Acceleration of Abstract Interpretation: Widening and Nowrrowing

Vladimír Štill

Parallel & Distributed Systems Laboratory

Masaryk University

Brno, Czech Republic

9th October 2017

Refresh I

Partially Ordered Set (poset)

- $(P, \sqsubseteq)$, $P \neq \emptyset$
- $\sqsubseteq$ is a binary relation which is reflexive, anti-symmetric, and transitive

Refresh I

Partially Ordered Set (poset)

- $(P, \sqsubseteq)$, $P \neq \emptyset$
- $\sqsubseteq$ is a binary relation which is reflexive, anti-symmetric, and transitive

Lattice

- let $(P, \sqsubseteq)$ be a poset
- if $\inf(x, y)$ and $\sup(x, y)$ exists for all $x, y \in P$, then $(P, \sqsubseteq)$ is a *lattice*
- if $\inf(X)$, $\sup(X)$ exists for all $X \subseteq P$, then $(P, \sqsubseteq)$ is a *complete lattice*

Refresh II

Galois Connection

- let $(C, \leqslant)$ and $(A, \sqsubseteq)$ be complete lattices
- functions $\alpha\colon C \to A, \gamma\colon A \to C$ such that
  $\forall c \in C, \forall a \in A : \alpha(c) \sqsubseteq a \Leftrightarrow c \leqslant \gamma(a)$
- also $A \underset{\gamma}{\overset{\alpha}{\leftrightarrows}} C$

Abstracting an Iterative Computation I

- complete lattice $(C, \leqslant)$
- suppose we have a monotone function $f\colon C \to C$

Abstracting an Iterative Computation I

- complete lattice $(C, \leqslant)$
- suppose we have a monotone function $f\colon C \to C$
- we want to calculate (overapproximation of) smallest fixpoint of f

$$\mathsf{lfp}(f) = \mathsf{inf}\{c \in C \mid c = f(c)\}$$

Abstracting an Iterative Computation I

- complete lattice $(C, \leqslant)$
- suppose we have a monotone function $f \colon C \to C$
- we want to calculate (overapproximation of) smallest fixpoint of f

$$\mathsf{lfp}(f) = \inf\{c \in C \mid c = f(c)\}$$
$$\mathsf{lfp}(f) \leqslant \gamma(\mathsf{lfp}(f^{\#}))$$

  - where $f^{\#} \colon A \to A$ approximates f

Abstracting an Iterative Computation I

- complete lattice $(C, \leqslant)$
- suppose we have a monotone function $f \colon C \to C$
- we want to calculate (overapproximation of) smallest fixpoint of f

$$\mathsf{lfp}(f) = \inf\{c \in C \mid c = f(c)\}$$
$$\mathsf{lfp}(f) \leqslant \gamma(\mathsf{lfp}(f^{\#})) = \inf\{\gamma(a) \mid a \in A, a = f^{\#}(a)\}$$

  - where $f^{\#} \colon A \to A$ approximates f

Abstracting an Iterative Computation I

- complete lattice $(C, \leqslant)$
- suppose we have a monotone function $f\colon C \to C$
- we want to calculate (overapproximation of) smallest fixpoint of f

$$\mathsf{lfp}(f) = \inf\{c \in C \mid c = f(c)\}$$
$$\mathsf{lfp}(f) \leqslant \gamma(\mathsf{lfp}(f^\#)) = \inf\{\gamma(a) \mid a \in A, a = f^\#(a)\}$$

  - where $f^\#\colon A \to A$ approximates f
- if A is finite or has no infinite strictly ascending chains:

$$\mathsf{lfp}(f^\#) = (f^\#)^n(\bot_A)$$

  - for some $n \in \mathbb{N}$, $\bot_A = \inf(A)$
  - can be calculated iteratively

Abstracting an Iterative Computation II

important abstract domains have strictly ascending chains:

- intervals: $[0,0] \sqsubseteq [0,1] \sqsubseteq [0,2] \sqsubseteq \cdots$

Abstracting an Iterative Computation II

important abstract domains have strictly ascending chains:

- intervals: $[0,0] \sqsubseteq [0,1] \sqsubseteq [0,2] \sqsubseteq \cdots$

but infinite abstract domains are useful

- we can trade precision for tractability

Abstracting an Iterative Computation II

important abstract domains have strictly ascending chains:

- intervals: $[0,0] \sqsubseteq [0,1] \sqsubseteq [0,2] \sqsubseteq \cdots$

but infinite abstract domains are useful

- we can trade precision for tractability
- accelerate iterative computation

Abstracting an Iterative Computation II

important abstract domains have strictly ascending chains:

- intervals: $[0,0] \sqsubseteq [0,1] \sqsubseteq [0,2] \sqsubseteq \cdots$

but infinite abstract domains are useful

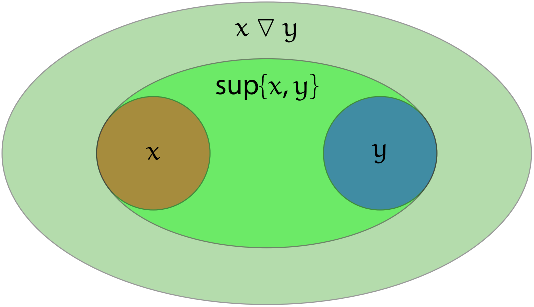
- we can trade precision for tractability
- accelerate iterative computation
- possibly find a element $a \sqsupseteq \mathsf{lfp}(f^{\#})$

Widening I

- let $(A, \sqsubseteq)$ be a poset
- (pair) widening operator: $\nabla : A \times A \to A$

Widening I

- let $(A, \sqsubseteq)$ be a poset
- (pair) widening operator: $\nabla \colon A \times A \to A$
  - *covering*: $\forall x, y \in A.\, x \sqsubseteq x \nabla y$ and $y \sqsubseteq x \nabla y$

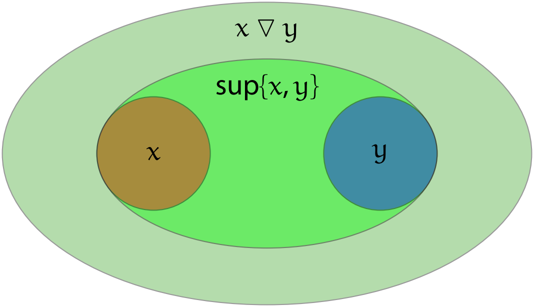


Widening I

- let $(A, \sqsubseteq)$ be a poset
- (pair) widening operator: $\nabla \colon A \times A \to A$
  - *covering*: $\forall x, y \in A.\, x \sqsubseteq x \mathbin{\nabla} y$ and $y \sqsubseteq x \mathbin{\nabla} y$
  - *termination*: for every ascending chain $\{x_i\}_{i \geqslant 0}$ the ascending chain

$$y_0 = x_0 \qquad\qquad y_{i+1} = y_i \mathbin{\nabla} x_{i+1}$$

    stabilizes after a finite number of terms

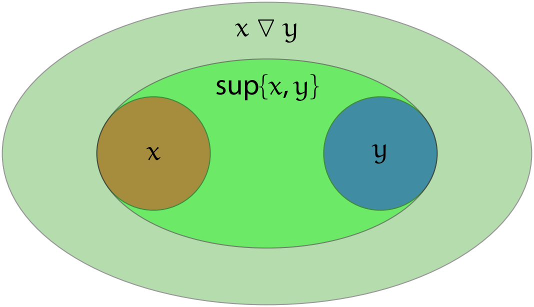


Widening I

- let $(A, \sqsubseteq)$ be a poset
- (pair) widening operator: $\nabla \colon A \times A \to A$
  - *covering*: $\forall x, y \in A.\, x \sqsubseteq x \nabla y$ and $y \sqsubseteq x \nabla y$
  - *termination*: for every ascending chain $\{x_i\}_{i \geqslant 0}$ the ascending chain

    $$y_0 = x_0 \qquad\qquad y_{i+1} = y_i \nabla x_{i+1}$$

    stabilizes after a finite number of terms
  - *note*: ∇ is often not symmetric



Widening II

- iterative calculation of $\widehat{x} \sqsupseteq \mathsf{lfp}(f^{\#})$:

$$\begin{aligned}\widehat{x}_0 &= \bot \\ \widehat{x}_{i+1} &= \widehat{x}_i \mathbin{\nabla} f^{\#}(\widehat{x}_i)\end{aligned}$$

Widening II

- iterative calculation of $\widehat{x} \sqsupseteq \mathrm{lfp}(f^{\#})$:

$$\widehat{x}_0 = \bot$$
$$\widehat{x}_{i+1} = \widehat{x}_i \triangledown f^{\#}(\widehat{x}_i)$$

- example of $\triangledown$ (intervals):

$$\bot \triangledown x = x$$
$$x \triangledown \bot = x$$
$$[l_0, u_0] \triangledown [l_1, u_1] = [\mathrm{ite}(l_1 < l_0, -\infty, l_0), \mathrm{ite}(u_0 < u_1, +\infty, u_0)]$$

Widening II

- iterative calculation of $\widehat{x} \sqsupseteq \mathsf{lfp}(f^{\#})$:

$$\widehat{x}_0 = \bot$$
$$\widehat{x}_{i+1} = \widehat{x}_i \mathbin{\triangledown} f^{\#}(\widehat{x}_i)$$

- example of $\triangledown$ (intervals):

$$\bot \mathbin{\triangledown} x = x$$
$$x \mathbin{\triangledown} \bot = x$$
$$[l_0, u_0] \mathbin{\triangledown} [l_1, u_1] = [\mathsf{ite}(l_1 < l_0, -\infty, l_0), \mathsf{ite}(u_0 < u_1, +\infty, u_0)]$$

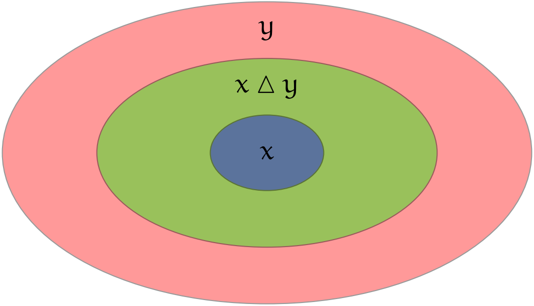
  - if the bound is expanding, extrapolate to infinity

Narrowing I

- let $(A, \sqsubseteq)$ be a poset
- (pair) narrowing operator: $\triangle : A \times A \to A$

Narrowing I

- let $(A, \sqsubseteq)$ be a poset
- (pair) narrowing operator: $\triangle : A \times A \to A$
  - *bounding*: $\forall x, y \in A.\, x \sqsubseteq y \implies x \sqsubseteq (x \triangle y) \sqsubseteq y$

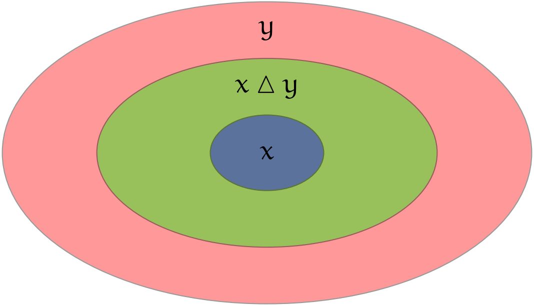


Narrowing I

- let $(A, \sqsubseteq)$ be a poset
- (pair) narrowing operator: $\triangle \colon A \times A \to A$
  - *bounding*: $\forall x, y \in A.\, x \sqsubseteq y \implies x \sqsubseteq (x \triangle y) \sqsubseteq y$
  - *termination*: for every descending chain $\{x_i\}_{i \geqslant 0}$ the chain

    $$y_0 = x_0 \qquad\qquad y_{i+1} = y_i \triangle x_{i+1}$$

    stabilizes after a finite number of terms

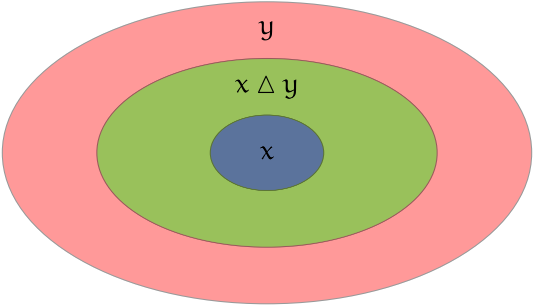


Narrowing I

- let $(A, \sqsubseteq)$ be a poset
- (pair) narrowing operator: $\triangle \colon A \times A \to A$
  - *bounding*: $\forall x, y \in A.\, x \sqsubseteq y \implies x \sqsubseteq (x \triangle y) \sqsubseteq y$
  - *termination*: for every descending chain $\{x_i\}_{i \geqslant 0}$ the chain

    $$y_0 = x_0 \qquad\qquad y_{i+1} = y_i \triangle x_{i+1}$$

    stabilizes after a finite number of terms
  - *note*: $\triangle$ is often not symmetric



Narrowing II

- iterative improving of precision of $\widehat{x} \sqsupseteq \mathsf{lfp}(f^\#)$:

$$\begin{aligned}\widehat{\widehat{x}}_0 &= \widehat{x}\\ \widehat{\widehat{x}}_{i+1} &= \widehat{\widehat{x}}_i \mathbin{\triangle} f^\#(\widehat{\widehat{x}}_i)\end{aligned}$$

Narrowing II

- iterative improving of precision of $\widehat{x} \sqsupseteq \text{lfp}(f^\#)$:

$$\begin{aligned}\widehat{\widehat{x}}_0 &= \widehat{x}\\ \widehat{\widehat{x}}_{i+1} &= \widehat{\widehat{x}}_i \mathbin{\triangle} f^\#(\widehat{\widehat{x}}_i)\end{aligned}$$

- obtains $\widehat{\widehat{x}}$ such that: $\text{lfp}(f^\#) \sqsubseteq \widehat{\widehat{x}} \sqsubseteq \widehat{x}$

Narrowing II

- iterative improving of precision of $\widehat{x} \sqsupseteq \mathsf{lfp}(f^\#)$:

$$\widehat{\widehat{x}}_0 = \widehat{x}$$
$$\widehat{\widehat{x}}_{i+1} = \widehat{\widehat{x}}_i \vartriangle f^\#(\widehat{\widehat{x}}_i)$$

- obtains $\widehat{\widehat{x}}$ such that: $\mathsf{lfp}(f^\#) \sqsubseteq \widehat{\widehat{x}} \sqsubseteq \widehat{x}$
- example of $\vartriangle$ (intervals):

$$\bot \vartriangle x = \bot$$
$$x \vartriangle \bot = \bot$$
$$[l_0, u_0] \vartriangle [l_1, u_1] = [\mathsf{ite}(l_0 = -\infty, l_1, l_0), \mathsf{ite}(u_0 = +\infty, u_1, u_0)]$$

Narrowing II

- iterative improving of precision of $\widehat{x} \sqsupseteq \mathsf{lfp}(f^\#)$:

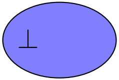
$$\widehat{\widehat{x}}_0 = \widehat{x}$$
$$\widehat{\widehat{x}}_{i+1} = \widehat{\widehat{x}}_i \bigtriangleup f^\#(\widehat{\widehat{x}}_i)$$

- obtains $\widehat{\widehat{x}}$ such that: $\mathsf{lfp}(f^\#) \sqsubseteq \widehat{\widehat{x}} \sqsubseteq \widehat{x}$
- example of $\bigtriangleup$ (intervals):

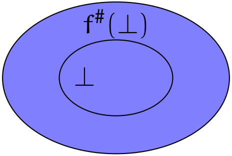
$$\bot \bigtriangleup x = \bot$$
$$x \bigtriangleup \bot = \bot$$
$$[l_0, u_0] \bigtriangleup [l_1, u_1] = [\mathsf{ite}(l_0 = -\infty, l_1, l_0), \mathsf{ite}(u_0 = +\infty, u_1, u_0)]$$

  - prefer finite bounds

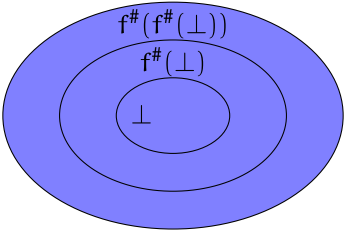
Widening and Narrowing in a Diagram



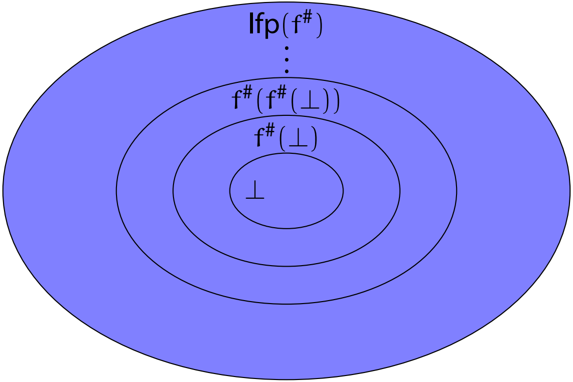
Widening and Narrowing in a Diagram



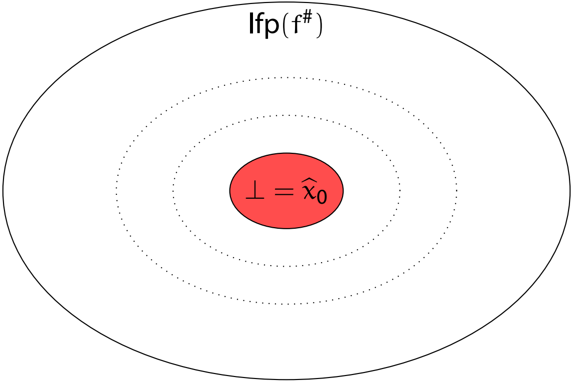
Widening and Narrowing in a Diagram



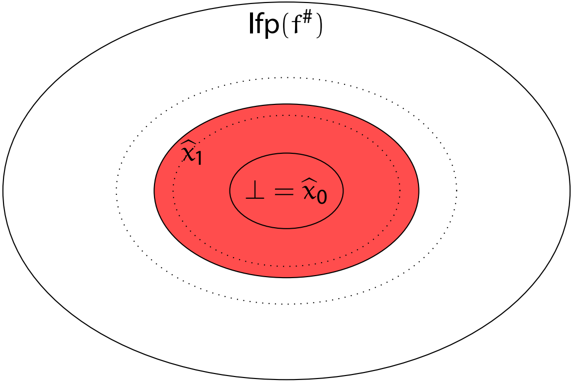
Widening and Narrowing in a Diagram



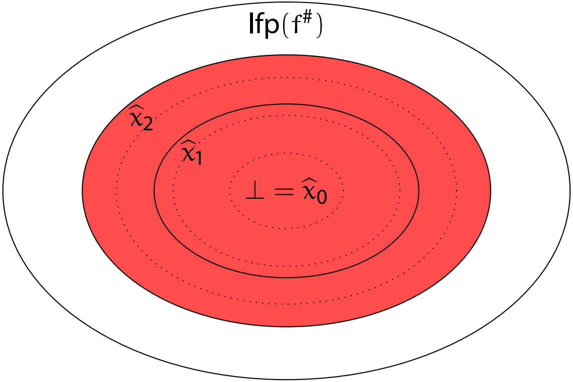
Widening and Narrowing in a Diagram



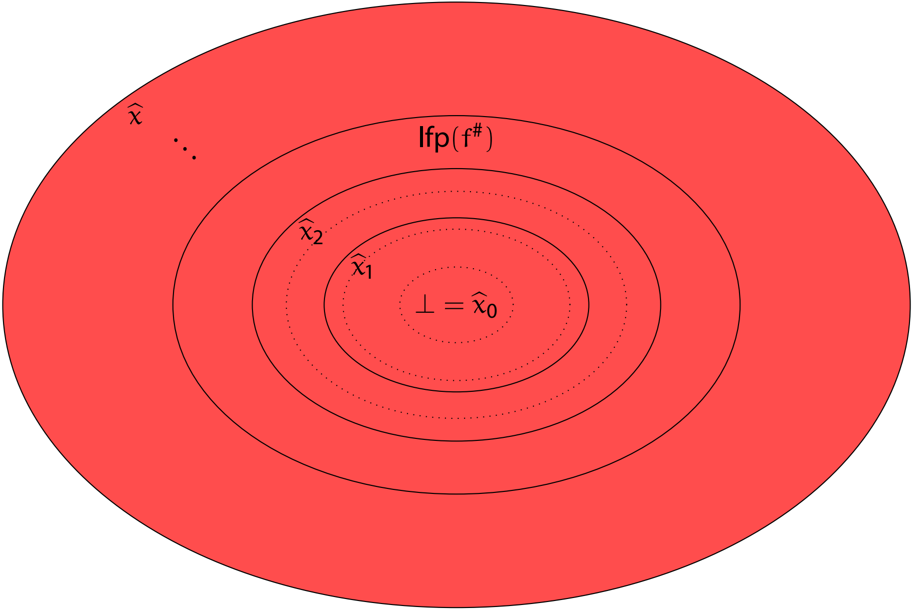
Widening and Narrowing in a Diagram



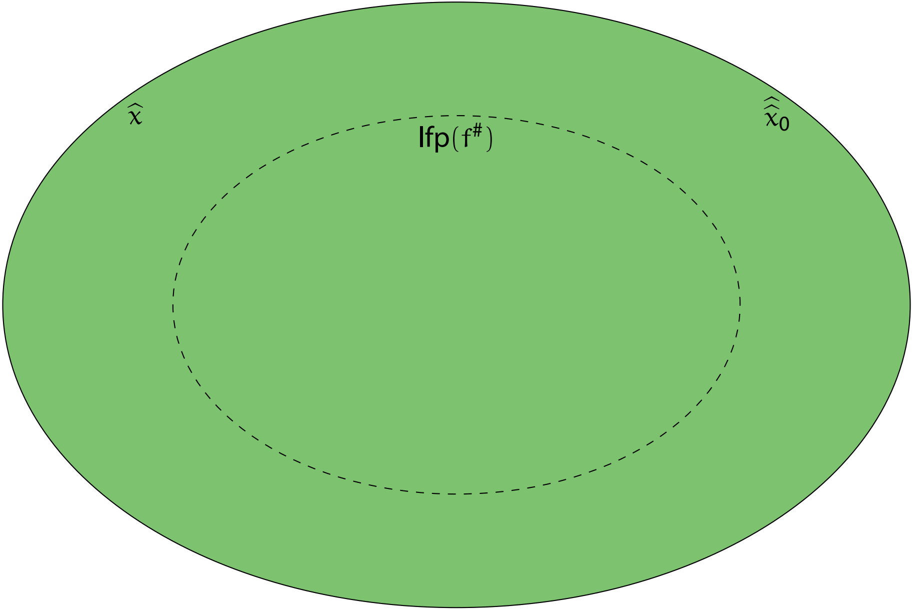
Widening and Narrowing in a Diagram



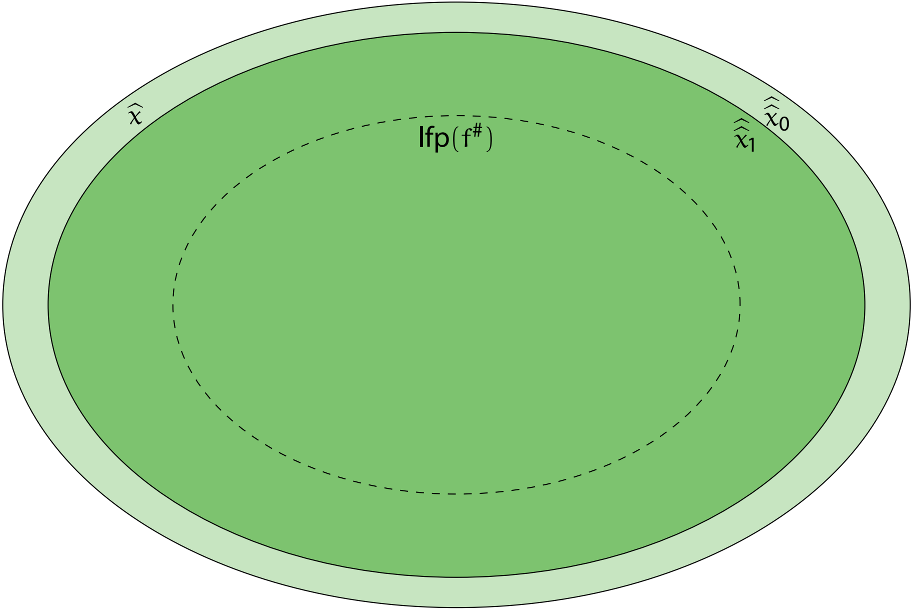
Widening and Narrowing in a Diagram



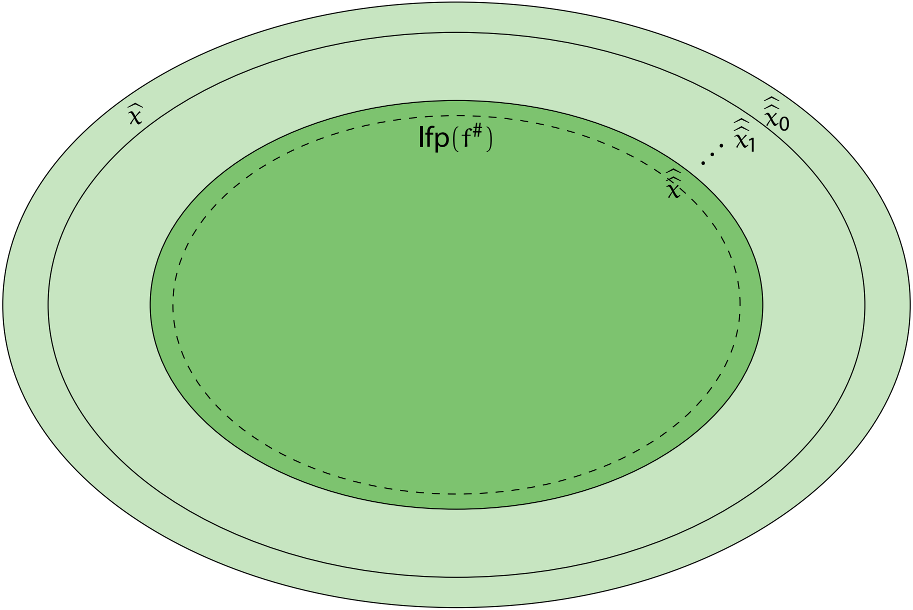
Widening and Narrowing in a Diagram



Widening and Narrowing in a Diagram



Widening and Narrowing in a Diagram



Widening and Narrowing Example

```
def foo():
    i = 1
    while i <= 100:
        i = i + 1
```

- possible values of `i` at the beginning of the cycle?

Widening and Narrowing Example

```python
def foo():
    i = 1
    while i <= 100:
        i = i + 1
```

- possible values of `i` at the beginning of the cycle?
- least fixed point of `f`:

$$X = f(X) = (\{1\} \cup \{i + 1 \mid i \in X\}) \cap \{i \in \mathbb{Z} \mid i \leqslant 100\}$$

Widening and Narrowing Example

```
def foo():
    i = 1
    while i <= 100:
        i = i + 1
```

- possible values of `i` at the beginning of the cycle?
- least fixed point of `f`:

$$X = f(X) = (\{1\} \cup \{i+1 \mid i \in X\}) \cap \{i \in \mathbb{Z} \mid i \leqslant 100\}$$

- with widening, on intervals:

$$X = f^{\#}(X) = ([1,1] \sqcup (X \oplus [1,1])) \sqcap [-\infty, 100]$$

Widening and Narrowing Example

```
def foo():
    i = 1
    while i <= 100:
        i = i + 1
```

- possible values of `i` at the beginning of the cycle?
- least fixed point of `f`:

$$X = f(X) = (\{1\} \cup \{i+1 \mid i \in X\}) \cap \{i \in \mathbb{Z} \mid i \leqslant 100\}$$

- with widening, on intervals:

$$X = f^{\#}(X) = ([1,1] \sqcup (X \oplus [1,1])) \sqcap [-\infty, 100]$$

$$\widehat{X}_0 = \bot$$

Widening and Narrowing Example

```
def foo():
    i = 1
    while i <= 100:
        i = i + 1
```

- possible values of `i` at the beginning of the cycle?
- least fixed point of `f`:

$$X = f(X) = (\{1\} \cup \{i+1 \mid i \in X\}) \cap \{i \in \mathbb{Z} \mid i \leqslant 100\}$$

- with widening, on intervals:

$$X = f^{\#}(X) = ([1,1] \sqcup (X \oplus [1,1])) \sqcap [-\infty, 100]$$

$$\widehat{X}_0 = \bot$$

$$\widehat{X}_1 = \widehat{X}_0 \triangledown (([1,1] \sqcup (\widehat{X}_0 \oplus [1,1])) \sqcap [-\infty, 100])$$

Widening and Narrowing Example

```
def foo():
    i = 1
    while i <= 100:
        i = i + 1
```

- possible values of `i` at the beginning of the cycle?
- least fixed point of `f`:

$$X = f(X) = (\{1\} \cup \{i+1 \mid i \in X\}) \cap \{i \in \mathbb{Z} \mid i \leqslant 100\}$$

- with widening, on intervals:

$$X = f^{\#}(X) = ([1,1] \sqcup (X \oplus [1,1])) \sqcap [-\infty, 100]$$

$$\widehat{X}_0 = \bot$$

$$\widehat{X}_1 = \bot \mathbin{\triangledown} (([1,1] \sqcup (\bot \oplus [1,1])) \sqcap [-\infty, 100])$$

Widening and Narrowing Example

```
def foo():
    i = 1
    while i <= 100:
        i = i + 1
```

- possible values of `i` at the beginning of the cycle?
- least fixed point of `f`:

$$X = f(X) = (\{1\} \cup \{i+1 \mid i \in X\}) \cap \{i \in \mathbb{Z} \mid i \leqslant 100\}$$

- with widening, on intervals:

$$X = f^{\#}(X) = ([1,1] \sqcup (X \oplus [1,1])) \sqcap [-\infty, 100]$$

$$\widehat{X}_0 = \bot$$

$$\widehat{X}_1 = \bot \triangledown (([1,1] \sqcup \bot) \sqcap [-\infty, 100])$$

Widening and Narrowing Example

```python
def foo():
    i = 1
    while i <= 100:
        i = i + 1
```

- possible values of `i` at the beginning of the cycle?
- least fixed point of `f`:

$$X = f(X) = (\{1\} \cup \{i+1 \mid i \in X\}) \cap \{i \in \mathbb{Z} \mid i \leqslant 100\}$$

- with widening, on intervals:

$$X = f^{\#}(X) = ([1,1] \sqcup (X \oplus [1,1])) \sqcap [-\infty, 100]$$

$$\widehat{X}_0 = \bot$$

$$\widehat{X}_1 = \bot \triangledown ([1,1] \sqcap [-\infty, 100])$$

Widening and Narrowing Example

```
def foo():
    i = 1
    while i <= 100:
        i = i + 1
```

- possible values of `i` at the beginning of the cycle?
- least fixed point of `f`:

$$X = f(X) = (\{1\} \cup \{i+1 \mid i \in X\}) \cap \{i \in \mathbb{Z} \mid i \leqslant 100\}$$

- with widening, on intervals:

$$X = f^{\#}(X) = ([1,1] \sqcup (X \oplus [1,1])) \sqcap [-\infty, 100]$$

$$\widehat{X}_0 = \bot$$

$$\widehat{X}_1 = \bot \mathbin{\triangledown} [1,1]$$

Widening and Narrowing Example

```
def foo():
    i = 1
    while i <= 100:
        i = i + 1
```

- possible values of `i` at the beginning of the cycle?
- least fixed point of `f`:

$$X = f(X) = (\{1\} \cup \{i+1 \mid i \in X\}) \cap \{i \in \mathbb{Z} \mid i \leqslant 100\}$$

- with widening, on intervals:

$$X = f^{\#}(X) = ([1,1] \sqcup (X \oplus [1,1])) \sqcap [-\infty, 100]$$

$$\widehat{X}_0 = \bot$$

$$\widehat{X}_1 = [1,1]$$

Widening and Narrowing Example

```
def foo():
    i = 1
    while i <= 100:
        i = i + 1
```

- possible values of `i` at the beginning of the cycle?
- least fixed point of `f`:

$$X = f(X) = (\{1\} \cup \{i+1 \mid i \in X\}) \cap \{i \in \mathbb{Z} \mid i \leqslant 100\}$$

- with widening, on intervals:

$$X = f^{\#}(X) = ([1,1] \sqcup (X \oplus [1,1])) \sqcap [-\infty, 100]$$

$$\widehat{X}_0 = \bot$$

$$\widehat{X}_1 = [1,1]$$

$$\widehat{X}_2 = \widehat{X}_1 \triangledown (([1,1] \sqcup (\widehat{X}_1 \oplus [1,1])) \sqcap [-\infty, 100])$$

Widening and Narrowing Example

```python
def foo():
    i = 1
    while i <= 100:
        i = i + 1
```

- possible values of `i` at the beginning of the cycle?
- least fixed point of `f`:

$$X = f(X) = (\{1\} \cup \{i+1 \mid i \in X\}) \cap \{i \in \mathbb{Z} \mid i \leqslant 100\}$$

- with widening, on intervals:

$$X = f^{\#}(X) = ([1,1] \sqcup (X \oplus [1,1])) \sqcap [-\infty, 100]$$

$$\widehat{X}_0 = \bot$$

$$\widehat{X}_1 = [1,1]$$

$$\widehat{X}_2 = [1,1] \mathbin{\triangledown} (([1,1] \sqcup ([1,1] \oplus [1,1])) \sqcap [-\infty, 100])$$

Widening and Narrowing Example

```
def foo():
    i = 1
    while i <= 100:
        i = i + 1
```

- possible values of `i` at the beginning of the cycle?
- least fixed point of `f`:

$$X = f(X) = (\{1\} \cup \{i + 1 \mid i \in X\}) \cap \{i \in \mathbb{Z} \mid i \leqslant 100\}$$

- with widening, on intervals:

$$X = f^{\#}(X) = ([1,1] \sqcup (X \oplus [1,1])) \sqcap [-\infty, 100]$$

$$\widehat{X}_0 = \bot$$

$$\widehat{X}_1 = [1,1]$$

$$\widehat{X}_2 = [1,1] \mathbin{\triangledown} (([1,1] \sqcup [2,2]) \sqcap [-\infty, 100])$$

Widening and Narrowing Example

```python
def foo():
    i = 1
    while i <= 100:
        i = i + 1
```

- possible values of `i` at the beginning of the cycle?
- least fixed point of `f`:

$$X = f(X) = (\{1\} \cup \{i+1 \mid i \in X\}) \cap \{i \in \mathbb{Z} \mid i \leqslant 100\}$$

- with widening, on intervals:

$$X = f^{\#}(X) = ([1,1] \sqcup (X \oplus [1,1])) \sqcap [-\infty, 100]$$

$$\widehat{X}_0 = \bot$$

$$\widehat{X}_1 = [1,1]$$

$$\widehat{X}_2 = [1,1] \mathbin{\nabla} ([1,2] \sqcap [-\infty, 100])$$

Widening and Narrowing Example

```python
def foo():
    i = 1
    while i <= 100:
        i = i + 1
```

- possible values of `i` at the beginning of the cycle?
- least fixed point of `f`:

$$X = f(X) = (\{1\} \cup \{i + 1 \mid i \in X\}) \cap \{i \in \mathbb{Z} \mid i \leqslant 100\}$$

- with widening, on intervals:

$$X = f^{\#}(X) = ([1,1] \sqcup (X \oplus [1,1])) \sqcap [-\infty, 100]$$

$$\widehat{X}_0 = \bot$$

$$\widehat{X}_1 = [1,1]$$

$$\widehat{X}_2 = [1,1] \mathbin{\triangledown} [1,2]$$

Widening and Narrowing Example

```python
def foo():
    i = 1
    while i <= 100:
        i = i + 1
```

- possible values of `i` at the beginning of the cycle?
- least fixed point of `f`:

$$X = f(X) = (\{1\} \cup \{i + 1 \mid i \in X\}) \cap \{i \in \mathbb{Z} \mid i \leqslant 100\}$$

- with widening, on intervals:

$$X = f^{\#}(X) = ([1,1] \sqcup (X \oplus [1,1])) \sqcap [-\infty, 100]$$

$$\widehat{X}_0 = \bot$$

$$\widehat{X}_1 = [1,1]$$

$$\widehat{X}_2 = [1,1] \mathbin{\triangledown} [1,2]$$

Widening and Narrowing Example

```
def foo():
    i = 1
    while i <= 100:
        i = i + 1
```

- possible values of `i` at the beginning of the cycle?
- least fixed point of `f`:

$$X = f(X) = (\{1\} \cup \{i+1 \mid i \in X\}) \cap \{i \in \mathbb{Z} \mid i \leqslant 100\}$$

- with widening, on intervals:

$$X = f^{\#}(X) = ([1,1] \sqcup (X \oplus [1,1])) \sqcap [-\infty, 100]$$

$$\widehat{X}_0 = \bot$$

$$\widehat{X}_1 = [1,1]$$

$$\widehat{X}_2 = [1,+\infty]$$

Widening and Narrowing Example

```
def foo():
    i = 1
    while i <= 100:
        i = i + 1
```

- possible values of `i` at the beginning of the cycle?
- least fixed point of `f`:

$$X = f(X) = (\{1\} \cup \{i + 1 \mid i \in X\}) \cap \{i \in \mathbb{Z} \mid i \leqslant 100\}$$

- with widening, on intervals:

$$X = f^{\#}(X) = ([1,1] \sqcup (X \oplus [1,1])) \sqcap [-\infty, 100]$$

$$\widehat{X}_0 = \bot$$

$$\widehat{X}_1 = [1,1]$$

$$\widehat{X}_2 = [1,+\infty]$$

$$\widehat{X}_3 = \widehat{X}_2 = [1,+\infty]$$

Widening and Narrowing Example

```python
def foo():
    i = 1
    while i <= 100:
        i = i + 1
```

- improve precision with narrowing:

$$\widehat{\widehat{X}}_0 = \widehat{X} = [1, +\infty]$$

Widening and Narrowing Example

```python
def foo():
    i = 1
    while i <= 100:
        i = i + 1
```

- improve precision with narrowing:

$$\widehat{\widehat{X}}_0 = \widehat{X} = [1, +\infty]$$

$$\widehat{\widehat{X}}_1 = \widehat{\widehat{X}}_0 \vartriangle (([1,1] \sqcup (\widehat{\widehat{X}}_0 \oplus [1,1])) \sqcap [-\infty, 100])$$

Widening and Narrowing Example

```
def foo():
    i = 1
    while i <= 100:
        i = i + 1
```

- improve precision with narrowing:

$$\widehat{\widehat{X}}_0 = \widehat{X} = [1, +\infty]$$

$$\widehat{\widehat{X}}_1 = [1, +\infty] \bigtriangleup \left(\left([1,1] \sqcup \left([1,+\infty] \oplus [1,1]\right)\right) \sqcap [-\infty, 100]\right)$$

Widening and Narrowing Example

```
def foo():
    i = 1
    while i <= 100:
        i = i + 1
```

- improve precision with narrowing:

$$\widehat{\widehat{X}}_0 = \widehat{X} = [1, +\infty]$$
$$\widehat{\widehat{X}}_1 = [1, +\infty] \vartriangle [1, 100]$$

Widening and Narrowing Example

```
def foo():
    i = 1
    while i <= 100:
        i = i + 1
```

- improve precision with narrowing:

$$\widehat{\widehat{X}}_0 = \widehat{X} = [1, +\infty]$$

$$\widehat{\widehat{X}}_1 = [1, 100]$$

Widening and Narrowing Example

```
def foo():
    i = 1
    while i <= 100:
        i = i + 1
```

- improve precision with narrowing:

$$\widehat{\widehat{X}}_0 = \widehat{X} = [1, +\infty]$$

$$\widehat{\widehat{X}}_1 = [1, 100]$$

$$\widehat{\widehat{X}}_2 = \widehat{\widehat{X}}_1 = [1, 100]$$

Widening and Narrowing Example

```
def foo():
    i = 1
    while i <= 100:
        i = i + 1
```

- improve precision with narrowing:

$$\widehat{\widehat{X}}_0 = \widehat{X} = [1, +\infty]$$

$$\widehat{\widehat{X}}_1 = [1, 100]$$

$$\widehat{\widehat{X}}_2 = \widehat{\widehat{X}}_1 = [1, 100]$$

  - it is not always possible to reach the least fixed point by widening + narrowing

Is Widening Necessary?

- only if the abstract domain contains infinite (or too long) ascending sequences

Is Widening Necessary?

- only if the abstract domain contains infinite (or too long) ascending sequences
- can the problem be always restated with finite abstraction?

Is Widening Necessary?

- only if the abstract domain contains infinite (or too long) ascending sequences
- can the problem be always restated with finite abstraction? no

Is Widening Necessary?

- only if the abstract domain contains infinite (or too long) ascending sequences
- can the problem be always restated with finite abstraction? no
  - suppose we have a class of programs which differ in a range of a variable computed by widening + narrowing

Is Widening Necessary?

- only if the abstract domain contains infinite (or too long) ascending sequences
- can the problem be always restated with finite abstraction? no
  - suppose we have a class of programs which differ in a range of a variable computed by widening + narrowing
  - the finite abstraction would need to contain all these ranges

Is Widening Necessary?

- only if the abstract domain contains infinite (or too long) ascending sequences
- can the problem be always restated with finite abstraction? no
  - suppose we have a class of programs which differ in a range of a variable computed by widening + narrowing
  - the finite abstraction would need to contain all these ranges
  - if the class of programs is infinite, there is no finite abstraction as precise as widening + narrowing approach

Is Widening Necessary?

- only if the abstract domain contains infinite (or too long) ascending sequences
- can the problem be always restated with finite abstraction? no
  - suppose we have a class of programs which differ in a range of a variable computed by widening + narrowing
  - the finite abstraction would need to contain all these ranges
  - if the class of programs is infinite, there is no finite abstraction as precise as widening + narrowing approach
  - the bounds might not be derivable from the program text
  - more in [CC92]

How to Design Widening and Narrowing? I

- want to define $\nabla, \Delta : L \times L \to L$
- use a finite lattice $\widehat{L}$, such that $L \underset{\gamma}{\overset{\alpha}{\rightleftarrows}} \widehat{L}$:

$$x \nabla y = \gamma(\text{sup}\{\alpha(x), \alpha(y)\})$$
$$x \Delta y = \text{inf}\{x, \gamma(\alpha(y)\}$$

How to Design Widening and Narrowing? II

- choose a specific thresholds and accelerate unstable bounds to nearest such threshold

How to Design Widening and Narrowing? II

- choose a specific thresholds and accelerate unstable bounds to nearest such threshold
  - e.g. $\{-\infty, 0, +\infty\}$ for intervals:

$$\begin{aligned}[l_0, u_0] \mathbin{\nabla} [l_1, u_1] = [\,&\mathrm{ite}(0 \leqslant l_1 < l_0, 0, \mathrm{ite}(l_1 < l_0, -\infty, l_0)),\\ &\mathrm{ite}(u_0 < u_1 \leqslant 0, 0, \mathrm{ite}(u_0 < u_1, +\infty, u_0))]\end{aligned}$$

How to Design Widening and Narrowing? II

- choose a specific thresholds and accelerate unstable bounds to nearest such threshold
  - e.g. $\{-\infty, 0, +\infty\}$ for intervals:

$$
\begin{aligned}
[l_0, u_0] \triangledown [l_1, u_1] = [\, & \mathsf{ite}(0 \leqslant l_1 < l_0, 0, \mathsf{ite}(l_1 < l_0, -\infty, l_0)), \\
& \mathsf{ite}(u_0 < u_1 \leqslant 0, 0, \mathsf{ite}(u_0 < u_1, +\infty, u_0))] \\
[l_0, u_0] \vartriangle [l_1, u_1] = [\, & \mathsf{ite}((l_0 \leqslant 0 \leqslant l_1) \vee (l_0 = -\infty), l_1, l_0), \\
& \mathsf{ite}((u_1 \leqslant 0 \leqslant u_0) \vee (l_0 = +\infty), u_1, u_0)]
\end{aligned}
$$

Final notes on Widening and Narrowing

- it is also possible to generalize widening (and narrowing) to work on more previous values
- or all previous values
  - *set widening/set narrowing*

Biblography

Patrick Cousot and Radhia Cousot. "Comparing the Galois connection and widening/narrowing approaches to abstract interpretation". In: *Programming Language Implementation and Logic Programming: 4th International Symposium, PLILP'92 Leuven, Belgium, August 26–28, 1992 Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1992, pp. 269–295. DOI: 10.1007/3-540-55844-6_142.