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. ::)

→ 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: \(x \leftarrow y + 2;\)

2: \(y = 5;\)

3: \(x = y + 3;\)

The value of \(x\) at program points 1, 2 is overwritten 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 \( k = (u, lab, v) \) 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 = e )</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] = e _1 \cup Vars (e _2) )</td>
<td>Vars ( (e _1) \cup Vars (e _2) )</td>
<td>( \emptyset )</td>
</tr>
</tbody>
</table>

Example:

A variable \( x \) which is not live at \( u \) along \( \pi \) (relative to \( X \)) is called dead at \( u \) along \( \pi \) (relative to \( X \)).

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

<table>
<thead>
<tr>
<th>live</th>
<th>dead</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 {y}</td>
<td>{x}</td>
</tr>
<tr>
<td>1 ( \emptyset )</td>
<td>{x,y}</td>
</tr>
<tr>
<td>2 {y}</td>
<td>{x}</td>
</tr>
<tr>
<td>3 ( \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 \( u \)?

Let \( L = 2^{\text{Vars}} \).

For \( k = (u, \_ , v) \), define \([k]^3\) by:

\[
\begin{align*}
[k]^3 L &= L \\
[\text{Pos}(e)]^3 L &= [\text{Neg}(e)]^3 L = L \cup \text{Vars}(e) \\
[x = c]^3 L &= (L \setminus \{x\}) \cup \text{Vars}(e) \\
[x = M[e]]^3 L &= (L \setminus \{x\}) \cup \text{Vars}(e) \\
[M[e_1] = e_2]^3 L &= L \cup \text{Vars}(e_1) \cup \text{Vars}(e_2)
\end{align*}
\]
Let \( L = 2^\text{Vars} \).

For \( k = (\_, lab, \_) \), define \( [k]^2 = [\text{lab}]^2 \) by:

\[
\begin{align*}
[k]^2 L &= L \\
[[\text{Pos}(e)]^2 L] &= L \cup \text{Vars}(e) \\
[[x = e]^2 L] &= (L \setminus \{x\}) \cup \text{Vars}(e) \\
[x = M[e_1]^2 L] &= (L \setminus \{x\}) \cup \text{Vars}(e) \\
[M[e_1] = e_2]^2 L &= L \cup \text{Vars}(e_1) \cup \text{Vars}(e_2)
\end{align*}
\]

\( [k]^2 \) can again be composed to the effects of \( [\pi]^2 \) of paths \( \pi = k_1 \ldots k_n \) by:

\[
[k]^2 = (k_1)^2 \circ \ldots \circ (k_n)^2
\]

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

\[
\begin{align*}
x &= y + 2; & y &= 5; & x &= y + 2; & M[y] &= x;
\end{align*}
\]

\( \mathcal{L}^* [u] = \bigcup \{[\pi]^2 X \mid \pi : u \rightarrow^* \text{stop} \} \)

... literally:

- The paths start in \( u \) \( :-) \).
  
  \[\begin{array}{c}
  \quad \\
  \quad \\
  \quad \\
  \end{array}\]
  
  As partial ordering for \( L \) we use \( \subseteq \subseteq \).

- The set of variables which are live at program exit is given by the set \( X \) \( :-) \).
Transformation 2:

\[ x = e; \quad x \notin L^*[e] \]

\[ x = M[e]; \quad x \notin L^*[e] \]

Correctness Proof:

→ **Correctness of the effects of edges:** If \( L \) is the set of variables which are live at the exit of the path \( \pi \), then \([\pi]^E L\) is the set of variables which are live at the beginning of \( \pi \) :-)

→ **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 :-)

→ **Correctness of the transformation:** In any execution of the transformed programs, the live variables always receive the same values :-))
Computation of the sets $L^*[u]$:

1. Collecting constraints:
   
   $L_{\text{def}(u)} \supseteq X$
   
   $L[u] \supseteq \llbracket k \rrbracket (L[u])$
   
   $k = (u, \ldots, v)$ edge

2. Solving the constraint system by means of RR iteration.
   
   Since $L$ is finite, the iteration will terminate :-)

3. If the exit is (formally) reachable from every program point, then the smallest solution $L$ of the constraint system equals $L^*$ since all $\llbracket k \rrbracket$ are distributive :-))

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*}

Correctness Proof:

$\therefore x = x \land a \land b$

→ Correctness of the effects of edges: If $L$ is the set of variables which are live at the exit of the path $\pi$, then $\llbracket \pi \rrbracket^2 L$ is the set of variables which are live at the beginning of $\pi$ :-)

→ 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 :-)

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

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

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

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 :-))

Caveat: The information is propagated backwards !!!!

Example:

Example:
The left-hand side of no assignment is dead :-)

Caveat:
Removal of assignments to dead variables may kill further variables:

```
1  x = y + 1;
2  z = 2 * x;
3  M[R] = y;
4  
```

Re-analyzing the program is inconvenient :-(

Idea: Analyze true liveness!

- \( x \) is called truly live at \( u \) along a path \( \pi \) (relative to \( X \)), either
  - if \( x \in X \), \( \pi \) 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 true use of \( x \) relative to \( \pi_2 \);
    - \( \pi_1 \) does not contain any definition of \( x \).
The set of truely used variables at an edge \( k = (\_, \text{lab}, v) \) is defined as:

<table>
<thead>
<tr>
<th>lab</th>
<th>truely used</th>
</tr>
</thead>
<tbody>
<tr>
<td>;</td>
<td>( \emptyset )</td>
</tr>
<tr>
<td>Pos ((e))</td>
<td>( \text{Vars}(e) )</td>
</tr>
<tr>
<td>Neg ((e))</td>
<td>( \text{Vars}(e) )</td>
</tr>
<tr>
<td>( x = e; )</td>
<td>( \text{Vars}(e) )</td>
</tr>
<tr>
<td>( x = M[e]; )</td>
<td>( \text{Vars}(e) )</td>
</tr>
<tr>
<td>( M[e_1] = e_2; )</td>
<td>( \text{Vars}(e_1) \cup \text{Vars}(e_2) )</td>
</tr>
</tbody>
</table>

(*) given that \( x \) is truely live at \( v \) w.r.t. \( \pi_2 \): ☐

Example:

\[
\begin{align*}
1 & : x = y + 1; \\
2 & : z = 2 * x; \\
3 & : M[R] = y; \\
4 & : \emptyset
\end{align*}
\]
Example:

1. $y, R$
2. $x = y + 1$
3. $y, R$
4. $z = 2 \times x$

Example:

1. $y, R$
2. $z = y + 1$
3. $y, R$
4. $M[R] = y$

The Effects of Edges:

\[
\begin{align*}
\llbracket \cdot \rrbracket^L L &= L \\
\llbracket \text{Pos}(e) \rrbracket^L L &= [\text{Neg}(e)]^L L = L \cup \text{Vars}(e) \\
\llbracket x = c \rrbracket^L L &= (L \setminus \{x\}) \cup \text{Vars}(c) \\
\llbracket x = M[e] \rrbracket^L L &= (L \setminus \{x\}) \cup \text{Vars}(c) \\
\llbracket M[e_1] = c_2 \rrbracket^L L &= L \cup \text{Vars}(e_1) \cup \text{Vars}(e_2)
\end{align*}
\]
Note:

- The effects of edges for truly live variables are more complicated than for live variables :-)
- Nonetheless, they are distributive !!

Note:

- The effects of edges for truly live variables are more complicated than for live variables :-)
- Nonetheless, they are distributive !!

To see this, consider for \( \mathcal{D} = 2^U \), \( f \ y = \{ u \in y \} \ ? \ b : \emptyset \). We verify:

\[
\begin{align*}
\quad f( y_1 \cup y_2 ) &= \{ u \in y_1 \cup y_2 \} \ ? \ b : \emptyset \\
&= \{ u \in y_1 \cup v \in y_2 \} \ ? \ b : \emptyset \\
&= \{ u \in y_1 \} \ ? \ b : \emptyset \cup \{ v \in y_2 \} \ ? \ b : \emptyset \\
&= f( y_1 ) \cup f( y_2 )
\end{align*}
\]

\[ \Rightarrow \text{the constraint system yields the MOP} \quad \Rightarrow \]

Note:

- True liveness detects more superfluous assignments than repeated liveness !!!