Final Question

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

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

\[
\begin{align*}
\mathcal{I}_{\text{start}} & \quad d_0 \\
\mathcal{I}[v] & \quad \mathcal{I}^{I[v]} (I[u]) \\
\end{align*}
\]

where $d_0 \in \mathcal{D}$ and all $\mathcal{I}^{I[v]} : \mathcal{D} \to \mathcal{D}$ are monotonic ...

---

Wanted: MOP (Merge Over all Paths)

\[
\mathcal{I}^*[v] = \bigcup \{ \mathcal{I}^{I[v]} d_0 | \pi : \text{start} \to^* v \}
\]
... end of background on: Complete Lattices

Final Question

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

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

\[
\mathcal{I}[\text{start}] \ni d_0
\]
\[
\mathcal{I}[v] \ni (k)^{\mathcal{I}[u]} \quad k = (u, v) \quad \text{edge}
\]

where $d_0 \in \mathcal{D}$ and all $(k)^{\mathcal{I}[u]} : \mathcal{D} \to \mathcal{D}$ are monotonic ...

$\implies$ Monotonic Analysis Framework

---

Wanted: MOP (Merge Over all Paths)

\[
\mathcal{I}^*[v] = \bigcup\{ ([\pi])^* d_0 \mid \pi : \text{start} \to^* v \}
\]

---

Proof: Induction on the length of $\pi$.

Theorem Kam, Ullman 1975

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

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

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

Then:

$[\pi]^2 d_0 = [\varepsilon]^2 d_0 = d_0 \subseteq \mathcal{I}[\text{start}]$

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

Proof: Induction on the length of $\pi$.

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

Then:

$[\pi]^2 d_0 = [\varepsilon]^2 d_0 = d_0 \subseteq \mathcal{I}[\text{start}]$

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

Then:

$[\pi']^1 d_0 \subseteq \mathcal{I}[u] \quad \text{by I.H. for } \pi$

$[\pi']^1 d_0 = [k] \mathcal{I}[\mathcal{I}[u]] \quad \text{since } [k]^2 \text{ monotonic}$

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

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

Answer

In general: yes

With the notable exception when all functions $[k]^4$ are distributive ...

Has developed the operating system CP/M and GUIs for PCs.
Has developed the operating system CP/M and GUIs for PCs.

The function \( f : D_1 \rightarrow D_2 \) is called

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

**Examples**

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

**Strictness:** \( f \emptyset = a \cap \emptyset \cup b = b = \emptyset \) whenever \( b = \emptyset \).
The function \( f : D_1 \rightarrow D_2 \) is called

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

**Examples**

- \( f x = x \cap a \cup b \) for \( \forall a, b \in D \).
- **Strictness**: \( f \emptyset = \emptyset \cap a \cup b = b = \emptyset \) whenever \( b = \emptyset \)
- **Distributivity**: \[
f(x_1 \cup x_2) = a \cap (x_1 \cup x_2) \cup b = a \cap x_1 \cup a \cap x_2 \cup b = f(x_1) \cup f(x_2)
\]

**D_1 = D_2 = \mathbb{N} \cup \{ \infty \}, \quad \text{inc} \ x = x + 1**

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

**Distributivity**: \[
f(\bigcup X) = \bigcup \{ f x + 1 \mid x \in X \} \quad \text{for} \quad \emptyset \neq X
\]
• $D_1 = D_2 = N \cup \{\infty\}, \quad \text{inc} \ x = x + 1$

  Strictness: $f_\bot = \text{inc} \ 0 = 1 \neq \bot$

  Distributivity: $f(\bigcup X) = \bigcup \{x + 1 \mid x \in X\}$ for $\emptyset \neq X$

• $D_1 = (N \cup \{\infty\})^2, \quad D_2 = N \cup \{\infty\}, \quad f(x_1, x_2) = x_1 + x_2$

Remark

If $f : D_1 \to D_2$ is distributive, then also monotonic.

• $D_1 = D_2 = N \cup \{\infty\}, \quad \text{inc} \ x = x + 1$

  Strictness: $f_\bot = \text{inc} \ 0 = 1 \neq \bot$

  Distributivity: $f(\bigcup X) = \bigcup \{x + 1 \mid x \in X\}$ for $\emptyset \neq X$

• $D_1 = (N \cup \{\infty\})^2, \quad D_2 = N \cup \{\infty\}, \quad f(x_1, x_2) = x_1 + x_2$

  Strictness: $f_\bot = 0 + 0 = 0$

• $D_1 = D_2 = N \cup \{\infty\}, \quad \text{inc} \ x = x + 1$

  Strictness: $f_\bot = \text{inc} \ 0 = 1 \neq \bot$

  Distributivity: $f(\bigcup X) = \bigcup \{x + 1 \mid x \in X\}$ for $\emptyset \neq X$

• $D_1 = (N \cup \{\infty\})^2, \quad D_2 = N \cup \{\infty\}, \quad f(x_1, x_2) = x_1 + x_2$

  Strictness: $f_\bot = 0 + 0 = 0$

  Distributivity:

  $f((1, 4) \cup (4, 1)) = f(4, 4) = 8$

  $\neq 5 = f(1, 4) \cup f(4, 1)$
Remark

If $f : D_1 \rightarrow D_2$ is distributive, then also monotonic.

Obviously: $a \sqsubseteq b$ iff $a \sqcup b = b$.

Remark

If $f : D_1 \rightarrow D_2$ is distributive, then also monotonic.

Obviously: $a \sqsubseteq b$ iff $a \sqcup b = b$.

From that follows:

$$fb = f(a \sqcup b) = fa \sqcup fb \implies fa \sqsubseteq fb$$
Assumption: all \( v \) are reachable from \( \text{start} \).

Then:

Theorem \quad \text{Kildall 1972}

If all effects of edges \( [k]^i \) are distributive, then: \( I^*[v] = I[v] \) for all \( v \).

Proof

It suffices to prove that \( I^* \) is a solution!

For this, we show that \( I^* \) satisfies all constraints.
(1) We prove for \( \text{start} \):

\[
I^* [\text{start}] = \bigsqcup \{ [\pi]^d d_0 \mid \pi : \text{start} \rightarrow^* \text{start} \}
\]
\[= \bigsqcup \{ [k]^d d_0 \mid k : \text{start} \rightarrow^* \text{start} \}
\]
\[= \bigsqcup \{ \{ \pi[k] \}^d d_0 \mid \pi' : \text{start} \rightarrow^* 0 \}
\]
\[= \bigsqcup \{ [k]^d \{ [\pi']^d d_0 \mid \pi' : \text{start} \rightarrow^* 0 \} \}
\]
\[= \bigsqcup \{ [k]^d (I^*[0]) \}
\]

since \( \{ \pi' \mid \pi' : \text{start} \rightarrow^* 0 \} \) is non-empty.

(2) For every \( k = (u, v) \) we prove:

\[
I^*[k] = \bigsqcup \{ [\pi]^d d_0 \mid \pi : \text{start} \rightarrow^* v \}
\]
\[= \bigsqcup \{ [\pi[u]]^d d_0 \mid \pi' : \text{start} \rightarrow^* u \}
\]
\[= \bigsqcup \{ [k]^d \{ [\pi'[]']^d d_0 \mid \pi' : \text{start} \rightarrow^* u \} \}
\]
\[= \bigsqcup \{ [k]^d \{ I^*[u] \} \}
\]

Caveat

- **Reachability** of all program points cannot be abandoned!

Consider:

\[\xymatrix{\cdot\ar[r] & 7 \ar[r] & 0 \ar[r] & 1 \ar[r] & \text{inc} \ar[r] & 2 \ar[r] & \cdot}
\]

where \( D = \mathbb{N} \cup \{ \infty \} \)

Then:

\[
I[2] = \text{inc} 0 = \{ 0 \}
\]

\[
I^*[2] = \bigsqcup \{ 0 \} = 0
\]

Summary and Application

\[
\begin{aligned}
(a \cup (x_1 \cap x_2)) \cap b &= ((a \cup x_1) \cap (a \cup x_2)) \cap b \\
&= ((a \cup x_1) \cap b) \cap ((a \cup x_2) \cap b)
\end{aligned}
\]
Summary and Application

→ The effects of edges of the analysis of availability of expressions are distributive:

\[
(a \cup (x_1 \cap x_2)) \setminus b = ((a \cup x_1) \cap (a \cup x_2)) \setminus b \\
= ((a \cup x_1) \setminus b) \cap ((a \cup x_2) \setminus b)
\]

→ If all effects of edges are distributive, then the MOP can be computed by means of the constraint system and RR-iteration.

→ The effects of edges of the analysis of availability of expressions are distributive:

\[
(a \cup (x_1 \cap x_2)) \setminus b = ((a \cup x_1) \cap (a \cup x_2)) \setminus b \\
= ((a \cup x_1) \setminus b) \cap ((a \cup x_2) \setminus b)
\]

→ If all effects of edges are distributive, then the MOP can be computed by means of the constraint system and RR-iteration.

→ If not all effects of edges are distributive, then RR-iteration for the constraint system at least returns a safe upper bound to the MOP.

1.2 Removing Assignments to Dead Variables

Example:

1: \(x = y + 2;\)
2: \(y = 5;\)
3: \(x = y + 3;\)

The value of \(x\) at program points 1, 2 is over-written before it can be used.

Therefore, we call the variable \(x\) dead at these program points.

Example:

1: \(x = y + 2;\)
2: \(y = 5;\)
3: \(x = y + 3;\)

The value of \(x\) at program points 1, 2 is over-written before it can be used.

Therefore, we call the variable \(x\) dead at these program points.

:-)
Note:

→ Assignments to dead variables can be removed :) 
→ Such inefficiencies may originate from other transformations.

Formal Definition:

The variable \( x \) is called live at \( u \) along the path \( \pi \) starting at \( u \) relative to a set \( X \) of variables either:

if \( x \in X \) and \( \pi \) does not contain a definition of \( x \); or:

if \( \pi \) can be decomposed into: \( \pi = \pi_1 k \pi_2 \) such that:

• \( k \) is a use of \( x \); and
• \( \pi_1 \) does not contain a definition of \( x \).

Thereby, the set of all defined or used variables at an edge \( k = (u, la, b, v) \) is defined by:

<table>
<thead>
<tr>
<th>lab</th>
<th>used</th>
<th>defined</th>
</tr>
</thead>
<tbody>
<tr>
<td>;</td>
<td>\emptyset</td>
<td>\emptyset</td>
</tr>
<tr>
<td>Pos(e)</td>
<td>Vars(e)</td>
<td>\emptyset</td>
</tr>
<tr>
<td>Neg(e)</td>
<td>Vars(e)</td>
<td>\emptyset</td>
</tr>
<tr>
<td>x = c;</td>
<td>Vars(e)</td>
<td>{x}</td>
</tr>
<tr>
<td>x = M[e];</td>
<td>Vars(e)</td>
<td>{x}</td>
</tr>
<tr>
<td>M[e_1] = c_2;</td>
<td>Vars(e_1) \cup Vars(e_2)</td>
<td>\emptyset</td>
</tr>
</tbody>
</table>
A variable \( x \) which is not live at \( u \) along \( \pi \) (relative to \( X \)) is called dead at \( u \) along \( \pi \) (relative to \( X \)).

Example:

\[
\begin{array}{c}
0 \\
1 \\
2 \\
3
\end{array}
\]

\[
\begin{array}{c} x = y + 2; \quad y = 5; \quad x = y + 3. \end{array}
\]

where \( X = \emptyset \). Then we observe:

<table>
<thead>
<tr>
<th>#</th>
<th>live</th>
<th>dead</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>{y}</td>
<td>{z}</td>
</tr>
<tr>
<td>1</td>
<td>\emptyset</td>
<td>{x, y}</td>
</tr>
<tr>
<td>2</td>
<td>{y}</td>
<td>{z}</td>
</tr>
<tr>
<td>3</td>
<td>\emptyset</td>
<td>{x, y}</td>
</tr>
</tbody>
</table>

The variable \( x \) is live at \( u \) (relative to \( X \)) if \( x \) is live at \( u \) along some path to the exit (relative to \( X \)). Otherwise, \( x \) is called dead at \( u \) (relative to \( X \)).

Question:

How can the sets of all dead/live variables be computed for every \( u \) ?

Idea:

For every edge \( k = (u, v) \), define a function \([k]^k\) which transforms the set of variables which are live at \( v \) into the set of variables which are live at \( u \).
Let \( L = 2^{\text{Vars}} \).

For \( k = (\_ , \text{lab} , \_ ) \), define \([k]^4 = [\text{lab}]^4\) by:

\[
\begin{align*}
[k]^4 L &= L \\
\left[\text{Pos}(e)\right]^4 L &= \left[\text{Neg}(e)\right]^4 L = L \cup \text{Vars}(e) \\
[x = e]^4 L &= (L \setminus \{x\}) \cup \text{Vars}(e) \\
[x = M[e]_1]^4 L &= (L \setminus \{x\}) \cup \text{Vars}(e) \\
[M[e_1] = e_2]^4 L &= L \cup \text{Vars}(e_1) \cup \text{Vars}(e_2)
\end{align*}
\]

\[\boxed{\mathfrak{X} = \mathfrak{X} + \mathfrak{y}} \]

We verify that these definitions are meaningful \( : \)

\[\begin{align*}
x = y + 2; & \quad y = 5; & \quad x = y + 2; & \quad M[y] = x;
\end{align*}\]
We verify that these definitions are meaningful \( \therefore \)

\[
x = y + 2; \quad y = 5; \quad x = y + 2; \quad M[y] = x;
\]

The set of variables which are live at \( u \) then is given by:

\[
\mathcal{L}^*[u] = \bigcup \{ \{x\} : x \in \mathcal{L}^*[u] \}
\]

... literally:

- The paths start in \( u \) \( \therefore \)
  \[ \implies \text{ As partial ordering for } L, \text{ we use } \subseteq = \subseteq. \]
- The set of variables which are live at program exit is given by the set \( X \) \( \therefore \)

Transformation 2:

\[
x \not\in \mathcal{L}^*[u]
\]

Correctness Proof:

\[ \therefore \text{Correctness of the effects of edges: If } L \text{ is the set of variables which are live at the exit of the path } \pi, \text{ then } \{x\} L \text{ is the set of variables which are live at the beginning of } \pi \therefore \]

\[ \therefore \text{Correctness of the transformation along a path: If the value of a variable is accessed, this variable is necessarily live. The value of dead variables thus is irrelevant } \therefore \]

\[ \therefore \text{Correctness of the transformation: In any execution of the transformed programs, the live variables always receive the same values } \therefore \]
Computation of the sets $\mathcal{L}^*[u]$:

1. Collecting constraints:
   \[
   \mathcal{L}_{\text{step}} \supseteq X \\
   \mathcal{L}[u] \supseteq [k]^2(\mathcal{L}[v]) \\
   k = (u, u, v) \text{ edge}
   \]

2. Solving the constraint system by means of RR iteration.
   Since $\mathcal{L}$ is finite, the iteration will terminate $\therefore$

3. If the exit is (formally) reachable from every program point, then the smallest solution $\mathcal{L}$ of the constraint system equals $\mathcal{L}^*$ since all $[k]^2$ are distributive $\therefore$

Transformation 2:

- $x \notin \mathcal{L}^*[v]$
- $x = c$
- $x \notin \mathcal{L}^*[v]$
- $x = M[e];$
- $x \notin \mathcal{L}^*[v]$