



# Static Program Analysis

**Lecture 8: Dataflow Analysis VII (DFA with Conditional Branches)**

**Summer Semester 2018**

**Thomas Noll**

**Software Modeling and Verification Group**

**RWTH Aachen University**

<https://moves.rwth-aachen.de/teaching/ss-18/spa/>

# Recap: Widening and Narrowing

---

## Outline of Lecture 8

Recap: Widening and Narrowing

Taking Conditional Branches into Account

Conditional Constant Propagation Analysis

Conditional Interval Analysis

Optimisation in the GNU C Compiler

# Recap: Widening and Narrowing

## Widening Operators

### Definition (Widening operator)

Let  $(D, \sqsubseteq)$  be a complete lattice. A mapping  $\nabla : D \times D \rightarrow D$  is called **widening operator** if

- for every  $d_1, d_2 \in D$ ,

$$d_1 \sqcup d_2 \sqsubseteq d_1 \nabla d_2$$

and

- for all ascending chains  $d_0 \sqsubseteq d_1 \sqsubseteq \dots$ , the ascending chain  $d_0^\nabla \sqsubseteq d_1^\nabla \sqsubseteq \dots$  eventually stabilises where

$$d_0^\nabla := d_0 \text{ and } d_{i+1}^\nabla := d_i^\nabla \nabla d_{i+1} \text{ for each } i \in \mathbb{N}$$

### Remarks:

- The requirement  $d_1 \sqcup d_2 \sqsubseteq d_1 \nabla d_2$  guarantees **soundness** of widening
- $(d_i^\nabla)_{i \in \mathbb{N}}$  is clearly an **ascending chain** as  $d_{i+1}^\nabla = d_i^\nabla \nabla d_{i+1} \sqsupseteq d_i^\nabla \sqcup d_{i+1} \sqsupseteq d_i^\nabla$
- In contrast to  $\sqcup$ ,  $\nabla$  does *not* have to be commutative, associative, monotonic, nor absorptive ( $d \nabla d = d$ )

# Recap: Widening and Narrowing

## Applying Widening to Interval Analysis

- A **widening operator**:  $\nabla : Int \times Int \rightarrow Int$  with

$$\begin{aligned}\emptyset \nabla J &:= J \nabla \emptyset := J \\ [x_1, x_2] \nabla [y_1, y_2] &:= [z_1, z_2] \quad \text{where} \\ z_1 &:= \begin{cases} x_1 & \text{if } x_1 \leq y_1 \\ -\infty & \text{otherwise} \end{cases} \\ z_2 &:= \begin{cases} x_2 & \text{if } x_2 \geq y_2 \\ +\infty & \text{otherwise} \end{cases}\end{aligned}$$

- Widening turns infinite ascending chain

$$J_0 = \emptyset \subseteq J_1 = [1, 1] \subseteq J_2 = [1, 2] \subseteq J_3 = [1, 3] \subseteq \dots$$

into a finite one:

$$\begin{aligned}J_0^\nabla &= J_0 = \emptyset \\ J_1^\nabla &= J_0^\nabla \nabla J_1 = \emptyset \nabla [1, 1] = [1, 1] \\ J_2^\nabla &= J_1^\nabla \nabla J_2 = [1, 1] \nabla [1, 2] = [1, +\infty] \\ J_3^\nabla &= J_2^\nabla \nabla J_3 = [1, +\infty] \nabla [1, 3] = [1, +\infty] \\ &\vdots\end{aligned}$$

- In fact, the maximal chain size arising with this operator is 4:

$$\emptyset \subseteq [3, 7] \subseteq [3, +\infty] \subseteq [-\infty, +\infty]$$

# Recap: Widening and Narrowing

---

## Narrowing

- **Observation:** widening can “shoot above the target”, i.e., lead to **unnecessarily imprecise results**
- **Solution:** improvement by **iterating again** from the result obtained by widening (i.e., from  $\text{fix}^\nabla(\Phi_S)$ )

$\implies$  compute  $\Phi_S^k(\text{fix}^\nabla(\Phi_S))$  for  $k = 1, 2, \dots$

- **Soundness:**  $\text{fix}^\nabla(\Phi_S) \sqsupseteq \text{fix}(\Phi_S)$  (cf. Algorithm 7.3)

$\implies \Phi_S^k(\text{fix}^\nabla(\Phi_S)) \sqsupseteq \Phi_S^k(\text{fix}(\Phi_S)) = \text{fix}(\Phi_S)$

(since  $\Phi_S$  and thus  $\Phi_S^k$  monotonic)

# Taking Conditional Branches into Account

---

## Outline of Lecture 8

Recap: Widening and Narrowing

Taking Conditional Branches into Account

Conditional Constant Propagation Analysis

Conditional Interval Analysis

Optimisation in the GNU C Compiler

# Taking Conditional Branches into Account

---

## Taking Conditional Branches into Account I

- **So far:** **values of conditions** have been ignored in analysis
- Essentially: **if** and **while** statements treated as **nondeterministic choice** between the two branches

# Taking Conditional Branches into Account

---

## Taking Conditional Branches into Account I

- **So far:** values of conditions have been ignored in analysis
- Essentially: `if` and `while` statements treated as nondeterministic choice between the two branches

### Example 8.1

```
y := 0;
z := 0;
while [x > 0]3 do
  if y < 17 then
    y := y + 1
  end;
  z := z + x;
  x := x - 1
end;
```



# Taking Conditional Branches into Account

## Taking Conditional Branches into Account I

- **So far:** values of conditions have been ignored in analysis
- Essentially: `if` and `while` statements treated as nondeterministic choice between the two branches

### Example 8.1

```
y := 0;
z := 0;
while [x > 0]3 do
  if y < 17 then
    y := y + 1
  end;
  z := z + x;
  x := x - 1
end;
```

- Interval analysis (with widening) yields for label 3:

$$\begin{aligned}x &\in [-\infty, +\infty] \\y &\in [0, +\infty] \\z &\in [-\infty, +\infty]\end{aligned}$$

# Taking Conditional Branches into Account

## Taking Conditional Branches into Account I

- **So far:** **values of conditions** have been ignored in analysis
- Essentially: **if** and **while** statements treated as **nondeterministic choice** between the two branches

### Example 8.1

```
y := 0;  
z := 0;  
while [x > 0]3 do  
  if y < 17 then  
    y := y + 1  
  end;  
  z := z + x;  
  x := x - 1  
end;
```

- Interval analysis (with widening) yields for label 3:

$$\begin{aligned}x &\in [-\infty, +\infty] \\y &\in [0, +\infty] \\z &\in [-\infty, +\infty]\end{aligned}$$

- **Too pessimistic!** In fact,

$$\begin{aligned}x &\in [-\infty, +\infty] \\y &\in [0, 17] \\z &\in [0, +\infty]\end{aligned}$$

# Taking Conditional Branches into Account

---

## Taking Conditional Branches into Account II

- **Solution:** introduce **transfer functions for branches**

# Taking Conditional Branches into Account

---

## Taking Conditional Branches into Account II

- **Solution:** introduce **transfer functions for branches**
- **First approach:** attach (negated) conditions as **labels to control flow edges**
  - advantage: no language modification required
  - disadvantage: entails extension of DFA framework
  - will not further be considered here

# Taking Conditional Branches into Account

---

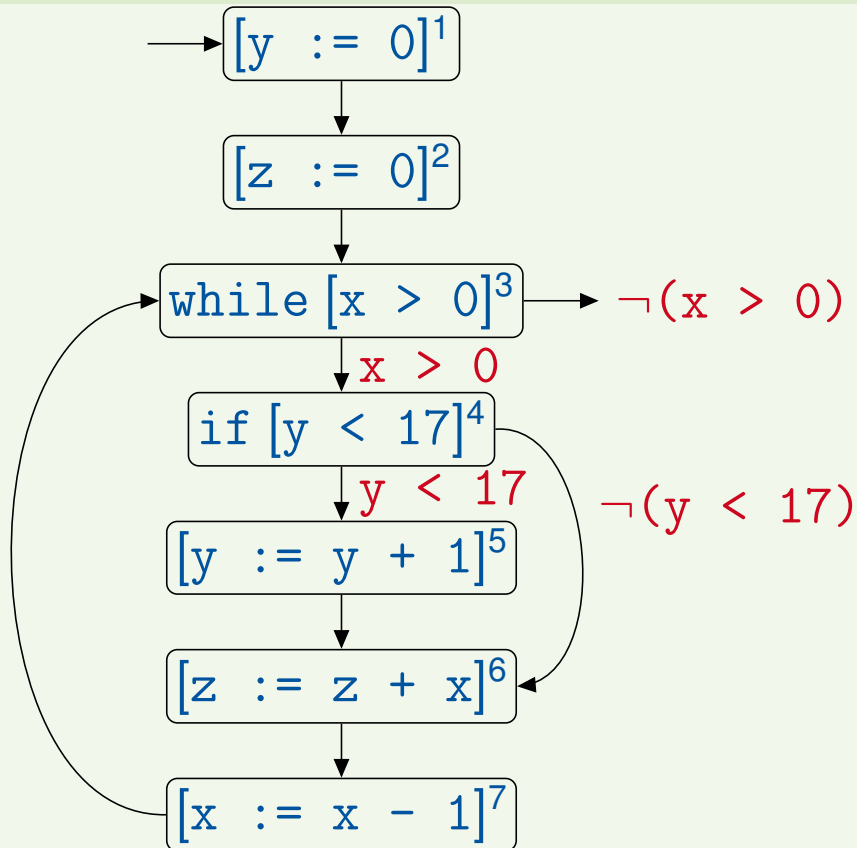
## Taking Conditional Branches into Account II

- **Solution:** introduce **transfer functions for branches**
- **First approach:** attach (negated) conditions as **labels to control flow edges**
  - advantage: no language modification required
  - disadvantage: entails extension of DFA framework
  - will not further be considered here
- **Second approach:** encode conditions as **assertions** (proper statements)
  - advantage: DFA framework can be reused
  - disadvantage: entails extension of WHILE language
  - the way we will follow

# Taking Conditional Branches into Account

## Conditions as Edge Labels

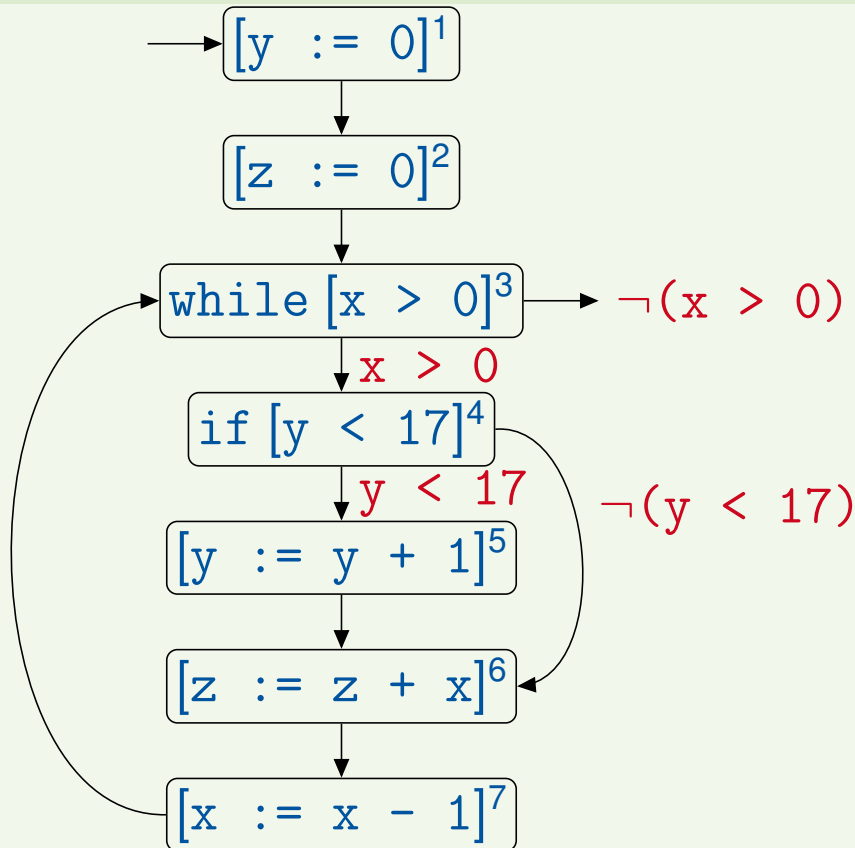
### Example 8.2 (cf. Example 8.1)



# Taking Conditional Branches into Account

## Conditions as Edge Labels vs. Conditions as Assertions

### Example 8.2 (cf. Example 8.1)



```
y := 0;  
z := 0;  
while x > 0 do  
  assert x > 0;  
  if y < 17 then  
    assert y < 17;  
    y := y + 1  
  end;  
  z := z + x;  
  x := x - 1  
end;  
assert  $\neg(x > 0)$ ;
```

# Taking Conditional Branches into Account

---

## Extending the Syntax of WHILE Programs

### Definition 8.3 (Labelled WHILE programs with assertions)

The **syntax of labelled WHILE programs with assertions** is defined by the following context-free grammar:

$$a ::= z \mid x \mid a_1 + a_2 \mid a_1 - a_2 \mid a_1 * a_2 \in AExp$$
$$b ::= t \mid a_1 = a_2 \mid a_1 > a_2 \mid \neg b \mid b_1 \wedge b_2 \mid b_1 \vee b_2 \in BExp$$
$$c ::= [\text{skip}]' \mid [x := a]' \mid c_1 ; c_2 \mid$$
$$\text{if } [b]' \text{ then } c_1 \text{ else } c_2 \text{ end} \mid \text{while } [b]' \text{ do } c \text{ end} \mid [\text{assert } b]' \in Cmd$$



# Taking Conditional Branches into Account

## Extending the Syntax of WHILE Programs

### Definition 8.3 (Labelled WHILE programs with assertions)

The **syntax of labelled WHILE programs with assertions** is defined by the following context-free grammar:

$$a ::= z \mid x \mid a_1 + a_2 \mid a_1 - a_2 \mid a_1 * a_2 \in AExp$$
$$b ::= t \mid a_1 = a_2 \mid a_1 > a_2 \mid \neg b \mid b_1 \wedge b_2 \mid b_1 \vee b_2 \in BExp$$
$$c ::= [\text{skip}]' \mid [x := a]' \mid c_1 ; c_2 \mid$$
$$\text{if } [b]' \text{ then } c_1 \text{ else } c_2 \text{ end} \mid \text{while } [b]' \text{ do } c \text{ end} \mid [\text{assert } b]' \in Cmd$$

### To be done:

- Definition of **transfer functions** for **assert** blocks (tailored to analysis problem)
- Idea: assertions as **filters** that let only “compatible” information pass

# Conditional Constant Propagation Analysis

---

## Outline of Lecture 8

Recap: Widening and Narrowing

Taking Conditional Branches into Account

**Conditional Constant Propagation Analysis**

Conditional Interval Analysis

Optimisation in the GNU C Compiler

# Conditional Constant Propagation Analysis

## Original Constant Propagation Analysis

- **Complete lattice**  $(D, \sqsubseteq)$  where
  - $D := \{\delta \mid \delta : \text{Var}_c \rightarrow \mathbb{Z} \cup \{\perp, \top\}\}$ 
    - $\delta(x) = z \in \mathbb{Z}$ :  $x$  has constant value  $z$  (i.e., possible values in  $\{z\}$ )
    - $\delta(x) = \perp$ :  $x$  undefined (i.e., possible values in  $\emptyset$ )
    - $\delta(x) = \top$ :  $x$  overdefined (i.e., possible values in  $\mathbb{Z}$ )
  - $\sqsubseteq \subseteq D \times D$  defined by pointwise extension of  $\perp \sqsubseteq z \sqsubseteq \top$  (for every  $z \in \mathbb{Z}$ )
- **Transfer functions**  $\{\varphi_l \mid l \in \text{Lab}\}$  defined by

$$\varphi_l(\delta) := \begin{cases} \delta & \text{if } B^l = \text{skip or } B^l \in \text{BExp} \\ \delta[x \mapsto \text{val}_\delta(a)] & \text{if } B^l = (x := a) \end{cases}$$

where

$$\begin{aligned} \text{val}_\delta(x) &:= \delta(x) \\ \text{val}_\delta(z) &:= z \end{aligned} \quad \text{val}_\delta(a_1 \text{ op } a_2) := \begin{cases} z_1 \text{ op } z_2 & \text{if } z_1, z_2 \in \mathbb{Z} \\ \perp & \text{if } z_1 = \perp \text{ or } z_2 = \perp \\ \top & \text{otherwise} \end{cases}$$

for  $z_1 := \text{val}_\delta(a_1)$  and  $z_2 := \text{val}_\delta(a_2)$

# Conditional Constant Propagation Analysis

---

## Transfer Functions of Assertions I

Additionally for  $B' = (\text{assert } b)$ ,  $\delta : \text{Var}_c \rightarrow \mathbb{Z} \cup \{\perp, \top\}$  and  $x \in \text{Var}_c$ :

$$\varphi_I(\delta)(x) := \begin{cases} \perp & \text{if } \nexists \sigma \in \Sigma_\delta : \text{val}_\sigma(b) = \text{true} \\ z & \text{if } \forall \sigma \in \Sigma_\delta : \text{val}_\sigma(b) = \text{true} \implies \sigma(x) = z \\ \top & \text{otherwise} \end{cases}$$

# Conditional Constant Propagation Analysis

---

## Transfer Functions of Assertions I

Additionally for  $B' = (\text{assert } b)$ ,  $\delta : \text{Var}_c \rightarrow \mathbb{Z} \cup \{\perp, \top\}$  and  $x \in \text{Var}_c$ :

$$\varphi_I(\delta)(x) := \begin{cases} \perp & \text{if } \nexists \sigma \in \Sigma_\delta : \text{val}_\sigma(b) = \text{true} \\ z & \text{if } \forall \sigma \in \Sigma_\delta : \text{val}_\sigma(b) = \text{true} \implies \sigma(x) = z \\ \top & \text{otherwise} \end{cases}$$

where

- the **set of  $\delta$ -states** is given by

$$\Sigma_\delta := \left\{ \sigma : \text{Var}_c \rightarrow \mathbb{Z} \mid \forall y \in \text{Var}_c : \sigma(y) \in \begin{cases} \emptyset & \text{if } \delta(y) = \perp \\ \{z\} & \text{if } \delta(y) = z \\ \mathbb{Z} & \text{if } \delta(y) = \top \end{cases} \right\}$$

(and thus  $\Sigma_\delta = \emptyset$  iff  $\delta(y) = \perp$  for some  $y \in \text{Var}_c$ )

# Conditional Constant Propagation Analysis

## Transfer Functions of Assertions I

Additionally for  $B' = (\text{assert } b)$ ,  $\delta : \text{Var}_c \rightarrow \mathbb{Z} \cup \{\perp, \top\}$  and  $x \in \text{Var}_c$ :

$$\varphi_I(\delta)(x) := \begin{cases} \perp & \text{if } \nexists \sigma \in \Sigma_\delta : \text{val}_\sigma(b) = \text{true} \\ z & \text{if } \forall \sigma \in \Sigma_\delta : \text{val}_\sigma(b) = \text{true} \implies \sigma(x) = z \\ \top & \text{otherwise} \end{cases}$$

where

- the **set of  $\delta$ -states** is given by

$$\Sigma_\delta := \left\{ \sigma : \text{Var}_c \rightarrow \mathbb{Z} \mid \forall y \in \text{Var}_c : \sigma(y) \in \begin{cases} \emptyset & \text{if } \delta(y) = \perp \\ \{z\} & \text{if } \delta(y) = z \\ \mathbb{Z} & \text{if } \delta(y) = \top \end{cases} \right\}$$

(and thus  $\Sigma_\delta = \emptyset$  iff  $\delta(y) = \perp$  for some  $y \in \text{Var}_c$ )

- the **evaluation function**  $\text{val}_\sigma : \text{BExp} \rightarrow \mathbb{B}$  for  $\sigma : \text{Var}_c \rightarrow \mathbb{Z}$  is defined by

$$\begin{aligned} \text{val}_\sigma(t) &:= t & \text{val}_\sigma(\neg b) &:= \begin{cases} \text{true} & \text{if } \text{val}_\sigma(b) = \text{false} \\ \text{false} & \text{otherwise} \end{cases} \\ \text{val}_\sigma(a_1 = a_2) &:= (\text{val}_\sigma(a_1) = \text{val}_\sigma(a_2)) & \text{val}_\sigma(b_1 \wedge b_2) &:= \begin{cases} \text{true} & \text{if } \text{val}_\sigma(b_1) = \text{val}_\sigma(b_2) = \text{true} \\ \text{false} & \text{otherwise} \end{cases} \end{aligned}$$

( $\text{val}_\sigma : \text{AExp} \rightarrow \mathbb{Z}$  on previous slide)

## Transfer Functions of Assertions II

### Example 8.4

1.  $Var_c = \{x, y, z\}$ ,  $\delta = (\underbrace{\perp}_x, \underbrace{1}_y, \underbrace{\top}_z)$

$$\implies \Sigma_\delta = \emptyset \implies \varphi_{\text{assert } b}(\delta) = (\perp, \perp, \perp) \text{ for every } b \in BExp$$

# Conditional Constant Propagation Analysis

## Transfer Functions of Assertions II

### Example 8.4

1.  $Var_c = \{x, y, z\}$ ,  $\delta = (\underbrace{\perp}_x, \underbrace{1}_y, \underbrace{\top}_z)$

$$\implies \Sigma_\delta = \emptyset \implies \varphi_{\text{assert } b}(\delta) = (\perp, \perp, \perp) \text{ for every } b \in BExp$$

2.  $Var_c = \{x, y, z\}$ ,  $\delta = (\underbrace{1}_x, \underbrace{2}_y, \underbrace{\top}_z)$

$$\implies \Sigma_\delta = \{(1, 2, z) \mid z \in \mathbb{Z}\} \implies \varphi_{\text{assert } x=y}(\delta) = (\perp, \perp, \perp)$$

$$\varphi_{\text{assert } y=z}(\delta) = (1, 2, 2)$$

$$\varphi_{\text{assert } y < z}(\delta) = (1, 2, \top)$$

$$\varphi_{\text{assert } x \leq z \wedge y > z}(\delta) = (1, 2, 1)$$



# Conditional Constant Propagation Analysis

## Transfer Functions of Assertions II

### Example 8.4

1.  $Var_c = \{x, y, z\}$ ,  $\delta = (\underbrace{\perp}_x, \underbrace{1}_y, \underbrace{\top}_z)$

$$\implies \Sigma_\delta = \emptyset \implies \varphi_{\text{assert } b}(\delta) = (\perp, \perp, \perp) \text{ for every } b \in BExp$$

2.  $Var_c = \{x, y, z\}$ ,  $\delta = (\underbrace{1}_x, \underbrace{2}_y, \underbrace{\top}_z)$

$$\implies \Sigma_\delta = \{(1, 2, z) \mid z \in \mathbb{Z}\} \implies \varphi_{\text{assert } x=y}(\delta) = (\perp, \perp, \perp)$$

$$\varphi_{\text{assert } y=z}(\delta) = (1, 2, 2)$$

$$\varphi_{\text{assert } y < z}(\delta) = (1, 2, \top)$$

$$\varphi_{\text{assert } x \leq z \wedge y > z}(\delta) = (1, 2, 1)$$

3.  $Var_c = \{x, y, z\}$ ,  $\delta = (\underbrace{1}_x, \underbrace{\top}_y, \underbrace{\top}_z)$

$$\implies \Sigma_\delta = \{(1, z_1, z_2) \mid z_1, z_2 \in \mathbb{Z}\} \implies \varphi_{\text{assert } x=y}(\delta) = (1, 1, \top)$$

$$\varphi_{\text{assert } y=z}(\delta) = (1, \top, \top)$$

# Conditional Constant Propagation Analysis

---

## Transfer Functions of Assertions III

### Remarks:

- For  $B' = (\text{assert } b)$  and  $\delta : \text{Var}_c \rightarrow \mathbb{Z} \cup \{\perp, \top\}$ ,  $\varphi_l(\delta) \sqsubseteq \delta$  and hence  $\Sigma_{\varphi_l(\delta)} \subseteq \Sigma_\delta$  (“filter”)

# Conditional Constant Propagation Analysis

---

## Transfer Functions of Assertions III

### Remarks:

- For  $B' = (\text{assert } b)$  and  $\delta : \text{Var}_c \rightarrow \mathbb{Z} \cup \{\perp, \top\}$ ,  $\varphi_l(\delta) \sqsubseteq \delta$  and hence  $\Sigma_{\varphi_l(\delta)} \subseteq \Sigma_\delta$  (“filter”)
- Constant propagation captures **interdependencies** between variables only when both are constant (cf. “`assert y=z`” in Example 8.4(2)/(3))

# Conditional Constant Propagation Analysis

---

## Transfer Functions of Assertions III

### Remarks:

- For  $B' = (\text{assert } b)$  and  $\delta : \text{Var}_c \rightarrow \mathbb{Z} \cup \{\perp, \top\}$ ,  $\varphi_I(\delta) \sqsubseteq \delta$  and hence  $\Sigma_{\varphi_I(\delta)} \subseteq \Sigma_\delta$  (“filter”)
- Constant propagation captures **interdependencies** between variables only when both are constant (cf. “`assert y=z`” in Example 8.4(2)/(3))
- $\varphi_I(\delta)$  can be determined (or at least approximated) by **Satisfiability Modulo Theories (SMT)** techniques

# Conditional Constant Propagation Analysis

---

## Transfer Functions of Assertions III

### Remarks:

- For  $B^l = (\text{assert } b)$  and  $\delta : \text{Var}_c \rightarrow \mathbb{Z} \cup \{\perp, \top\}$ ,  $\varphi_l(\delta) \sqsubseteq \delta$  and hence  $\Sigma_{\varphi_l(\delta)} \subseteq \Sigma_\delta$  (“filter”)
- Constant propagation captures **interdependencies** between variables only when both are constant (cf. “`assert y=z`” in Example 8.4(2)/(3))
- $\varphi_l(\delta)$  can be determined (or at least approximated) by **Satisfiability Modulo Theories (SMT)** techniques
- If  $\text{CP}_l(x) = \perp$  for some  $l \in \text{Lab}_c$  and  $x \in \text{Var}_c$ , then  $l$  is **unreachable** (and  $\text{CP}_l(y) = \perp$  for all  $y \in \text{Var}_c$ )

# Conditional Constant Propagation Analysis

---

## An Example

### Example 8.5

```
if [x = 1]1 then
  [assert x = 1]2;
  [y := x + 1]3
else
  [assert ¬(x = 1)]4;
  [y := 2]5
end;
[skip]6
```

# Conditional Constant Propagation Analysis

## An Example

### Example 8.5

```
if [x = 1]1 then  
  [assert x = 1]2;  
  [y := x + 1]3
```

```
else
```

```
  [assert ¬(x = 1)]4;  
  [y := 2]5
```

```
end;
```

```
[skip]6
```

**Without assertions**

$CP_1 = (\top, \top)$

**With assertions**

$CP_1 = (\top, \top)$

# Conditional Constant Propagation Analysis

## An Example

### Example 8.5

```
if [x = 1]1 then  
  [assert x = 1]2;  
  [y := x + 1]3
```

```
else
```

```
  [assert ¬(x = 1)]4;  
  [y := 2]5
```

```
end;
```

```
[skip]6
```

**Without assertions**

$CP_1 = (\top, \top)$

**With assertions**

$CP_1 = (\top, \top)$

$CP_2 = (\top, \top)$



# Conditional Constant Propagation Analysis

## An Example

### Example 8.5

```
if [x = 1]1 then  
  [assert x = 1]2;  
  [y := x + 1]3
```

```
else  
  [assert ¬(x = 1)]4;  
  [y := 2]5
```

```
end;  
[skip]6
```

#### Without assertions

$CP_1 = (\top, \top)$

$CP_3 = (\top, \top)$

#### With assertions

$CP_1 = (\top, \top)$

$CP_2 = (\top, \top)$

$CP_3 = (1, \top)$

# Conditional Constant Propagation Analysis

## An Example

### Example 8.5

```
if [x = 1]1 then  
  [assert x = 1]2;  
  [y := x + 1]3
```

```
else  
  [assert ¬(x = 1)]4;  
  [y := 2]5
```

```
end;  
[skip]6
```

#### Without assertions

$CP_1 = (\top, \top)$

$CP_3 = (\top, \top)$

#### With assertions

$CP_1 = (\top, \top)$

$CP_2 = (\top, \top)$

$CP_3 = (1, \top)$

$CP_4 = (\top, \top)$

# Conditional Constant Propagation Analysis

## An Example

### Example 8.5

```
if [x = 1]1 then  
  [assert x = 1]2;  
  [y := x + 1]3
```

```
else  
  [assert ¬(x = 1)]4;  
  [y := 2]5
```

```
end;  
[skip]6
```

#### Without assertions

$CP_1 = (\top, \top)$

$CP_3 = (\top, \top)$

$CP_5 = (\top, \top)$

#### With assertions

$CP_1 = (\top, \top)$

$CP_2 = (\top, \top)$

$CP_3 = (1, \top)$

$CP_4 = (\top, \top)$

$CP_5 = (\top, \top)$

# Conditional Constant Propagation Analysis

## An Example

### Example 8.5

```
if [x = 1]1 then  
  [assert x = 1]2;  
  [y := x + 1]3
```

```
else  
  [assert ¬(x = 1)]4;  
  [y := 2]5
```

```
end;  
[skip]6
```

#### Without assertions

$$CP_1 = (\top, \top)$$

$$CP_3 = (\top, \top)$$

$$CP_5 = (\top, \top)$$

$$CP_6 = (\top, \top) \sqcup (\top, 2) \\ = (\top, \top)$$

#### With assertions

$$CP_1 = (\top, \top)$$

$$CP_2 = (\top, \top)$$

$$CP_3 = (1, \top)$$

$$CP_4 = (\top, \top)$$

$$CP_5 = (\top, \top)$$

$$CP_6 = (1, 2) \sqcup (\top, 2) \\ = (\top, 2)$$

# Conditional Interval Analysis

---

## Outline of Lecture 8

Recap: Widening and Narrowing

Taking Conditional Branches into Account

Conditional Constant Propagation Analysis

**Conditional Interval Analysis**

Optimisation in the GNU C Compiler

# Conditional Interval Analysis

---

## Original Interval Analysis

- **Intervals** over  $\mathbb{Z}$  defined by

$$Int := \{[z_1, z_2] \mid z_1 \in \mathbb{Z} \cup \{-\infty\}, z_2 \in \mathbb{Z} \cup \{+\infty\}, z_1 \leq z_2\} \cup \{\emptyset\}$$

where

- $-\infty \leq z \leq +\infty$  (for all  $z \in \mathbb{Z}$ )
- $\emptyset \subseteq J$  (for all  $J \in Int$ )
- $[y_1, y_2] \subseteq [z_1, z_2]$  iff  $z_1 \leq y_1$  and  $y_2 \leq z_2$
- **Complete lattice**  $(D, \sqsubseteq)$  where
  - $D := \{\delta \mid \delta : Var_c \rightarrow Int\}$
  - $\delta_1 \sqsubseteq \delta_2$  iff  $\delta_1(x) \subseteq \delta_2(x)$  for every  $x \in Var_c$
- **Transfer functions**  $\{\varphi_l \mid l \in Lab\}$  are defined by

$$\varphi_l(\delta) := \begin{cases} \delta & \text{if } B' = \text{skip or } B' \in BExp \\ \delta[x \mapsto val_\delta(a)] & \text{if } B' = (x := a) \end{cases}$$

where

$$\begin{array}{ll} val_\delta(x) := \delta(x) & val_\delta(a_1 + a_2) := val_\delta(a_1) \oplus val_\delta(a_2) \\ val_\delta(z) := [z, z] & val_\delta(a_1 - a_2) := val_\delta(a_1) \ominus val_\delta(a_2) \\ & val_\delta(a_1 * a_2) := val_\delta(a_1) \odot val_\delta(a_2) \end{array}$$

# Conditional Interval Analysis

---

## Transfer Functions of Assertions I

Additionally for  $B' = (\text{assert } b)$ ,  $\delta : \text{Var}_c \rightarrow \text{Int}$  and  $x \in \text{Var}_c$ :

$$\varphi_I(\delta)(x) := \begin{cases} \emptyset & \text{if } Z = \emptyset \\ \left[ \prod_{Z \cup \{-\infty\}} Z, \bigsqcup_{Z \cup \{+\infty\}} Z \right] & \text{otherwise} \end{cases}$$

where

- $Z := \{\sigma(x) \mid \sigma \in \Sigma_\delta, \text{val}_\sigma(b) = \text{true}\}$
- $\Sigma_\delta := \{\sigma : \text{Var}_c \rightarrow \mathbb{Z} \mid \forall y \in \text{Var}_c : \sigma(y) \in \delta(y)\}$   
(and thus  $\Sigma_\delta = \emptyset$  iff  $\delta(y) = \emptyset$  for some  $y \in \text{Var}_c$ )
- $\text{val}_\sigma : \text{BExp} \rightarrow \mathbb{B}$ : see Slide 8.13

# Conditional Interval Analysis

## Transfer Functions of Assertions II

### Example 8.6

$$\text{Var}_c = \{x, y\}, \delta = (\underbrace{[-\infty, 2]}_x, \underbrace{[0, +\infty]}_y)$$

$$\implies \varphi_{\text{assert } x > 0}(\delta) = ([1, 2], [0, +\infty])$$

$$\varphi_{\text{assert } x=y}(\delta) = ([0, 2], [0, 2])$$

$$\varphi_{\text{assert } x > y}(\delta) = ([1, 2], [0, 1])$$

$$\varphi_{\text{assert } x < y}(\delta) = ([-\infty, 2], [0, +\infty])$$



# Conditional Interval Analysis

## Transfer Functions of Assertions II

### Example 8.6

$$Var_c = \{x, y\}, \delta = (\underbrace{[-\infty, 2]}_x, \underbrace{[0, +\infty]}_y)$$

$$\implies \varphi_{\text{assert } x>0}(\delta) = ([1, 2], [0, +\infty])$$

$$\varphi_{\text{assert } x=y}(\delta) = ([0, 2], [0, 2])$$

$$\varphi_{\text{assert } x>y}(\delta) = ([1, 2], [0, 1])$$

$$\varphi_{\text{assert } x<y}(\delta) = ([-\infty, 2], [0, +\infty])$$

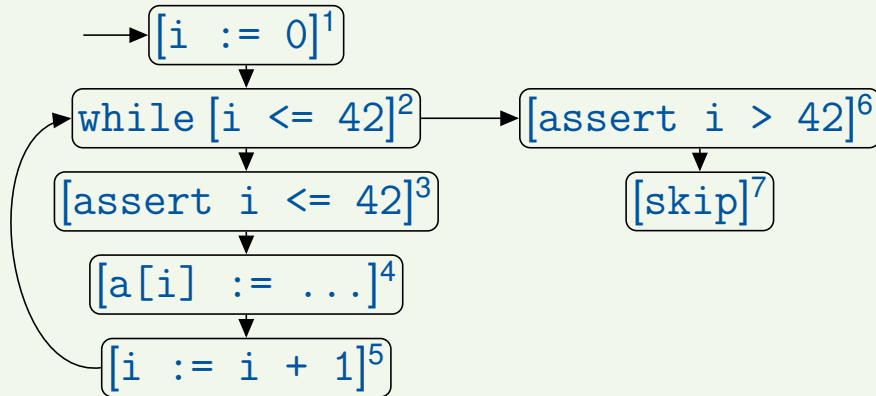
### Remarks:

- Again for  $B^l = (\text{assert } b)$  and  $\delta : Var_c \rightarrow Int$ ,  $\varphi_l(\delta) \sqsubseteq \delta$  and hence  $\Sigma_{\varphi_l(\delta)} \subseteq \Sigma_\delta$  (“filter”)
- Again if  $AI_l(x) = \emptyset$  for some  $l \in Lab_c$  and  $x \in Var_c$ , then  $l$  is **unreachable** (and  $AI_l(y) = \emptyset$  for all  $y \in Var_c$ )

# Conditional Interval Analysis

## An Example

### Example 8.7 (Interval analysis for array index; cf. Example 6.7)



$$\varphi_1(J) = [0, 0]$$

$$\varphi_2(J) = J$$

$$\varphi_3(J) = J \cap [-\infty, 42]$$

$$\varphi_4(J) = J$$

$$\varphi_5(\emptyset) = \emptyset$$

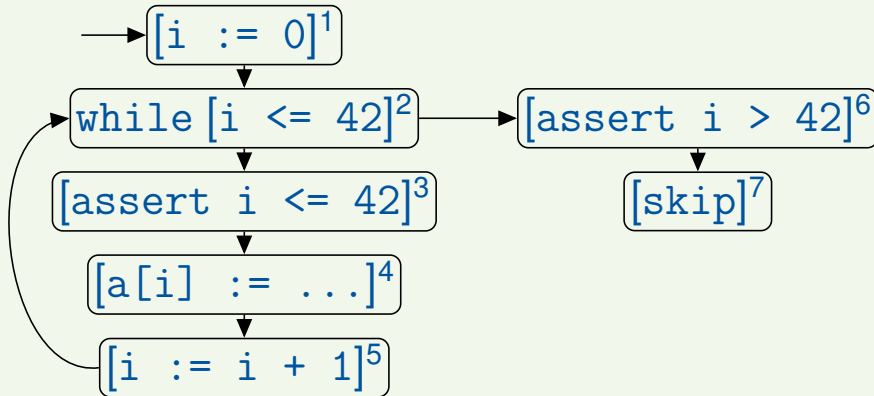
$$\varphi_5([i_1, i_2]) = [i_1 + 1, i_2 + 1]$$

$$\varphi_6(J) = J \cap [43, +\infty]$$

# Conditional Interval Analysis

## An Example

### Example 8.7 (Interval analysis for array index; cf. Example 6.7)



$$\varphi_1(J) = [0, 0]$$

$$\varphi_2(J) = J$$

$$\varphi_3(J) = J \cap [-\infty, 42]$$

$$\varphi_4(J) = J$$

$$\varphi_5(\emptyset) = \emptyset$$

$$\varphi_5([i_1, i_2]) = [i_1 + 1, i_2 + 1]$$

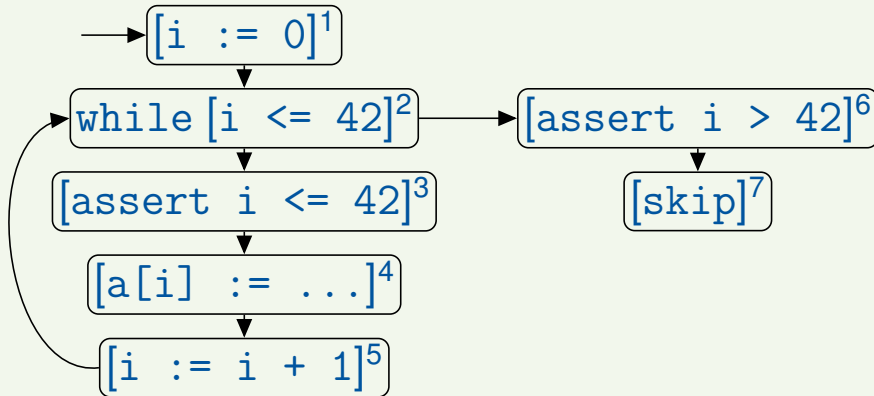
$$\varphi_6(J) = J \cap [43, +\infty]$$

Worklist w. widening	AI <sub>1</sub>	AI <sub>2</sub>	AI <sub>3</sub>	AI <sub>4</sub>	AI <sub>5</sub>	AI <sub>6</sub>	AI <sub>7</sub>
12, 23, 34, 45, 52, 26, 67	$[-\infty, +\infty]$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

# Conditional Interval Analysis

## An Example

### Example 8.7 (Interval analysis for array index; cf. Example 6.7)



$$\varphi_1(J) = [0, 0]$$

$$\varphi_2(J) = J$$

$$\varphi_3(J) = J \cap [-\infty, 42]$$

$$\varphi_4(J) = J$$

$$\varphi_5(\emptyset) = \emptyset$$

$$\varphi_5([i_1, i_2]) = [i_1 + 1, i_2 + 1]$$

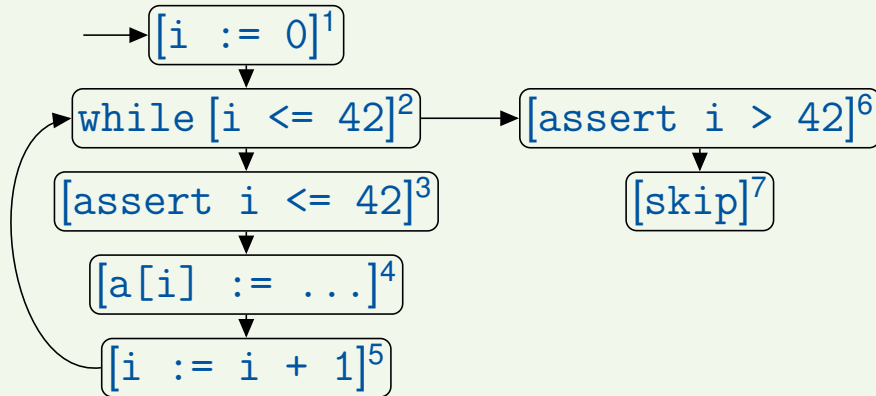
$$\varphi_6(J) = J \cap [43, +\infty]$$

Worklist w. widening	AI <sub>1</sub>	AI <sub>2</sub>	AI <sub>3</sub>	AI <sub>4</sub>	AI <sub>5</sub>	AI <sub>6</sub>	AI <sub>7</sub>
12, 23, 34, 45, 52, 26, 67	$[-\infty, +\infty]$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
23, 34, 45, 52, 26, 67	$[-\infty, +\infty]$	$[0, 0]$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

# Conditional Interval Analysis

## An Example

### Example 8.7 (Interval analysis for array index; cf. Example 6.7)



$$\varphi_1(J) = [0, 0]$$

$$\varphi_2(J) = J$$

$$\varphi_3(J) = J \cap [-\infty, 42]$$

$$\varphi_4(J) = J$$

$$\varphi_5(\emptyset) = \emptyset$$

$$\varphi_5([i_1, i_2]) = [i_1 + 1, i_2 + 1]$$

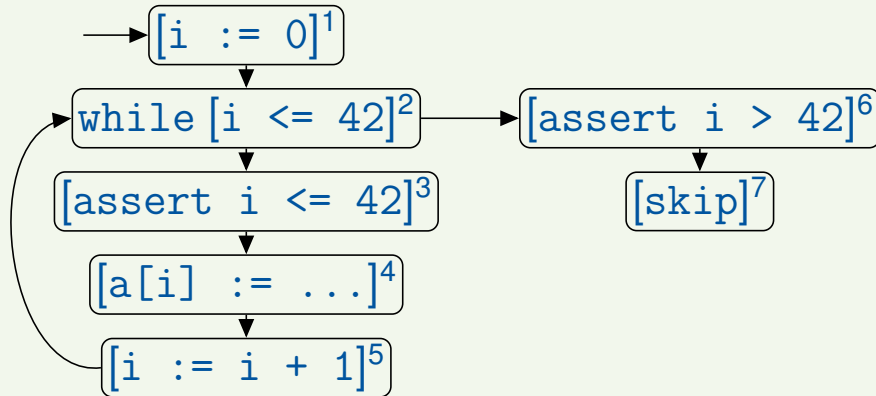
$$\varphi_6(J) = J \cap [43, +\infty]$$

Worklist w. widening	AI <sub>1</sub>	AI <sub>2</sub>	AI <sub>3</sub>	AI <sub>4</sub>	AI <sub>5</sub>	AI <sub>6</sub>	AI <sub>7</sub>
12, 23, 34, 45, 52, 26, 67	$[-\infty, +\infty]$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
23, 34, 45, 52, 26, 67	$[-\infty, +\infty]$	$[0, 0]$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
34, 45, 52, 26, 67	$[-\infty, +\infty]$	$[0, 0]$	$[0, 0]$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

# Conditional Interval Analysis

## An Example

### Example 8.7 (Interval analysis for array index; cf. Example 6.7)



$$\varphi_1(J) = [0, 0]$$

$$\varphi_2(J) = J$$

$$\varphi_3(J) = J \cap [-\infty, 42]$$

$$\varphi_4(J) = J$$

$$\varphi_5(\emptyset) = \emptyset$$

$$\varphi_5([i_1, i_2]) = [i_1 + 1, i_2 + 1]$$

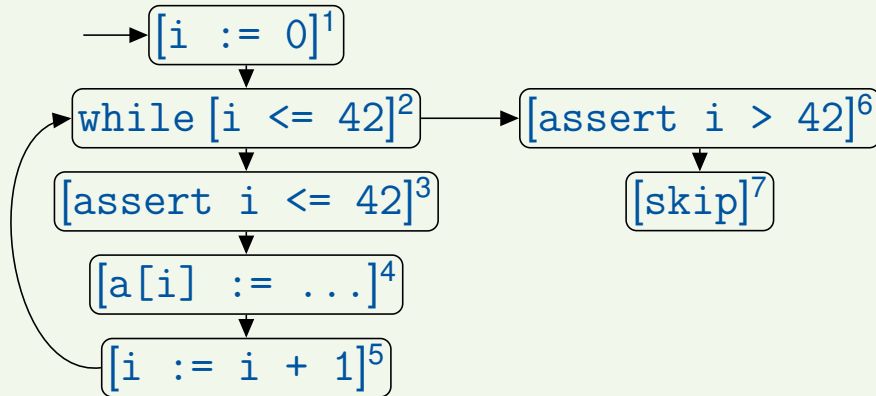
$$\varphi_6(J) = J \cap [43, +\infty]$$

Worklist w. widening	AI <sub>1</sub>	AI <sub>2</sub>	AI <sub>3</sub>	AI <sub>4</sub>	AI <sub>5</sub>	AI <sub>6</sub>	AI <sub>7</sub>
12, 23, 34, 45, 52, 26, 67	$[-\infty, +\infty]$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
23, 34, 45, 52, 26, 67	$[-\infty, +\infty]$	$[0, 0]$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
34, 45, 52, 26, 67	$[-\infty, +\infty]$	$[0, 0]$	$[0, 0]$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
45, 52, 26, 67	$[-\infty, +\infty]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$\emptyset$	$\emptyset$	$\emptyset$

# Conditional Interval Analysis

## An Example

### Example 8.7 (Interval analysis for array index; cf. Example 6.7)



$$\varphi_1(J) = [0, 0]$$

$$\varphi_2(J) = J$$

$$\varphi_3(J) = J \cap [-\infty, 42]$$

$$\varphi_4(J) = J$$

$$\varphi_5(\emptyset) = \emptyset$$

$$\varphi_5([i_1, i_2]) = [i_1 + 1, i_2 + 1]$$

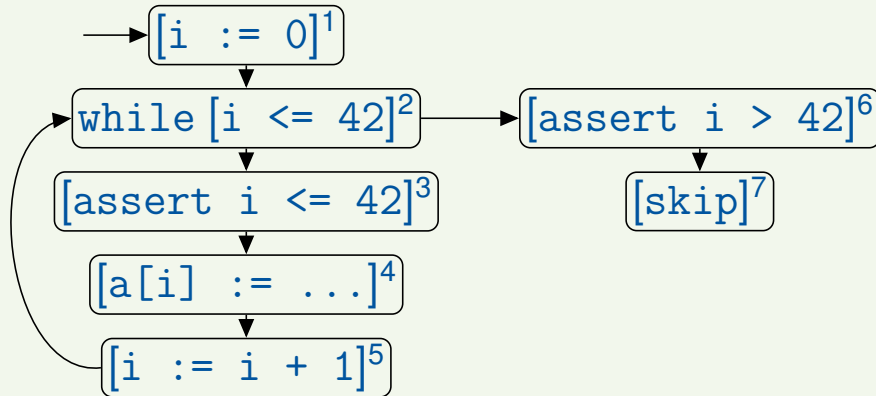
$$\varphi_6(J) = J \cap [43, +\infty]$$

Worklist w. widening	AI <sub>1</sub>	AI <sub>2</sub>	AI <sub>3</sub>	AI <sub>4</sub>	AI <sub>5</sub>	AI <sub>6</sub>	AI <sub>7</sub>
12, 23, 34, 45, 52, 26, 67	$[-\infty, +\infty]$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
23, 34, 45, 52, 26, 67	$[-\infty, +\infty]$	$[0, 0]$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
34, 45, 52, 26, 67	$[-\infty, +\infty]$	$[0, 0]$	$[0, 0]$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
45, 52, 26, 67	$[-\infty, +\infty]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$\emptyset$	$\emptyset$	$\emptyset$
52, 26, 67	$[-\infty, +\infty]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$\emptyset$	$\emptyset$

# Conditional Interval Analysis

## An Example

### Example 8.7 (Interval analysis for array index; cf. Example 6.7)



$$\varphi_1(J) = [0, 0]$$

$$\varphi_2(J) = J$$

$$\varphi_3(J) = J \cap [-\infty, 42]$$

$$\varphi_4(J) = J$$

$$\varphi_5(\emptyset) = \emptyset$$

$$\varphi_5([i_1, i_2]) = [i_1 + 1, i_2 + 1]$$

$$\varphi_6(J) = J \cap [43, +\infty]$$

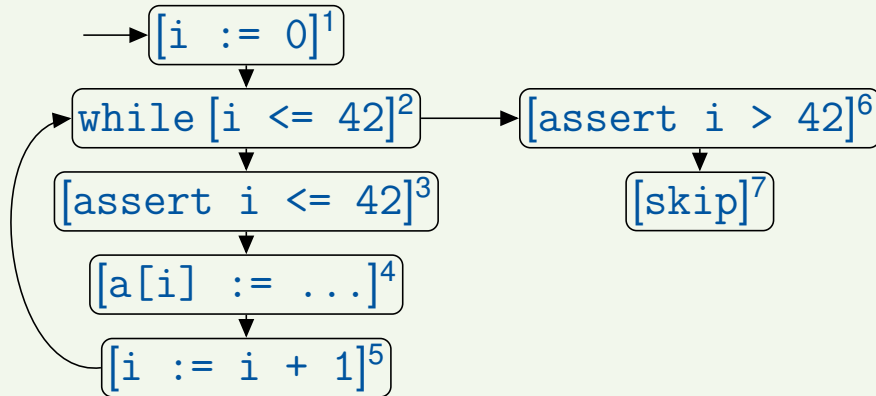
Worklist w. widening	AI <sub>1</sub>	AI <sub>2</sub>	AI <sub>3</sub>	AI <sub>4</sub>	AI <sub>5</sub>	AI <sub>6</sub>	AI <sub>7</sub>
12, 23, 34, 45, 52, 26, 67	$[-\infty, +\infty]$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
23, 34, 45, 52, 26, 67	$[-\infty, +\infty]$	$[0, 0]$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
34, 45, 52, 26, 67	$[-\infty, +\infty]$	$[0, 0]$	$[0, 0]$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
45, 52, 26, 67	$[-\infty, +\infty]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$\emptyset$	$\emptyset$	$\emptyset$
52, 26, 67	$[-\infty, +\infty]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$\emptyset$	$\emptyset$
23, 26, 67	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$\emptyset$	$\emptyset$



# Conditional Interval Analysis

## An Example

### Example 8.7 (Interval analysis for array index; cf. Example 6.7)



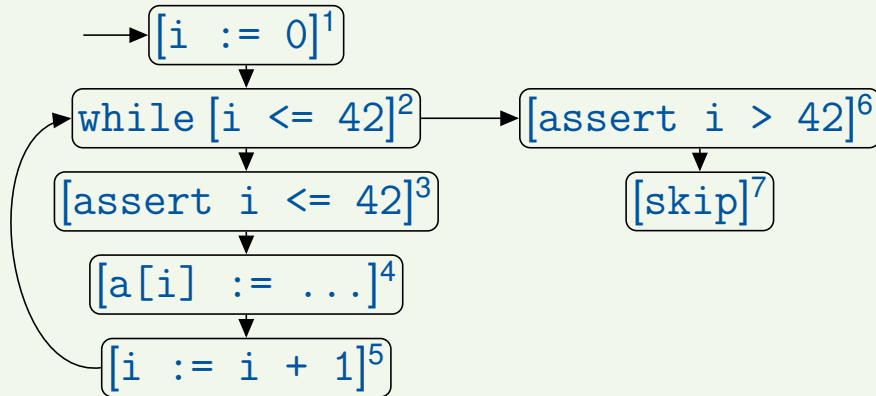
$$\begin{aligned} \varphi_1(J) &= [0, 0] \\ \varphi_2(J) &= J \\ \varphi_3(J) &= J \cap [-\infty, 42] \\ \varphi_4(J) &= J \\ \varphi_5(\emptyset) &= \emptyset \\ \varphi_5([i_1, i_2]) &= [i_1 + 1, i_2 + 1] \\ \varphi_6(J) &= J \cap [43, +\infty] \end{aligned}$$

Worklist w. widening	AI <sub>1</sub>	AI <sub>2</sub>	AI <sub>3</sub>	AI <sub>4</sub>	AI <sub>5</sub>	AI <sub>6</sub>	AI <sub>7</sub>
12, 23, 34, 45, 52, 26, 67	$[-\infty, +\infty]$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
23, 34, 45, 52, 26, 67	$[-\infty, +\infty]$	$[0, 0]$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
34, 45, 52, 26, 67	$[-\infty, +\infty]$	$[0, 0]$	$[0, 0]$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
45, 52, 26, 67	$[-\infty, +\infty]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$\emptyset$	$\emptyset$	$\emptyset$
52, 26, 67	$[-\infty, +\infty]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$\emptyset$	$\emptyset$
23, 26, 67	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$\emptyset$	$\emptyset$
34, 26, 67	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, 0]$	$[0, 0]$	$\emptyset$	$\emptyset$

# Conditional Interval Analysis

## An Example

### Example 8.7 (Interval analysis for array index; cf. Example 6.7)



$$\varphi_1(J) = [0, 0]$$

$$\varphi_2(J) = J$$

$$\varphi_3(J) = J \cap [-\infty, 42]$$

$$\varphi_4(J) = J$$

$$\varphi_5(\emptyset) = \emptyset$$

$$\varphi_5([i_1, i_2]) = [i_1 + 1, i_2 + 1]$$

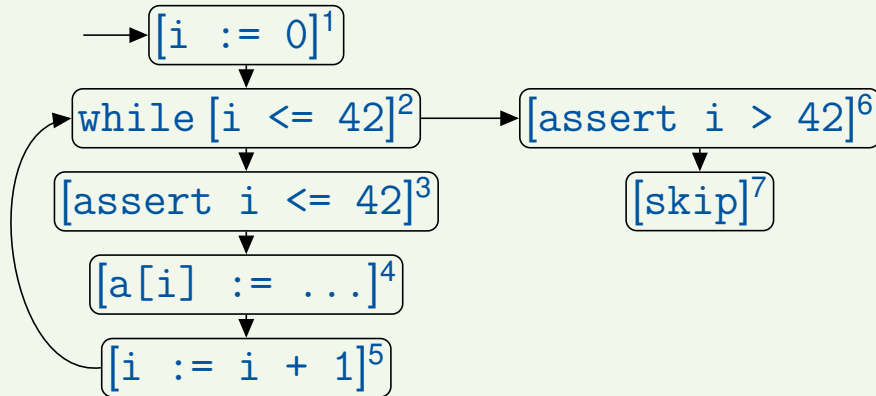
$$\varphi_6(J) = J \cap [43, +\infty]$$

Worklist w. widening	AI <sub>1</sub>	AI <sub>2</sub>	AI <sub>3</sub>	AI <sub>4</sub>	AI <sub>5</sub>	AI <sub>6</sub>	AI <sub>7</sub>
23, 34, 45, 52, 26, 67	$[-\infty, +\infty]$	$[0, 0]$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
34, 45, 52, 26, 67	$[-\infty, +\infty]$	$[0, 0]$	$[0, 0]$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
45, 52, 26, 67	$[-\infty, +\infty]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$\emptyset$	$\emptyset$	$\emptyset$
52, 26, 67	$[-\infty, +\infty]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$\emptyset$	$\emptyset$
23, 26, 67	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$\emptyset$	$\emptyset$
34, 26, 67	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, 0]$	$[0, 0]$	$\emptyset$	$\emptyset$
45, 26, 67	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, 0]$	$\emptyset$	$\emptyset$

# Conditional Interval Analysis

## An Example

### Example 8.7 (Interval analysis for array index; cf. Example 6.7)



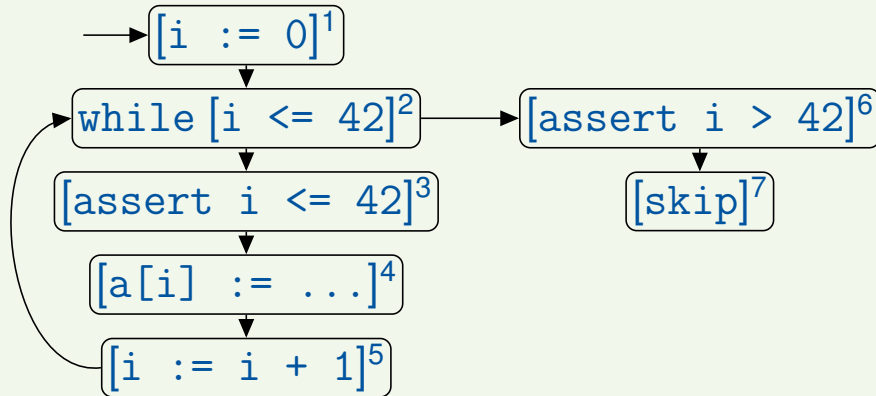
$$\begin{aligned} \varphi_1(J) &= [0, 0] \\ \varphi_2(J) &= J \\ \varphi_3(J) &= J \cap [-\infty, 42] \\ \varphi_4(J) &= J \\ \varphi_5(\emptyset) &= \emptyset \\ \varphi_5([i_1, i_2]) &= [i_1 + 1, i_2 + 1] \\ \varphi_6(J) &= J \cap [43, +\infty] \end{aligned}$$

Worklist w. widening	AI <sub>1</sub>	AI <sub>2</sub>	AI <sub>3</sub>	AI <sub>4</sub>	AI <sub>5</sub>	AI <sub>6</sub>	AI <sub>7</sub>
34, 45, 52, 26, 67	$[-\infty, +\infty]$	$[0, 0]$	$[0, 0]$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
45, 52, 26, 67	$[-\infty, +\infty]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$\emptyset$	$\emptyset$	$\emptyset$
52, 26, 67	$[-\infty, +\infty]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$\emptyset$	$\emptyset$
23, 26, 67	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$\emptyset$	$\emptyset$
34, 26, 67	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, 0]$	$[0, 0]$	$\emptyset$	$\emptyset$
45, 26, 67	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, 0]$	$\emptyset$	$\emptyset$
52, 26, 67	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$\emptyset$	$\emptyset$

# Conditional Interval Analysis

## An Example

### Example 8.7 (Interval analysis for array index; cf. Example 6.7)



$$\varphi_1(J) = [0, 0]$$

$$\varphi_2(J) = J$$

$$\varphi_3(J) = J \cap [-\infty, 42]$$

$$\varphi_4(J) = J$$

$$\varphi_5(\emptyset) = \emptyset$$

$$\varphi_5([i_1, i_2]) = [i_1 + 1, i_2 + 1]$$

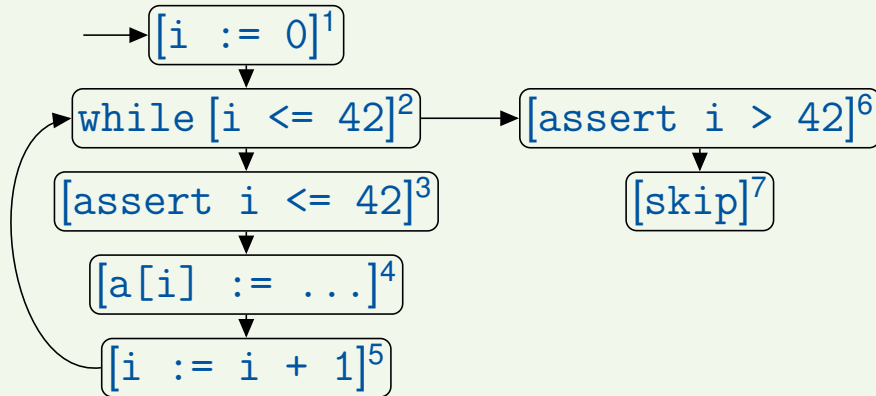
$$\varphi_6(J) = J \cap [43, +\infty]$$

Worklist w. widening	AI <sub>1</sub>	AI <sub>2</sub>	AI <sub>3</sub>	AI <sub>4</sub>	AI <sub>5</sub>	AI <sub>6</sub>	AI <sub>7</sub>
45, 52, 26, 67	$[-\infty, +\infty]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$\emptyset$	$\emptyset$	$\emptyset$
52, 26, 67	$[-\infty, +\infty]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$\emptyset$	$\emptyset$
23, 26, 67	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$\emptyset$	$\emptyset$
34, 26, 67	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, 0]$	$[0, 0]$	$\emptyset$	$\emptyset$
45, 26, 67	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, 0]$	$\emptyset$	$\emptyset$
52, 26, 67	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$\emptyset$	$\emptyset$
26, 67	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$\emptyset$	$\emptyset$

# Conditional Interval Analysis

## An Example

### Example 8.7 (Interval analysis for array index; cf. Example 6.7)



$$\varphi_1(J) = [0, 0]$$

$$\varphi_2(J) = J$$

$$\varphi_3(J) = J \cap [-\infty, 42]$$

$$\varphi_4(J) = J$$

$$\varphi_5(\emptyset) = \emptyset$$

$$\varphi_5([i_1, i_2]) = [i_1 + 1, i_2 + 1]$$

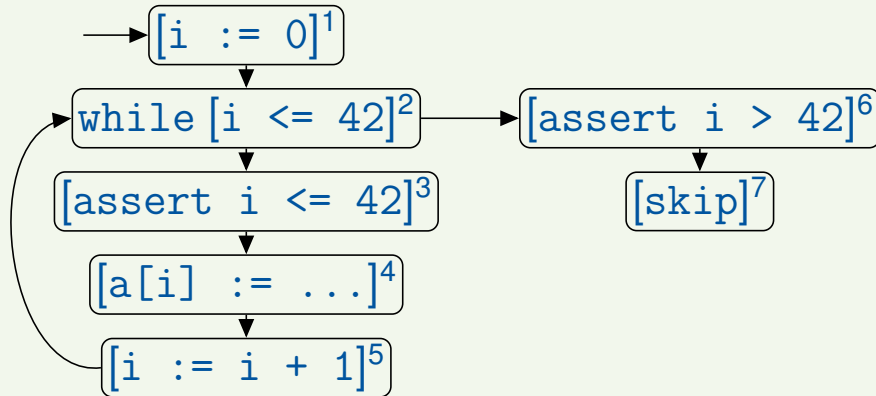
$$\varphi_6(J) = J \cap [43, +\infty]$$

Worklist w. widening	AI <sub>1</sub>	AI <sub>2</sub>	AI <sub>3</sub>	AI <sub>4</sub>	AI <sub>5</sub>	AI <sub>6</sub>	AI <sub>7</sub>
52, 26, 67	$[-\infty, +\infty]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$\emptyset$	$\emptyset$
23, 26, 67	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$\emptyset$	$\emptyset$
34, 26, 67	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, 0]$	$[0, 0]$	$\emptyset$	$\emptyset$
45, 26, 67	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, 0]$	$\emptyset$	$\emptyset$
52, 26, 67	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$\emptyset$	$\emptyset$
26, 67	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$\emptyset$	$\emptyset$
67	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$\emptyset$

# Conditional Interval Analysis

## An Example

### Example 8.7 (Interval analysis for array index; cf. Example 6.7)



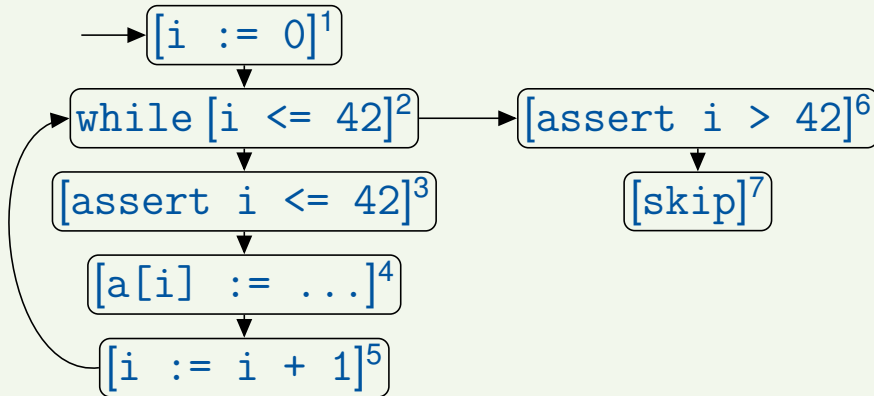
$$\begin{aligned} \varphi_1(J) &= [0, 0] \\ \varphi_2(J) &= J \\ \varphi_3(J) &= J \cap [-\infty, 42] \\ \varphi_4(J) &= J \\ \varphi_5(\emptyset) &= \emptyset \\ \varphi_5([i_1, i_2]) &= [i_1 + 1, i_2 + 1] \\ \varphi_6(J) &= J \cap [43, +\infty] \end{aligned}$$

Worklist w. widening	AI <sub>1</sub>	AI <sub>2</sub>	AI <sub>3</sub>	AI <sub>4</sub>	AI <sub>5</sub>	AI <sub>6</sub>	AI <sub>7</sub>
23, 26, 67	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, 0]$	$[0, 0]$	$[0, 0]$	$\emptyset$	$\emptyset$
34, 26, 67	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, 0]$	$[0, 0]$	$\emptyset$	$\emptyset$
45, 26, 67	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, 0]$	$\emptyset$	$\emptyset$
52, 26, 67	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$\emptyset$	$\emptyset$
26, 67	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$\emptyset$	$\emptyset$
67	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$\emptyset$
$\epsilon$	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[43, +\infty]$

# Conditional Interval Analysis

## An Example (continued)

### Example 8.7 (Interval analysis for array index; continued)



$$\varphi_1(J) = [0, 0]$$

$$\varphi_2(J) = J$$

$$\varphi_3(J) = J \cap [-\infty, 42]$$

$$\varphi_4(J) = J$$

$$\varphi_5(\emptyset) = \emptyset$$

$$\varphi_5([i_1, i_2]) = [i_1 + 1, i_2 + 1]$$

$$\varphi_6(J) = J \cap [43, +\infty]$$

**Narrowing**

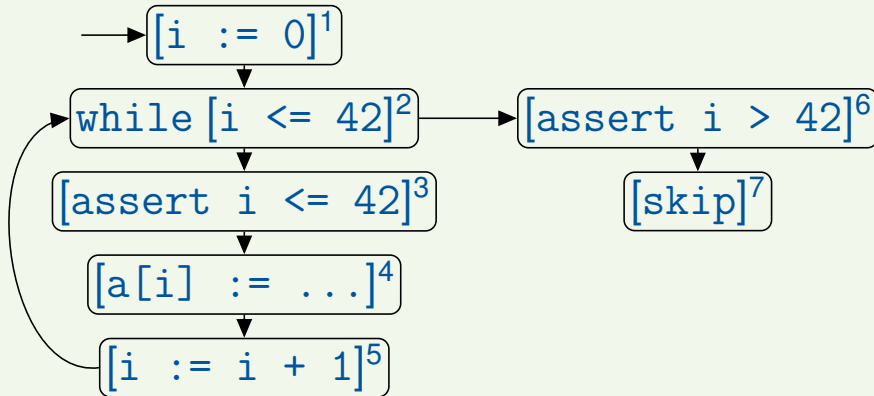
$\text{fix}^\nabla(\Phi_S)$

	$AI_1$	$AI_2$	$AI_3$	$AI_4$	$AI_5$	$AI_6$	$AI_7$
	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[43, +\infty]$

# Conditional Interval Analysis

## An Example (continued)

### Example 8.7 (Interval analysis for array index; continued)



$$\varphi_1(J) = [0, 0]$$

$$\varphi_2(J) = J$$

$$\varphi_3(J) = J \cap [-\infty, 42]$$

$$\varphi_4(J) = J$$

$$\varphi_5(\emptyset) = \emptyset$$

$$\varphi_5([i_1, i_2]) = [i_1 + 1, i_2 + 1]$$

$$\varphi_6(J) = J \cap [43, +\infty]$$

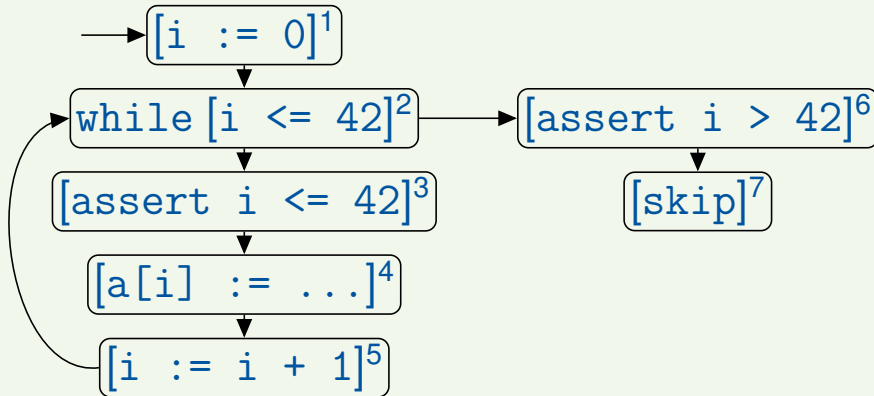
Narrowing	Al <sub>1</sub>	Al <sub>2</sub>	Al <sub>3</sub>	Al <sub>4</sub>	Al <sub>5</sub>	Al <sub>6</sub>	Al <sub>7</sub>
$\text{fix}^\nabla(\Phi_S)$	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[43, +\infty]$
$\Phi_S(\text{fix}^\nabla(\Phi_S))$	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, 42]$	$[0, +\infty]$	$[0, +\infty]$	$[43, +\infty]$



# Conditional Interval Analysis

## An Example (continued)

### Example 8.7 (Interval analysis for array index; continued)



$$\varphi_1(J) = [0, 0]$$

$$\varphi_2(J) = J$$

$$\varphi_3(J) = J \cap [-\infty, 42]$$

$$\varphi_4(J) = J$$

$$\varphi_5(\emptyset) = \emptyset$$

$$\varphi_5([i_1, i_2]) = [i_1 + 1, i_2 + 1]$$

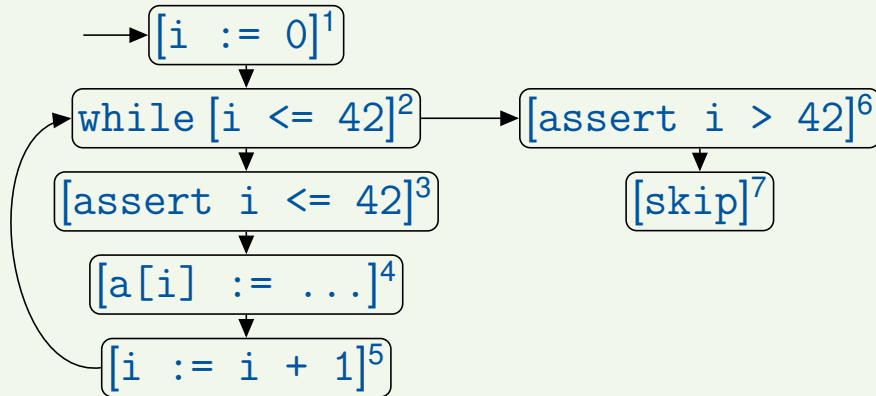
$$\varphi_6(J) = J \cap [43, +\infty]$$

Narrowing	Al <sub>1</sub>	Al <sub>2</sub>	Al <sub>3</sub>	Al <sub>4</sub>	Al <sub>5</sub>	Al <sub>6</sub>	Al <sub>7</sub>
$\text{fix}^\nabla(\Phi_S)$	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[43, +\infty]$
$\Phi_S(\text{fix}^\nabla(\Phi_S))$	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, 42]$	$[0, +\infty]$	$[0, +\infty]$	$[43, +\infty]$
$\Phi_S^2(\text{fix}^\nabla(\Phi_S))$	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, 42]$	$[0, 42]$	$[0, +\infty]$	$[43, +\infty]$

# Conditional Interval Analysis

## An Example (continued)

### Example 8.7 (Interval analysis for array index; continued)



$$\varphi_1(J) = [0, 0]$$

$$\varphi_2(J) = J$$

$$\varphi_3(J) = J \cap [-\infty, 42]$$

$$\varphi_4(J) = J$$

$$\varphi_5(\emptyset) = \emptyset$$

$$\varphi_5([i_1, i_2]) = [i_1 + 1, i_2 + 1]$$

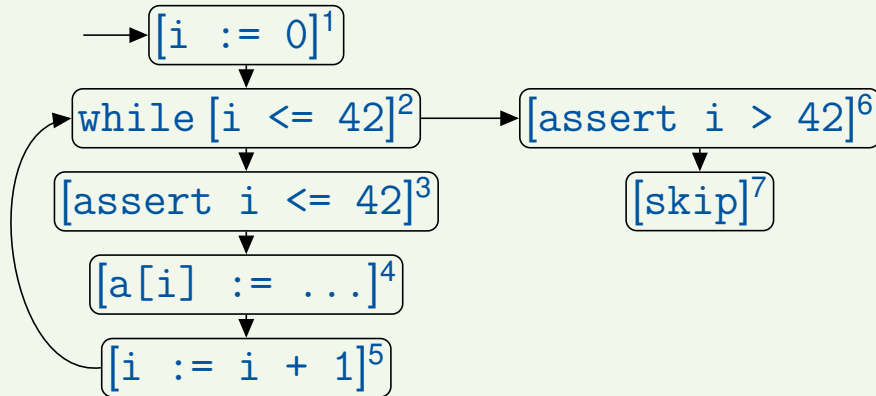
$$\varphi_6(J) = J \cap [43, +\infty]$$

Narrowing	Al <sub>1</sub>	Al <sub>2</sub>	Al <sub>3</sub>	Al <sub>4</sub>	Al <sub>5</sub>	Al <sub>6</sub>	Al <sub>7</sub>
$\text{fix}^\nabla(\Phi_S)$	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[43, +\infty]$
$\Phi_S(\text{fix}^\nabla(\Phi_S))$	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, 42]$	$[0, +\infty]$	$[0, +\infty]$	$[43, +\infty]$
$\Phi_S^2(\text{fix}^\nabla(\Phi_S))$	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, 42]$	$[0, 42]$	$[0, +\infty]$	$[43, +\infty]$
$\Phi_S^3(\text{fix}^\nabla(\Phi_S))$	$[-\infty, +\infty]$	$[0, 43]$	$[0, +\infty]$	$[0, 42]$	$[0, 42]$	$[0, +\infty]$	$[43, +\infty]$

# Conditional Interval Analysis

## An Example (continued)

### Example 8.7 (Interval analysis for array index; continued)



$$\varphi_1(J) = [0, 0]$$

$$\varphi_2(J) = J$$

$$\varphi_3(J) = J \cap [-\infty, 42]$$

$$\varphi_4(J) = J$$

$$\varphi_5(\emptyset) = \emptyset$$

$$\varphi_5([i_1, i_2]) = [i_1 + 1, i_2 + 1]$$

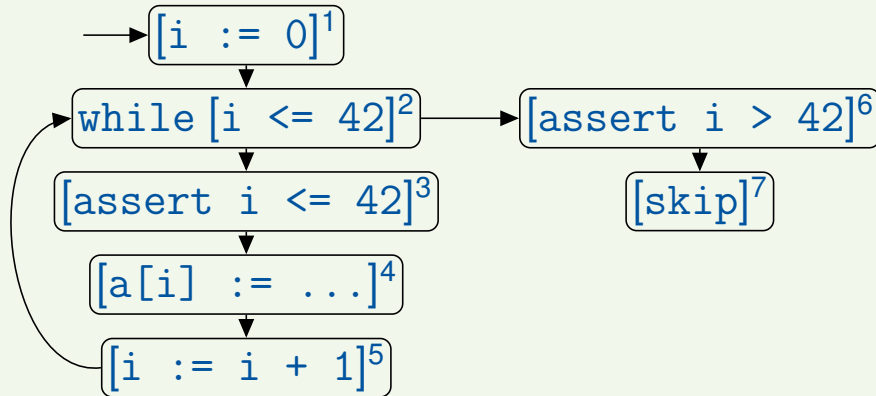
$$\varphi_6(J) = J \cap [43, +\infty]$$

Narrowing	Al <sub>1</sub>	Al <sub>2</sub>	Al <sub>3</sub>	Al <sub>4</sub>	Al <sub>5</sub>	Al <sub>6</sub>	Al <sub>7</sub>
$\text{fix}^\nabla(\Phi_S)$	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[43, +\infty]$
$\Phi_S(\text{fix}^\nabla(\Phi_S))$	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, 42]$	$[0, +\infty]$	$[0, +\infty]$	$[43, +\infty]$
$\Phi_S^2(\text{fix}^\nabla(\Phi_S))$	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, 42]$	$[0, 42]$	$[0, +\infty]$	$[43, +\infty]$
$\Phi_S^3(\text{fix}^\nabla(\Phi_S))$	$[-\infty, +\infty]$	$[0, 43]$	$[0, +\infty]$	$[0, 42]$	$[0, 42]$	$[0, +\infty]$	$[43, +\infty]$
$\Phi_S^4(\text{fix}^\nabla(\Phi_S))$	$[-\infty, +\infty]$	$[0, 43]$	$[0, 43]$	$[0, 42]$	$[0, 42]$	$[0, 43]$	$[43, +\infty]$

# Conditional Interval Analysis

## An Example (continued)

### Example 8.7 (Interval analysis for array index; continued)



$$\varphi_1(J) = [0, 0]$$

$$\varphi_2(J) = J$$

$$\varphi_3(J) = J \cap [-\infty, 42]$$

$$\varphi_4(J) = J$$

$$\varphi_5(\emptyset) = \emptyset$$

$$\varphi_5([i_1, i_2]) = [i_1 + 1, i_2 + 1]$$

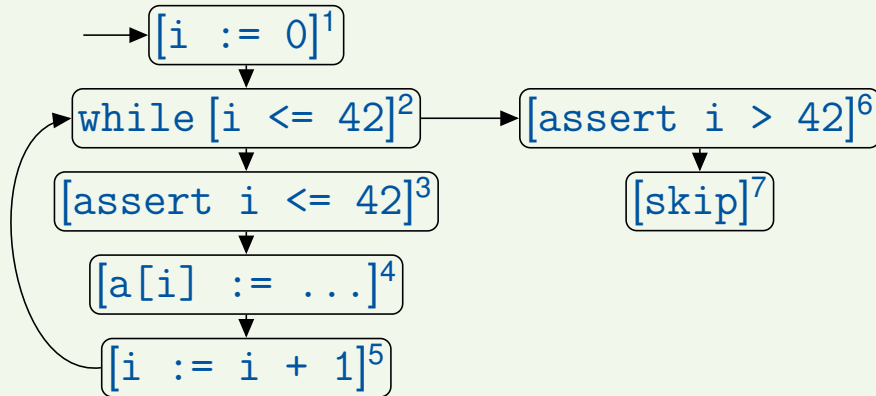
$$\varphi_6(J) = J \cap [43, +\infty]$$

Narrowing	Al <sub>1</sub>	Al <sub>2</sub>	Al <sub>3</sub>	Al <sub>4</sub>	Al <sub>5</sub>	Al <sub>6</sub>	Al <sub>7</sub>
$\text{fix}^\nabla(\Phi_S)$	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[43, +\infty]$
$\Phi_S(\text{fix}^\nabla(\Phi_S))$	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, 42]$	$[0, +\infty]$	$[0, +\infty]$	$[43, +\infty]$
$\Phi_S^2(\text{fix}^\nabla(\Phi_S))$	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, 42]$	$[0, 42]$	$[0, +\infty]$	$[43, +\infty]$
$\Phi_S^3(\text{fix}^\nabla(\Phi_S))$	$[-\infty, +\infty]$	$[0, 43]$	$[0, +\infty]$	$[0, 42]$	$[0, 42]$	$[0, +\infty]$	$[43, +\infty]$
$\Phi_S^4(\text{fix}^\nabla(\Phi_S))$	$[-\infty, +\infty]$	$[0, 43]$	$[0, 43]$	$[0, 42]$	$[0, 42]$	$[0, 43]$	$[43, +\infty]$
$\Phi_S^5(\text{fix}^\nabla(\Phi_S))$	$[-\infty, +\infty]$	$[0, 43]$	$[0, 43]$	$[0, 42]$	$[0, 42]$	$[0, 43]$	$[43, 43]$

# Conditional Interval Analysis

## An Example (continued)

### Example 8.7 (Interval analysis for array index; continued)



$$\varphi_1(J) = [0, 0]$$

$$\varphi_2(J) = J$$

$$\varphi_3(J) = J \cap [-\infty, 42]$$

$$\varphi_4(J) = J$$

$$\varphi_5(\emptyset) = \emptyset$$

$$\varphi_5([i_1, i_2]) = [i_1 + 1, i_2 + 1]$$

$$\varphi_6(J) = J \cap [43, +\infty]$$

Narrowing	Al <sub>1</sub>	Al <sub>2</sub>	Al <sub>3</sub>	Al <sub>4</sub>	Al <sub>5</sub>	Al <sub>6</sub>	Al <sub>7</sub>
$\text{fix}^\nabla(\Phi_S)$	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[43, +\infty]$
$\Phi_S(\text{fix}^\nabla(\Phi_S))$	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, 42]$	$[0, +\infty]$	$[0, +\infty]$	$[43, +\infty]$
$\Phi_S^2(\text{fix}^\nabla(\Phi_S))$	$[-\infty, +\infty]$	$[0, +\infty]$	$[0, +\infty]$	$[0, 42]$	$[0, 42]$	$[0, +\infty]$	$[43, +\infty]$
$\Phi_S^3(\text{fix}^\nabla(\Phi_S))$	$[-\infty, +\infty]$	$[0, 43]$	$[0, +\infty]$	$[0, 42]$	$[0, 42]$	$[0, +\infty]$	$[43, +\infty]$
$\Phi_S^4(\text{fix}^\nabla(\Phi_S))$	$[-\infty, +\infty]$	$[0, 43]$	$[0, 43]$	$[0, 42]$	$[0, 42]$	$[0, 43]$	$[43, +\infty]$
$\Phi_S^5(\text{fix}^\nabla(\Phi_S))$	$[-\infty, +\infty]$	$[0, 43]$	$[0, 43]$	$[0, 42]$	$[0, 42]$	$[0, 43]$	$[43, 43]$
$\Phi_S^6(\text{fix}^\nabla(\Phi_S))$	$[-\infty, +\infty]$	$[0, 43]$	$[0, 43]$	$[0, 42]$	$[0, 42]$	$[0, 43]$	$[43, 43]$

# Optimisation in the GNU C Compiler

---

## Outline of Lecture 8

Recap: Widening and Narrowing

Taking Conditional Branches into Account

Conditional Constant Propagation Analysis

Conditional Interval Analysis

Optimisation in the GNU C Compiler

# Optimisation in the GNU C Compiler

---

## Live Variables

Source:

```
int f () {  
    int a;  
    a = 1;  
    a = 2;  
    return a;  
}
```

# Optimisation in the GNU C Compiler

---

## Live Variables

Source:

```
int f() {  
    int a;  
    a = 1;  
    a = 2;  
    return a;  
}
```

W/o optimisation:

```
...  
_f:                                     ...  
## @f  
                                     .cfi_startproc  
## BB#0:  
    pushq   %rbp  
Ltmp0:                                     ...  
                                     .cfi_def_cfa_offset 16  
Ltmp1:                                     ...  
                                     .cfi_offset %rbp, 16  
    movq    %rsp, %rbp  
Ltmp2:                                     ...  
                                     .cfi_def_cfa_register %rbp  
    movl   $1, 4(%rbp)  
    movl   $2, 4(%rbp)  
    movl   4(%rbp), %eax  
    popq   %rbp  
    retq  
                                     .cfi_endproc  
...
```

With -O3:

```
...  
_f:                                     ...  
## @f  
                                     .cfi_startproc  
## BB#0:  
    pushq   %rbp  
Ltmp0:                                     ...  
                                     .cfi_def_cfa_offset 16  
Ltmp1:                                     ...  
                                     .cfi_offset %rbp, 16  
    movq    %rsp, %rbp  
Ltmp2:                                     ...  
                                     .cfi_def_cfa_register %rbp  
    movl   $2, %eax  
    popq   %rbp  
    retq  
                                     .cfi_endproc  
...
```



## Available Expressions

Source:

```
int f(int a, int b){
    int c;
    c = a*b;
    if (c > 0)
        {return a*b+1;}
    else
        {return a*b+2;}
}
```

# Optimisation in the GNU C Compiler

## Available Expressions

Source:

```
int f(int a,int b){
  int c;
  c = a*b;
  if (c > 0)
    {return a*b+1;}
  else
    {return a*b+2;}
}
```

W/o optimisation:

```
_f:      ...                ## @f
        .cfi_startproc
## BB#0:
        pushq   %rbp
Ltmp0:   .cfi_def_cfa_offset 16
Ltmp1:   .cfi_offset %rbp, 16
        movq    %rsp, %rbp
Ltmp2:   .cfi_def_cfa_register %rbp
        movl   %edi, 8(%rbp)
        movl   %esi, 12(%rbp)
        movl   8(%rbp), %esi
        imull  12(%rbp), %esi ; a*b
        movl   %esi, 16(%rbp) ; store c
        cmpl   $0, 16(%rbp) ; c > 0?
        jle   LBB0_2
## BB#1:
        movl   8(%rbp), %eax
        imull  12(%rbp), %eax ; a*b
        addl   $1, %eax ; add 1
        movl   %eax, 4(%rbp)
        jmp   LBB0_3
LBB0_2:
        movl   8(%rbp), %eax
        imull  12(%rbp), %eax ; a*b
        addl   $2, %eax ; add 2
        movl   %eax, 4(%rbp)
LBB0_3:
        movl   4(%rbp), %eax
        popq   %rbp
        retq
        .cfi_endproc
        ...
```

With -O3:

```
_f:      ...                ## @f
        .cfi_startproc
## BB#0:
        pushq   %rbp
Ltmp0:   .cfi_def_cfa_offset 16
Ltmp1:   .cfi_offset %rbp, 16
        movq    %rsp, %rbp
Ltmp2:   .cfi_def_cfa_register %rbp
        imull  %esi, %edi ; a*b
        testl  %edi, %edi ; a*b = 0?
        setle %al
        movzbl %al, %eax
        leal   1(%rax,%rdi), %eax
        popq   %rbp
        retq
        .cfi_endproc
        ...
```