Script generated by TTT

Title: Seidl: Programmoptimierung (04.11.2015)

Date: Wed Nov 04 10:12:09 CET 2015

Duration: 83:37 min

Pages: 60

Wanted: minimally small solution for: $\sqsupseteq f_i(\perp)$

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

where all $f_i : \mathbb{D}^n \to \mathbb{D}$ are monotonic.

Idea $f_i(F^2\perp) \sqsupseteq f_i(F^2\perp)$

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

Hope: We eventually reach a solution ... ???

Example: $\mathbb{D} = 2^{\{a,b,c\}}, \quad \sqsubseteq \,=\, \subseteq$

$$\begin{array}{lcl} x_1 & \supseteq & \boxed{\{a\} \cup x_3} \\ x_2 & \supseteq & \boxed{x_3 \cap \{a, b\}} \\ x_3 & \supseteq & \boxed{x_1 \cup \{c\}} \end{array}$$

Example: $\mathbb{D} = 2^{\{a,b,c\}}, \quad \sqsubseteq \,=\, \subseteq$

$$\begin{array}{lcl} x_1 & \supseteq & \{a\} \cup x_3 \\ x_2 & \supseteq & x_3 \cap \{a, b\} \\ x_3 & \supseteq & x_1 \cup \{c\} \end{array}$$

The Iteration:

	0	1	2	3	4
x_1	$\emptyset$	a			
x_2	$\emptyset$	b			
x_3	$\emptyset$	c			

Example: $\mathbb{D} = 2^{\{a,b,c\}}, \quad \sqsubseteq = \subseteq$

$$\begin{aligned} x_1 &\supseteq \{a\} \cup x_3 \\ x_2 &\supseteq x_3 \cap \{a,b\} \\ x_3 &\supseteq x_1 \cup \{c\} \end{aligned}$$

The Iteration:

	0	1	2	3	4
x_1	$\emptyset$	$\{a\}$	a, a		
x_2	$\emptyset$	$\emptyset$	$\emptyset$		
x_3	$\emptyset$	$\{c\}$	a, c		

Example: $\mathbb{D} = 2^{\{a,b,c\}}, \quad \sqsubseteq = \subseteq$

$$\begin{aligned} x_1 &\supseteq \{a\} \cup x_3 \\ x_2 &\supseteq x_3 \cap \{a,b\} \\ x_3 &\supseteq x_1 \cup \{c\} \end{aligned}$$

The Iteration:

	0	1	2	3	4
x_1	$\emptyset$	$\{a\}$	$\{a,c\}$	$\{a,c\}$	
x_2	$\emptyset$	$\emptyset$	$\emptyset$	$\{a\}$	
x_3	$\emptyset$	$\{c\}$	$\{a,c\}$	$\{a,c\}$	

Example: $\mathbb{D} = 2^{\{a,b,c\}}, \quad \sqsubseteq = \subseteq$

$$\begin{aligned} x_1 &\supseteq \{a\} \cup x_3 \\ x_2 &\supseteq x_3 \cap \{a,b\} \\ x_3 &\supseteq x_1 \cup \{c\} \end{aligned}$$

The Iteration:

	0	1	2	3	4
x_1	$\emptyset$	$\{a\}$	$\{a,c\}$	$\{a,c\}$	ditto
x_2	$\emptyset$	$\emptyset$	$\emptyset$	$\{a\}$	
x_3	$\emptyset$	$\{c\}$	$\{a,c\}$	$\{a,c\}$	

Theorem

- $\underline{\perp}, F\,\underline{\perp}, F^2\,\underline{\perp}, \ldots$ form an ascending chain :

$$\boxed{\underline{\perp} \quad \sqsubseteq \quad F\,\underline{\perp} \quad \sqsubseteq \quad F^2\,\underline{\perp} \quad \sqsubseteq \quad \ldots}$$

- If $F^k\,\underline{\perp} = F^{k+1}\,\underline{\perp}$, a solution is obtained which is the least one.
- If all ascending chains are finite, such a k always exists.

Theorem

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

$$\bot \quad \sqsubseteq \quad F\,\bot \quad \sqsubseteq \quad F^2\,\bot \quad \sqsubseteq \quad \ldots$$

- 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.

Proof

The first claim follows by complete induction:

Foundation: $F^0\,\bot = \bot \sqsubseteq F^1\,\bot$.

Step: Assume $F^{i-1}\,\bot \sqsubseteq F^i\,\bot$. Then

$$F^i\,\bot = F\,(F^{i-1}\,\bot) \sqsubseteq F\,(F^i\,\bot) = F^{i+1}\,\bot$$

since F monotonic.

Theorem

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

$$\bot \quad \sqsubseteq \quad F\,\bot \quad \sqsubseteq \quad F^2\,\bot \quad \sqsubseteq \quad \ldots$$

- 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.

Proof

The first claim follows by complete induction:

Foundation: $F^0\,\bot = \bot \sqsubseteq F^1\,\bot$.

Step: Assume $F^{i-1}\,\bot \sqsubseteq F^i\,\bot$. Then

$$F^i\,\bot = F\,(F^{i-1}\,\bot) \sqsubseteq F\,(F^i\,\bot) = F^{i+1}\,\bot$$

since F monotonic.

Theorem

- $\perp, F\,\perp, F^2\,\perp, \ldots$ form an ascending chain :

$$\perp \;\sqsubseteq\; F\,\perp \;\sqsubseteq\; F^2\,\perp \;\sqsubseteq\; \ldots$$

- 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: $F^0\,\perp = \perp \sqsubseteq F^1\,\perp$.

Theorem

- $\perp, F\,\perp, F^2\,\perp, \ldots$ form an ascending chain :

$$\perp \;\sqsubseteq\; F\,\perp \;\sqsubseteq\; F^2\,\perp \;\sqsubseteq\; \ldots$$

- 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.

d solution, i.e. $d \sqsupseteq F\,d$

Claim: $\forall i.\ F^i\,\perp \sqsubseteq d$

$i = 0$: $F^0\,\perp = \perp \sqsubseteq d$ ✓

$i > 0$: Hyp.: $F^{i-1}\,\perp \sqsubseteq d$

$F^i\,\perp = F(F^{i-1}\,\perp) \sqsubseteq F\,d \sqsubseteq d$

Step: Assume $F^{i-1}\,\perp \;\sqsubseteq\; F^i\,\perp$. Then

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

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 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 $d_0 = \bigsqcap P$.

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 $P = \{d \in \mathbb{D} \mid f\, d \sqsubseteq d\}$. $x \sqsupseteq f\, x$

Then $d_0 = \bigsqcap P$.

Bronisław Knaster (1893-1980), topology

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) $d_0 \in P$:

$d_0 \sqsupseteq f\, d_0$

$d_0 \sqsubseteq f\, d_0$

d_0 is least

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) $d_0 \in P$:

$f\, d_0 \sqsubseteq f\, d \sqsubseteq d$ for all $d \in P$

$\Longrightarrow$ $f\, d_0$ is a lower bound of P

$\Longrightarrow$ $f\, d_0 \sqsubseteq d_0$ since $d_0 = \bigsqcap P$

$\Longrightarrow$ $d_0 \in P$

(2) $f\, d_0 \overset{\sqsubseteq}{=} d_0$: $\sqsupseteq$

(2) $f\, d_0 = d_0$:

$f\, d_0 \sqsubseteq d_0$ by (1)

$\Longrightarrow$ $f\,(\boxed{f\, d_0}) \sqsubseteq \boxed{f\, d_0}$ by monotonicity of f

$\Longrightarrow$ $\boxed{f\, d_0 \in P}$

$\Longrightarrow$ $d_0 \sqsubseteq f\, d_0$ and the claim follows.

(3) d_0 is least fixpoint:

(2) $f\, d_0 = d_0$:

$f\, d_0 \sqsubseteq d_0$ by (1)

$\Longrightarrow$ $f\,(f\, d_0) \sqsubseteq f\, d_0$ by monotonicity of f

$\Longrightarrow$ $f\, d_0 \in P$

$\Longrightarrow$ $d_0 \sqsubseteq f\, d_0$ and the claim follows.

(3) d_0 is least fixpoint:

(2) $f\, d_0 = d_0$:

$f\, d_0 \sqsubseteq d_0$ by (1)

$\Longrightarrow$ $f\,(f\, d_0) \sqsubseteq f\, d_0$ by monotonicity of f

$\Longrightarrow$ $f\, d_0 \in P$

$\Longrightarrow$ $d_0 \sqsubseteq f\, d_0$ and the claim follows.

(3) d_0 is least fixpoint: $f\, d_1 \sqsubseteq d_1$

$f\, d_1 = d_1 \sqsubseteq d_1$ an other fixpoint

$\Longrightarrow$ $d_1 \in P$

$\Longrightarrow$ $d_0 \sqsubseteq d_1$

Remark

The least fixpoint d_0 is in P and a lower bound.

$\Longrightarrow$ 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.

$\Longrightarrow$ d_0 is the least value x with $x \sqsupseteq f\,x$

Application

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

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

Remark

The least fixpoint d_0 is in P and a lower bound.

$\Longrightarrow$ d_0 is the least value x with $x \sqsupseteq f\,x$

Application

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

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

$\Longrightarrow$ least solution of $(*)$ $==$ least fixpoint of F.

Remark

The least fixpoint d_0 is in P and a lower bound.

$\Longrightarrow$ d_0 is the least value x with $x \sqsupseteq f\,x$

Application

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

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

$\Longrightarrow$ least solution of $(*)$ $==$ least fixpoint of F.

$F : \mathbb{D}^n \to \mathbb{D}^n$

Example 1 $\quad \mathbb{D} = 2^U, \quad f\,x = x \cap a \cup b$

Example 1 $\quad \mathbb{D} = 2^U, \quad f\,x = x \cap a \cup b$

f	$f^k\,\bot$	$f^k\,\top$
0	$\emptyset$	U

Example 1 $\quad \mathbb{D} = 2^U, \quad f\,x = x \cap a \cup b$

f	$f^k\,\bot$ $\subseteq$	$f^k\,\top$ $\supseteq$
0	$\emptyset$	U
1	b	$a \cup b$

$U \cap a \cup b$

$= a \cup b$

Example 1 $\quad \mathbb{D} = 2^U, \quad f\,x = x \cap a \cup b$

f	$f^k\,\bot$	$f^k\,\top$
0	$\emptyset$	U
1	b	$a \cup b$
2	b	$a \cup b$

Example 2 $\quad \mathbb{D} = \mathbb{N} \cup \{\infty\}$

Assume $\quad f\,x = x + 1$. Then

$$f^i\,\bot = f^i\,0 = i \quad \sqsubset \quad i + 1 = f^{i+1}\,\bot$$

$\Longrightarrow$ Ordinary iteration will never reach a fixpoint !

$\Longrightarrow$ Sometimes, transfinite iteration is needed.

Example 1 $\mathbb{D} = 2^U, \quad f\,x = x \cap a \cup b$

f	$f^k\,\bot$	$f^k\,\top$
0	$\emptyset$	U
1	b	$a \cup b$
2	b	$a \cup b$

Example 2 $\mathbb{D} = \mathbb{N} \cup \{\infty\}$

Assume $f\,x = x + 1$. Then

$$f^i\,\bot = f^i\,0 = i \quad \sqsubset \quad i + 1 = f^{i+1}\,\bot$$

$\Longrightarrow$ Ordinary iteration will never reach a fixpoint !

$\Longrightarrow$ 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.

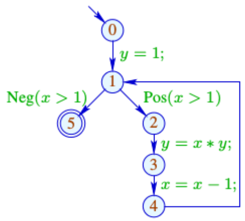
Caveat Naive fixpoint iteration is rather inefficient.

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



	1	2
0	$\emptyset$	$\emptyset$
1	$\{1, x > 1, x - 1\}$	$\{1\}$
2	$Expr$	$\{1, x > 1, x - 1\}$
3	$\{1, x > 1, x - 1\}$	$\{1, x > 1, x - 1\}$
4	$\{1\}$	$\{1\}$
5	$Expr$	$\{1, x > 1, x - 1\}$

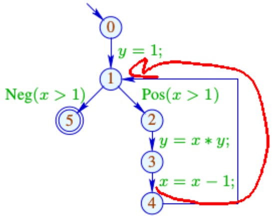
Conclusion

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



	1	2	3	4	5
0	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	
1	$\{1, x > 1, x-1\}$	$\{1\}$	$\{1\}$	$\{1\}$	
2	*Expr*	$\{1, x > 1, x-1\}$	$\{1, x > 1\}$	$\{1, x > 1\}$	
3	$\{1, x > 1, x-1\}$	$\{1, x > 1, x-1\}$	$\{1, x > 1, x-1\}$	$\{1, x > 1\}$	ditto
4	$\{1\}$	$\{1\}$	$\{1\}$	$\{1\}$	
5	*Expr*	$\{1, x > 1, x-1\}$	$\{1, x > 1\}$	$\{1, x > 1\}$	

Idea: Round Robin Iteration

Instead of accessing the values of the last iteration, always use the current values of unknowns.

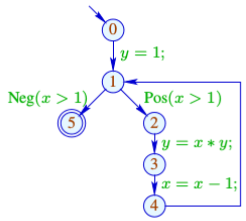
Idea: Round Robin Iteration

Instead of accessing the values of the last iteration, always use the current values of unknowns.

Idea: Round Robin Iteration

Instead of accessing the values of the last iteration, always use the current values of unknowns.

Example



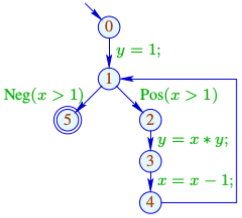
	1
0	$\emptyset$
1	$\{1\}$
2	$\{1, x > 1\}$
3	$\{1, x > 1\}$
4	$\{1\}$
5	$\{1, x > 1\}$

$(\emptyset \cup \{1\}) \setminus Expr_2 = \{1\}$

Idea: Round Robin Iteration

Instead of accessing the values of the last iteration, always use the current values of unknowns.

Example

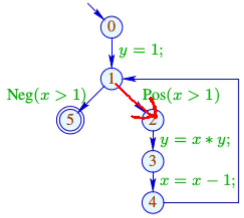


0
1
2
3
4
5

Idea: Round Robin Iteration

Instead of accessing the values of the last iteration, always use the current values of unknowns.

Example

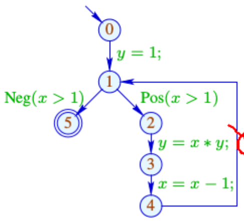


	1
0	$\emptyset$
1	$\{1\}$
2	$\{1, x > 1\}$
3	$\{1, x > 1\}$
4	$\{1\}$
5	$\{1, x > 1\}$

Idea: Round Robin Iteration

Instead of accessing the values of the last iteration, always use the current values of unknowns.

Example

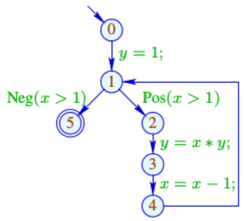


0
1
2
3
4
5

Idea: Round Robin Iteration

Instead of accessing the values of the last iteration, always use the current values of unknowns.

Example

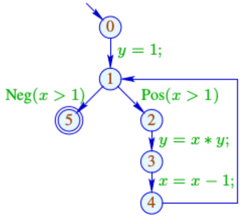


	1
0	$\emptyset$
1	$\{1\}$
2	$\{1, x > 1\}$
3	$\{1, x > 1\}$
4	$\{1\}$
5	$\{1, x > 1\}$

Idea: Round Robin Iteration

Instead of accessing the values of the last iteration, always use the current values of unknowns.

Example



	1	2
0	$\emptyset$	
1	$\{1\}$	
2	$\{1, x > 1\}$	
3	$\{1, x > 1\}$	ditto
4	$\{1\}$	
5	$\{1, x > 1\}$	

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 $F^d\,\underline{\perp}$.

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 $F^d\,\underline{\perp}$.

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

One proves:

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

Correctness

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

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 $F^d\,\underline{\perp}$.

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

One proves: ind. on d

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

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

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

Caveat

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

The code for Round Robin Iteration in Java looks as follows:

for $(i = 1; i \leq n; i{+}{+})\; x_i = \perp;$
do {
    $\mathit{finished} =$ true;
    for $(i = 1; i \leq n; i{+}{+})$ {
        $\mathit{new} = f_i(x_1, \ldots, x_n);$
        if $(!(x_i \sqsupseteq \mathit{new}))$ {
            $\mathit{finished} =$ false;
            $x_i = x_i \sqcup \mathit{new};$
        }
    }
} while $(!\mathit{finished});$

Correctness

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

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 $(z_1, \ldots, z_n)$.

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

Caveat

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

Caveat

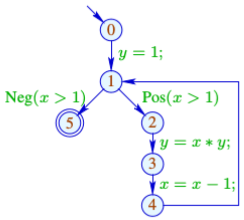
The efficiency of RR-iteration depends on the ordering of the unknowns !!!

Good:

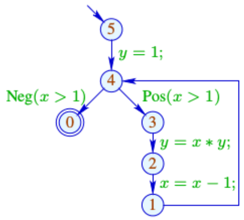
→ u before v, if $u \rightarrow^* v$;

→ entry condition before loop body.

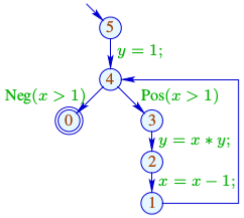
Good:



Bad:

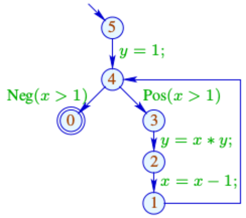


Inefficient Round Robin Iteration



0
1
2
3
4
5

Inefficient Round Robin Iteration



	1	2	3	4
0	$Expr$	$\{1, x > 1\}$	$\{1, x > 1\}$	
1	$\{1\}$	$\{1\}$	$\{1\}$	
2	$\{1, x - 1, x > 1\}$	$\{1, x - 1, x > 1\}$	$\{1, x > 1\}$	ditto
3	$Expr$	$\{1, x > 1\}$	$\{1, x > 1\}$	
4	$\{1\}$	$\{1\}$	$\{1\}$	
5	$\emptyset$	$\emptyset$	$\emptyset$	

$\Longrightarrow$ 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:

$$\begin{array}{lll} \mathcal{I}[start] & \sqsupseteq & d_0 \\ \mathcal{I}[v] & \sqsupseteq & [\![k]\!]^\sharp\,(\mathcal{I}[u]) \qquad k = (u, \_, v) \quad \text{edge} \end{array}$$

where $d_0 \in \mathbb{D}$ and all $[\![k]\!]^\sharp : \mathbb{D} \to \mathbb{D}$ are monotonic ...