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 $d_0 = \sqcap P$. 
Bronislaw Knaster (1893-1980), topology
Theorem \hspace{1cm} \text{Knaster – Tarski}

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

Let $P = \{d \in \mathbb{D} \mid f \cdot d \sqsubseteq d\}$.
Then $d_0 = \bigsqcap P$.

\textbf{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 $d_0 = \sqcap P$.

Proof:

(1) $d_0 \in P$:

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

$\implies f d_0$ is a lower bound of $P$

$\implies f d_0 \sqsubseteq d_0$ since $d_0 = \sqcap P$

$\implies d_0 \in P$ 

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

\[
\begin{align*}
f d_0 \subseteq d_0 & \quad \text{by (1)} \\
\implies f(f d_0) \subseteq f d_0 & \quad \text{by monotonicity of } f \\
\implies f d_0 \in P \\
\implies d_0 \subseteq f d_0 & \quad \text{and the claim follows} \quad :-(
\end{align*}
\]
(2) \[ f d_0 = d_0 : \]

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

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

\[ \implies f d_0 \in P \]

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

(3) \( d_0 \) is least fixpoint:
(2) \( f d_0 = d_0 :\)

\[
fd_0 \sqsubseteq d_0 \quad \text{by (1)}
\]

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

\[
\implies \quad fd_0 \in P
\]

\[
\implies \quad d_0 \sqsubseteq fd_0 \quad \text{and the claim follows :-)}
\]

(3) \( d_0 \) is least fixpoint:

\[
fd_1 = d_1 \sqsubseteq d_1 \quad \text{an other fixpoint}
\]

\[
\implies \quad d_1 \in P
\]

\[
\implies \quad d_0 \sqsubseteq d_1 \quad :-))
\]
Remark:

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

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

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

$\implies d_0$ is the least value $x$ with $x \sqsubseteq f\ x$

Application:

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

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

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

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

Application:

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

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

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

\[
\begin{array}{|c|c|c|}
\hline
f & f^k \perp & f^k \top \\
\hline
0 & \emptyset & U \\
\hline
\end{array}
\]
Example 1: \( \mathbb{D} = 2^u, \ f x = x \cap a \cup b \)

<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: \( \mathbb{D} = 2^u, \ f x = x \cap a \cup b \)

<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 1: \( \mathbb{D} = 2^U, \ f x = x \cap a \cup b \)

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

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

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

<table>
<thead>
<tr>
<th>( f )</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

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

\( \implies \) Ordinary iteration will never reach a fixpoint \( :-( \)

\( \implies \) 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  :-)

Warning: 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  :-)

Warning: 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  :-)

Warning: 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  :-)

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

**Example:**

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

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

Warning: Naive fixpoint iteration is rather inefficient :( 

Example:

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

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

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

**Example:**

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

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

Warning: Naive fixpoint iteration is rather inefficient :-(

Example:
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:
Idea: Round Robin Iteration

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

Example:

```
0
  y = 1;

1
Pos(x > 1)
  y = x * y;
Neg(x > 1)

2

3
x = x - 1;

4

5
```

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>∅</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>{1}</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>{1, x &gt; 1}</td>
<td>dito</td>
</tr>
<tr>
<td>3</td>
<td>{1, x &gt; 1}</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>{1}</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>{1, x &gt; 1}</td>
<td></td>
</tr>
</tbody>
</table>
The code for Round Robin Iteration in Java looks as follows:

```
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);
```
Correctness:

Assume $y_i^{(d)}$ is the $i$-th component of $F^d \perp$.

Assume $x_i^{(d)}$ is the value of $x_i$ after the $d$-th RR-iteration.
Correctness:

Assume \( y_i^{(d)} \) is the \( i \)-th component of \( F^d \perp \).
Assume \( x_i^{(d)} \) is the value of \( x_i \) after the \( i \)-th RR-iteration.

One proves:

(1) \( y_i^{(d)} \subseteq x_i^{(d)} \) :-)

Correctness:

Assume $y_i^{(d)}$ is the $i$-th component of $F_d \perp$.
Assume $x_i^{(d)}$ is the value of $x_i$ after the $i$-th RR-iteration.

One proves:

(1) $y_i^{(d)} \sqsubseteq x_i^{(d)}$ :-)

(2) $x_i^{(d)} \sqsubseteq z_i$ for every solution $(z_1, \ldots, z_n) :-)$
Correctness:

Assume \( y_i^{(d)} \) is the \( i \)-th component of \( F^d \perp \).
Assume \( x_i^{(d)} \) is the value of \( x_i \) after the \( i \)-th RR-iteration.

One proves:

(1) \( y_i^{(d)} \sqsubseteq x_i^{(d)} \quad :-(\)

(2) \( x_i^{(d)} \sqsubseteq z_i \quad \text{for every solution} \quad (z_1, \ldots, z_n) \quad :-(\)

(3) If RR-iteration terminates after \( d \) rounds, then
\( (x_1^{(d)}, \ldots, x_n^{(d)}) \) is a solution \( \quad :-(\)\)
Warning:

The efficiency of RR-iteration depends on the ordering of the unknowns !!!
Warning:

The efficiency of RR-iteration depends on the ordering of the unknowns !!!

Good:

→ $u$ before $v$, if $u \rightarrow^* v$;
→ entry condition before loop body :-)
Warning:

The efficiency of RR-iteration depends on the ordering of the unknowns !!!

Good:

→ $u$ before $v$, if $u \rightarrow^* v$;

→ entry condition before loop body :-)

Bad:

e.g., post-order DFS of the CFG, starting at start :-)
Good:

\[ y = 1; \]
\[ x = x - 1; \]
\[ y = x \times y; \]
\[ x = x - 1; \]

Bad:

\[ y = 1; \]
\[ y = x \times y; \]
\[ x = x - 1; \]
Inefficient Round Robin Iteration:

\[ x = x - 1; \]
\[ y = x \times y; \]
\[ y = 1; \]
Inefficient Round Robin Iteration:

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

\[
\begin{array}{c|c}
\hline
\text{Pos} & \text{Neg} \\
\hline
\{1\} & \emptyset \\
\{\{1\}\} & \emptyset
\end{array}
\]
Inefficient Round Robin Iteration:

\[ x = x - 1; \]
\[ y = x \times y; \]

\[
\begin{array}{c|c|c}
\text{Pos}(x > 1) & 1 & 2 \\
\hline
0 & \{1\} & \{1, x > 1\} \\
1 & \{1\} & \{1\} \\
2 & \{1, x - 1, x > 1\} & \{1, x - 1, x > 1\} \\
3 & Expr & \{1, x > 1\} \\
4 & \{1\} & \{1\} \\
5 & \emptyset & \emptyset
\end{array}
\]
Inefficient Round Robin Iteration:

\[
y = 1;
\]

\[
\text{Neg}(x > 1) \quad \text{Pos}(x > 1)
\]

\[
\begin{align*}
0 & \quad 1 \\
3 & \quad 2 \\
5 & \quad 4 \\
\end{align*}
\]

\[
\begin{align*}
x = x - 1; \\
y = x * y;
\end{align*}
\]

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

\[ y = 1; \]
\[ y = x \times y; \]
\[ x = x - 1; \]

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

\[ \Rightarrow \text{significantly less efficient :)} \]
... end of background on: Complete Lattices
... end of background on: Complete Lattices

Final Question:

Why is a (or the least) solution of the constraint system useful ???
... end of background on: Complete Lattices

Final Question:

Why is a (or the least) solution of the constraint system useful ???

For a complete lattice $\mathbb{D}$, consider systems:

$$\mathcal{I}[start] \supseteq d_0$$

$$\mathcal{I}[v] \supseteq \left[ k \right]^\sharp (\mathcal{I}[u]) \quad k = (u, \_, v) \quad \text{edge}$$

where $d_0 \in \mathbb{D}$ and all $\left[ k \right]^\sharp : \mathbb{D} \rightarrow \mathbb{D}$ are monotonic ...
Final Question:

Why is a (or the least) solution of the constraint system useful???

For a complete lattice $\mathbb{D}$, consider systems:

\[
\begin{align*}
\mathcal{I}[\text{start}] & \supseteq d_0 \\
\mathcal{I}[v] & \supseteq [k]^\#(\mathcal{I}[u]) \quad k = (u, _, v) \quad \text{edge}
\end{align*}
\]

where $d_0 \in \mathbb{D}$ and all $[k]^\# : \mathbb{D} \rightarrow \mathbb{D}$ are monotonic ...

\[\Rightarrow\quad \text{Monotonic Analysis Framework}\]
Wanted: MOP (Merge Over all Paths)

\[ I^*[v] = \bigsqcup \{ [\pi]^d_0 \mid \pi : start \rightarrow^* v \} \]
Wanted: MOP (Merge Over all Paths)

\[ \mathcal{I}^*[v] = \bigsqcup \{ [[\pi]]^d_0 \mid \pi : \text{start} \rightarrow^* v \} \]

Theorem Kam, Ullman 1975

Assume \( \mathcal{I} \) is a solution of the constraint system. Then:

\[ \mathcal{I}[v] \supseteq \mathcal{I}^*[v] \quad \text{for every} \quad v \]
Wanted: MOP  (Merge Over all Paths)

\[ \mathcal{I}^*[v] = \bigsqcup \{ [[\pi]]^# d_0 \mid \pi : \text{start} \rightarrow^* v \} \]

Theorem  
Kam, Ullman 1975

Assume \( \mathcal{I} \) is a solution of the constraint system. Then:

\[ \mathcal{I}[v] \supseteq \mathcal{I}^*[v] \quad \text{for every } v \]

In particular:

\[ \mathcal{I}[v] \supseteq [[\pi]]^# d_0 \quad \text{for every } \pi : \text{start} \rightarrow^* v \]
Proof: Induction on the length of $\pi$. 
Proof: Induction on the length of $\pi$.

Foundation: $\pi = \epsilon$ (empty path)
Proof: Induction on the length of $\pi$.

Foundation: $\pi = \epsilon$ (empty path)
Then:

$$[[\pi]]^\# d_0 = [[\epsilon]]^\# d_0 = d_0 \subseteq \mathcal{I}[\text{start}]$$
Proof: Induction on the length of $\pi$.

Foundation: $\pi = \epsilon$ (empty path)

Then:

$$[[\pi]] d_0 = [[\epsilon]] d_0 = d_0 \subseteq \mathcal{I}[start]$$

Step: $\pi = \pi'k$ for $k = (u, _, v)$ edge.
Proof: Induction on the length of $\pi$.

**Foundation:** $\pi = \epsilon$ (empty path)

Then:

$$\llbracket \pi \rrbracket^d d_0 = \llbracket \epsilon \rrbracket^d d_0 = d_0 \subseteq \mathcal{I}[start]$$

**Step:** $\pi = \pi'k$ for $k = (u,_,v)$ edge.

Then:

$$\llbracket \pi' \rrbracket^d d_0 \subseteq \mathcal{I}[u] \quad \text{by I.H. for } \pi$$

$$\implies \llbracket \pi \rrbracket^d d_0 = \llbracket k \rrbracket^d (\llbracket \pi' \rrbracket^d d_0)$$

$$\subseteq \llbracket k \rrbracket^d (\mathcal{I}[u]) \quad \text{since } \llbracket k \rrbracket^d \text{ monotonic}$$

$$\subseteq \mathcal{I}[v] \quad \text{since } \mathcal{I} \text{ solution :-(}}$$
Disappointment:

Are solutions of the constraint system just upper bounds ???
Disappointment:

Are solutions of the constraint system just upper bounds ???

Answer:

In general: yes  :-(

Disappointment:

Are solutions of the constraint system just upper bounds ???

Answer:

In general: yes :-(

With the notable exception when all functions $[k]^\#$ are distributive ... :-)
The function \( f : \mathbb{D}_1 \rightarrow \mathbb{D}_2 \) is called

- **distributive**, if \( f (\sqcup X) = \sqcup \{ f x \mid x \in X \} \) for all \( \emptyset \neq X \subseteq \mathbb{D} \);
- **strict**, if \( f \bot = \bot \).
- **totally distributive**, if \( f \) is distributive and strict.
The function \( f : \mathbb{D}_1 \to \mathbb{D}_2 \) is called

- **distributive**, if \( f (\sqcup X) = \bigcup \{ f x \mid x \in X \} \) for all \( \emptyset \neq X \subseteq \mathbb{D} \);
- **strict**, if \( f \bot = \bot \).
- **totally distributive**, if \( f \) is distributive and strict.

**Examples:**

- \( f x = x \cap a \cup b \) for \( a, b \subseteq U \).
The function \( f : \mathbb{D}_1 \rightarrow \mathbb{D}_2 \) is called

- **distributive**, if \( f (\sqcup X) = \sqcup \{ f x \mid x \in X \} \) for all \( \emptyset \neq X \subseteq \mathbb{D} \);
- **strict**, if \( f \bot = \bot \).
- **totally distributive**, if \( f \) is distributive and strict.

**Examples:**

- \( f x = x \cap a \cup b \) for \( a, b \subseteq U \).

**Strictness:** \( f \emptyset = a \cap \emptyset \cup b = b = \emptyset \) whenever \( b = \emptyset \) :-(
The function \( f : \mathbb{D}_1 \rightarrow \mathbb{D}_2 \) is called

- **distributive**, if \( f (\bigcup X) = \bigcup \{ f x \mid x \in X \} \) for all \( \emptyset \neq X \subseteq \mathbb{D} \);
- **strict**, if \( f \perp = \perp \).
- **totally distributive**, if \( f \) is distributive and strict.

**Examples:**

- \( f x = x \cap a \cup b \) for \( a, b \subseteq U \).

  **Strictness:** \( f \emptyset = a \cap \emptyset \cup b = b = \emptyset \) whenever \( b = \emptyset \) :-(

  **Distributivity:**

  \[
  f (x_1 \cup x_2) = a \cap (x_1 \cup x_2) \cup b \\
  = a \cap x_1 \cup a \cap x_2 \cup b \\
  = f x_1 \cup f x_2 
  \]
\[ \mathcal{D}_1 = \mathcal{D}_2 = \mathbb{N} \cup \{\infty\}, \quad \text{inc } x = x + 1 \]
\( \mathbb{D}_1 = \mathbb{D}_2 = \mathbb{N} \cup \{\infty\}, \quad \text{inc} \ x = x + 1 \)

**Strictness:** \( f \bot = \text{inc} \ 0 = 1 \neq \bot \) :-(
• \( \mathbb{D}_1 = \mathbb{D}_2 = \mathbb{N} \cup \{\infty\} \), \( \text{inc} \, x = x + 1 \)

**Strictness:** \( f \perp = \text{inc} \, 0 = 1 \neq \perp :-( \)

**Distributivity:** \( f \,(\sqcup X) = \sqcup \{x + 1 \mid x \in X\} \) for \( \emptyset \neq X \ :-( \)
• $\mathbb{D}_1 = \mathbb{D}_2 = \mathbb{N} \cup \{\infty\}$, \hspace{1em} $\text{inc} \ x = x + 1$

**Strictness:** \hspace{1em} $f \bot = \text{inc} \ 0 = 1 \neq \bot$ \hspace{1em} :-(

**Distributivity:** \hspace{1em} $f(\bigsqcup X) = \bigsqcup \{x + 1 \mid x \in X\}$ \hspace{1em} for $\emptyset \neq X$ \hspace{1em} :-) 

• $\mathbb{D}_1 = (\mathbb{N} \cup \{\infty\})^2$, \hspace{1em} $\mathbb{D}_2 = \mathbb{N} \cup \{\infty\}$, \hspace{1em} $f(x_1, x_2) = x_1 + x_2$
• \( \mathbb{D}_1 = \mathbb{D}_2 = \mathbb{N} \cup \{\infty\} \), \hspace{1em} \text{inc} \ x = x + 1

**Strictness:** \( f \bot = \text{inc} \ 0 = 1 \neq \bot \) :-(

**Distributivity:** \( f (\bigsqcup X) = \bigsqcup \{x + 1 \mid x \in X\} \) for \( \emptyset \neq X \) :-)

• \( \mathbb{D}_1 = (\mathbb{N} \cup \{\infty\})^2 \), \hspace{1em} \mathbb{D}_2 = \mathbb{N} \cup \{\infty\}, \hspace{1em} f(x_1, x_2) = x_1 + x_2 :$

**Strictness:** \( f \bot = 0 + 0 = 0 \) :-)
\( \mathbb{D}_1 = \mathbb{D}_2 = \mathbb{N} \cup \{\infty\}, \quad \text{inc } x = x + 1 \)

**Strictness:** \( f \perp = \text{inc } 0 = 1 \neq \perp :-( \)

**Distributivity:** \( f (\biguplus X) = \biguplus \{x + 1 \mid x \in X\} \quad \text{for} \quad \emptyset \neq X :-( \)

\[ \]

\( \mathbb{D}_1 = (\mathbb{N} \cup \{\infty\})^2, \quad \mathbb{D}_2 = \mathbb{N} \cup \{\infty\}, \quad f(x_1, x_2) = x_1 + x_2 : \)

**Strictness:** \( f \perp = 0 + 0 = 0 :-( \)

**Distributivity:**

\[
\begin{align*}
f ((1, 4) \sqcup (4, 1)) & = f (4, 4) = 8 \\
& \neq 5 = f (1, 4) \sqcup f (4, 1) :-( \)
\end{align*}
\]

177