By means of $\bot$ we construct the complete lattice:

$$\mathbb{D}_1 = (\text{Vars} \rightarrow \downarrow) \bot$$

**Description Relation:**

$$\rho \triangle D \iff D \neq \bot \land \forall x \in \text{Vars}: (\rho x) \triangle (D x)$$

The abstract evaluation of expressions is defined analogously to constant propagation. We have:

$$([e] \rho) \triangle ([e] D) \quad \text{whenever} \quad \rho \triangle D$$

Example:

$$\begin{align*}
[1, 2] <^d [9, 42] &= \langle 1, 1 \rangle \\
[0, 7] <^d [0, 7] &= \langle 0, 1 \rangle \\
[3, 4] <^d [1, 2] &= \langle 0, 0 \rangle
\end{align*}$$

The Effects of Edges:

$$\begin{align*}
[i] D &= D \\
[x = e]^D D &= D \uplus \{ x \mapsto [e]^D D \} \\
[x = M[e]]^D D &= D \uplus \{ x \mapsto \top \} \\
[M[e_1] = e_2]^D D &= D \\
[\text{Pos}(e)]^D D &= \begin{cases} \bot & \text{if } [0, 0] \subseteq [e]^D D \\ D & \text{otherwise} \end{cases} \\
[\text{Neg}(e)]^D D &= \begin{cases} D & \text{if } [0, 0] \subseteq [e]^D D \\ \bot & \text{otherwise} \end{cases}
\end{align*}$$

... given that $D \neq \bot$. 
Better Exploitation of Conditions (cont.)

\[ [\text{Neg}(c)]^2 D = \begin{cases} \bot & \text{if } [0, 0] \not\subseteq [c]^2 D \\ D_1 & \text{otherwise} \end{cases} \]

where:

\[ D_1 = \begin{cases} D \oplus \{ x \mapsto (D x) \cap ([e_1]^2 D) \} & \text{if } e \equiv x \neq e_1 \\ D \oplus \{ x \mapsto (D x) \cap \llbracket -\infty, u \rrbracket \} & \text{if } e \equiv x > e_1, [e_1]^2 D = \llbracket -\infty, u \rrbracket \\ D \oplus \{ x \mapsto (D x) \cap [l, \infty) \} & \text{if } e \equiv x < e_1, [e_1]^2 D = [l, \infty) \end{cases} \]

Better Exploitation of Conditions

\[ [\text{Pos}(c)]^2 D = \begin{cases} \bot & \text{if } [0, 0] = [c]^2 D \\ D_1 & \text{otherwise} \end{cases} \]

where:

\[ D_1 = \begin{cases} D \oplus \{ x \mapsto (D x) \cap ([e_1]^2 D) \} & \text{if } e \equiv x = e_1 \\ D \oplus \{ x \mapsto (D x) \cap \llbracket -\infty, u \rrbracket \} & \text{if } e \equiv x \leq e_1, [e_1]^2 D = \llbracket -\infty, u \rrbracket \\ D \oplus \{ x \mapsto (D x) \cap [l, \infty) \} & \text{if } e \equiv x \geq e_1, [e_1]^2 D = [l, \infty) \end{cases} \]

Example

\[ \llbracket 0, \infty \rrbracket \cap \llbracket 0, 42 \rrbracket = \llbracket 0, 42 \rrbracket \]

<table>
<thead>
<tr>
<th>( i )</th>
<th>( t )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>( -\infty )</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>( -\infty )</td>
</tr>
<tr>
<td>8</td>
<td>42</td>
</tr>
</tbody>
</table>
Problem

→ The solution can be computed with RR-iteration —
   after about 42 rounds !

→ On some programs, iteration may never terminate ...

Idea 1  Widening

• Accelerate the iteration — at the prize of imprecision.
• Allow only a bounded number of modifications of values !!!

... in the Example

• dis-allow updates of interval bounds in \( \mathbb{Z} \ ...

\[ [3, 17] \subseteq [3, +\infty] \subseteq [-\infty, +\infty] \]

\[ 332 \]

(c) The sequence \( G^k \downarrow \), \( k \geq 0 \), is an ascending chain:

\[ \bot \subseteq G \subseteq \ldots \subseteq G^k \subseteq \ldots \]

(d) If \( G^k \downarrow = G^{k+1} \downarrow = y \), then \( y \) is a solution of (1).

(e) If \( \mathbb{D} \) has infinite strictly ascending chains, then (d) is not yet sufficient ...

but: we could consider the modified system of equations:

\[ x_i = x_i \downarrow \sqcup f_i(x_1, \ldots, x_n), \quad i = 1, \ldots, n \]  \quad (3)

for a binary operation widening:

\[ \downarrow \sqcup : \mathbb{D} \rightarrow \mathbb{D} \quad \text{with} \quad v_1 \sqcup v_2 \subseteq v_1 \sqcup v_2 \]

(RR)-iteration for (3) will still compute a solution of (1).

\[ 334 \]

Formalization of the Approach

Let \( x_i \sqsubseteq f_i(x_1, \ldots, x_n), \quad i = 1, \ldots, n \) \quad (1)

denote a system of constraints over \( \mathbb{D} \) where the \( f_i \) are not necessarily monotonic.

Nonetheless, an accumulating iteration can be defined. Consider the system of equations:

\[ x_i = x_i \sqcup f_i(x_1, \ldots, x_n), \quad i = 1, \ldots, n \]  \quad (2)

We obviously have:

(a) \( \mathbf{x} \) is a solution of (1) iff \( \mathbf{x} \) is a solution of (2).

(b) The function \( G : \mathbb{D}^n \rightarrow \mathbb{D}^n \) with

\[ G(x_1, \ldots, x_n) = (y_1, \ldots, y_n), \quad y_i = x_i \sqcup f_i(x_1, \ldots, x_n) \]

is increasing, i.e., \( \mathbf{x} \subseteq G \mathbf{x} \) for all \( \mathbf{x} \in \mathbb{D}^n \).

\[ 333 \]

... for Interval Analysis:

• The complete lattice is: \( \mathbb{D}_1 = (\text{Vars} \rightarrow \mathbb{I})_\bot \)

• the widening \( \sqcup \) is defined by:

\[ \bot \sqcup D = D \sqcup \bot = D \quad \text{and for} \quad D_1 \neq \bot \neq D_2: \]

\[ (D_1 \sqcup D_2) \mathbf{x} = (D_1 \mathbf{x}) \sqcup (D_2 \mathbf{x}) \quad \text{where} \]

\[ l_1, u_1 \sqsubseteq [l_2, u_2] \quad \text{with} \]

\[ l = \begin{cases} l_1 & \text{if } l_1 \leq l_2 \\ \infty & \text{otherwise} \end{cases} \]

\[ u = \begin{cases} u_1 & \text{if } u_1 \geq u_2 \\ +\infty & \text{otherwise} \end{cases} \]

\[ \Rightarrow \quad \sqcup \quad \text{is not commutative} !!! \]

\[ 335 \]
Example

\[
[0,2] \sqcup [1,2] = [0,2]
\]
\[
[1,2] \sqcup [0,2] = [-\infty,2]
\]
\[
[1,5] \sqcup [3,7] = [1,\infty]
\]

→ Widening returns larger values more quickly.
→ It should be constructed in such a way that termination of iteration is guaranteed.
→ For interval analysis, widening bounds the number of iterations by:

\[\text{#points} \cdot (1 + 2 \cdot \text{#Vars})\]

Conclusion

- In order to determine a solution of (1) over a complete lattice with infinite ascending chains, we define a suitable widening and then solve (3).
- Caveat The construction of suitable widenings is a dark art!!!
  
  Often \(\sqcup\) is chosen dynamically during iteration such that
  → the abstract values do not get too complicated;
  → the number of updates remains bounded ...

Our Example

| 1 | i = 0;
|---|---
| 0 | i = 0;
| 1 | i = 0;
| 2 | A := A + i;
| 3 | M[A1] = i;
| 4 | i := i + 1;
| 5 | neg[i < 42] Pos[i < 42]
| 6 | neg[0 \leq i < 42] Pos[0 \leq i < 42]

Our Example

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>i = 0;</td>
<td>0 \infty \infty \infty</td>
<td>0 \infty \infty \infty</td>
</tr>
<tr>
<td>0 \infty \infty \infty</td>
<td>0 \infty \infty \infty</td>
<td></td>
</tr>
<tr>
<td>1 0 0 0</td>
<td>1 0 0 0</td>
<td></td>
</tr>
<tr>
<td>2 0 0 0</td>
<td>2 0 0 0</td>
<td></td>
</tr>
<tr>
<td>3 0 0 0</td>
<td>3 0 0 0</td>
<td></td>
</tr>
<tr>
<td>4 0 0 0</td>
<td>4 0 0 0</td>
<td></td>
</tr>
<tr>
<td>5 0 0 0</td>
<td>5 0 0 0</td>
<td></td>
</tr>
<tr>
<td>6 1 1 \infty</td>
<td>6 1 1 \infty</td>
<td></td>
</tr>
<tr>
<td>7 \perp 42 \infty</td>
<td>7 \perp 42 \infty</td>
<td></td>
</tr>
<tr>
<td>8 \perp 42 \infty</td>
<td>8 \perp 42 \infty</td>
<td></td>
</tr>
</tbody>
</table>

\([0,0] \sqcup (20,3) = \boxed{20,3}\)
... obviously, the result is disappointing!

Idea 2

In fact, acceleration with \( \sqcup \) need only be applied at sufficiently many places!

A set \( I \) is a loop separator, if every loop contains at least one point from \( I \).

If we apply widening only at program points from such a set \( I \), then RR-iteration still terminates !!!
The Analysis with $I = \{1\}$:

\[
\begin{align*}
&\text{Neg}(i \leq 42) \quad \text{Pos}(0 \leq i < 42) \\
&\text{Neg}(0 \leq i < 42) \quad \text{Pos}(0 \leq i < 42)
\end{align*}
\]

\[
\begin{array}{c|c|c|c|c|c|c}
1 & 2 & 3 & & & & \\
\hline
l & u & l & u & l & u \\
\hline
0 & +\infty & +\infty & +\infty & +\infty & +\infty \\
1 & 0 & 0 & 0 & +\infty & +\infty \\
2 & 0 & 0 & 0 & 41 & +\infty \\
3 & 0 & 0 & 0 & 41 & +\infty \\
4 & 0 & 0 & 0 & 41 & +\infty \\
5 & 0 & 0 & 0 & 41 & +\infty \\
6 & 1 & 1 & 1 & 42 & +\infty \\
7 & \bot & \bot & \bot & \bot & +\infty \\
8 & \bot & 42 & +\infty & +\infty & +\infty \\
\end{array}
\]

The Analysis with $I = \{2\}$:

\[
\begin{align*}
&\text{Neg}(i \leq 42) \quad \text{Pos}(i < 42) \\
&\text{Neg}(0 \leq i < 42) \quad \text{Pos}(0 \leq i < 42)
\end{align*}
\]

\[
\begin{array}{c|c|c|c|c|c|c}
1 & 2 & 3 & & & & \\
\hline
l & u & l & u & l & u \\
\hline
0 & +\infty & +\infty & +\infty & +\infty & +\infty \\
1 & 0 & 0 & 1 & 0 & 42 \\
2 & 0 & 0 & 0 & +\infty & 0 & +\infty \\
3 & 0 & 0 & 0 & 41 & 0 & 41 \\
4 & 0 & 0 & 0 & 41 & 0 & 41 \\
5 & 0 & 0 & 0 & 41 & 0 & 41 \\
6 & 1 & 1 & 1 & 42 & 1 & 42 \\
7 & \bot & 42 & +\infty & 42 & +\infty \\
8 & \bot & \bot & \bot & 42 & 42 \\
\end{array}
\]
Discussion

- Both runs of the analysis determine interesting information.
- The run with \( J = \{2\} \) proves that always \( i = 42 \) after leaving the loop.
- Only the run with \( J = \{1\} \) finds, however, that the outer check makes the inner check superfluous!

How can we find a suitable loop separator \( I \)???

---

**Idea 3:** Narrowing

Let \( \bar{x} \) denote any solution of (1), i.e., \( x_i \not\in f_i \bar{x} \), \( i = 1, \ldots, n \)

Then for monotonic \( f_i \),

\[ \bar{x} \not\in F \bar{x} \not\in F^2 \bar{x} \not\in \cdots \not\in F^k \bar{x} \not\in \cdots \]

\[ \text{// Narrowing Iteration} \]

Every tuple \( F^k \bar{x} \) is a solution of (1).

Termination is no problem anymore: we stop whenever we want!

\[ \text{// The same also holds for RR-iteration.} \]
Narrowing Iteration in the Example

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>l</td>
<td>u</td>
</tr>
<tr>
<td>l</td>
<td>u</td>
</tr>
<tr>
<td>0</td>
<td>$-\infty$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>42</td>
</tr>
<tr>
<td>8</td>
<td>42</td>
</tr>
</tbody>
</table>

Discussion

→ We start with a safe approximation.
→ We find that the inner check is redundant.
→ We find that at exit from the loop, always $i = 42$.
→ It was not necessary to construct an optimal loop separator!

Final Question

Do we have to accept that narrowing may not terminate ???

4. Idea: Accelerated Narrowing

Assume that we have a solution $x = (x_1, \ldots, x_n)$ of the system of constraints:

$$x_i \supseteq f_i(x_1, \ldots, x_n), \quad i = 1, \ldots, n$$

(1)

Then consider the system of equations:

$$x_i = x_i \cap f_i(x_1, \ldots, x_n), \quad i = 1, \ldots, n$$

(4)

Obviously, we have for monotonic $f_i : H^{k_1} \subseteq F^{k_2}$ where $H(x_1, \ldots, x_n) = (y_1, \ldots, y_n)$, $y_i = x_i \cap f_i(x_1, \ldots, x_n)$.

In (4), we replace $\cap$ by the novel operator $\nabla$ where:

$$a_1 \nabla a_2 \subseteq a_1 \cap a_2 \subseteq a_1$$

348

349

350

351
Accelerated Narrowing in the Example

\[
\begin{array}{c}
\text{Neg}(i < 42) \quad \text{Pos}(i < 42)
\end{array}
\]

\[
\begin{array}{c}
\text{Neg}(0 \leq i < 42) \quad \text{Pos}(0 \leq i < 42)
\end{array}
\]

\[
A_i = A + i;
\]

\[
M[A_i] = i;
\]

\[
i = i + 1;
\]

\[
\begin{array}{c|c|c|c|c|c|c|c|c|c|c}
& 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
\hline
l & -\infty & +\infty & -\infty & +\infty & -\infty & +\infty & 0 & +\infty & 0 & +\infty & 0 & 42 \\
\hline
u & -\infty & +\infty & -\infty & +\infty & -\infty & +\infty & 0 & +\infty & 0 & +\infty & 0 & 42 \\
\hline
\end{array}
\]

\[
\begin{array}{c}
\text{Neg}(i < 42) \quad \text{Pos}(i < 42)
\end{array}
\]

\[
\begin{array}{c}
\text{Neg}(0 \leq i < 42) \quad \text{Pos}(0 \leq i < 42)
\end{array}
\]

\[
A_i = A + i;
\]

\[
M[A_i] = i;
\]

\[
i = i + 1;
\]

\[
\begin{array}{c|c|c|c|c|c|c|c|c|c|c}
& 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
\hline
l & -\infty & +\infty & -\infty & +\infty & -\infty & +\infty & 0 & +\infty & 0 & +\infty & 0 & 42 \\
\hline
u & -\infty & +\infty & -\infty & +\infty & -\infty & +\infty & 0 & +\infty & 0 & +\infty & 0 & 42 \\
\hline
\end{array}
\]

... for Interval Analysis

We preserve finite interval bounds.

Therefore, \( \bot \cap D = D \cap \bot = \bot \) and for \( D_1 \neq \bot \neq D_2 \):

\[
(D_1 \cap D_2)x = (D_1x) \cap (D_2x)
\]

where

\[
[l_1, u_1] \cap [l_2, u_2] = [l, u]
\]

with

\[
l = \begin{cases} l_2 & \text{if } l_1 = -\infty \\ l_1 & \text{otherwise} \end{cases}
\]

\[
u = \begin{cases} u_2 & \text{if } u_1 = \infty \\ u_1 & \text{otherwise} \end{cases}
\]

\[
\cap \text{ is not commutative} \!
\]

\[
\begin{array}{c}
\text{Neg}(i < 42) \quad \text{Pos}(i < 42)
\end{array}
\]

\[
\begin{array}{c}
\text{Neg}(0 \leq i < 42) \quad \text{Pos}(0 \leq i < 42)
\end{array}
\]

\[
A_i = A + i;
\]

\[
M[A_i] = i;
\]

\[
i = i + 1;
\]

\[
\begin{array}{c|c|c|c|c|c|c|c|c|c|c}
& 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
\hline
l & -\infty & +\infty & -\infty & +\infty & -\infty & +\infty & 0 & +\infty & 0 & +\infty & 0 & 42 \\
\hline
u & -\infty & +\infty & -\infty & +\infty & -\infty & +\infty & 0 & +\infty & 0 & +\infty & 0 & 42 \\
\hline
\end{array}
\]

\[
\begin{array}{c}
\text{Neg}(i < 42) \quad \text{Pos}(i < 42)
\end{array}
\]

\[
\begin{array}{c}
\text{Neg}(0 \leq i < 42) \quad \text{Pos}(0 \leq i < 42)
\end{array}
\]

\[
A_i = A + i;
\]

\[
M[A_i] = i;
\]

\[
i = i + 1;
\]

\[
\begin{array}{c|c|c|c|c|c|c|c|c|c|c}
& 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
\hline
l & -\infty & +\infty & -\infty & +\infty & -\infty & +\infty & 0 & +\infty & 0 & +\infty & 0 & 42 \\
\hline
u & -\infty & +\infty & -\infty & +\infty & -\infty & +\infty & 0 & +\infty & 0 & +\infty & 0 & 42 \\
\hline
\end{array}
\]
Discussion

→ **Caveat:** Widening also returns for non-monotonic $f_i$ a solution. Narrowing is only applicable to monotonic $f_i$. ⚠️

→ In the example, accelerated narrowing already returns the optimal result.

→ If the operator $\cap$ only allows for finitely many improvements of values, we may execute narrowing until stabilization.

→ In case of interval analysis these are at most:

$$\#points \cdot (1 + 2 \cdot \#Vars)$$

---

1.6 **Pointer Analysis**

**Questions**

→ Are two addresses possibly equal?

→ Are two addresses definitively equal?

Discussion

→ **Caveat:** Widening also returns for non-monotonic $f_i$ a solution. Narrowing is only applicable to monotonic $f_i$. ⚠️

→ In the example, accelerated narrowing already returns the optimal result.

→ If the operator $\cap$ only allows for finitely many improvements of values, we may execute narrowing until stabilization.

→ In case of interval analysis these are at most:

$$\#points \cdot (1 + 2 \cdot \#Vars)$$
1.6 Pointer Analysis

Questions

- Are two addresses possibly equal?
- Are two addresses definitively equal?