Theorem:

If $f_1 : \mathbb{D}_1 \rightarrow \mathbb{D}_2$ and $f_2 : \mathbb{D}_2 \rightarrow \mathbb{D}_3$ are monotonic, then also $f_2 \circ f_1 : \mathbb{D}_1 \rightarrow \mathbb{D}_3$.

Theorem:

If $\mathbb{D}_2$ is a complete lattice, then the set $[\mathbb{D}_1 \rightarrow \mathbb{D}_2]$ of monotonic functions $f : \mathbb{D}_1 \rightarrow \mathbb{D}_2$ is also a complete lattice where

$f \sqsubseteq g \iff f \, x \sqsubseteq g \, x$ for all $x \in \mathbb{D}_1$.
Theorem:
If $f_1 : D_1 \to D_2$ and $f_2 : D_2 \to D_3$ are monotonic, then also $f_2 \circ f_1 : D_1 \to D_3$.

Theorem:
If $D_2$ is a complete lattice, then the set $[D_1 \to D_2]$ of monotonic functions $f : D_1 \to D_2$ is also a complete lattice where

$$f \sqsubseteq g \quad \text{iff} \quad f x \sqsubseteq g x \quad \text{for all} \quad x \in D_1$$

In particular for $F \subseteq [D_1 \to D_2]$,

$$\bigsqcup F = f \quad \text{mit} \quad f x = \bigsqcup \{g x \mid g \in F\}$$
For functions \( f_i x = a_i \cap x \cup b_i \), the operations “\( \circ \)”, “\( \sqcup \)” and “\( \sqcap \)” can be explicitly defined by:

\[
(f_2 \circ f_1) x = (a_1 \cap a_2) \cap x \cup a_2 \cap b_1 \cup b_2
\]

\[
(f_1 \sqcup f_2) x = (a_1 \cup a_2) \cap x \cup b_1 \cup b_2
\]

\[
(f_1 \sqcap f_2) x = (a_1 \cup b_1) \cap (a_2 \cup b_2) \cap x \cup b_1 \cap b_2
\]
Wanted: minimally small solution for:

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

where all \( f_i : \mathbb{D}^n \rightarrow \mathbb{D} \) are monotonic.
Wanted: minimally small solution for:

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

\[ (\ast) \]

where all \( f_i : \mathbb{D}^n \rightarrow \mathbb{D} \) are monotonic.

Idea:

- Consider \( F : \mathbb{D}^n \rightarrow \mathbb{D}^n \) where

\[ F(x_1, \ldots, x_n) = (y_1, \ldots, y_n) \quad \text{with} \quad y_i = f_i(x_1, \ldots, x_n). \]
Wanted: minimally small solution for:

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

where all \( f_i : \mathbb{D}^n \rightarrow \mathbb{D} \) are monotonic.

Idea:

- Consider \( F : \mathbb{D}^n \rightarrow \mathbb{D}^n \) where

\[ F(x_1, \ldots, x_n) = (y_1, \ldots, y_n) \quad \text{with} \quad y_i = f_i(x_1, \ldots, x_n). \]

- If all \( f_i \) are monotonic, then also \( F \) :-)

98
Wanted: minimally small solution for:

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

where all \( f_i : \mathbb{D}^n \rightarrow \mathbb{D} \) are monotonic.

Idea:

- Consider \( F : \mathbb{D}^n \rightarrow \mathbb{D}^n \) where
  \[ F(x_1, \ldots, x_n) = (y_1, \ldots, y_n) \quad \text{with} \quad y_i = f_i(x_1, \ldots, x_n). \]
- If all \( f_i \) are monotonic, then also \( F : -) \)
- We successively approximate a solution. We construct:
  \[ \bot, \quad F \bot, \quad F^2 \bot, \quad F^3 \bot, \quad \ldots \]

Hope: We eventually reach a solution ... ???
Example:  \[ \mathbb{D} = 2\{a, b, c\}, \ \subseteq = \subseteq \]

\begin{align*}
x_1 & \supseteq \{a\} \cup x_3 \\
x_2 & \supseteq x_3 \cap \{a, b\} \\
x_3 & \supseteq x_1 \cup \{c\}
\end{align*}
Example: \[ \mathbb{D} = 2\{a, b, c\}, \quad \subseteq = \subseteq \]

\[
\begin{align*}
x_1 & \supseteq \{a\} \cup x_3 \\
x_2 & \supseteq x_3 \cap \{a, b\} \\
x_3 & \supseteq x_1 \cup \{c\}
\end{align*}
\]

The Iteration:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x_1)</td>
<td>(\emptyset)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(x_2)</td>
<td>(\emptyset)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(x_3)</td>
<td>(\emptyset)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example:
\[ \mathcal{D} = 2\{a,b,c\}, \quad \subseteq = \subseteq \]

\[ x_1 \supseteq \{a\} \cup x_3 \]

\[ x_2 \supseteq x_3 \cap \{a, b\} \]

\[ x_3 \supseteq x_1 \cup \{c\} \]

The Iteration:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x_1)</td>
<td>\emptyset</td>
<td>{a}</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(x_2)</td>
<td>\emptyset</td>
<td>\emptyset</td>
<td>\emptyset</td>
<td>\emptyset</td>
<td>\emptyset</td>
</tr>
<tr>
<td>(x_3)</td>
<td>\emptyset</td>
<td>{c}</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example: \( \mathcal{D} = 2^{\{a,b,c\}} \), \( \subseteq = \subseteq \)

\[
\begin{align*}
x_1 & \supseteq \{a\} \cup x_3 \\
x_2 & \supseteq x_3 \cap \{a, b\} \\
x_3 & \supseteq x_1 \cup \{c\}
\end{align*}
\]

The Iteration:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1 )</td>
<td>( \emptyset )</td>
<td>( {a} )</td>
<td>( {a, c} )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( x_2 )</td>
<td>( \emptyset )</td>
<td>( \emptyset )</td>
<td>( \emptyset )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( x_3 )</td>
<td>( \emptyset )</td>
<td>( {c} )</td>
<td>( {a, c} )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example: \[ \mathcal{D} = 2\{a,b,c\}, \quad \sqsubseteq = \subseteq \]

\[
x_1 \supseteq \{a\} \cup x_3
\]
\[
x_2 \supseteq x_3 \cap \{a, b\}
\]
\[
x_3 \supseteq x_1 \cup \{c\}
\]

The Iteration:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x_1)</td>
<td>(\emptyset)</td>
<td>({a})</td>
<td>({a, c})</td>
<td>({a, c})</td>
<td>({a, c})</td>
</tr>
<tr>
<td>(x_2)</td>
<td>(\emptyset)</td>
<td>(\emptyset)</td>
<td>(\emptyset)</td>
<td>(\emptyset)</td>
<td>({a})</td>
</tr>
<tr>
<td>(x_3)</td>
<td>(\emptyset)</td>
<td>({c})</td>
<td>({a, c})</td>
<td>({a, c})</td>
<td>({a, c})</td>
</tr>
</tbody>
</table>
Example:  \[ \mathcal{D} = 2\{a,b,c\}, \quad \sqsubseteq = \subseteq \]

\[ x_1 \supseteq \{a\} \cup x_3 \]
\[ x_2 \supseteq x_3 \cap \{a, b\} \]
\[ x_3 \supseteq x_1 \cup \{c\} \]

The Iteration:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1 )</td>
<td>\emptyset</td>
<td>{a}</td>
<td>{a, c}</td>
<td>{a, c}</td>
<td>dito</td>
</tr>
<tr>
<td>( x_2 )</td>
<td>\emptyset</td>
<td>\emptyset</td>
<td>\emptyset</td>
<td>\emptyset</td>
<td>{a}</td>
</tr>
<tr>
<td>( x_3 )</td>
<td>\emptyset</td>
<td>{c}</td>
<td>{a, c}</td>
<td>{a, c}</td>
<td>{a, c}</td>
</tr>
</tbody>
</table>
Theorem

• $\bot, F \bot, F^2 \bot, \ldots$ form an ascending chain:

$$
\bot \subseteq F \bot \subseteq F^2 \bot \subseteq \ldots
$$

• If $F^k \bot = F^{k+1} \bot$, a solution is obtained which is the least one

• If all ascending chains are finite, such a $k$ always exists.
Theorem

• ⊥, \( F \perp \), \( F^2 \perp \), . . . form an ascending chain:

\[
\perp \subseteq F \perp \subseteq F^2 \perp \subseteq \ldots
\]

• If \( F^k \perp = F^{k+1} \perp \), a solution is obtained which is the least one :-)

• If all ascending chains are finite, such a \( k \) always exists.

Proof

The first claim follows by complete induction:

**Foundation:** \( F^0 \perp = \perp \subseteq F^1 \perp \) :-)}
**Step:** Assume $F^{i-1} \perp \subseteq F^i \perp$. Then

$$F^i \perp = F(F^{i-1} \perp) \subseteq F(F^i \perp) = F^{i+1} \perp$$

since $F$ monotonic :-)

108
Step: Assume \( F^{i-1} \perp \subseteq F^i \perp \). Then

\[
F^i \perp = F (F^{i-1} \perp) \subseteq F (F^i \perp) = F^{i+1} \perp
\]
since \( F \) monotonic :-)

Conclusion:

If \( \mathbb{D} \) is finite, a solution can be found which is definitely the least :-)

Question:

What, if \( \mathbb{D} \) is not finite ???
Theorem \hspace{2cm} \textbf{Knaster – Tarski}

Assume $\mathbb{D}$ is a complete lattice. Then every \textit{monotonic} function $f : \mathbb{D} \to \mathbb{D}$ has a \textbf{least fixpoint} $d_0 \in \mathbb{D}$.

Let $P = \{d \in \mathbb{D} \mid f \ d \sqsubseteq d\}$.

Then $d_0 = \bigsqcap P$.
Bronislaw Knaster (1893-1980), topology
Theorem  Knaster – Tarski

Assume \( D \) is a complete lattice. Then every monotonic function \( f : D \to D \) has a least fixpoint \( d_0 \in D \).

Let \( P = \{ d \in D \mid f \, d \sqsubseteq d \} \).

Then \( d_0 = \bigcap P \).

Proof:

(1) \( d_0 \in P \) :
Theorem  Knaster – Tarski

Assume $\mathbb{D}$ is a complete lattice. Then every monotonic function $f : \mathbb{D} \to \mathbb{D}$ has a least fixpoint $d_0 \in \mathbb{D}$.

Let $P = \{d \in \mathbb{D} \mid f \ d \sqsubseteq d\}$.

Then $d_0 = \bigsqcap P$.

Proof:

(1) $d_0 \in P$:

\[ f \ d_0 \sqsubseteq f \ d \sqsubseteq d \quad \text{for all } d \in P \]

\[ \implies f \ d_0 \quad \text{is a lower bound of } P \]

\[ \implies f \ d_0 \sqsubseteq d_0 \quad \text{since } d_0 = \bigsqcap P \]

\[ \implies d_0 \in P \quad \therefore \]

113
(2) \[ f d_0 = d_0 : \]
(2) \( f d_0 = d_0 \):

\[ f d_0 \sqsubseteq d_0 \quad \text{by (1)} \]

\[ \implies f(f d_0) \sqsubseteq f d_0 \quad \text{by monotonicity of } f \]

\[ \implies f d_0 \in P \]

\[ \implies d_0 \sqsubseteq f d_0 \quad \text{and the claim follows } :-) \]
(2) \[ f d_0 = d_0 : \]

\[ f d_0 \sqsubseteq d_0 \quad \text{by (1)} \]

\implies \quad f(f d_0) \sqsubseteq f d_0 \quad \text{by monotonicity of } f

\implies \quad f d_0 \in P

\implies \quad d_0 \sqsubseteq f d_0 \quad \text{and the claim follows} \quad :-)

(3) \[ d_0 \quad \text{is least fixpoint:} \]
(2) \[ f d_0 = d_0 : \]
\[ f d_0 \sqsubseteq d_0 \text{ by (1)} \]
\[ \implies f (f d_0) \sqsubseteq f d_0 \text{ by monotonicity of } f \]
\[ \implies f d_0 \in P \]
\[ \implies d_0 \sqsubseteq f d_0 \text{ and the claim follows :)} \]

(3) \[ d_0 \text{ is least fixpoint:} \]
\[ f d_1 = d_1 \sqsubseteq d_1 \text{ an other fixpoint} \]
\[ \implies d_1 \in P \]
\[ \implies d_0 \sqsubseteq d_1 \text{ :))} \]
Remark:

The least fixpoint $d_0$ is in $P$ and a lower bound $:\implies$

$\implies$ $d_0$ is the least value $x$ with $x \sqsupseteq f x$
Remark:

The least fixpoint $d_0$ is in $P$ and a lower bound $:-)$

$\iff d_0$ is the least value $x$ with $x \sqsupseteq f\ x$

Application:

Assume $x_i \sqsupseteq f_i(x_1, \ldots, x_n), \quad i = 1, \ldots, n$ \hfill (\ast)

is a system of constraints where all $f_i : \mathbb{D}^n \rightarrow \mathbb{D}$ are monotonic.
Remark:

The least fixpoint $d_0$ is in $P$ and a lower bound :-)

$\implies d_0$ is the least value $x$ with $x \sqsupseteq f(x)$

Application:

Assume $x_i \sqsupseteq f_i(x_1, \ldots, x_n), \quad i = 1, \ldots, n$ (*)

is a system of constraints where all $f_i : \mathbb{D}^n \rightarrow \mathbb{D}$ are monotonic.

$\implies \text{least solution of}(*) \equiv \text{least fixpoint of } F \quad :-)$
Example 1:  \( \mathcal{D} = 2^U, \ f \ x = x \cap a \cup b \)
Example 1: \( \mathbb{D} = 2^U, \quad f x = x \cap a \cup b \)

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( f )</td>
<td>( f^k \perp )</td>
<td>( f^k \top )</td>
</tr>
<tr>
<td>0</td>
<td>( \emptyset )</td>
<td>( U )</td>
</tr>
</tbody>
</table>
Example 1: \[ D = 2^U, \quad f x = x \cap a \cup b \]

<table>
<thead>
<tr>
<th></th>
<th>( f^k \bot )</th>
<th>( f^k \top )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>( \emptyset )</td>
<td>( U )</td>
</tr>
<tr>
<td>1</td>
<td>( b )</td>
<td>( a \cup b )</td>
</tr>
</tbody>
</table>
Example 1: \( \mathbb{D} = 2^U, \quad f x = x \cap a \cup b \)

<table>
<thead>
<tr>
<th>( f )</th>
<th>( f^k \perp )</th>
<th>( f^k \top )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>( \emptyset )</td>
<td>( U )</td>
</tr>
<tr>
<td>1</td>
<td>( b )</td>
<td>( a \cup b )</td>
</tr>
<tr>
<td>2</td>
<td>( b )</td>
<td>( a \cup b )</td>
</tr>
</tbody>
</table>
Example 1: \( \mathbb{D} = 2^U, \quad f x = x \cap a \cup b \)

<table>
<thead>
<tr>
<th></th>
<th>( f^k \bot )</th>
<th>( f^k \top )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>( \emptyset )</td>
<td>( U )</td>
</tr>
<tr>
<td>1</td>
<td>( b )</td>
<td>( a \cup b )</td>
</tr>
<tr>
<td>2</td>
<td>( b )</td>
<td>( a \cup b )</td>
</tr>
</tbody>
</table>

Example 2: \( \mathbb{D} = \mathbb{N} \cup \{\infty\} \)

Assume \( f x = x + 1 \). Then

\[
    f^i \bot = f^i 0 = i \quad \square \quad i + 1 = f^{i+1} \bot
\]
Example 1: \( \mathbb{D} = 2^U, \quad f(x) = x \cap a \cup b \)

<table>
<thead>
<tr>
<th></th>
<th>( f^k \perp )</th>
<th>( f^k \top )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>( \emptyset )</td>
<td>( U )</td>
</tr>
<tr>
<td>1</td>
<td>( b )</td>
<td>( a \cup b )</td>
</tr>
<tr>
<td>2</td>
<td>( b )</td>
<td>( a \cup b )</td>
</tr>
</tbody>
</table>

Example 2: \( \mathbb{D} = \mathbb{N} \cup \{\infty\} \)

Assume \( f(x) = x + 1 \). Then

\[
\begin{align*}
  f^i \perp &= f^i 0 = i \quad \square \quad i + 1 = f^{i+1} \perp \\
\end{align*}
\]

\[\Rightarrow \text{ Ordinary iteration will never reach a fixpoint } :-(
\]
\[\Rightarrow \text{ Sometimes, transfinite iteration is needed } :-)
\]
Conclusion:

Systems of inequations can be solved through fixpoint iteration, i.e., by repeated evaluation of right-hand sides  :-)}
Conclusion:

Systems of inequations can be solved through fixpoint iteration, i.e., by repeated evaluation of right-hand sides :-)

Caveat: Naive fixpoint iteration is rather inefficient :-(

Conclusion:

Systems of inequations can be solved through fixpoint iteration, i.e., by repeated evaluation of right-hand sides :-)  

Caveat: Naive fixpoint iteration is rather inefficient:(

Example:

\[
\begin{align*}
0 & \quad y = 1; \\
1 & \quad \text{Pos}(x > 1) \\
2 & \quad y = x \ast y; \\
3 & \quad x = x - 1; \\
4 & \\
\end{align*}
\]
Conclusion:

Systems of inequations can be solved through fixpoint iteration, i.e., by repeated evaluation of right-hand sides :-)

Caveat: Naive fixpoint iteration is rather inefficient :-(
Conclusion:

Systems of inequations can be solved through fixpoint iteration, i.e., by repeated evaluation of right-hand sides :-)

Caveat: Naive fixpoint iteration is rather inefficient :-(

Example:
Conclusion:

Systems of inequations can be solved through fixpoint iteration, i.e., by repeated evaluation of right-hand sides  :-)

Caveat:  Naive fixpoint iteration is rather inefficient  :-(

Example:

\[
\begin{array}{c|c|c}
 & 1 & 2 \\
\hline
0 & \emptyset & \emptyset \\
1 & \{1, x > 1, x - 1\} & \{1\} \\
2 & \{1, x > 1, x - 1\} & \{1, x > 1\} \\
3 & \{1\} & \{1\} \\
4 & \{1, x > 1, x - 1\} & \{1, x > 1, x - 1\} \\
5 & \{1, x > 1, x - 1\} & \{1, x > 1\}
\end{array}
\]
Conclusion:

Systems of inequations can be solved through **fixpoint iteration**, i.e., by repeated evaluation of right-hand sides  :-)

Caveat:  Naive fixpoint iteration is rather **inefficient**  :-(

Example:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>∅</td>
<td>∅</td>
<td>∅</td>
<td>∅</td>
</tr>
<tr>
<td>1</td>
<td>{1, x &gt; 1, x − 1}</td>
<td>{1}</td>
<td>{1}</td>
<td>{1}</td>
</tr>
<tr>
<td>2</td>
<td><em>Expr</em></td>
<td>{1, x &gt; 1, x − 1}</td>
<td>{1, x &gt; 1}</td>
<td>{1, x &gt; 1}</td>
</tr>
<tr>
<td>3</td>
<td>{1, x &gt; 1, x − 1}</td>
<td>{1, x &gt; 1, x − 1}</td>
<td>{1, x &gt; 1, x − 1}</td>
<td>{1, x &gt; 1}</td>
</tr>
<tr>
<td>4</td>
<td>{1}</td>
<td>{1}</td>
<td>{1}</td>
<td>{1}</td>
</tr>
<tr>
<td>5</td>
<td><em>Expr</em></td>
<td>{1, x &gt; 1, x − 1}</td>
<td>{1, x &gt; 1}</td>
<td>{1, x &gt; 1}</td>
</tr>
</tbody>
</table>
Conclusion:

Systems of inequations can be solved through fixpoint iteration, i.e., by repeated evaluation of right-hand sides :-)

Caveat: Naive fixpoint iteration is rather inefficient :-(

Example:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>∅</td>
<td>∅</td>
<td>∅</td>
<td>∅</td>
<td>dito</td>
</tr>
<tr>
<td>1</td>
<td>{1, x &gt; 1, x - 1}</td>
<td>{1}</td>
<td>{1}</td>
<td>{1}</td>
<td>{1}</td>
</tr>
<tr>
<td>2</td>
<td>Expr</td>
<td>{1, x &gt; 1, x - 1}</td>
<td>{1, x &gt; 1}</td>
<td>{1, x &gt; 1}</td>
<td>{1, x &gt; 1}</td>
</tr>
<tr>
<td>3</td>
<td>{1, x &gt; 1, x - 1}</td>
<td>{1, x &gt; 1, x - 1}</td>
<td>{1, x &gt; 1, x - 1}</td>
<td>{1, x &gt; 1}</td>
<td>{1, x &gt; 1}</td>
</tr>
<tr>
<td>4</td>
<td>{1}</td>
<td>{1}</td>
<td>{1}</td>
<td>{1}</td>
<td>{1}</td>
</tr>
<tr>
<td>5</td>
<td>Expr</td>
<td>{1, x &gt; 1, x - 1}</td>
<td>{1, x &gt; 1}</td>
<td>{1, x &gt; 1}</td>
<td>{1, x &gt; 1}</td>
</tr>
</tbody>
</table>

Neg(x > 1) ▼ 0

Pos(x > 1) ▼ 1

y = 1;

y = x * y;

x = x - 1;
Idea: Round Robin Iteration

Instead of accessing the values of the last iteration, always use the current values of unknowns :-)}
Idea: Round Robin Iteration

Instead of accessing the values of the last iteration, always use the current values of unknowns :-)

Example:
Idea: Round Robin Iteration

Instead of accessing the values of the last iteration, always use the current values of unknowns :-)

Example:

```
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>y = 1;</td>
</tr>
<tr>
<td>1</td>
<td>Neg(x &gt; 1)</td>
</tr>
<tr>
<td></td>
<td>Pos(x &gt; 1)</td>
</tr>
<tr>
<td>2</td>
<td>y = x * y;</td>
</tr>
<tr>
<td>3</td>
<td>x = x - 1;</td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>∅</td>
</tr>
<tr>
<td>1</td>
<td>{1}</td>
</tr>
<tr>
<td>2</td>
<td>{1, x &gt; 1}</td>
</tr>
<tr>
<td>3</td>
<td>{1, x &gt; 1}</td>
</tr>
<tr>
<td>4</td>
<td>{1}</td>
</tr>
<tr>
<td>5</td>
<td>{1, x &gt; 1}</td>
</tr>
</tbody>
</table>
```
Idea: Round Robin Iteration

Instead of accessing the values of the last iteration, always use the current values of unknowns :-) 

Example:

\[
\begin{array}{c|c|c}
\text{Pos}(x > 1) & 1 & 2 \\
\hline
0 & \emptyset & \text{dito} \\
1 & \{1\} & \\
2 & \{1, x > 1\} & \\
3 & \{1, x > 1\} & \\
4 & \{1\} & \\
5 & \{1, x > 1\} & \\
\end{array}
\]
The code for **Round Robin Iteration in Java** looks as follows:

```java
for (i = 1; i ≤ n; i++) x_i = ⊥;

do {
    finished = true;
    for (i = 1; i ≤ n; i++) {
        new = f_i(x_1, ..., x_n);
        if (!(x_i ⊑ new)) {
            finished = false;
            x_i = x_i ⊔ new;
        }
    }
} while (!finished);
```