 & \text{if } x = \{0\} \\ + & \text{if } x \subseteq \{r \in \mathbb{R} \mid r > 0\} \\ -0 & \text{if } \{0\} \subset x \subseteq \{r \in \mathbb{R} \mid r \leq 0\} \\ 0+ & \text{if } \{0\} \subset x \subseteq \{r \in \mathbb{R} \mid r \geq 0\} \\ \top & \text{otherwise} \end{cases}$$

| $\overline{y}$ | $\gamma(\overline{y})$ |
|---|---|
| $\bot$ | $\emptyset$ |
| $-$ | $\{r \in \mathbb{R} \mid r < 0\}$ |
| $0$ | $\{0\}$ |
| $+$ | $\{r \in \mathbb{R} \mid r > 0\}$ |
| $-0$ | $\{r \in \mathbb{R} \mid r \leq 0\}$ |
| $0+$ | $\{r \in \mathbb{R} \mid r \geq 0\}$ |
| $\top$ | $\mathbb{R}$ |

# Galois Connections: Example


$\gamma$
$(L, \sqsubseteq)$
$(\mathcal{P}(\mathbb{R}), \subseteq)$
$\alpha$
$\top$
$-0$
$0+$
$-$
$0$
$+$
$\bot$
$(\overline{L}, \overline{\sqsubseteq})$

$$\alpha(x) = \begin{cases} \bot & \text{if } x = \emptyset \\ - & \text{if } x \subseteq \{r \in \mathbb{R} \mid r < 0\} \\ 0 & \text{if } x = \{0\} \\ + & \text{if } x \subseteq \{r \in \mathbb{R} \mid r > 0\} \\ -0 & \text{if } \{0\} \subset x \subseteq \{r \in \mathbb{R} \mid r \leq 0\} \\ 0+ & \text{if } \{0\} \subset x \subseteq \{r \in \mathbb{R} \mid r \geq 0\} \\ \top & \text{otherwise} \end{cases}$$

| $\overline{y}$ | $\gamma(\overline{y})$ |
|---|---|
| $\bot$ | $\emptyset$ |
| $-$ | $\{r \in \mathbb{R} \mid r < 0\}$ |
| $0$ | $\{0\}$ |
| $+$ | $\{r \in \mathbb{R} \mid r > 0\}$ |
| $-0$ | $\{r \in \mathbb{R} \mid r \leq 0\}$ |
| $0+$ | $\{r \in \mathbb{R} \mid r \geq 0\}$ |
| $\top$ | $\mathbb{R}$ |

# Galois Connections: Characterization

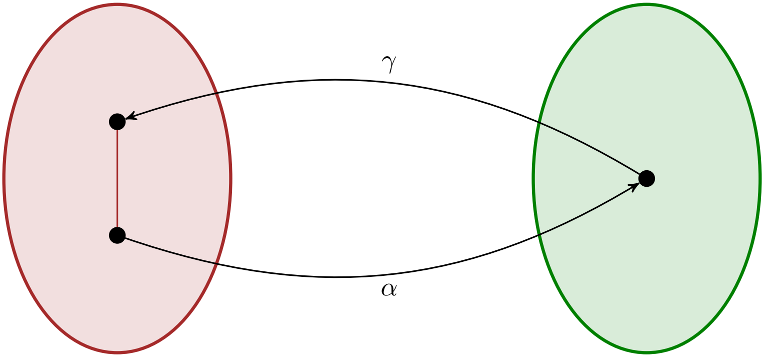
Consider two lattices $(L, \sqsubseteq)$ and $(\overline{L}, \overline{\sqsubseteq})$.

For any two functions $\alpha : L \rightarrow \overline{L}$ et $\gamma : \overline{L} \rightarrow L$, we have

$$(L, \sqsubseteq) \overset{\gamma}{\underset{\alpha}{\leftrightarrows}} (\overline{L}, \overline{\sqsubseteq}) \qquad \text{iff} \qquad \begin{cases} x \sqsubseteq \gamma \circ \alpha(x) & (\text{for all } x \in L) \\ \alpha \circ \gamma(\overline{y}) \mathrel{\overline{\sqsubseteq}} \overline{y} & (\text{for all } \overline{y} \in \overline{L}) \\ \alpha \text{ is monotonic} & \\ \gamma \text{ is monotonic} & \end{cases}$$

# Galois Connections: Characterization

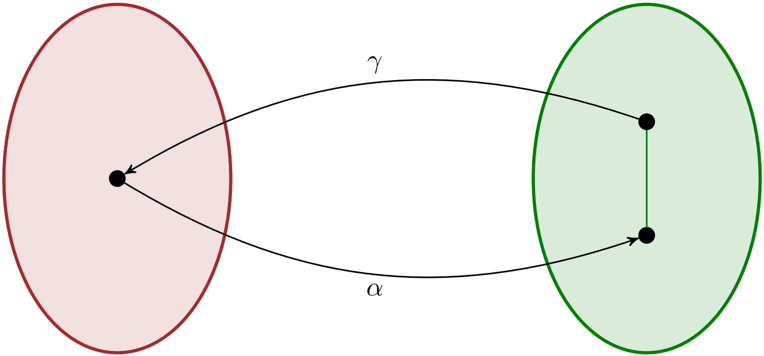
$$\begin{tikzcd}(L, \sqsubseteq) \arrow[r, shift right, "\alpha"'] & (\overline{L}, \overline{\sqsubseteq}) \arrow[l, shift right, "\gamma"']\end{tikzcd}$$



$x \sqsubseteq \gamma \circ \alpha(x)$ $(\gamma \circ \alpha$ extensive$)$

# Galois Connections: Characterization

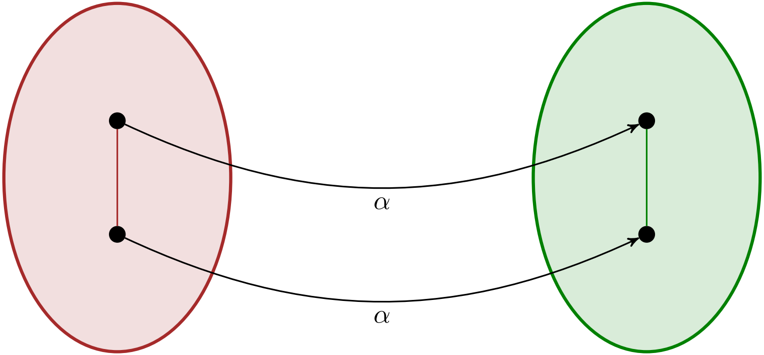
$$\begin{tikzcd}(L, \sqsubseteq) \arrow[r, shift right, "\alpha"'] & (\overline{L}, \overline{\sqsubseteq}) \arrow[l, shift right, "\gamma"']\end{tikzcd}$$



$\alpha \circ \gamma(\overline{y}) \ \overline{\sqsubseteq} \ \overline{y}$ ($\alpha \circ \gamma$ reductive)

# Galois Connections: Characterization

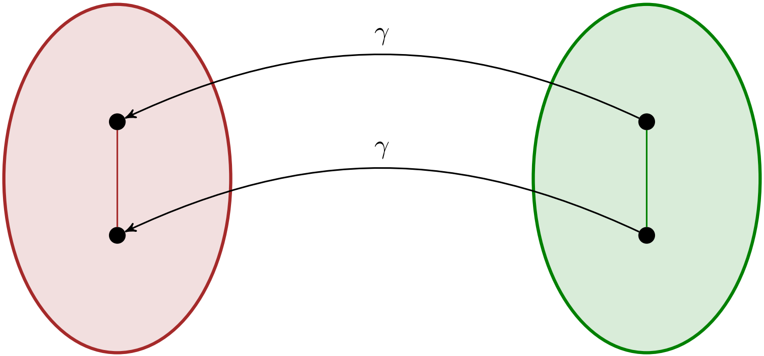
$$\begin{tikzcd}(L, \sqsubseteq) \arrow[r, shift right, "\alpha"'] & (\overline{L}, \overline{\sqsubseteq}) \arrow[l, shift right, "\gamma"']\end{tikzcd}$$



$\alpha$ is monotonic

# Galois Connections: Characterization

$$\begin{tikzcd}(L, \sqsubseteq) \arrow[r, shift right, "\alpha"'] & (\overline{L}, \overline{\sqsubseteq}) \arrow[l, shift right, "\gamma"']\end{tikzcd}$$



$\gamma$ is monotonic

# Galois Connections: Properties

For any Galois connection $(L, \sqsubseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (\overline{L}, \overline{\sqsubseteq})$, we have

$$\alpha = \alpha \circ \gamma \circ \alpha \qquad \gamma = \gamma \circ \alpha \circ \gamma$$

$\alpha$ is surjective iff $\gamma$ is injective iff $\alpha \circ \gamma = \lambda \overline{y} \,.\, \overline{y}$

## Definition

A Galois insertion between a lattice $(L, \sqsubseteq)$ and a lattice $(\overline{L}, \overline{\sqsubseteq})$ is any Galois connection $(L, \sqsubseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (\overline{L}, \overline{\sqsubseteq})$ where $\alpha$ is surjective.

Notation for Galois insertions: $(L, \sqsubseteq) \underset{\alpha}{\overset{\gamma}{\leftarrows\!\!\twoheadrightarrow}} (\overline{L}, \overline{\sqsubseteq})$

# Galois Connections: Properties

For any Galois connection $(L, \sqsubseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (\overline{L}, \overline{\sqsubseteq})$, we have

$$\alpha \;=\; \alpha \circ \gamma \circ \alpha \qquad\qquad \gamma \;=\; \gamma \circ \alpha \circ \gamma$$

$\alpha$ is surjective iff $\gamma$ is injective iff $\alpha \circ \gamma = \lambda\, \overline{y} \,.\, \overline{y}$

## Definition

A Galois insertion between a lattice $(L, \sqsubseteq)$ and a lattice $(\overline{L}, \overline{\sqsubseteq})$ is any Galois connection $(L, \sqsubseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (\overline{L}, \overline{\sqsubseteq})$ where $\alpha$ is surjective.

Notation for Galois insertions: $(L, \sqsubseteq) \underset{\alpha}{\overset{\gamma}{\leftarrows\!\!\twoheadrightarrow}} (\overline{L}, \overline{\sqsubseteq})$

# Galois Connections: Properties

For any Galois connection $(L, \sqsubseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (\overline{L}, \overline{\sqsubseteq})$, we have

$$\alpha \;=\; \alpha \circ \gamma \circ \alpha \qquad\qquad \gamma \;=\; \gamma \circ \alpha \circ \gamma$$

$\alpha$ is surjective iff $\gamma$ is injective iff $\alpha \circ \gamma = \lambda \overline{y} \,.\, \overline{y}$

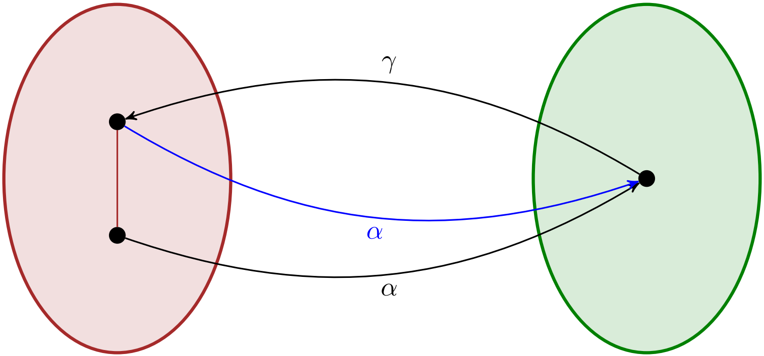
## Definition

A Galois insertion between a lattice $(L, \sqsubseteq)$ and a lattice $(\overline{L}, \overline{\sqsubseteq})$ is any Galois connection $(L, \sqsubseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (\overline{L}, \overline{\sqsubseteq})$ where $\alpha$ is surjective.

Notation for Galois insertions: $(L, \sqsubseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (\overline{L}, \overline{\sqsubseteq})$ (with a two-headed arrow $\twoheadrightarrow$ for $\alpha$)

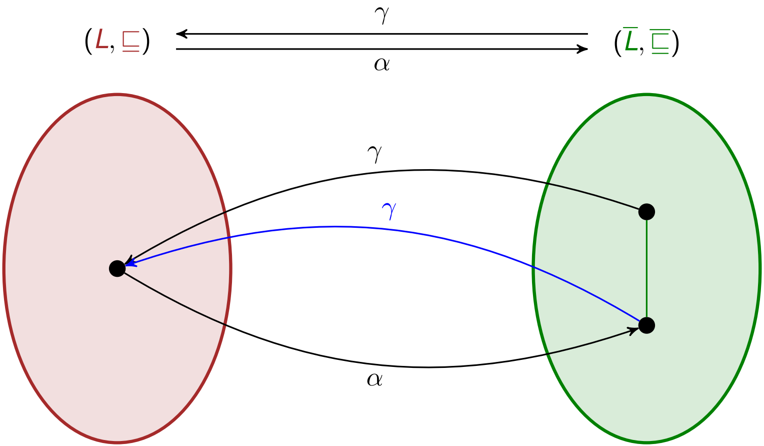
# Galois Connections: Properties

$$\begin{tikzcd}(L, \sqsubseteq) \arrow[r, shift right, "\alpha"'] & (\overline{L}, \overline{\sqsubseteq}) \arrow[l, shift right, "\gamma"']\end{tikzcd}$$



$$\alpha \;=\; \alpha \circ \gamma \circ \alpha$$

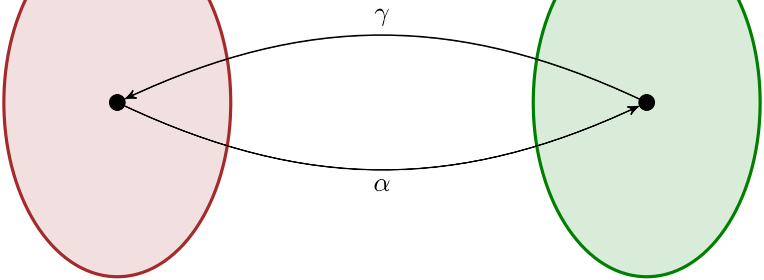
# Galois Connections: Properties



$$\gamma \;=\; \gamma \circ \alpha \circ \gamma$$

# Galois Connections: Properties

$$\begin{tikzcd}(L, \sqsubseteq) \arrow[r, two heads, shift right, "\alpha"'] & (\overline{L}, \overline{\sqsubseteq}) \arrow[l, shift right, "\gamma"']\end{tikzcd}$$



$$\alpha \circ \gamma = \lambda \overline{y} \,.\, \overline{y}$$ (Galois insertion)

# Galois Connections: Properties for Complete Lattices

For any Galois connection $(L, \sqsubseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (\overline{L}, \overline{\sqsubseteq})$ on complete lattices,

$$\begin{aligned} \alpha(x) &= \overline{\bigsqcap} \{\overline{y} \in \overline{L} \mid x \sqsubseteq \gamma(\overline{y})\} && \text{(for all } x \in L\text{)} \\ \gamma(\overline{y}) &= \bigsqcup \{x \in L \mid \alpha(x) \overline{\sqsubseteq} \overline{y}\} && \text{(for all } \overline{y} \in \overline{L}\text{)} \end{aligned}$$

$$\begin{aligned} \alpha\left(\bigsqcup X\right) &= \overline{\bigsqcup} \{\alpha(x) \mid x \in X\} && \text{(for all } X \subseteq L\text{)} \\ \gamma\left(\overline{\bigsqcap} \overline{Y}\right) &= \bigsqcap \{\gamma(\overline{y}) \mid \overline{y} \in \overline{Y}\} && \text{(for all } \overline{Y} \subseteq \overline{L}\text{)} \end{aligned}$$

## Informally

$\alpha$ uniquely determines $\gamma$ and $\gamma$ uniquely determines $\alpha$.

$\alpha$ preserves least upper bounds, $\gamma$ preserves greatest lower bounds.

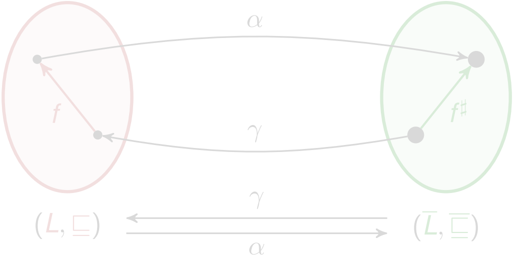
# Best Abstraction of a Monotonic Concrete Function

Consider a Galois connection $(L, \sqsubseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (\overline{L}, \overline{\sqsubseteq})$ on complete lattices.

## Definition

For any *monotonic* function $f : L \rightarrow L$, the best abstraction of $f$ is the *monotonic* function $f^\sharp : \overline{L} \rightarrow \overline{L}$ defined by:

$$f^\sharp \quad = \quad \alpha \circ f \circ \gamma$$



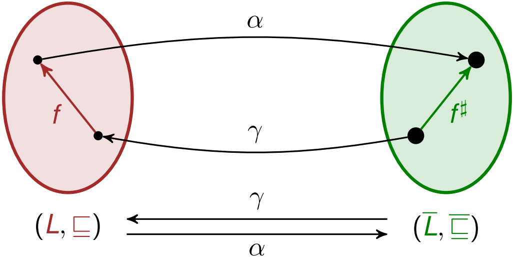
# Best Abstraction of a Monotonic Concrete Function

Consider a Galois connection $(L, \sqsubseteq) \overset{\gamma}{\underset{\alpha}{\leftrightarrows}} (\overline{L}, \overline{\sqsubseteq})$ on complete lattices.

## Definition

For any *monotonic* function $f : L \rightarrow L$, the best abstraction of $f$ is the *monotonic* function $f^\sharp : \overline{L} \rightarrow \overline{L}$ defined by:

$$f^\sharp \quad = \quad \alpha \circ f \circ \gamma$$



# Best Abstraction: Justification

Given a monotonic function $f : L \rightarrow L$, we look for a monotonic function $\overline{g} : \overline{L} \rightarrow \overline{L}$ that is a sound approximation of $f$:

$$f(x) \sqsubseteq \gamma \circ \overline{g} \circ \alpha(x)$$

or equivalently (when $\overline{g}$ is monotonic):

$$\alpha(x) \mathrel{\overline{\sqsubseteq}} \overline{y} \quad \Longrightarrow \quad \overline{g}(\overline{y}) \sqsupseteq \alpha \circ f(x)$$

The most precise function satisfying the above condition is defined by:

$$\overline{g}(\overline{y}) \quad = \quad \overline{\bigsqcup} \{\alpha \circ f(x) \mid \alpha(x) \mathrel{\overline{\sqsubseteq}} \overline{y}\}$$

# Best Abstraction: Justification

Given a monotonic function $f : L \to L$, we look for a monotonic function $\overline{g} : \overline{L} \to \overline{L}$ that is a sound approximation of $f$:

$$f(x) \sqsubseteq \gamma \circ \overline{g} \circ \alpha(x)$$

or equivalently (when $\overline{g}$ is monotonic):

$$\alpha(x) \mathrel{\overline{\sqsubseteq}} \overline{y} \quad \Longrightarrow \quad \overline{g}(\overline{y}) \mathrel{\overline{\sqsupseteq}} \alpha \circ f(x)$$

The most precise function satisfying the above condition is defined by:

$$\overline{g}(\overline{y}) \quad = \quad \overline{\bigsqcup} \left\{\alpha \circ f(x) \mid \alpha(x) \mathrel{\overline{\sqsubseteq}} \overline{y}\right\}$$

# Best Abstraction: Justification

$$\overline{g}(\overline{y}) \quad = \quad \overline{\bigsqcup} \{\alpha \circ f(x) \mid \alpha(x) \sqsubseteq \overline{y}\}$$

Recall that $\alpha$ preserves least upper bounds, hence:

$$\overline{g}(\overline{y}) \quad = \quad \alpha\left(\bigsqcup \{f(x) \mid x \in X\}\right)$$

where $X = \{x \in L \mid \alpha(x) \sqsubseteq \overline{y}\}$. Since $f$ is monotonic,

$$\bigsqcup \{f(x) \mid x \in X\} \quad \sqsubseteq \quad f\left(\bigsqcup X\right)$$

Recall that $\alpha \circ \gamma(\overline{y}) \sqsubseteq \overline{y}$, hence $\gamma(\overline{y}) \in X$ and we come to:

$$\bigsqcup \{f(x) \mid x \in X\} \quad \sqsupseteq \quad f(\gamma(\overline{y}))$$

# Best Abstraction: Justification

$$\overline{g}(\overline{y}) \quad = \quad \overline{\bigsqcup} \{\alpha \circ f(x) \mid \alpha(x) \sqsubseteq \overline{y}\}$$

Recall that $\alpha$ preserves least upper bounds, hence:

$$\overline{g}(\overline{y}) \quad = \quad \alpha\left(\bigsqcup \{f(x) \mid x \in X\}\right)$$

where $X = \left\{x \in L \;\middle|\; \alpha(x) \sqsubseteq \overline{y}\right\}$. Since $f$ is monotonic,

$$\bigsqcup \{f(x) \mid x \in X\} \quad \sqsubseteq \quad f\left(\bigsqcup X\right)$$

Recall that $\alpha \circ \gamma(\overline{y}) \sqsubseteq \overline{y}$, hence $\gamma(\overline{y}) \in X$ and we come to:

$$\bigsqcup \{f(x) \mid x \in X\} \quad \sqsupseteq \quad f(\gamma(\overline{y}))$$

# Best Abstraction: Justification

$$\overline{g}(\overline{y}) \quad = \quad \overline{\bigsqcup} \{\alpha \circ f(x) \mid \alpha(x) \overline{\sqsubseteq} \overline{y}\}$$

Recall that $\alpha$ preserves least upper bounds, hence:

$$\overline{g}(\overline{y}) \quad = \quad \alpha\left(\bigsqcup \{f(x) \mid x \in X\}\right)$$

where $X = \left\{x \in L \;\middle|\; \alpha(x) \overline{\sqsubseteq} \overline{y}\right\}$. Since $f$ is monotonic,

$$\bigsqcup \{f(x) \mid x \in X\} \quad \sqsubseteq \quad f\left(\bigsqcup X\right)$$

Recall that $\alpha \circ \gamma(\overline{y}) \overline{\sqsubseteq} \overline{y}$, hence $\gamma(\overline{y}) \in X$ and we come to:

$$\bigsqcup \{f(x) \mid x \in X\} \quad \sqsupseteq \quad f(\gamma(\overline{y}))$$

# Best Abstraction: Justification

$$\overline{g}(\overline{y}) \quad = \quad \overline{\bigsqcup} \{\alpha \circ f(x) \mid \alpha(x) \overline{\sqsubseteq} \overline{y}\}$$

Recall that $\alpha$ preserves least upper bounds, hence:

$$\overline{g}(\overline{y}) \quad = \quad \alpha\left(\bigsqcup \{f(x) \mid x \in X\}\right)$$

where $X = \{x \in L \mid \alpha(x) \overline{\sqsubseteq} \overline{y}\}$. Since $f$ is monotonic,

$$\bigsqcup \{f(x) \mid x \in X\} \quad \sqsubseteq \quad f\left(\bigsqcup X\right)$$

Recall that $\alpha \circ \gamma(\overline{y}) \overline{\sqsubseteq} \overline{y}$, hence $\gamma(\overline{y}) \in X$ and we come to:

$$\bigsqcup \{f(x) \mid x \in X\} \quad \sqsupseteq \quad f(\gamma(\overline{y}))$$

# Best Abstraction: Justification

$$\overline{g}(\overline{y}) \quad = \quad \alpha\left(\bigsqcup \{f(x) \mid x \in X\}\right)$$

where $X = \left\{x \in L \;\middle|\; \alpha(x) \sqsubseteq \overline{y}\right\}$.

$$f(\gamma(\overline{y})) \quad \sqsubseteq \quad \bigsqcup \{f(x) \mid x \in X\} \quad \sqsubseteq \quad f\left(\bigsqcup X\right)$$

Recall that $\gamma(\overline{y}) = \bigsqcup X$, hence:

$$\bigsqcup \{f(x) \mid x \in X\} \quad = \quad f\left(\bigsqcup X\right) \quad = \quad f(\gamma(\overline{y}))$$

We obtain that:

$$\overline{g}(\overline{y}) \quad = \quad \alpha \circ f \circ \gamma(\overline{y})$$

And... all is well, since $\alpha \circ f \circ \gamma$ is monotonic!

# Best Abstraction: Justification

$$\overline{g}(\overline{y}) \quad = \quad \alpha\left(\bigsqcup \{f(x) \mid x \in X\}\right)$$

where $X = \left\{x \in L \;\middle|\; \alpha(x) \mathrel{\overline{\sqsubseteq}} \overline{y}\right\}$.

$$f(\gamma(\overline{y})) \quad \sqsubseteq \quad \bigsqcup \{f(x) \mid x \in X\} \quad \sqsubseteq \quad f\left(\bigsqcup X\right)$$

Recall that $\gamma(\overline{y}) = \bigsqcup X$, hence:

$$\bigsqcup \{f(x) \mid x \in X\} \quad = \quad f\left(\bigsqcup X\right) \quad = \quad f(\gamma(\overline{y}))$$

We obtain that:

$$\overline{g}(\overline{y}) \quad = \quad \alpha \circ f \circ \gamma(\overline{y})$$

And... all is well, since $\alpha \circ f \circ \gamma$ is monotonic!

# Best Abstraction: Justification

$$\overline{g}(\overline{y}) \quad = \quad \alpha\left(\bigsqcup \{f(x) \mid x \in X\}\right)$$

where $X = \left\{x \in L \;\middle|\; \alpha(x) \mathrel{\overline{\sqsubseteq}} \overline{y}\right\}$.

$$f(\gamma(\overline{y})) \quad \sqsubseteq \quad \bigsqcup \{f(x) \mid x \in X\} \quad \sqsubseteq \quad f\left(\bigsqcup X\right)$$

Recall that $\gamma(\overline{y}) = \bigsqcup X$, hence:

$$\bigsqcup \{f(x) \mid x \in X\} \quad = \quad f\left(\bigsqcup X\right) \quad = \quad f(\gamma(\overline{y}))$$

We obtain that:

$$\overline{g}(\overline{y}) \quad = \quad \alpha \circ f \circ \gamma(\overline{y})$$

And... all is well, since $\alpha \circ f \circ \gamma$ is monotonic!

# Galois Connections: Fixpoint Abstraction

Consider a Galois connection $(L, \sqsubseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (\overline{L}, \overline{\sqsubseteq})$ on complete lattices.

Recall that for any monotonic function $f : L \rightarrow L$, we denote by $f^\sharp$ the monotonic function:

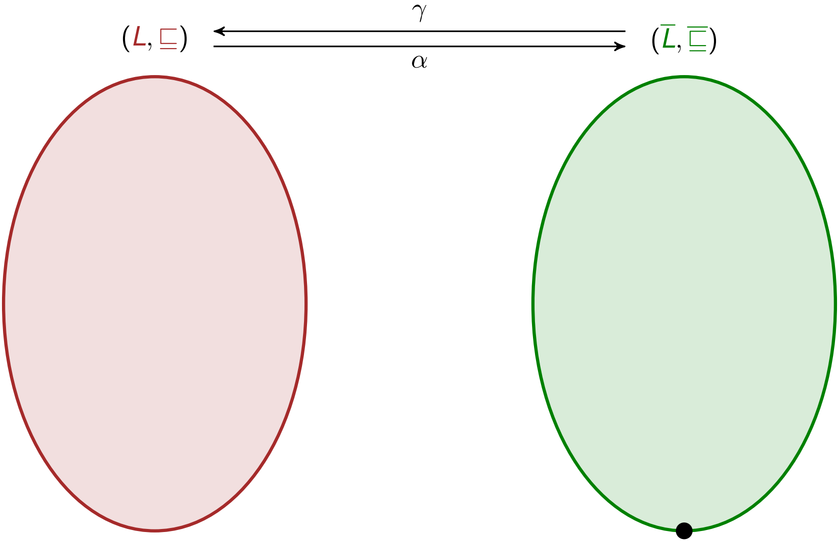
$$f^\sharp \quad = \quad \alpha \circ f \circ \gamma$$

## Theorem

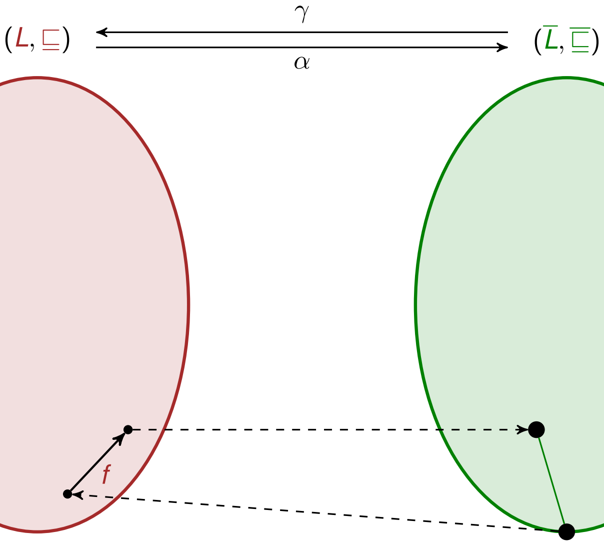
*For any monotonic function $f : L \rightarrow L$, the least fixpoints of $f$ and $f^\sharp$ satisfy:*

$$\mathrm{lfp}(f) \quad \sqsubseteq \quad \gamma\left(\mathrm{lfp}(f^\sharp)\right)$$

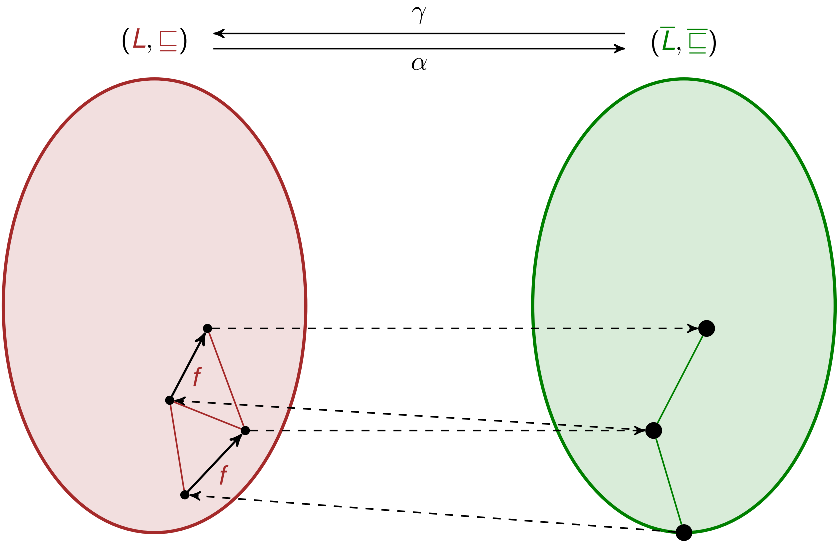
# Galois Connections: Fixpoint Abstraction



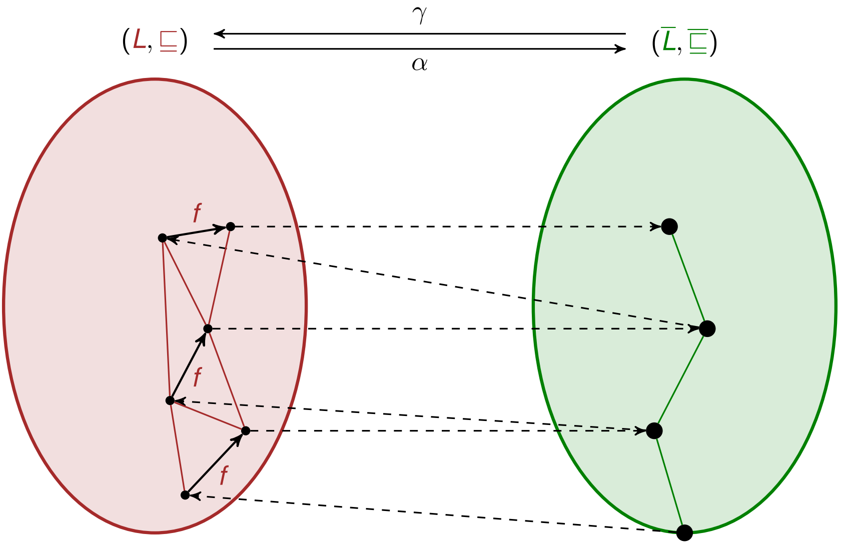
# Galois Connections: Fixpoint Abstraction



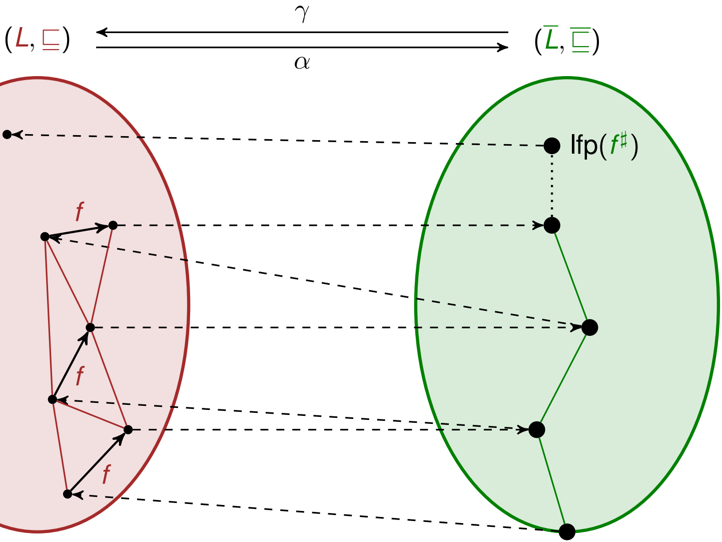
# Galois Connections: Fixpoint Abstraction



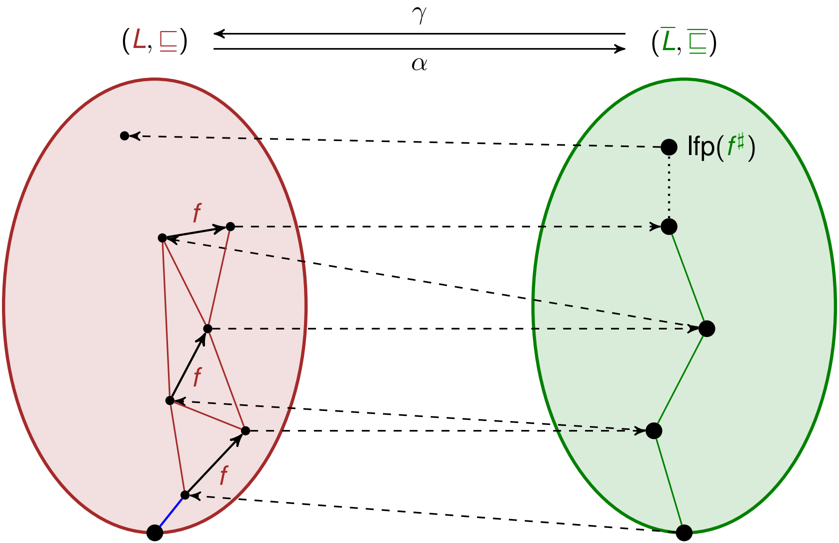
# Galois Connections: Fixpoint Abstraction



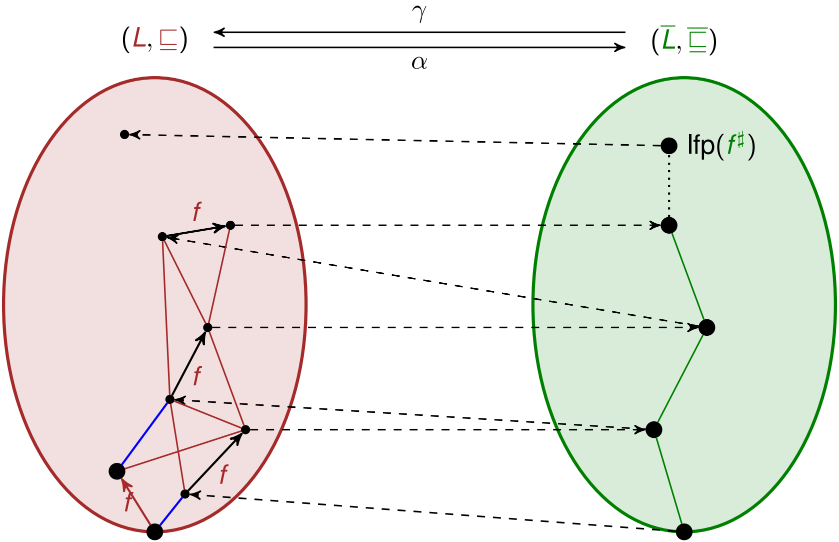
# Galois Connections: Fixpoint Abstraction



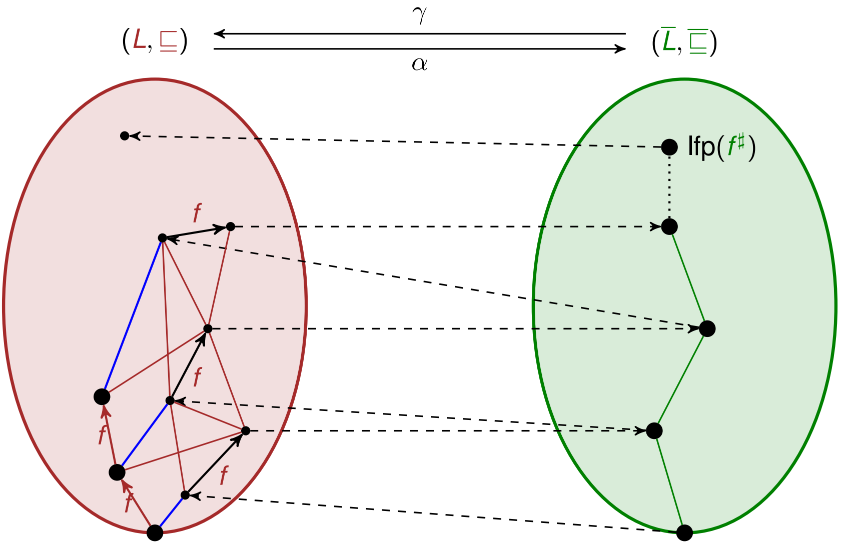
# Galois Connections: Fixpoint Abstraction



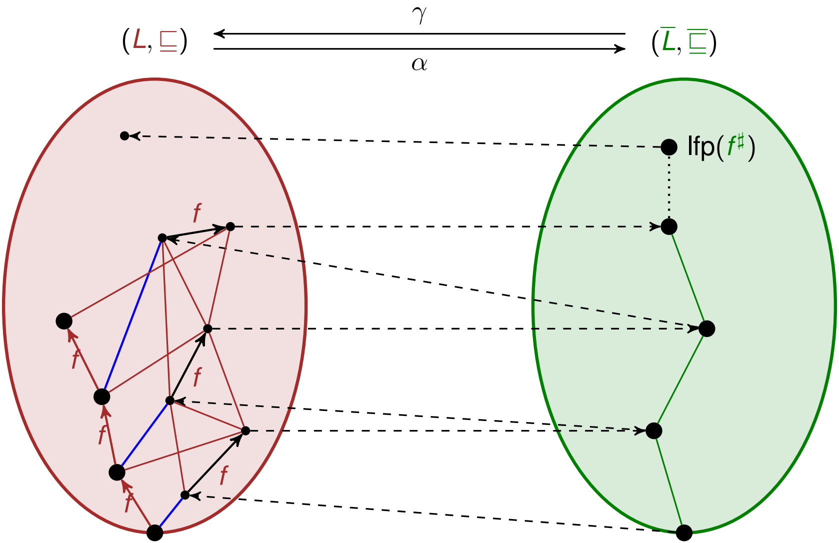
# Galois Connections: Fixpoint Abstraction



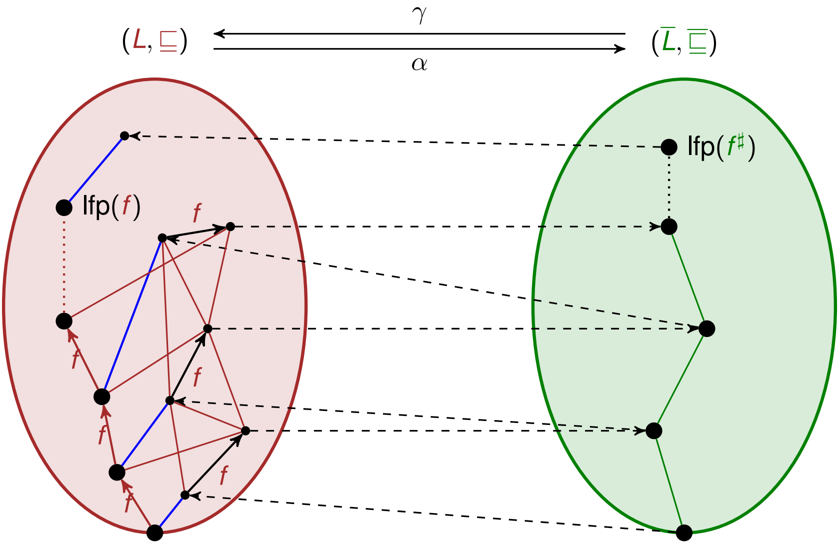
# Galois Connections: Fixpoint Abstraction



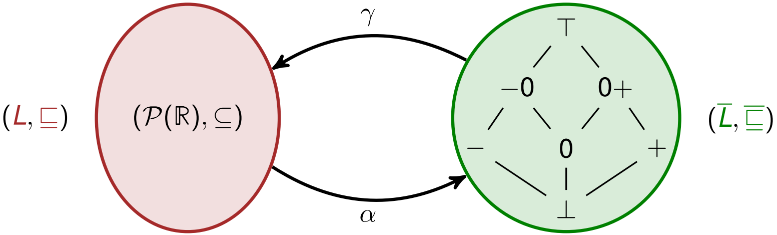
# Galois Connections: Fixpoint Abstraction



# Galois Connections: Fixpoint Abstraction



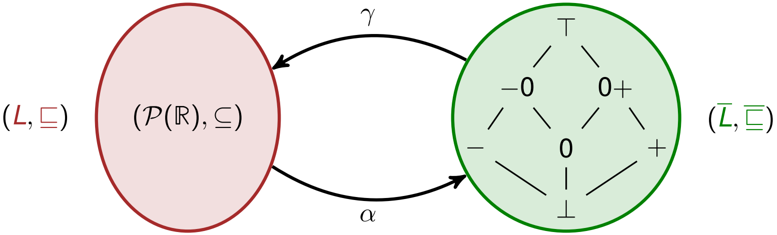
# Best Abstraction & Fixpoint Abstraction: Example


$(L, \sqsubseteq)$
$(\mathcal{P}(\mathbb{R}), \subseteq)$
$\gamma$
$\alpha$
$\top$
$-0$
$0+$
$-$
$0$
$+$
$\bot$
$(\overline{L}, \overline{\sqsubseteq})$

| $f$ | | |
|---|---|---|
| $\mathcal{P}(\mathbb{R})$ | $\rightarrow$ | $\mathcal{P}(\mathbb{R})$ |
| $x$ | $\mapsto$ | $\{r+2 \mid r \in x\} \cup \{5\}$ |

| $f^\sharp$ | | | | | | | |
|---|---|---|---|---|---|---|---|
| $\overline{y}$ | $\bot$ | $-$ | $0$ | $+$ | $-0$ | $0+$ | $\top$ |
| $f^\sharp(\overline{y})$ | | | | | | | |

# Best Abstraction & Fixpoint Abstraction: Example



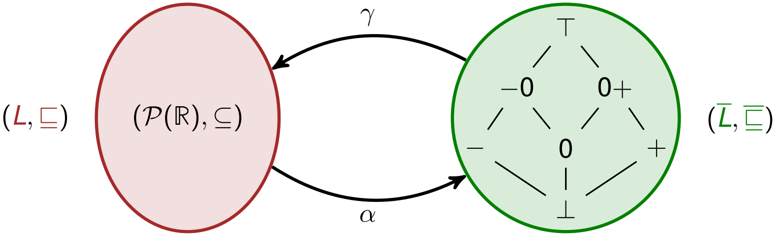
$(L, \sqsubseteq)$
$(\mathcal{P}(\mathbb{R}), \subseteq)$
$\gamma$
$\alpha$
$\top$
$-0$
$0+$
$-$
$0$
$+$
$\bot$
$(\overline{L}, \overline{\sqsubseteq})$

| $f$ | | |
|---|---|---|
| $\mathcal{P}(\mathbb{R})$ | $\rightarrow$ | $\mathcal{P}(\mathbb{R})$ |
| $x$ | $\mapsto$ | $\{r + 2 \mid r \in x\} \cup \{5\}$ |

| $f^\sharp$ | | | | | | | |
|---|---|---|---|---|---|---|---|
| $\overline{y}$ | $\bot$ | $-$ | $0$ | $+$ | $-0$ | $0+$ | $\top$ |
| $f^\sharp(\overline{y})$ | $+$ | | | | | | |

$$\begin{aligned} f^\sharp(\bot) &= \alpha \circ f \circ \gamma(\bot) \\ &= \alpha \circ f(\emptyset) \\ &= \alpha(\{5\}) \\ &= + \end{aligned}$$

# Best Abstraction & Fixpoint Abstraction: Example

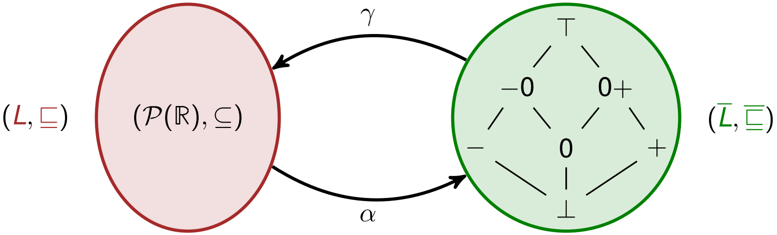

$(L, \sqsubseteq)$
$(\mathcal{P}(\mathbb{R}), \subseteq)$
$\gamma$
$\alpha$
$\top$
$-0$
$0+$
$-$
$0$
$+$
$\bot$
$(\overline{L}, \overline{\sqsubseteq})$

| $f$ | | |
|---|---|---|
| $\mathcal{P}(\mathbb{R})$ | $\rightarrow$ | $\mathcal{P}(\mathbb{R})$ |
| $x$ | $\mapsto$ | $\{r+2 \mid r \in x\} \cup \{5\}$ |

| $f^\sharp$ | | | | | | | |
|---|---|---|---|---|---|---|---|
| $\overline{y}$ | $\bot$ | $-$ | $0$ | $+$ | $-0$ | $0+$ | $\top$ |
| $f^\sharp(\overline{y})$ | $+$ | $\top$ | | | | | |

$$
\begin{aligned}
f^\sharp(-) &= \alpha \circ f \circ \gamma(-) \\
&= \alpha \circ f(\{r \in \mathbb{R} \mid r < 0\}) \\
&= \alpha(\{r+2 \mid r < 0\} \cup \{5\}) \\
&= \top
\end{aligned}
$$

# Best Abstraction & Fixpoint Abstraction: Example


$(L, \sqsubseteq)$
$(\mathcal{P}(\mathbb{R}), \subseteq)$
$\gamma$
$\alpha$
$\top$
$-0$
$0+$
$-$
$0$
$+$
$\bot$
$(\overline{L}, \overline{\sqsubseteq})$

**$f$**

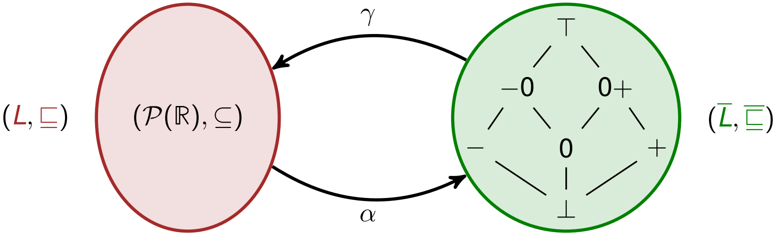
$$\begin{array}{rcl} \mathcal{P}(\mathbb{R}) & \rightarrow & \mathcal{P}(\mathbb{R}) \\ x & \mapsto & \{r+2 \mid r \in x\} \cup \{5\} \end{array}$$

**$f^\sharp$**

| $\overline{y}$ | $\bot$ | $-$ | $0$ | $+$ | $-0$ | $0+$ | $\top$ |
|---|---|---|---|---|---|---|---|
| $f^\sharp(\overline{y})$ | $+$ | $\top$ | $+$ | | | | |

$$\begin{aligned} f^\sharp(0) &= \alpha \circ f \circ \gamma(0) \\ &= \alpha \circ f(\{0\}) \\ &= \alpha(\{2\} \cup \{5\}) \\ &= + \end{aligned}$$

# Best Abstraction & Fixpoint Abstraction: Example


$(L, \sqsubseteq)$
$(\mathcal{P}(\mathbb{R}), \subseteq)$
$\gamma$
$\alpha$
$\top$
$-0$
$0+$
$-$
$0$
$+$
$\bot$
$(\overline{L}, \overline{\sqsubseteq})$

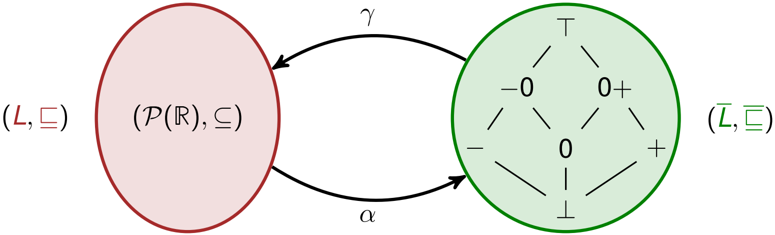
$f$

$$\begin{array}{rcl} \mathcal{P}(\mathbb{R}) & \to & \mathcal{P}(\mathbb{R}) \\ x & \mapsto & \{r+2 \mid r \in x\} \cup \{5\} \end{array}$$

$f^\sharp$

| $\overline{y}$ | $\bot$ | $-$ | $0$ | $+$ | $-0$ | $0+$ | $\top$ |
|---|---|---|---|---|---|---|---|
| $f^\sharp(\overline{y})$ | $+$ | $\top$ | $+$ | $+$ | $\top$ | $+$ | $\top$ |

# Best Abstraction & Fixpoint Abstraction: Example



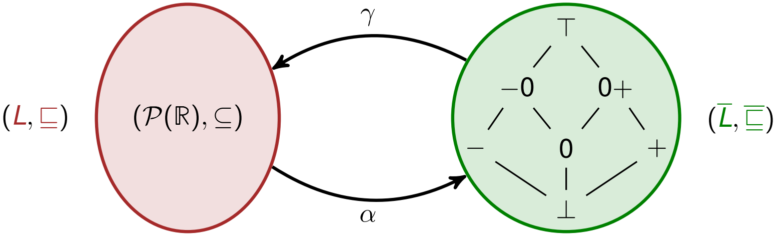
$(L, \sqsubseteq)$
$(\mathcal{P}(\mathbb{R}), \subseteq)$
$\gamma$
$\alpha$
$\top$
$-0$
$0+$
$-$
$0$
$+$
$\bot$
$(\overline{L}, \overline{\sqsubseteq})$

| $f$ | | |
|---|---|---|
| $\mathcal{P}(\mathbb{R})$ | $\rightarrow$ | $\mathcal{P}(\mathbb{R})$ |
| $x$ | $\mapsto$ | $\{r+2 \mid r \in x\} \cup \{5\}$ |

$$\begin{aligned} f(\emptyset) &= \{5\} \\ f^2(\emptyset) &= \{5,7\} \\ f^3(\emptyset) &= \{5,7,9\} \\ \operatorname{lfp} f &= \{5+2k \mid k \in \mathbb{N}\} \end{aligned}$$

| $f^\sharp$ | | | | | | | |
|---|---|---|---|---|---|---|---|
| $\overline{y}$ | $\bot$ | $-$ | $0$ | $+$ | $-0$ | $0+$ | $\top$ |
| $f^\sharp(\overline{y})$ | $+$ | $\top$ | $+$ | $+$ | $\top$ | $+$ | $\top$ |

# Best Abstraction & Fixpoint Abstraction: Example


$(L, \sqsubseteq)$
$(\mathcal{P}(\mathbb{R}), \subseteq)$
$\gamma$
$\alpha$
$\top$
$-0$
$0+$
$-$
$0$
$+$
$\bot$
$(\overline{L}, \overline{\sqsubseteq})$

$f$

$$\begin{array}{rcl} \mathcal{P}(\mathbb{R}) & \to & \mathcal{P}(\mathbb{R}) \\ x & \mapsto & \{r+2 \mid r \in x\} \cup \{5\} \end{array}$$

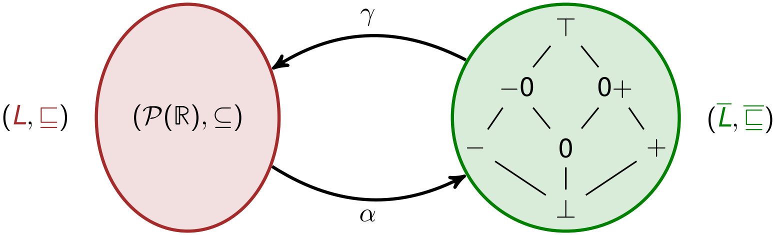
$$\begin{array}{rcl} f(\emptyset) & = & \{5\} \\ f^2(\emptyset) & = & \{5, 7\} \\ f^3(\emptyset) & = & \{5, 7, 9\} \\ \mathrm{lfp}\ f & = & \{5 + 2\,k \mid k \in \mathbb{N}\} \end{array}$$

$f^\sharp$

| $\overline{y}$ | $\bot$ | $-$ | $0$ | $+$ | $-0$ | $0+$ | $\top$ |
|---|---|---|---|---|---|---|---|
| $f^\sharp(\overline{y})$ | $+$ | $\top$ | $+$ | $+$ | $\top$ | $+$ | $\top$ |

$$\begin{array}{rcl} f^\sharp(\bot) & = & + \\ f^{\sharp 2}(\bot) & = & + \\ \mathrm{lfp}\ f^\sharp & = & + \end{array}$$

# Best Abstraction & Fixpoint Abstraction: Example



$(L, \sqsubseteq)$
$(\mathcal{P}(\mathbb{R}), \subseteq)$
$\gamma$
$\alpha$
$\top$
$-0$
$0+$
$-$
$0$
$+$
$\bot$
$(\overline{L}, \overline{\sqsubseteq})$

| $f$ | | |
|---|---|---|
| $\mathcal{P}(\mathbb{R})$ | $\rightarrow$ | $\mathcal{P}(\mathbb{R})$ |
| $x$ | $\mapsto$ | $\{r + 2 \mid r \in x\} \cup \{5\}$ |

$$\begin{aligned} f(\emptyset) &= \{5\} \\ f^2(\emptyset) &= \{5, 7\} \\ f^3(\emptyset) &= \{5, 7, 9\} \\ \text{lfp } f &= \{5 + 2\,k \mid k \in \mathbb{N}\} \end{aligned}$$

| $f^\sharp$ | | | | | | | |
|---|---|---|---|---|---|---|---|
| $\overline{y}$ | $\bot$ | $-$ | $0$ | $+$ | $-0$ | $0+$ | $\top$ |
| $f^\sharp(\overline{y})$ | $+$ | $\top$ | $+$ | $+$ | $\top$ | $+$ | $\top$ |

$$\text{lfp } f^\sharp = +$$

$$\gamma\left(\text{lfp}(f^\sharp)\right) = \{r \in \mathbb{R} \mid r > 0\}$$

# Galois Connections: Summary & Application

We want to "compute" the least fixpoint $\text{lfp}(f)$ of monotonic function $f : L \to L$ on a complete lattice $(L, \sqsubseteq)$.

If Kleene iteration $\bot \sqsubseteq f(\bot) \sqsubseteq \cdots \sqsubseteq f^i(\bot) \sqsubseteq \cdots$ diverges then:

1. design an abstract complete lattice $(\overline{L}, \overline{\sqsubseteq})$, simpler than $(L, \sqsubseteq)$, and formalize the "meaning" of abstract values by a Galois connection

$$(L, \sqsubseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (\overline{L}, \overline{\sqsubseteq})$$

2. compute $\text{lfp}(f^\sharp)$, where $f^\sharp = \alpha \circ f \circ \gamma$ is the best abstraction of $f$.

By Fixpoint Abstraction Theorem, $\gamma\left(\text{lfp}(f^\sharp)\right)$ soundly approximates $\text{lfp}(f)$

$$\text{lfp}(f) \quad \sqsubseteq \quad \gamma\left(\text{lfp}(f^\sharp)\right) \qquad \text{or equivalently} \qquad \alpha(\text{lfp}(f)) \quad \overline{\sqsubseteq} \quad \text{lfp}(f^\sharp)$$

# Outline — Abstract Interpretation

8 Some More Lattice Theory: Galois Connections

9 Abstract Interpretation-Based Data Flow Analysis

- Design of Approximate Transfer Mappings for Sign Analysis

10 Convergence Acceleration with Widening and Narrowing

# Short Introduction to Abstract Interpretation

**Recall that to ensure correctness of data flow analyses. . .**

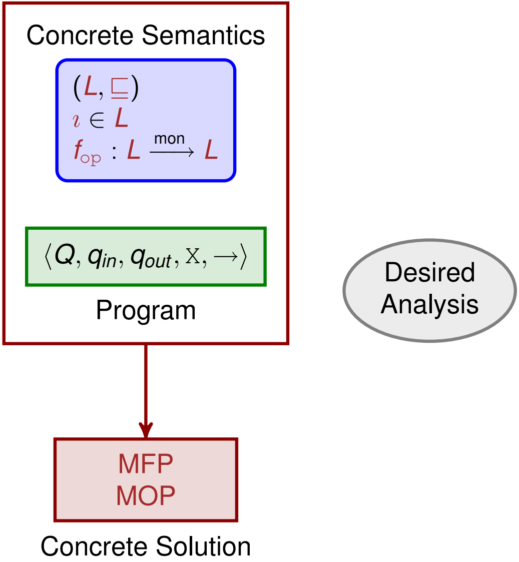
. . . we need a formal link to relate data flow facts and transfer functions with the formal semantics.

Abstract interpretation relies on Galois connections to formally express these relationships.

- Formal meaning of data flow facts by a concretization function
- Transfer mapping that soundly approximates the formal semantics
- Sound fixpoint approximation

Data flow analyses that are correct by design: crucial for verification!

# Systematic Design of Correct of Data Flow Analyses



Concrete Semantics

$(L, \sqsubseteq)$

$\imath \in L$

$f_{\mathrm{op}} : L \xrightarrow{\text{mon}} L$

$\langle Q, q_{in}, q_{out}, \mathrm{X}, \rightarrow \rangle$

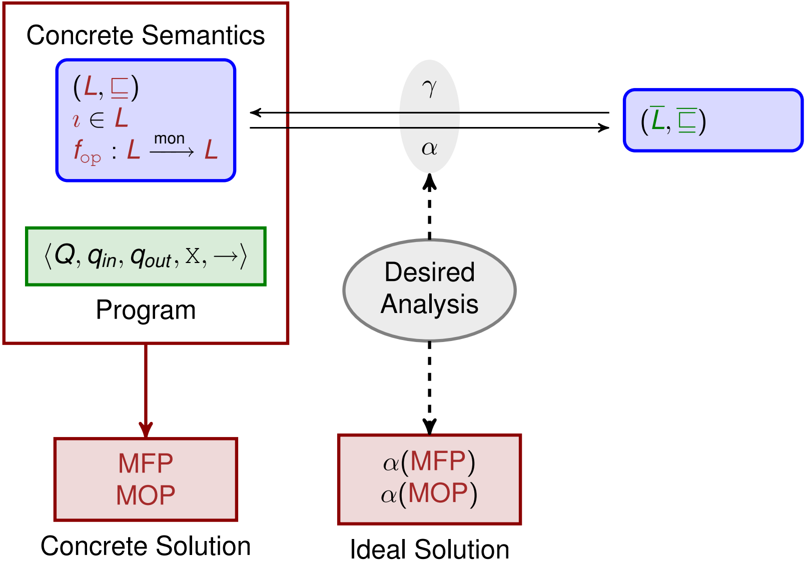
Program

Desired Analysis

MFP
MOP

Concrete Solution

# Systematic Design of Correct of Data Flow Analyses



Concrete Semantics

$(L, \sqsubseteq)$

$\imath \in L$

$f_{\text{op}} : L \xrightarrow{\text{mon}} L$

$\langle Q, q_{in}, q_{out}, \mathrm{X}, \longrightarrow \rangle$

Program

$\gamma$

$\alpha$

$(\overline{L}, \overline{\sqsubseteq})$

Desired Analysis

MFP

MOP

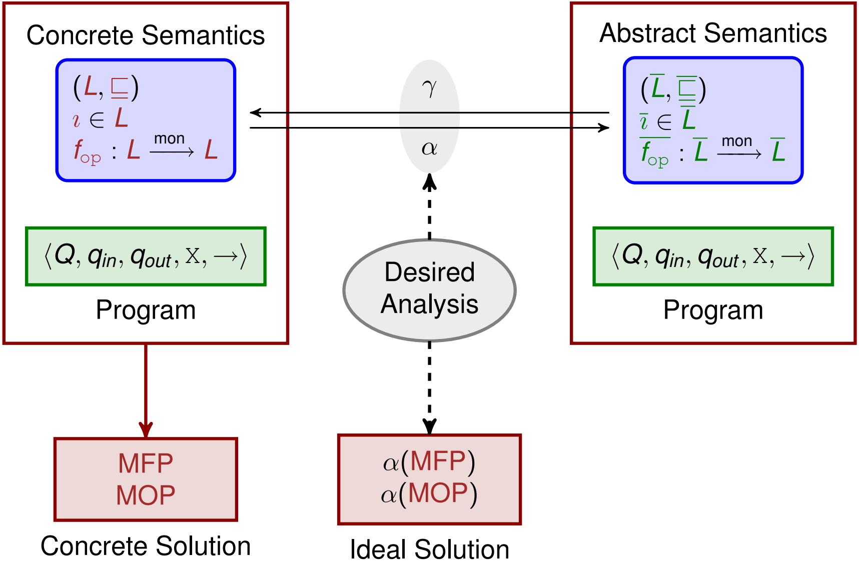
Concrete Solution

$\alpha(\text{MFP})$

$\alpha(\text{MOP})$

Ideal Solution

# Systematic Design of Correct of Data Flow Analyses



Concrete Semantics

$(L, \sqsubseteq)$

$\imath \in L$

$f_{\mathrm{op}} : L \xrightarrow{\text{mon}} L$

$\langle Q, q_{in}, q_{out}, \mathrm{X}, \rightarrow \rangle$

Program

$\gamma$

$\alpha$

Abstract Semantics

$(\overline{L}, \overline{\sqsubseteq})$

$\overline{\imath} \in \overline{L}$

$\overline{f_{\mathrm{op}}} : \overline{L} \xrightarrow{\text{mon}} \overline{L}$

$\langle Q, q_{in}, q_{out}, \mathrm{X}, \rightarrow \rangle$

Program

Desired Analysis

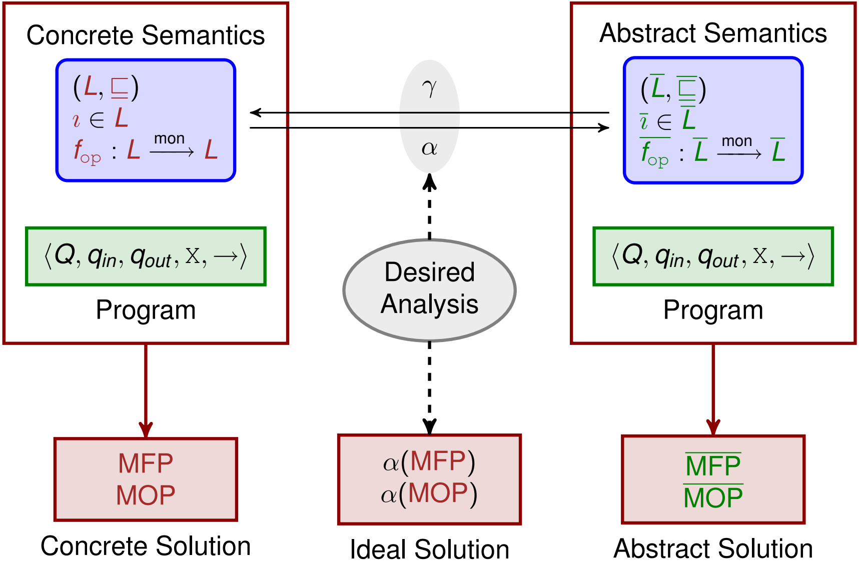
MFP
MOP

Concrete Solution

$\alpha$(MFP)
$\alpha$(MOP)

Ideal Solution

# Systematic Design of Correct of Data Flow Analyses



Concrete Semantics

$(L, \sqsubseteq)$

$\imath \in L$

$f_{\mathrm{op}} : L \xrightarrow{\text{mon}} L$

$\langle Q, q_{in}, q_{out}, \mathrm{X}, \rightarrow \rangle$

Program

$\gamma$

$\alpha$

Desired Analysis

Abstract Semantics

$(\overline{L}, \overline{\sqsubseteq})$

$\overline{\imath} \in \overline{L}$

$\overline{f}_{\mathrm{op}} : \overline{L} \xrightarrow{\text{mon}} \overline{L}$

$\langle Q, q_{in}, q_{out}, \mathrm{X}, \rightarrow \rangle$

Program

MFP

MOP

Concrete Solution

$\alpha$(MFP)

$\alpha$(MOP)

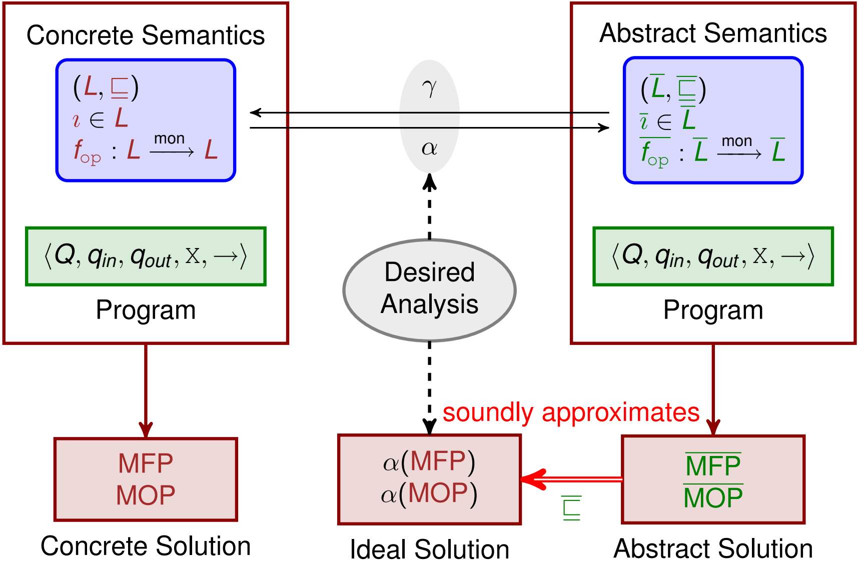
Ideal Solution

$\overline{\text{MFP}}$

$\overline{\text{MOP}}$

Abstract Solution

# Systematic Design of Correct of Data Flow Analyses



Concrete Semantics

$(L, \sqsubseteq)$

$\imath \in L$

$f_{\mathrm{op}} : L \xrightarrow{\text{mon}} L$

$\langle Q, q_{in}, q_{out}, \mathrm{X}, \rightarrow \rangle$

Program

$\gamma$

$\alpha$

Desired Analysis

Abstract Semantics

$(\overline{L}, \overline{\sqsubseteq})$

$\overline{\imath} \in \overline{L}$

$\overline{f_{\mathrm{op}}} : \overline{L} \xrightarrow{\text{mon}} \overline{L}$

$\langle Q, q_{in}, q_{out}, \mathrm{X}, \rightarrow \rangle$

Program

MFP

MOP

Concrete Solution

$\alpha$(MFP)

$\alpha$(MOP)

Ideal Solution

soundly approximates

$\overline{\sqsubseteq}$

$\overline{\text{MFP}}$

$\overline{\text{MOP}}$

Abstract Solution

# Systematic Design of Correct of Data Flow Analyses

The MOP solution of the concrete semantics is the strongest property (i.e. the most precise fact) that is satisfied by all runs of the program.

The ideal solution to a given analysis is an approximation of the concrete MOP solution.

## Natural Limitation

The class of possible analyses depends on the choice of

“standard” concrete semantics.

Abstract data flow facts and transfer functions cannot be more precise than concrete ones.

Our operational semantics: $\langle Q \times (\mathbb{X} \to \mathbb{R}), \mathit{Init}, \mathit{Out}, \mathrm{Op}, \to \rangle$

Focus on numerical analyses

# Systematic Design of Correct of Data Flow Analyses

The MOP solution of the concrete semantics is the strongest property (i.e. the most precise fact) that is satisfied by all runs of the program.

The ideal solution to a given analysis is an approximation of the concrete MOP solution.

## Natural Limitation

The class of possible analyses depends on the choice of

“standard” concrete semantics.

Abstract data flow facts and transfer functions cannot be more precise than concrete ones.

Our operational semantics: $\langle Q \times (\mathtt{X} \to \mathbb{R}), \mathit{Init}, \mathit{Out}, \mathtt{Op}, \to \rangle$

Focus on numerical analyses

# Standard Concrete Semantics

Control Flow Automaton: $\langle Q, q_{in}, q_{out}, \mathtt{X}, \rightarrow \rangle$

## Recall: semantics $[\![\mathtt{op}]\!]$ of operations $\mathtt{op} \in \mathtt{Op}$

$$[\![\mathtt{op}]\!] \quad \subseteq \quad (\mathtt{X} \rightarrow \mathbb{R}) \times (\mathtt{X} \rightarrow \mathbb{R})$$

## Monotone Framework

- Complete lattice $(L, \sqsubseteq)$ of data flow facts: $(\mathcal{P}(\mathtt{X} \rightarrow \mathbb{R}), \subseteq)$
- Set $\mathcal{F}$ of monotonic transfer functions:

$$\mathcal{F} \;\; = \;\; \{\lambda\,\phi\,.\,R[\phi] \mid R \subseteq (\mathtt{X} \rightarrow \mathbb{R}) \times (\mathtt{X} \rightarrow \mathbb{R})\}$$

## Data Flow Instance $\overrightarrow{\mathcal{S}}$ for Forward Analysis

- Initial data flow value: $\top = \mathtt{X} \rightarrow \mathbb{R}$
- Transfer mapping: $f_{\mathtt{op}}(\phi) \;=\; [\![\mathtt{op}]\!][\phi]$

# Standard Concrete Semantics

Control Flow Automaton: $\langle Q, q_{in}, q_{out}, \mathtt{X}, \rightarrow \rangle$

## Recall: semantics $[\![\mathtt{op}]\!]$ of operations $\mathtt{op} \in \mathtt{Op}$

$$[\![\mathtt{op}]\!] \quad \subseteq \quad (\mathtt{X} \rightarrow \mathbb{R}) \times (\mathtt{X} \rightarrow \mathbb{R})$$

## Monotone Framework

- Complete lattice $(L, \sqsubseteq)$ of data flow facts: $(\mathcal{P}(\mathtt{X} \rightarrow \mathbb{R}), \subseteq)$
- Set $\mathcal{F}$ of monotonic transfer functions:

$$\mathcal{F} \quad = \quad \{\lambda\, \phi \,.\, R[\phi] \mid R \subseteq (\mathtt{X} \rightarrow \mathbb{R}) \times (\mathtt{X} \rightarrow \mathbb{R})\}$$

## Data Flow Instance $\overleftarrow{\mathcal{S}}$ for Backward Analysis

- Initial data flow value: $\top = \mathtt{X} \rightarrow \mathbb{R}$
- Transfer mapping: $f_{\mathtt{op}}(\phi) \ = \ [\![\mathtt{op}]\!]^{-1}[\phi]$

# Standard Concrete Semantics

Control Flow Automaton: $\langle Q, q_{in}, q_{out}, \mathtt{X}, \rightarrow \rangle$

## Recall: semantics $[\![\mathtt{op}]\!]$ of operations $\mathtt{op} \in \mathtt{Op}$

$$[\![\mathtt{op}]\!] \quad \subseteq \quad (\mathtt{X} \rightarrow \mathbb{R}) \times (\mathtt{X} \rightarrow \mathbb{R})$$

## Monotone Framework: Distributive

- Complete lattice $(L, \sqsubseteq)$ of data flow facts: $(\mathcal{P}(\mathtt{X} \rightarrow \mathbb{R}), \subseteq)$
- Set $\mathcal{F}$ of monotonic transfer functions:

$$\mathcal{F} \quad = \quad \{\lambda\,\phi\,.\,R[\phi] \mid R \subseteq (\mathtt{X} \rightarrow \mathbb{R}) \times (\mathtt{X} \rightarrow \mathbb{R})\}$$

## Data Flow Instance $\overleftarrow{\mathcal{S}}$ for Backward Analysis

- Initial data flow value: $\top = \mathtt{X} \rightarrow \mathbb{R}$
- Transfer mapping: $f_{\mathtt{op}}(\phi) \;=\; [\![\mathtt{op}]\!]^{-1}[\phi]$

# Post$^*$ and Pre$^*$ as Data Flow Analysis Solutions

Consider a control flow automaton: $\langle Q, q_{in}, q_{out}, \mathtt{X}, \rightarrow \rangle$. Recall that:

$$\mathrm{Post}^* \;=\; \bigcup_{q_{in} \xrightarrow{\mathtt{op}_0} \cdots \xrightarrow{\mathtt{op}_k} q} \{q\} \times ([\![\mathtt{op}_k]\!] \circ \cdots \circ [\![\mathtt{op}_0]\!])\,[(\mathtt{X} \to \mathbb{R})]$$

$$\mathrm{Pre}^* \;=\; \bigcup_{q \xrightarrow{\mathtt{op}_0} \cdots \xrightarrow{\mathtt{op}_k} q_{out}} \{q\} \times \left(([\![\mathtt{op}_k]\!] \circ \cdots \circ [\![\mathtt{op}_0]\!])^{-1}\right)[(\mathtt{X} \to \mathbb{R})]$$

$$\mathrm{Post}^* \;=\; \overrightarrow{\mathrm{MOP}}\left(\overrightarrow{\mathbb{S}}\right) \;=\; \overrightarrow{\mathrm{MFP}}\left(\overrightarrow{\mathbb{S}}\right)$$

$$\mathrm{Pre}^* \;=\; \overleftarrow{\mathrm{MOP}}\left(\overleftarrow{\mathbb{S}}\right) \;=\; \overleftarrow{\mathrm{MFP}}\left(\overleftarrow{\mathbb{S}}\right)$$

# Post$^*$ and Pre$^*$ as Data Flow Analysis Solutions

Consider a control flow automaton: $\langle Q, q_{in}, q_{out}, \mathrm{X}, \rightarrow \rangle$. Recall that:

$$\mathrm{Post}^* = \bigcup_{q_{in} \xrightarrow{\mathrm{op}_0} \cdots \xrightarrow{\mathrm{op}_k} q} \{q\} \times (\llbracket \mathrm{op}_k \rrbracket \circ \cdots \circ \llbracket \mathrm{op}_0 \rrbracket)\,[(\mathrm{X} \to \mathbb{R})]$$

$$\mathrm{Pre}^* = \bigcup_{q \xrightarrow{\mathrm{op}_0} \cdots \xrightarrow{\mathrm{op}_k} q_{out}} \{q\} \times \left((\llbracket \mathrm{op}_k \rrbracket \circ \cdots \circ \llbracket \mathrm{op}_0 \rrbracket)^{-1}\right)[(\mathrm{X} \to \mathbb{R})]$$

$$\mathrm{Post}^* = \overrightarrow{\mathrm{MOP}}\left(\overrightarrow{\mathcal{S}}\right) = \overrightarrow{\mathrm{MFP}}\left(\overrightarrow{\mathcal{S}}\right)$$

$$\mathrm{Pre}^* = \overleftarrow{\mathrm{MOP}}\left(\overleftarrow{\mathcal{S}}\right) = \overleftarrow{\mathrm{MFP}}\left(\overleftarrow{\mathcal{S}}\right)$$

# Abstraction of the Concrete Semantics: Intuition

### Concrete Semantics

$\langle (\mathcal{P}(\mathrm{X} \to \mathbb{R}), \subseteq), \mathcal{F}, Q, q_{in}, q_{out}, \mathrm{X}, \to, f, \imath \rangle$

### Galois connection

$(\mathcal{P}(\mathrm{X} \to \mathbb{R}), \subseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (\overline{L}, \overline{\sqsubseteq})$

$\overline{L}$ is a set of machine-representable “properties” of the variables.

### Example

$$\overline{L} \;=\; \{\mathrm{x} \text{ is even},\;\; \mathrm{y} \text{ is odd or negative},\;\; \mathrm{x} \geq \mathrm{y} \;\Rightarrow\; \mathrm{x} = 2^i\}$$

$\gamma(\overline{\psi})$ is the meaning of an abstract “property” $\overline{\psi}$.

$\alpha(\phi)$ encodes a sound approximation of $\phi$, the most precise one.

$\overline{\sqsubseteq}$ corresponds to entailment between “properties”, and abstracts $\subseteq$.

# Abstraction of the Concrete Semantics: Intuition

## Concrete Semantics

$\langle (\mathcal{P}(\mathrm{X} \to \mathbb{R}), \subseteq), \mathcal{F}, Q, q_{in}, q_{out}, \mathrm{X}, \to, f, \imath \rangle$

## Galois connection

$(\mathcal{P}(\mathrm{X} \to \mathbb{R}), \subseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (\overline{L}, \overline{\sqsubseteq})$

$\overline{L}$ is a set of machine-representable “properties” of the variables.

## Example

$$\overline{L} \;=\; \{\mathrm{x} \text{ is even},\;\; \mathrm{y} \text{ is odd or negative},\;\; \mathrm{x} \geq \mathrm{y} \;\Rightarrow\; \mathrm{x} = 2^i\}$$

$\gamma(\overline{\psi})$ is the meaning of an abstract “property” $\overline{\psi}$.

$\alpha(\phi)$ encodes a sound approximation of $\phi$, the most precise one.

$\overline{\sqsubseteq}$ corresponds to entailment between “properties”, and abstracts $\subseteq$.

# Abstract Semantics Induced by a Galois Connection

Consider a data flow instance $\mathcal{A} = \langle (L, \sqsubseteq), \mathcal{F}, Q, q_{in}, q_{out}, \mathrm{X}, \rightarrow, f, \imath \rangle$ and a Galois connection $(L, \sqsubseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (\overline{L}, \overline{\sqsubseteq})$.

## Definition

The abstract data flow instance $\overline{\mathcal{A}}$ induced by $\mathcal{A}$ and $(L, \sqsubseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (\overline{L}, \overline{\sqsubseteq})$ is $\overline{\mathcal{A}} = \langle (\overline{L}, \overline{\sqsubseteq}), \overline{\mathcal{F}}, Q, q_{in}, q_{out}, \mathrm{X}, \rightarrow, \overline{f}, \overline{\imath} \rangle$ where:

$$\begin{aligned} \overline{\mathcal{F}} &= \overline{L} \xrightarrow{\text{mon}} \overline{L} \\ \overline{f} &= \lambda\, \mathrm{op} \,.\, f^{\sharp}_{\mathrm{op}} \\ \overline{\imath} &= \alpha(\imath) \end{aligned}$$

Recall that $f^{\sharp}_{\mathrm{op}} = \alpha \circ f_{\mathrm{op}} \circ \gamma$ is the *best abstraction* of $f_{\mathrm{op}}$.

# Correctness of Induced Abstract Data Flow Analysis

## Extension of Galois Connections to Functions

For any set $Q$ and Galois connection $(L, \sqsubseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (\overline{L}, \overline{\sqsubseteq})$, we have

$$(Q \to L, \sqsubseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (Q \to \overline{L}, \overline{\sqsubseteq}) \quad \text{where:} \quad \begin{aligned} \alpha(a) &= \lambda q \,.\, \alpha(a(q)) \\ \gamma(\overline{b}) &= \lambda q \,.\, \gamma(\overline{b}(q)) \end{aligned}$$

## Theorem (Correctness of Induced Abstract Forward Analysis)

*For any data flow instance $\mathcal{A}$ and Galois connection $(L, \sqsubseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (\overline{L}, \overline{\sqsubseteq})$, the induced abstract data flow instance $\overline{\mathcal{A}}$ satisfies:*

$$\begin{aligned} \overrightarrow{\text{MFP}}(\mathcal{A}) &\sqsubseteq \gamma\left(\overrightarrow{\text{MFP}}(\overline{\mathcal{A}})\right) & \qquad \alpha\left(\overrightarrow{\text{MFP}}(\mathcal{A})\right) &\mathrel{\overline{\sqsubseteq}} \overrightarrow{\text{MFP}}(\overline{\mathcal{A}}) \\ \overrightarrow{\text{MOP}}(\mathcal{A}) &\sqsubseteq \gamma\left(\overrightarrow{\text{MOP}}(\overline{\mathcal{A}})\right) & \qquad \alpha\left(\overrightarrow{\text{MOP}}(\mathcal{A})\right) &\mathrel{\overline{\sqsubseteq}} \overrightarrow{\text{MOP}}(\overline{\mathcal{A}}) \end{aligned}$$

# Correctness of Induced Abstract Data Flow Analysis

## Extension of Galois Connections to Functions

For any set $Q$ and Galois connection $(L, \sqsubseteq) \mathrel{\underset{\alpha}{\overset{\gamma}{\leftrightarrows}}} (\overline{L}, \overline{\sqsubseteq})$, we have

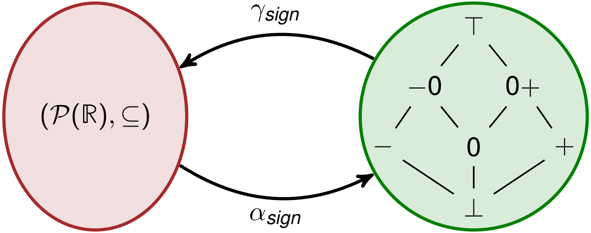
$$(Q \to L, \sqsubseteq) \mathrel{\underset{\alpha}{\overset{\gamma}{\leftrightarrows}}} (Q \to \overline{L}, \overline{\sqsubseteq}) \quad \text{where:} \quad \begin{array}{rcl} \alpha(a) & = & \lambda\, q \,.\, \alpha(a(q)) \\ \gamma(\overline{b}) & = & \lambda\, q \,.\, \gamma(\overline{b}(q)) \end{array}$$

## Theorem (Correctness of Induced Abstract Backward Analysis)

*For any data flow instance $\mathcal{A}$ and Galois connection $(L, \sqsubseteq) \mathrel{\underset{\alpha}{\overset{\gamma}{\leftrightarrows}}} (\overline{L}, \overline{\sqsubseteq})$, the induced abstract data flow instance $\overline{\mathcal{A}}$ satisfies:*

$$\begin{array}{rclcrcl} \overleftarrow{\mathrm{MFP}}\,(\mathcal{A}) & \sqsubseteq & \gamma\left(\overleftarrow{\mathrm{MFP}}\,(\overline{\mathcal{A}})\right) & \qquad & \alpha\left(\overleftarrow{\mathrm{MFP}}\,(\mathcal{A})\right) & \overline{\sqsubseteq} & \overleftarrow{\mathrm{MFP}}\,(\overline{\mathcal{A}}) \\ \overleftarrow{\mathrm{MOP}}\,(\mathcal{A}) & \sqsubseteq & \gamma\left(\overleftarrow{\mathrm{MOP}}\,(\overline{\mathcal{A}})\right) & \qquad & \alpha\left(\overleftarrow{\mathrm{MOP}}\,(\mathcal{A})\right) & \overline{\sqsubseteq} & \overleftarrow{\mathrm{MOP}}\,(\overline{\mathcal{A}}) \end{array}$$

# Back Again to Sign Analysis: Galois Connection



| $\gamma$ | | | | | | | |
|---|---|---|---|---|---|---|---|
| $\overline{y}$ | $\bot$ | $-$ | $0$ | $+$ | $-0$ | $0+$ | $\top$ |
| $\gamma(\overline{y})$ | $\emptyset$ | $\{r \mid r < 0\}$ | $\{0\}$ | $\{r \mid r > 0\}$ | $\{r \mid r \leq 0\}$ | $\{r \mid r \geq 0\}$ | $\mathbb{R}$ |

## Objective

Design a Galois Connection between:

- $(\mathcal{P}(\mathtt{X} \to \mathbb{R}), \subseteq)$, concrete data flow facts from standard semantics
- $(\mathtt{X} \to \mathit{Sign}, \sqsubseteq)$, abstract data flow facts

# Intermediate Galois Connection: Projection

Convenient intermediate step for non-relational analyses

## Objective of Projection

Design a Galois Connection between:

- $(\mathcal{P}(\mathrm{X} \to \mathbb{R}), \subseteq)$, concrete data flow facts from standard semantics
- $(\mathrm{X} \to \mathcal{P}(\mathbb{R}), \overline{\subseteq})$, projected data flow facts

where $\overline{\subseteq}$ is as expected: $\overline{\psi} \;\overline{\subseteq}\; \overline{\psi'}$ if $\overline{\psi}(x) \subseteq \overline{\psi'}(x)$ for all $x \in \mathrm{X}$.

## Fact

$$(\mathcal{P}(\mathrm{X} \to \mathbb{R}), \subseteq) \underset{\alpha_\pi}{\overset{\gamma_\pi}{\leftrightarrows}} (\mathrm{X} \to \mathcal{P}(\mathbb{R}), \overline{\subseteq})$$

*where:*

$$\begin{aligned} \alpha_\pi(\phi) &= \lambda x\,.\,\{v(x) \mid v \in \phi\} \\ \gamma_\pi(\overline{\psi}) &= \{v \in \mathrm{X} \to \mathbb{R} \mid v(x) \in \overline{\psi}(x) \textit{ for all } x \in \mathrm{X}\} \end{aligned}$$

# Intermediate Galois Connection: Projection

Convenient intermediate step for non-relational analyses

## Objective of Projection

Design a Galois Connection between:

- $(\mathcal{P}(\mathtt{X} \to \mathbb{R}), \subseteq)$, concrete data flow facts from standard semantics
- $(\mathtt{X} \to \mathcal{P}(\mathbb{R}), \overline{\subseteq})$, projected data flow facts

where $\overline{\subseteq}$ is as expected: $\overline{\psi} \;\overline{\subseteq}\; \overline{\psi'}$ if $\overline{\psi}(x) \subseteq \overline{\psi'}(x)$ for all $x \in \mathtt{X}$.

## Fact

$$(\mathcal{P}(\mathtt{X} \to \mathbb{R}), \subseteq) \underset{\alpha_\pi}{\overset{\gamma_\pi}{\leftrightarrows}} (\mathtt{X} \to \mathcal{P}(\mathbb{R}), \overline{\subseteq})$$

*where:*

$$\begin{aligned} \alpha_\pi(\phi) &= \lambda x\,.\,\{v(x) \mid v \in \phi\} \\ \gamma_\pi(\overline{\psi}) &= \{v \in \mathtt{X} \to \mathbb{R} \mid v(x) \in \overline{\psi}(x) \textit{ for all } x \in \mathtt{X}\} \end{aligned}$$

# Intermediate Galois Connection: Projection

Convenient intermediate step for non-relational analyses

## Objective of Projection

Design a Galois Connection between:

- $(\mathcal{P}(\mathrm{X} \to \mathbb{R}), \subseteq)$, concrete data flow facts from standard semantics
- $(\mathrm{X} \to \mathcal{P}(\mathbb{R}), \overline{\subseteq})$, projected data flow facts

where $\overline{\subseteq}$ is as expected: $\overline{\psi} \;\overline{\subseteq}\; \overline{\psi'}$ if $\overline{\psi}(x) \subseteq \overline{\psi'}(x)$ for all $x \in \mathrm{X}$.

## Fact

$$(\mathcal{P}(\mathrm{X} \to \mathbb{R}), \subseteq) \underset{\alpha_\pi}{\overset{\gamma_\pi}{\leftrightarrows}} (\mathrm{X} \to \mathcal{P}(\mathbb{R}), \overline{\subseteq})$$

*where:*

$$\begin{aligned} \alpha_\pi(\phi) &= \lambda x \,.\, \{v(x) \mid v \in \phi\} \\ \gamma_\pi(\overline{\psi}) &= \left\{ v \in \mathrm{X} \to \mathbb{R} \;\middle|\; v(x) \in \overline{\psi}(x) \textit{ for all } x \in \mathrm{X} \right\} \end{aligned}$$

# Back Again to Sign Analysis: Galois Connection

## Projection

$$(\mathcal{P}(\mathrm{X} \to \mathbb{R}), \subseteq) \underset{\alpha_\pi}{\overset{\gamma_\pi}{\leftrightarrows}} (\mathrm{X} \to \mathcal{P}(\mathbb{R}), \subseteq)$$

## Sign

$$(\mathcal{P}(\mathbb{R}), \subseteq) \underset{\alpha_{sign}}{\overset{\gamma_{sign}}{\leftrightarrows}} (\mathit{Sign}, \sqsubseteq)$$

## Extension of Sign to Functions

$$(\mathrm{X} \to \mathcal{P}(\mathbb{R}), \subseteq) \underset{\alpha_{sign}}{\overset{\gamma_{sign}}{\leftrightarrows}} (\mathrm{X} \to \mathit{Sign}, \sqsubseteq)$$

$$(\mathcal{P}(\mathrm{X} \to \mathbb{R}), \subseteq) \underset{\alpha_{sign} \circ \alpha_\pi}{\overset{\gamma_\pi \circ \gamma_{sign}}{\leftrightarrows}} (\mathrm{X} \to \mathit{Sign}, \sqsubseteq)$$

The composition of Galois connections is a Galois connection.

# Back Again to Sign Analysis: Induced Instance

$$(\mathcal{P}(\mathtt{X} \to \mathbb{R}), \subseteq) \underset{\alpha_\pi}{\overset{\gamma_\pi}{\leftrightarrows}} (\mathtt{X} \to \mathcal{P}(\mathbb{R}), \subseteq) \underset{\alpha_{sign}}{\overset{\gamma_{sign}}{\leftrightarrows}} (\mathtt{X} \to \mathit{Sign}, \overline{\sqsubseteq})$$

$$\begin{aligned} \overline{\mathcal{F}} &= (\mathtt{X} \to \mathit{Sign}) \xrightarrow{\text{mon}} (\mathtt{X} \to \mathit{Sign}) \\ \overline{f} &= \lambda\, \mathtt{op} \,.\, f^\sharp_{\mathtt{op}} \\ \overline{\imath} &= \alpha_{sign} \circ \alpha_\pi(\top) = \lambda\, x \,.\, \overline{\top} \end{aligned}$$

But this data flow instance looks similar to what we did previously (less painfully) without Galois connections. . .

What do we get?

The most precise transfer mapping that soundly approximates the standard semantics

# Back Again to Sign Analysis: Induced Instance

$$(\mathcal{P}(\mathtt{X} \to \mathbb{R}), \subseteq) \underset{\alpha_\pi}{\overset{\gamma_\pi}{\leftrightarrows}} (\mathtt{X} \to \mathcal{P}(\mathbb{R}), \subseteq) \underset{\alpha_{sign}}{\overset{\gamma_{sign}}{\leftrightarrows}} (\mathtt{X} \to \mathit{Sign}, \overline{\sqsubseteq})$$

$$\begin{aligned} \overline{\mathcal{F}} &= (\mathtt{X} \to \mathit{Sign}) \xrightarrow{\text{mon}} (\mathtt{X} \to \mathit{Sign}) \\ \overline{f} &= \lambda\, \mathrm{op} \,.\, f^\sharp_{\mathrm{op}} \\ \overline{\imath} &= \alpha_{sign} \circ \alpha_\pi(\top) = \lambda\, x \,.\, \overline{\top} \end{aligned}$$

But this data flow instance looks similar to what we did previously (less painfully) without Galois connections...

What do we get?

The most precise transfer mapping that soundly approximates the standard semantics

# Forward Sign Analysis: Transfer Mapping

$$(\mathcal{P}(\mathrm{X} \to \mathbb{R}), \subseteq) \underset{\alpha_\pi}{\overset{\gamma_\pi}{\leftrightarrows}} (\mathrm{X} \to \mathcal{P}(\mathbb{R}), \subseteq) \underset{\alpha_{sign}}{\overset{\gamma_{sign}}{\leftrightarrows}} (\mathrm{X} \to \mathit{Sign}, \sqsubseteq)$$

$$f^\sharp_{\mathrm{op}}(\overline{\psi}) \quad = \quad \alpha_{sign} \circ \alpha_\pi \left( [\![\mathrm{op}]\!] \left[ \gamma_\pi \circ \gamma_{sign}(\overline{\psi}) \right] \right)$$

Extensions of $[\![e]\!]$ and $[\![g]\!]$ to subsets of $\mathbb{R}$

| $[\![e]\!]$ |
|---|
| $\mathcal{P}(\mathrm{X} \to \mathbb{R}) \quad \to \quad \mathcal{P}(\mathbb{R})$<br>$\phi \quad \mapsto \quad \{[\![e]\!]_v \mid v \in \phi\}$ |

| $[\![g]\!]$ |
|---|
| $\mathcal{P}(\mathrm{X} \to \mathbb{R}) \quad \to \quad \mathcal{P}(\mathrm{X} \to \mathbb{R})$<br>$\phi \quad \mapsto \quad \{v \in \phi \mid v \models g\}$ |

$$f^\sharp_{x:=e}(\overline{v}) \quad = \quad \lambda y \,.\, \begin{cases} \overline{v}(y) & \text{if } y \neq x \\ \alpha_{sign} \circ [\![e]\!] \circ \gamma(\overline{v}) & \text{if } y = x \end{cases}$$

$$f^\sharp_g \quad = \quad \alpha_{sign} \circ \alpha_\pi \circ [\![g]\!] \circ \gamma_\pi \circ \gamma_{sign}$$

# Forward Sign Analysis: Transfer Mapping

$$(\mathcal{P}(\mathtt{X} \to \mathbb{R}), \subseteq) \underset{\alpha_\pi}{\overset{\gamma_\pi}{\leftrightarrows}} (\mathtt{X} \to \mathcal{P}(\mathbb{R}), \subseteq) \underset{\alpha_{sign}}{\overset{\gamma_{sign}}{\leftrightarrows}} (\mathtt{X} \to Sign, \overline{\sqsubseteq})$$

$$f^\sharp_{\mathrm{op}}(\overline{\psi}) \quad = \quad \alpha_{sign} \circ \alpha_\pi \left( [\![\mathrm{op}]\!] \left[ \gamma_\pi \circ \gamma_{sign}(\overline{\psi}) \right] \right)$$

Extensions of $[\![e]\!]$ and $[\![g]\!]$ to subsets of $\mathbb{R}$

| $[\![e]\!]$ | | |
|---|---|---|
| $\mathcal{P}(\mathtt{X} \to \mathbb{R})$ | $\to$ | $\mathcal{P}(\mathbb{R})$ |
| $\phi$ | $\mapsto$ | $\{[\![e]\!]_v \mid v \in \phi\}$ |

| $[\![g]\!]$ | | |
|---|---|---|
| $\mathcal{P}(\mathtt{X} \to \mathbb{R})$ | $\to$ | $\mathcal{P}(\mathtt{X} \to \mathbb{R})$ |
| $\phi$ | $\mapsto$ | $\{v \in \phi \mid v \models g\}$ |

$$f^\sharp_{x:=e}(\overline{v}) \quad = \quad \lambda y\,.\begin{cases} \overline{v}(y) & \text{if } y \neq x \\ \alpha_{sign} \circ [\![e]\!] \circ \gamma(\overline{v}) & \text{if } y = x \end{cases}$$

$$f^\sharp_g \quad = \quad \alpha_{sign} \circ \alpha_\pi \circ [\![g]\!] \circ \gamma_\pi \circ \gamma_{sign}$$

# Forward Sign Analysis: Transfer Mapping

$$(\mathcal{P}(\mathrm{X} \to \mathbb{R}), \subseteq) \underset{\alpha_\pi}{\overset{\gamma_\pi}{\leftrightarrows}} (\mathrm{X} \to \mathcal{P}(\mathbb{R}), \subseteq) \underset{\alpha_{sign}}{\overset{\gamma_{sign}}{\leftrightarrows}} (\mathrm{X} \to \mathit{Sign}, \sqsubseteq)$$

$$f^\sharp_{\mathrm{op}}(\overline{\psi}) \quad = \quad \alpha_{sign} \circ \alpha_\pi \left( [\![\mathrm{op}]\!] \left[ \gamma_\pi \circ \gamma_{sign}(\overline{\psi}) \right] \right)$$

Extensions of $[\![e]\!]$ and $[\![g]\!]$ to subsets of $\mathbb{R}$

| $[\![e]\!]$ | | |
|---|---|---|
| $\mathcal{P}(\mathrm{X} \to \mathbb{R})$ | $\to$ | $\mathcal{P}(\mathbb{R})$ |
| $\phi$ | $\mapsto$ | $\{[\![e]\!]_v \mid v \in \phi\}$ |

| $[\![g]\!]$ | | |
|---|---|---|
| $\mathcal{P}(\mathrm{X} \to \mathbb{R})$ | $\to$ | $\mathcal{P}(\mathrm{X} \to \mathbb{R})$ |
| $\phi$ | $\mapsto$ | $\{v \in \phi \mid v \models g\}$ |

$$f^\sharp_{x:=e}(\overline{v}) \quad = \quad \lambda y \,.\, \begin{cases} \overline{v}(y) & \text{if } y \neq x \\ \alpha_{sign} \circ [\![e]\!] \circ \gamma(\overline{v}) & \text{if } y = x \end{cases}$$

$$f^\sharp_g \quad = \quad \alpha_{sign} \circ \alpha_\pi \circ [\![g]\!] \circ \gamma_\pi \circ \gamma_{sign}$$

# Forward Sign Analysis: Transfer Mapping

$$(\mathcal{P}(\mathrm{X} \to \mathbb{R}), \subseteq) \underset{\alpha_\pi}{\overset{\gamma_\pi}{\leftrightarrows}} (\mathrm{X} \to \mathcal{P}(\mathbb{R}), \subseteq) \underset{\alpha_{sign}}{\overset{\gamma_{sign}}{\leftrightarrows}} (\mathrm{X} \to \mathit{Sign}, \sqsubseteq)$$

$$f^\sharp_{x:=e}(\overline{v}) \quad = \quad \lambda\, y\,.\begin{cases} \overline{v}(y) & \text{if } y \neq x \\ \alpha_{sign} \circ [\![e]\!] \circ \gamma(\overline{v}) & \text{if } y = x \end{cases}$$

Not easy to compute!

## Example

Consider a (non-constant) multivariate polynomial expression *e* and the operation $\mathrm{op} = \mathrm{x} := e * e$.

$$f^\sharp_{\mathrm{op}}(\top) \quad = \quad \lambda\, y\,.\begin{cases} \top & \text{if } y \neq \mathrm{x} \\ 0+ & \text{if } y = \mathrm{x} \text{ and } e \text{ has a root} \\ + & \text{if } y = \mathrm{x} \text{ and } e \text{ has no root} \end{cases}$$

# Forward Sign Analysis: Transfer Mapping

$$(\mathcal{P}(\mathrm{X} \to \mathbb{R}), \subseteq) \underset{\alpha_\pi}{\overset{\gamma_\pi}{\leftrightarrows}} (\mathrm{X} \to \mathcal{P}(\mathbb{R}), \subseteq) \underset{\alpha_{sign}}{\overset{\gamma_{sign}}{\leftrightarrows}} (\mathrm{X} \to \mathit{Sign}, \sqsubseteq)$$

$$f^\sharp_{x:=e}(\overline{v}) \quad = \quad \lambda y \, . \begin{cases} \overline{v}(y) & \text{if } y \neq x \\ \alpha_{sign} \circ [\![ e ]\!] \circ \gamma(\overline{v}) & \text{if } y = x \end{cases}$$

Not easy to compute! Even for the simple *Sign* lattice!

## Example

Consider a (non-constant) multivariate polynomial expression *e* and the operation $\mathrm{op} = \mathrm{x} := e \star e$.

$$f^\sharp_{\mathrm{op}}(\overline{\top}) \quad = \quad \lambda y \, . \begin{cases} \top & \text{if } y \neq \mathrm{x} \\ 0+ & \text{if } y = \mathrm{x} \text{ and } e \text{ has a root} \\ + & \text{if } y = \mathrm{x} \text{ and } e \text{ has no root} \end{cases}$$

# Forward Sign Analysis: Transfer Mapping

$$(\mathcal{P}(\mathrm{X} \to \mathbb{R}), \subseteq) \underset{\alpha_\pi}{\overset{\gamma_\pi}{\leftrightarrows}} (\mathrm{X} \to \mathcal{P}(\mathbb{R}), \subseteq) \underset{\alpha_{sign}}{\overset{\gamma_{sign}}{\leftrightarrows}} (\mathrm{X} \to \mathit{Sign}, \sqsubseteq)$$

$$f^\sharp_{x:=e}(\overline{v}) \quad = \quad \lambda\, y\, .\begin{cases} \overline{v}(y) & \text{if } y \neq x \\ \alpha_{sign} \circ [\![e]\!] \circ \gamma(\overline{v}) & \text{if } y = x \end{cases}$$

Not easy to compute! Even for the simple *Sign* lattice!

## Example

Consider a (non-constant) multivariate polynomial expression *e* and the operation $\mathrm{op} = \mathrm{x} := e \star e$.

$$f^\sharp_{\mathrm{op}}(\overline{\top}) \quad = \quad \lambda\, y\, .\begin{cases} \overline{\top} & \text{if } y \neq \mathrm{x} \\ 0+ & \text{if } y = \mathrm{x} \text{ and } e \text{ has a root} \\ + & \text{if } y = \mathrm{x} \text{ and } e \text{ has no root} \end{cases}$$

# Forward Sign Analysis: Transfer Mapping

What can be done?

# Forward Sign Analysis: Transfer Mapping

What can be done?

Approximate!

But soundly ☺

# Forward Sign Analysis: Transfer Mapping

What can be done?

Approximate!

But soundly ☺

## Approximate Transfer Mapping

Replace each $f^\sharp_{\mathrm{op}}$ with an approximate transfer function $\overline{h_{\mathrm{op}}}$ that

- exploits the structure of operations to obtain
- better performance at the expense of precision.

# Last Bit of Lattice Theory

## Theorem

*For any two monotonic functions $f, g$ on a complete lattice $(L, \sqsubseteq)$,*

$$\textit{if} \quad f(x) \sqsubseteq g(x) \quad \textit{for all } x \in L \quad \textit{then} \quad \mathrm{lfp}(f) \sqsubseteq \mathrm{lfp}(g)$$

## Proof.

$$\{x \in L \mid f(x) \sqsubseteq x\} \quad \supseteq \quad \{x \in L \mid g(x) \sqsubseteq x\}$$

Hence

$$\mathrm{lfp}(f) \;=\; \bigsqcap \{x \in L \mid f(x) \sqsubseteq x\} \;\sqsubseteq\; \bigsqcap \{x \in L \mid g(x) \sqsubseteq x\} \;=\; \mathrm{lfp}(g)$$

□

# Last Bit of Lattice Theory

## Theorem

*For any two monotonic functions $f, g$ on a complete lattice $(L, \sqsubseteq)$,*

$$\textit{if} \quad f(x) \ \sqsubseteq \ g(x) \quad \textit{for all } x \in L \quad \textit{then} \quad \mathrm{lfp}(f) \ \sqsubseteq \ \mathrm{lfp}(g)$$

## Proof.

$$\{x \in L \mid f(x) \sqsubseteq x\} \ \supseteq \ \{x \in L \mid g(x) \sqsubseteq x\}$$

Hence

$$\mathrm{lfp}(f) \ = \ \bigsqcap \{x \in L \mid f(x) \sqsubseteq x\} \ \sqsubseteq \ \bigsqcap \{x \in L \mid g(x) \sqsubseteq x\} \ = \ \mathrm{lfp}(g)$$

□

# Correctness of Approximate Transfer Mapping

Consider a data flow instance $\mathcal{A}$ with a set $\mathcal{F}$ of transfer functions and a transfer mapping $f : \mathtt{Op} \rightarrow \mathcal{F}$.

For any monotonic function $h : \mathtt{Op} \rightarrow \mathcal{F}$ verifying

$$f_{\mathrm{op}}(x) \;\sqsubseteq\; h_{\mathrm{op}}(x) \qquad \text{(for all } \mathrm{op} \in \mathtt{Op}, x \in L\text{)}$$

the data flow instance $\mathcal{B}$ obtained from $\mathcal{A}$ by replacing $f$ with $h$ satisfies:

$$\overrightarrow{\mathrm{MFP}}(\mathcal{A}) \;\sqsubseteq\; \overrightarrow{\mathrm{MFP}}(\mathcal{B}) \qquad\qquad \overrightarrow{\mathrm{MOP}}(\mathcal{A}) \;\sqsubseteq\; \overrightarrow{\mathrm{MOP}}(\mathcal{B})$$

## Application to Induced Abstract Data Flow Instances

Replace $f^\sharp_{\mathrm{op}}$ with a simpler monotonic $\overline{h_{\mathrm{op}}}$ verifying

$$f^\sharp_{\mathrm{op}}(\overline{x}) \;\sqsupseteq\; \overline{h_{\mathrm{op}}}(\overline{x}) \qquad \text{(for all } \mathrm{op} \in \mathtt{Op}, \overline{x} \in \overline{L}\text{)}$$

# Correctness of Approximate Transfer Mapping

Consider a data flow instance $\mathcal{A}$ with a set $\mathcal{F}$ of transfer functions and a transfer mapping $f : \mathtt{Op} \to \mathcal{F}$.

For any monotonic function $h : \mathtt{Op} \to \mathcal{F}$ verifying

$$f_{\mathrm{op}}(x) \ \sqsubseteq \ h_{\mathrm{op}}(x) \qquad \text{(for all } \mathrm{op} \in \mathtt{Op}, x \in L\text{)}$$

the data flow instance $\mathcal{B}$ obtained from $\mathcal{A}$ by replacing $f$ with $h$ satisfies:

$$\overleftarrow{\mathrm{MFP}}(\mathcal{A}) \ \sqsubseteq \ \overleftarrow{\mathrm{MFP}}(\mathcal{B}) \qquad \overleftarrow{\mathrm{MOP}}(\mathcal{A}) \ \sqsubseteq \ \overleftarrow{\mathrm{MOP}}(\mathcal{B})$$

## Application to Induced Abstract Data Flow Instances

Replace $f^{\sharp}_{\mathrm{op}}$ with a simpler monotonic $\overline{h_{\mathrm{op}}}$ verifying

$$f^{\sharp}_{\mathrm{op}}(\overline{x}) \ \sqsupseteq \ \overline{h_{\mathrm{op}}}(\overline{x}) \qquad \text{(for all } \mathrm{op} \in \mathtt{Op}, \overline{x} \in L\text{)}$$

# Correctness of Approximate Transfer Mapping

Consider a data flow instance $\mathcal{A}$ with a set $\mathcal{F}$ of transfer functions and a transfer mapping $f : \mathtt{Op} \rightarrow \mathcal{F}$.

For any monotonic function $h : \mathtt{Op} \rightarrow \mathcal{F}$ verifying

$$f_{\mathtt{op}}(x) \sqsubseteq h_{\mathtt{op}}(x) \qquad \text{(for all } \mathtt{op} \in \mathtt{Op}, x \in L\text{)}$$

the data flow instance $\mathcal{B}$ obtained from $\mathcal{A}$ by replacing $f$ with $h$ satisfies:

$$\overleftarrow{\mathrm{MFP}}(\mathcal{A}) \sqsubseteq \overleftarrow{\mathrm{MFP}}(\mathcal{B}) \qquad \overleftarrow{\mathrm{MOP}}(\mathcal{A}) \sqsubseteq \overleftarrow{\mathrm{MOP}}(\mathcal{B})$$

## Application to Induced Abstract Data Flow Instances

Replace $f^{\sharp}_{\mathtt{op}}$ with a simpler monotonic $\overline{h_{\mathtt{op}}}$ verifying

$$f^{\sharp}_{\mathtt{op}}(\overline{x}) \sqsupseteq \overline{h_{\mathtt{op}}}(\overline{x}) \qquad \text{(for all } \mathtt{op} \in \mathtt{Op}, \overline{x} \in L\text{)}$$

# Design of Approximate Transfer Mapping

Given a Galois connection $(\mathcal{P}(\mathtt{X} \to \mathbb{R}), \subseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (\overline{L}, \overline{\sqsubseteq})$ the resulting abstract data flow instance is obtained systematically.

But in practice, $f^\sharp$ is rarely used: an approximate transfer mapping is required.

Tradeoff between computational cost and precision: many possibilities!

**General principle: exploit the structure operations**

1. define an abstract conservative semantics for arithmetic operators and comparators, ideally the most precise one
2. derive inductively an abstract semantics for operations, as usual

# Design of Approximate Transfer Mapping

Given a Galois connection $(\mathcal{P}(\mathbb{X} \to \mathbb{R}), \subseteq) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (\overline{L}, \overline{\sqsubseteq})$ the resulting abstract data flow instance is obtained systematically.

But in practice, $f^\sharp$ is rarely used: an approximate transfer mapping is required.

Tradeoff between computational cost and precision: many possibilities!

## General principle: exploit the structure operations

1. define an abstract conservative semantics for arithmetic operators and comparators, ideally the most precise one
2. derive inductively an abstract semantics for operations, as usual

# Sign Analysis: Abstract Arithmetic Operators

$$(\mathcal{P}(\mathbb{R}), \subseteq) \underset{\alpha_{sign}}{\overset{\gamma_{sign}}{\leftrightarrows}} (Sign, \sqsubseteq)$$

## Extension of Arithmetic Operators to Subsets of $\mathbb{R}$

For each function $* \in \{+, -, \times, \ldots\}$ from $\mathbb{R} \times \mathbb{R}$ to $\mathbb{R}$, define the function $* : (\mathcal{P}(\mathbb{R}) \times \mathcal{P}(\mathbb{R})) \rightarrow \mathcal{P}(\mathbb{R})$ by:

$$U * V \quad = \quad \{u * v \mid u \in U, v \in V\}$$

## Abstract Arithmetic Operators

Define the best abstraction $*^\sharp : (Sign \times Sign) \rightarrow Sign$ of each function $* \in \{+, -, \times, \ldots\}$ by:

$$\overline{x} \; *^\sharp \; \overline{y} \quad = \quad \alpha_{sign}\left(\gamma_{sign}(\overline{x}) * \gamma_{sign}(\overline{y})\right)$$

# Abstract Arithmetic Operators: Table for $+^\sharp$

$$\overline{x} \; +^\sharp \; \overline{y} \quad = \quad \alpha_{sign}\left(\gamma_{sign}(\overline{x}) + \gamma_{sign}(\overline{y})\right)$$

| $+^\sharp$ | $\bot$ | $-$ | $0$ | $+$ | $-0$ | $0+$ | $\top$ |
|---|---|---|---|---|---|---|---|
| $\bot$ | $\bot$ | $\bot$ | $\bot$ | $\bot$ | $\bot$ | $\bot$ | $\bot$ |
| $-$ | $\bot$ | $-$ | $-$ | $\top$ | $-$ | $\top$ | $\top$ |
| $0$ | $\bot$ | $-$ | $0$ | $+$ | $-0$ | $0+$ | $\top$ |
| $+$ | $\bot$ | $\top$ | $+$ | $+$ | $\top$ | $+$ | $\top$ |
| $-0$ | $\bot$ | $-$ | $-0$ | $\top$ | $-0$ | $\top$ | $\top$ |
| $0+$ | $\bot$ | $\top$ | $0+$ | $+$ | $\top$ | $0+$ | $\top$ |
| $\top$ | $\bot$ | $\top$ | $\top$ | $\top$ | $\top$ | $\top$ | $\top$ |

After mechanical inspection of all cases, we derive the above table.

We can derive similar tables for $-^\sharp$ and $\times^\sharp$.

# Sign Analysis: Abstract Semantics of Expressions

$$(\mathcal{P}(\mathrm{X} \to \mathbb{R}), \subseteq) \overset{\gamma_\pi}{\underset{\alpha_\pi}{\leftrightarrows}} (\mathrm{X} \to \mathcal{P}(\mathbb{R}), \subseteq) \overset{\gamma_{sign}}{\underset{\alpha_{sign}}{\leftrightarrows}} (\mathrm{X} \to \mathit{Sign}, \sqsubseteq)$$

For any abstract valuation $\overline{v} : \mathrm{X} \to \mathit{Sign}$, define $\overline{[\![}e\overline{]\!]}_{\overline{v}}$ inductively:

$$\begin{array}{rcll} \overline{[\![}c\overline{]\!]}_{\overline{v}} & = & \alpha_{sign}(\{c\}) & [c \in \mathbb{Q}] \\ \overline{[\![}x\overline{]\!]}_{\overline{v}} & = & \overline{v}(x) & [x \in \mathrm{X}] \\ \overline{[\![}e_1 + e_2\overline{]\!]}_{\overline{v}} & = & \overline{[\![}e_1\overline{]\!]}_{\overline{v}} +^\sharp \overline{[\![}e_2\overline{]\!]}_{\overline{v}} & \\ \overline{[\![}e_1 - e_2\overline{]\!]}_{\overline{v}} & = & \overline{[\![}e_1\overline{]\!]}_{\overline{v}} -^\sharp \overline{[\![}e_2\overline{]\!]}_{\overline{v}} & \\ \overline{[\![}e_1 * e_2\overline{]\!]}_{\overline{v}} & = & \overline{[\![}e_1\overline{]\!]}_{\overline{v}} \times^\sharp \overline{[\![}e_2\overline{]\!]}_{\overline{v}} & \end{array}$$

### Fact (Conservative Approximation)

$$\overline{[\![}e\overline{]\!]}(\overline{v}) \quad \sqsupseteq \quad \alpha_{sign} \circ [\![e]\!] \circ \gamma(\overline{v})$$

# Sign Analysis: Abstract Semantics of Expressions

$$(\mathcal{P}(\mathrm{X} \to \mathbb{R}), \subseteq) \underset{\alpha_\pi}{\overset{\gamma_\pi}{\leftrightarrows}} (\mathrm{X} \to \mathcal{P}(\mathbb{R}), \subseteq) \underset{\alpha_{sign}}{\overset{\gamma_{sign}}{\leftrightarrows}} (\mathrm{X} \to \mathit{Sign}, \sqsubseteq)$$

For any abstract valuation $\overline{v} : \mathrm{X} \to \mathit{Sign}$, define $\overline{\llbracket e \rrbracket}_{\overline{v}}$ inductively:

$$\begin{array}{rcll}
\overline{\llbracket c \rrbracket}_{\overline{v}} & = & \alpha_{sign}(\{c\}) & [c \in \mathbb{Q}] \\
\overline{\llbracket x \rrbracket}_{\overline{v}} & = & \overline{v}(x) & [x \in \mathrm{X}] \\
\overline{\llbracket e_1 + e_2 \rrbracket}_{\overline{v}} & = & \overline{\llbracket e_1 \rrbracket}_{\overline{v}} +^\sharp \overline{\llbracket e_2 \rrbracket}_{\overline{v}} & \\
\overline{\llbracket e_1 - e_2 \rrbracket}_{\overline{v}} & = & \overline{\llbracket e_1 \rrbracket}_{\overline{v}} -^\sharp \overline{\llbracket e_2 \rrbracket}_{\overline{v}} & \\
\overline{\llbracket e_1 * e_2 \rrbracket}_{\overline{v}} & = & \overline{\llbracket e_1 \rrbracket}_{\overline{v}} \times^\sharp \overline{\llbracket e_2 \rrbracket}_{\overline{v}} &
\end{array}$$

## Fact (Conservative Approximation)

$$\overline{\llbracket e \rrbracket}(\overline{v}) \quad \sqsupseteq \quad \alpha_{sign} \circ \llbracket e \rrbracket \circ \gamma(\overline{v})$$

# Sign Analysis: Abstract Arithmetic Comparators

$$(\mathcal{P}(\mathbb{R}), \subseteq) \mathrel{\substack{\xleftarrow{\gamma_{sign}} \\ \xrightarrow[\alpha_{sign}]{}}} (\mathit{Sign}, \overline{\sqsubseteq})$$

## Extension of Arithmetic Comparators to Subsets of $\mathbb{R}$

For each binary relation $\bowtie \in \{<, \leq, =, \neq, >, \geq, \ldots\}$ on $\mathbb{R}$, define the binary relation $\bowtie$ on $\mathcal{P}(\mathbb{R})$ by:

$$U \bowtie V \quad \text{if} \quad u \bowtie v \ \text{ for some } u \in U \text{ and } v \in V$$

## Abstract Arithmetic Comparators

Define the best abstraction $\bowtie^\sharp \subseteq \mathit{Sign} \times \mathit{Sign}$ of each binary relation $\bowtie \in \{<, \leq, =, \neq, >, \geq, \ldots\}$ by:

$$\overline{x} \bowtie^\sharp \overline{y} \quad \text{if} \quad \gamma_{sign}(\overline{x}) \ \bowtie \ \gamma_{sign}(\overline{y})$$

# Abstract Arithmetic Comparators: Table for $<^\sharp$

$$\overline{x} <^\sharp \overline{y} \quad \text{if} \quad \gamma_{sign}(\overline{x}) < \gamma_{sign}(\overline{y})$$

| $<^\sharp$ | $\bot$ | $-$ | $0$ | $+$ | $-0$ | $0+$ | $\top$ |
|---|---|---|---|---|---|---|---|
| $\bot$ | | | | | | | |
| $-$ | | ● | ● | ● | ● | ● | ● |
| $0$ | | | | ● | | ● | ● |
| $+$ | | | | ● | | ● | ● |
| $-0$ | | ● | ● | ● | ● | ● | ● |
| $0+$ | | | | ● | | ● | ● |
| $\top$ | | ● | ● | ● | ● | ● | ● |

After mechanical inspection of all cases, we derive the above table.

We can derive similar tables for $\leq^\sharp$, $=^\sharp$, $\neq^\sharp$, $>^\sharp$, and $\geq^\sharp$.

# Sign Analysis: Abstract Semantics of Guards

$$(\mathcal{P}(\mathrm{X} \to \mathbb{R}), \subseteq) \mathrel{\underset{\alpha_\pi}{\overset{\gamma_\pi}{\leftrightarrows}}} (\mathrm{X} \to \mathcal{P}(\mathbb{R}), \subseteq) \mathrel{\underset{\alpha_{sign}}{\overset{\gamma_{sign}}{\leftrightarrows}}} (\mathrm{X} \to \mathit{Sign}, \sqsubseteq)$$

For any abstract valuation $\overline{v} : \mathrm{X} \to \mathit{Sign}$, define $\overline{v} \models g$ inductively:

$$\begin{array}{lllll}
\overline{v} & \models & e_1 < e_2 & \mathit{if} & [\![e_1]\!]_{\overline{v}} \;<^\sharp\; [\![e_2]\!]_{\overline{v}} \\
\overline{v} & \models & e_1 \leq e_2 & \mathit{if} & [\![e_1]\!]_{\overline{v}} \;\leq^\sharp\; [\![e_2]\!]_{\overline{v}} \\
\overline{v} & \models & e_1 = e_2 & \mathit{if} & [\![e_1]\!]_{\overline{v}} \;=^\sharp\; [\![e_2]\!]_{\overline{v}} \\
\overline{v} & \models & e_1 \neq e_2 & \mathit{if} & [\![e_1]\!]_{\overline{v}} \;\neq^\sharp\; [\![e_2]\!]_{\overline{v}} \\
\overline{v} & \models & e_1 \geq e_2 & \mathit{if} & [\![e_1]\!]_{\overline{v}} \;\geq^\sharp\; [\![e_2]\!]_{\overline{v}} \\
\overline{v} & \models & e_1 > e_2 & \mathit{if} & [\![e_1]\!]_{\overline{v}} \;>^\sharp\; [\![e_2]\!]_{\overline{v}}
\end{array}$$

## Fact (Conservative Approximation)

$$\mathit{if} \quad v \models g \quad \mathit{for\ some} \quad v \in \gamma(\overline{v}) \qquad \mathit{then} \qquad \overline{v} \models g$$

# Sign Analysis: Abstract Semantics of Guards

$$(\mathcal{P}(\mathrm{X} \to \mathbb{R}), \subseteq) \underset{\alpha_\pi}{\overset{\gamma_\pi}{\leftrightarrows}} (\mathrm{X} \to \mathcal{P}(\mathbb{R}), \subseteq) \underset{\alpha_{sign}}{\overset{\gamma_{sign}}{\leftrightarrows}} (\mathrm{X} \to \mathit{Sign}, \sqsubseteq)$$

For any abstract valuation $\overline{v} : \mathrm{X} \to \mathit{Sign}$, define $\overline{v} \models g$ inductively:

$$\begin{array}{lllll}
\overline{v} & \models & e_1 < e_2 & \textit{if} & [\![e_1]\!]_{\overline{v}} <^\sharp [\![e_2]\!]_{\overline{v}} \\
\overline{v} & \models & e_1 \leq e_2 & \textit{if} & [\![e_1]\!]_{\overline{v}} \leq^\sharp [\![e_2]\!]_{\overline{v}} \\
\overline{v} & \models & e_1 = e_2 & \textit{if} & [\![e_1]\!]_{\overline{v}} =^\sharp [\![e_2]\!]_{\overline{v}} \\
\overline{v} & \models & e_1 \neq e_2 & \textit{if} & [\![e_1]\!]_{\overline{v}} \neq^\sharp [\![e_2]\!]_{\overline{v}} \\
\overline{v} & \models & e_1 \geq e_2 & \textit{if} & [\![e_1]\!]_{\overline{v}} \geq^\sharp [\![e_2]\!]_{\overline{v}} \\
\overline{v} & \models & e_1 > e_2 & \textit{if} & [\![e_1]\!]_{\overline{v}} >^\sharp [\![e_2]\!]_{\overline{v}}
\end{array}$$

## Fact (Conservative Approximation)

$$\textit{if} \;\; v \models g \;\; \textit{for some} \;\; v \in \gamma(\overline{v}) \qquad \textit{then} \qquad \overline{v} \models g$$

# Forward Sign Analysis: Approximate Transfer Mapping

$$(\mathcal{P}(\mathrm{X} \to \mathbb{R}), \subseteq) \underset{\alpha_\pi}{\overset{\gamma_\pi}{\leftrightarrows}} (\mathrm{X} \to \mathcal{P}(\mathbb{R}), \subseteq) \underset{\alpha_{sign}}{\overset{\gamma_{sign}}{\leftrightarrows}} (\mathrm{X} \to \mathit{Sign}, \overline{\sqsubseteq})$$

$$\overline{h_{x:=e}}(\overline{v}) = \lambda y \,.\begin{cases} \overline{v}(y) & \text{if } y \neq x \\ \overline{[\![e]\!]}_{\overline{v}} & \text{if } y = x \end{cases} \qquad \overline{h_g}(\overline{v}) = \begin{cases} \overline{\bot} & \text{if } \overline{v} \not\models g \\ \overline{v} & \text{if } \overline{v} \models g \end{cases}$$

## Fact (Conservative Approximation)

$$f^\sharp_{\mathrm{op}}(\overline{x}) \quad \overline{\sqsupseteq} \quad \overline{h_{\mathrm{op}}}(\overline{x}) \qquad (\textit{for all } \mathrm{op} \in \mathrm{Op}, \overline{x} \in \mathrm{X} \to \mathit{Sign})$$

## Vs Transfer Mapping Previously Designed by Hand

- ☺ guaranteed to lead to a correct data flow analysis
- ☺ more precise since the previous one was the identity on guards

# Forward Sign Analysis: Approximate Transfer Mapping

$$(\mathcal{P}(\mathrm{X} \to \mathbb{R}), \subseteq) \mathrel{\substack{\xleftarrow{\ \gamma_\pi\ } \\ \xrightarrow[\ \alpha_\pi\ ]{}}} (\mathrm{X} \to \mathcal{P}(\mathbb{R}), \subseteq) \mathrel{\substack{\xleftarrow{\ \gamma_{sign}\ } \\ \xrightarrow[\ \alpha_{sign}\ ]{}}} (\mathrm{X} \to \mathit{Sign}, \overline{\sqsubseteq})$$

$$\overline{h_{x:=e}}(\overline{v}) \quad = \quad \lambda y \,.\, \begin{cases} \overline{v}(y) & \text{if } y \neq x \\ \overline{[\![}e\overline{]\!]}_{\overline{v}} & \text{if } y = x \end{cases} \qquad\qquad \overline{h_g}(\overline{v}) \quad = \quad \begin{cases} \overline{\bot} & \text{if } \overline{v} \not\models g \\ \overline{v} & \text{if } \overline{v} \models g \end{cases}$$

## Fact (Conservative Approximation)

$$f^\sharp_{\mathrm{op}}(\overline{x}) \quad \overline{\sqsupseteq} \quad \overline{h_{\mathrm{op}}}(\overline{x}) \qquad\qquad (\textit{for all } \mathrm{op} \in \mathrm{Op}, \overline{x} \in \mathrm{X} \to \mathit{Sign})$$

## Vs Transfer Mapping Previously Designed by Hand

- ☺ guaranteed to lead to a correct data flow analysis
- ☺ more precise since the previous one was the identity on guards

# Forward Sign Analysis: Approximate Transfer Mapping

$$(\mathcal{P}(\mathrm{X} \to \mathbb{R}), \subseteq) \underset{\alpha_\pi}{\overset{\gamma_\pi}{\leftrightarrows}} (\mathrm{X} \to \mathcal{P}(\mathbb{R}), \subseteq) \underset{\alpha_{sign}}{\overset{\gamma_{sign}}{\leftrightarrows}} (\mathrm{X} \to Sign, \overline{\sqsubseteq})$$

$$\overline{h_{x:=e}}(\overline{v}) \;=\; \lambda y\,.\begin{cases} \overline{v}(y) & \text{if } y \neq x \\ \overline{[\![e]\!]}_{\overline{v}} & \text{if } y = x \end{cases} \qquad\qquad \overline{h_g}(\overline{v}) \;=\; \begin{cases} \overline{\bot} & \text{if } \overline{v} \not\models g \\ \overline{v} & \text{if } \overline{v} \models g \end{cases}$$

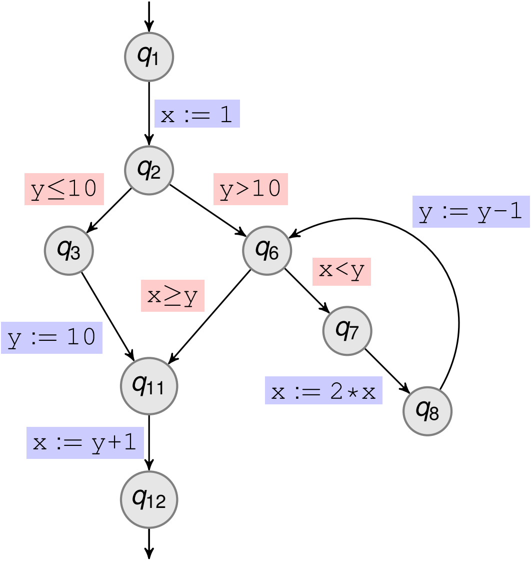
## Fact (Conservative Approximation)

$$f^\sharp_{\mathrm{op}}(\overline{x}) \quad \overline{\sqsupseteq} \quad \overline{h_{\mathrm{op}}}(\overline{x}) \qquad\qquad (\textit{for all } \mathrm{op} \in \mathrm{Op}, \overline{x} \in \mathrm{X} \to Sign)$$

## Vs Transfer Mapping Previously Designed by Hand

- ☺ guaranteed to lead to a correct data flow analysis
- ☺ more precise since the previous one was the identity on guards

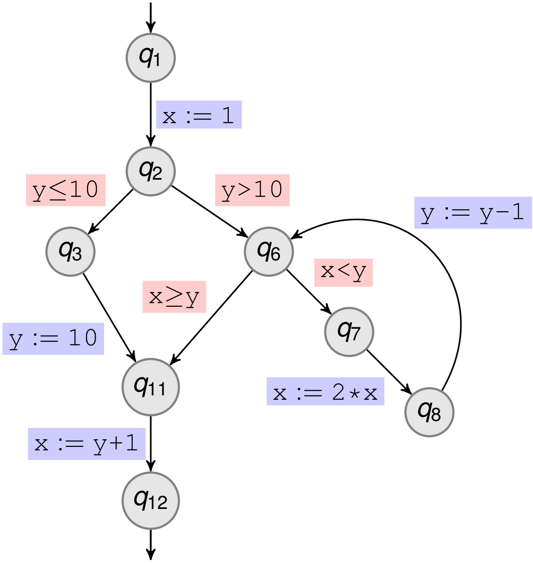
# Forward Sign Analysis on Running Example



**Goal**

Show that $\mathtt{x} > 0$ at $q_{12}$

# Forward Sign Analysis on Running Example

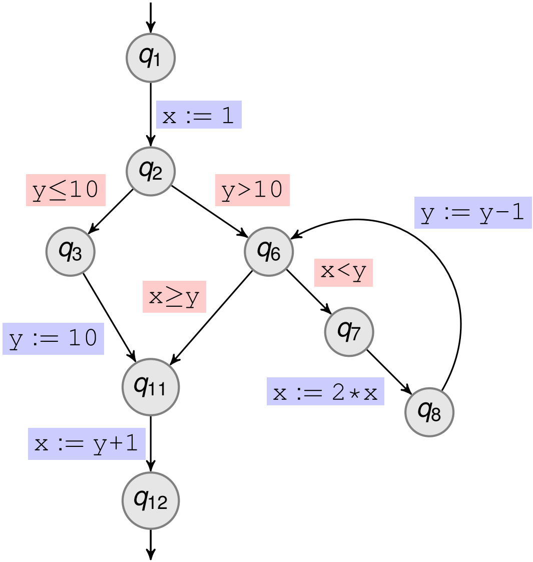


**Goal**

Show that x > 0 at $q_{12}$

| | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $\bot$ | $\bot$ |
| $q_3$ | $\bot$ | $\bot$ |
| $q_6$ | $\bot$ | $\bot$ |
| $q_7$ | $\bot$ | $\bot$ |
| $q_8$ | $\bot$ | $\bot$ |
| $q_{11}$ | $\bot$ | $\bot$ |
| $q_{12}$ | $\bot$ | $\bot$ |

# Forward Sign Analysis on Running Example

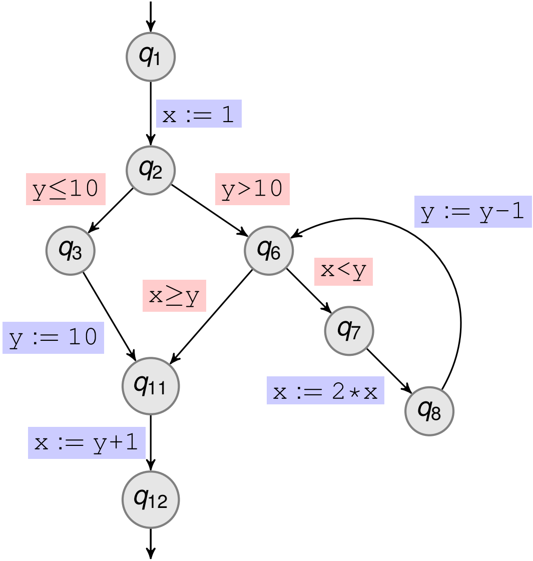


**Goal**

Show that $x > 0$ at $q_{12}$

| | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $\bot$ | $\bot$ |
| $q_6$ | $\bot$ | $\bot$ |
| $q_7$ | $\bot$ | $\bot$ |
| $q_8$ | $\bot$ | $\bot$ |
| $q_{11}$ | $\bot$ | $\bot$ |
| $q_{12}$ | $\bot$ | $\bot$ |

# Forward Sign Analysis on Running Example

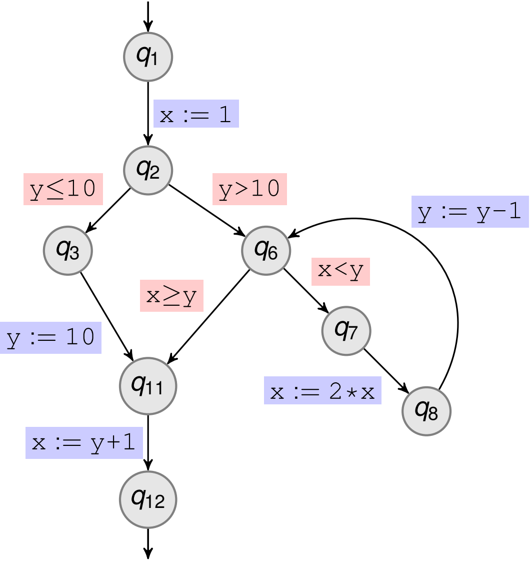


**Goal**

Show that $x > 0$ at $q_{12}$

|  | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $\bot$ | $\bot$ |
| $q_6$ | $\bot$ | $\bot$ |
| $q_7$ | $\bot$ | $\bot$ |
| $q_8$ | $\bot$ | $\bot$ |
| $q_{11}$ | $\bot$ | $\bot$ |
| $q_{12}$ | $\bot$ | $\bot$ |

# Forward Sign Analysis on Running Example

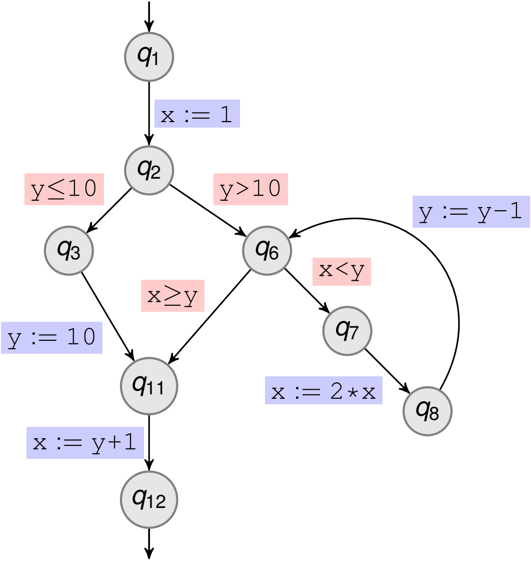


**Goal**

Show that $x > 0$ at $q_{12}$

| | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $\bot$ | $\bot$ |
| $q_7$ | $\bot$ | $\bot$ |
| $q_8$ | $\bot$ | $\bot$ |
| $q_{11}$ | $\bot$ | $\bot$ |
| $q_{12}$ | $\bot$ | $\bot$ |

# Forward Sign Analysis on Running Example

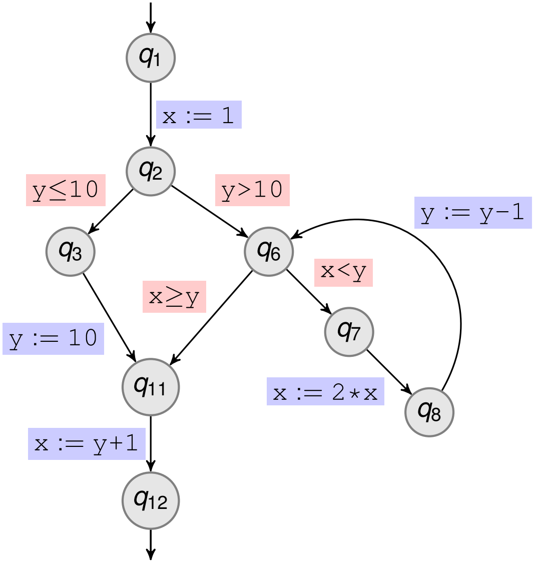


**Goal**

Show that $x > 0$ at $q_{12}$

|  | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $\perp$ | $\perp$ |
| $q_7$ | $\perp$ | $\perp$ |
| $q_8$ | $\perp$ | $\perp$ |
| $q_{11}$ | $\perp$ | $\perp$ |
| $q_{12}$ | $\perp$ | $\perp$ |

# Forward Sign Analysis on Running Example

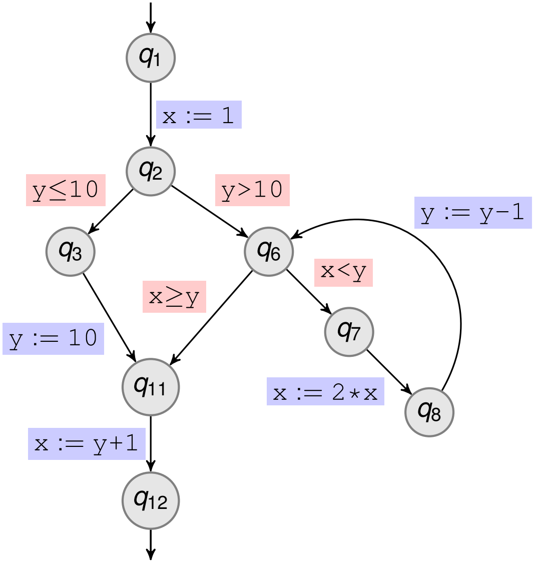


**Goal**

Show that $\mathtt{x} > 0$ at $q_{12}$

| | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $\bot$ | $\bot$ |
| $q_7$ | $\bot$ | $\bot$ |
| $q_8$ | $\bot$ | $\bot$ |
| $q_{11}$ | $+$ | $+$ |
| $q_{12}$ | $\bot$ | $\bot$ |

# Forward Sign Analysis on Running Example

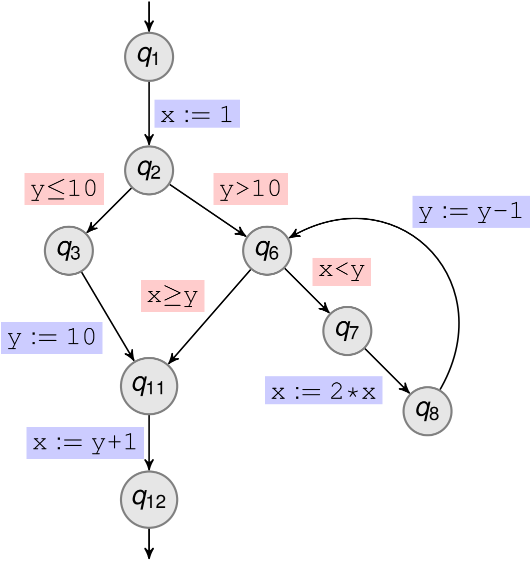


**Goal**

Show that $x > 0$ at $q_{12}$

|  | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $\bot$ | $\bot$ |
| $q_7$ | $\bot$ | $\bot$ |
| $q_8$ | $\bot$ | $\bot$ |
| $q_{11}$ | $+$ | $+$ |
| $q_{12}$ | $\bot$ | $\bot$ |

# Forward Sign Analysis on Running Example

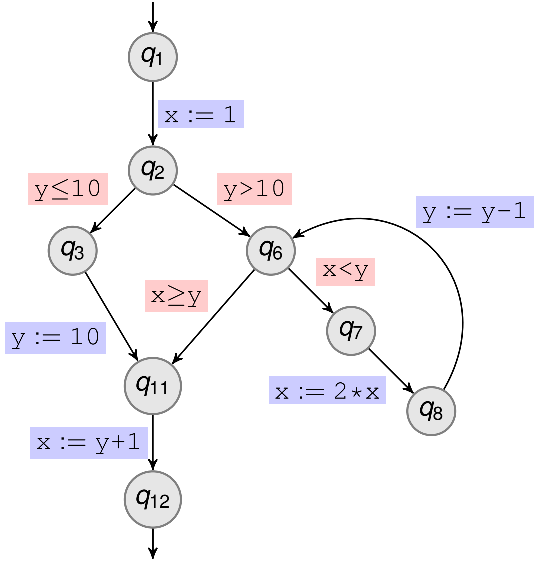


**Goal**

Show that $x > 0$ at $q_{12}$

| | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $\perp$ | $\perp$ |
| $q_7$ | $\perp$ | $\perp$ |
| $q_8$ | $\perp$ | $\perp$ |
| $q_{11}$ | $+$ | $+$ |
| $q_{12}$ | $+$ | $+$ |

# Forward Sign Analysis on Running Example

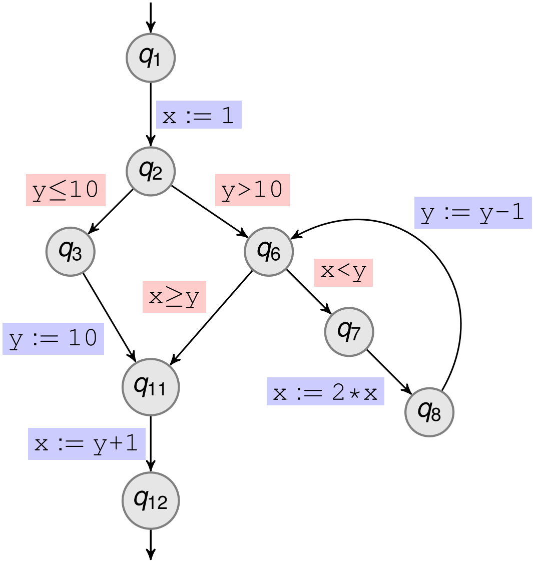


**Goal**

Show that $x > 0$ at $q_{12}$

| | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $\bot$ | $\bot$ |
| $q_7$ | $\bot$ | $\bot$ |
| $q_8$ | $\bot$ | $\bot$ |
| $q_{11}$ | $+$ | $+$ |
| $q_{12}$ | $+$ | $+$ |

# Forward Sign Analysis on Running Example

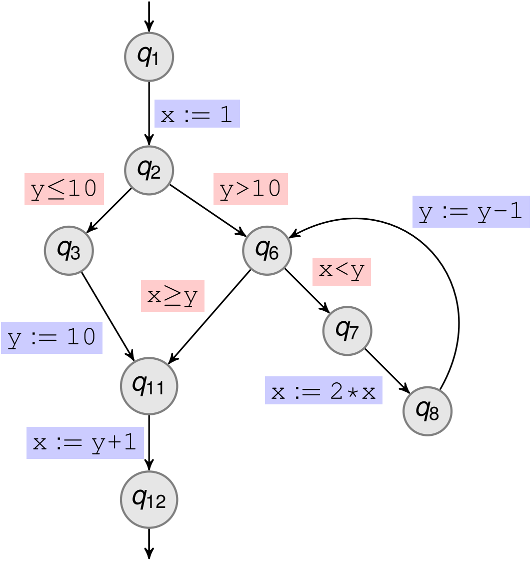


**Goal**

Show that $\texttt{x} > 0$ at $q_{12}$

| | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $+$ | $\top$ |
| $q_7$ | $\bot$ | $\bot$ |
| $q_8$ | $\bot$ | $\bot$ |
| $q_{11}$ | $+$ | $+$ |
| $q_{12}$ | $+$ | $+$ |

# Forward Sign Analysis on Running Example

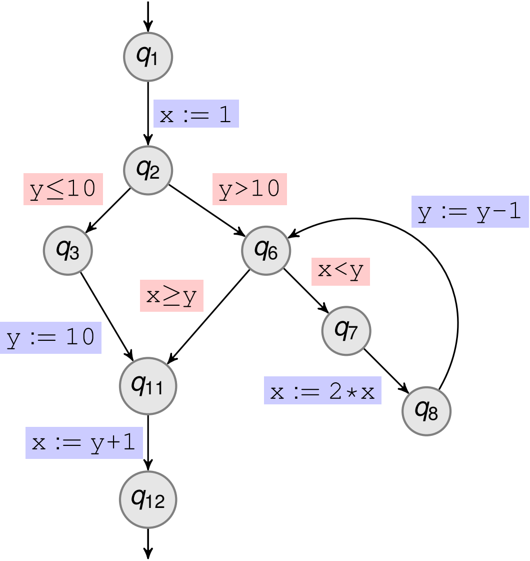


**Goal**

Show that x > 0 at $q_{12}$

|  | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $+$ | $\top$ |
| $q_7$ | $\bot$ | $\bot$ |
| $q_8$ | $\bot$ | $\bot$ |
| $q_{11}$ | $+$ | $+$ |
| $q_{12}$ | $+$ | $+$ |

# Forward Sign Analysis on Running Example

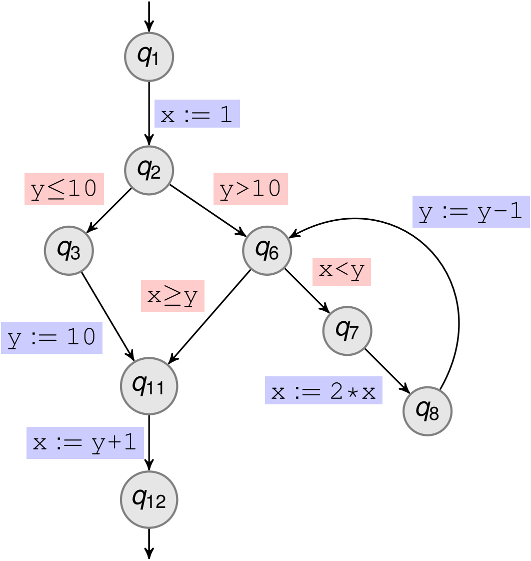


**Goal**

Show that $x > 0$ at $q_{12}$

| | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $+$ | $\top$ |
| $q_7$ | $+$ | $\top$ |
| $q_8$ | $\bot$ | $\bot$ |
| $q_{11}$ | $+$ | $+$ |
| $q_{12}$ | $+$ | $+$ |

# Forward Sign Analysis on Running Example

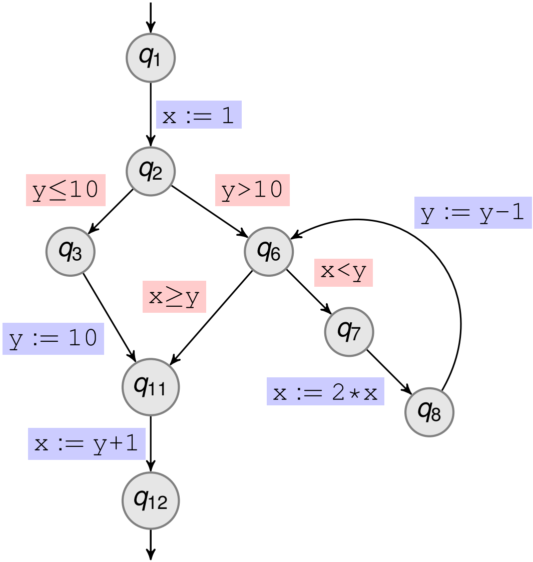


**Goal**

Show that $\mathtt{x} > 0$ at $q_{12}$

| | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $+$ | $\top$ |
| $q_7$ | $+$ | $\top$ |
| $q_8$ | $\bot$ | $\bot$ |
| $q_{11}$ | $+$ | $+$ |
| $q_{12}$ | $+$ | $+$ |

# Forward Sign Analysis on Running Example

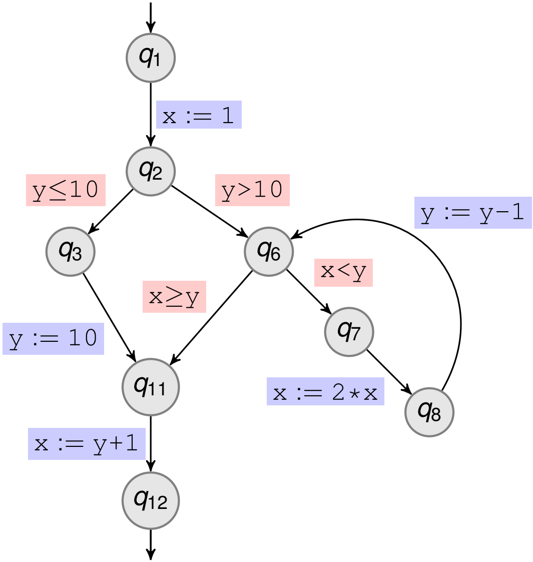


**Goal**

Show that $\mathtt{x} > 0$ at $q_{12}$

| | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $+$ | $\top$ |
| $q_7$ | $+$ | $\top$ |
| $q_8$ | $+$ | $\top$ |
| $q_{11}$ | $+$ | $+$ |
| $q_{12}$ | $+$ | $+$ |

# Forward Sign Analysis on Running Example

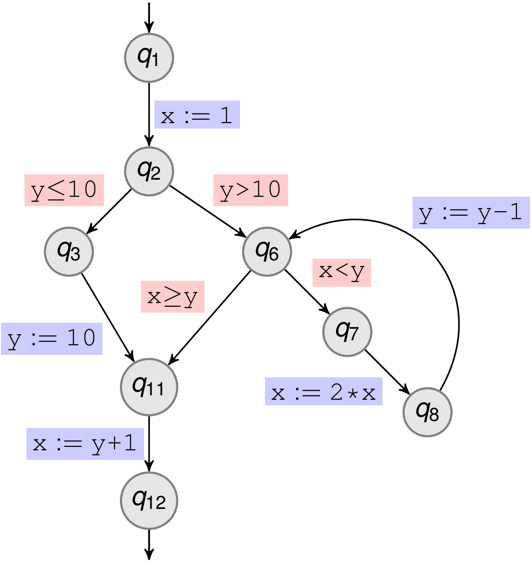


**Goal**

Show that $\mathtt{x} > 0$ at $q_{12}$

| | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $+$ | $\top$ |
| $q_7$ | $+$ | $\top$ |
| $q_8$ | $+$ | $\top$ |
| $q_{11}$ | $+$ | $+$ |
| $q_{12}$ | $+$ | $+$ |

# Forward Sign Analysis on Running Example

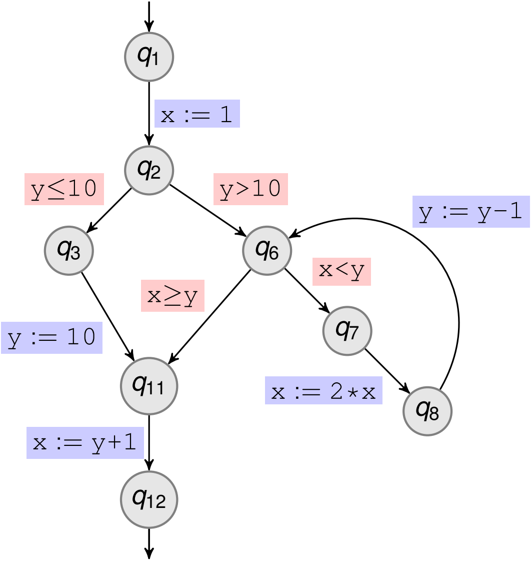


**Goal**

Show that $x > 0$ at $q_{12}$

| | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $+$ | $\top$ |
| $q_7$ | $+$ | $\top$ |
| $q_8$ | $+$ | $\top$ |
| $q_{11}$ | $+$ | $\top$ |
| $q_{12}$ | $+$ | $+$ |

# Forward Sign Analysis on Running Example

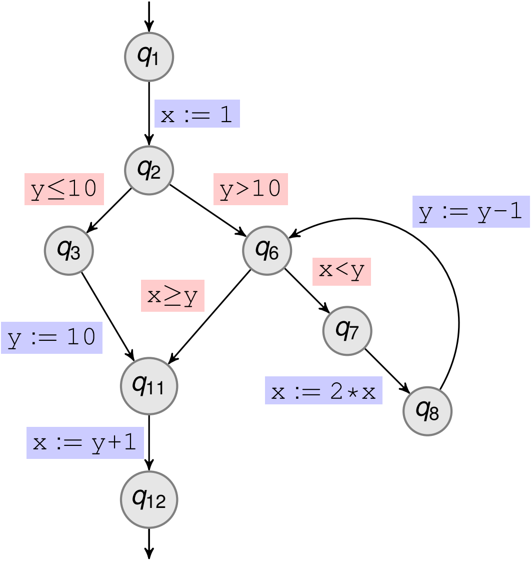


**Goal**

Show that $x > 0$ at $q_{12}$

| | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $+$ | $\top$ |
| $q_7$ | $+$ | $\top$ |
| $q_8$ | $+$ | $\top$ |
| $q_{11}$ | $+$ | $\top$ |
| $q_{12}$ | $+$ | $+$ |

# Forward Sign Analysis on Running Example

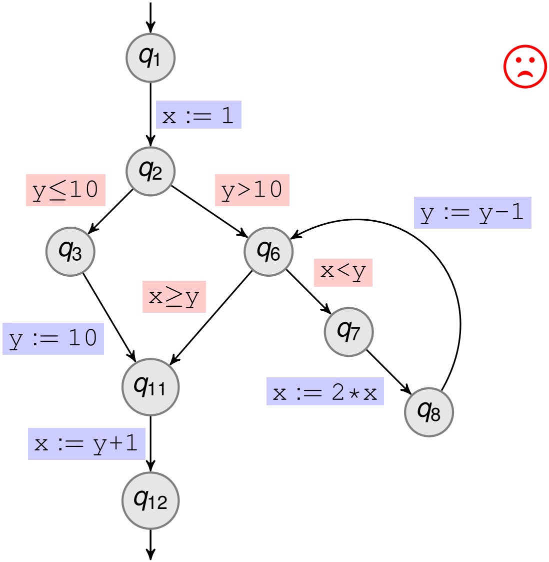


**Goal**

Show that $\mathtt{x} > 0$ at $q_{12}$

| | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $+$ | $\top$ |
| $q_7$ | $+$ | $\top$ |
| $q_8$ | $+$ | $\top$ |
| $q_{11}$ | $+$ | $\top$ |
| $q_{12}$ | $\top$ | $\top$ |

# Forward Sign Analysis on Running Example



☹

**Goal**

Show that $\mathtt{x} > 0$ at $q_{12}$

|  | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $+$ | $\top$ |
| $q_7$ | $+$ | $\top$ |
| $q_8$ | $+$ | $\top$ |
| $q_{11}$ | $+$ | $\top$ |
| $q_{12}$ | $\top$ | $\top$ |

# Loss of Precision with Approximate Transfer Mapping

## Example (Assignment $\mathtt{op} \;=\; \mathtt{x} := \mathtt{z} * \mathtt{z}$)

$$f^{\sharp}_{\mathrm{op}}(\overline{\top}) \;=\; \lambda y\,.\begin{cases}\overline{\top} & \text{if } y \neq \mathtt{x}\\ 0+ & \text{if } y = \mathtt{x}\end{cases} \qquad\qquad \overline{h_{\mathrm{op}}}(\overline{\top}) \;=\; \lambda y\,.\begin{cases}\overline{\top} & \text{if } y \neq \mathtt{x}\\ \overline{\top} & \text{if } y = \mathtt{x}\end{cases}$$

Indeed with $\overline{h_{\mathrm{op}}}$ the new value for $\mathtt{x}$ is: $\overline{[\![\mathtt{z} * \mathtt{z}]\!]}_{\overline{\top}} \;=\; \overline{[\![\mathtt{z}]\!]}_{\overline{\top}} \times^{\sharp} \overline{[\![\mathtt{z}]\!]}_{\overline{\top}} \;=\; \overline{\top}$.

## Example (Guard $\mathtt{op} \;=\; \mathtt{x} = 0$)

$$f^{\sharp}_{\mathrm{op}}(\overline{\top}) \;=\; \lambda y\,.\begin{cases}\overline{\top} & \text{if } y \neq \mathtt{x}\\ 0 & \text{if } y = \mathtt{x}\end{cases} \qquad\qquad \overline{h_{\mathrm{op}}}(\overline{\top}) \;=\; \lambda y\,.\begin{cases}\overline{\top} & \text{if } y \neq \mathtt{x}\\ \overline{\top} & \text{if } y = \mathtt{x}\end{cases}$$

Indeed $\overline{h_{\mathrm{op}}}(\overline{v})$ is either $\overline{\bot}$ (if $\overline{v} \not\models g$) or $\overline{v}$.

## Example (Guard $\mathtt{op} \;=\; \mathtt{x} > \mathtt{x}$)

$$f^{\sharp}_{\mathrm{op}}(\overline{\top}) \;=\; \overline{\bot} \qquad\qquad \overline{h_{\mathrm{op}}}(\overline{\top}) \;=\; \overline{\top}$$

# Loss of Precision with Approximate Transfer Mapping

## Example (Assignment $\mathrm{op} \ = \ \mathrm{x} := \mathrm{z} * \mathrm{z}$)

$$f^\sharp_{\mathrm{op}}(\overline{\top}) \ = \ \lambda y \, . \begin{cases} \overline{\top} & \text{if } y \neq \mathrm{x} \\ 0+ & \text{if } y = \mathrm{x} \end{cases} \qquad \overline{h_{\mathrm{op}}}(\overline{\top}) \ = \ \lambda y \, . \begin{cases} \overline{\top} & \text{if } y \neq \mathrm{x} \\ \overline{\top} & \text{if } y = \mathrm{x} \end{cases}$$

Indeed with $\overline{h_{\mathrm{op}}}$ the new value for $\mathrm{x}$ is: $\overline{[\![\mathrm{z} * \mathrm{z}]\!]}_{\overline{\top}} \ = \ \overline{[\![\mathrm{z}]\!]}_{\overline{\top}} \times^\sharp \overline{[\![\mathrm{z}]\!]}_{\overline{\top}} \ = \ \overline{\top}$.

## Example (Guard $\mathrm{op} \ = \ \mathrm{x} = 0$)

$$f^\sharp_{\mathrm{op}}(\overline{\top}) \ = \ \lambda y \, . \begin{cases} \overline{\top} & \text{if } y \neq \mathrm{x} \\ 0 & \text{if } y = \mathrm{x} \end{cases} \qquad \overline{h_{\mathrm{op}}}(\overline{\top}) \ = \ \lambda y \, . \begin{cases} \overline{\top} & \text{if } y \neq \mathrm{x} \\ \overline{\top} & \text{if } y = \mathrm{x} \end{cases}$$

Indeed $\overline{h_{\mathrm{op}}}(\overline{v})$ is either $\overline{\bot}$ (if $\overline{v} \not\models g$) or $\overline{v}$.

## Example (Guard $\mathrm{op} \ = \ \mathrm{x} > \mathrm{x}$)

$$f^\sharp_{\mathrm{op}}(\overline{\top}) \ = \ \overline{\bot} \qquad \overline{h_{\mathrm{op}}}(\overline{\top}) \ = \ \overline{\top}$$

# Loss of Precision with Approximate Transfer Mapping

## Example (Assignment $\mathrm{op} = \mathrm{x} := \mathrm{z} \ast \mathrm{z}$)

$$f^\sharp_{\mathrm{op}}(\overline{\top}) = \lambda y\,.\begin{cases} \overline{\top} & \text{if } y \neq \mathrm{x} \\ 0+ & \text{if } y = \mathrm{x} \end{cases} \qquad \overline{h_{\mathrm{op}}}(\overline{\top}) = \lambda y\,.\begin{cases} \overline{\top} & \text{if } y \neq \mathrm{x} \\ \overline{\top} & \text{if } y = \mathrm{x} \end{cases}$$

Indeed with $\overline{h_{\mathrm{op}}}$ the new value for $\mathrm{x}$ is: $\overline{[\![}\mathrm{z} \ast \mathrm{z}\overline{]\!]}_{\overline{\top}} = \overline{[\![}\mathrm{z}\overline{]\!]}_{\overline{\top}} \times^\sharp \overline{[\![}\mathrm{z}\overline{]\!]}_{\overline{\top}} = \overline{\top}$.

## Example (Guard $\mathrm{op} = \mathrm{x} = 0$)

$$f^\sharp_{\mathrm{op}}(\overline{\top}) = \lambda y\,.\begin{cases} \overline{\top} & \text{if } y \neq \mathrm{x} \\ 0 & \text{if } y = \mathrm{x} \end{cases} \qquad \overline{h_{\mathrm{op}}}(\overline{\top}) = \lambda y\,.\begin{cases} \overline{\top} & \text{if } y \neq \mathrm{x} \\ \overline{\top} & \text{if } y = \mathrm{x} \end{cases}$$

Indeed $\overline{h_{\mathrm{op}}}(\overline{v})$ is either $\overline{\bot}$ (if $\overline{v} \not\models g$) or $\overline{v}$.

## Example (Guard $\mathrm{op} = \mathrm{x} > \mathrm{x}$)

$$f^\sharp_{\mathrm{op}}(\overline{\top}) = \overline{\bot} \qquad \overline{h_{\mathrm{op}}}(\overline{\top}) = \overline{\top}$$

# Enhanced Precision with Functional Comparators

Gain information from guards

## Functional Extension of Arithmetic Comparators to Subsets of $\mathbb{R}$

For each binary relation $\bowtie \in \{<, \leq, =, \neq, >, \geq, \ldots\}$ on $\mathbb{R}$, define the function $\bowtie : (\mathcal{P}(\mathbb{R}) \times \mathcal{P}(\mathbb{R})) \to (\mathcal{P}(\mathbb{R}) \times \mathcal{P}(\mathbb{R}))$ by:

$$U \bowtie V \quad = \quad (\{u \in U \mid \exists v \in V, u \bowtie v\}, \{v \in V \mid \exists u \in U, u \bowtie v\})$$

## Functional Abstract Arithmetic Comparators

Define the best abstraction $\bowtie^\sharp : (\mathit{Sign} \times \mathit{Sign}) \to (\mathit{Sign} \times \mathit{Sign})$ of each function $\bowtie \in \{<, \leq, =, \neq, >, \geq, \ldots\}$ by:

$$\overline{x} \bowtie^\sharp \overline{y} \quad = \quad (\alpha_{sign}(U), \alpha_{sign}(V))$$

$$\text{where} \qquad (U, V) \quad = \quad \gamma_{sign}(\overline{x}) \bowtie \gamma_{sign}(\overline{y})$$

# Enhanced Precision with Functional Comparators

Gain information from guards

## Functional Extension of Arithmetic Comparators to Subsets of $\mathbb{R}$

For each binary relation $\bowtie \in \{<, \leq, =, \neq, >, \geq, \ldots\}$ on $\mathbb{R}$, define the function $\bowtie : (\mathcal{P}(\mathbb{R}) \times \mathcal{P}(\mathbb{R})) \to (\mathcal{P}(\mathbb{R}) \times \mathcal{P}(\mathbb{R}))$ by:

$$U \bowtie V \;=\; (\{u \in U \mid \exists v \in V, u \bowtie v\}, \{v \in V \mid \exists u \in U, u \bowtie v\})$$

## Functional Abstract Arithmetic Comparators

Define the best abstraction $\bowtie^\sharp : (Sign \times Sign) \to (Sign \times Sign)$ of each function $\bowtie \in \{<, \leq, =, \neq, >, \geq, \ldots\}$ by:

$$\begin{array}{rrcl} & \overline{x} \bowtie^\sharp \overline{y} & = & (\alpha_{sign}(U), \alpha_{sign}(V)) \\ \text{where} & (U, V) & = & \gamma_{sign}(\overline{x}) \bowtie \gamma_{sign}(\overline{y}) \end{array}$$

# Enhanced Precision with Functional Comparators

Gain information from guards

## Functional Extension of Arithmetic Comparators to Subsets of $\mathbb{R}$

For each binary relation $\bowtie \in \{<, \leq, =, \neq, >, \geq, \ldots\}$ on $\mathbb{R}$, define the function $\bowtie : (\mathcal{P}(\mathbb{R}) \times \mathcal{P}(\mathbb{R})) \to (\mathcal{P}(\mathbb{R}) \times \mathcal{P}(\mathbb{R}))$ by:

$$U \bowtie V \quad = \quad (\{u \in U \mid \exists v \in V, u \bowtie v\}, \{v \in V \mid \exists u \in U, u \bowtie v\})$$

## Functional Abstract Arithmetic Comparators

Define the best abstraction $\bowtie^\sharp : (\mathit{Sign} \times \mathit{Sign}) \to (\mathit{Sign} \times \mathit{Sign})$ of each function $\bowtie \in \{<, \leq, =, \neq, >, \geq, \ldots\}$ by:

$$\begin{array}{lrcl} & \overline{x} \bowtie^\sharp \overline{y} & = & (\alpha_{sign}(U), \alpha_{sign}(V)) \\ \text{where} & (U, V) & = & \gamma_{sign}(\overline{x}) \bowtie \gamma_{sign}(\overline{y}) \end{array}$$

# Functional Abstract Comparators: Table for $\leq^\sharp$

$$\begin{array}{lrcl} & \overline{x} \bowtie^\sharp \overline{y} & = & (\alpha_{sign}(U), \alpha_{sign}(V)) \\ \text{where} & (U, V) & = & \gamma_{sign}(\overline{x}) \bowtie \gamma_{sign}(\overline{y}) \end{array}$$

| $\leq^\sharp$ | $\bot$ | $-$ | $0$ | $+$ | $-0$ | $0+$ | $\top$ |
|---|---|---|---|---|---|---|---|
| $\bot$ | $(\bot, \bot)$ | $(\bot, \bot)$ | $(\bot, \bot)$ | $(\bot, \bot)$ | $(\bot, \bot)$ | $(\bot, \bot)$ | $(\bot, \bot)$ |
| $-$ | $(\bot, \bot)$ | $(-, -)$ | $(-, 0)$ | $(-, +)$ | $(-, -0)$ | $(-, 0+)$ | $(-, \top)$ |
| $0$ | $(\bot, \bot)$ | $(\bot, \bot)$ | $(0, 0)$ | $(0, +)$ | $(0, 0)$ | $(0, 0+)$ | $(0, \top)$ |
| $+$ | $(\bot, \bot)$ | $(\bot, \bot)$ | $(\bot, \bot)$ | $(+, +)$ | $(\bot, \bot)$ | $(+, 0+)$ | $(+, \top)$ |
| $-0$ | $(\bot, \bot)$ | $(-, -)$ | $(-0, 0)$ | $(-0, +)$ | $(-0, -0)$ | $(-0, 0+)$ | $(-0, \top)$ |
| $0+$ | $(\bot, \bot)$ | $(\bot, \bot)$ | $(0, 0)$ | $(0+, +)$ | $(0, 0)$ | $(0+, 0+)$ | $(0+, \top)$ |
| $\top$ | $(\bot, \bot)$ | $(-, -)$ | $(-0, 0)$ | $(\top, +)$ | $(-0, -0)$ | $(\top, 0+)$ | $(\top, \top)$ |

After mechanical inspection of all cases, we derive the above table.

We can derive similar tables for $\leq^\sharp$, $=^\sharp$, $\neq^\sharp$, $>^\sharp$, and $\geq^\sharp$.

# Enhanced Approximate Transfer Mapping

$$\overline{h_{x:=e}}(\overline{v}) \;=\; \lambda y\,.\begin{cases}\overline{v}(y) & \text{if } y \neq x\\ \overline{[\![e]\!]}_{\overline{v}} & \text{if } y = x\end{cases} \qquad \overline{h_g}(\overline{v}) \;=\; \begin{cases}\overline{\bot} & \text{if } \overline{v} \not\models g\\ \overline{\theta_g}(\overline{v}) & \text{if } \overline{v} \models g\end{cases}$$

$g \;=\; x \;\bowtie\; x$

$$\overline{\theta_g}(\overline{v}) \;=\; \lambda y\,.\begin{cases}\overline{v}(y) & \text{if } \bowtie \in \{=, \leq, \geq\}\\ \overline{\bot} & \text{if } \bowtie \in \{\neq, <, >\}\end{cases}$$

$g \;=\; x_1 \;\bowtie\; x_2 \quad$ with $x_1 \neq x_2$

$$\overline{\theta_g}(\overline{v}) \;=\; \lambda y\,.\begin{cases}\overline{t_1} & \text{if } y = x_1\\ \overline{t_2} & \text{if } y = x_2\\ \overline{v}(y) & \text{otherwise}\end{cases} \qquad \text{where } (\overline{t_1}, \overline{t_2}) \;=\; \overline{[\![x_1]\!]}_{\overline{v}} \bowtie^\sharp \overline{[\![x_2]\!]}_{\overline{v}}$$

# Enhanced Approximate Transfer Mapping

$$\overline{h_{x:=e}}(\overline{v}) \quad = \quad \lambda y \, . \begin{cases} \overline{v}(y) & \text{if } y \neq x \\ [\![e]\!]_{\overline{v}} & \text{if } y = x \end{cases} \qquad\qquad \overline{h_g}(\overline{v}) \quad = \quad \begin{cases} \overline{\bot} & \text{if } \overline{v} \not\models g \\ \overline{\theta_g}(\overline{v}) & \text{if } \overline{v} \models g \end{cases}$$

### $g \;=\; x \;\bowtie\; x$

$$\overline{\theta_g}(\overline{v}) \quad = \quad \lambda y \, . \begin{cases} \overline{v}(y) & \text{if } \bowtie \in \{=, \leq, \geq\} \\ \overline{\bot} & \text{if } \bowtie \in \{\neq, <, >\} \end{cases}$$

### $g \;=\; x_1 \;\bowtie\; x_2$ with $x_1 \neq x_2$

$$\overline{\theta_g}(\overline{v}) \quad = \quad \lambda y \, . \begin{cases} \overline{t_1} & \text{if } y = x_1 \\ \overline{t_2} & \text{if } y = x_2 \\ \overline{v}(y) & \text{otherwise} \end{cases} \qquad \text{where } (\overline{t_1}, \overline{t_2}) \; = \; [\![x_1]\!]_{\overline{v}} \bowtie^\sharp [\![x_2]\!]_{\overline{v}}$$

# Enhanced Approximate Transfer Mapping

$$\overline{h_{x:=e}}(\overline{v}) \;=\; \lambda y \,.\begin{cases} \overline{v}(y) & \text{if } y \neq x \\ \overline{[\![e]\!]}_{\overline{v}} & \text{if } y = x \end{cases} \qquad \overline{h_g}(\overline{v}) \;=\; \begin{cases} \overline{\bot} & \text{if } \overline{v} \not\models g \\ \overline{\theta_g}(\overline{v}) & \text{if } \overline{v} \models g \end{cases}$$

## $g \;=\; x \;\bowtie\; e$ with $e$ not reduced to a variable

$$\overline{\theta_g}(\overline{v}) \;=\; \lambda y \,.\begin{cases} \overline{t} & \text{if } y = x \\ \overline{v}(y) & \text{otherwise} \end{cases} \qquad \text{where} \quad (\overline{t}, \_) \;=\; \overline{[\![x]\!]}_{\overline{v}} \bowtie^{\sharp} \overline{[\![e]\!]}_{\overline{v}}$$

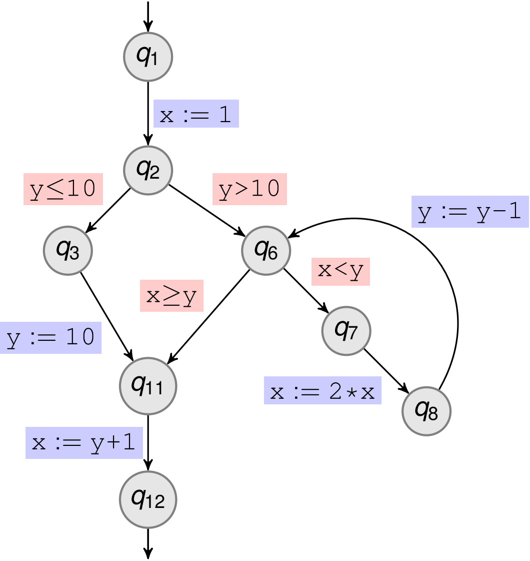
## $g \;=\; e \;\bowtie\; x$ with $e$ not reduced to a variable

$$\overline{\theta_g}(\overline{v}) \;=\; \lambda y \,.\begin{cases} \overline{t} & \text{if } y = x \\ \overline{v}(y) & \text{otherwise} \end{cases} \qquad \text{where} \quad (\_, \overline{t}) \;=\; \overline{[\![e]\!]}_{\overline{v}} \bowtie^{\sharp} \overline{[\![x]\!]}_{\overline{v}}$$

## $g \;=\; e_1 \;\bowtie\; e_2$ with $e_1, e_2$ not reduced to a variable

$$\overline{\theta_g}(\overline{v}) \;=\; \overline{v}$$

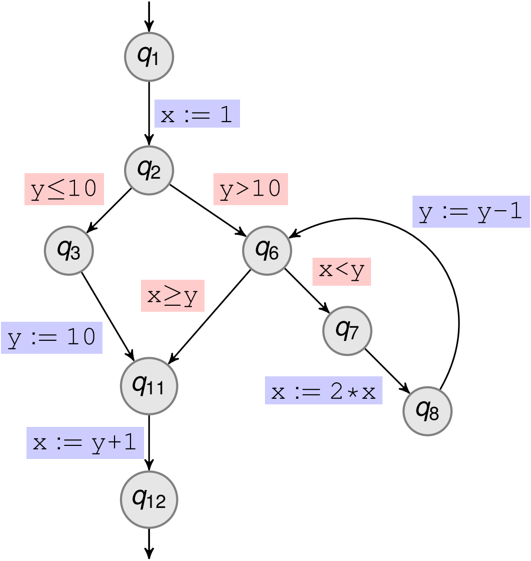
# Forward Sign Analysis on Example with Enhanced $\overline{h}_{\mathrm{op}}$



**Goal**

Show that $\texttt{x} > 0$ at $q_{12}$

# Forward Sign Analysis on Example with Enhanced $\overline{h}_{\mathrm{op}}$

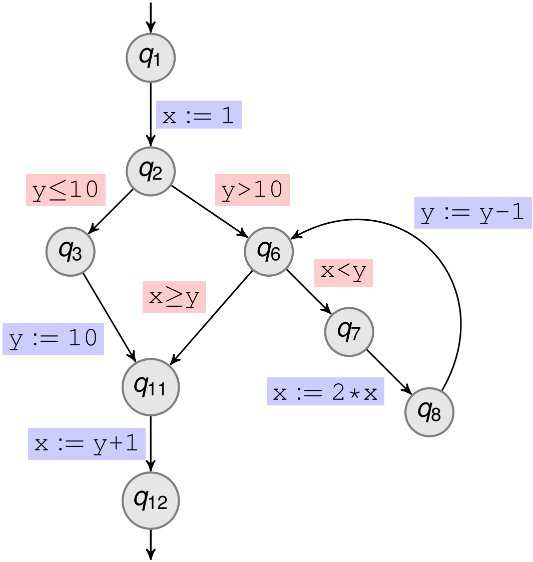


**Goal**

Show that $\mathtt{x} > 0$ at $q_{12}$

|  | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $\bot$ | $\bot$ |
| $q_3$ | $\bot$ | $\bot$ |
| $q_6$ | $\bot$ | $\bot$ |
| $q_7$ | $\bot$ | $\bot$ |
| $q_8$ | $\bot$ | $\bot$ |
| $q_{11}$ | $\bot$ | $\bot$ |
| $q_{12}$ | $\bot$ | $\bot$ |

# Forward Sign Analysis on Example with Enhanced $\overline{h}_{\mathrm{op}}$

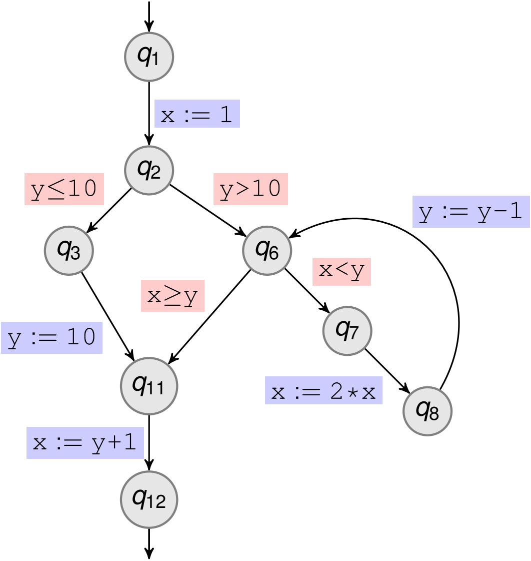


**Goal**

Show that x $>$ 0 at $q_{12}$

| | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $\perp$ | $\perp$ |
| $q_6$ | $\perp$ | $\perp$ |
| $q_7$ | $\perp$ | $\perp$ |
| $q_8$ | $\perp$ | $\perp$ |
| $q_{11}$ | $\perp$ | $\perp$ |
| $q_{12}$ | $\perp$ | $\perp$ |

# Forward Sign Analysis on Example with Enhanced $\overline{h}_{\mathrm{op}}$

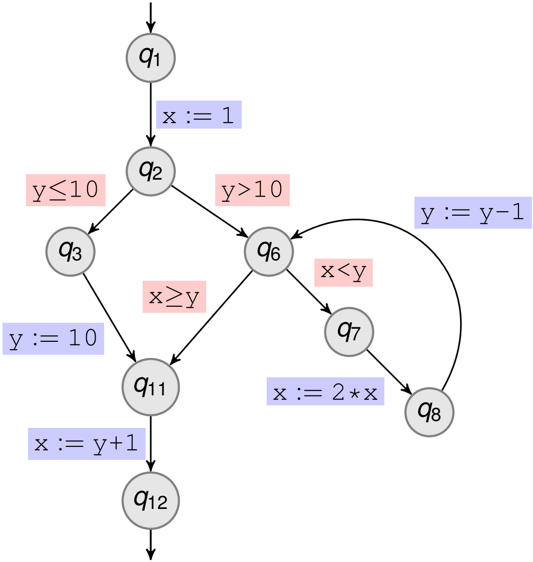


## Goal

Show that $x > 0$ at $q_{12}$

| | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $\bot$ | $\bot$ |
| $q_6$ | $\bot$ | $\bot$ |
| $q_7$ | $\bot$ | $\bot$ |
| $q_8$ | $\bot$ | $\bot$ |
| $q_{11}$ | $\bot$ | $\bot$ |
| $q_{12}$ | $\bot$ | $\bot$ |

# Forward Sign Analysis on Example with Enhanced $\overline{h_{\mathrm{op}}}$

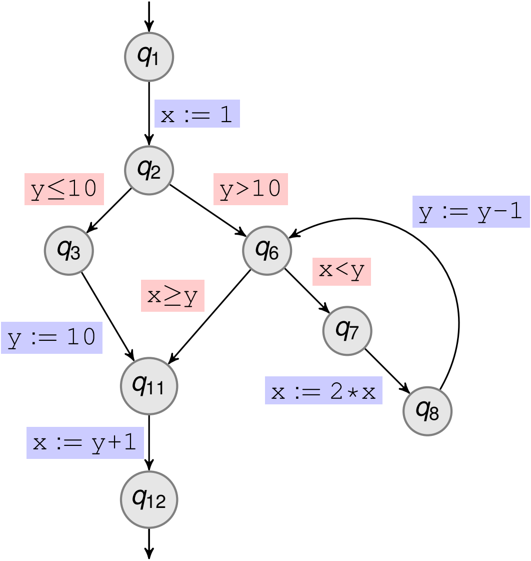


**Goal**

Show that x $>$ 0 at $q_{12}$

| | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $\perp$ | $\perp$ |
| $q_7$ | $\perp$ | $\perp$ |
| $q_8$ | $\perp$ | $\perp$ |
| $q_{11}$ | $\perp$ | $\perp$ |
| $q_{12}$ | $\perp$ | $\perp$ |

# Forward Sign Analysis on Example with Enhanced $\overline{h}_{\mathrm{op}}$

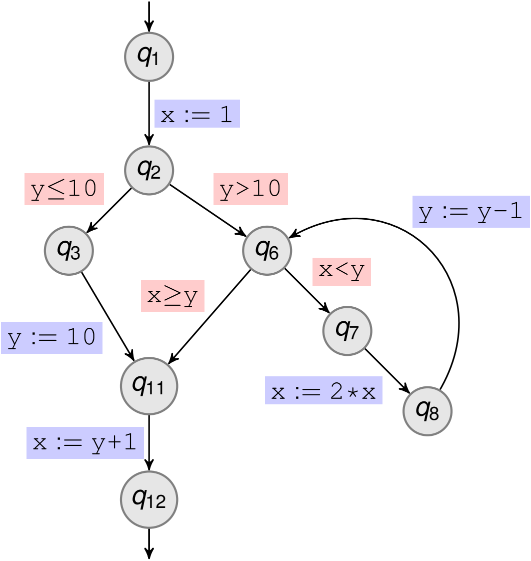


**Goal**

Show that $\mathtt{x} > 0$ at $q_{12}$

| | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $\bot$ | $\bot$ |
| $q_7$ | $\bot$ | $\bot$ |
| $q_8$ | $\bot$ | $\bot$ |
| $q_{11}$ | $\bot$ | $\bot$ |
| $q_{12}$ | $\bot$ | $\bot$ |

# Forward Sign Analysis on Example with Enhanced $\overline{h}_{\mathrm{op}}$

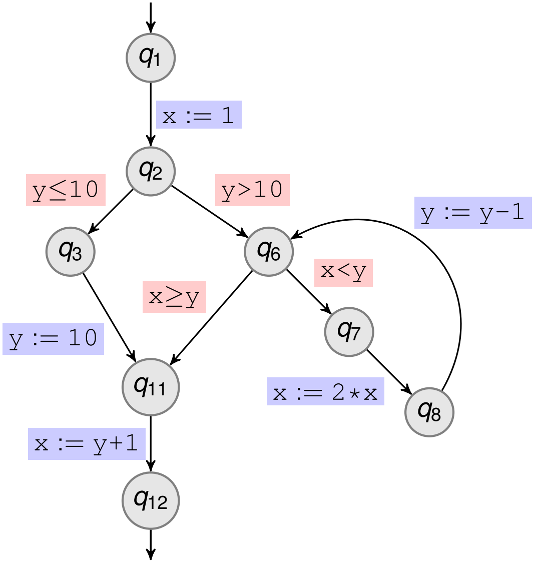


**Goal**

Show that x $> 0$ at $q_{12}$

| | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $\bot$ | $\bot$ |
| $q_7$ | $\bot$ | $\bot$ |
| $q_8$ | $\bot$ | $\bot$ |
| $q_{11}$ | $+$ | $+$ |
| $q_{12}$ | $\bot$ | $\bot$ |

# Forward Sign Analysis on Example with Enhanced $\overline{h}_{\mathrm{op}}$

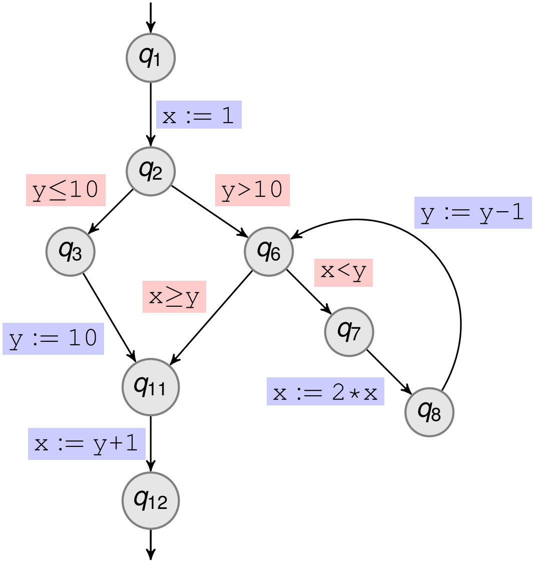


**Goal**

Show that $\mathtt{x} > 0$ at $q_{12}$

| | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $\bot$ | $\bot$ |
| $q_7$ | $\bot$ | $\bot$ |
| $q_8$ | $\bot$ | $\bot$ |
| $q_{11}$ | $+$ | $+$ |
| $q_{12}$ | $\bot$ | $\bot$ |

# Forward Sign Analysis on Example with Enhanced $\overline{h}_{\mathrm{op}}$

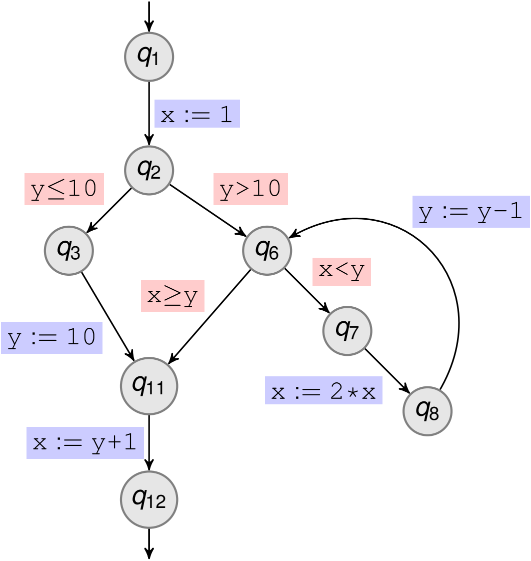


**Goal**

Show that $\mathtt{x} > 0$ at $q_{12}$

|  | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $\bot$ | $\bot$ |
| $q_7$ | $\bot$ | $\bot$ |
| $q_8$ | $\bot$ | $\bot$ |
| $q_{11}$ | $+$ | $+$ |
| $q_{12}$ | $+$ | $+$ |

# Forward Sign Analysis on Example with Enhanced $\overline{h}_{\mathrm{op}}$

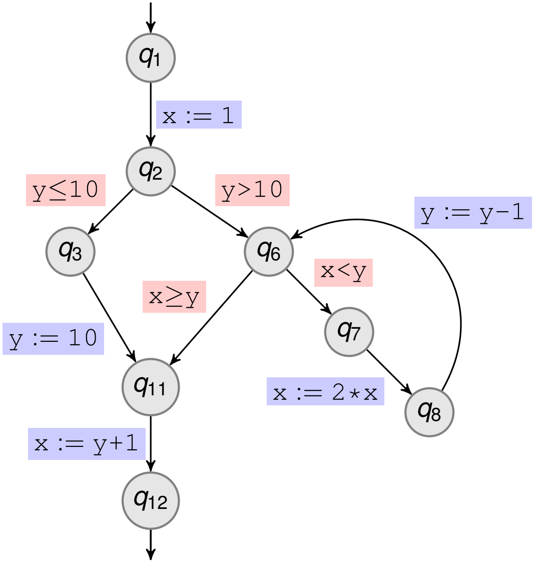


**Goal**

Show that x $>$ 0 at $q_{12}$

|  | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $\bot$ | $\bot$ |
| $q_7$ | $\bot$ | $\bot$ |
| $q_8$ | $\bot$ | $\bot$ |
| $q_{11}$ | $+$ | $+$ |
| $q_{12}$ | $+$ | $+$ |

# Forward Sign Analysis on Example with Enhanced $\overline{h}_{\mathrm{op}}$

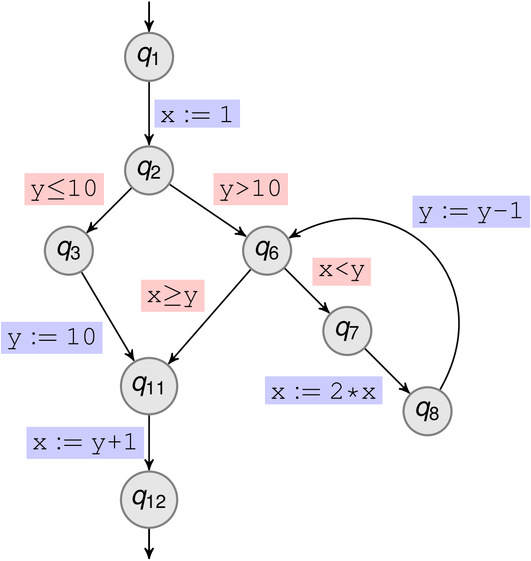


**Goal**

Show that $\mathtt{x} > 0$ at $q_{12}$

| | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $+$ | $+$ |
| $q_7$ | $\bot$ | $\bot$ |
| $q_8$ | $\bot$ | $\bot$ |
| $q_{11}$ | $+$ | $+$ |
| $q_{12}$ | $+$ | $+$ |

# Forward Sign Analysis on Example with Enhanced $\overline{h}_{\mathrm{op}}$

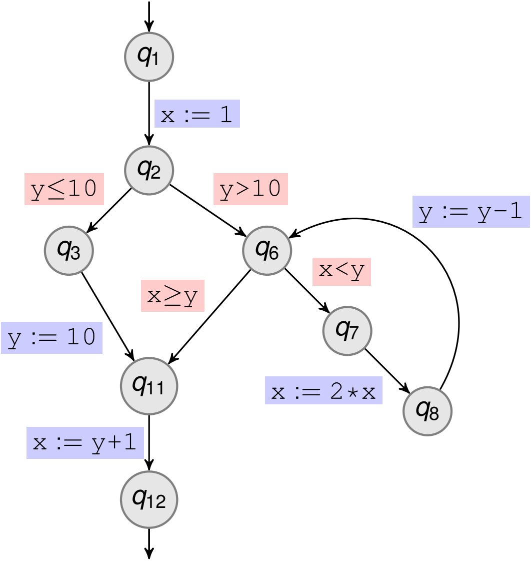


**Goal**

Show that $\mathtt{x} > 0$ at $q_{12}$

|  | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $+$ | $+$ |
| $q_7$ | $\bot$ | $\bot$ |
| $q_8$ | $\bot$ | $\bot$ |
| $q_{11}$ | $+$ | $+$ |
| $q_{12}$ | $+$ | $+$ |

# Forward Sign Analysis on Example with Enhanced $\overline{h}_{\mathrm{op}}$

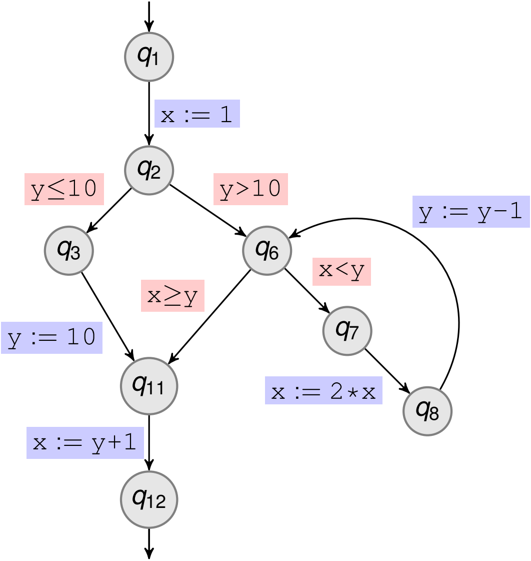


**Goal**

Show that $\mathtt{x} > 0$ at $q_{12}$

|  | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $+$ | $+$ |
| $q_7$ | $+$ | $+$ |
| $q_8$ | $\bot$ | $\bot$ |
| $q_{11}$ | $+$ | $+$ |
| $q_{12}$ | $+$ | $+$ |

# Forward Sign Analysis on Example with Enhanced $\overline{h}_{\mathrm{op}}$

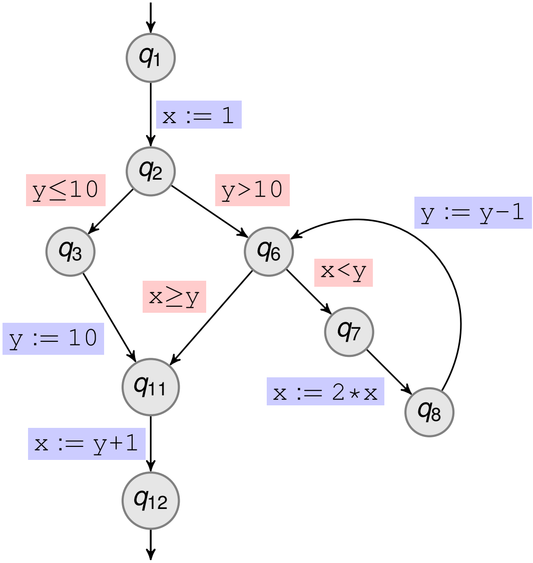


**Goal**

Show that $\mathtt{x} > 0$ at $q_{12}$

| | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $+$ | $+$ |
| $q_7$ | $+$ | $+$ |
| $q_8$ | $\bot$ | $\bot$ |
| $q_{11}$ | $+$ | $+$ |
| $q_{12}$ | $+$ | $+$ |

# Forward Sign Analysis on Example with Enhanced $\overline{h}_{\mathrm{op}}$

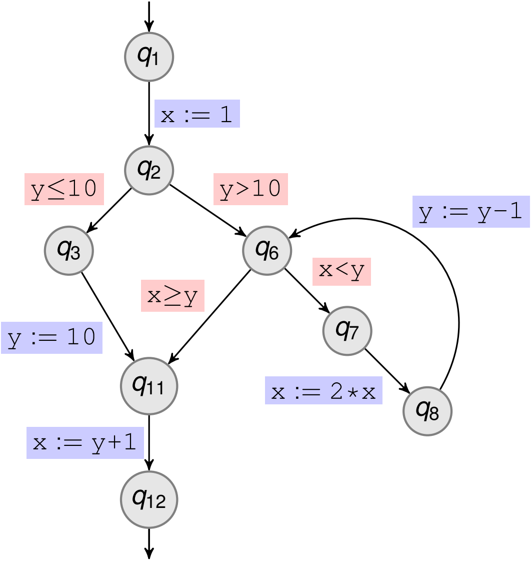


**Goal**

Show that $x > 0$ at $q_{12}$

| | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $+$ | $+$ |
| $q_7$ | $+$ | $+$ |
| $q_8$ | $+$ | $+$ |
| $q_{11}$ | $+$ | $+$ |
| $q_{12}$ | $+$ | $+$ |

# Forward Sign Analysis on Example with Enhanced $\overline{h}_{\mathrm{op}}$

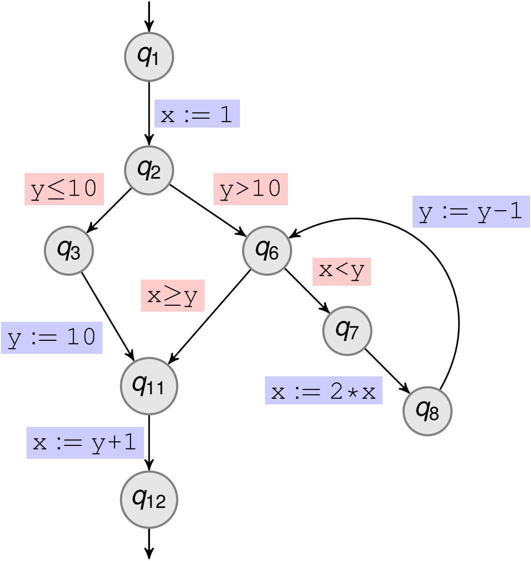


**Goal**

Show that $\mathtt{x} > 0$ at $q_{12}$

| | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $+$ | $+$ |
| $q_7$ | $+$ | $+$ |
| $q_8$ | $+$ | $+$ |
| $q_{11}$ | $+$ | $+$ |
| $q_{12}$ | $+$ | $+$ |

# Forward Sign Analysis on Example with Enhanced $\overline{h_{\mathrm{op}}}$

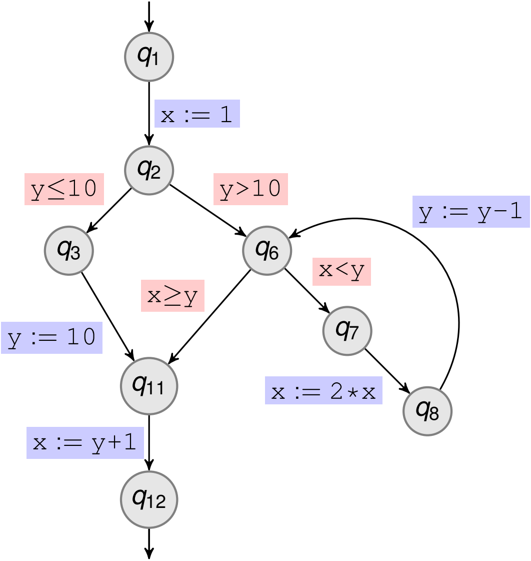


**Goal**

Show that x $>$ 0 at $q_{12}$

| | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $+$ | $\top$ |
| $q_7$ | $+$ | $+$ |
| $q_8$ | $+$ | $+$ |
| $q_{11}$ | $+$ | $+$ |
| $q_{12}$ | $+$ | $+$ |

# Forward Sign Analysis on Example with Enhanced $\overline{h}_{\mathrm{op}}$

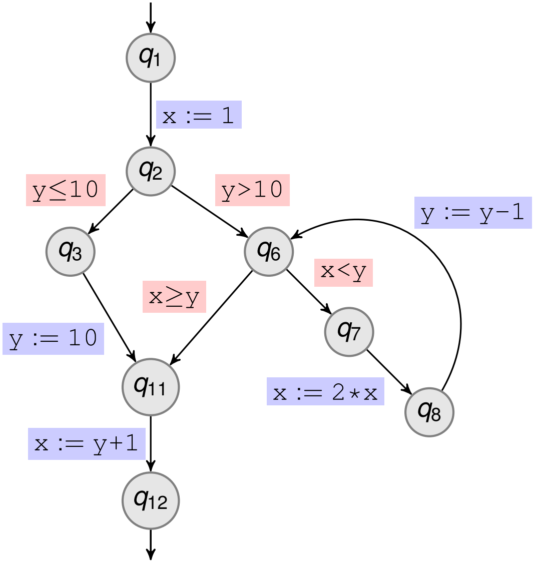


**Goal**

Show that $\mathtt{x} > 0$ at $q_{12}$

|  | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $+$ | $\top$ |
| $q_7$ | $+$ | $+$ |
| $q_8$ | $+$ | $+$ |
| $q_{11}$ | $+$ | $+$ |
| $q_{12}$ | $+$ | $+$ |

# Forward Sign Analysis on Example with Enhanced $\overline{h}_{\mathrm{op}}$

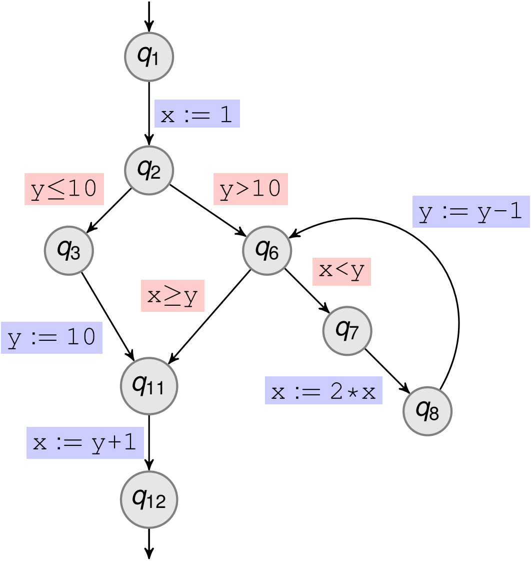


**Goal**

Show that $\mathtt{x} > 0$ at $q_{12}$

| | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $+$ | $\top$ |
| $q_7$ | $+$ | $+$ |
| $q_8$ | $+$ | $+$ |
| $q_{11}$ | $+$ | $+$ |
| $q_{12}$ | $+$ | $+$ |

# Forward Sign Analysis on Example with Enhanced $\overline{h}_{\mathrm{op}}$

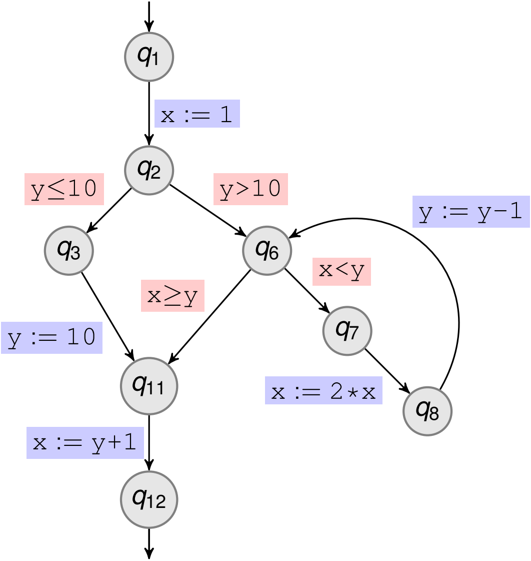


**Goal**

Show that $x > 0$ at $q_{12}$

|  | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $+$ | $\top$ |
| $q_7$ | $+$ | $+$ |
| $q_8$ | $+$ | $+$ |
| $q_{11}$ | $+$ | $+$ |
| $q_{12}$ | $+$ | $+$ |

# Forward Sign Analysis on Example with Enhanced $\overline{h}_{\mathrm{op}}$

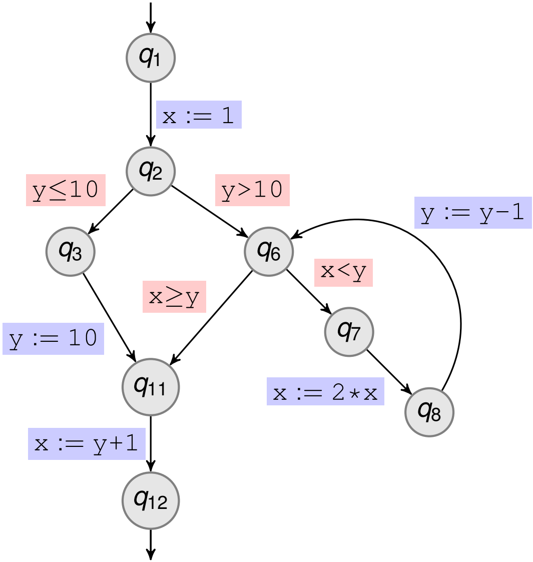


**Goal**

Show that $\mathtt{x} > 0$ at $q_{12}$

| | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $+$ | $\top$ |
| $q_7$ | $+$ | $+$ |
| $q_8$ | $+$ | $+$ |
| $q_{11}$ | $+$ | $\top$ |
| $q_{12}$ | $+$ | $+$ |

# Forward Sign Analysis on Example with Enhanced $\overline{h}_{\mathrm{op}}$

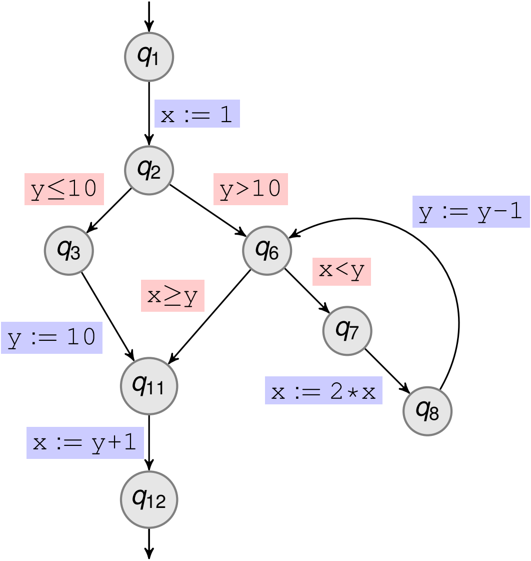


**Goal**

Show that $x > 0$ at $q_{12}$

| | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $+$ | $\top$ |
| $q_7$ | $+$ | $+$ |
| $q_8$ | $+$ | $+$ |
| $q_{11}$ | $+$ | $\top$ |
| $q_{12}$ | $+$ | $+$ |

# Forward Sign Analysis on Example with Enhanced $\overline{h}_{\mathrm{op}}$

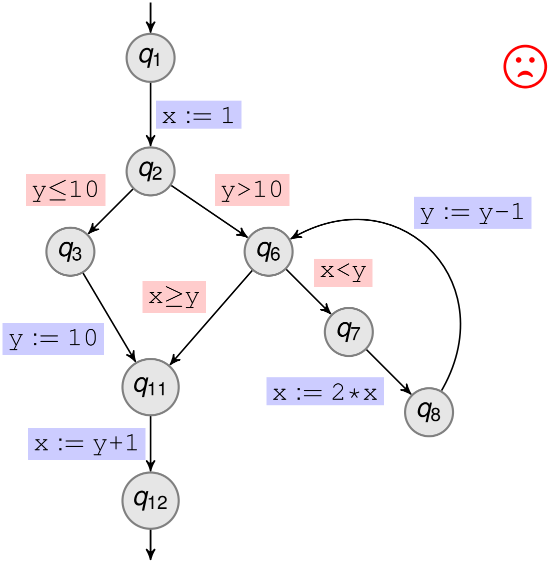


**Goal**

Show that x $> 0$ at $q_{12}$

| | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $+$ | $\top$ |
| $q_7$ | $+$ | $+$ |
| $q_8$ | $+$ | $+$ |
| $q_{11}$ | $+$ | $\top$ |
| $q_{12}$ | $\top$ | $\top$ |

# Forward Sign Analysis on Example with Enhanced $\overline{h}_{\mathrm{op}}$



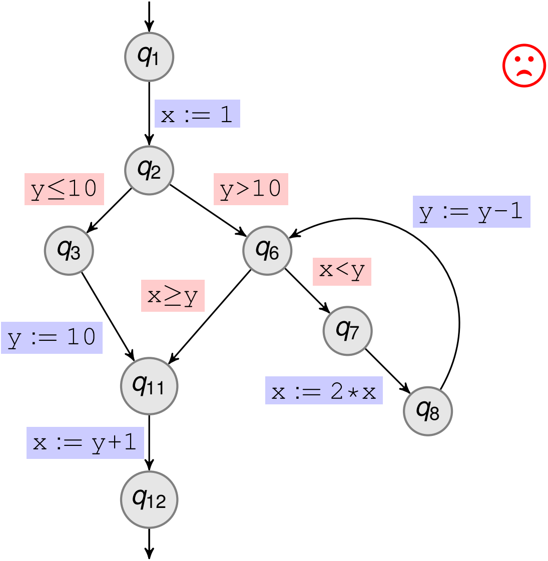
☹

**Goal**

Show that $x > 0$ at $q_{12}$

| | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $+$ | $\top$ |
| $q_7$ | $+$ | $+$ |
| $q_8$ | $+$ | $+$ |
| $q_{11}$ | $+$ | $\top$ |
| $q_{12}$ | $\top$ | $\top$ |

# Forward Sign Analysis on Example with Enhanced $\overline{h_{\mathrm{op}}}$



☹

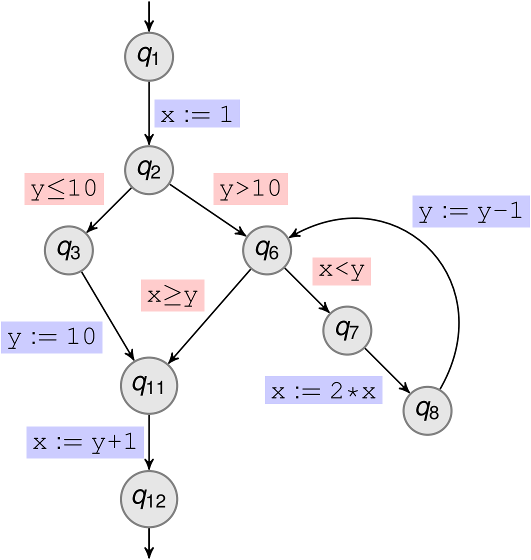
**Goal**

Show that $x > 0$ at $q_{12}$

|  | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $+$ | $\top$ |
| $q_7$ | $+$ | $+$ |
| $q_8$ | $+$ | $+$ |
| $q_{11}$ | $+$ | $\top$ |
| $q_{12}$ | $\top$ | $\top$ |

Getting closer…

# Forward Sign Analysis on Example with Enhanced $\overline{h}_{\mathrm{op}}$

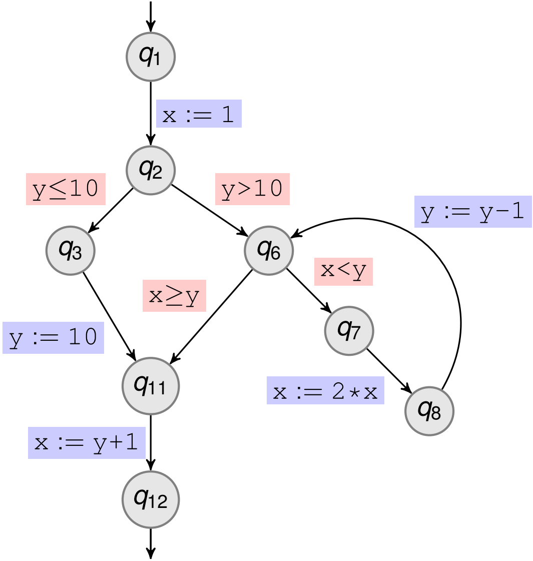


Show that $\mathtt{x} > 0$ at $q_{12}$

**Assumption**

Variables range over $\mathbb{Z}$

# Forward Sign Analysis on Example with Enhanced $\overline{h_{\mathrm{op}}}$



Show that $\mathtt{x} > 0$ at $q_{12}$

**Assumption**

Variables range over $\mathbb{Z}$

$$\gamma_{sign}(+) \quad = \quad \{1, 2, \ldots\}$$

**Tuned Semantics**

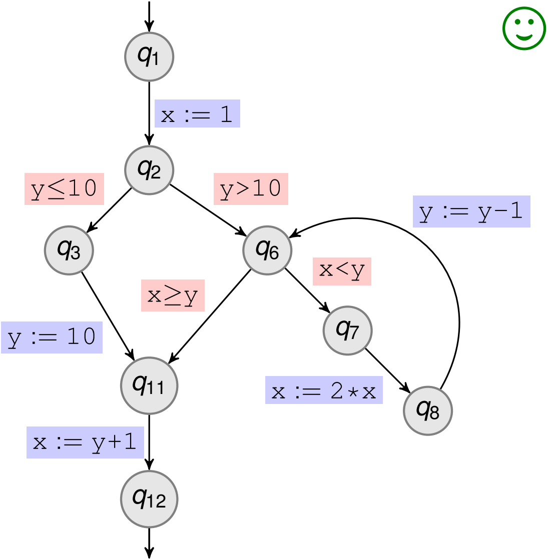
To exploit this new $\gamma_{sign}$

$\mathrm{op} \;=\; x := e - 1$

If $\;\overline{[\![e]\!]}_{\overline{v}} = +\;$ then

$$\overline{h_{\mathrm{op}}}(\overline{v})(x) \;=\; 0+$$

# Forward Sign Analysis on Example with Enhanced $\overline{h}_{\mathrm{op}}$



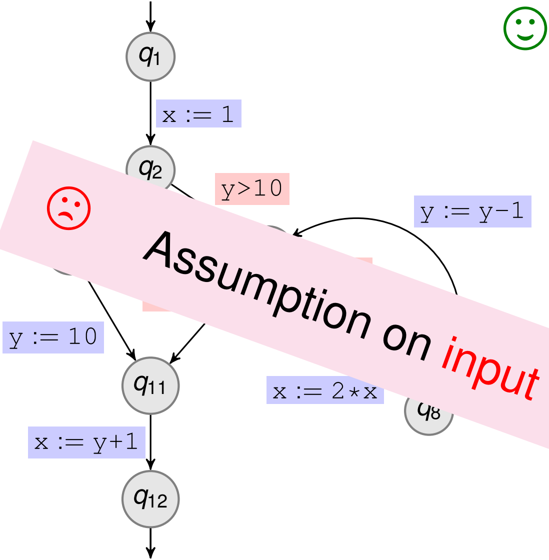
☺ Show that $\mathtt{x} > 0$ at $q_{12}$

**Assumption**

Variables range over $\mathbb{Z}$

| | x | y |
|---|---|---|
| $q_1$ | $\top$ | $\top$ |
| $q_2$ | $+$ | $\top$ |
| $q_3$ | $+$ | $\top$ |
| $q_6$ | $+$ | $0+$ |
| $q_7$ | $+$ | $+$ |
| $q_8$ | $+$ | $+$ |
| $q_{11}$ | $+$ | $0+$ |
| $q_{12}$ | $+$ | $+$ |

# Forward Sign Analysis on Example with Enhanced $\overline{h}_{\mathrm{op}}$



☺ Show that $\mathtt{x} > 0$ at $q_{12}$

**Assumption**

Variables range over $\mathbb{Z}$

|  | x | y |
|---|---|---|
| $q_1$ | ⊤ | ⊤ |
| $q_2$ | + | ⊤ |
| $q_3$ | + | ⊤ |
|  | + | 0+ |
|  |  | + |
|  |  |  |
| $q_{11}$ |  |  |
| $q_{12}$ | + |  |

☹ Assumption on input variable y ☹

# Beyond Sign Analysis. . .

A careful inspection of the control flow automaton shows that the desired property $\mathtt{x} > 0$ at $q_{12}$ holds for any real initial value of $\mathtt{y}$.

This property cannot be obtained even with best abstract transfer mapping $f^\sharp_{\mathtt{op}}$.

The *Sign* abstract lattice is not sufficient!

## Solution

Try with a finer abstract lattice!

# Beyond Sign Analysis...

A careful inspection of the control flow automaton shows that the desired property $\mathtt{x} > 0$ at $q_{12}$ holds for any real initial value of $\mathtt{y}$.

This property cannot be obtained even with best abstract transfer mapping $f^{\sharp}_{\mathtt{op}}$.

The *Sign* abstract lattice is not sufficient!

## Solution

Try with a finer abstract lattice!

# Outline — Abstract Interpretation

8 Some More Lattice Theory: Galois Connections

9 Abstract Interpretation-Based Data Flow Analysis

10 Convergence Acceleration with Widening and Narrowing

# Short Introduction to Widening and Narrowing

So far: finite height lattices. Iterative computation of MOP converges.

Finite height lattices not sufficient for (precise) numerical analysis

## Software Verification by Invariant Generation

Good precision is required for generation of useful invariants

Infinite height abstract lattices required to obtain good precision

But iterative computation of MOP may diverge in infinite height lattices.

## Solution

1. Use widening to compute a sound approximation of MOP.
2. Use narrowing to improve the precision of the approximation.

# Short Introduction to Widening and Narrowing

So far: finite height lattices. Iterative computation of MOP converges.

Finite height lattices not sufficient for (precise) numerical analysis

## Software Verification by Invariant Generation

Good precision is required for generation of useful invariants

Infinite height abstract lattices required to obtain good precision

But iterative computation of MOP may diverge in infinite height lattices.

## Solution

1. Use widening to compute a sound approximation of MOP.
2. Use narrowing to improve the precision of the approximation.

# Illustration with Range Analysis

## Objective of Range Analysis

Discover for each location the range of possible run-time values that variables can have at that location.

Generalizes both sign analysis and constant propagation analysis.

We will first design a Galois Connection between:

- $(\mathcal{P}(\mathrm{X} \to \mathbb{R}), \subseteq)$, concrete data flow facts from standard semantics
- $(\mathrm{X} \to \overline{L}, \sqsubseteq)$, where $(\overline{L}, \sqsubseteq)$ is an abstract lattice to represent range information.

Forward analysis

# Complete Lattice $\mathcal{Z}$: Extension of $\mathbb{Z}$ with $-\infty$ and $+\infty$

Let $\mathcal{Z} = \mathbb{Z} \cup \{-\infty, +\infty\}$ and define the partial order $\leq$ on $\mathcal{Z}$ with:

$$-\infty < \cdots < -2 < -1 < 0 < 1 < 2 < \cdots < +\infty$$

$(\mathcal{Z}, \leq)$ is a complete lattice

- Least element is $-\infty$ and greatest element is $+\infty$.
- Least upper bound sup $X$ is either $\max(X)$ if it exists, or $+\infty$.
- Greatest lower bound inf $X$ is either $\min(X)$ if it exists, or $-\infty$.

# Abstract Lattice of Intervals

$$-\infty < \cdots < -2 < -1 < 0 < 1 < 2 < \cdots < +\infty$$

$$Int \quad = \quad \{\bot\} \cup \{(l, u) \in (\mathbb{Z} \cup \{-\infty\}) \times (\mathbb{Z} \cup \{+\infty\}) \mid l \leq u\}$$

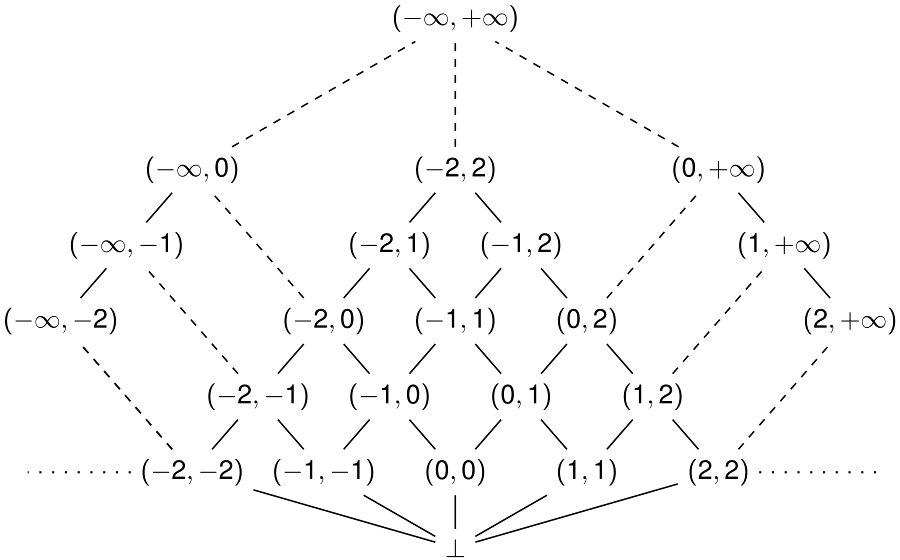
Define the binary relation $\sqsubseteq$ on *Int* as follows:

$$\begin{array}{ccccccc} \bot & \sqsubseteq & \bot & & \bot \ \sqsubseteq \ (l, u) & \not\sqsubseteq & \bot \\ (l_1, u_1) & \sqsubseteq & (l_2, u_2) & \text{if} & l_1 \leq l_2 \leq u_2 \leq u_1 & & \end{array}$$

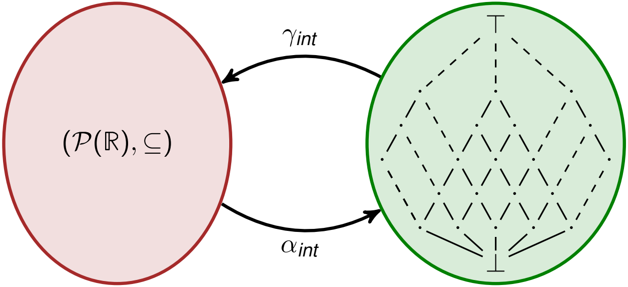
$(Int, \sqsubseteq)$ is a complete lattice

- Least element is $\bot$ and greatest element is $(-\infty, +\infty)$.
- $\bigsqcup X$ is either $\bot$ or $(\inf \{l \mid (l, u) \in X\}, \sup \{u \mid (l, u) \in X\})$.
- $\bigsqcap X$ is either $\bot$ or $(\sup \{l \mid (l, u) \in X\}, \inf \{u \mid (l, u) \in X\})$.

# Abstract Lattice of Intervals



# Interpretation of Abstract Intervals: Galois Connection



$$\alpha_{int}(\phi) = \begin{cases} \overline{\bot} & \text{if } \phi = \emptyset \\ (\inf \{\lfloor r \rfloor \mid r \in \phi\}, \sup \{\lceil r \rceil \mid r \in \phi\}) & \text{otherwise} \end{cases}$$

$$\begin{aligned} \gamma_{int}(\overline{\bot}) &= \emptyset \\ \gamma_{int}(\overline{(l, u)}) &= \{r \in \mathbb{R} \mid l \leq r \leq u\} \end{aligned}$$

# Range Analysis with Intervals: Galois Connection

## Projection

$$(\mathcal{P}(\mathrm{X} \to \mathbb{R}), \subseteq) \underset{\alpha_\pi}{\overset{\gamma_\pi}{\leftrightarrows}} (\mathrm{X} \to \mathcal{P}(\mathbb{R}), \subseteq)$$

## Intervals

$$(\mathcal{P}(\mathbb{R}), \subseteq) \underset{\alpha_{int}}{\overset{\gamma_{int}}{\leftrightarrows}} (\mathit{Int}, \sqsubseteq)$$

## Extension of Intervals to Functions

$$(\mathrm{X} \to \mathcal{P}(\mathbb{R}), \subseteq) \underset{\alpha_{int}}{\overset{\gamma_{int}}{\leftrightarrows}} (\mathrm{X} \to \mathit{Int}, \sqsubseteq)$$

$$(\mathcal{P}(\mathrm{X} \to \mathbb{R}), \subseteq) \underset{\alpha_{int} \circ \alpha_\pi}{\overset{\gamma_\pi \circ \gamma_{int}}{\leftrightarrows}} (\mathrm{X} \to \mathit{Int}, \sqsubseteq)$$

# Range Analysis with Intervals: Galois Connection

### Projection

$$(\mathcal{P}(X \to \mathbb{R}), \subseteq) \underset{\alpha_\pi}{\overset{\gamma_\pi}{\leftrightarrows}} (X \to \mathcal{P}(\mathbb{R}), \subseteq)$$

### Intervals

$$(\mathcal{P}(\mathbb{R}), \subseteq) \xleftarrow{\gamma_{int}}$$

### als to Functions

$$(\mathbb{R}), \subseteq) \underset{\alpha_{int}}{\overset{\gamma_{int}}{\leftrightarrows}} (X \to \mathit{Int}, \sqsubseteq)$$

☺ Same as sign analysis ☺

$$(\mathcal{P}(X \to \mathbb{R}), \subseteq) \underset{\alpha_{int} \circ \alpha_\pi}{\overset{\gamma_\pi \circ \gamma_{int}}{\leftrightarrows}} (X \to \mathit{Int}, \sqsubseteq)$$

# Range Analysis: Induced Abstract Data Flow Instance

$$(\mathcal{P}(\mathrm{X} \to \mathbb{R}), \subseteq) \underset{\alpha_\pi}{\overset{\gamma_\pi}{\leftrightarrows}} (\mathrm{X} \to \mathcal{P}(\mathbb{R}), \subseteq) \underset{\alpha_{int}}{\overset{\gamma_{int}}{\leftrightarrows}} (\mathrm{X} \to \mathit{Int}, \sqsubseteq)$$

$$\begin{aligned} \overline{\mathcal{F}} &= (\mathrm{X} \to \mathit{Int}) \xrightarrow{\text{mon}} (\mathrm{X} \to \mathit{Int}) \\ \overline{f} &= \lambda\, \mathrm{op} \,.\, f^\sharp_{\mathrm{op}} \\ \overline{\imath} &= \alpha_{int} \circ \alpha_\pi(\top) = \lambda\, x \,.\, \overline{\top} \end{aligned}$$

$$f^\sharp_{x:=e}(\overline{v}) = \lambda\, y \,.\, \begin{cases} \overline{v}(y) & \text{if } y \neq x \\ \alpha_{int} \circ [\![e]\!] \circ \gamma(\overline{v}) & \text{if } y = x \end{cases}$$

$$f^\sharp_g = \alpha_{int} \circ \alpha_\pi \circ [\![g]\!] \circ \gamma_\pi \circ \gamma_{int}$$

# Range Analysis: Induced Abstract Data Flow Instance

$$(\mathcal{P}(\mathrm{X} \to \mathbb{R}), \subseteq) \underset{\alpha_\pi}{\overset{\gamma_\pi}{\leftrightarrows}} (\mathrm{X} \to \mathcal{P}(\mathbb{R}), \subseteq) \underset{\alpha_{int}}{\overset{\gamma_{int}}{\leftrightarrows}} (\mathrm{X} \to Int, \sqsubseteq)$$

$$\overline{\mathcal{F}} = (\mathrm{X} \to Int) \xrightarrow{\text{mon}} (\mathrm{X} \to In$$

$$\overline{f} = \lambda\, \mathrm{op} \,.\, f^\sharp_{\mathrm{op}}$$

$$\overline{\imath} = \alpha_{int} \circ$$

☺ Same as sign analysis ☺

$$\lambda y \,.\, \begin{cases} \overline{v}(y) & \text{if } y \neq x \\ \alpha_{int} \circ [\![e]\!] \circ \gamma(\overline{v}) & \text{if } y = x \end{cases}$$

$$f^\sharp_g = \alpha_{int} \circ \alpha_\pi \circ [\![g]\!] \circ \gamma_\pi \circ \gamma_{int}$$

# Range Analysis: Approximate Transfer Mapping

$$\overline{h_{x:=e}}(\overline{v}) \;=\; \lambda y\,.\begin{cases}\overline{v}(y) & \text{if } y \neq x\\ \overline{\llbracket} e \overline{\rrbracket}_{\overline{v}} & \text{if } y = x\end{cases} \qquad \overline{h_g}(\overline{v}) \;=\; \begin{cases}\overline{\perp} & \text{if } \overline{v} \not\models g\\ \overline{v} & \text{if } \overline{v} \models g\end{cases}$$

## Similar to Sign Analysis

- Definition of abstract arithmetic operators and comparators
- Definition of $\overline{\llbracket} e \overline{\rrbracket}_{\overline{v}}$ and of $\overline{v} \models g$
- Precision enhancement by gaining information from guards ("refinement" of $\overline{h_g}(\overline{v})$)

Care about effective computability of $f^\sharp_{\mathrm{op}}$? Not in this section. . .

We will use $f^\sharp$ for range analysis

# Range Analysis: Approximate Transfer Mapping

$$\overline{h_{x:=e}}(\overline{v}) \;=\; \lambda y \,.\, \begin{cases} \overline{v}(y) & \text{if } y \neq x \\ \overline{[\![e]\!]}_{\overline{v}} & \text{if } y = x \end{cases} \qquad\qquad \overline{h_g}(\overline{v}) \;=\; \begin{cases} \overline{\bot} & \text{if } \overline{v} \not\models g \\ \overline{v} & \text{if } \overline{v} \models g \end{cases}$$

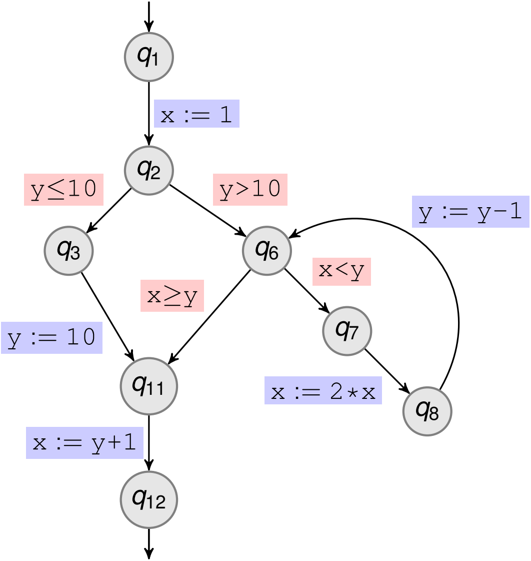
## Similar to Sign Analysis

- Definition of abstract arithmetic operators and comparators
- Definition of $\overline{[\![e]\!]}_{\overline{v}}$ and of $\overline{v} \models g$
- Precision enhancement by gaining information from guards ("refinement" of $\overline{h_g}(\overline{v})$)

Care about effective computability of $f^{\sharp}_{\mathrm{op}}$? Not in this section. . .

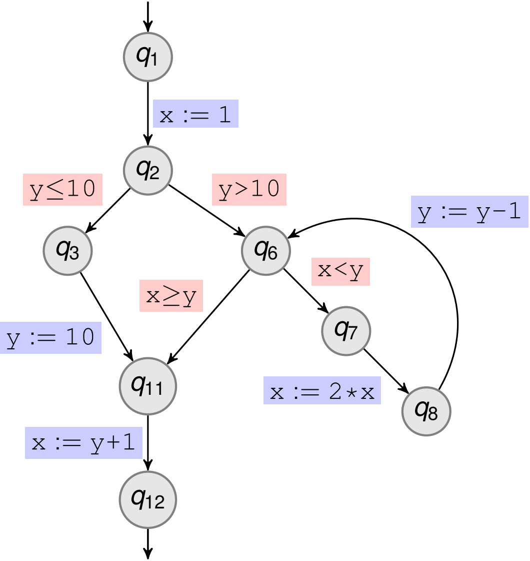
We will use $f^{\sharp}$ for range analysis

# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

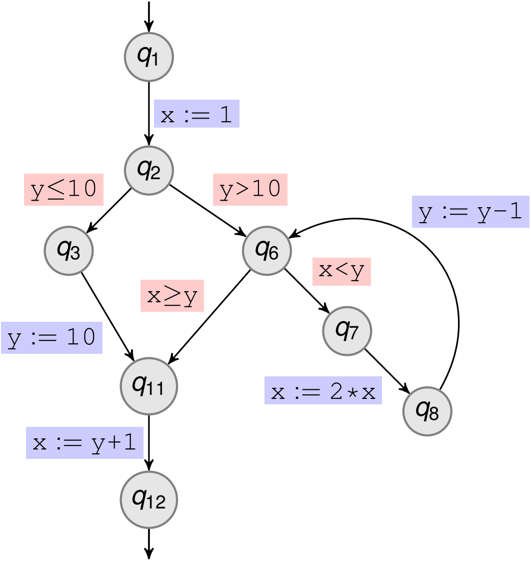
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_3$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_6$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_7$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_8$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_{11}$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_{12}$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |

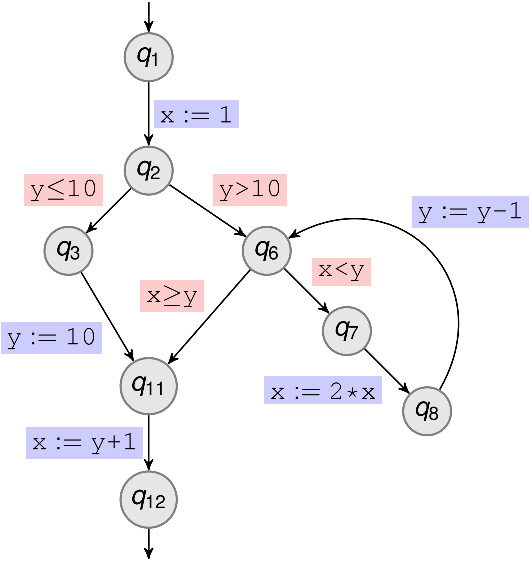
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

|  | x |  | y |  |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_6$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_7$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_8$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_{11}$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_{12}$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |

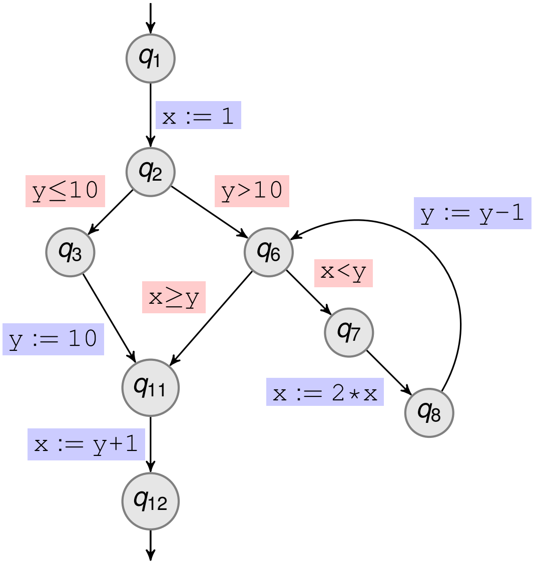
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

|  | x |  | y |  |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_6$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_7$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_8$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_{11}$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_{12}$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |

# Range Analysis on Running Example

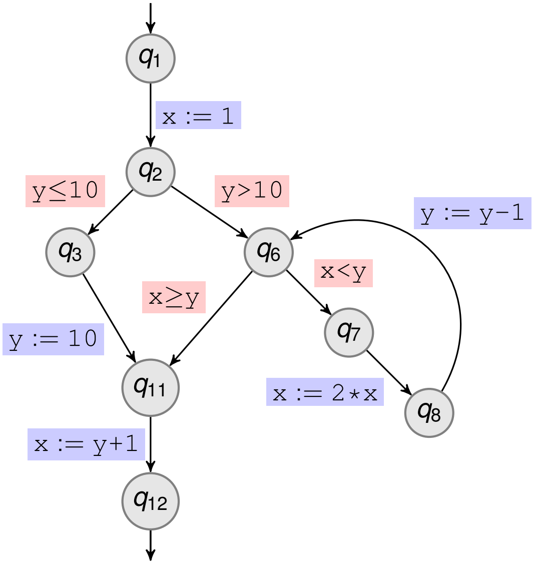


Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_7$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_8$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_{11}$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_{12}$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |

Gained from guard $\mathtt{y} \leq \mathtt{10}$

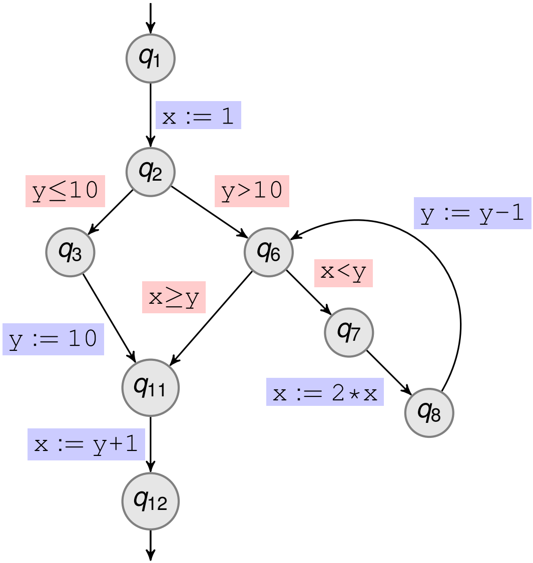
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_7$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_8$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_{11}$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_{12}$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |

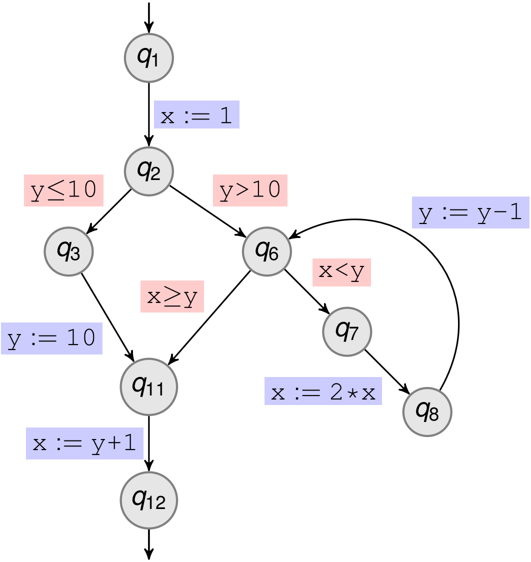
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | $\bot$ | $\bot$ | $\bot$ | $\bot$ |
| $q_7$ | $\bot$ | $\bot$ | $\bot$ | $\bot$ |
| $q_8$ | $\bot$ | $\bot$ | $\bot$ | $\bot$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | $\bot$ | $\bot$ | $\bot$ | $\bot$ |

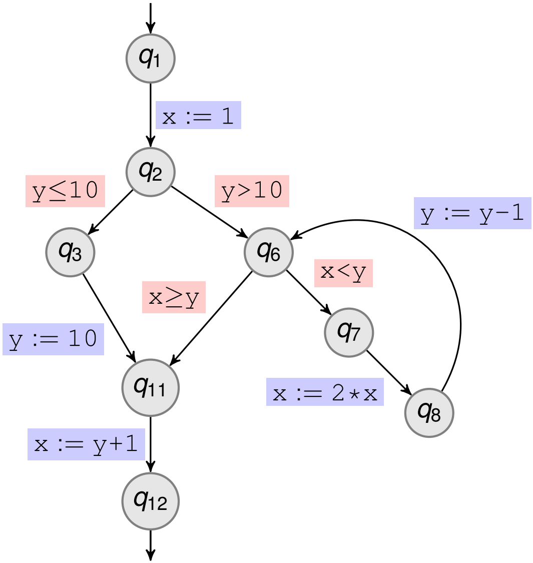
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_7$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_8$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |

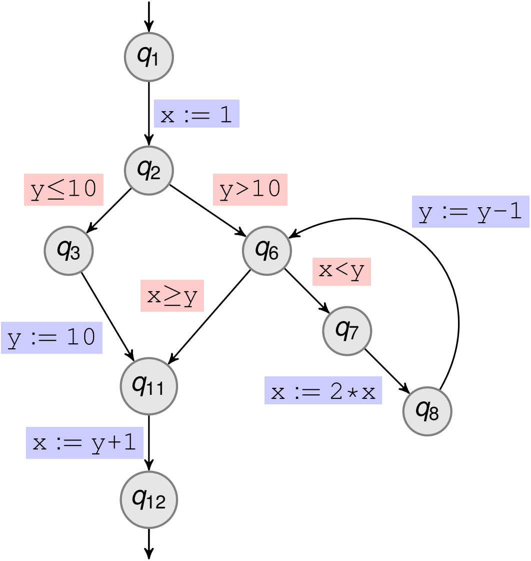
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_7$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_8$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

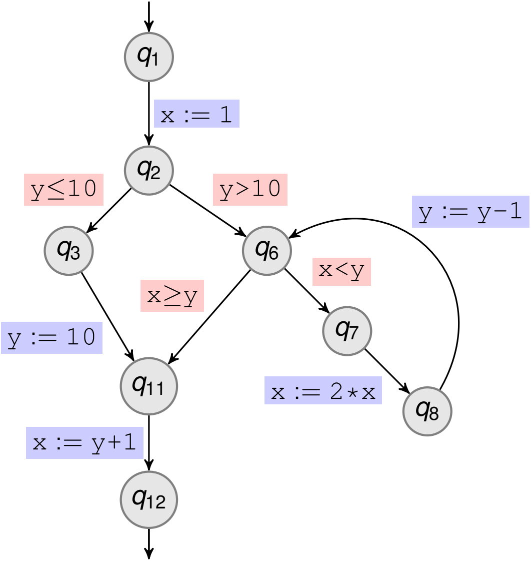
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_7$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_8$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

# Range Analysis on Running Example

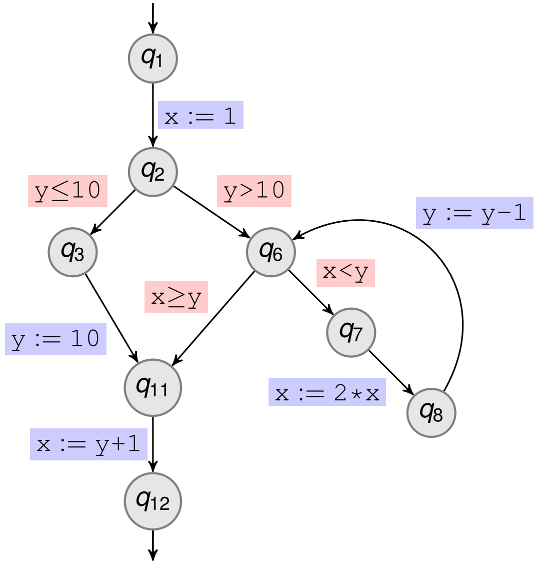


Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 1 | 10 | $+\infty$ |
| $q_7$ | $\bot$ | $\bot$ | $\bot$ | $\bot$ |
| $q_8$ | $\bot$ | $\bot$ | $\bot$ | $\bot$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

Gained from guard $\mathtt{y} > \mathtt{10}$

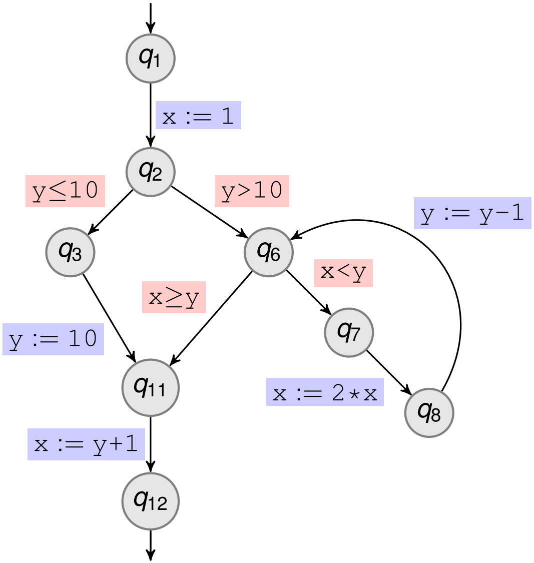
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 1 | 10 | $+\infty$ |
| $q_7$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_8$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

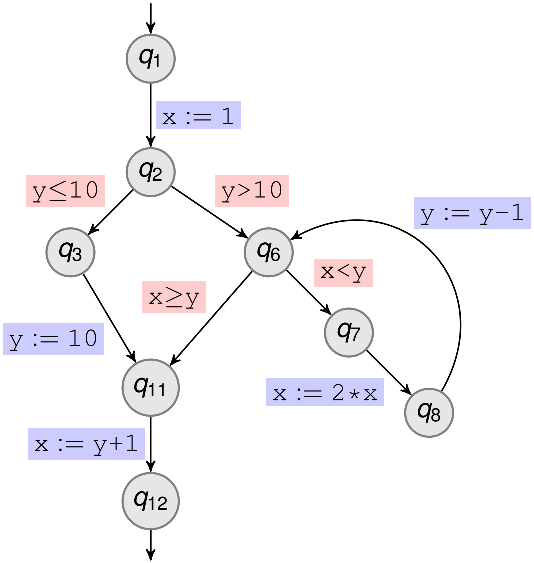
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 1 | 10 | $+\infty$ |
| $q_7$ | 1 | 1 | 10 | $+\infty$ |
| $q_8$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

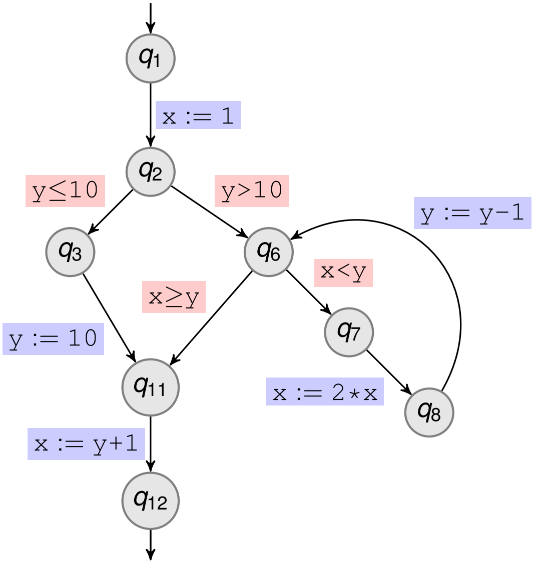
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 1 | 10 | $+\infty$ |
| $q_7$ | 1 | 1 | 10 | $+\infty$ |
| $q_8$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

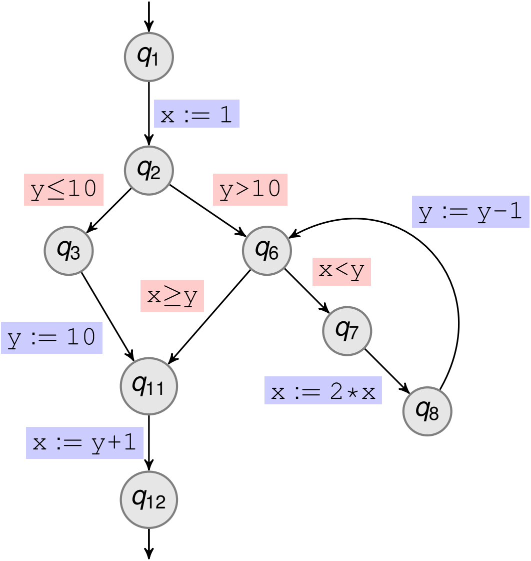
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 1 | 10 | $+\infty$ |
| $q_7$ | 1 | 1 | 10 | $+\infty$ |
| $q_8$ | 2 | 2 | 10 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

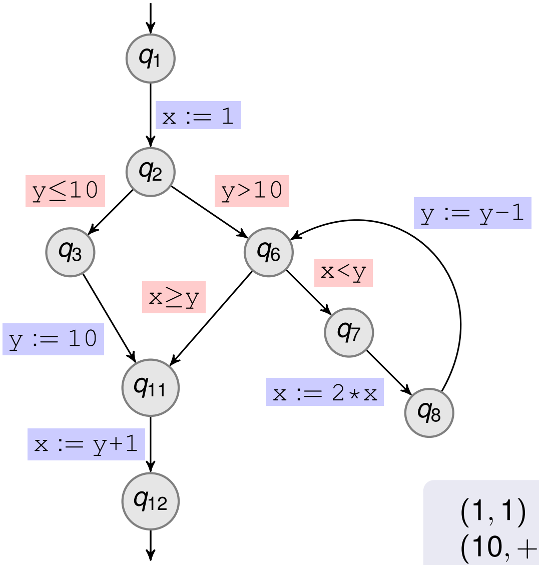
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 1 | 10 | $+\infty$ |
| $q_7$ | 1 | 1 | 10 | $+\infty$ |
| $q_8$ | 2 | 2 | 10 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

# Range Analysis on Running Example

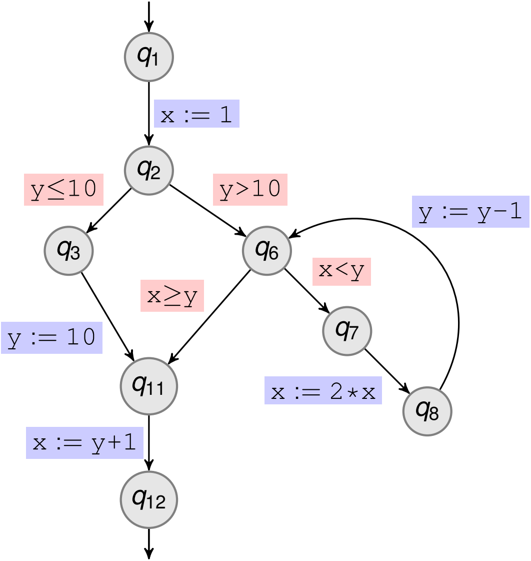


Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 2 | 9 | $+\infty$ |
| $q_7$ | 1 | 1 | 10 | $+\infty$ |
| $q_8$ | 2 | 2 | 10 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

$$\begin{array}{llll} (1,1) & \sqcup\ (2,2) & = & (1,2) \\ (10,+\infty) & \sqcup\ (9,+\infty) & = & (9,+\infty) \end{array}$$

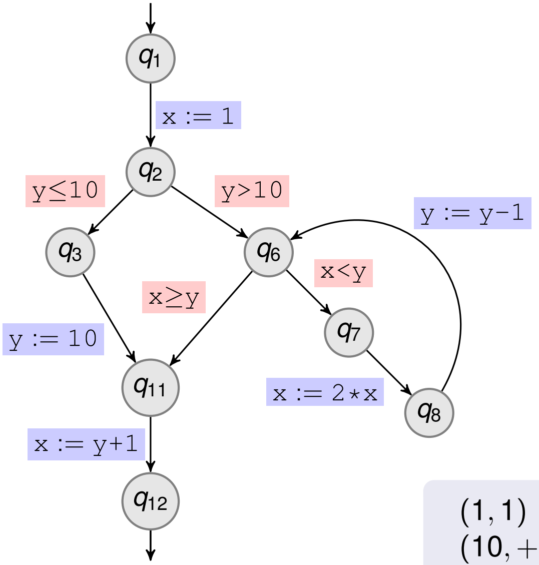
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 2 | 9 | $+\infty$ |
| $q_7$ | 1 | 1 | 10 | $+\infty$ |
| $q_8$ | 2 | 2 | 10 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

# Range Analysis on Running Example

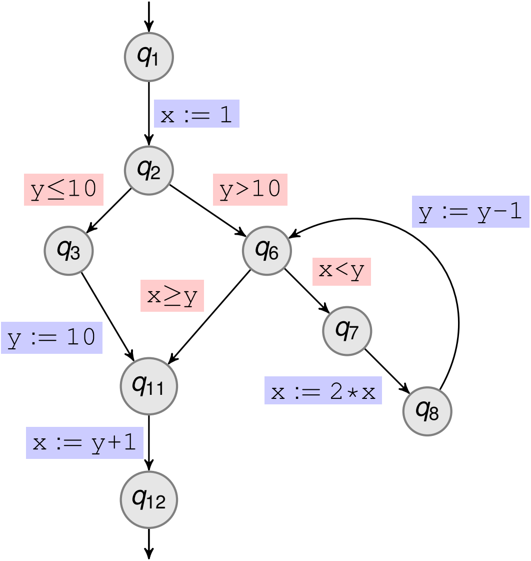


Recall that $\top = (-\infty, +\infty)$

|  | x |  | y |  |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 2 | 9 | $+\infty$ |
| $q_7$ | 1 | 2 | 9 | $+\infty$ |
| $q_8$ | 2 | 2 | 10 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

$$\begin{array}{llll} (1,1) & \sqcup\ (1,2) & = & (1,2) \\ (10,+\infty) & \sqcup\ (9,+\infty) & = & (9,+\infty) \end{array}$$

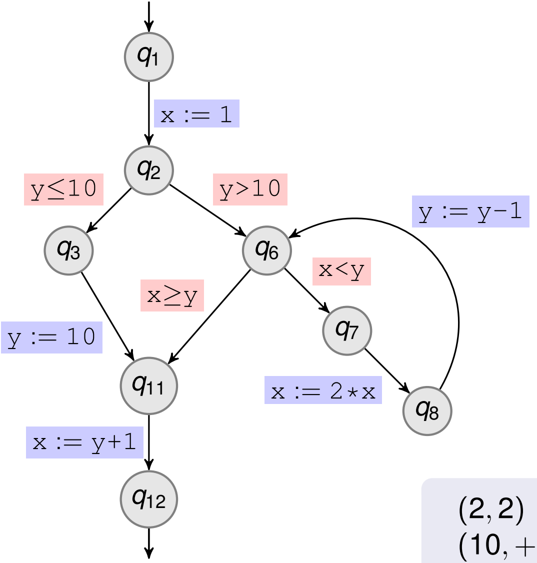
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 2 | 9 | $+\infty$ |
| $q_7$ | 1 | 2 | 9 | $+\infty$ |
| $q_8$ | 2 | 2 | 10 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

# Range Analysis on Running Example

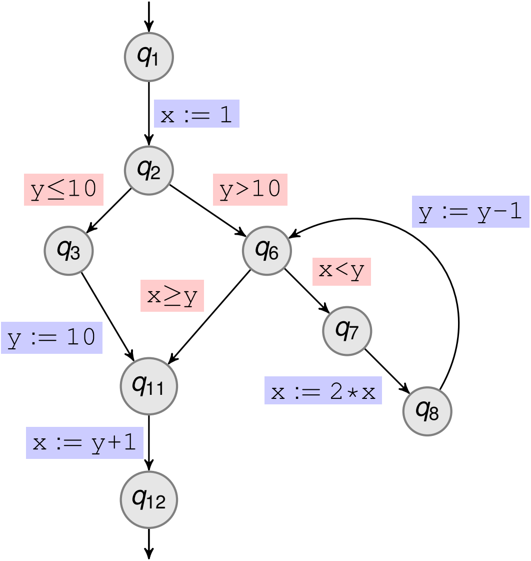


Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 2 | 9 | $+\infty$ |
| $q_7$ | 1 | 2 | 9 | $+\infty$ |
| $q_8$ | 2 | 4 | 9 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

$$(2,2) \sqcup (2,4) = (2,4)$$
$$(10,+\infty) \sqcup (9,+\infty) = (9,+\infty)$$

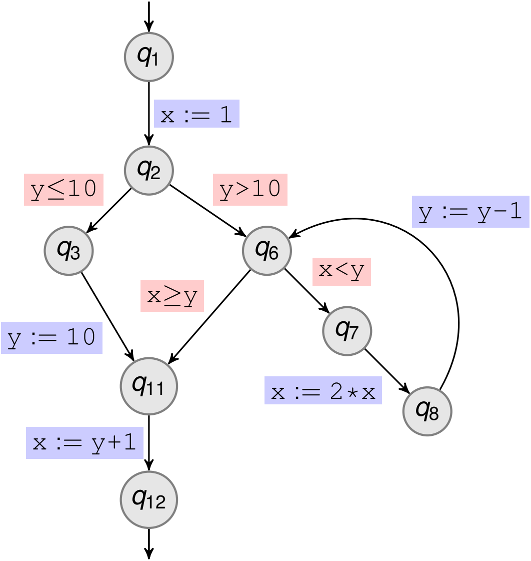
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 2 | 9 | $+\infty$ |
| $q_7$ | 1 | 2 | 9 | $+\infty$ |
| $q_8$ | 2 | 4 | 9 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

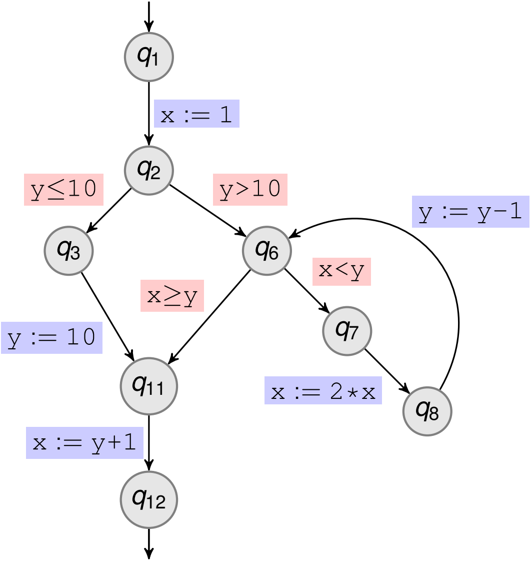
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 4 | 8 | $+\infty$ |
| $q_7$ | 1 | 2 | 9 | $+\infty$ |
| $q_8$ | 2 | 4 | 9 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

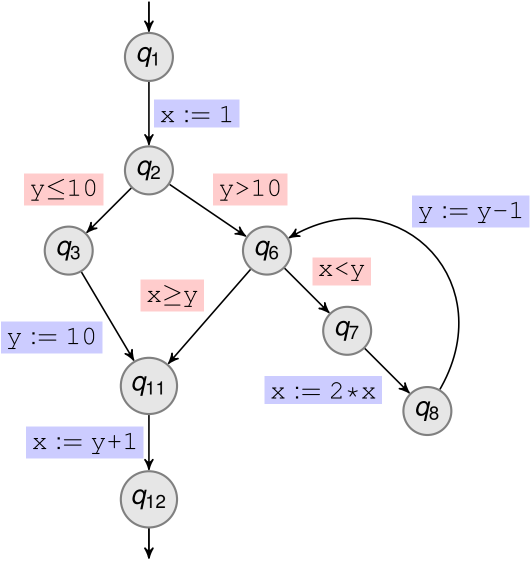
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 4 | 8 | $+\infty$ |
| $q_7$ | 1 | 2 | 9 | $+\infty$ |
| $q_8$ | 2 | 4 | 9 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

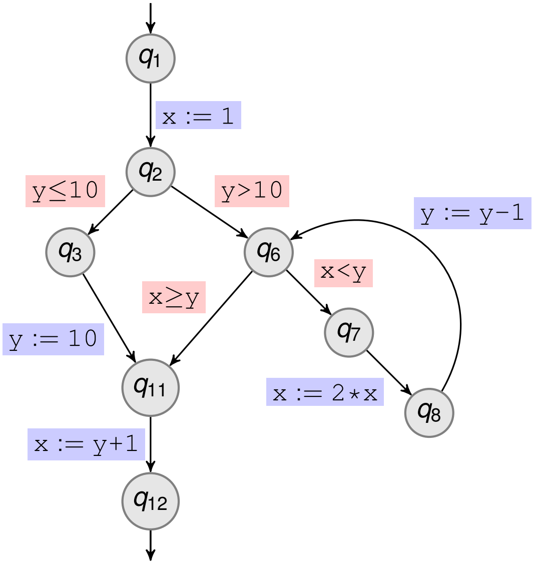
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 4 | 8 | $+\infty$ |
| $q_7$ | 1 | 4 | 8 | $+\infty$ |
| $q_8$ | 2 | 4 | 9 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

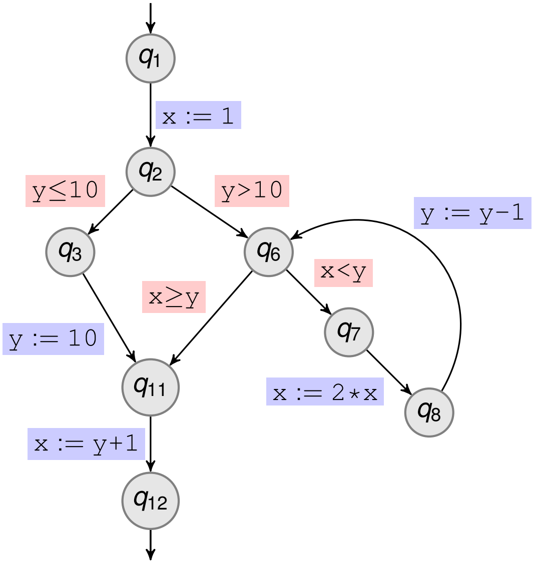
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 4 | 8 | $+\infty$ |
| $q_7$ | 1 | 4 | 8 | $+\infty$ |
| $q_8$ | 2 | 4 | 9 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

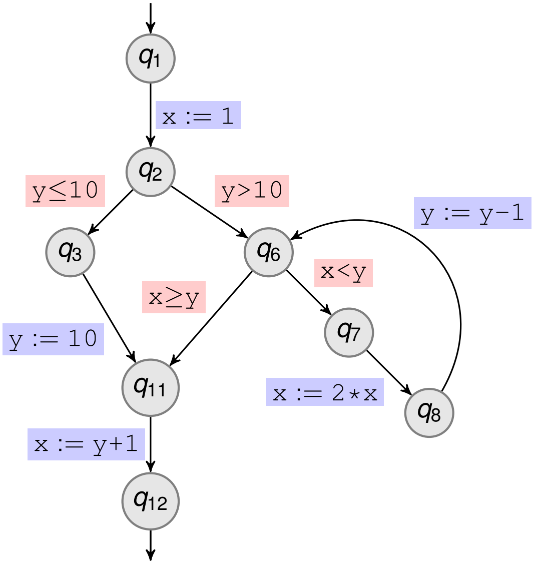
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 4 | 8 | $+\infty$ |
| $q_7$ | 1 | 4 | 8 | $+\infty$ |
| $q_8$ | 2 | 8 | 8 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

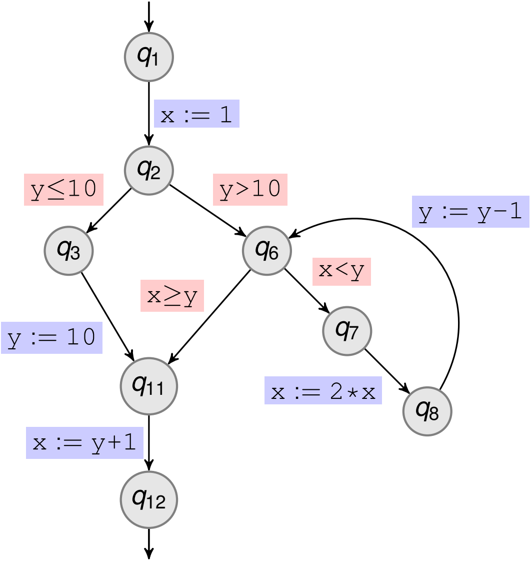
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 4 | 8 | $+\infty$ |
| $q_7$ | 1 | 4 | 8 | $+\infty$ |
| $q_8$ | 2 | 8 | 8 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

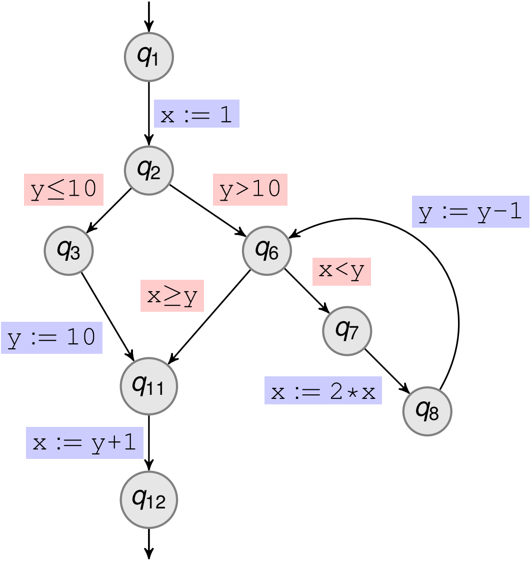
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 8 | 7 | $+\infty$ |
| $q_7$ | 1 | 4 | 8 | $+\infty$ |
| $q_8$ | 2 | 8 | 8 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

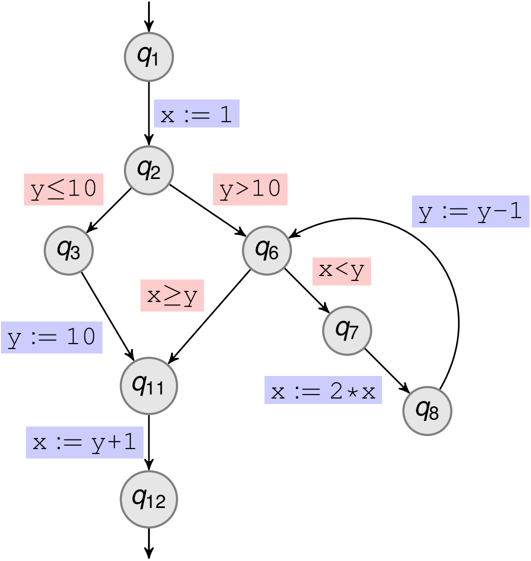
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 8 | 7 | $+\infty$ |
| $q_7$ | 1 | 4 | 8 | $+\infty$ |
| $q_8$ | 2 | 8 | 8 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

# Range Analysis on Running Example

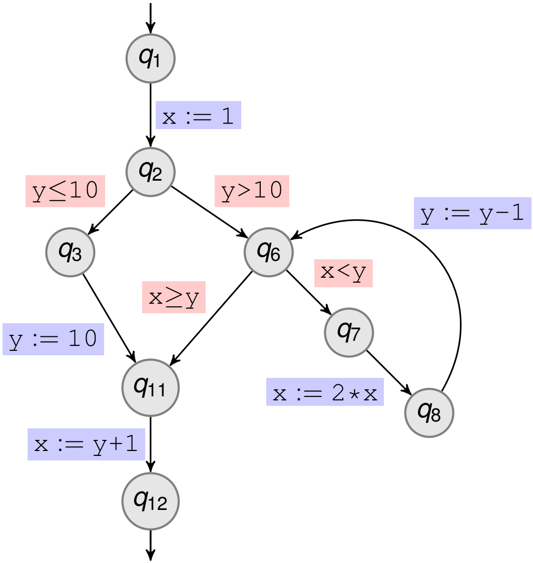


Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 8 | 7 | $+\infty$ |
| $q_7$ | 1 | 8 | 7 | $+\infty$ |
| $q_8$ | 2 | 8 | 8 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

Nothing to be gained from guard $x < y$

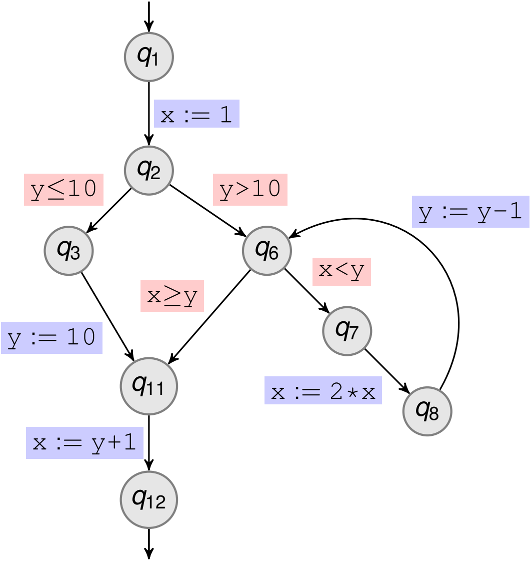
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 8 | 7 | $+\infty$ |
| $q_7$ | 1 | 8 | 7 | $+\infty$ |
| $q_8$ | 2 | 8 | 8 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

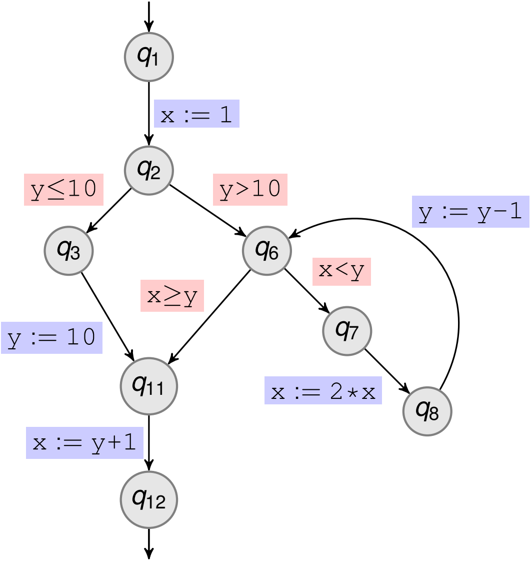
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 8 | 7 | $+\infty$ |
| $q_7$ | 1 | 8 | 7 | $+\infty$ |
| $q_8$ | 2 | 16 | 7 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

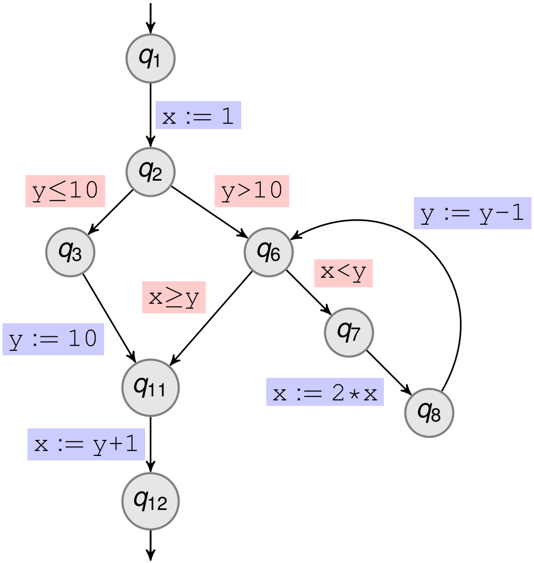
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 8 | 7 | $+\infty$ |
| $q_7$ | 1 | 8 | 7 | $+\infty$ |
| $q_8$ | 2 | 16 | 7 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

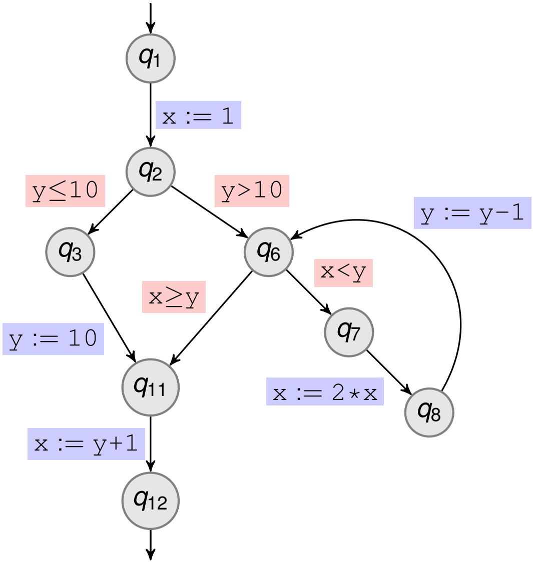
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 16 | 6 | $+\infty$ |
| $q_7$ | 1 | 16 | 6 | $+\infty$ |
| $q_8$ | 2 | 32 | 6 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

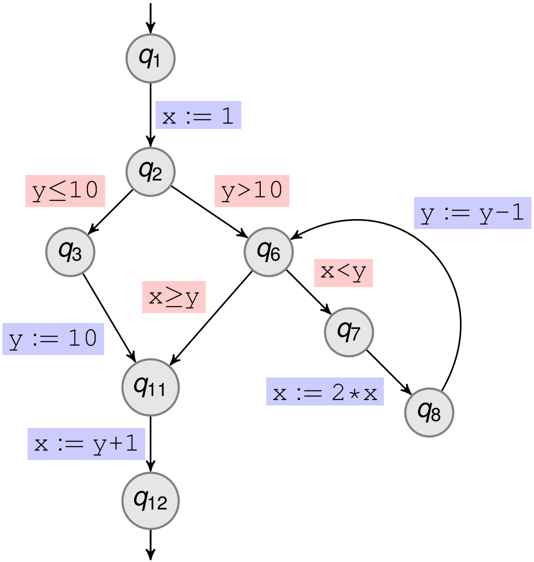
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 32 | 5 | $+\infty$ |
| $q_7$ | 1 | 32 | 5 | $+\infty$ |
| $q_8$ | 2 | $2^6$ | 5 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

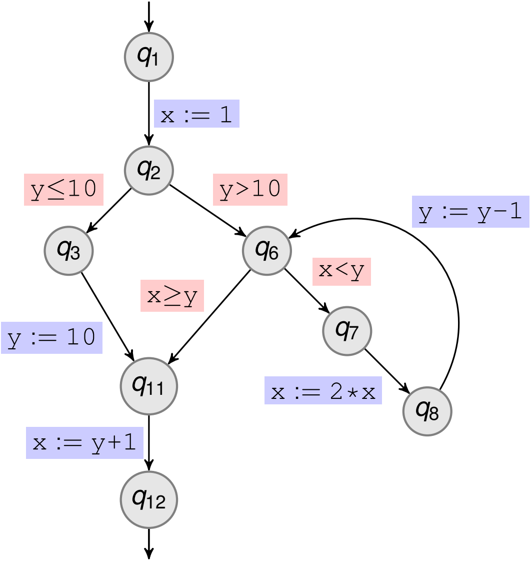
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $2^6$ | 4 | $+\infty$ |
| $q_7$ | 1 | $2^6$ | 4 | $+\infty$ |
| $q_8$ | 2 | $2^7$ | 4 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

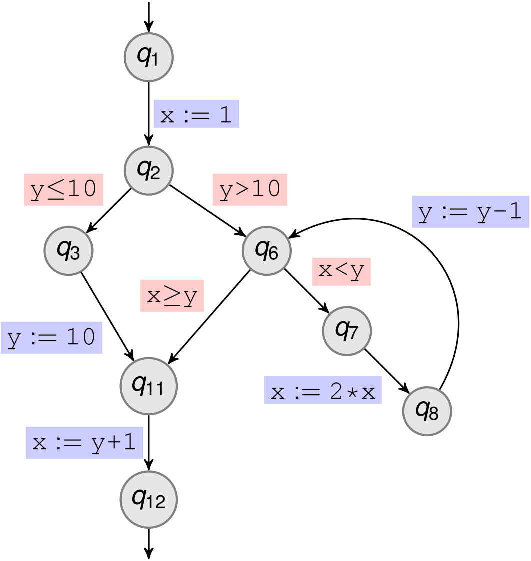
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

|  | x |  | y |  |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $2^7$ | 3 | $+\infty$ |
| $q_7$ | 1 | $2^7$ | 3 | $+\infty$ |
| $q_8$ | 2 | $2^8$ | 3 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

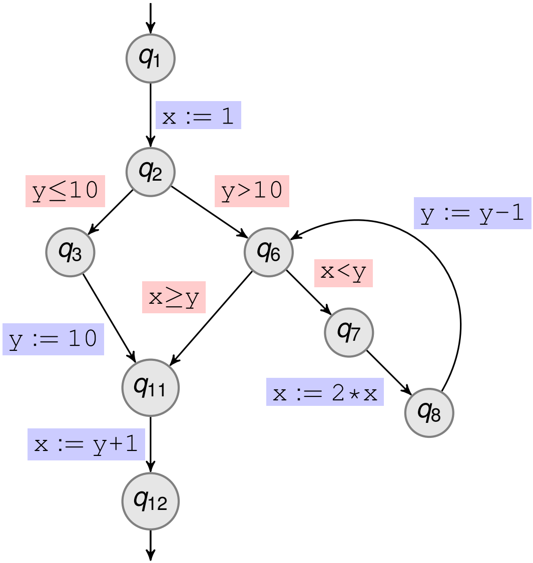
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $2^8$ | 2 | $+\infty$ |
| $q_7$ | 1 | $2^8$ | 2 | $+\infty$ |
| $q_8$ | 2 | $2^9$ | 2 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

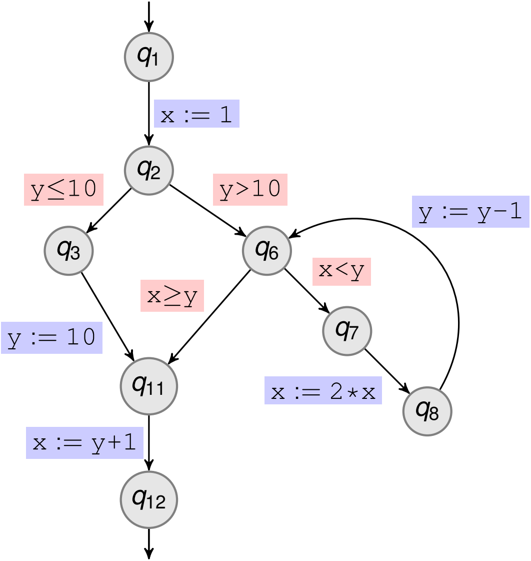
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $2^9$ | 1 | $+\infty$ |
| $q_7$ | 1 | $2^9$ | 1 | $+\infty$ |
| $q_8$ | 2 | $2^{10}$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

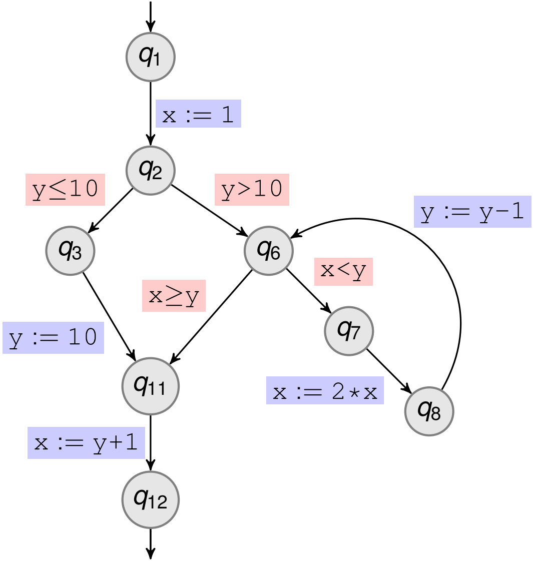
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $2^9$ | 1 | $+\infty$ |
| $q_7$ | 1 | $2^9$ | 1 | $+\infty$ |
| $q_8$ | 2 | $2^{10}$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

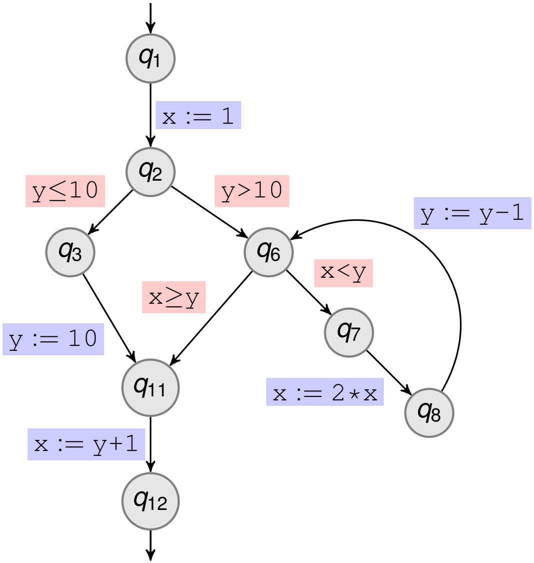
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $2^{10}$ | 0 | $+\infty$ |
| $q_7$ | 1 | $2^{9}$ | 1 | $+\infty$ |
| $q_8$ | 2 | $2^{10}$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

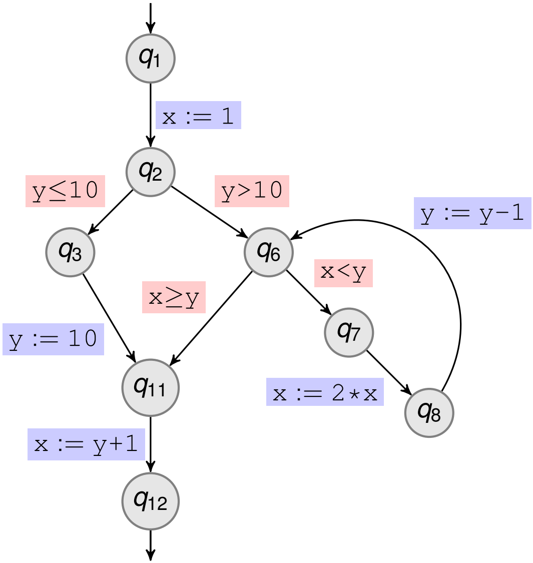
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $2^{10}$ | 0 | $+\infty$ |
| $q_7$ | 1 | $2^9$ | 1 | $+\infty$ |
| $q_8$ | 2 | $2^{10}$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

# Range Analysis on Running Example

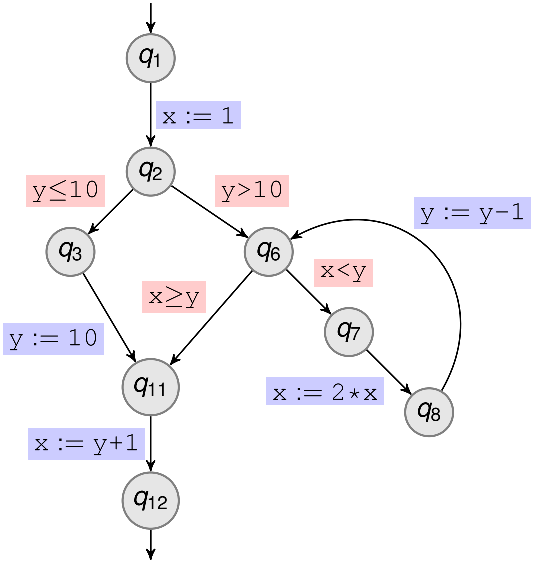


Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $2^{10}$ | 0 | $+\infty$ |
| $q_7$ | 1 | $2^{10}$ | 1 | $+\infty$ |
| $q_8$ | 2 | $2^{10}$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

Gained from guard $x < y$

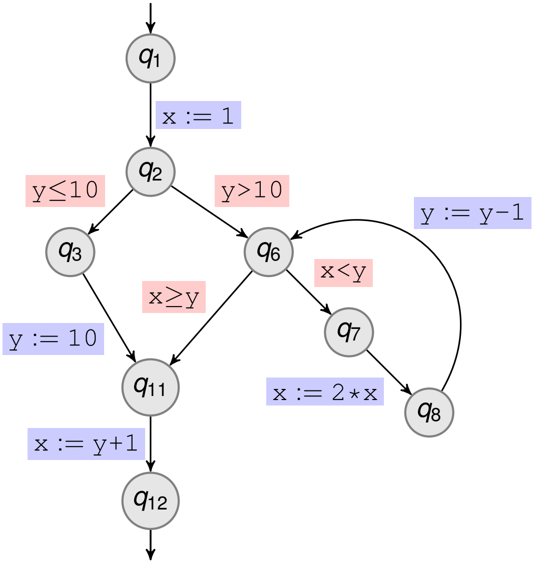
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $2^{10}$ | 0 | $+\infty$ |
| $q_7$ | 1 | $2^{10}$ | 1 | $+\infty$ |
| $q_8$ | 2 | $2^{10}$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

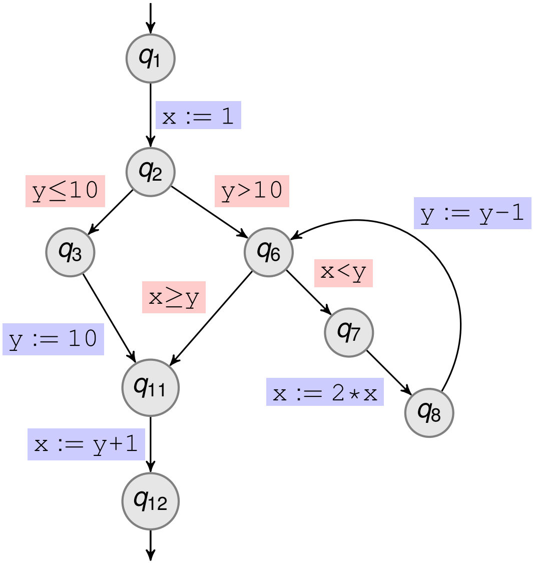
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $2^{10}$ | 0 | $+\infty$ |
| $q_7$ | 1 | $2^{10}$ | 1 | $+\infty$ |
| $q_8$ | 2 | $2^{11}$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

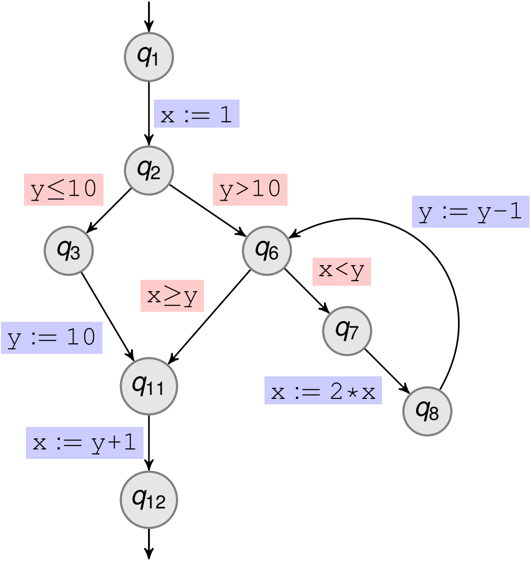
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $2^{10}$ | 0 | $+\infty$ |
| $q_7$ | 1 | $2^{10}$ | 1 | $+\infty$ |
| $q_8$ | 2 | $2^{11}$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

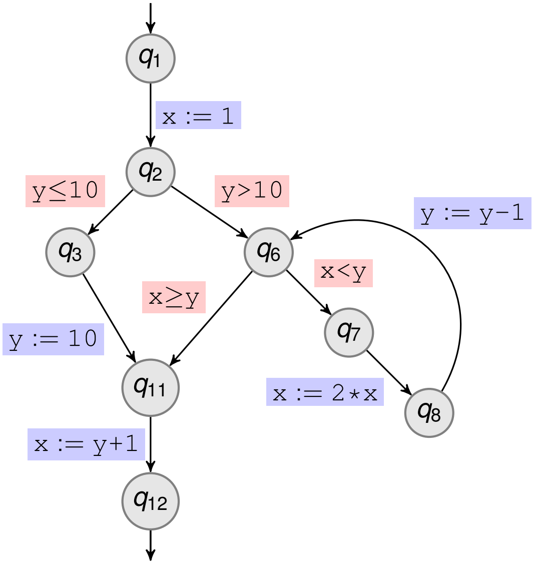
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $2^{11}$ | 0 | $+\infty$ |
| $q_7$ | 1 | $2^{11}$ | 1 | $+\infty$ |
| $q_8$ | 2 | $2^{12}$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

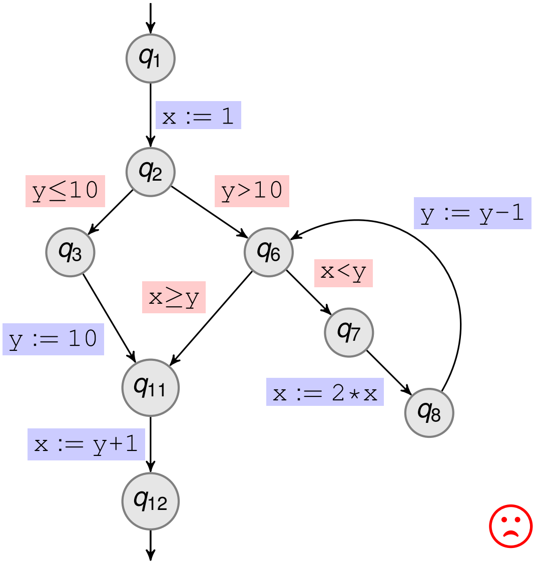
# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $2^{12}$ | 0 | $+\infty$ |
| $q_7$ | 1 | $2^{12}$ | 1 | $+\infty$ |
| $q_8$ | 2 | $2^{13}$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

# Range Analysis on Running Example



Recall that $\top = (-\infty, +\infty)$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $\cdots$ | 0 | $+\infty$ |
| $q_7$ | 1 | $\cdots$ | 1 | $+\infty$ |
| $q_8$ | 2 | $\cdots$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

☹ Does not converge!

# Dynamic Approximation: Widening Operators

Consider a complete lattice $(L, \sqsubseteq)$.

## Objective of Widening Operators

Soundly extrapolate “limits” of ascending chains

## Definition

A widening operator for $(L, \sqsubseteq)$ is a function $\nabla : (L \times L) \to L$ such that:

1. $$x \sqcup y \;\sqsubseteq\; x \nabla y \qquad \text{(for all } x, y \in L\text{)}$$
2. for any ascending chain $x_0 \sqsubseteq x_1 \sqsubseteq \cdots$ of elements of $L$, the ascending chain $y_0 \sqsubseteq y_1 \sqsubseteq \cdots$ defined by

   $$\begin{cases} y_0 \;=\; x_0 \\ y_{i+1} = y_i \nabla x_{i+1} \quad \text{for all } i \in \mathbb{N} \end{cases}$$

   is not strictly increasing (i.e. $y_{i+1} = y_i$ for some $i \in \mathbb{N}$).

# Correctness of Kleene Iteration with Widening

Consider a complete lattice $(L, \sqsubseteq)$ and a monotonic function $f : L \rightarrow L$.

## Theorem

*If $\nabla : (L \times L) \rightarrow L$ is a widening operator then the ascending chain $x_0 \sqsubseteq x_1 \sqsubseteq \cdots$ defined by*

$$
\begin{array}{rcl}
x_0 & = & \bot \\
x_{i+1} & = & \begin{cases} x_i & \text{if } f(x_i) \sqsubseteq x_i \\ x_i \nabla f(x_{i+1}) & \text{otherwise} \end{cases}
\end{array}
$$

*is eventually stationary, and its limit satisfies $\bigsqcup \{x_i \mid i \in \mathbb{N}\} \;\sqsupseteq\; \mathrm{lfp}(f)$.*

## Application to MFP Approximation in Data Flow Analysis

Replacing $\sqcup$ with $\nabla$ in Kleene / round-robin / worklist algorithms

- guarantees termination, but
- at the expense of precision.

# Correctness of Kleene Iteration with Widening

Consider a complete lattice $(L, \sqsubseteq)$ and a monotonic function $f : L \rightarrow L$.

## Theorem

*If $\nabla : (L \times L) \rightarrow L$ is a widening operator then the ascending chain $x_0 \sqsubseteq x_1 \sqsubseteq \cdots$ defined by*

$$\begin{aligned} x_0 &= \bot \\ x_{i+1} &= \begin{cases} x_i & \textit{if } f(x_i) \sqsubseteq x_i \\ x_i \nabla f(x_{i+1}) & \textit{otherwise} \end{cases} \end{aligned}$$

*is eventually stationary, and its limit satisfies $\bigsqcup \{x_i \mid i \in \mathbb{N}\} \sqsupseteq \mathrm{lfp}(f)$.*

## Application to MFP Approximation in Data Flow Analysis

Replacing $\sqcup$ with $\nabla$ in Kleene / round-robin / worklist algorithms

- guarantees termination, but
- at the expense of precision.

# (Forward) Round-Robin Iteration with Widening

Consider a data flow instance $\langle (L, \sqsubseteq), \mathcal{F}, Q, q_{in}, q_{out}, \mathrm{X}, \rightarrow, f, \imath \rangle$.

```
foreach $q \in Q$
    a[$q$] $\leftarrow$ $\bot$
a[$q_{in}$] $\leftarrow$ $\imath$
do
    change $\leftarrow$ false
    foreach $q \xrightarrow{\mathrm{op}} q'$
        new $\leftarrow$ $f_{\mathrm{op}}$(a[$q$])
        if new $\not\sqsubseteq$ a[$q'$]
            a[$q'$] $\leftarrow$ a[$q'$] $\nabla$ new
            change $\leftarrow$ true
while change
return a
```

If $\nabla$ is a widening operator on $(L, \sqsubseteq)$ then:

- this algorithm terminates for any data flow instance on $(L, \sqsubseteq)$.
- the returned $a \in Q \rightarrow L$ satisfies:

$$\overrightarrow{\mathrm{MFP}}(q) \;\sqsubseteq\; a(q)$$

for every $q \in Q$.

# Widening Operator for Range Analysis: Intuition

## Objective of Widening Operators

Soundly extrapolate “limits” of ascending chains

Put $\infty$ when the bound is moving towards $\infty$

## Examples

$$\begin{array}{rcl}
\ldots,\ (1,2),\ (1,3),\ (1,4) & \longrightarrow & (1,+\infty) \\
\ldots,\ (1,2),\ (-1,2),\ (-6,2) & \longrightarrow & (-\infty,2) \\
\ldots,\ (1,2),\ (-9,3),\ (-19,4) & \longrightarrow & (-\infty,+\infty) \\
\ldots,\ (1,+\infty),\ (-9,+\infty),\ (-19,+\infty) & \longrightarrow & (-\infty,+\infty)
\end{array}$$

$\nabla$ only looks at the last two elements of the sequence

# Widening Operator for Range Analysis

## Widening Operator on the Complete Lattice ($Int, \sqsubseteq$) of Intervals

$$\bot \nabla \bot \;=\; \bot \qquad\qquad \bot \nabla (l, u) \;=\; (l, u) \nabla \bot \;=\; (l, u)$$

$$(l_1, u_1) \nabla (l_2, u_2) \;=\; (l_\nabla, u_\nabla) \quad \text{where} \quad \begin{cases} l_\nabla \;=\; \begin{cases} -\infty & \text{if } l_2 < l_1 \\ l_1 & \text{otherwise} \end{cases} \\ u_\nabla \;=\; \begin{cases} +\infty & \text{if } u_2 > u_1 \\ u_1 & \text{otherwise} \end{cases} \end{cases}$$

## Widening Operator on the Complete Lattice ($\mathbb{X} \rightarrow Int, \overline{\sqsubseteq}$)

Extension $\overline{\nabla}$ of the widening $\nabla$ on ($Int, \sqsubseteq$) to ($\mathbb{X} \rightarrow Int, \overline{\sqsubseteq}$), defined by:

$$\overline{v_1} \,\overline{\nabla}\, \overline{v_2} \;=\; \lambda q \,.\, \overline{v_1}(q) \nabla \overline{v_2}(q)$$

# Widening Operator for Range Analysis

## Widening Operator on the Complete Lattice $(Int, \sqsubseteq)$ of Intervals

$$\bot \nabla \bot \; = \; \bot \qquad \bot \nabla (l, u) \; = \; (l, u) \nabla \bot \; = \; (l, u)$$

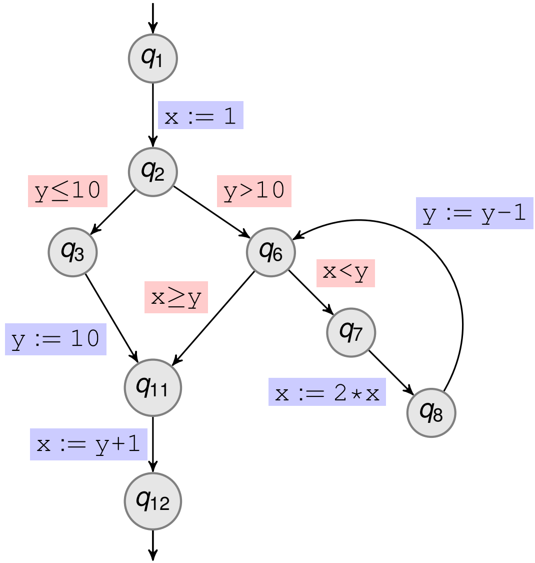
$$(l_1, u_1) \nabla (l_2, u_2) \; = \; (l_\nabla, u_\nabla) \quad \text{where} \quad \begin{cases} l_\nabla & = \begin{cases} -\infty & \text{if } l_2 < l_1 \\ l_1 & \text{otherwise} \end{cases} \\ u_\nabla & = \begin{cases} +\infty & \text{if } u_2 > u_1 \\ u_1 & \text{otherwise} \end{cases} \end{cases}$$

## Widening Operator on the Complete Lattice $(\mathrm{X} \to Int, \overline{\sqsubseteq})$

Extension $\overline{\nabla}$ of the widening $\nabla$ on $(Int, \sqsubseteq)$ to $(\mathrm{X} \to Int, \overline{\sqsubseteq})$, defined by:

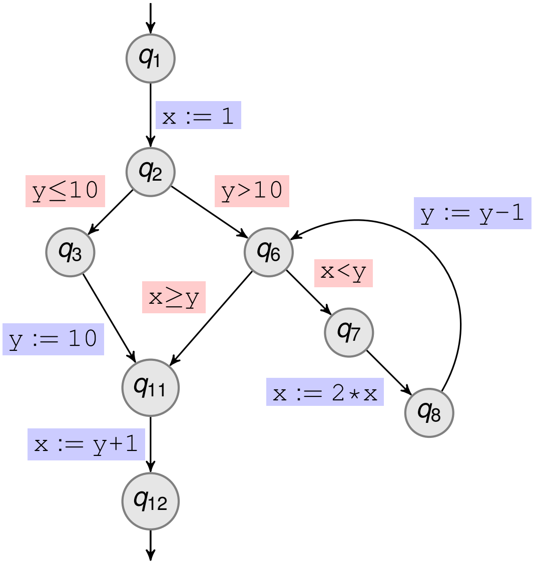
$$\overline{v_1} \; \overline{\nabla} \; \overline{v_2} \; = \; \lambda q \, . \, \overline{v_1}(q) \nabla \overline{v_2}(q)$$

# Range Analysis on Example with Widening



Show that $x > 0$ at $q_{12}$

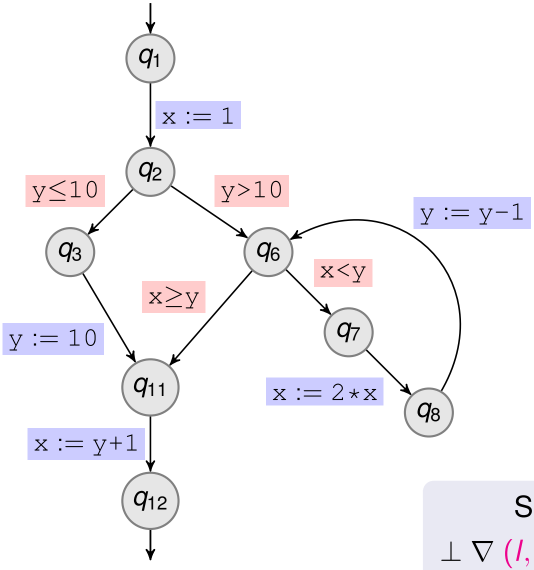
# Range Analysis on Example with Widening



Show that x > 0 at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_3$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_6$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_7$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_8$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_{11}$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_{12}$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |

# Range Analysis on Example with Widening



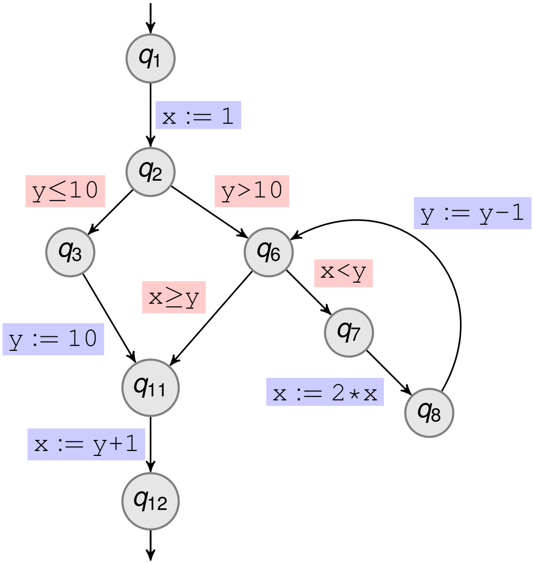
Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 1 | 10 | $+\infty$ |
| $q_7$ | 1 | 1 | 10 | $+\infty$ |
| $q_8$ | 2 | 2 | 10 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

Same as without $\nabla$ since

$$\bot \, \nabla \, (l, u) \quad = \quad \bot \sqcup (l, u) \quad = \quad (l, u)$$

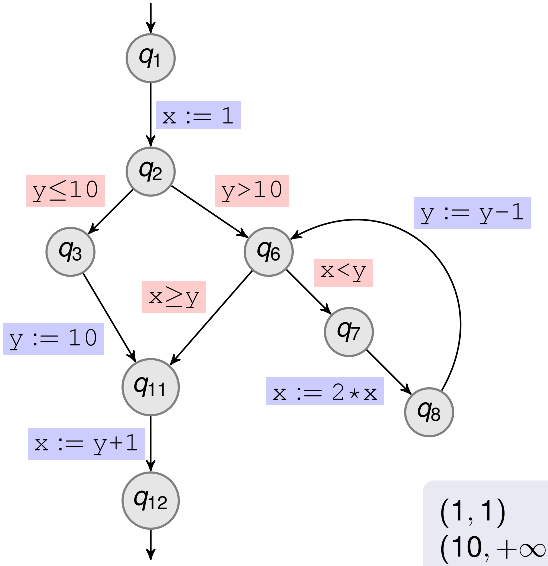
# Range Analysis on Example with Widening



Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 1 | 10 | $+\infty$ |
| $q_7$ | 1 | 1 | 10 | $+\infty$ |
| $q_8$ | 2 | 2 | 10 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

# Range Analysis on Example with Widening

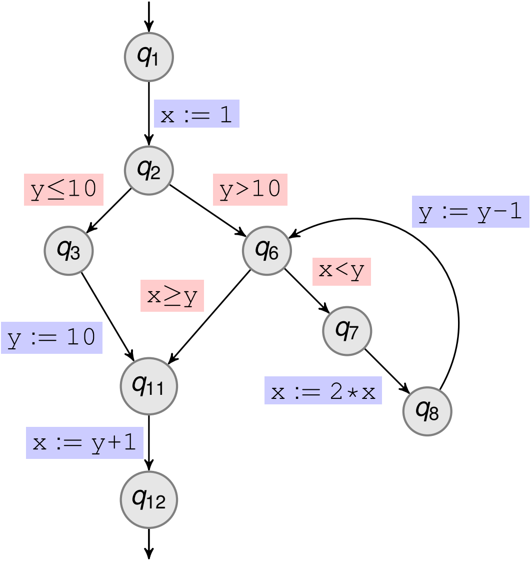


Show that $\mathtt{x} > 0$ at $q_{12}$

|  | x |  | y |  |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_7$ | 1 | 1 | 10 | $+\infty$ |
| $q_8$ | 2 | 2 | 10 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

$$(1,1) \;\nabla\; (2,2) = (1,+\infty)$$
$$(10,+\infty) \;\nabla\; (9,+\infty) = (-\infty,+\infty)$$

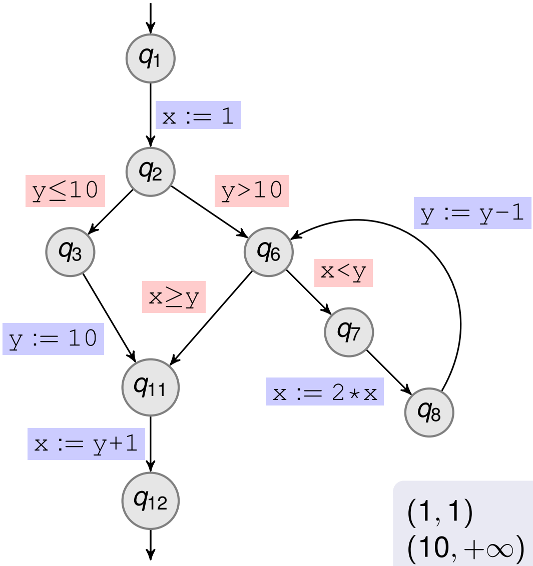
# Range Analysis on Example with Widening



Show that x > 0 at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_7$ | 1 | 1 | 10 | $+\infty$ |
| $q_8$ | 2 | 2 | 10 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

# Range Analysis on Example with Widening

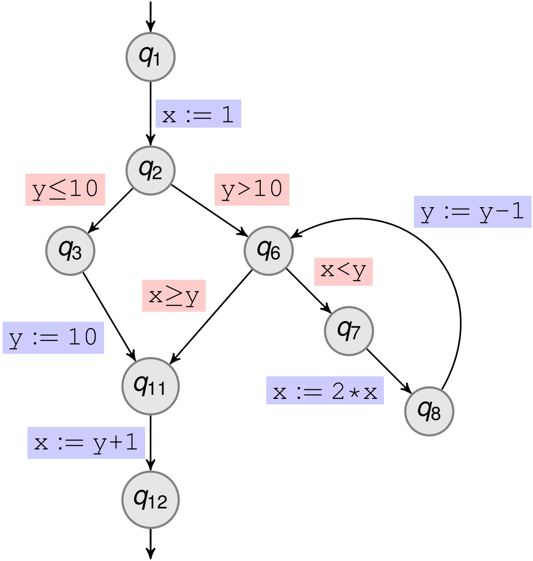


Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_7$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_8$ | 2 | 2 | 10 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

$$\begin{aligned} (1,1) \quad &\nabla \quad (1,+\infty) &&= \quad (1,+\infty) \\ (10,+\infty) \quad &\nabla \quad (-\infty,+\infty) &&= \quad (-\infty,+\infty) \end{aligned}$$

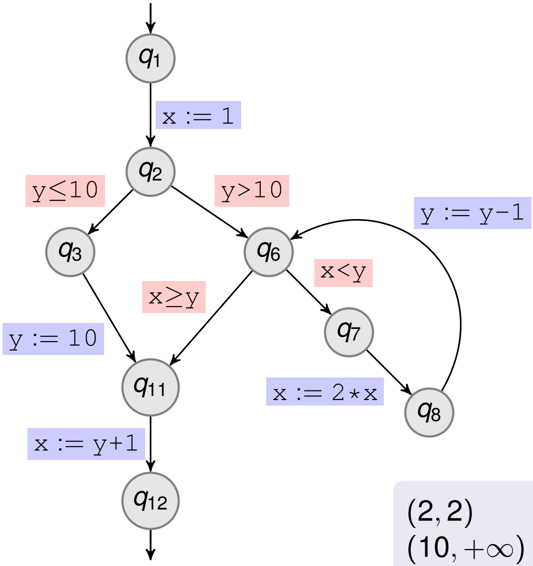
# Range Analysis on Example with Widening



Show that x > 0 at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_7$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_8$ | 2 | 2 | 10 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

# Range Analysis on Example with Widening

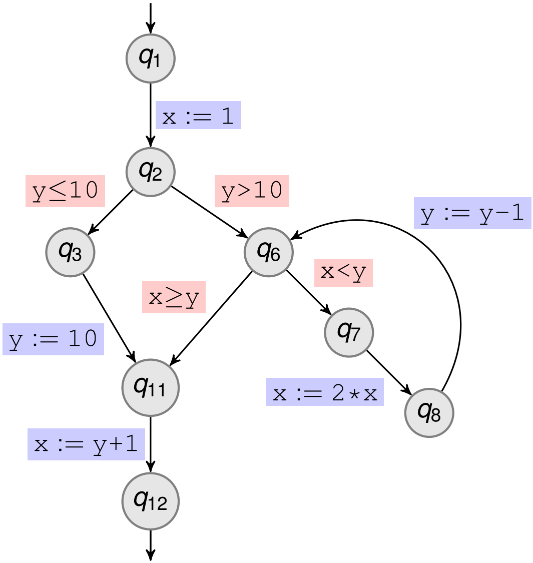


Show that $x > 0$ at $q_{12}$

| | $x$ | | $y$ | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_7$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_8$ | 2 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

$$\begin{aligned}
(2,2) \quad &\nabla \quad (2,+\infty) &&= (2,+\infty) \\
(10,+\infty) \quad &\nabla \quad (-\infty,+\infty) &&= (-\infty,+\infty)
\end{aligned}$$

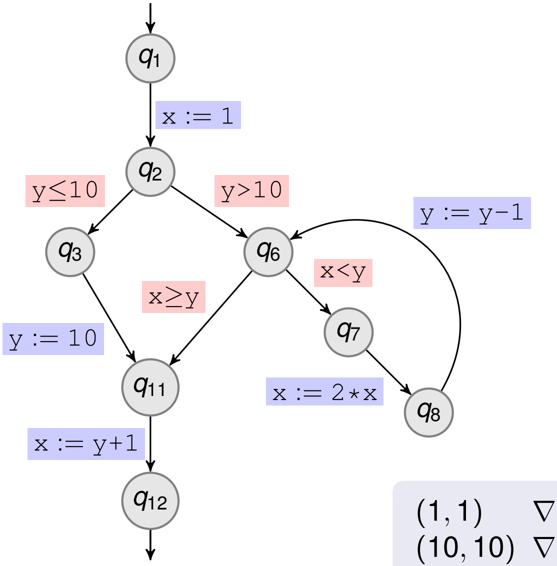
# Range Analysis on Example with Widening



Show that x > 0 at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_7$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_8$ | 2 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

# Range Analysis on Example with Widening

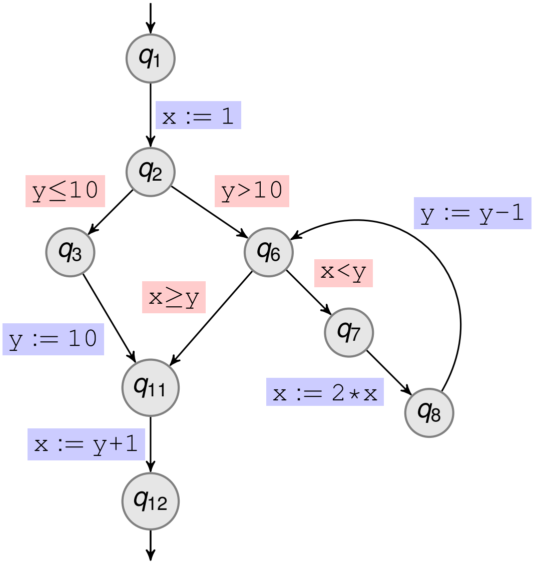


Show that $\mathtt{x} > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_7$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_8$ | 2 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_{11}$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_{12}$ | 11 | 11 | 10 | 10 |

$$\begin{array}{llll}(1,1) & \nabla\ (1,+\infty) & = & (1,+\infty) \\ (10,10) & \nabla\ (-\infty,+\infty) & = & (-\infty,+\infty)\end{array}$$

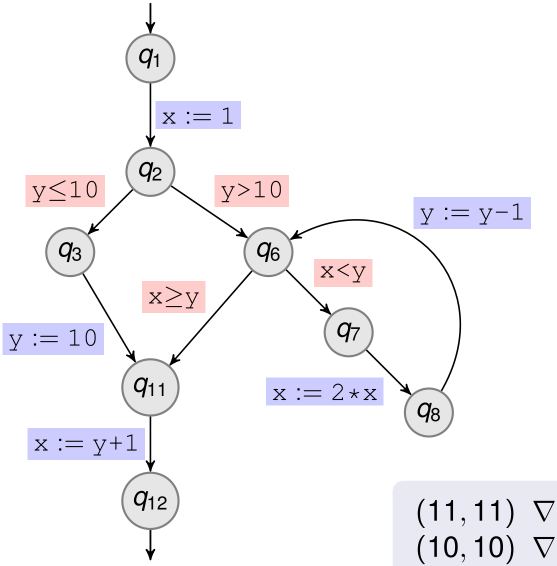
# Range Analysis on Example with Widening



Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_7$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_8$ | 2 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_{11}$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_{12}$ | 11 | 11 | 10 | 10 |

# Range Analysis on Example with Widening

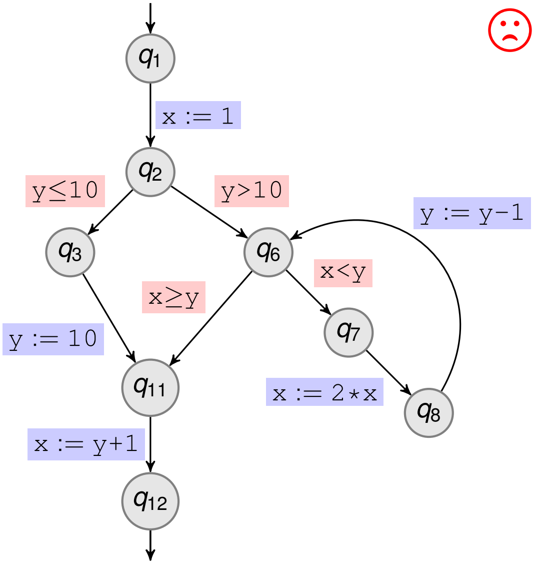


Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_7$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_8$ | 2 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_{11}$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_{12}$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |

$$\begin{array}{lcl}
(11,11) \ \nabla \ (-\infty,+\infty) & = & (-\infty,+\infty) \\
(10,10) \ \nabla \ (-\infty,+\infty) & = & (-\infty,+\infty)
\end{array}$$

# Range Analysis on Example with Widening



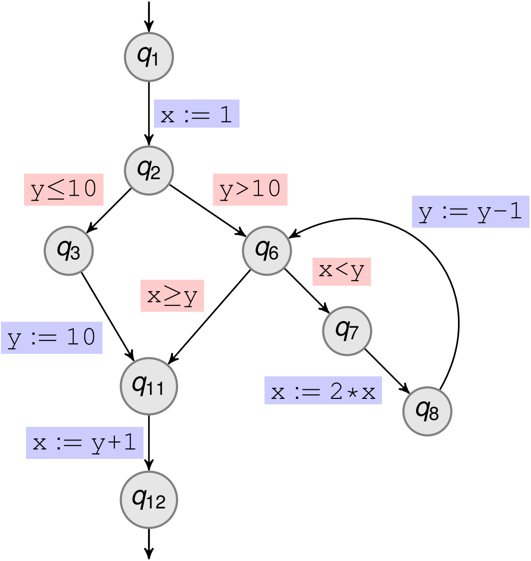
☹

Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_7$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_8$ | 2 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_{11}$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_{12}$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |

Too coarse!

# Range Analysis on Example with Delayed Widening

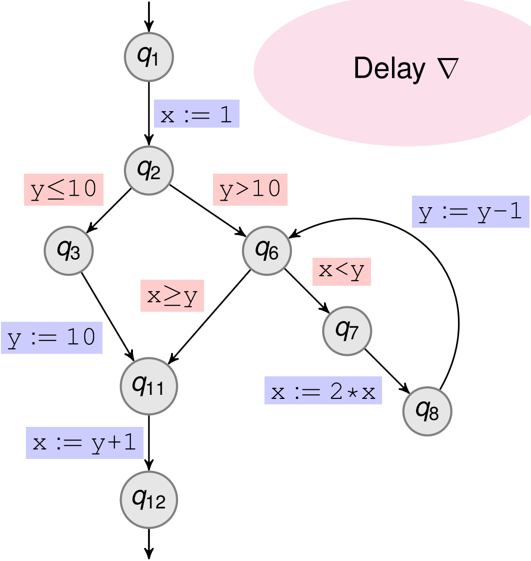


Show that $x > 0$ at $q_{12}$

## Delayed Widening

1. Keep $\sqcup$ for the first iterations
2. Track number of “updates” for each location
3. Switch to $\nabla$ after a suitable “delay”

# Range Analysis on Example with Delayed Widening

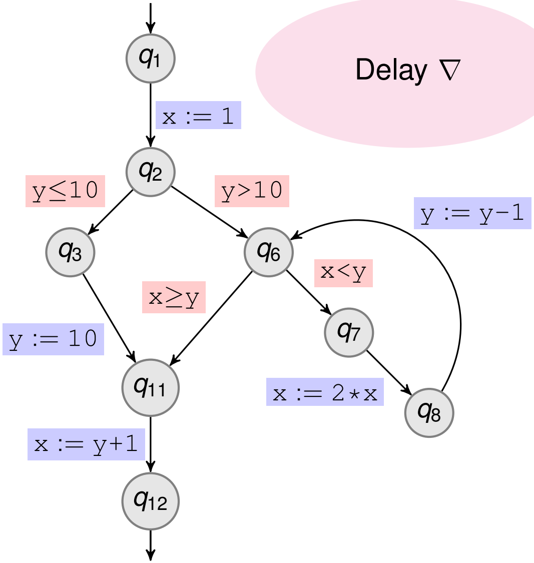


Delay ∇

Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_3$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_6$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_7$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_8$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_{11}$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_{12}$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |

# Range Analysis on Example with Delayed Widening



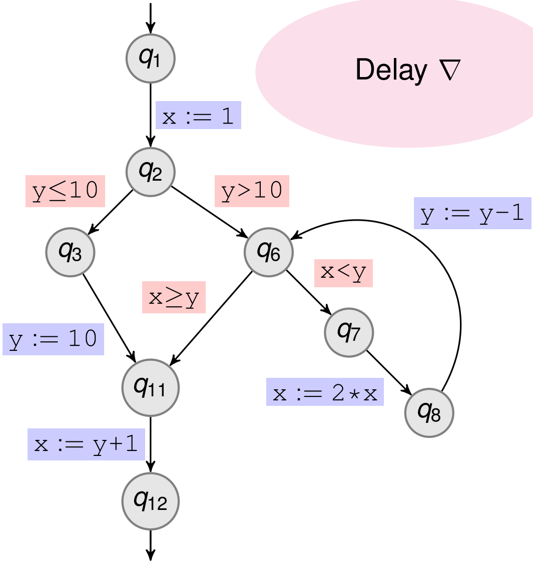
Delay $\nabla$

Show that $\mathtt{x} > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 1 | 10 | $+\infty$ |
| $q_7$ | 1 | 1 | 10 | $+\infty$ |
| $q_8$ | 2 | 2 | 10 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

Same as without $\nabla$

# Range Analysis on Example with Delayed Widening

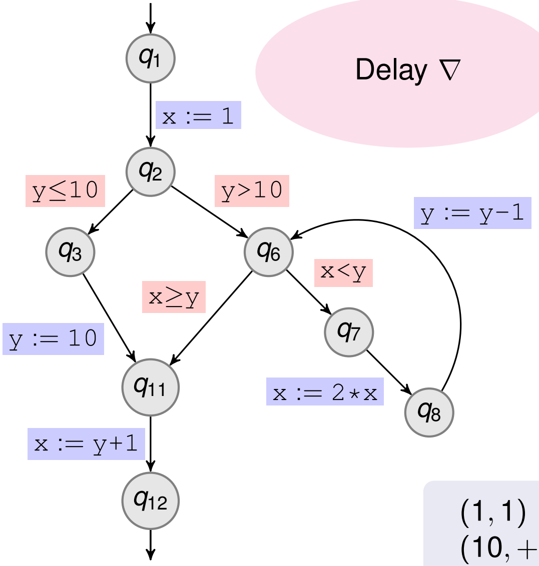


Delay $\nabla$

Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 1 | 10 | $+\infty$ |
| $q_7$ | 1 | 1 | 10 | $+\infty$ |
| $q_8$ | 2 | 2 | 10 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

# Range Analysis on Example with Delayed Widening



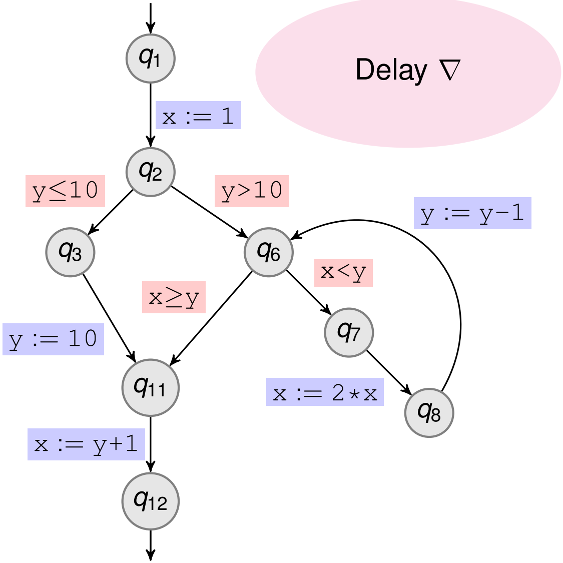
Delay ∇

Show that x $>$ 0 at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 2 | 9 | $+\infty$ |
| $q_7$ | 1 | 1 | 10 | $+\infty$ |
| $q_8$ | 2 | 2 | 10 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

$$\begin{array}{llll} (1,1) & \sqcup\ (2,2) & = & (1,2) \\ (10,+\infty) & \sqcup\ (9,+\infty) & = & (9,+\infty) \end{array}$$

# Range Analysis on Example with Delayed Widening

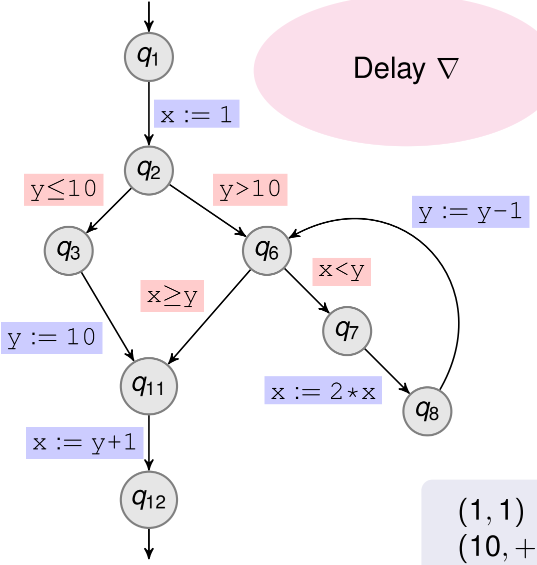


Delay ∇

Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 2 | 9 | $+\infty$ |
| $q_7$ | 1 | 1 | 10 | $+\infty$ |
| $q_8$ | 2 | 2 | 10 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

# Range Analysis on Example with Delayed Widening



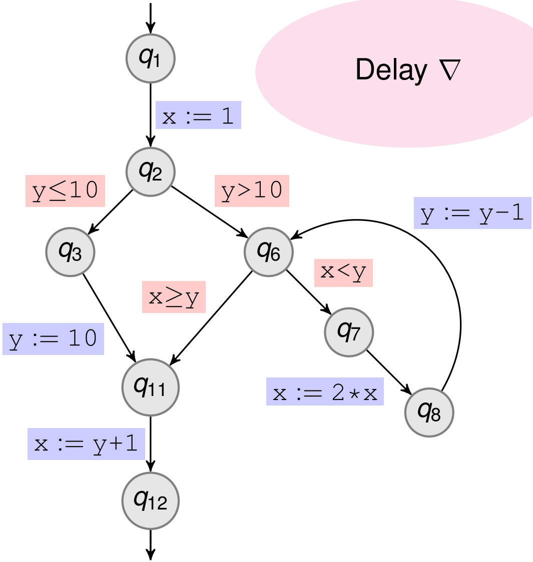
Delay $\nabla$

Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 2 | 9 | $+\infty$ |
| $q_7$ | 1 | 2 | 9 | $+\infty$ |
| $q_8$ | 2 | 2 | 10 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

$$\begin{array}{lclcl} (1,1) & \sqcup & (1,2) & = & (1,2) \\ (10,+\infty) & \sqcup & (9,+\infty) & = & (9,+\infty) \end{array}$$

# Range Analysis on Example with Delayed Widening

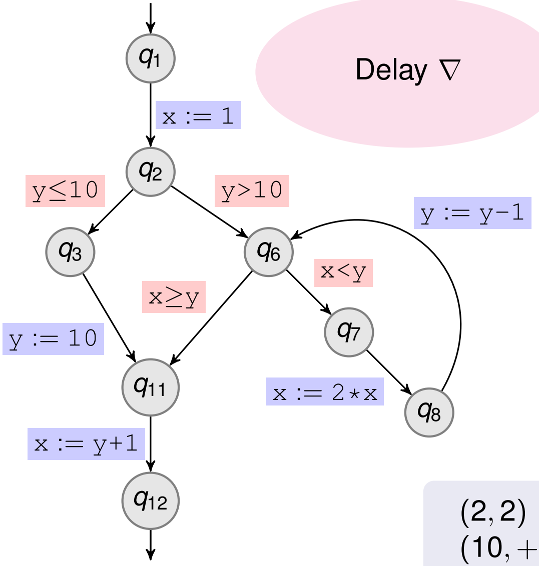


Delay ∇

Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 2 | 9 | $+\infty$ |
| $q_7$ | 1 | 2 | 9 | $+\infty$ |
| $q_8$ | 2 | 2 | 10 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

# Range Analysis on Example with Delayed Widening

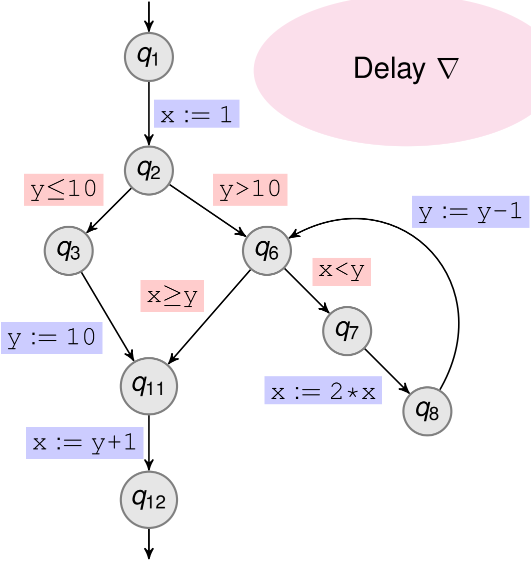


Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 2 | 9 | $+\infty$ |
| $q_7$ | 1 | 2 | 9 | $+\infty$ |
| $q_8$ | 2 | 4 | 9 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

$$\begin{array}{llll} (2,2) & \sqcup\ (2,4) & = & (2,4) \\ (10,+\infty) & \sqcup\ (9,+\infty) & = & (9,+\infty) \end{array}$$

# Range Analysis on Example with Delayed Widening

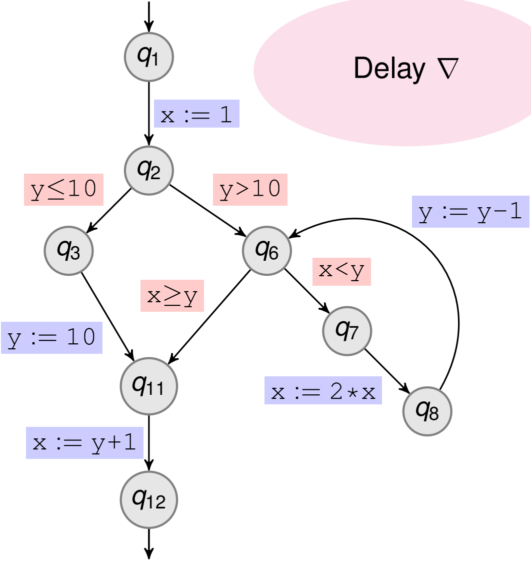


Delay $\nabla$

Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 2 | 9 | $+\infty$ |
| $q_7$ | 1 | 2 | 9 | $+\infty$ |
| $q_8$ | 2 | 4 | 9 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

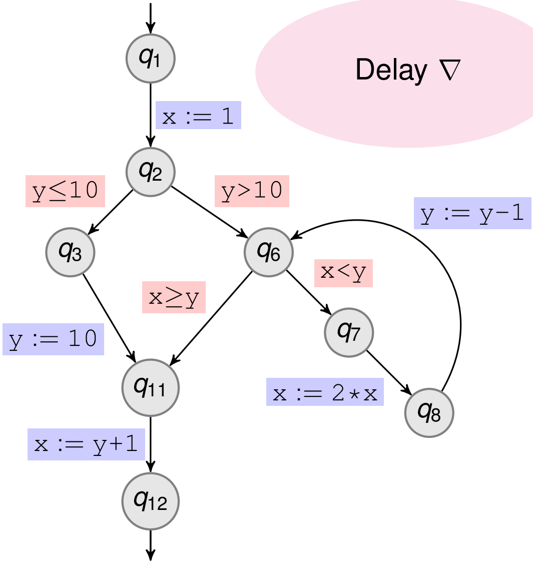
# Range Analysis on Example with Delayed Widening



Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 4 | 8 | $+\infty$ |
| $q_7$ | 1 | 4 | 8 | $+\infty$ |
| $q_8$ | 2 | 8 | 8 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

# Range Analysis on Example with Delayed Widening

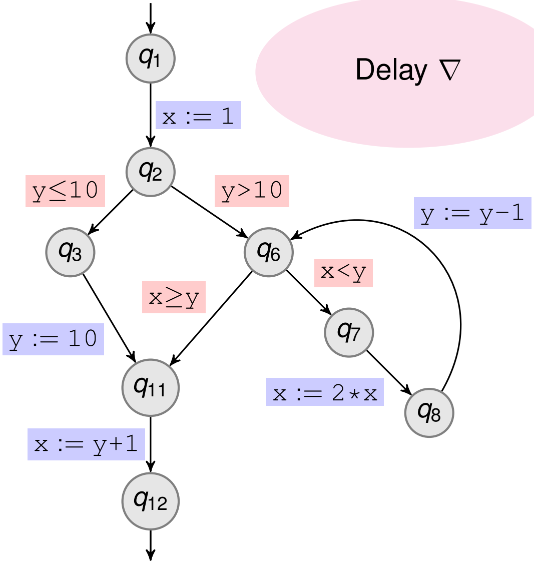


Delay ∇

Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 8 | 7 | $+\infty$ |
| $q_7$ | 1 | 8 | 7 | $+\infty$ |
| $q_8$ | 2 | 16 | 7 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

# Range Analysis on Example with Delayed Widening

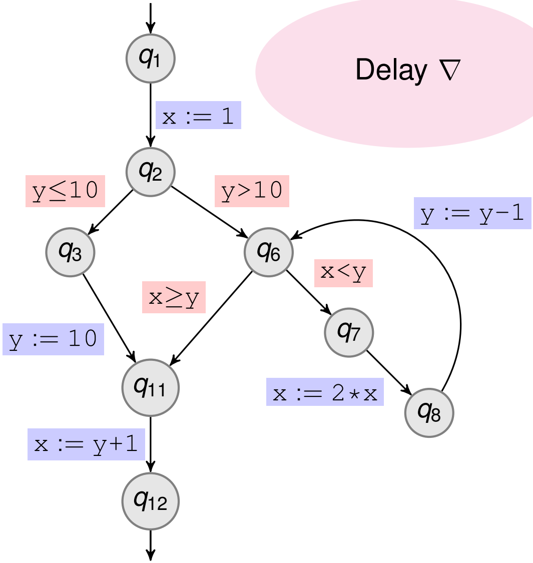


Delay ∇

Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 16 | 6 | $+\infty$ |
| $q_7$ | 1 | 16 | 6 | $+\infty$ |
| $q_8$ | 2 | 32 | 6 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

# Range Analysis on Example with Delayed Widening

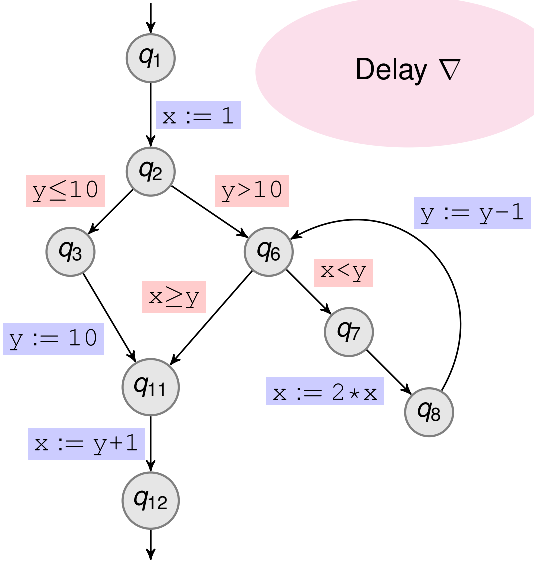


Delay $\nabla$

Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 32 | 5 | $+\infty$ |
| $q_7$ | 1 | 32 | 5 | $+\infty$ |
| $q_8$ | 2 | $2^6$ | 5 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

# Range Analysis on Example with Delayed Widening



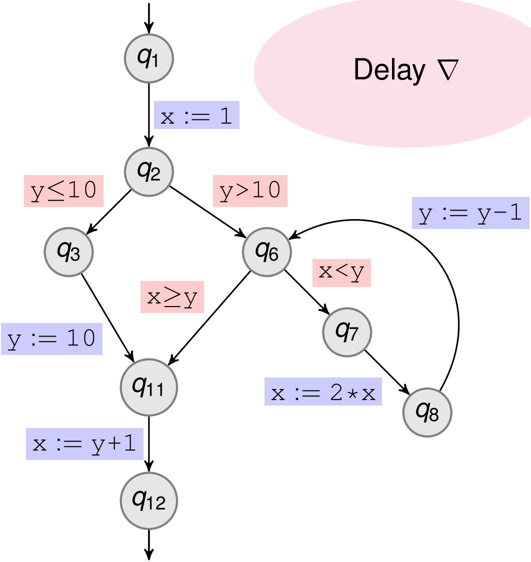
Delay ∇

Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $2^6$ | 4 | $+\infty$ |
| $q_7$ | 1 | $2^6$ | 4 | $+\infty$ |
| $q_8$ | 2 | $2^7$ | 4 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

# Range Analysis on Example with Delayed Widening

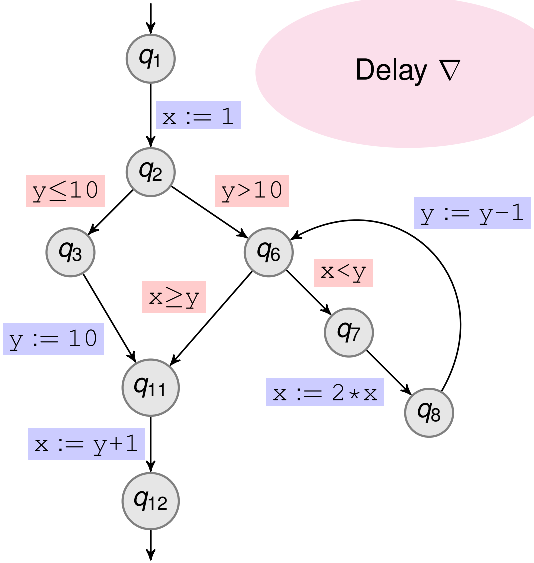
Delay ∇



Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $2^7$ | 3 | $+\infty$ |
| $q_7$ | 1 | $2^7$ | 3 | $+\infty$ |
| $q_8$ | 2 | $2^8$ | 3 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

# Range Analysis on Example with Delayed Widening

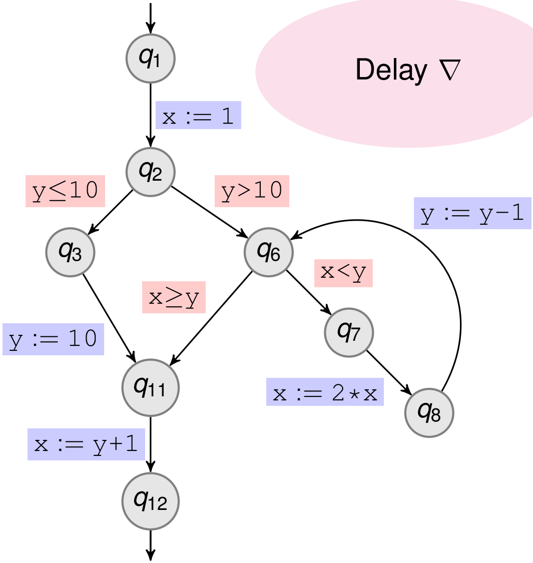


Delay ∇

Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $2^8$ | 2 | $+\infty$ |
| $q_7$ | 1 | $2^8$ | 2 | $+\infty$ |
| $q_8$ | 2 | $2^9$ | 2 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

# Range Analysis on Example with Delayed Widening

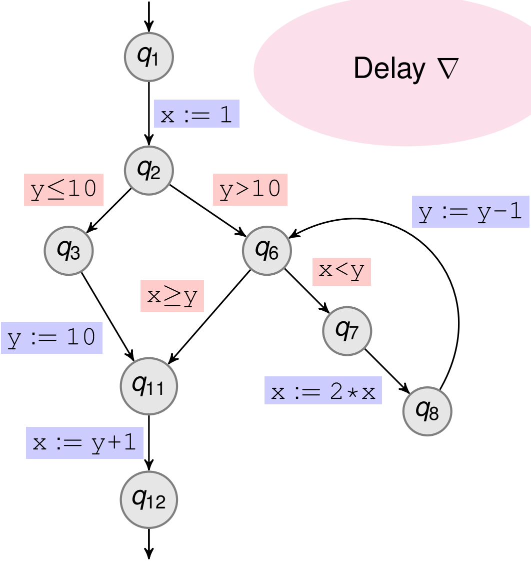


Delay $\nabla$

Show that $\mathtt{x} > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $2^9$ | 1 | $+\infty$ |
| $q_7$ | 1 | $2^9$ | 1 | $+\infty$ |
| $q_8$ | 2 | $2^{10}$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

# Range Analysis on Example with Delayed Widening

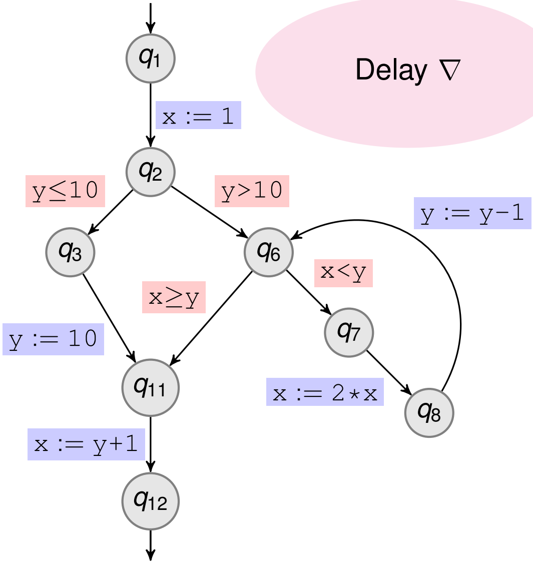


Delay ∇

Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $2^9$ | 1 | $+\infty$ |
| $q_7$ | 1 | $2^9$ | 1 | $+\infty$ |
| $q_8$ | 2 | $2^{10}$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

# Range Analysis on Example with Delayed Widening

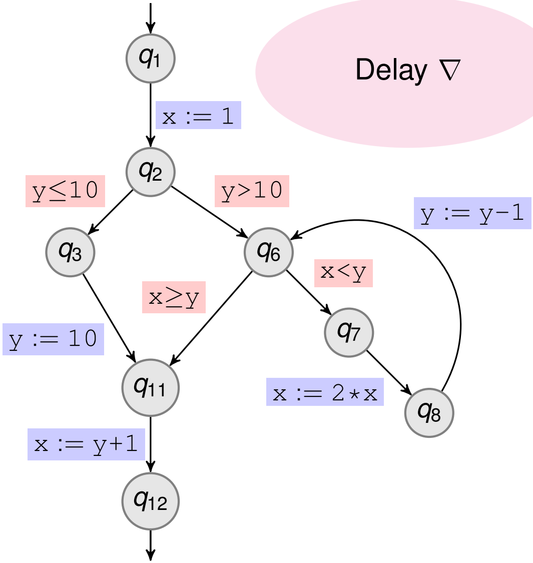


Delay $\nabla$

Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $2^{10}$ | 0 | $+\infty$ |
| $q_7$ | 1 | $2^9$ | 1 | $+\infty$ |
| $q_8$ | 2 | $2^{10}$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

# Range Analysis on Example with Delayed Widening

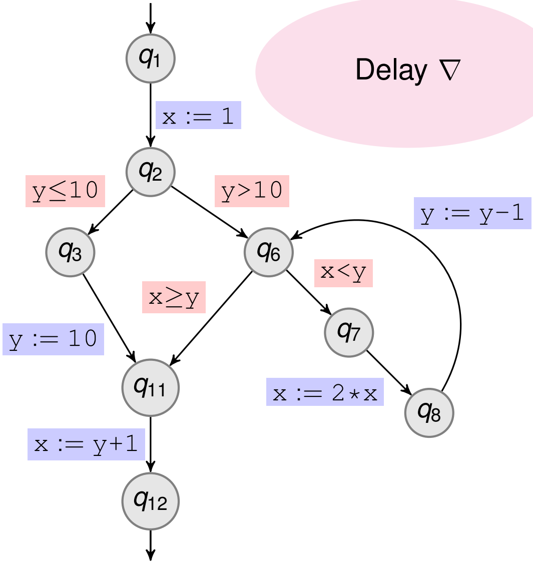


Delay ∇

Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $2^{10}$ | 0 | $+\infty$ |
| $q_7$ | 1 | $2^9$ | 1 | $+\infty$ |
| $q_8$ | 2 | $2^{10}$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

# Range Analysis on Example with Delayed Widening



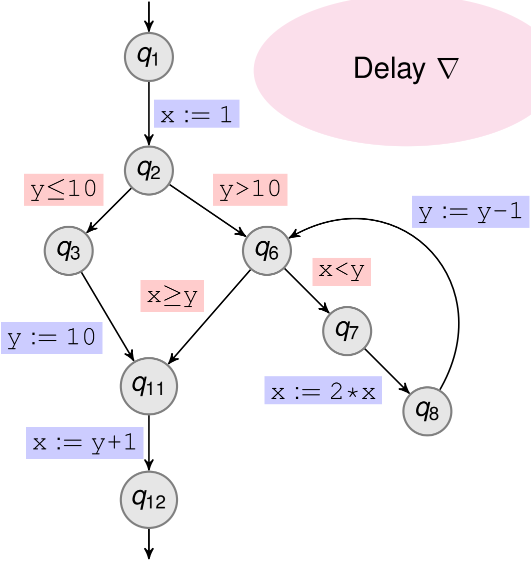
Delay $\nabla$

Show that $\mathtt{x} > 0$ at $q_{12}$

|  | x |  | y |  |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $2^{10}$ | 0 | $+\infty$ |
| $q_7$ | 1 | $2^{10}$ | 1 | $+\infty$ |
| $q_8$ | 2 | $2^{10}$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

Gained from guard $\mathtt{x} < \mathtt{y}$

# Range Analysis on Example with Delayed Widening

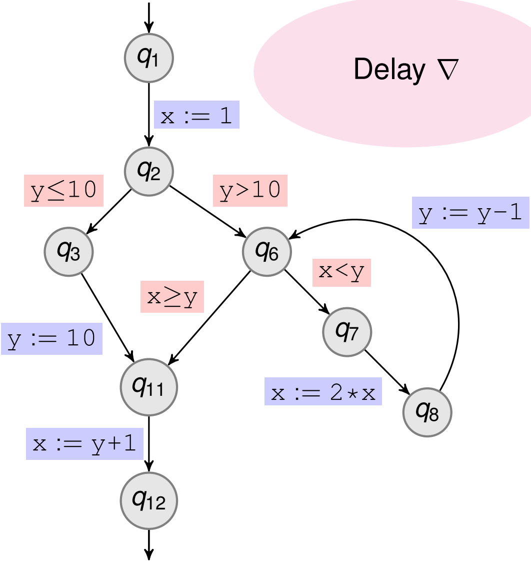


Delay ∇

Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $2^{10}$ | 0 | $+\infty$ |
| $q_7$ | 1 | $2^{10}$ | 1 | $+\infty$ |
| $q_8$ | 2 | $2^{10}$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

# Range Analysis on Example with Delayed Widening



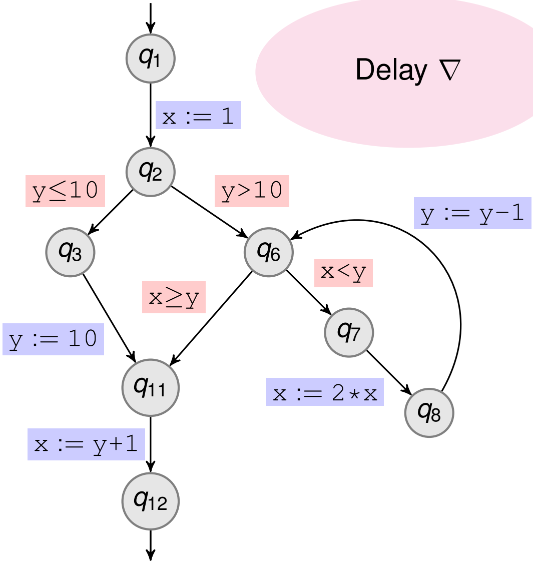
Delay ∇

Show that x > 0 at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $2^{10}$ | 0 | $+\infty$ |
| $q_7$ | 1 | $2^{10}$ | 1 | $+\infty$ |
| $q_8$ | 2 | $2^{11}$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

# Range Analysis on Example with Delayed Widening

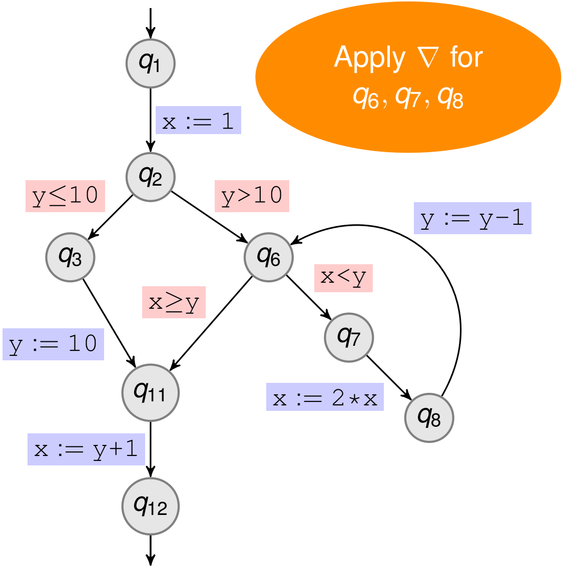
Delay $\nabla$



Show that $\mathtt{x} > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $2^{10}$ | 0 | $+\infty$ |
| $q_7$ | 1 | $2^{10}$ | 1 | $+\infty$ |
| $q_8$ | 2 | $2^{11}$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

# Range Analysis on Example with Delayed Widening

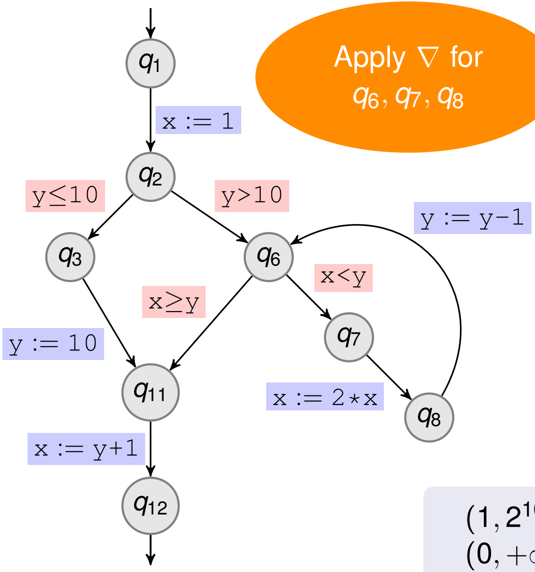


Apply $\nabla$ for $q_6, q_7, q_8$

Show that $\mathtt{x} > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $2^{10}$ | 0 | $+\infty$ |
| $q_7$ | 1 | $2^{10}$ | 1 | $+\infty$ |
| $q_8$ | 2 | $2^{11}$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

# Range Analysis on Example with Delayed Widening



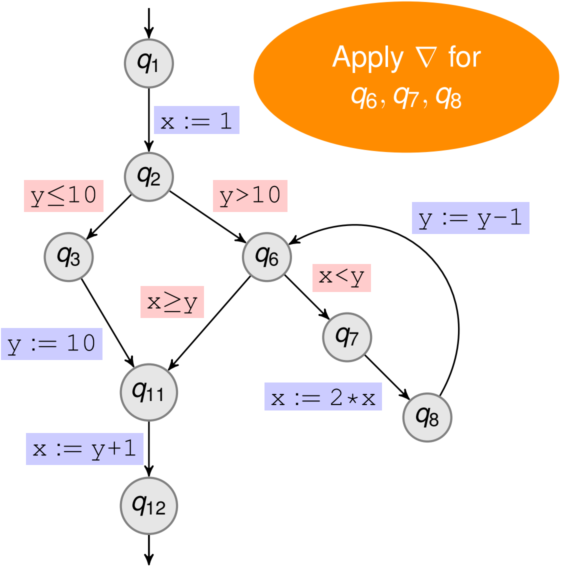
Apply $\nabla$ for $q_6, q_7, q_8$

Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | 0 | $+\infty$ |
| $q_7$ | 1 | $2^{10}$ | 1 | $+\infty$ |
| $q_8$ | 2 | $2^{11}$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

$$(1, 2^{10}) \;\nabla\; (2, 2^{11}) \;=\; (1, +\infty)$$
$$(0, +\infty) \;\nabla\; (0, +\infty) \;=\; (0, +\infty)$$

# Range Analysis on Example with Delayed Widening

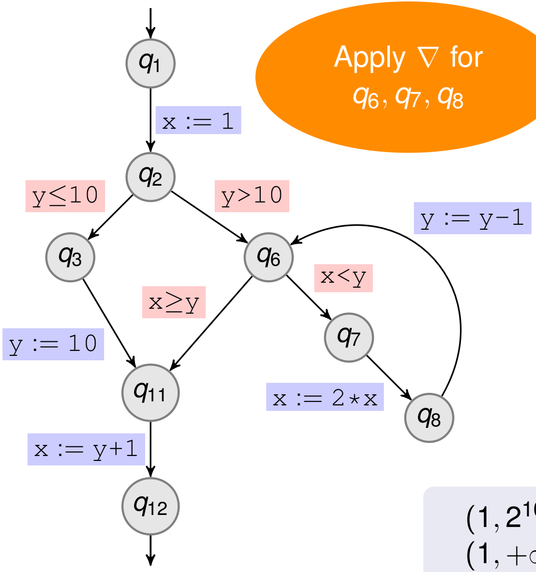


Apply $\nabla$ for $q_6, q_7, q_8$

Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | 0 | $+\infty$ |
| $q_7$ | 1 | $2^{10}$ | 1 | $+\infty$ |
| $q_8$ | 2 | $2^{11}$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

# Range Analysis on Example with Delayed Widening



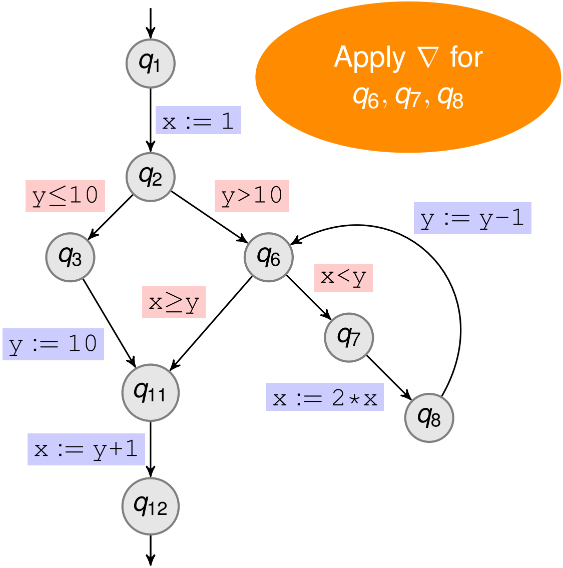
Apply $\nabla$ for $q_6, q_7, q_8$

Show that $\mathtt{x} > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | 0 | $+\infty$ |
| $q_7$ | 1 | $+\infty$ | 1 | $+\infty$ |
| $q_8$ | 2 | $2^{11}$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

$$(1, 2^{10}) \ \nabla \ (1, +\infty) = (1, +\infty)$$
$$(1, +\infty) \ \nabla \ (1, +\infty) = (1, +\infty)$$

# Range Analysis on Example with Delayed Widening

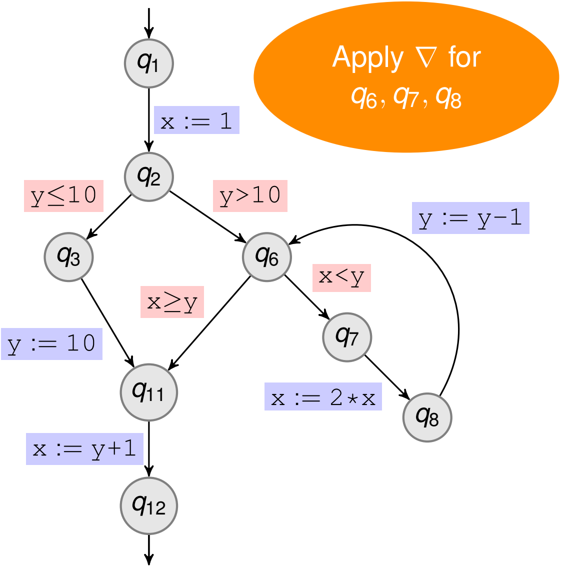


Apply $\nabla$ for $q_6, q_7, q_8$

Show that $\mathtt{x} > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | 0 | $+\infty$ |
| $q_7$ | 1 | $+\infty$ | 1 | $+\infty$ |
| $q_8$ | 2 | $2^{11}$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

# Range Analysis on Example with Delayed Widening

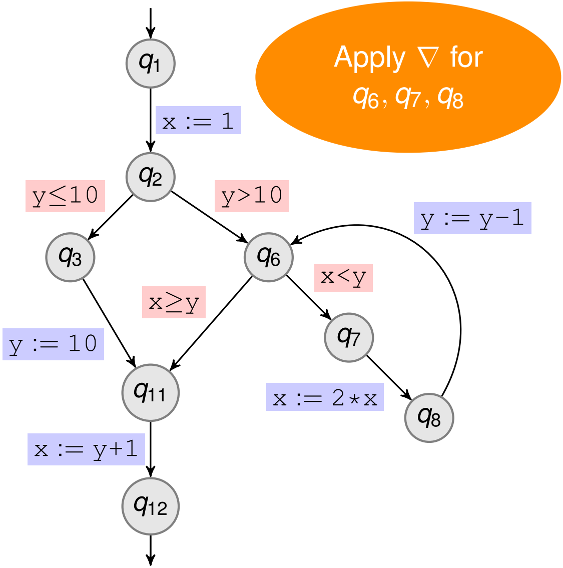


Apply $\nabla$ for $q_6, q_7, q_8$

Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | 0 | $+\infty$ |
| $q_7$ | 1 | $+\infty$ | 1 | $+\infty$ |
| $q_8$ | 2 | $+\infty$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

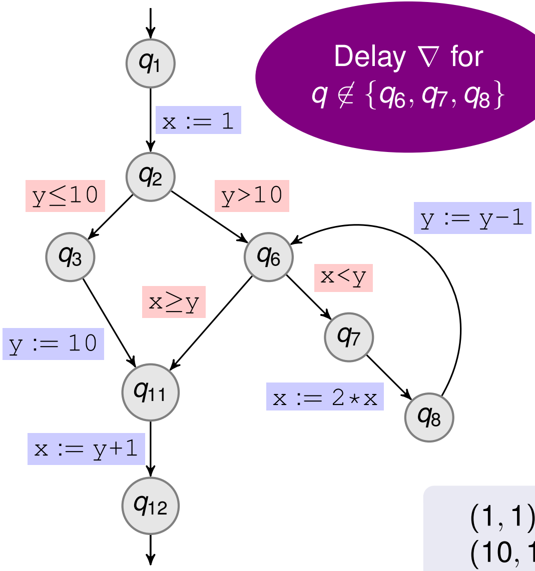
# Range Analysis on Example with Delayed Widening



Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | 0 | $+\infty$ |
| $q_7$ | 1 | $+\infty$ | 1 | $+\infty$ |
| $q_8$ | 2 | $+\infty$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

# Range Analysis on Example with Delayed Widening



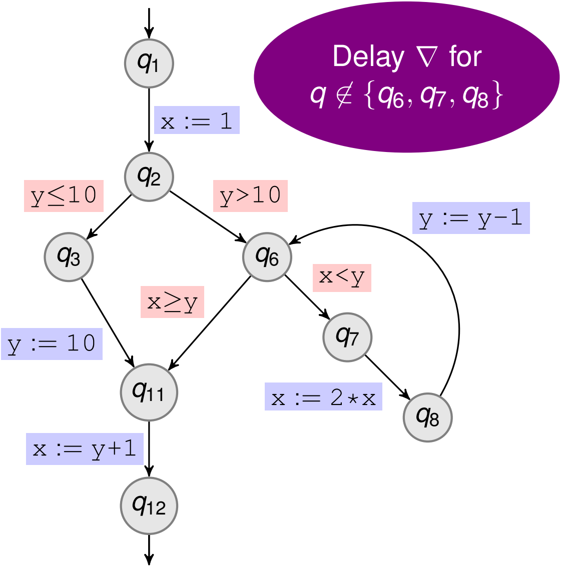
Delay $\nabla$ for
$q \notin \{q_6, q_7, q_8\}$

Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | 0 | $+\infty$ |
| $q_7$ | 1 | $+\infty$ | 1 | $+\infty$ |
| $q_8$ | 2 | $+\infty$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | $+\infty$ | 0 | $+\infty$ |
| $q_{12}$ | 11 | 11 | 10 | 10 |

$$\begin{array}{llll} (1,1) & \sqcup\ (1,+\infty) & = & (1,+\infty) \\ (10,10) & \sqcup\ (0,+\infty) & = & (0,+\infty) \end{array}$$

# Range Analysis on Example with Delayed Widening

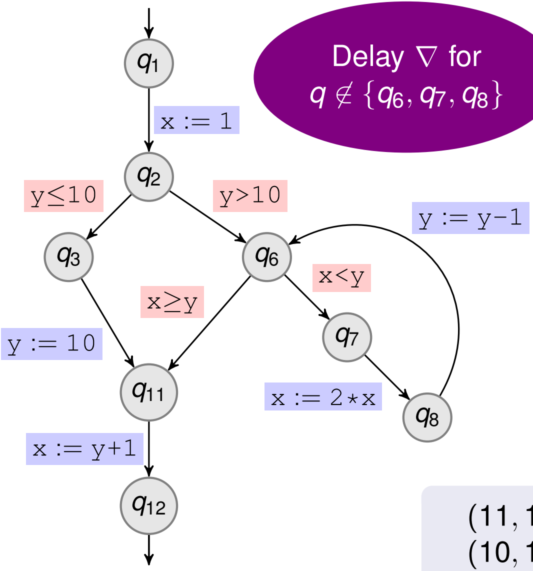


Delay $\nabla$ for $q \notin \{q_6, q_7, q_8\}$

Show that $\mathtt{x} > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | 0 | $+\infty$ |
| $q_7$ | 1 | $+\infty$ | 1 | $+\infty$ |
| $q_8$ | 2 | $+\infty$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | $+\infty$ | 0 | $+\infty$ |
| $q_{12}$ | 11 | 11 | 10 | 10 |

# Range Analysis on Example with Delayed Widening



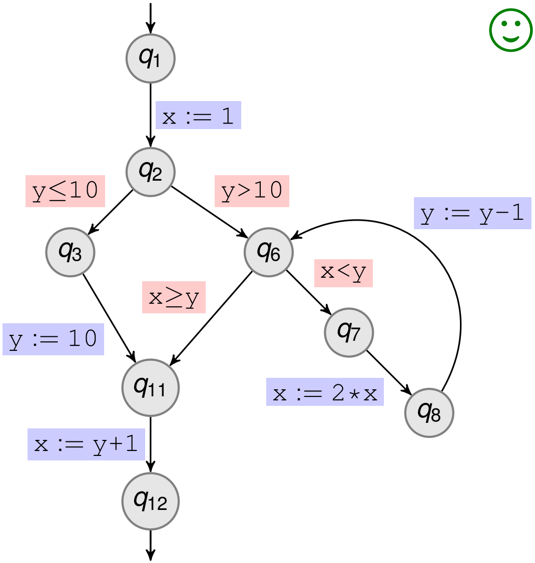
Delay $\nabla$ for $q \notin \{q_6, q_7, q_8\}$

Show that x > 0 at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | 0 | $+\infty$ |
| $q_7$ | 1 | $+\infty$ | 1 | $+\infty$ |
| $q_8$ | 2 | $+\infty$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | $+\infty$ | 0 | $+\infty$ |
| $q_{12}$ | 0 | $+\infty$ | 0 | $+\infty$ |

$$(11,11) \sqcup (1,+\infty) = (1,+\infty)$$
$$(10,10) \sqcup (0,+\infty) = (0,+\infty)$$

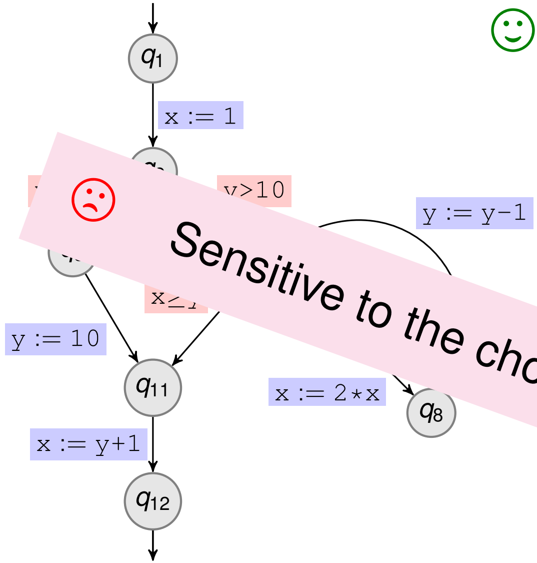
# Range Analysis on Example with Delayed Widening



Show that $\texttt{x} > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | 0 | $+\infty$ |
| $q_7$ | 1 | $+\infty$ | 1 | $+\infty$ |
| $q_8$ | 2 | $+\infty$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | $+\infty$ | 0 | $+\infty$ |
| $q_{12}$ | 0 | $+\infty$ | 0 | $+\infty$ |

# Range Analysis on Example with Delayed Widening



☺ Show that $\mathtt{x} > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | 0 | $+\infty$ |
| $q_7$ | 1 | $+\infty$ | 1 | $+\infty$ |
| [illegible] | 2 | $+\infty$ | 1 | $+\infty$ |
| [illegible] | [illegible] | $+\infty$ | 0 | $+\infty$ |
| [illegible] | [illegible] | [illegible] | [illegible] | $+\infty$ |

☹ Sensitive to the choice of delay ☹

# Precision Improvement with Narrowing

Consider a complete lattice $(L, \sqsubseteq)$.

## Objective of Narrowing Operators

Soundly improve the precision of an approximation obtained with $\nabla$

## Definition

A narrowing operator for $(L, \sqsubseteq)$ is a function $\Delta : (L \times L) \rightarrow L$ such that:

1. $$y \sqsubseteq x \quad \Longrightarrow \quad y \sqsubseteq (x \,\Delta\, y) \sqsubseteq x \qquad \text{(for all } x, y \in L\text{)}$$
2. for any descending chain $x_0 \sqsupseteq x_1 \sqsupseteq \cdots$ of elements of $L$, the descending chain $y_0 \sqsupseteq y_1 \sqsupseteq \cdots$ defined by

$$\begin{cases} y_0 = x_0 \\ y_{i+1} = y_i \,\Delta\, x_{i+1} & \text{for all } i \in \mathbb{N} \end{cases}$$

is not strictly decreasing (i.e. $y_{i+1} = y_i$ for some $i \in \mathbb{N}$).

# Correctness of Decreasing Iteration with Narrowing

Consider a complete lattice $(L, \sqsubseteq)$ and a monotonic function $f : L \rightarrow L$.

A post-fixpoint of $f$ is any element $a \in L$ satisfying $a \sqsupseteq f(a)$.

### Theorem

*If $\Delta : (L \times L) \rightarrow L$ is a narrowing operator then for any post-fixpoint $a$ of $f$, the descending chain $x_0 \sqsupseteq x_1 \sqsupseteq \cdots$ defined by*

$$\begin{array}{rcl} x_0 & = & a \\ x_{i+1} & = & x_i \,\Delta\, f(x_{i+1}) \end{array}$$

*is eventually stationary, and its limit satisfies $\bigsqcap \{x_i \mid i \in \mathbb{N}\} \;\sqsupseteq\; \mathrm{lfp}(f)$.*

### Application to Precision Improvement of MFP Approximations

1. Compute an approximation of *MFP* by Kleene iteration with $\nabla$.
2. Then perform a decreasing iteration with $\Delta$ to regain precision.

# Correctness of Decreasing Iteration with Narrowing

Consider a complete lattice $(L, \sqsubseteq)$ and a monotonic function $f : L \rightarrow L$.

A post-fixpoint of $f$ is any element $a \in L$ satisfying $a \sqsupseteq f(a)$.

## Theorem

*If $\Delta : (L \times L) \rightarrow L$ is a narrowing operator then for any post-fixpoint $a$ of $f$, the descending chain $x_0 \sqsupseteq x_1 \sqsupseteq \cdots$ defined by*

$$\begin{aligned} x_0 &= a \\ x_{i+1} &= x_i \, \Delta \, f(x_{i+1}) \end{aligned}$$

*is eventually stationary, and its limit satisfies $\bigsqcap \{x_i \mid i \in \mathbb{N}\} \sqsupseteq \mathrm{lfp}(f)$.*

## Application to Precision Improvement of MFP Approximations

1. Compute an approximation of *MFP* by Kleene iteration with $\nabla$.
2. Then perform a decreasing iteration with $\Delta$ to regain precision.

# Narrowing Operator for Range Analysis: Intuition

## Objective of Narrowing Operators

Soundly improve the precision of an approximation obtained with $\nabla$

$\nabla$ may have introduced infinite bounds to accelerate convergence.

Improve infinite bounds when possible (leave the non-infinite ones)

## Examples

$$\begin{array}{rcl}
(1, +\infty) \,\Delta\, (1, 4) & = & (1, 4) \\
(1, 10) \,\Delta\, (1, 4) & = & (1, 10) \\
(-\infty, 10) \,\Delta\, (1, 4) & = & (1, 10)
\end{array}$$

# Narrowing Operator for Range Analysis

## Narrowing Operator on the Complete Lattice $(Int, \sqsubseteq)$ of Intervals

$$\bot \,\Delta\, \bot \;=\; \bot \qquad\qquad \bot \,\Delta\, (l, u) \;=\; (l, u) \,\Delta\, \bot \;=\; \bot$$

$$(l_1, u_1) \,\Delta\, (l_2, u_2) \;=\; (l_\Delta, u_\Delta) \quad \text{where} \quad \begin{cases} l_\Delta \;=\; \begin{cases} l_2 & \text{if } l_1 = -\infty \\ l_1 & \text{otherwise} \end{cases} \\ \\ u_\Delta \;=\; \begin{cases} u_2 & \text{if } u_1 = +\infty \\ u_1 & \text{otherwise} \end{cases} \end{cases}$$

## Narrowing Operator on the Complete Lattice $(\mathrm{X} \to Int, \overline{\sqsubseteq})$

Extension $\Delta$ of the narrowing $\Delta$ on $(Int, \sqsubseteq)$ to $(\mathrm{X} \to Int, \overline{\sqsubseteq})$, defined by:

$$\overline{v_1} \,\Delta\, \overline{v_2} \;=\; \lambda q \,.\, \overline{v_1}(q) \,\Delta\, \overline{v_2}(q)$$

# Narrowing Operator for Range Analysis

## Narrowing Operator on the Complete Lattice $(Int, \sqsubseteq)$ of Intervals

$$\bot \,\Delta\, \bot \;=\; \bot \qquad\qquad \bot \,\Delta\, (l, u) \;=\; (l, u) \,\Delta\, \bot \;=\; \bot$$

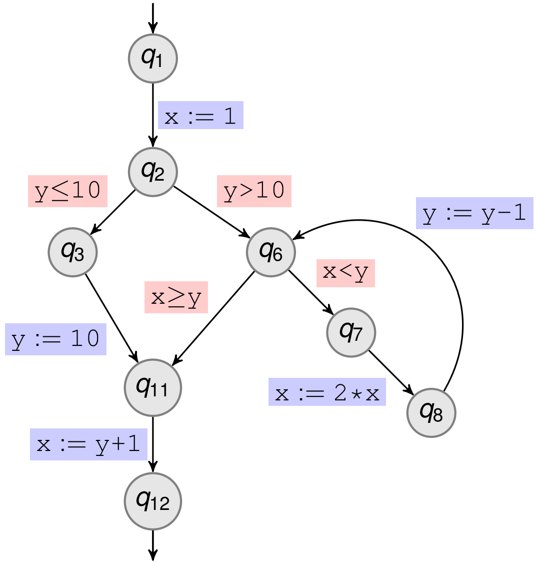
$$(l_1, u_1) \,\Delta\, (l_2, u_2) \;=\; (l_\Delta, u_\Delta) \quad \text{where} \quad \begin{cases} l_\Delta \;=\; \begin{cases} l_2 & \text{if } l_1 = -\infty \\ l_1 & \text{otherwise} \end{cases} \\ \\ u_\Delta \;=\; \begin{cases} u_2 & \text{if } u_1 = +\infty \\ u_1 & \text{otherwise} \end{cases} \end{cases}$$

## Narrowing Operator on the Complete Lattice $(\mathrm{X} \to Int, \overline{\sqsubseteq})$

Extension $\Delta$ of the narrowing $\Delta$ on $(Int, \sqsubseteq)$ to $(\mathrm{X} \to Int, \overline{\sqsubseteq})$, defined by:

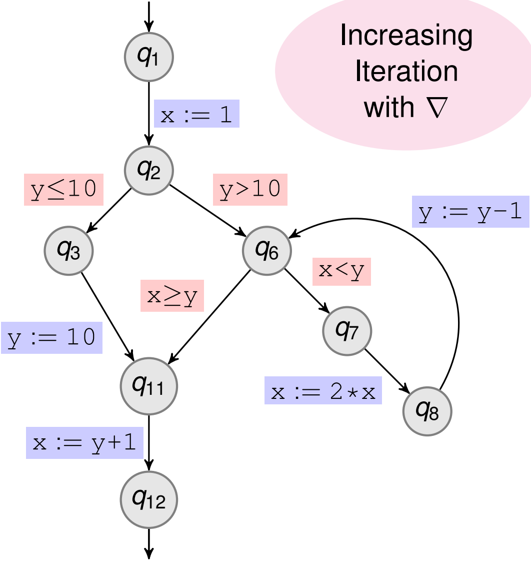
$$\overline{v_1} \,\Delta\, \overline{v_2} \;=\; \lambda\, q \,.\, \overline{v_1}(q) \,\Delta\, \overline{v_2}(q)$$

# Range Analysis on Example with $\nabla$ and $\Delta$



Show that $x > 0$ at $q_{12}$

# Range Analysis on Example with $\nabla$ and $\Delta$

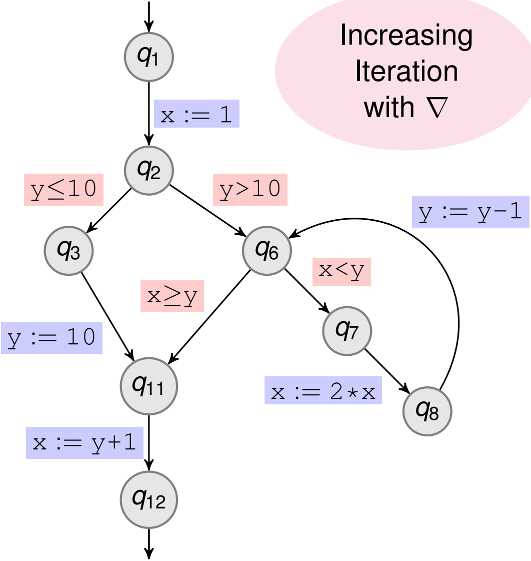


Increasing Iteration with $\nabla$

Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_3$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_6$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_7$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_8$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_{11}$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |
| $q_{12}$ | $\perp$ | $\perp$ | $\perp$ | $\perp$ |

# Range Analysis on Example with ∇ and ∆



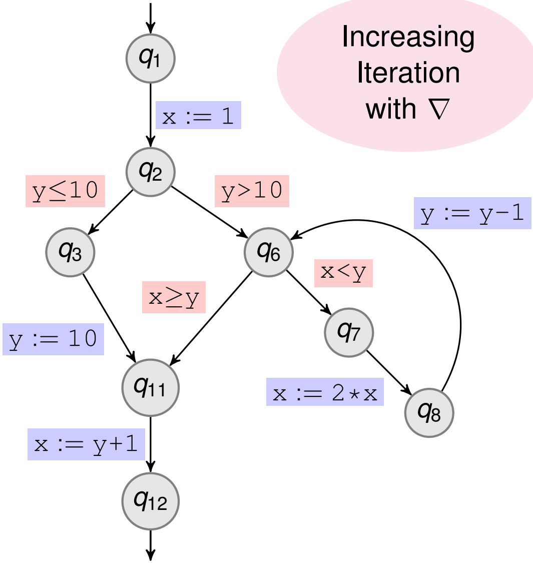
Increasing Iteration with ∇

Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 1 | 10 | $+\infty$ |
| $q_7$ | 1 | 1 | 10 | $+\infty$ |
| $q_8$ | 2 | 2 | 10 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

Same as with non-delayed widening

# Range Analysis on Example with $\nabla$ and $\Delta$



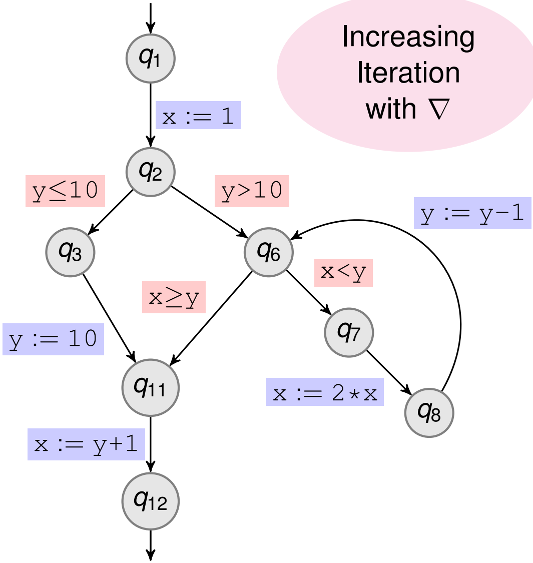
Increasing Iteration with $\nabla$

Show that $x > 0$ at $q_{12}$

|  | x |  | y |  |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | 1 | 10 | $+\infty$ |
| $q_7$ | 1 | 1 | 10 | $+\infty$ |
| $q_8$ | 2 | 2 | 10 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

Same as with non-delayed widening

# Range Analysis on Example with $\nabla$ and $\Delta$



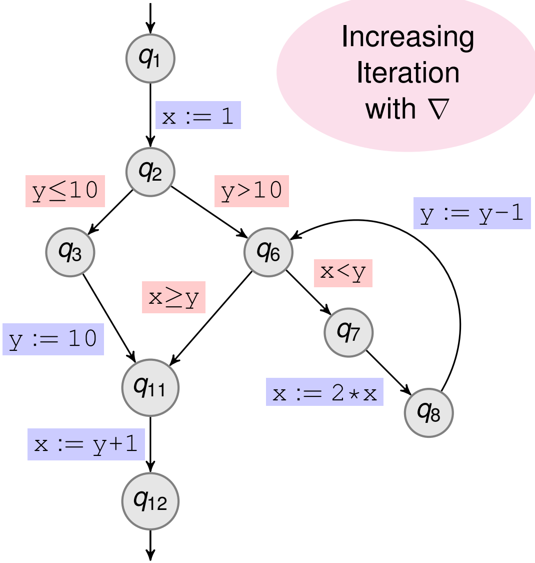
Increasing Iteration with $\nabla$

Show that $x > 0$ at $q_{12}$

|  | x |  | y |  |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_7$ | 1 | 1 | 10 | $+\infty$ |
| $q_8$ | 2 | 2 | 10 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

Same as with non-delayed widening

# Range Analysis on Example with ∇ and Δ



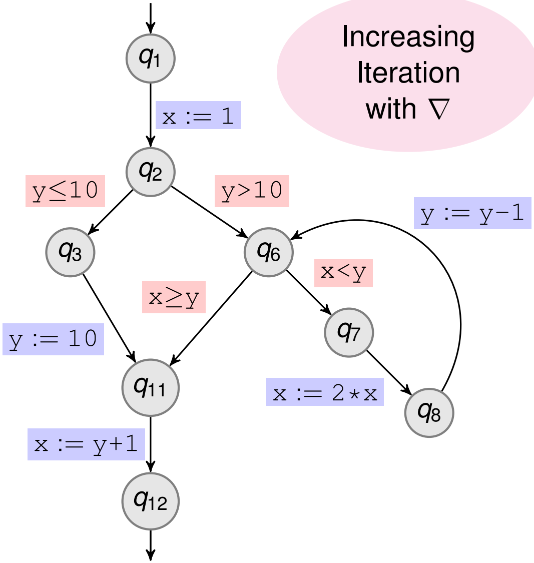
Increasing Iteration with ∇

Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_7$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_8$ | 2 | 2 | 10 | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

Same as with non-delayed widening

# Range Analysis on Example with ∇ and Δ



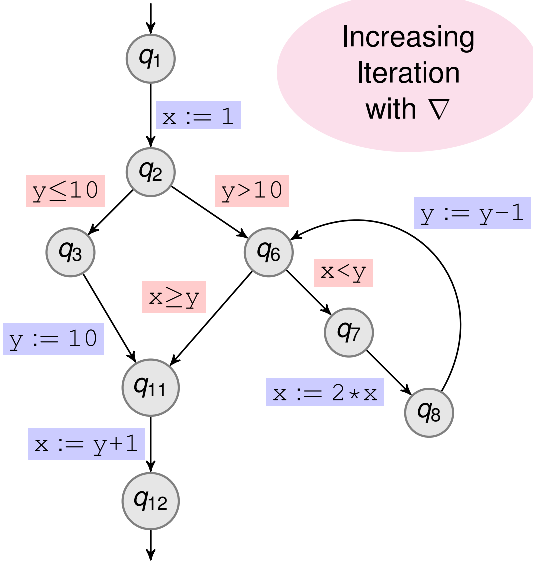
Increasing Iteration with ∇

Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_7$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_8$ | 2 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_{11}$ | 1 | 1 | 10 | 10 |
| $q_{12}$ | 11 | 11 | 10 | 10 |

Same as with non-delayed widening

# Range Analysis on Example with $\nabla$ and $\Delta$



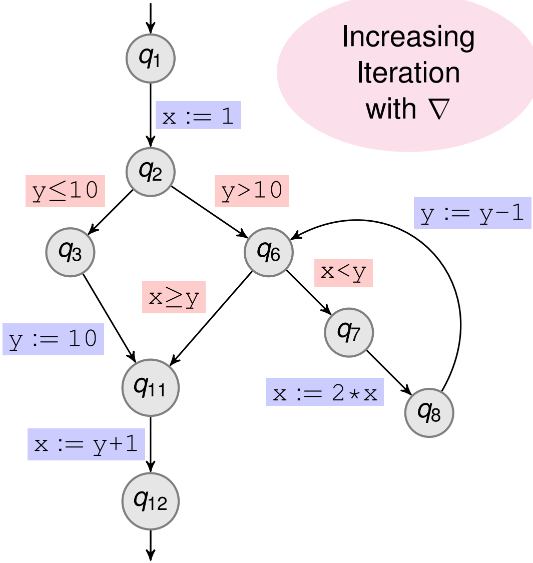
Increasing Iteration with $\nabla$

Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_7$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_8$ | 2 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_{11}$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_{12}$ | 11 | 11 | 10 | 10 |

Same as with non-delayed widening

# Range Analysis on Example with ∇ and Δ



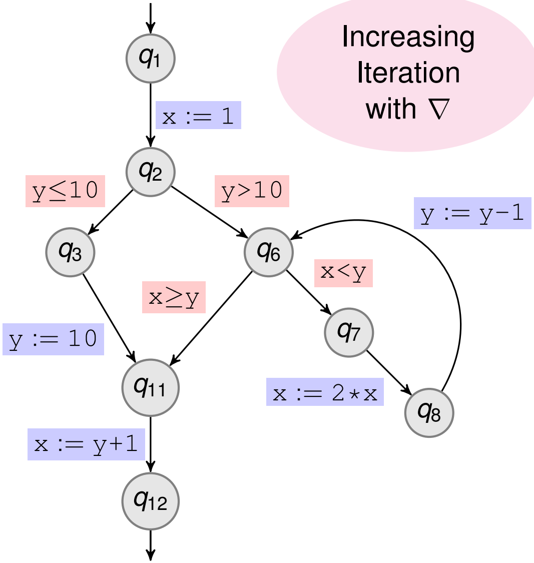
Increasing Iteration with ∇

Show that $\mathtt{x} > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_7$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_8$ | 2 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_{11}$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_{12}$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |

Same as with non-delayed widening

# Range Analysis on Example with $\nabla$ and $\Delta$

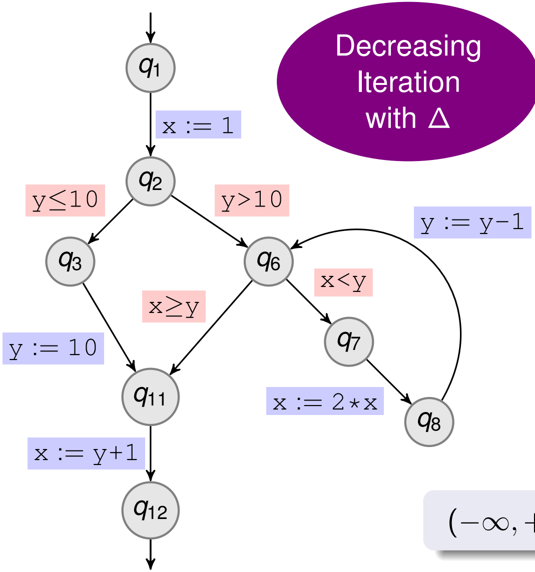


Increasing Iteration with $\nabla$

Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_7$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_8$ | 2 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_{11}$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_{12}$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |

# Range Analysis on Example with $\nabla$ and $\Delta$



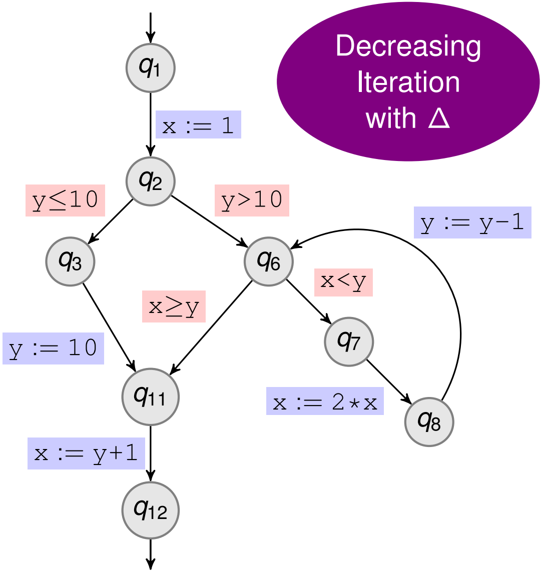
Decreasing Iteration with $\Delta$

Show that $\mathtt{x} > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_7$ | 1 | $+\infty$ | 1 | $+\infty$ |
| $q_8$ | 2 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_{11}$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_{12}$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |

$$(-\infty, +\infty) \;\; \Delta \;\; (1, +\infty) \;\; = \;\; (1, +\infty)$$

# Range Analysis on Example with ∇ and Δ

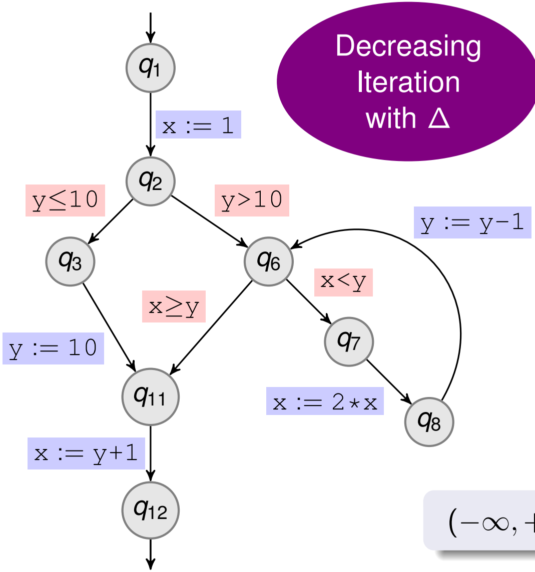


Decreasing Iteration with Δ

Show that $x > 0$ at $q_{12}$

|  | x |  | y |  |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_7$ | 1 | $+\infty$ | 1 | $+\infty$ |
| $q_8$ | 2 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_{11}$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_{12}$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |

# Range Analysis on Example with ∇ and Δ



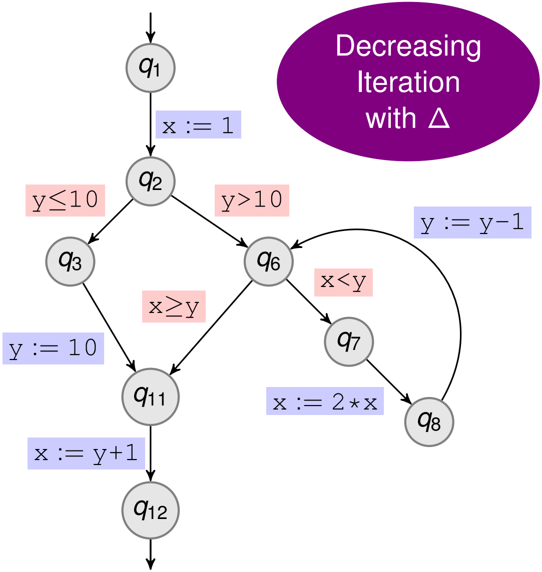
Decreasing Iteration with Δ

Show that x > 0 at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_7$ | 1 | $+\infty$ | 1 | $+\infty$ |
| $q_8$ | 2 | $+\infty$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_{12}$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |

$$(-\infty,+\infty) \;\Delta\; (1,+\infty) \;=\; (1,+\infty)$$

# Range Analysis on Example with $\nabla$ and $\Delta$

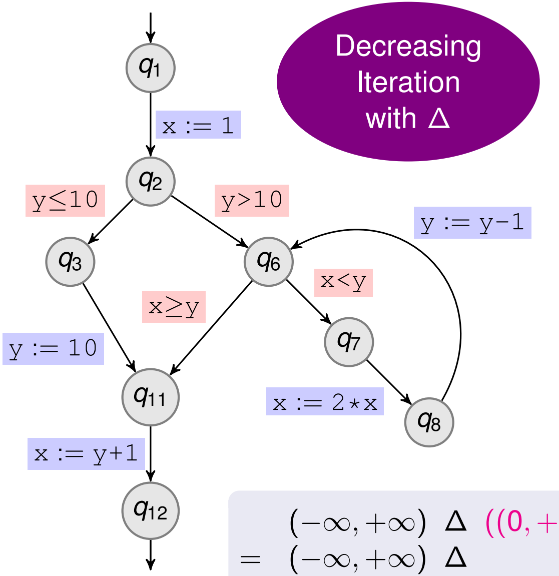


Decreasing Iteration with $\Delta$

Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_7$ | 1 | $+\infty$ | 1 | $+\infty$ |
| $q_8$ | 2 | $+\infty$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_{12}$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |

# Range Analysis on Example with ∇ and Δ



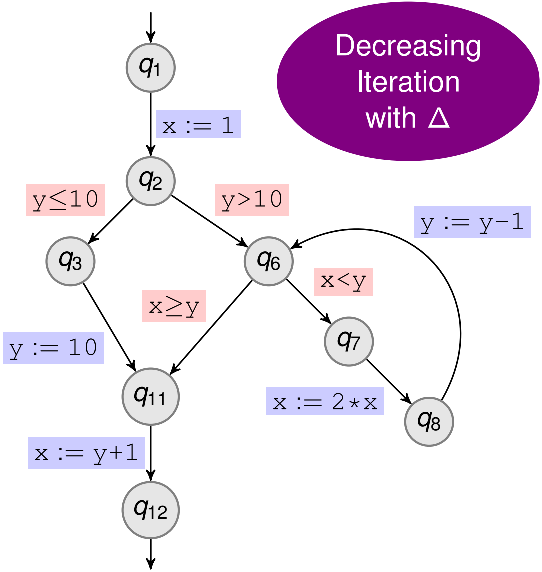
Decreasing Iteration with Δ

Show that $\texttt{x} > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | 0 | $+\infty$ |
| $q_7$ | 1 | $+\infty$ | 1 | $+\infty$ |
| $q_8$ | 2 | $+\infty$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_{12}$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |

$$\begin{array}{rlll} & (-\infty,+\infty) & \Delta & ((0,+\infty)\sqcup(10,+\infty)) \\ = & (-\infty,+\infty) & \Delta & (0,+\infty) \qquad = (0,+\infty) \end{array}$$

# Range Analysis on Example with ∇ and Δ

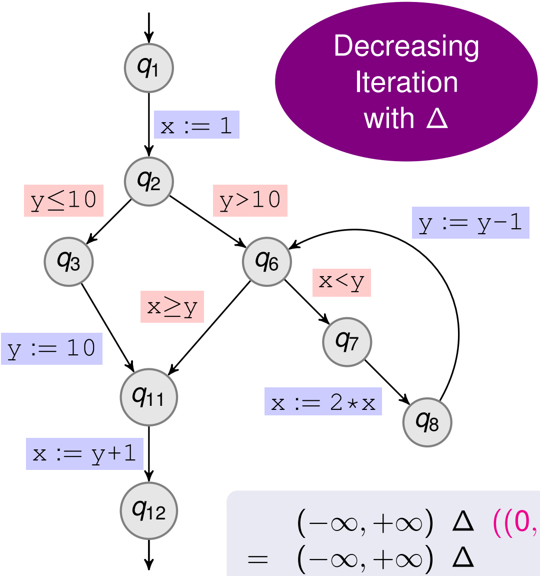


Decreasing Iteration with Δ

Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | 0 | $+\infty$ |
| $q_7$ | 1 | $+\infty$ | 1 | $+\infty$ |
| $q_8$ | 2 | $+\infty$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_{12}$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |

# Range Analysis on Example with $\nabla$ and $\Delta$



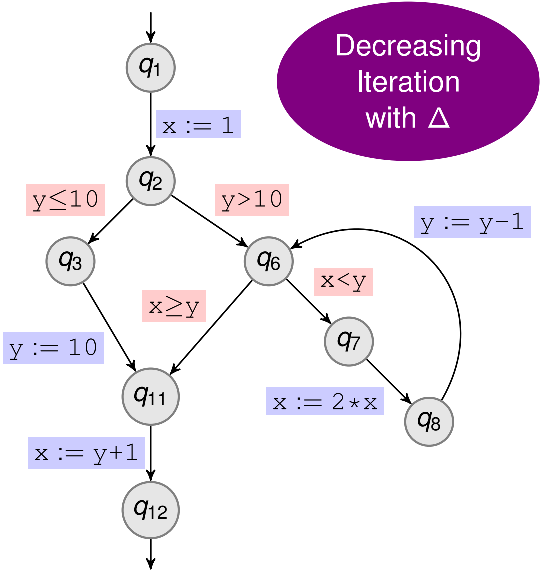
Decreasing Iteration with $\Delta$

Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | 0 | $+\infty$ |
| $q_7$ | 1 | $+\infty$ | 1 | $+\infty$ |
| $q_8$ | 2 | $+\infty$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | $+\infty$ | 0 | $+\infty$ |
| $q_{12}$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |

$$\begin{array}{rllll} & (-\infty,+\infty) & \Delta & ((0,+\infty)\sqcup(10,10)) & \\ = & (-\infty,+\infty) & \Delta & (0,+\infty) & = (0,+\infty)\end{array}$$

# Range Analysis on Example with ∇ and Δ

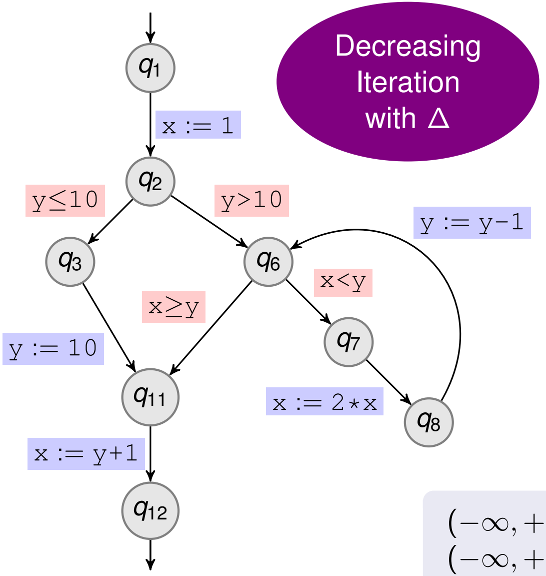


Decreasing Iteration with Δ

Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | 0 | $+\infty$ |
| $q_7$ | 1 | $+\infty$ | 1 | $+\infty$ |
| $q_8$ | 2 | $+\infty$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | $+\infty$ | 0 | $+\infty$ |
| $q_{12}$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |

# Range Analysis on Example with ∇ and Δ



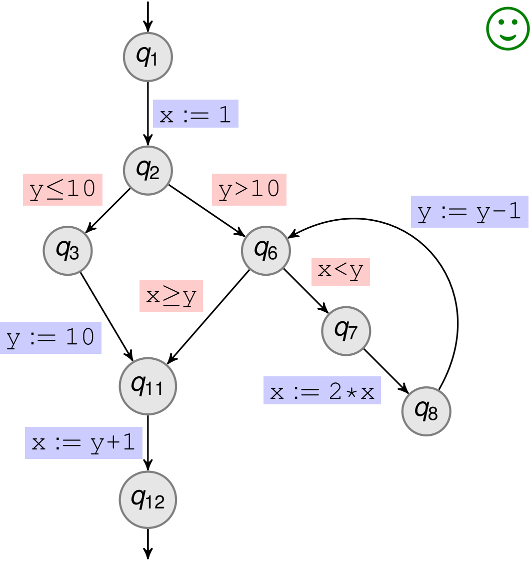
Decreasing Iteration with Δ

Show that $x > 0$ at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | 0 | $+\infty$ |
| $q_7$ | 1 | $+\infty$ | 1 | $+\infty$ |
| $q_8$ | 2 | $+\infty$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | $+\infty$ | 0 | $+\infty$ |
| $q_{12}$ | 1 | $+\infty$ | 0 | $+\infty$ |

$$(-\infty,+\infty)\ \Delta\ (1,+\infty) = (1,+\infty)$$
$$(-\infty,+\infty)\ \Delta\ (0,+\infty) = (0,+\infty)$$

# Range Analysis on Example with ∇ and ∆



☺ Show that x > 0 at $q_{12}$

| | x | | y | |
|---|---|---|---|---|
| $q_1$ | $-\infty$ | $+\infty$ | $-\infty$ | $+\infty$ |
| $q_2$ | 1 | 1 | $+\infty$ | $+\infty$ |
| $q_3$ | 1 | 1 | $-\infty$ | 10 |
| $q_6$ | 1 | $+\infty$ | 0 | $+\infty$ |
| $q_7$ | 1 | $+\infty$ | 1 | $+\infty$ |
| $q_8$ | 2 | $+\infty$ | 1 | $+\infty$ |
| $q_{11}$ | 1 | $+\infty$ | 0 | $+\infty$ |
| $q_{12}$ | 1 | $+\infty$ | 0 | $+\infty$ |

# Selective Application of Widening

Widening introduces imprecision that often cannot be regained by narrowing.

To ensure convergence it is enough to only apply widening at cut points

Cut points: set of locations that cut each loop (in the control flow automaton's graph)

## Other Methods to Reduce Precision Loss of Widening

- Delayed widening
- Widening "up to"

  Given a finite set $M \subseteq L$, use $(x \nabla y) \sqcap \prod \{m \in M \mid a \sqcup b \sqsubseteq m\}$.
- Look-ahead widening
- ...

# Selective Application of Widening

Widening introduces imprecision that often cannot be regained by narrowing.

To ensure convergence it is enough to only apply widening at cut points

Cut points: set of locations that cut each loop (in the control flow automaton's graph)

## Other Methods to Reduce Precision Loss of Widening

- Delayed widening
- Widening "up to"

  Given a finite set $M \subseteq L$, use $(x \nabla y) \sqcap \prod \{m \in M \mid a \sqcup b \sqsubseteq m\}$.
- Look-ahead widening
- . . .

# Part V

# Software Verification by Static Analysis

# Outline — Software Verification by Static Analysis

11 Summary

12 Applications of Static Analysis to Software Verification

13 Limitations of Static Analysis for Software Verification

14 Some References

# Outline — Software Verification by Static Analysis

11 Summary

12 Applications of Static Analysis to Software Verification

13 Limitations of Static Analysis for Software Verification

14 Some References

# Summary: Data Flow Analysis

Compile-time techniques to gather run-time information about data in programs without actually running them

- Live Variables
- Available Expressions
- Uninitialized Variables
- Constant Propagation

Monotone Data Flow Analysis Frameworks

### Minimal Fixpoint

☺ Computable in finite-height lattices

☹ Loss of Precision

### Meet Over All Paths

☺ Most precise solution

☹ Undecidable (constant propagation)

# Summary: Abstract Interpretation

Semantics-based systematic design of correct data flow analyses

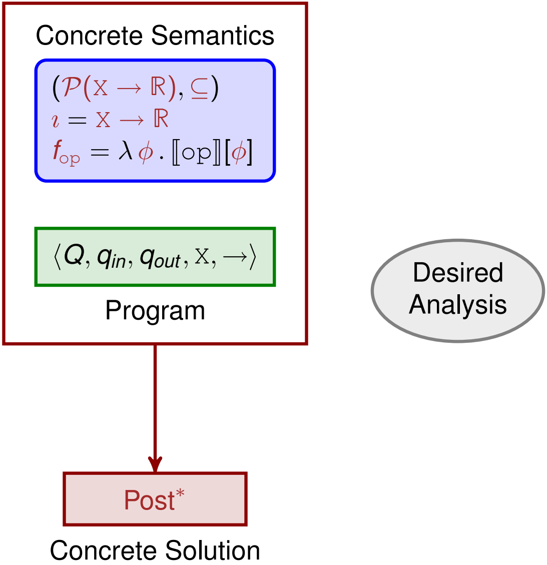
Galois connections to formally relate abstract and concrete semantics

Safe approximations of the “best” abstract semantics

Convergence acceleration with widening and narrowing

- Sign Analysis
- Range Analysis

# Abstract Interpretation-Based Data Flow Analysis



Concrete Semantics

$(\mathcal{P}(\mathrm{X} \to \mathbb{R}), \subseteq)$

$\imath = \mathrm{X} \to \mathbb{R}$

$f_{\mathrm{op}} = \lambda \phi \, . \, [\![\mathrm{op}]\!][\phi]$

$\langle Q, q_{in}, q_{out}, \mathrm{X}, \to \rangle$

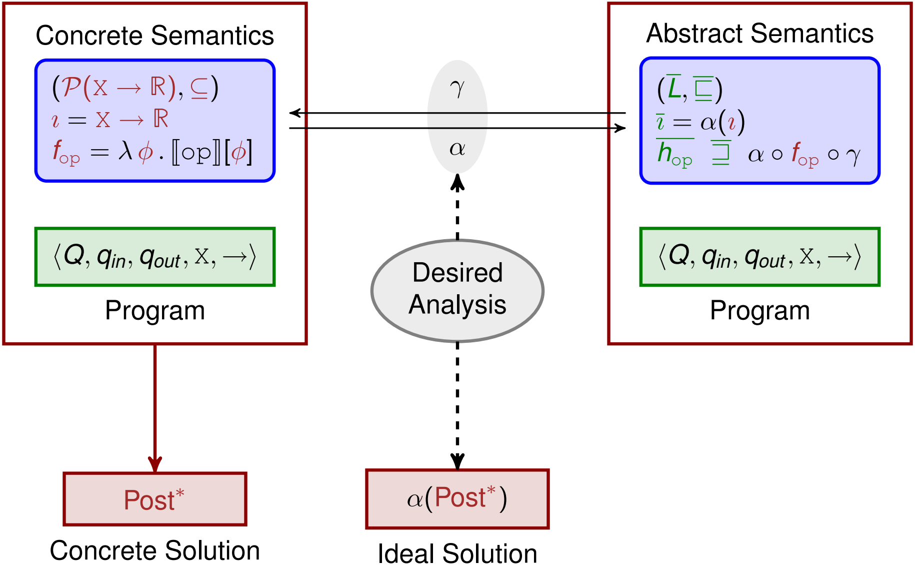
Program

Desired Analysis

Post$^*$

Concrete Solution

# Abstract Interpretation-Based Data Flow Analysis



Concrete Semantics

$(\mathcal{P}(\mathrm{X} \to \mathbb{R}), \subseteq)$

$\imath = \mathrm{X} \to \mathbb{R}$

$f_{\mathrm{op}} = \lambda \phi \,.\, [\![\mathrm{op}]\!][\phi]$

$\langle Q, q_{in}, q_{out}, \mathrm{X}, \to \rangle$

Program

$\gamma$

$\alpha$

Abstract Semantics

$(\overline{L}, \overline{\sqsubseteq})$

$\overline{\imath} = \alpha(\imath)$

$\overline{h_{\mathrm{op}}} \;\; \overline{\sqsupseteq} \;\; \alpha \circ f_{\mathrm{op}} \circ \gamma$

$\langle Q, q_{in}, q_{out}, \mathrm{X}, \to \rangle$

Program

Desired Analysis

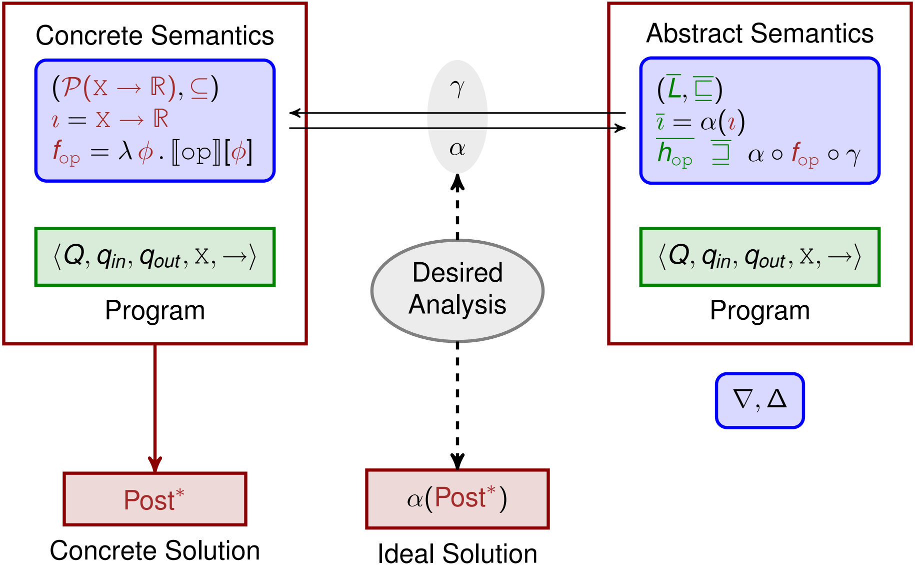
Post$^*$

Concrete Solution

$\alpha($Post$^*)$

Ideal Solution

# Abstract Interpretation-Based Data Flow Analysis



Concrete Semantics

$(\mathcal{P}(\mathrm{X} \to \mathbb{R}), \subseteq)$

$\imath = \mathrm{X} \to \mathbb{R}$

$f_{\mathrm{op}} = \lambda \phi \,.\, [\![\mathrm{op}]\!][\phi]$

$\langle Q, q_{in}, q_{out}, \mathrm{X}, \to \rangle$

Program

$\gamma$

$\alpha$

Desired Analysis

Abstract Semantics

$(\overline{L}, \overline{\sqsubseteq})$

$\overline{\imath} = \alpha(\imath)$

$\overline{h_{\mathrm{op}}} \mathrel{\overline{\sqsupseteq}} \alpha \circ f_{\mathrm{op}} \circ \gamma$

$\langle Q, q_{in}, q_{out}, \mathrm{X}, \to \rangle$

Program

$\nabla, \Delta$

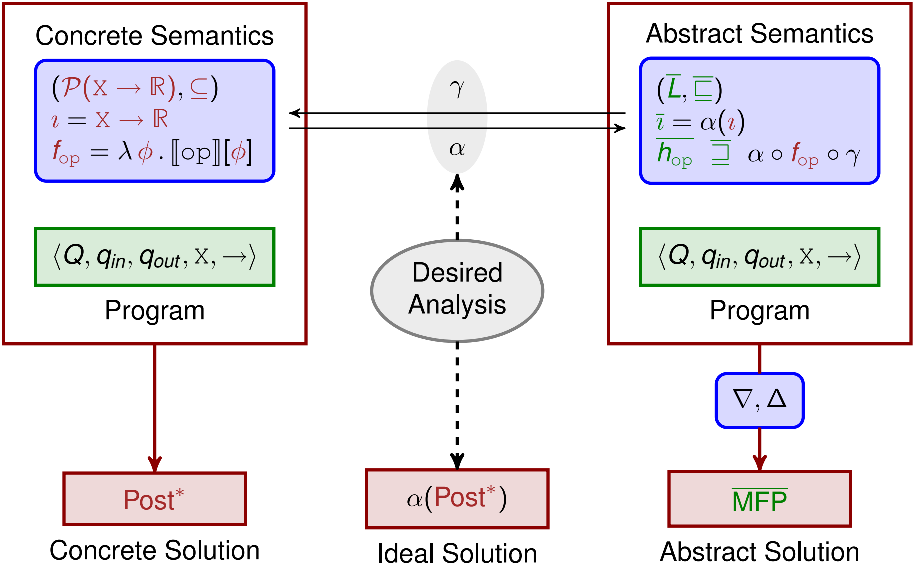
$\mathrm{Post}^*$

Concrete Solution

$\alpha(\mathrm{Post}^*)$

Ideal Solution

# Abstract Interpretation-Based Data Flow Analysis



Concrete Semantics

$(\mathcal{P}(\mathrm{X} \to \mathbb{R}), \subseteq)$

$\imath = \mathrm{X} \to \mathbb{R}$

$f_{\mathrm{op}} = \lambda\, \phi \,.\, [\![\mathrm{op}]\!][\phi]$

$\langle Q, q_{in}, q_{out}, \mathrm{X}, \to \rangle$

Program

Post$^*$

Concrete Solution

$\gamma$

$\alpha$

Desired Analysis

$\alpha(\mathrm{Post}^*)$

Ideal Solution

Abstract Semantics

$(\overline{L}, \overline{\sqsubseteq})$

$\overline{\imath} = \alpha(\imath)$

$\overline{h_{\mathrm{op}}} \;\overline{\sqsupseteq}\; \alpha \circ f_{\mathrm{op}} \circ \gamma$

$\langle Q, q_{in}, q_{out}, \mathrm{X}, \to \rangle$

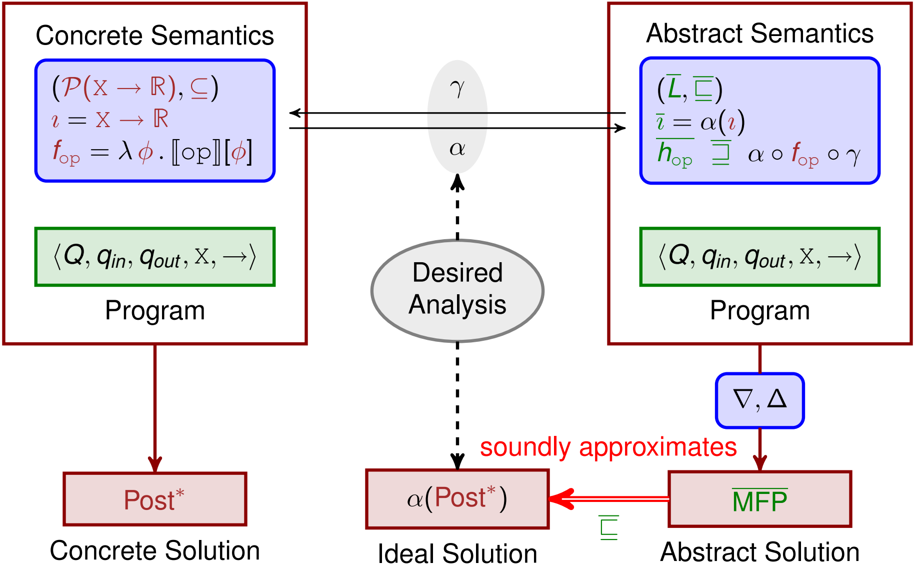
Program

$\nabla, \Delta$

$\overline{\mathrm{MFP}}$

Abstract Solution

# Abstract Interpretation-Based Data Flow Analysis



Concrete Semantics

$(\mathcal{P}(\mathrm{X} \to \mathbb{R}), \subseteq)$

$\imath = \mathrm{X} \to \mathbb{R}$

$f_{\mathrm{op}} = \lambda \phi . [\![\mathrm{op}]\!][\phi]$

$\langle Q, q_{in}, q_{out}, \mathrm{X}, \to \rangle$

Program

$\gamma$

$\alpha$

Abstract Semantics

$(\overline{L}, \overline{\sqsubseteq})$

$\overline{\imath} = \alpha(\imath)$

$\overline{h}_{\mathrm{op}} \;\overline{\sqsupseteq}\; \alpha \circ f_{\mathrm{op}} \circ \gamma$

$\langle Q, q_{in}, q_{out}, \mathrm{X}, \to \rangle$

Program

Desired Analysis

$\nabla, \Delta$

soundly approximates

$\mathrm{Post}^*$

Concrete Solution

$\alpha(\mathrm{Post}^*)$

Ideal Solution

$\overline{\sqsubseteq}$

$\overline{\mathrm{MFP}}$

Abstract Solution

# Outline — Software Verification by Static Analysis

11 Summary

12 Applications of Static Analysis to Software Verification

13 Limitations of Static Analysis for Software Verification

14 Some References

# Applications of Classical Data Flow Analysis

## Very Common Sources of Bugs

Uninitialized variables

Dead code

...

Can be detected by gen / kill data flow analyses

Data flow analysis in every compiler!

# Classical Data Flow Analysis in Compilers

```java
1 class Foo1 {
2     static void foo1(int x) {
3         int i, y;
4 
5         for (i = 0 ; i < x ; i++) {
6             y = y + (i * i);
7         }
8     }
9 }
```

```
$ javac Foo1.java
Foo1.java:6: variable y might not have been initialized
            y = y + (i * i);
                ^
1 error
```

# Abstract Interpretation-Based Invariant Generation

## Software Verification: Is $\text{Post}^*$ disjoint from *Bad*?

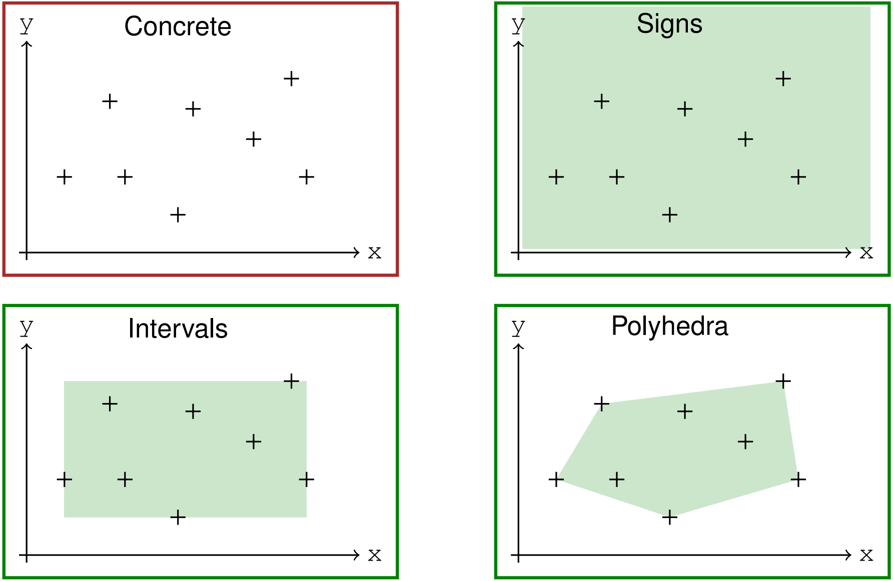
1. Compute an invariant $\mathit{Inv} \supseteq \text{Post}^*$
2. If *Inv* is disjoint from *Bad* then return “program safe”

The MFP solution obtained by abstract interpretation is an invariant ☺

## Tradeoff between computational cost and precision

- Numerical abstract domains
- Approximate transfer mappings
- Widenings and narrowings

# Some Numerical Abstract Domains



Concrete

Signs

Intervals

Polyhedra

# Some Commercial Static Analysis Tools

PolySpace$^{\text{TM}}$ Embedded Software Verification, The MathWorks$^{\text{TM}}$

> *« PolySpace$^{\text{TM}}$ products verify C, C++, and Ada code for embedded applications by detecting run-time errors before code is compiled and executed. »*

Coverity Prevent$^{\text{TM}}$ Static Analysis for C/C++, for C#, and for Java

> *« The foundation of Coverity's leading automated approach to identifying and resolving the most critical defects in C, C++, C# and Java source code. »*

Coverity periodically runs Coverity Prevent$^{\text{TM}}$ on open source projects

- Program Analyzer Generator (Saarland Univ. & AbsInt GmbH)
- Purify (IBM), Klocwork, . . .

# Some Academic Static Analysis Tools

## ASTRÉE Static Analyzer — P. Cousot, R. Cousot, . . .

- Abstract interpretation-based analysis of C
- Application to safety critical embedded software

Verification of the primary flight control software of the Airbus A340 and A380 fly-by-wire systems

## APRON Numerical Abstract Domain Library — B. Jeannet, . . .

- Common interface to various abstract domains
  - includes intervals, polyhedra, octagons, linear congruences
- Online demonstration of the Interproc analyzer
- Open-source, released under the GNU LGPL

# Outline — Software Verification by Static Analysis

11 Summary

12 Applications of Static Analysis to Software Verification

13 Limitations of Static Analysis for Software Verification

14 Some References

# Main Limitation of Data Flow Analysis

False Positives and False Negatives

## Example of False Positive

Variable detected as not initialized, but in fact it is initialized for all runs of the program.

## Example of False Negative

No code detected as dead, but in fact some program point cannot be reached by any run.

# Main Limitation of Data Flow Analysis

False Positives and False Negatives

### Example of False Positive

Variable detected as not initialized, but in fact it is initialized for all runs of the program.

### Example of False Negative

No code detected as dead, but in fact some program point cannot be reached by any run.

# Main Limitation of Data Flow Analysis

False Positives and False Negatives

## Example of False Positive

Variable detected as not initialized, but in fact it is initialized for all runs of the program.

## Example of False Negative

No code detected as dead, but in fact some program point cannot be reached by any run.

# Classical Data Flow Analysis in Compilers

```
class Foo2 {
    static int foo2(int x) {
        int y;

        if (x == 0) { y = 5; }
        else        { y = 2; }
        return y;
    }
}
```

```
$ javac Foo2.java
$
```

# Classical Data Flow Analysis in Compilers

```java
1 class Foo3 {
2     static int foo3(int x) {
3         int y;
4
5         if (x == 0) { y = 5; }
6         if (x != 0) { y = 2; }
7         return y;
8     }
9 }
```

```
$ javac Foo3.java
Foo3.java:7: variable y might not have been initialized
       return y;
            ^
1 error
```

# Main Limitation of Abstract Interpretation

False Positives

**Software Verification: Is Post$^*$ disjoint from *Bad*?**

1. Compute an invariant *Inv* $\supseteq$ Post$^*$
2. If *Inv* is disjoint from *Bad* then return “program safe”
3. If *Inv* intersects *Bad* then return “alarm”

In practice, there might be too many false alarms. . .

# Main Limitation of Abstract Interpretation

False Positives

### Software Verification: Is $\mathrm{Post}^*$ disjoint from *Bad*?

1. Compute an invariant $Inv \supseteq \mathrm{Post}^*$
2. If *Inv* is disjoint from *Bad* then return “program safe”
3. If *Inv* intersects *Bad* then return “alarm”

In practice, there might be too many false alarms. . .

# Main Limitation of Abstract Interpretation

False Positives

**Software Verification: Is Post$^*$ disjoint from *Bad*?**

1. Compute an invariant $Inv \supseteq \mathrm{Post}^*$
2. If *Inv* is disjoint from *Bad* then return “program safe”
3. If *Inv* intersects *Bad* then return “alarm”

In practice, there might be too many false alarms. . .

# What can we do about it?

### Software Verification by Static Analysis: Workflow

While the analysis returns alarms

1. Inspect alarms to determine whether they are spurious or not
2. If alarms are spurious then refine the analysis to gain precision

Why not automate this process?

Trade termination guarantee with fully automatic model-checking

- ☹ Not acceptable for compile-time static analyses
- ☺ Acceptable for verification

Topic of next part. . .

# What can we do about it?

## Software Verification by Static Analysis: Workflow

While the analysis returns alarms

1. Inspect alarms to determine whether they are spurious or not
2. If alarms are spurious then refine the analysis to gain precision

Why not automate this process?

Trade termination guarantee with fully automatic model-checking

- ☹ Not acceptable for compile-time static analyses
- ☺ Acceptable for verification

Topic of next part. . .

# Outline — Software Verification by Static Analysis

11 Summary

12 Applications of Static Analysis to Software Verification

13 Limitations of Static Analysis for Software Verification

14 Some References

# Some References

- F. Nielson, H. R. Nielson, and C. Hankin.
  *Principles of Program Analysis*.
  Springer, 1999.

- P. Cousot and R. Cousot.
  Systematic design of program analysis frameworks.
  In *Proc. 6th ACM Symp. Principles of Programming Languages, San Antonio, TX, USA*, pages 269–282. ACM Press, 1979.

- The ASTRÉE Static Analyzer.
  `http://www.astree.ens.fr/`
- The APRON Library for Numerical Abstract Domains.
  `http://apron.cri.ensmp.fr/library/`
- Coverity's Scan.
  `http://scan.coverity.com/`

Part VI

# Abstract Model Refinement

# Outline — Abstract Model Refinement

15 Introduction and Overview

16 Basic Theory on Property-Preserving Abstractions

17 Abstraction Schemes

18 Counterexample Guided Refinement

# Outline — Abstract Model Refinement

15 Introduction and Overview

16 Basic Theory on Property-Preserving Abstractions

17 Abstraction Schemes

18 Counterexample Guided Refinement

# Software Verification by Static Analysis (Repetition)

## Software Verification: Is Post$^*$ disjoint from *Bad*?

1. Compute an invariant *Inv* $\supseteq$ Post$^*$
2. If *Inv* is disjoint from *Bad* then return “program safe”
3. If *Inv* intersects *Bad* then return “alarm”

Alarms must be inspected manually ☹

If an alarm is a real bug, then the analysis is useful ☺

Otherwise…

An improved analysis must be designed to eliminate alarms

# Software Verification by Static Analysis (Repetition)

## Software Verification: Is Post$^*$ disjoint from *Bad*?

1. Compute an invariant *Inv* $\supseteq$ Post$^*$
2. If *Inv* is disjoint from *Bad* then return “program safe”
3. If *Inv* intersects *Bad* then return “alarm”

Alarms must be inspected manually ☹

If an alarm is a real bug, then the analysis is useful ☺

Otherwise…

An improved analysis must be designed to eliminate alarms

# Software Verification by Static Analysis (Repetition)

## Software Verification: Is Post$^*$ disjoint from *Bad*?

1. Compute an invariant *Inv* $\supseteq$ Post$^*$
2. If *Inv* is disjoint from *Bad* then return “program safe”
3. If *Inv* intersects *Bad* then return “alarm”

Alarms must be inspected manually ☹

If an alarm is a real bug, then the analysis is useful ☺

Otherwise. . .

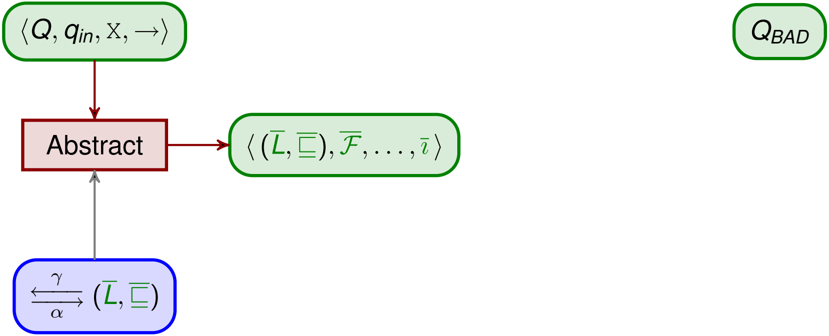
An improved analysis must be designed to eliminate alarms

# Software Verification by Static Analysis: Workflow

$\langle Q, q_{in}, \mathrm{X}, \rightarrow \rangle$

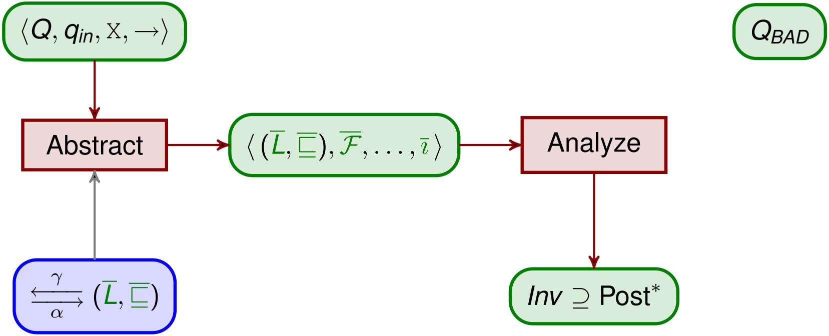
$Q_{BAD}$

# Software Verification by Static Analysis: Workflow

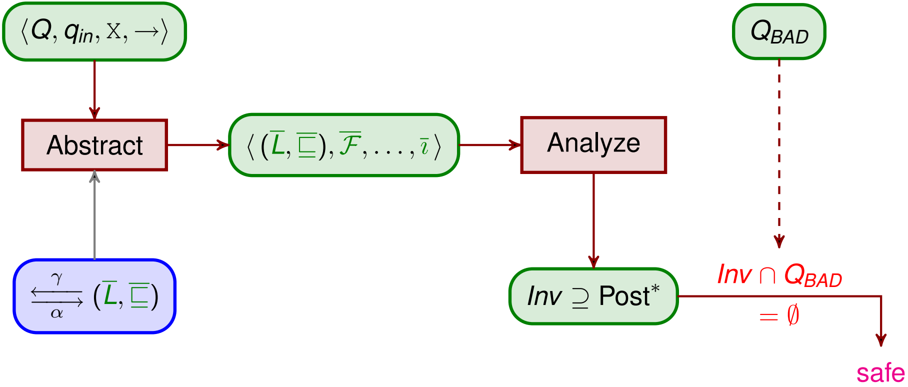


$\langle Q, q_{in}, \mathrm{X}, \rightarrow \rangle$
$Q_{BAD}$
Abstract
$\langle (\overline{L}, \overline{\sqsubseteq}), \overline{\mathcal{F}}, \ldots, \overline{\imath} \rangle$
$\overset{\gamma}{\underset{\alpha}{\leftrightarrows}} (\overline{L}, \overline{\sqsubseteq})$

# Software Verification by Static Analysis: Workflow

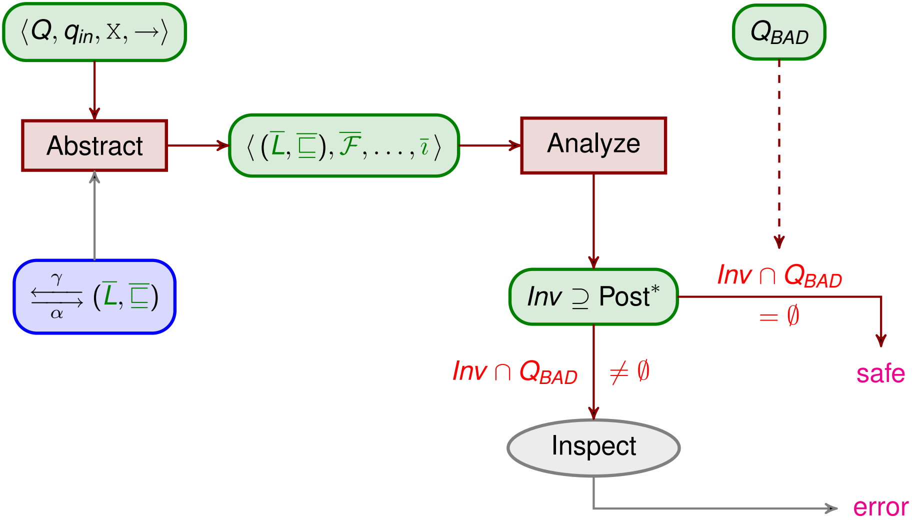

$\langle Q, q_{in}, \mathrm{X}, \rightarrow \rangle$
$Q_{BAD}$
Abstract
$\langle (\overline{L}, \overline{\sqsubseteq}), \overline{\mathcal{F}}, \ldots, \overline{\imath} \rangle$
Analyze
$\xrightleftharpoons[\alpha]{\gamma} (\overline{L}, \overline{\sqsubseteq})$
$Inv \supseteq \mathrm{Post}^*$

# Software Verification by Static Analysis: Workflow



$\langle Q, q_{in}, \mathrm{X}, \rightarrow \rangle$
Abstract
$\langle (\overline{L}, \overline{\sqsubseteq}), \overline{\mathcal{F}}, \ldots, \bar{\imath} \rangle$
Analyze
$Q_{BAD}$
$\underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (\overline{L}, \overline{\sqsubseteq})$
$Inv \supseteq \mathrm{Post}^*$
$Inv \cap Q_{BAD}$
$= \emptyset$
safe

# Software Verification by Static Analysis: Workflow



$\langle Q, q_{in}, \mathrm{X}, \rightarrow \rangle$

Abstract

$\langle (\overline{L}, \overline{\sqsubseteq}), \overline{\mathcal{F}}, \ldots, \overline{\imath} \rangle$

Analyze

$Q_{BAD}$

$\underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (\overline{L}, \overline{\sqsubseteq})$

$\mathit{Inv} \supseteq \mathrm{Post}^*$

$\mathit{Inv} \cap Q_{BAD} = \emptyset$

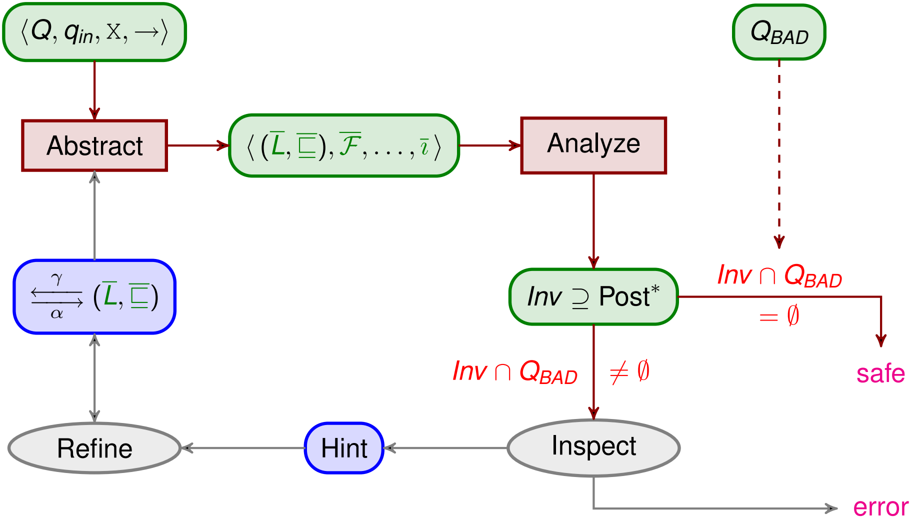
safe

$\mathit{Inv} \cap Q_{BAD} \neq \emptyset$

Inspect

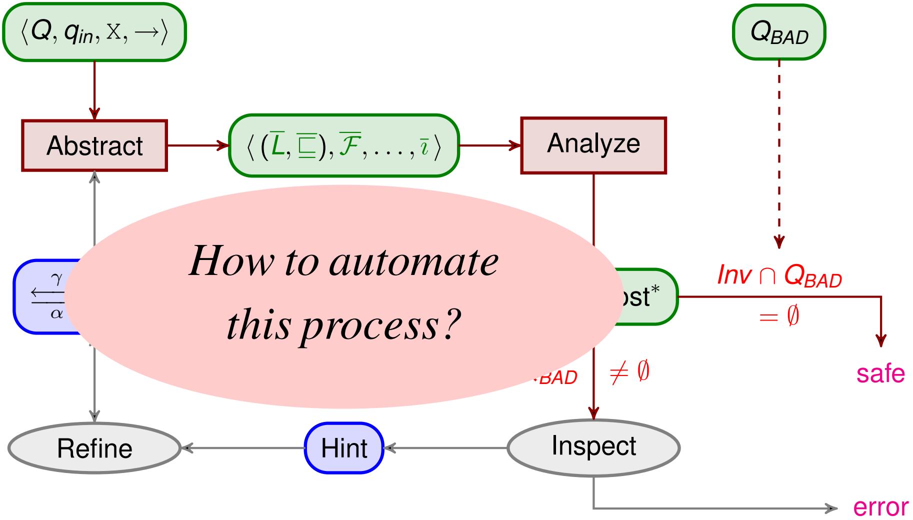
error

# Software Verification by Static Analysis: Workflow



$\langle Q, q_{in}, \mathrm{X}, \rightarrow \rangle$
$Q_{BAD}$
Abstract
$\langle (\overline{L}, \overline{\sqsubseteq}), \overline{\mathcal{F}}, \ldots, \overline{\imath} \rangle$
Analyze
$\underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (\overline{L}, \overline{\sqsubseteq})$
$Inv \supseteq Post^*$
$Inv \cap Q_{BAD}$
$= \emptyset$
safe
$Inv \cap Q_{BAD}$
$\neq \emptyset$
Refine
Hint
Inspect
error

# Software Verification by Static Analysis: Workflow


$\langle Q, q_{in}, \mathrm{X}, \rightarrow \rangle$
$Q_{BAD}$
Abstract
$\langle (\overline{L}, \overline{\sqsubseteq}), \overline{\mathcal{F}}, \ldots, \overline{\imath} \rangle$
Analyze
$\xleftarrow[\alpha]{\gamma}$
ost*
$Inv \cap Q_{BAD}$
$= \emptyset$
safe
BAD
$\neq \emptyset$
Refine
Hint
Inspect
error

*How to automate this process?*

# Inspection of Alarms: Not a Simple Task!

## Objective

Given an abstract invariant $\overline{Inv}$ whose concretization $\gamma(\overline{Inv})$ intersects $Bad = Q_{BAD} \times (\mathbb{X} \to \mathbb{R})$, determine whether Post* intersects *Bad*.

All configurations in $\gamma(\overline{Inv}) \cap Bad$ are potentially reachable…

How are these configurations potentially reached?

It would be nice to have an "abstract run" of the form:

$$(q_{in}, \overline{\psi_0}) \xrightarrow{\mathrm{op}_0} \cdots \xrightarrow{\mathrm{op}_{k-1}} (q_k, \overline{\psi_k}) \quad \text{with} \quad \begin{cases} q_k \in Q_{BAD} \\ \gamma(\overline{\psi_k}) \cap Bad \neq \emptyset \end{cases}$$

Checking whether this abstract run is spurious reduces to checking emptiness of the relation: $(\llbracket \mathrm{op}_k \rrbracket \circ \cdots \circ \llbracket \mathrm{op}_0 \rrbracket)$.

# Inspection of Alarms: Not a Simple Task!

## Objective

Given an abstract invariant $\overline{Inv}$ whose concretization $\gamma(\overline{Inv})$ intersects $Bad = Q_{BAD} \times (\mathbb{X} \to \mathbb{R})$, determine whether Post$^*$ intersects $Bad$.

All configurations in $\gamma(\overline{Inv}) \cap Bad$ are potentially reachable. . .

How are these configurations potentially reached?

It would be nice to have an "abstract run" of the form:

$$(q_{in}, \overline{\psi_0}) \xrightarrow{\mathrm{op}_0} \cdots \xrightarrow{\mathrm{op}_{k-1}} (q_k, \overline{\psi_k}) \quad \text{with} \quad \begin{cases} q_k \in Q_{BAD} \\ \gamma(\overline{\psi_k}) \cap Bad \neq \emptyset \end{cases}$$

Checking whether this abstract run is spurious reduces to checking emptiness of the relation: $(\llbracket \mathrm{op}_k \rrbracket \circ \cdots \circ \llbracket \mathrm{op}_0 \rrbracket)$.

# Inspection of Alarms: Not a Simple Task!

## Objective

Given an abstract invariant $\overline{Inv}$ whose concretization $\gamma(\overline{Inv})$ intersects $Bad = Q_{BAD} \times (\mathbb{X} \rightarrow \mathbb{R})$, determine whether Post$^*$ intersects *Bad*.

All configurations in $\gamma(\overline{Inv}) \cap Bad$ are potentially reachable...

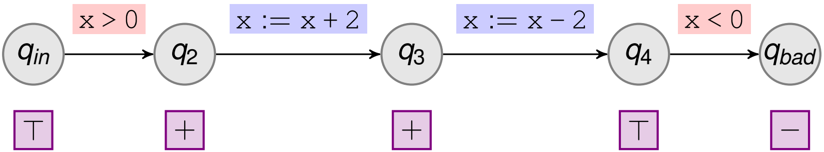
How are these configurations potentially reached?

It would be nice to have an "abstract run" of the form:

$$(q_{in}, \overline{\psi_0}) \xrightarrow{\mathrm{op}_0} \cdots \xrightarrow{\mathrm{op}_{k-1}} (q_k, \overline{\psi_k}) \quad \text{with} \quad \begin{cases} q_k \in Q_{BAD} \\ \gamma(\overline{\psi_k}) \cap Bad \neq \emptyset \end{cases}$$

Checking whether this abstract run is spurious reduces to checking emptiness of the relation: $(\llbracket \mathrm{op}_k \rrbracket \circ \cdots \circ \llbracket \mathrm{op}_0 \rrbracket)$.

# Inspection of Abstract Runs: Example



## Semantics of Operations (Repetition)

$$(v, v') \in [\![g]\!] \quad \textit{if} \quad v \models g \;\; \textit{and} \;\; v' = v$$

$$(v, v') \in [\![x := e]\!] \quad \textit{if} \quad \begin{cases} v'(x) & = & [\![e]\!]_v \\ v'(y) & = & v'(y) \quad \text{for all } y \neq x \end{cases}$$

$$[\![x < 0]\!] \circ [\![x := x - 2]\!] \circ [\![x := x + 2]\!] \circ [\![x > 0]\!] \quad = \quad \emptyset$$

$x > 0 \;\wedge\; x' = x + 2 \;\wedge\; x'' = x' - 2 \;\wedge\; x'' < 0$ *unsastisfiable*

# Refinement of Abstract Domains: Not a Simple Task!

## Objective

Given an abstract invariant $\overline{Inv}$ and a subset $U \subseteq \gamma(\overline{Inv}) \setminus \text{Post}^*$, design a new abstract domain where the resulting $\overline{Inv}$ is disjoint from $U$.

*U* would be a set of configurations identified as false alarms.

Quite challenging!

## More Reasonable Objective

Given a spurious “abstract run”, design a new abstract domain that eliminates this “abstract run”.

# Refinement of Abstract Domains: Not a Simple Task!

## Objective

Given an abstract invariant $\overline{\mathit{Inv}}$ and a subset $U \subseteq \gamma(\overline{\mathit{Inv}}) \setminus \mathrm{Post}^*$, design a new abstract domain where the resulting $\overline{\mathit{Inv}}$ is disjoint from $U$.

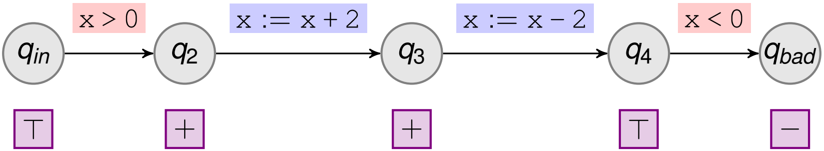
$U$ would be a set of configurations identified as false alarms.

Quite challenging!

## More Reasonable Objective

Given a spurious “abstract run”, design a new abstract domain that eliminates this “abstract run”.

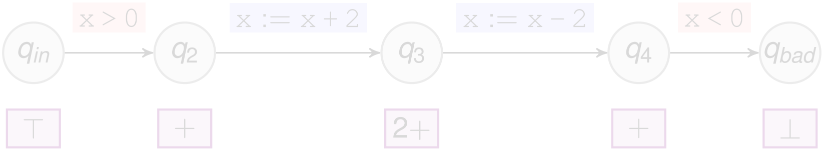
# Refinement Based on Abstract Runs: Example



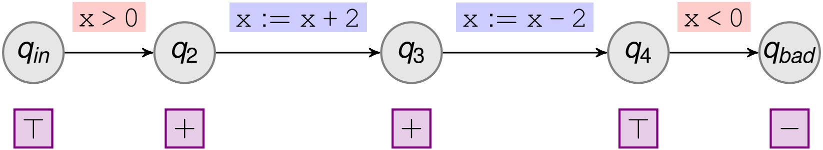
In $q_3$, the set of reachable valuations is:

$$[\![\mathtt{x} := \mathtt{x} + 2]\!] \circ [\![\mathtt{x} > 0]\!][(\mathtt{X} \to \mathbb{R})] \quad = \quad \{v \in \mathtt{X} \to \mathbb{R} \mid v(\mathtt{x}) > 2\}$$

We lack the "property" $\mathtt{x} > 2$. Let us add it (as 2+) to the *Sign* domain.



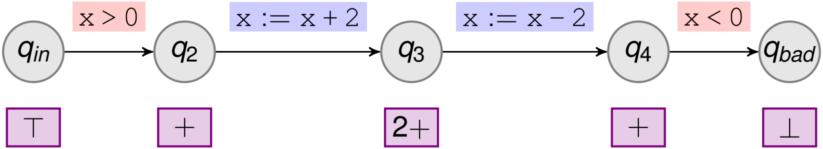
# Refinement Based on Abstract Runs: Example



In $q_3$, the set of reachable valuations is:

$$[\![\mathrm{x} := \mathrm{x} + 2]\!] \circ [\![\mathrm{x} > 0]\!][(\mathrm{X} \to \mathbb{R})] \quad = \quad \{v \in \mathrm{X} \to \mathbb{R} \mid v(\mathrm{x}) > 2\}$$

We lack the “property” $\mathrm{x} > 2$. Let us add it (as $2+$) to the *Sign* domain.



# Hypothetical Workflow Based on Abstract Runs

Abstract "counterexample" runs are key to:

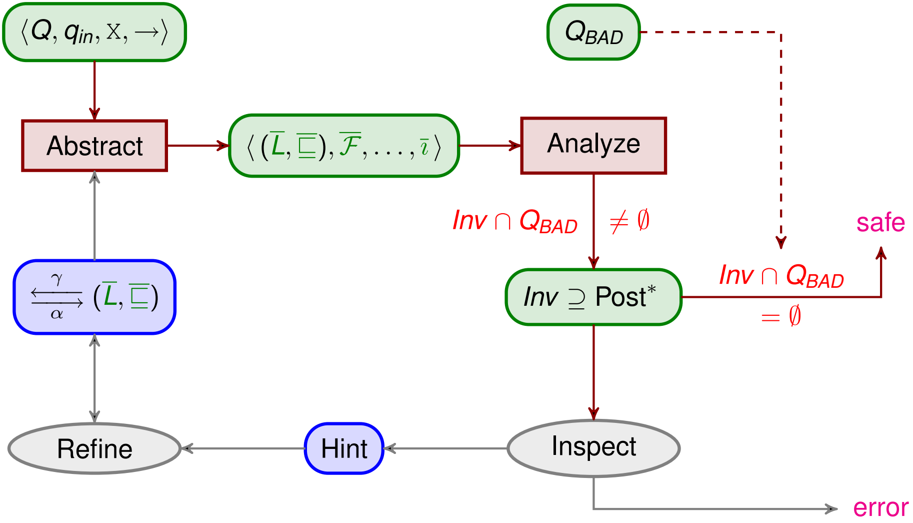
- inspection of alarms
- refinement of abstract domains

## Enhanced Workflow Based on Abstract Runs

Imagine a hypothetical workflow where the analyzer returns:

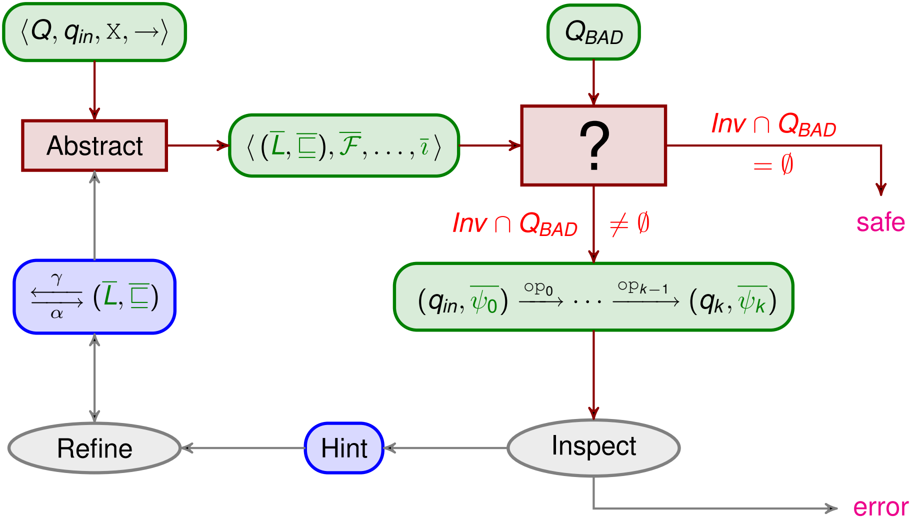
- either "program safe" if it finds an invariant *Inv* disjoint from *Bad*
- or "alarm" with an abstract run as a potential counterexample

# Hypothetical Workflow Based on Abstract Runs

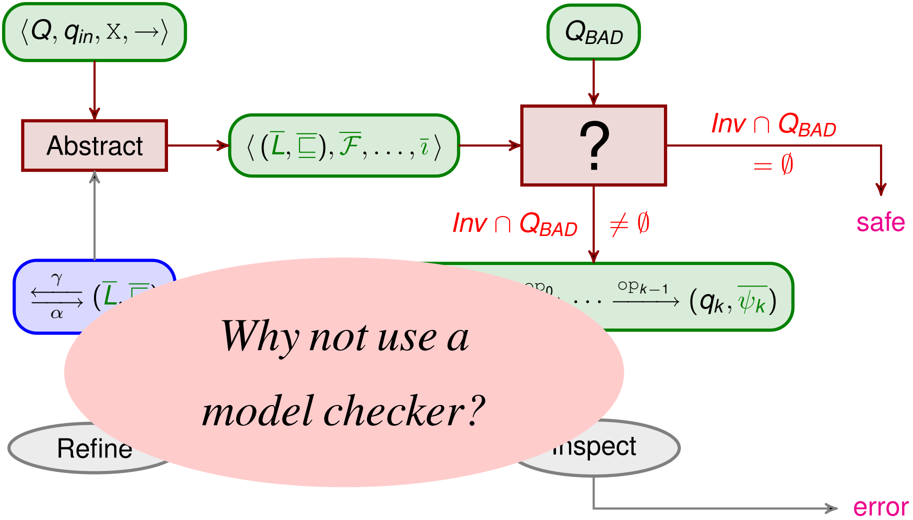


$\langle Q, q_{in}, \mathrm{X}, \rightarrow \rangle$
$Q_{BAD}$
Abstract
$\langle (\overline{L}, \overline{\sqsubseteq}), \overline{\mathcal{F}}, \ldots, \overline{\imath} \rangle$
Analyze
$Inv \cap Q_{BAD} \neq \emptyset$
safe
$\xleftarrow[\alpha]{\gamma} (\overline{L}, \overline{\sqsubseteq})$
$Inv \supseteq \mathrm{Post}^*$
$Inv \cap Q_{BAD} = \emptyset$
Refine
Hint
Inspect
error

# Hypothetical Workflow Based on Abstract Runs


$\langle Q, q_{in}, \mathrm{X}, \rightarrow \rangle$
Abstract
$\langle (\overline{L}, \overline{\sqsubseteq}), \overline{\mathcal{F}}, \ldots, \overline{\imath} \rangle$
$Q_{BAD}$
?
$Inv \cap Q_{BAD}$
$= \emptyset$
safe
$Inv \cap Q_{BAD}$
$\neq \emptyset$
$\underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (\overline{L}, \overline{\sqsubseteq})$
$(q_{in}, \overline{\psi_0}) \xrightarrow{\mathrm{op}_0} \cdots \xrightarrow{\mathrm{op}_{k-1}} (q_k, \overline{\psi_k})$
Refine
Hint
Inspect
error

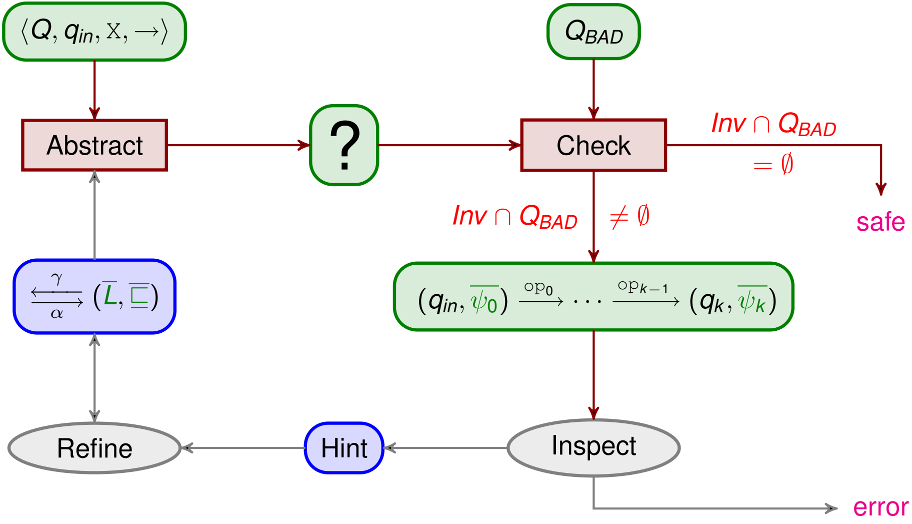
# Hypothetical Workflow Based on Abstract Runs



$\langle Q, q_{in}, \mathrm{X}, \rightarrow \rangle$
$Q_{BAD}$
Abstract
$\langle (\overline{L}, \overline{\sqsubseteq}), \overline{\mathcal{F}}, \ldots, \bar{\imath} \rangle$
?
$Inv \cap Q_{BAD}$
$= \emptyset$
safe
$Inv \cap Q_{BAD}$
$\neq \emptyset$
$\xleftarrow[\alpha]{\gamma} (\overline{L}, \overline{\sqsubseteq})$
$\xrightarrow{op_0} \cdots \xrightarrow{op_{k-1}} (q_k, \overline{\psi_k})$
Refine
Inspect
error

*Why not use a model checker?*

# Hypothetical Workflow Based on Abstract Runs



$\langle Q, q_{in}, \mathrm{X}, \rightarrow \rangle$

$Q_{BAD}$

Abstract

?

Check

$Inv \cap Q_{BAD} = \emptyset$

safe

$Inv \cap Q_{BAD} \neq \emptyset$

$\underset{\alpha}{\overset{\gamma}{\leftrightarrows}} (\overline{L}, \overline{\sqsubseteq})$

$(q_{in}, \overline{\psi_0}) \xrightarrow{\mathrm{op}_0} \cdots \xrightarrow{\mathrm{op}_{k-1}} (q_k, \overline{\psi_k})$

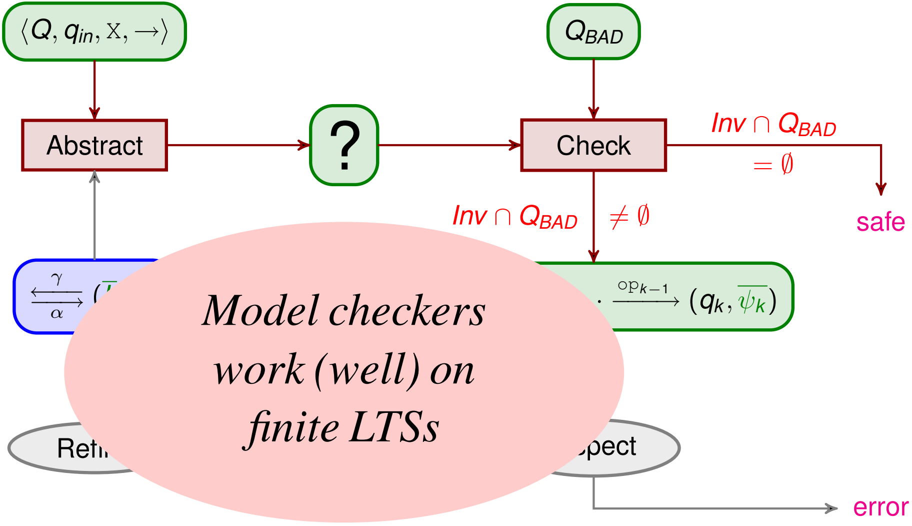
Refine

Hint

Inspect

error

# Hypothetical Workflow Based on Abstract Runs



$\langle Q, q_{in}, \mathrm{X}, \rightarrow \rangle$

$Q_{BAD}$

Abstract

?

Check

$Inv \cap Q_{BAD}$ $= \emptyset$

safe

$Inv \cap Q_{BAD}$ $\neq \emptyset$

$\underset{\alpha}{\overset{\gamma}{\leftrightarrows}}$ $($

$\cdot \xrightarrow{\mathrm{op}_{k-1}} (q_k, \overline{\psi_k})$

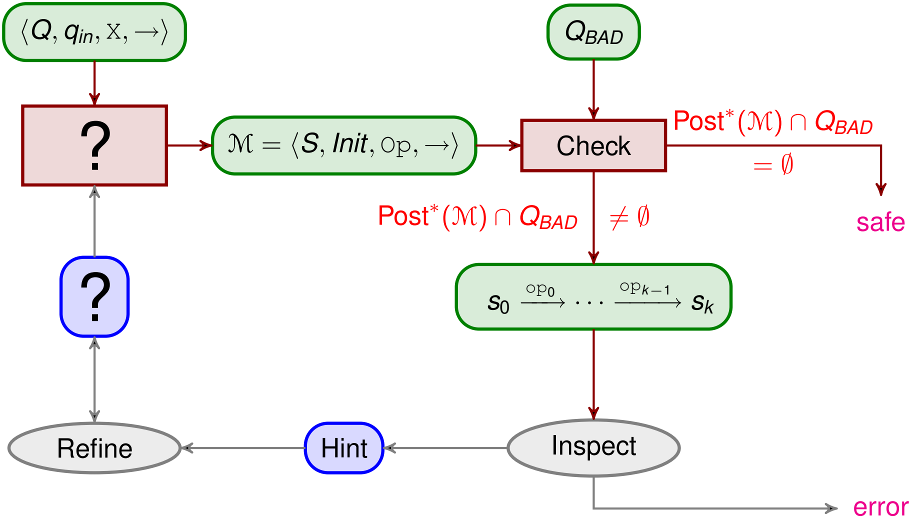
Refi

pect

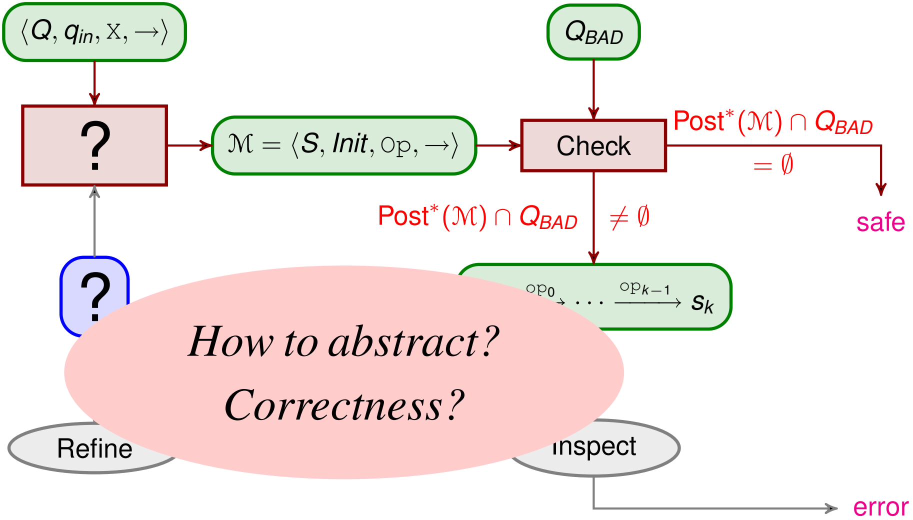
error

*Model checkers work (well) on finite LTSs*

# Hypothetical Workflow Based on Abstract Runs


$\langle Q, q_{in}, \mathrm{X}, \rightarrow \rangle$
?
$\mathcal{M} = \langle S, Init, \mathrm{Op}, \rightarrow \rangle$
$Q_{BAD}$
Check
$\mathrm{Post}^*(\mathcal{M}) \cap Q_{BAD}$
$= \emptyset$
safe
$\mathrm{Post}^*(\mathcal{M}) \cap Q_{BAD}$
$\neq \emptyset$
?
$s_0 \xrightarrow{\mathrm{op}_0} \cdots \xrightarrow{\mathrm{op}_{k-1}} s_k$
Refine
Hint
Inspect
error

# Hypothetical Workflow Based on Abstract Runs



$\langle Q, q_{in}, \mathtt{X}, \rightarrow \rangle$

?

$\mathcal{M} = \langle S, Init, \mathtt{Op}, \rightarrow \rangle$

$Q_{BAD}$

Check

$\mathrm{Post}^*(\mathcal{M}) \cap Q_{BAD} = \emptyset$

safe

$\mathrm{Post}^*(\mathcal{M}) \cap Q_{BAD} \neq \emptyset$

$\xrightarrow{\mathtt{op}_0} \cdots \xrightarrow{\mathtt{op}_{k-1}} s_k$

?

Refine

Inspect

error

*How to abstract?*

*Correctness?*

# Verification by Model-Checking Abstract Models

This hypothetical workflow… … is not hypothetical at all!

## Automatic Generation of Property-Preserving Abstractions

- First designed for large finite-state concurrent systems
- Inspired from abstract interpretation (use of Galois connections)
- Extended to (infinite-state) programs with theorem provers

## Credits: Pioneers (1990's)

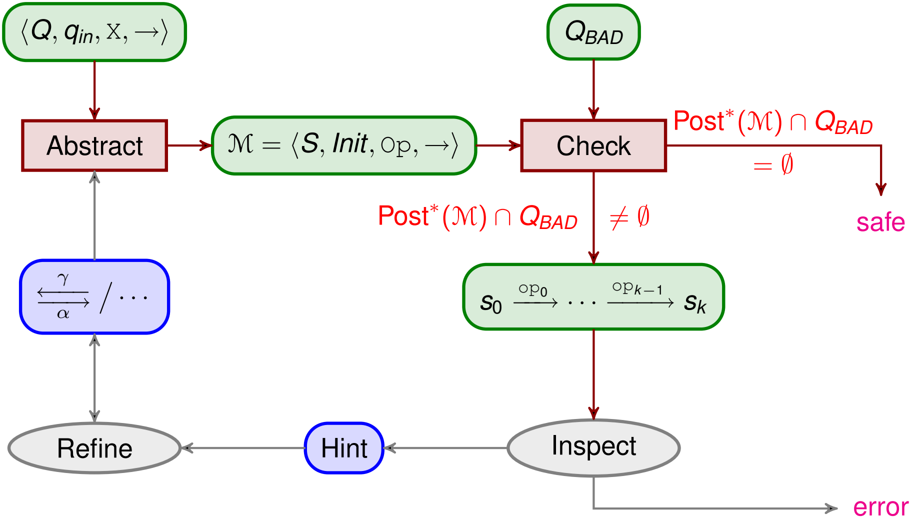
Joseph Sifakis & Claire Loiseaux

Dennis Dams & Rob Gerth & Orna Grumberg

Susanne Graf & Hassen Saïdi

…

# Verification by Model-Checking Abstract Models



$\langle Q, q_{in}, \mathtt{X}, \rightarrow \rangle$
$Q_{BAD}$
Abstract
$\mathcal{M} = \langle S, Init, \mathtt{Op}, \rightarrow \rangle$
Check
$\mathrm{Post}^*(\mathcal{M}) \cap Q_{BAD}$
$= \emptyset$
safe
$\mathrm{Post}^*(\mathcal{M}) \cap Q_{BAD}$
$\neq \emptyset$
$\underset{\alpha}{\overset{\gamma}{\rightleftarrows}} / \cdots$
$s_0 \xrightarrow{\mathtt{op}_0} \cdots \xrightarrow{\mathtt{op}_{k-1}} s_k$
Refine
Hint
Inspect
error

# Automatic Inspection and Refinement: a Dream?

## Goal

Automate the tasks Inspect and Refine

## Counterexample Guided Refinement (2000)

- First designed for large finite-state systems (hardware)
- Extended to (infinite-state) programs with theorem provers
- Subject of active research

## Credits: Pioneers (2000)

Edmund Clarke & Orna Grumberg

Thomas Ball & Sriram Rajamani

...

# Summary and Outlook: Key Ingredients

## Property-Preserving Abstraction

Conservatively extract finite-state models from programs

## Model-Checking

Can use a readily available finite-state model checker ☺

## Inspection of Abstract Counterexamples

Reduces to satisfiability checking (use of theorem provers)

## Refinement Guided by Abstract Counterexamples

Driven by the safety property to check: precision where required

Monotonic: the model after refinement has less counterexamples

All these tasks can be automated ☺

# Outline — Abstract Model Refinement

15 Introduction and Overview

16 Basic Theory on Property-Preserving Abstractions

17 Abstraction Schemes

18 Counterexample Guided Refinement

# Objectives of the Basic Theory

## Property-Preserving Abstraction

Conservatively extract finite-state models from programs

We focus on safety properties

## Model

Labeled Kripke Structure

=

LTS + Bad

## Notations

Concrete LKS: $\mathcal{M}^c$

Abstract LKS: $\mathcal{M}^a$

## Theory Intentionally Limited (Only What We Need. . . )

- Notions of abstraction and refinement (simpler than $\underset{\alpha}{\overset{\gamma}{\leftrightarrows}}$ ☺)
- Theorem for preservation of safety

# Labeled Kripke Structures for Safety

## Definition

A labeled Kripke structure is a quintuple $\langle S, \mathit{Init}, \mathit{Bad}, \Sigma, \rightarrow\rangle$ where :

- $S$ is a set of *states*
- $\mathit{Init} \subseteq S$ is a set of *initial states*
- $\mathit{Bad} \subseteq S$ is a set of *bad states*
- $\Sigma$ is a finite set of *actions*
- $\rightarrow \subseteq S \times \Sigma \times S$ is a set of *transitions*

## Simplified Definition!

Kripke structures are classically defined with a mapping from $S$ to $\mathcal{P}(AP)$ where $AP$ is a finite set of atomic propositions.

In our context $AP = \{\mathit{bad}\}$, hence it suffices to take $\mathit{Bad} \subseteq S$.

# Labeled Kripke Structures for Safety

## Definition

A labeled Kripke structure is a quintuple $\langle S, \mathit{Init}, \mathit{Bad}, \Sigma, \rightarrow\rangle$ where :

- $S$ is a set of *states*
- $\mathit{Init} \subseteq S$ is a set of *initial states*
- $\mathit{Bad} \subseteq S$ is a set of *bad states*
- $\Sigma$ is a finite set of *actions*
- $\rightarrow \subseteq S \times \Sigma \times S$ is a set of *transitions*

## Simplified Definition!

Kripke structures are classically defined with a mapping from $S$ to $\mathcal{P}(AP)$ where $AP$ is a finite set of atomic propositions.

In our context $AP = \{\mathit{bad}\}$, hence it suffices to take $\mathit{Bad} \subseteq S$.

# Lab. Kripke Structures Versus Lab. Transition Systems

## Labeled Transition System

$$\langle C, \mathit{Init}, \Sigma, \rightarrow \rangle$$

Elements of $C$ are called configurations.

Use: concrete operational semantics of control flow automata.

## Labeled Kripke Structures

$$\begin{aligned} \mathcal{M} &= \langle S, \mathit{Init}, \mathit{Bad}, \Sigma, \rightarrow \rangle \\ &= \mathit{LTS} + \mathit{Bad} \end{aligned}$$

Elements of $S$ are called states.

Use: models (in general abstract ones) for abstraction refinement.

# Simulation Relation: Definition

Consider two labeled Kripke structures:

$$\mathcal{M}^c \quad = \quad \langle S^c, Init^c, Bad^c, \Sigma, \rightarrow^c \rangle \qquad\qquad \mathcal{M}^a \quad = \quad \langle S^a, Init^a, Bad^a, \Sigma, \rightarrow^a \rangle$$

A simulation relation from $\mathcal{M}^c$ to $\mathcal{M}^a$ is any binary relation $\prec \; \subseteq S^c \times S^a$ satisfying:

$$\begin{tikzcd} \forall & & \\ & s^c \arrow[d, "\sigma"'] \arrow[r, phantom, "\prec"] & s^a \\ & t^c & \end{tikzcd}$$

# Simulation Relation: Definition

Consider two labeled Kripke structures:

$$\mathcal{M}^c = \langle S^c, \mathit{Init}^c, \mathit{Bad}^c, \Sigma, \rightarrow^c \rangle \qquad \mathcal{M}^a = \langle S^a, \mathit{Init}^a, \mathit{Bad}^a, \Sigma, \rightarrow^a \rangle$$

A simulation relation from $\mathcal{M}^c$ to $\mathcal{M}^a$ is any binary relation $\prec \subseteq S^c \times S^a$ satisfying:

$$\begin{tikzcd}
\forall & & & \\
& s^c \arrow[r, phantom, "\prec"] \arrow[d, "\sigma"'] & s^a \arrow[d, "\sigma"] & \\
& t^c \arrow[r, phantom, "\prec"] & t^a & \\
& & & \exists
\end{tikzcd}$$

# Abstraction and Refinement

Consider two labeled Kripke structures:

$$\mathcal{M}^c = \langle S^c, Init^c, Bad^c, \Sigma, \rightarrow^c \rangle \qquad \mathcal{M}^a = \langle S^a, Init^a, Bad^a, \Sigma, \rightarrow^a \rangle$$

If there exists a simulation relation $\prec$ from $\mathcal{M}^c$ to $\mathcal{M}^a$ such that

$$\begin{cases} \forall s^c \in Init^c \quad \cdot \quad \exists s^a \in Init^a \quad \cdot \quad s^c \prec s^a \\ \forall (s^c, s^a) \in \prec \quad \cdot \quad s^c \in Bad^c \implies s^a \in Bad^a \end{cases}$$

then we say that:

$\mathcal{M}^a$ is an abstraction of $\mathcal{M}^c$

$\mathcal{M}^c$ is a refinement of $\mathcal{M}^a$

# Preservation of Safety Properties

A labeled Kripke structure $\mathcal{M} = \langle S, \mathit{Init}, \mathit{Bad}, \Sigma, \rightarrow \rangle$ is safe if it contains no path

$$s_0 \xrightarrow{\sigma_0} s_1 \cdots s_{k-1} \xrightarrow{\sigma_k} s_k \quad \text{with} \quad \begin{cases} s_0 \in \mathit{Init} \\ s_k \in \mathit{Bad} \end{cases}$$

**Theorem (Safety Preservation)**

*For any two labeled Kripke structures $\mathcal{M}^c$ and $\mathcal{M}^a$,*

*if $\mathcal{M}^a$ is an abstraction of $\mathcal{M}^c$ and $\mathcal{M}^a$ is safe then $\mathcal{M}^c$ is safe.*

The converse does not hold.

# Preservation of Safety Properties

A labeled Kripke structure $\mathcal{M} = \langle S, Init, Bad, \Sigma, \rightarrow \rangle$ is safe if it contains no path

$$s_0 \xrightarrow{\sigma_0} s_1 \cdots s_{k-1} \xrightarrow{\sigma_k} s_k \quad \text{with} \quad \begin{cases} s_0 \in Init \\ s_k \in Bad \end{cases}$$

## Theorem (Safety Preservation)

*For any two labeled Kripke structures $\mathcal{M}^c$ and $\mathcal{M}^a$,*

*if $\mathcal{M}^a$ is an abstraction of $\mathcal{M}^c$ and $\mathcal{M}^a$ is safe then $\mathcal{M}^c$ is safe.*

The converse does not hold.

# Preservation of Safety Properties: Application

We want to show that a concrete labeled Kripke structure $\mathcal{M}^c$ is safe.

If $\mathcal{M}^c$ cannot be directly model-checked then:

1. design an abstract labeled Kripke structure $\mathcal{M}^a$, simpler than $\mathcal{M}^c$, and exhibit a simulation relation $\prec$ that shows that $\mathcal{M}^a$ is an abstraction of $\mathcal{M}^c$.
2. check that $\mathcal{M}^a$ is safe

If $\mathcal{M}^a$ is safe then $\mathcal{M}^c$ is safe

However, If $\mathcal{M}^a$ is not safe then we cannot conclude that $\mathcal{M}^c$ is not safe.

# Preservation of Safety Properties: Completeness

Consider a labeled Kripke structure $\mathcal{M}^c = \langle S^c, Init^c, Bad^c, \Sigma, \rightarrow^c \rangle$.

$$(\mathcal{M}^a) \qquad \begin{array}{rcl} S^a & = & \{reach\} \\ \rightarrow^a & = & \{reach\} \times \Sigma \times \{reach\} \end{array} \qquad \begin{array}{rcl} Init^a & = & \{reach\} \\ Bad^a & = & \emptyset \end{array}$$

The relation $\prec \; = \; \text{Post}^*(\mathcal{M}^c) \times \{reach\}$ is obviously a simulation relation from $\mathcal{M}^c$ to $\mathcal{M}^a$. Note that $\mathcal{M}^a$ is safe. Moreover:

$$\text{if } \mathcal{M}^c \text{ is safe then } \begin{cases} \forall s^c \in Init^c \cdot \exists s^a \in Init^a \cdot s^c \prec s^a \\ \forall (s^c, s^a) \in \prec \cdot s^c \in Bad^c \implies s^a \in Bad^a \end{cases}$$

## Theorem (Relative Completeness)

*For any safe labeled Kripke structure $\mathcal{M}^c$, there exists a finite-state abstraction $\mathcal{M}^a$ of $\mathcal{M}^c$ such that $\mathcal{M}^a$ is safe.*

Finite-state abstractions are sufficient to prove safety of any model.

# Outline — Abstract Model Refinement

15 Introduction and Overview

16 Basic Theory on Property-Preserving Abstractions

17 Abstraction Schemes
- Partition Abstraction
- Boolean Predicate Abstraction
- Cartesian Predicate Abstraction
- Application of Predicate Abstraction to Programs
- Summary

18 Counterexample Guided Refinement

# Two Steps

Presentation of abstraction schemes at the Semantic Level

Forget about control flow automata and programs

But keep them in mind for intuitions ☺

Implementation of predicate abstraction for control flow automata

# Two Steps

Presentation of abstraction schemes at the Semantic Level

Forget about control flow automata and programs

But keep them in mind for intuitions ☺

Implementation of predicate abstraction for control flow automata

# Partition Abstraction: Definition

Consider a labeled Kripke structure $\mathcal{M}^c = \langle S^c, \mathit{Init}^c, \mathit{Bad}^c, \Sigma, \rightarrow^c \rangle$.

Partition given by $(S^a, \alpha)$ where $S^a$ is a finite set and $\alpha : S^c \rightarrow S^a$

## Partition Abstraction $\mathcal{M}^a$ Induced by $(S^a, \alpha)$

$$\begin{aligned}
\mathit{Init}^a &= \{\alpha(s^c) \mid s^c \in \mathit{Init}^c\} \\
\mathit{Bad}^a &= \{\alpha(s^c) \mid s^c \in \mathit{Bad}^c\} \\
\rightarrow^a &= \{(\alpha(s^c), \sigma, \alpha(t^c)) \mid (s^c, \sigma, t^c) \in \rightarrow^c\}
\end{aligned}$$

The simulation relation $\prec = \{(s^c, \alpha(s^c)) \mid s^c \in S^c\}$ shows that

$\mathcal{M}^a$ is an abstraction of $\mathcal{M}^c$

# Partition Abstraction: Explanation

### Partition $(S^a, \alpha)$

- $S^a$ finite set
- $\alpha : S^c \rightarrow S^a$

### Partition Abstraction Induced by $(S^a, \alpha)$

$$\begin{array}{lcll} \mathit{Init}^a & = & \{\alpha(s^c) \mid s^c \in \mathit{Init}^c\} & (\mathit{Bad}^a \ldots) \\ \rightarrow^a & = & \{(\alpha(s^c), \sigma, \alpha(t^c)) \mid (s^c, \sigma, t^c) \in \rightarrow^c\} & \end{array}$$

Induced equivalence relation $\sim$ defined by: $s^c \sim t^c$ if $\alpha(s^c) = \alpha(t^c)$.

### Abstraction Function $\alpha : S^c \rightarrow S^a$

All concrete states in an equivalence class are merged together.

### Induced Concretization Function $\gamma : S^a \rightarrow \mathcal{P}(S^c)$

$$\gamma(s^a) \;=\; \{s^c \mid \alpha(s^c) = s^a\}$$

### Not a Galois Connection

$(\alpha, \gamma)$ becomes a Galois Connection when lifted to powersets.

# Partition Abstraction: Explanation

## Partition $(S^a, \alpha)$

- $S^a$ finite set
- $\alpha : S^c \to S^a$

## Partition Abstraction Induced by $(S^a, \alpha)$

$$\begin{array}{rcl} \mathit{Init}^a & = & \{\alpha(s^c) \mid s^c \in \mathit{Init}^c\} \qquad (\mathit{Bad}^a \ldots) \\ \to^a & = & \{(\alpha(s^c), \sigma, \alpha(t^c)) \mid (s^c, \sigma, t^c) \in \to^c\} \end{array}$$

$\mathit{Init}^a$, $\mathit{Bad}^a$ and $\to^a$ are existential lifts of their concrete counterparts:

$$s^a \in \mathit{Init}^a \quad \text{iff} \quad \exists s^c \cdot \begin{cases} \alpha(s^c) = s^a & \wedge \\ s^c \in \mathit{Init}^c \end{cases}$$

$$(s^a, \sigma, t^a) \in \to^a \quad \text{iff} \quad \exists s^c \, \exists t^c \cdot \begin{cases} \alpha(s^c) = s^a & \wedge \\ \alpha(t^c) = t^a & \wedge \\ (s^c, \sigma, t^c) \in \to^c \end{cases}$$

# Partition Abstraction: Computation of $\mathcal{M}^a$

Consider a labeled Kripke structure $\mathcal{M}^c = \langle S^c, \mathit{Init}^c, \mathit{Bad}^c, \Sigma, \rightarrow^c \rangle$.

## Computation of $\mathit{Init}^a$

```
I ← ∅
foreach $\mathtt{s}^a \in S^a$
    if $\exists s^c \cdot (s^c \in \gamma(\mathtt{s}^a) \wedge s^c \in \mathit{Init}^c)$
        I ← I ∪ $\{\mathtt{s}^a\}$
return I
```

## Computation of $\rightarrow^a$

```
R ← ∅
foreach $(\mathtt{s}^a, \sigma, \mathtt{t}^a) \in S^a \times \Sigma \times S^a$
    if $\exists s^c \; \exists t^c \cdot (s^c \in \gamma(\mathtt{s}^a) \wedge t^c \in \gamma(\mathtt{t}^a) \wedge (s^c, \sigma, t^c) \in \rightarrow^c)$
        R ← R ∪ $\{(\mathtt{s}^a, \sigma, \mathtt{t}^a)\}$
return R
```

# Partition Abstraction: Implementation Issues

- Machine representation of $\alpha : S^c \rightarrow S^a$ or $\gamma : S^a \rightarrow \mathcal{P}(S^c)$
  - Examples: BDDs (if $S^c = \{0,1\}^n$), NDDs (if $S^c = \mathbb{Z}^n$), ...
- Algorithms to decide the conditions

$$\exists s^c \cdot (s^c \in \gamma(s^a) \ \wedge \ s^c \in Init^c)$$

$$\exists s^c \cdot (s^c \in \gamma(s^a) \ \wedge \ s^c \in Bad^c)$$

$$\exists s^c \ \exists t^c \cdot (s^c \in \gamma(s^a) \ \wedge \ t^c \in \gamma(t^a) \ \wedge \ (s^c, \sigma, t^c) \in \rightarrow^c)$$

## Partial Algorithms (yes / no / ?) Are Sufficient

Safety preservation from $\mathcal{M}^a$ to $\mathcal{M}^c$ still holds if $Init^a$, $Bad^a$ and $\rightarrow^a$ are larger than the “optimal ones”. We may soundly consider “?” as “yes”.

# Partition Abstraction: Refinement

Given two equivalence relations $\sim_1$ and $\sim_2$ on some set $S$, we say that $\sim_2$ is finer than $\sim_1$ if $\sim_2 \subseteq \sim_1$, or equivalently if each equivalence class of $\sim_1$ is a union of equivalence classes of $\sim_2$.

Consider two partitions $(S^a_1, \alpha_1)$ and $(S^a_2, \alpha_2)$.

If $\sim_2$ is finer than $\sim_1$ then $\mathcal{M}^a(S^a_2, \alpha_2)$ is a refinement of $\mathcal{M}^a(S^a_1, \alpha_1)$.

## Informally

To refine a partition abstraction, split some equivalence classes.

## Recomputation of $\mathcal{M}^a$ after refinement

- Refinement is local to equivalence classes that are split.
- If $\mathcal{M}^a$ is stored explicitly then the refined $\rightarrow^a$ can be efficiently computed from the previous $\rightarrow^a$.

# Predicate Language

## Predicates

Formulas in first-order logic over some vocabulary

## Example

For control flow automata, take the same vocabulary as in expressions:

$$\langle \ldots, -1, 0, 1, \ldots \; ; \; +, -, * \; ; \; <, \leq, =, \neq, \geq, > \rangle$$

At the semantic level, we view predicates as sets of states.

# Predicate Language

## Predicates

Formulas in first-order logic over some vocabulary

## Example

For control flow automata, take the same vocabulary as in expressions:

$$\langle \ldots, -1, 0, 1, \ldots \;;\; +, -, * \;;\; <, \leq, =, \neq, \geq, > \rangle$$

At the semantic level, we view predicates as sets of states.

# Boolean Predicate Abstraction: Definition

Consider a labeled Kripke structure $\mathcal{M}^c = \langle S^c, \mathit{Init}^c, \mathit{Bad}^c, \Sigma, \rightarrow^c \rangle$.

Support predicates given by a finite set $\Phi$ of subsets of $S^c$

### Characteristic Function of $\phi \in \Phi$

$$\begin{array}{rcl} \mathbf{1}_\phi \; : \; S^c & \rightarrow & \{0,1\} \\ s^c & \mapsto & \begin{cases} 1 & \text{if } s^c \in \phi \\ 0 & \text{if } s^c \notin \phi \end{cases} \end{array}$$

### Partition $(S^a_\Phi, \alpha_\Phi)$

$$\begin{array}{rcl} S^a_\Phi & = & \Phi \rightarrow \{0,1\} \\ \alpha_\Phi(s^c) & = & \lambda\,\phi\,.\,\mathbf{1}_\phi(s^c) \end{array}$$

### Boolean Predicate Abstraction $\mathcal{M}^a$ Induced by $\Phi$

Partition abstraction induced by the partition $(S^a_\Phi, \alpha_\Phi)$

# Boolean Predicate Abstraction: Explanation

## Partition $(S^a_\Phi, \alpha_\Phi)$

$$\begin{aligned} S^a_\Phi &= \Phi \to \{0, 1\} \\ \alpha_\Phi(s^c) &= \lambda\,\phi\,.\,\mathbf{1}_\phi(s^c) \end{aligned}$$

## Intuition

Abstract state: truth value for each predicate

$\alpha_\Phi$ merges concrete states that satisfy the same predicates.

## Induced Concretization Function $\gamma : S^a \to \mathcal{P}(S^c)$

$$\gamma_\Phi(s^a) = \bigcap_{s^a(\phi)=1} \phi \quad \cap \quad \bigcap_{s^a(\phi)=0} S^c \setminus \phi$$

## Not a Galois Connection

$(\alpha_\Phi, \gamma_\Phi)$ becomes a Galois Connection when lifted to powersets.

# Boolean Predicate Abstraction: Computation of $\mathcal{M}^a$

$Init^a$, $Bad^a$ and $\rightarrow^a$ can be computed as for partition abstractions, but:

### Exponential complexity

Number of abstract states: $2^{|\Phi|}$

Exponential number of decisions $\exists s^c \, \exists t^c \cdot (\cdots)$ to compute $\rightarrow^a$

Exploit the structure of the partition to get better algorithms (in practice)

### Computation of $\alpha(U) = \{\alpha(s^c) \mid s^c \in U\}$ where $U \subseteq S^c$

If $U \subseteq \phi$ then every $s^a \in \alpha(U)$ necessarily satisfies $s^a(\phi) = 1$.

In that case, there is no need to examine candidates where $s^a(\phi) = 0$.

$$\Phi_1 = \{\phi \in \Phi \mid U \subseteq \phi\} \qquad \Phi_0 = \{\phi \in \Phi \mid U \subseteq S^c \setminus \phi\}$$

New complexity linear in $|\Phi_0| + |\Phi_1|$ and exponential in $|\Phi \setminus (\Phi_0 \cup \Phi_1)|$

# Boolean Predicate Abstraction: Computation of $\mathcal{M}^a$

$Init^a$, $Bad^a$ and $\rightarrow^a$ can be computed as for partition abstractions, but:

### Exponential complexity

Number of abstract states: $2^{|\Phi|}$

Exponential number of decisions $\exists s^c \, \exists t^c \cdot (\cdots)$ to compute $\rightarrow^a$

Exploit the structure of the partition to get better algorithms (in practice)

### Computation of $\alpha(U) = \{\alpha(s^c) \mid s^c \in U\}$ where $U \subseteq S^c$

If $U \subseteq \phi$ then every $s^a \in \alpha(U)$ necessarily satisfies $s^a(\phi) = 1$.

In that case, there is no need to examine candidates where $s^a(\phi) = 0$.

$$\Phi_1 = \{\phi \in \Phi \mid U \subseteq \phi\} \qquad \Phi_0 = \{\phi \in \Phi \mid U \subseteq S^c \setminus \phi\}$$

New complexity linear in $|\Phi_0| + |\Phi_1|$ and exponential in $|\Phi \setminus (\Phi_0 \cup \Phi_1)|$

# Boolean Predicate Abstraction: Computation of $\mathcal{M}^a$

$Init^a$, $Bad^a$ and $\rightarrow^a$ can be computed as for partition abstractions, but:

## Exponential complexity

Number of abstract states: $2^{|\Phi|}$

Exponential number of decisions $\exists s^c \ \exists t^c \ \cdot \ (\cdots)$ to compute $\rightarrow^a$

Exploit the structure of the partition to get better algorithms (in practice)

## Computation of $\alpha(U) = \{\alpha(s^c) \mid s^c \in U\}$ where $U \subseteq S^c$

If $U \subseteq \phi$ then every $s^a \in \alpha(U)$ necessarily satisfies $s^a(\phi) = 1$.

In that case, there is no need to examine candidates where $s^a(\phi) = 0$.

$$\Phi_1 \ = \ \{\phi \in \Phi \mid U \subseteq \phi\} \qquad\qquad \Phi_0 \ = \ \{\phi \in \Phi \mid U \subseteq S^c \setminus \phi\}$$

New complexity linear in $|\Phi_0| + |\Phi_1|$ and exponential in $|\Phi \setminus (\Phi_0 \cup \Phi_1)|$

# Boolean Predicate Abstraction: Implementation Issues

Each abstract state is a truth valuation of the predicates.

Sets of abstract states (e.g. $Init^a$, $Bad^a$) are sets of truth valuations.

## Natural Encoding

Propositional Formulas

Introduce propositional variables $\mathrm{p}_\phi, \mathrm{p}'_\phi$ for each predicate $\phi$.

$$s^a \quad \leadsto \quad \bigwedge_{\phi \in \Phi} \overline{\mathrm{p}_\phi} \qquad \text{(conjunction of literals)}$$

$$Init^a, Bad^a \quad \leadsto \quad \bigvee \bigwedge_{\phi \in \Phi} \overline{\mathrm{p}_\phi} \qquad \text{(formula on } \mathrm{p}_\phi\text{)}$$

$$\rightarrow^a \quad \leadsto \quad \bigvee \bigwedge_{\phi \in \Phi} \overline{\mathrm{p}_\phi} \bigwedge_{\phi \in \Phi} \overline{\mathrm{p}'_\phi} \qquad \text{(formula on } \mathrm{p}_\phi, \mathrm{p}'_\phi\text{)}$$

Use BDDs to represent these propositional formulas ☺

# Boolean Predicate Abstraction: Implementation Issues

Each abstract state is a truth valuation of the predicates.

Sets of abstract states (e.g. $\mathit{Init}^a$, $\mathit{Bad}^a$) are sets of truth valuations.

## Natural Encoding

Propositional Formulas

Introduce propositional variables $\mathrm{p}_\phi, \mathrm{p}'_\phi$ for each predicate $\phi$.

$$s^a \quad \leftrightsquigarrow \quad \bigwedge_{\phi \in \Phi} \overline{\mathrm{p}_\phi} \qquad \text{(conjunction of literals)}$$

$$\mathit{Init}^a, \mathit{Bad}^a \quad \leftrightsquigarrow \quad \bigvee \bigwedge_{\phi \in \Phi} \overline{\mathrm{p}_\phi} \qquad \text{(formula on } \mathrm{p}_\phi\text{)}$$

$$\rightarrow^a \quad \leftrightsquigarrow \quad \bigvee \bigwedge_{\phi \in \Phi} \overline{\mathrm{p}_\phi} \bigwedge_{\phi \in \Phi} \overline{\mathrm{p}'_\phi} \qquad \text{(formula on } \mathrm{p}_\phi, \mathrm{p}'_\phi\text{)}$$

Use BDDs to represent these propositional formulas ☺

# Boolean Predicate Abstraction: Refinement

If $\Phi_2 \supseteq \Phi_1$ then $\mathcal{M}^a(\Phi_2)$ is a refinement of $\mathcal{M}^a(\Phi_1)$.

## Informally

To refine a boolean predicate abstraction, add new predicates.

## Recomputation of $\mathcal{M}^a$ after refinement

- Refinement is global, since it can impact all abstract states.

# Cartesian Predicate Abstraction: Introduction

Support predicates given by a finite set $\Phi$ of subsets of $S^c$

## Objective

Avoid exponential cost in the abstraction of a set $U$ of concrete states

A monomial is a conjunction of literals $\bigwedge_{\phi \in \Phi'} \overline{\mathrm{p}_\phi}$ for some $\Phi' \subseteq \Phi$.

## Solution

Replace disjunctions of abstract states by the most precise monomial.

$$\text{Boolean:} \quad U \subseteq S^c \quad \overset{\alpha}{\rightsquigarrow} \quad \bigvee \bigwedge_{\phi \in \Phi} \overline{\mathrm{p}_\phi}$$

$$\text{Cartesian:} \quad U \subseteq S^c \quad \overset{\alpha}{\rightsquigarrow} \quad \bigwedge_{\phi \in \Phi'} \overline{\mathrm{p}_\phi} \qquad (\Phi' \subseteq \Phi)$$

# Cartesian Predicate Abstraction: Trivectors

## Encoding of Monomials

Encode $\bigwedge_{\phi \in \Phi'} \overline{\mathrm{p}_\phi}$ as the valuation

$$v(\phi) = \begin{cases} 1 & \text{if } \overline{\mathrm{p}_\phi} = p_\phi \\ 0 & \text{if } \overline{\mathrm{p}_\phi} = \neg p_\phi \\ * & \text{if } \phi \notin \Phi' \end{cases}$$

## 3-Valued Characteristic Function

$$\mathbf{1}_\phi \ : \ \mathcal{P}(S^c) \ \rightarrow \ \{0, 1, *\}$$

$$U \neq \emptyset \ \mapsto \ \begin{cases} 1 & \text{if } U \subseteq \phi \\ 0 & \text{if } U \subseteq S^c \setminus \phi \\ * & \text{otherwise} \end{cases}$$

## Cartesian Abstraction and Concretization Functions

$$\begin{aligned} S^a_\Phi &= \Phi \rightarrow \{0, 1, *\} \\ \alpha_\Phi(U) &= \lambda \phi \,.\, \mathbf{1}_\phi(U) & (U \neq \emptyset) \\ \gamma_\Phi(s^a) &= \bigcap_{s^a(\phi)=1} \phi \ \cap \bigcap_{s^a(\phi)=0} S^c \setminus \phi \end{aligned}$$

# Cartesian Predicate Abstraction: Definition

## Notation: Concrete Post Operator

$$\mathrm{Post}^c(U, \sigma) \;=\; \{t^c \in \mathit{State}^c \mid \exists s^c \in U \;\cdot\; (s^c, \sigma, t^c) \in \rightarrow^c\}$$

## Cartesian Predicate Abstraction $\mathcal{M}^a$ Induced by $\Phi$

$$\begin{aligned}
S^a_\Phi &= \Phi \rightarrow \{0, 1, *\} \\
\mathit{Init}^a &= \{\alpha_\Phi(s^c) \mid s^c \in \mathit{Init}^c\} \\
\mathit{Bad}^a &= \{s^a \mid s^a \in S^a, \gamma_\Phi(s^a) \cap \mathit{Bad}^c \neq \emptyset\} \\
\rightarrow^a &= \{(s^a, \sigma, \alpha_\Phi \circ \mathrm{Post}^c(\gamma_\Phi(s^a), \sigma)) \mid s^a \in S^a, \sigma \in \Sigma\}
\end{aligned}$$

The simulation relation $\prec \;=\; \{(s^c, s^a) \mid s^c \in \gamma_\Phi(s^a)\}$ shows that

$\mathcal{M}^a$ is an abstraction of $\mathcal{M}^c$

# Cartesian Predicate Abstraction: Remarks

## Cartesian Predicate Abstraction $\mathcal{M}^a$ Induced by $\Phi$

$$\begin{aligned} S^a_\Phi &= \Phi \to \{0, 1, *\} \\ \gamma_\Phi(s^a) &= \bigcap_{s^a(\phi)=1} \phi \quad \cap \quad \bigcap_{s^a(\phi)=0} S^c \setminus \phi \\ \to^a &= \left\{(s^a, \sigma, \alpha_\Phi \circ \mathrm{Post}^c \circ \gamma_\Phi(s^a)) \;\middle|\; s^a \in S^a, \sigma \in \Sigma\right\} \end{aligned}$$

Abstract state: truth value in $\{0, 1, *\}$ for each $\phi \in \Phi$. Not a partition!

The special value $*$ is conservatively treated as “don’t know” in $\gamma_\Phi$.

The transition relation $\to^a$ is deterministic (at most one successor).

## Galois Connection

$(\alpha_\Phi, \gamma_\Phi)$ is a Galois Connection (with $0, 1 \sqsubseteq *$).

# Cartesian Predicate Abstraction: Computation of $\mathcal{M}^a$

## Computation of $Init^a$

Same as boolean case

## Computation of $\alpha(U)$

```
foreach $\phi \in \Phi$
    if $U \subseteq S^c \setminus \phi$
        $s^a[\phi] \leftarrow 0$
    else if $U \subseteq \phi$
        $s^a[\phi] \leftarrow 1$
    else
        $s^a[\phi] \leftarrow \star$
return $s^a$
```

## Computation of $\rightarrow^a$

```
$R \leftarrow \emptyset$
foreach $(s^a, \sigma) \in S^a \times \Sigma \mid \mathrm{Post}^c(\gamma(s^a), \sigma) \neq \emptyset$
    foreach $\phi \in \Phi$
        if $\mathrm{Post}^c(\gamma(s^a), \sigma) \subseteq S^c \setminus \phi$
            $t^a[\phi] \leftarrow 0$
        else if $\mathrm{Post}^c(\gamma(s^a), \sigma) \subseteq \phi$
            $t^a[\phi] \leftarrow 1$
        else
            $t^a[\phi] \leftarrow \star$
    $R \leftarrow R \cup \{(s^a, \sigma, t^a)\}$
return $R$
```

Linear number of decisions $\mathrm{Post}^c(\gamma(s^a), \sigma) \subseteq \ldots$ to compute the successor $\rightarrow^a (s^a, \sigma)$ of a given abstract state $s^a$ and action $\sigma \in \Sigma$.

# Cartesian Pred. Abstraction: Implementation Issues

Similar to boolean predicate abstraction:

- Encoding with 3-valued propositional variables $\mathtt{p}_\phi, \mathtt{p}'_\phi$
- Representation with TDDs (or BDDs via binary encoding)

For concrete labeled Kripke structures obtained from programs, the cartesian predicate abstraction can be presented as boolean program.

# Cartesian Predicate Abstraction: Refinement

If $\Phi_2 \supseteq \Phi_1$ then $\mathcal{M}^a(\Phi_2)$ is a refinement of $\mathcal{M}^a(\Phi_1)$.

## Informally

To refine a cartesian predicate abstraction, add new predicates.

## Recomputation of $\mathcal{M}^a$ after refinement

- Refinement is global, since it can impact all abstract states.

# How about Programs?

Control flow automaton: $\langle Q, q_{in}, \mathbb{X}, \rightarrow \rangle$. Set $Q_{BAD} \subseteq Q$ of bad locations.

## Concrete Labeled Kripke Structure $\mathcal{M}^c$

$$\begin{array}{rclrcl} S^c & = & Q \times (\mathbb{X} \rightarrow \mathbb{R}) & \mathit{Init}^c & = & \{q_{in}\} \times (\mathbb{X} \rightarrow \mathbb{R}) \\ \Sigma & = & \mathrm{Op} & \mathit{Bad}^c & = & Q_{BAD} \times (\mathbb{X} \rightarrow \mathbb{R}) \end{array}$$

$$\rightarrow^c \;=\; \left\{ ((q, u^c), \sigma, (q', v^c)) \;\middle|\; q \xrightarrow{\mathrm{op}} q' \text{ and } (u^c, v^c) \in [\![\mathrm{op}]\!]^c \right\}$$

The usual semantics $[\![\mathrm{op}]\!]$ of operations is now written $[\![\mathrm{op}]\!]^c$.

## Nothing Surprising Here!

This is the usual labeled transition system (operational semantics of control flow automata) equipped with the usual bad configurations.

# Predicate Language

Control flow automaton: $\langle Q, q_{in}, \mathtt{X}, \rightarrow \rangle$.

## Vocabulary

$$\langle \ldots, -1, 0, 1, \ldots \; ; \; +, -, * \; ; \; <, \leq, =, \neq, \geq, > \rangle$$

Additive and multiplicative theory of the reals is decidable.

## Finite Set Φ of Support Predicates

(Quantifier-free) first-order formulas with free variables in $\mathtt{X}$

## Semantics of Support Predicates

The interpretation $[\![\varphi]\!]$ of a predicate $\varphi$ is a subset of $\mathtt{X} \rightarrow \mathbb{R}$.

## Link With Semantic Level Abstraction Schemes

The interpretations $[\![\varphi]\!]$ replace the "semantic support predicates" $\phi$.

# Boolean Predicate Abstraction: Definition

## Boolean Predicate Abstraction $\mathcal{M}^a$ Induced by $\Phi$

$$
\begin{array}{rcl@{\qquad}rcl}
S^a & = & Q \times (\Phi \to \{0,1\}) & Init^a & = & \{q_{in}\} \times (\Phi \to \{0,1\}) \\
\Sigma & = & \mathtt{Op} & Bad^a & = & Q_{BAD} \times (\Phi \to \{0,1\}) \\
\to^a & = & \multicolumn{4}{l}{\left\{((q,u^a),\sigma,(q',v^a)) \;\middle|\; q \xrightarrow{\mathtt{op}} q' \text{ and } (u^a,v^a) \in [\![\mathtt{op}]\!]^a\right\}}
\end{array}
$$

- Concrete valuations in $\mathtt{X} \to \mathbb{R}$ are replaced by abstract valuations in $\Phi \to \{0,1\}$.
- The control flow automaton's graph is kept intact.
- All the work is done in the abstract semantics of operations.

# Boolean Predicate Abstraction: Definition

## Syntactic Concretization

Concretization formula $\gamma(v^a)$ of a valuation $v^a \in \Phi \rightarrow \{0, 1\}$ defined by

$$\gamma(v^a) \quad = \quad \bigwedge_{\substack{\varphi \in \Phi \\ v^a(\varphi)=1}} \varphi \quad \wedge \quad \bigwedge_{\substack{\varphi \in \Phi \\ v^a(\varphi)=0}} \neg\varphi$$

Abstract semantics $[\![\mathrm{op}]\!]^a$ of operations defined as a binary relation

$$[\![\mathrm{op}]\!]^a \quad \subseteq \quad (\Phi \rightarrow \{0, 1\}) \times (\Phi \rightarrow \{0, 1\})$$

Guards: $(u^a, v^a) \in [\![g]\!]^a$ if $v^a = u^a$ and $\gamma(u^a) \wedge g$ sat.

Assignments: $(u^a, v^a) \in [\![x := e]\!]^a$ if $\gamma(u^a) \wedge \gamma(v^a)[e/x]$ sat.

# Boolean Predicate Abstraction: Computation of $\mathcal{M}^a$

Safety checking of $\mathcal{M}^a$ usually performed by forward graph exploration.

## Computation of $\{v^a \in \Phi \to \{0,1\} \mid (u^a, v^a) \in [\![g]\!]^a\}$

```
if $\gamma(u^a) \wedge g$ is satisfiable
    return $\{u^a\}$
else
    return $\emptyset$
```

## Computation of $\{v^a \in \Phi \to \{0,1\} \mid (u^a, v^a) \in [\![x := e]\!]^a\}$

```
S $\leftarrow \emptyset$
foreach $v^a \in \Phi \to \{0,1\}$                    (exponential ☹)
    if $\gamma(u^a) \wedge \gamma(v^a)[e/x]$ is satisfiable
        S $\leftarrow$ S $\cup$ $\{v^a\}$
return S
```

# Cartesian Predicate Abstraction: Definition

## Cartesian Predicate Abstraction $\mathcal{M}^a$ Induced by $\Phi$

$$
\begin{aligned}
S^a &= Q \times (\Phi \to \{0, 1, *\}) & \quad Init^a &= \{q_{in}\} \times \{\lambda\,\varphi\,.\,*\} \\
\Sigma &= \mathtt{Op} & \quad Bad^a &= Q_{BAD} \times \{\lambda\,\varphi\,.\,*\} \\
\to^a &= \left\{ ((q, u^a), \sigma, (q', v^a)) \;\middle|\; q \xrightarrow{\mathtt{op}} q' \text{ and } v^a = [\![\mathtt{op}]\!]^a(u^a) \right\}
\end{aligned}
$$

## Syntactic Concretization

Concretization formula $\gamma(v^a)$ of a valuation $v^a \in \Phi \to \{0, 1, *\}$ defined by

$$
\gamma(v^a) = \bigwedge_{\substack{\varphi \in \Phi \\ v^a(\varphi)=1}} \varphi \quad \wedge \quad \bigwedge_{\substack{\varphi \in \Phi \\ v^a(\varphi)=0}} \neg\varphi
$$

# Cartesian Predicate Abstraction: Definition

Abstract semantics $[\![\mathrm{op}]\!]^a$ of operations defined as a partial function

$$[\![\mathrm{op}]\!]^a : (\Phi \to \{0,1,*\}) \to (\Phi \to \{0,1,*\})$$

## Guards

If $\gamma(u^a) \ \wedge \ g$ is unsatisfiable then $[\![\mathrm{op}]\!]^a(u^a)$ is undefined

$$\text{Otherwise} \quad [\![\mathrm{op}]\!]^a(u^a) \ = \ \lambda\,\varphi\,.\begin{cases} 0 & \text{if } \ (\gamma(u^a) \ \wedge \ g) \Rightarrow \neg\varphi \ \text{ is valid} \\ 1 & \text{if } \ (\gamma(u^a) \ \wedge \ g) \Rightarrow \varphi \ \text{ is valid} \\ * & \text{otherwise} \end{cases}$$

## Assignments

If $\gamma(u^a)$ is unsatisfiable then $[\![x := e]\!]^a(u^a)$ is undefined

$$\text{Otherwise} \quad [\![x := e]\!]^a(u^a) \ = \ \lambda\,\varphi\,.\begin{cases} 0 & \text{if } \ \gamma(u^a) \Rightarrow \neg\varphi[e/x] \ \text{ is valid} \\ 1 & \text{if } \ \gamma(u^a) \Rightarrow \varphi[e/x] \ \text{ is valid} \\ * & \text{otherwise} \end{cases}$$

# Cartesian Predicate Abstraction: Computation of $\mathcal{M}^a$

Safety checking of $\mathcal{M}^a$ usually performed by forward graph exploration.

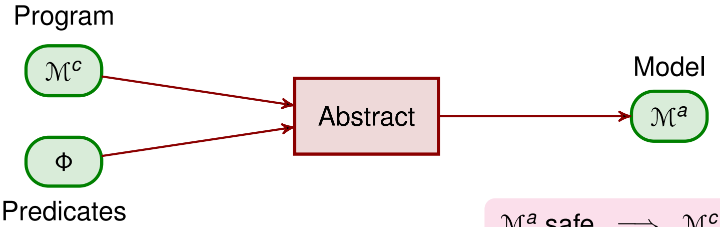
## Computation of $[\![g]\!]^a(u^a)$

```
if $\gamma(u^a) \ \wedge \ g$ is unsatisfiable
    return undefined
foreach $\varphi \in \Phi$          (linear ☺)
    if $\models (\gamma(u^a) \wedge g) \Rightarrow \neg\varphi$
        $v^a[\varphi] \leftarrow 0$
    else if $\models (\gamma(u^a) \wedge g) \Rightarrow \varphi$
        $v^a[\varphi] \leftarrow 1$
    else
        $v^a[\varphi] \leftarrow \star$
return $v^a$
```

## Computation of $[\![x := e]\!]^a(u^a)$

```
if $\gamma(u^a)$ is unsatisfiable
    return undefined
foreach $\varphi \in \Phi$          (linear ☺)
    if $\models \gamma(u^a) \Rightarrow \neg\varphi[e/x]$
        $v^a[\varphi] \leftarrow 0$
    else if $\models \gamma(u^a) \Rightarrow \varphi[e/x]$
        $v^a[\varphi] \leftarrow 1$
    else
        $v^a[\varphi] \leftarrow \star$
return $v^a$
```

# Summary: Automatic Predicate Abstraction



Program

$\mathcal{M}^c$

$\Phi$

Predicates

Abstract

Model

$\mathcal{M}^a$

$\mathcal{M}^a$ safe $\Longrightarrow$ $\mathcal{M}^c$ safe

Refinement consists in adding new support predicates

## Boolean Abstraction

Partition induced by $\Phi$

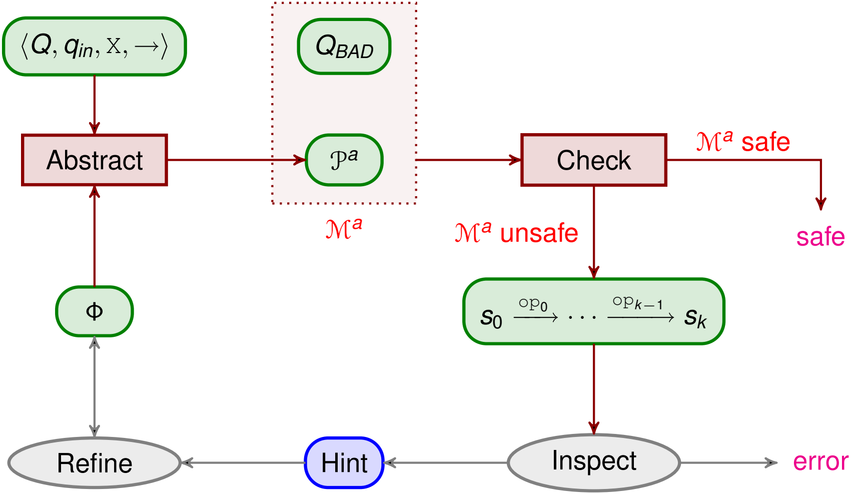
- ☺ Most precise abstraction based of $\Phi$
- ☹ Exponential (for successors)

## Cartesian Abstraction

Monomials induced by $\Phi$

- ☹ Less precise than boolean abstraction
- ☺ Linear (for successors)

# Verification by Model-Checking Abstract Models



$\langle Q, q_{in}, \mathrm{X}, \rightarrow\rangle$
$Q_{BAD}$
Abstract
$\mathcal{P}^a$
Check
$\mathcal{M}^a$ safe
$\mathcal{M}^a$
$\mathcal{M}^a$ unsafe
safe
$\Phi$
$s_0 \xrightarrow{\mathrm{op}_0} \cdots \xrightarrow{\mathrm{op}_{k-1}} s_k$
Refine
Hint
Inspect
error

# Outline — Abstract Model Refinement

15 Introduction and Overview

16 Basic Theory on Property-Preserving Abstractions

17 Abstraction Schemes

18 Counterexample Guided Refinement

- Inspection of Abstract Counterexamples
- Counterexample Guided Refinement
- Counterexample Guided Abstraction Refinement Algorithms

# Inspection of Abstract Counterexamples

Control flow automaton: $\langle Q, q_{in}, \mathrm{X}, \rightarrow\rangle$. Set $Q_{BAD} \subseteq Q$ of bad locations.

$\mathcal{M}^a = \langle S^a, Init^a, Bad^a, \Sigma, \rightarrow^a\rangle$ obtained by predicate abstraction

### Abstract counterexample

$$(q_{in}, v_0^a) \xrightarrow{\mathrm{op}_0} (q_1, v_1^a) \cdots (q_k, v_k^a) \xrightarrow{\mathrm{op}_k} (q_{bad}, v_{k+1}^a)$$

The abstract counterexample is feasible if there is a concrete run

$$(q_{in}, v_0^c) \xrightarrow{\mathrm{op}_0} (q_1, v_1^c) \cdots (q_k, v_k^c) \xrightarrow{\mathrm{op}_k} (q_{bad}, v_{k+1}^c) \quad \text{with } v_i^c \in \gamma(v_i^a)$$

Better to directly check for all possible abstract predicate valuations!

### Objective

Check whether a control path $q_{in} \xrightarrow{\mathrm{op}_0} q_1 \cdots q_k \xrightarrow{\mathrm{op}_k} q_{bad}$ is feasible

# Inspection of Abstract Counterexamples

Control flow automaton: $\langle Q, q_{in}, \mathtt{X}, \rightarrow \rangle$. Set $Q_{BAD} \subseteq Q$ of bad locations.

$\mathcal{M}^a = \langle S^a, Init^a, Bad^a, \Sigma, \rightarrow^a \rangle$ obtained by predicate abstraction

## Abstract counterexample

$$(q_{in}, v_0^a) \xrightarrow{\mathrm{op}_0} (q_1, v_1^a) \cdots (q_k, v_k^a) \xrightarrow{\mathrm{op}_k} (q_{bad}, v_{k+1}^a)$$

The abstract counterexample is feasible if there is a concrete run

$$(q_{in}, v_0^c) \xrightarrow{\mathrm{op}_0} (q_1, v_1^c) \cdots (q_k, v_k^c) \xrightarrow{\mathrm{op}_k} (q_{bad}, v_{k+1}^c) \quad \text{with } v_i^c \in \gamma(v_i^a)$$

Better to directly check for all possible abstract predicate valuations!

## Objective

Check whether a control path $q_{in} \xrightarrow{\mathrm{op}_0} q_1 \cdots q_k \xrightarrow{\mathrm{op}_k} q_{bad}$ is feasible

# Inspection of Abstract Counterexamples

Control flow automaton: $\langle Q, q_{in}, \mathtt{X}, \rightarrow \rangle$. Set $Q_{BAD} \subseteq Q$ of bad locations.

$\mathcal{M}^a = \langle S^a, Init^a, Bad^a, \Sigma, \rightarrow^a \rangle$ obtained by predicate abstraction

## Abstract counterexample

$$(q_{in}, v_0^a) \xrightarrow{\mathrm{op}_0} (q_1, v_1^a) \cdots (q_k, v_k^a) \xrightarrow{\mathrm{op}_k} (q_{bad}, v_{k+1}^a)$$

The abstract counterexample is feasible if there is a concrete run

$$(q_{in}, v_0^c) \xrightarrow{\mathrm{op}_0} (q_1, v_1^c) \cdots (q_k, v_k^c) \xrightarrow{\mathrm{op}_k} (q_{bad}, v_{k+1}^c) \quad \text{with } v_i^c \in \gamma(v_i^a)$$

Better to directly check for all possible abstract predicate valuations!

## Objective

Check whether a control path $q_{in} \xrightarrow{\mathrm{op}_0} q_1 \cdots q_k \xrightarrow{\mathrm{op}_k} q_{bad}$ is feasible

# Checking Feasibility of Control Paths

## Feasibility at the Semantic Level

$$q_{in} \xrightarrow{\mathrm{op}_0} q_1 \cdots q_k \xrightarrow{\mathrm{op}_k} q_{bad} \text{ feasible} \quad \text{iff} \quad [\![\mathrm{op}_k]\!] \circ \cdots \circ [\![\mathrm{op}_0]\!] \neq \emptyset$$

Recall that expressions $e$ used in guards and assignments are over $\mathrm{X}$.

## Syntactic Effect of Operations: Formula $\langle\!\langle \mathrm{op} \rangle\!\rangle$ over $\mathrm{X}, \mathrm{X}'$

$$\langle\!\langle g \rangle\!\rangle \;=\; g \wedge \bigwedge_{x \in \mathrm{X}} x' = x \qquad \langle\!\langle x := e \rangle\!\rangle \;=\; x' = e \;\wedge \bigwedge_{y \in \mathrm{X}, y \neq x} y' = y$$

For each $\mathrm{op} \in \mathrm{Op}$: $[\![\langle\!\langle \mathrm{op} \rangle\!\rangle]\!] \;=\; [\![\mathrm{op}]\!]$

Multiply-primed copies of variables: $x^{(i)}$ is the copy of $x$ with $i$ primes.

## Feasibility at the Syntactic Level

$$q_{in} \xrightarrow{\mathrm{op}_0} q_1 \cdots q_k \xrightarrow{\mathrm{op}_k} q_{bad} \text{ feasible} \quad \text{iff} \quad \langle\!\langle \mathrm{op}_0 \rangle\!\rangle^{(0)} \wedge \cdots \wedge \langle\!\langle \mathrm{op}_k \rangle\!\rangle^{(k)} \text{ sat.}$$

# Checking Feasibility of Control Paths

## Feasibility at the Semantic Level

$q_{in} \xrightarrow{\mathtt{op}_0} q_1 \cdots q_k \xrightarrow{\mathtt{op}_k} q_{bad}$ feasible iff $[\![\mathtt{op}_k]\!] \circ \cdots \circ [\![\mathtt{op}_0]\!] \neq \emptyset$

Recall that expressions $e$ used in guards and assignments are over $\mathtt{X}$.

## Syntactic Effect of Operations: Formula $\langle\!\langle \mathtt{op} \rangle\!\rangle$ over $\mathtt{X}, \mathtt{X}'$

$$\langle\!\langle g \rangle\!\rangle = g \wedge \bigwedge_{x \in \mathtt{X}} x' = x \qquad \langle\!\langle x := e \rangle\!\rangle = x' = e \wedge \bigwedge_{y \in \mathtt{X}, y \neq x} y' = y$$

For each $\mathtt{op} \in \mathtt{Op}$: $[\![\langle\!\langle \mathtt{op} \rangle\!\rangle]\!] = [\![\mathtt{op}]\!]$

Multiply-primed copies of variables: $x^{(i)}$ is the copy of $x$ with $i$ primes.

## Feasibility at the Syntactic Level

$q_{in} \xrightarrow{\mathtt{op}_0} q_1 \cdots q_k \xrightarrow{\mathtt{op}_k} q_{bad}$ feasible iff $\langle\!\langle \mathtt{op}_0 \rangle\!\rangle^{(0)} \wedge \cdots \wedge \langle\!\langle \mathtt{op}_k \rangle\!\rangle^{(k)}$ sat.

# Checking Feasibility of Control Paths

## Feasibility at the Syntactic Level

$q_{in} \xrightarrow{\mathrm{op}_0} q_1 \cdots q_k \xrightarrow{\mathrm{op}_k} q_{bad}$ feasible iff $\langle\!\langle \mathrm{op}_0 \rangle\!\rangle^{(0)} \wedge \cdots \wedge \langle\!\langle \mathrm{op}_k \rangle\!\rangle^{(k)}$ sat.

Number of variables grows linearly with the length of the control path.

To help the prover, we may replace $\langle\!\langle \mathrm{op} \rangle\!\rangle$ with the weakest precondition

$$\mathrm{wp}(\mathrm{op}, \varphi) \quad = \quad \begin{cases} g \wedge \varphi & \text{if } \mathrm{op} = g \\ \varphi[e/x] & \text{if } \mathrm{op} = x := e \end{cases}$$

## Feasibility with Weakest Precondition

$q_{in} \xrightarrow{*} q_{bad}$ feasible iff $\mathrm{wp}(\mathrm{op}_0, \mathrm{wp}(\mathrm{op}_1, \ldots, \mathrm{wp}(\mathrm{op}_k, \mathit{true}) \cdots ))$ sat.

But it might actually be better to rely on the prover's powerful engine!

# Checking Feasibility of Control Paths

## Feasibility at the Syntactic Level

$q_{in} \xrightarrow{\mathrm{op}_0} q_1 \cdots q_k \xrightarrow{\mathrm{op}_k} q_{bad}$ feasible iff $\langle\!\langle \mathrm{op}_0 \rangle\!\rangle^{(0)} \wedge \cdots \wedge \langle\!\langle \mathrm{op}_k \rangle\!\rangle^{(k)}$ sat.

Number of variables grows linearly with the length of the control path.

To help the prover, we may replace $\langle\!\langle \mathrm{op} \rangle\!\rangle$ with the weakest precondition

$$\mathrm{wp}(\mathrm{op}, \varphi) = \begin{cases} g \wedge \varphi & \text{if } \mathrm{op} = g \\ \varphi[e/x] & \text{if } \mathrm{op} = x := e \end{cases}$$

## Feasibility with Weakest Precondition

$q_{in} \xrightarrow{*} q_{bad}$ feasible iff $\mathrm{wp}(\mathrm{op}_0, \mathrm{wp}(\mathrm{op}_1, \ldots, \mathrm{wp}(\mathrm{op}_k, \mathit{true}) \cdots ))$ sat.

But it might actually be better to rely on the prover's powerful engine!

# Refinement Challenge: Finding Relevant Predicates

Assume that the counterexample $q_{in} \xrightarrow{\mathtt{op}_0} q_1 \cdots q_k \xrightarrow{\mathtt{op}_k} q_{bad}$ is spurious

$$\langle\!\langle \mathtt{op}_0 \rangle\!\rangle^{(0)} \wedge \cdots \wedge \langle\!\langle \mathtt{op}_k \rangle\!\rangle^{(k)} \text{ unsatisfiable} \qquad (1)$$

Refinement consists in adding new predicates, but as few as possible.

## Goal

Find predicates that remove the counterexample from the abstraction

## Practical Approach

Some conjuncts in (1) do not "participate" in unsatisfiability.

Natural idea: try to find a small unsatisfiable subset of useful conjuncts.

For instance pick the leaves in a proof of unsatisfiability.

Might or might not work… … Let us look back at the goal!

# Refinement: Computation of Path Invariants

Consider an unfeasible control path $q_{in} \xrightarrow{\mathtt{op}_0} q_1 \cdots q_k \xrightarrow{\mathtt{op}_k} q_{bad}$.

## Path Safety Invariant

Sequence $(\phi_i)_{0 \leq i \leq k+1}$ of subsets of $\mathtt{X} \rightarrow \mathbb{R}$ such that

$$\phi_0 \;=\; \mathtt{X} \rightarrow \mathbb{R} \qquad\qquad \phi_{i+1} \;\supseteq\; [\![\mathtt{op}_i]\!][\phi_i] \qquad\qquad \phi_{k+1} \;=\; \emptyset$$

## Intuition

A path safety invariant gives an explanation of unfeasibility

## Example: Sequence of Reachable Valuations Along the Path

$$\phi_i \;=\; [\![\mathtt{op}_{i-1}]\!] \circ \cdots \circ [\![\mathtt{op}_0]\!][\mathtt{X} \rightarrow \mathbb{R}]$$

## Objective

Compute simple path safety invariants

# Refinement: Path Safety Invariants from Proofs

Consider an unfeasible control path $q_{in} \xrightarrow{\mathtt{op}_0} q_1 \cdots q_k \xrightarrow{\mathtt{op}_k} q_{bad}$.

## Path Safety Invariant (Syntactic Definition)

Sequence $(\varphi_i)_{0 \leq i \leq k+1}$ of formulas with free variables in $\mathtt{X}$ such that

$$\varphi_0 = \mathit{true} \qquad \models \varphi_i \wedge \langle\langle \mathtt{op}_i \rangle\rangle \Rightarrow \varphi_{i+1}^{(1)} \qquad \varphi_{k+1} = \mathit{false}$$

Path safety invariants can be obtained from proofs of unsatisfiability

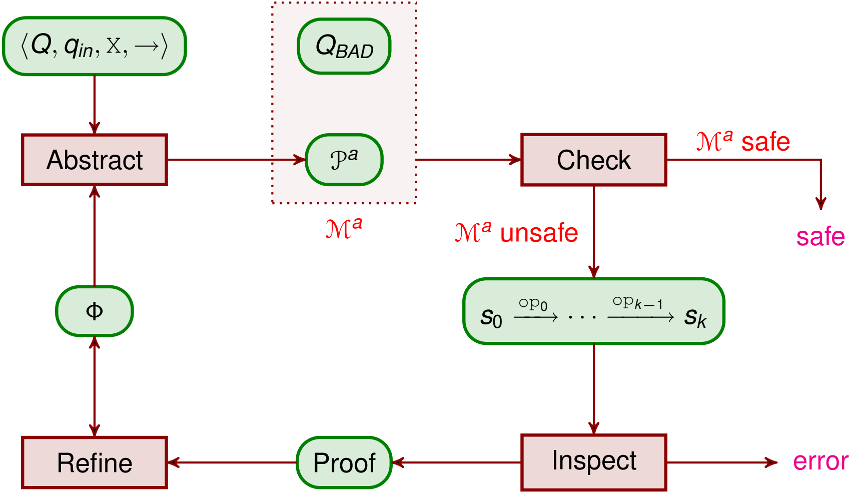
## Refinement

New predicates are atomic predicates from the path invariant.

This guarantees that the counterexample will be eliminated.

No quantifier ever introduced! ☺

# CounterExample-Guided Abstract model Refinement



$\langle Q, q_{in}, \mathrm{X}, \rightarrow \rangle$
$Q_{BAD}$
Abstract
$\mathcal{P}^a$
Check
$\mathcal{M}^a$ safe
safe
$\mathcal{M}^a$
$\mathcal{M}^a$ unsafe
$\Phi$
$s_0 \xrightarrow{\mathrm{op}_0} \cdots \xrightarrow{\mathrm{op}_{k-1}} s_k$
Refine
Proof
Inspect
error

# Classical CEGAR Algorithm

```
CEGAR $(\mathcal{P} = \langle Q, q_{in}, \mathrm{X}, \rightarrow \rangle, Q_{BAD}, \Phi_0)$
    $\Phi \leftarrow \Phi_0$
    while (true)
        $\mathcal{M}^a \leftarrow$ PredicateAbstraction $(\langle Q, q_{in}, \mathrm{X}, \rightarrow \rangle, Q_{BAD}, \Phi)$
        check $\leftarrow$ ModelCheck $(\mathcal{M}^a)$
        if check is $\mathcal{M}^a$ safe
            return $\mathcal{P}$ safe
        // check is $(q_{in}, v_0^a) \xrightarrow{\mathtt{op}_0} (q_1, v_1^a) \cdots (q_k, v_k^a) \xrightarrow{\mathtt{op}_k} (q_{bad}, v_{k+1}^a)$
        insp $\leftarrow$ Inspect $(q_{in} \xrightarrow{\mathtt{op}_0} q_1 \cdots q_k \xrightarrow{\mathtt{op}_k} q_{bad}))$
        if insp is feasible
            return $q_{in} \xrightarrow{\mathtt{op}_0} q_1 \cdots q_k \xrightarrow{\mathtt{op}_k} q_{bad}$ feasible
        // insp is unfeasible
        construct a path invariant and extract new predicates $\Phi'$ from it
        $\Phi \leftarrow \Phi \cup \Phi'$
```

# Drawbacks of the Classical CEGAR Algorithm

Batch-oriented integration

No sharing of data structures

No reuse of previous computations

Re-explores the same error-free parts of the configuration space

Abstraction fully computed before the model-checking phase

Useless expensive work

# Some Variants of the Classical CEGAR Algorithm

## Lazy CEGAR

Integrated CEGAR loop driven by the model-checker

Builds a reachability tree with abstract successors on demand

- Nodes labeled by support predicates
- Refinement only locally refines subparts of the tree

## Lazy Interpolation

Builds a reachability tree with no abstract successor computation

Uses interpolation to:

- rule out each spurious control path
- label counterexample paths in the tree with path invariants

Part VII

# Conclusions

# Outline — Conclusions

19 Summary

20 Applications of CEGAR to Software Verification

21 Concluding Remarks

22 Some References

# Outline — Conclusions

19 Summary

20 Applications of CEGAR to Software Verification

21 Concluding Remarks

22 Some References

# Summary: Abstract Model Refinement

Fully automatic software verification technique based on model-checking and refinement of finite-state abstractions

## Property-Preserving Abstraction

Conservatively extract finite-state models from programs

## Inspection of Abstract Counterexamples

Reduces to satisfiability checking

## Refinement Guided by Abstract Counterexamples

Based on the construction of path invariants

New predicates obtained from proofs of unsatisfiability

Each of these three phases relies on theorem provers

# Outline — Conclusions

19 Summary

20 Applications of CEGAR to Software Verification

21 Concluding Remarks

22 Some References

# Some CEGAR-based Software Verification Tools

## SLAM — Thomas Ball, Sriram Rajamani, . . .

Analysis of programs written in C

- ☹ Classical batch-oriented CEGAR algorithm
- ☺ Interprocedural analysis (abstraction into boolean programs)

Now integrated in Static Driver Verifier, part of the Windows Driver Kit

## BLAST — Thomas Henzinger, . . .

Analysis of programs written in C

- ☺ Lazy CEGAR algorithm
- ☹ Bounded-recursion interprocedural analysis

Open source, distributed under the BSD license

MAGIC, YASM, . . .

# Application: Verification of Device Drivers

Why device drivers?

## High Impact

Bugs lead to system crash (e.g. BSOD)

Developed by third-party vendors

## Not So Complex

Simple safety properties (e.g. locking discipline)

Only a small part of the code is relevant to the properties

Medium-sized ($\leq$ 25 000 lines)

# Outline — Conclusions

19 Summary

20 Applications of CEGAR to Software Verification

21 Concluding Remarks

22 Some References

# Static Analysis and Abstraction Refinement

Verification of software: computation of strong enough invariants

## Abstraction Process

Interpret programs according to a simplified, “abstract” semantics.

## Property-Preserving Abstraction

Formally relate the “abstract” semantics with the “standard” semantics, so as to preserve relevant properties.

Main challenge: suitable refinement of abstractions

# Static Analysis versus Abstraction Refinement

### Static Analysis

- ☺ Always terminates
- ☹ False positives
- ☹ Manual refinement
- ☺ Infinite domains
- ☹ Same precision everywhere

### Abstraction Refinement

- ☹ May not terminate
- ☺ Definite answer (yes / no)
- ☺ Automatic refinement
- ☹ Finite abstract domains
- ☺ Adaptive precision
- ☺ Driven by the property

### Inspection & Refinement

Smart mind

### Inspection & Refinement

Smart prover

# Static Analysis versus Abstraction Refinement

## Static Analysis

- ☺ Always terminates
- ☹ False positives
- ☹ Manual refinement
- ☺ Infinite domains
- ☹ Same precision everywhere

## Inspection & Refinement

Smart mind

## Abstraction Refinement

- ☹ May not terminate
- ☺ Definite answer (yes / no)
- ☺ Automatic refinement
- ☹ Finite abstract domains
- ☺ Adaptive precision
- ☺ Driven by the property

## Inspection & Refinement

Smart prover

# Extensions and Remaining Challenges

Not Covered in the Lecture

### Computational Models

- Pointer analysis, arrays
- Recursion, threads
- Hybrid systems, . . .

### Beyond Safety

- Termination
- Liveness properties
- $\mu$-calculus (Modal LKS)

Software Verification remains a challenging problem!

### Room for Improvement

- Generation of smart predicates for refinement
- Path invariants for control paths with loops

# Outline — Conclusions

19 Summary

20 Applications of CEGAR to Software Verification

21 Concluding Remarks

22 Some References

# Some References

- S. Graf and H. Saïdi.
  Construction of abstract state graphs with PVS.
  In *Proc. 9th Int. Conf. Computer Aided Verification, Haifa, Israel*, LNCS 1254, pages 72–83. Springer, 1997.

- E. Clarke, O. Grumberg, S. Jha, Y. Lu, H. Veith.
  Counterexample-guided abstraction refinement.
  In *Proc. 12th Int. Conf. Computer Aided Verification, Haifa, Israel*, LNCS 1855, pages 154–169. Springer, 2000.

- The SLAM Project.
  `http://research.microsoft.com/slam/`

- The Berkeley Lazy Abstraction Software Verification Tool.
  `http://mtc.epfl.ch/software-tools/blast/`

Thank you!