Problem Sheet 3: Data Flow Analysis
Sample Solutions

Chris Poskitt*
ETH Zürich

Starred exercises (*) are more challenging than the others.

1 Reaching Definitions Analysis

i-ii. The control flow graph and the results of the reaching definitions analysis are given in the diagram below:

iii. We give the use-definition information for x and y in the table below (you could also annotate the diagram above with additional arrows).

*Solutions adapted from an earlier version of the course, when Stephan van Staden was the teaching assistant.
2 Live Variables Analysis

i. Below we identify the blocks of the program:

\[
\begin{align*}
[x := y] &^1 \\
[x := x-1] &^2 \\
[x := 4] &^3 \\
\text{while } [y < x] &^4 \text{ do} \\
\text{[y := y+x]} &^5 \\
\text{end} \\
[y := 0] &^6
\end{align*}
\]

ii. The system of equations for a live variable analysis are as follows:

\[
\begin{align*}
LV_{\text{entry}}(1) &= (LV_{\text{exit}}(1) - \{x\}) \cup \{y\} \\
LV_{\text{entry}}(2) &= (LV_{\text{exit}}(2) - \{x\}) \cup \{x\} \\
LV_{\text{entry}}(3) &= LV_{\text{exit}}(3) - \{x\} \\
LV_{\text{entry}}(4) &= LV_{\text{exit}}(4) \cup \{x, y\} \\
LV_{\text{entry}}(5) &= (LV_{\text{exit}}(5) - \{y\}) \cup \{x, y\} \\
LV_{\text{entry}}(6) &= LV_{\text{exit}}(6) - \{y\} \\
LV_{\text{exit}}(1) &= LV_{\text{entry}}(2) \\
LV_{\text{exit}}(2) &= LV_{\text{entry}}(3) \\
LV_{\text{exit}}(3) &= LV_{\text{entry}}(4) \\
LV_{\text{exit}}(4) &= LV_{\text{entry}}(5) \cup LV_{\text{entry}}(6) \\
LV_{\text{exit}}(5) &= LV_{\text{entry}}(4) \\
LV_{\text{exit}}(6) &= \emptyset
\end{align*}
\]

iii. We begin the iteration by initialising every set to $\emptyset$. Then, we iteratively update the sets by applying the equation system above. (For simplicity, the columns omit sets when a particular iteration does not update the previous value.)
iv. We eliminate blocks $b$ of the form $[x := \ldots]^b$ if $x$ is not an element of $\text{LV}_{\text{exit}}(b)$:

$$
\begin{align*}
[x & := y]^1 \\
[x & := 4]^3 \\
\text{while } [y < x]^4 \text{ do} \\
[y & := y+x]^5 \\
\text{end}
\end{align*}
$$

v. (∗) The program is not yet free of dead variables: $x$ in block 1 is still dead. We strengthen the definition of $\text{LV}_{\text{entry}}$:

$$
\text{LV}_{\text{entry}}(b) = \begin{cases} 
(\text{LV}_{\text{exit}}(b) - \text{kill}_{\text{LV}}(b)) \cup \text{gen}_{\text{LV}}(b) & \text{if } \text{kill}_{\text{LV}}(b) \subseteq \text{LV}_{\text{exit}}(b) \\
\text{LV}_{\text{exit}}(b) & \text{otherwise}
\end{cases}
$$

The rationale is this: if a block assigns to a variable that is not live afterwards, then it must be eliminated, and should not influence the analysis by adding the variables it reads to the live variable set.

Performing a chaotic iteration with this new equation yields the following results:

<table>
<thead>
<tr>
<th>LV Sets</th>
<th>Final Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{LV}_{\text{entry}}(1)$</td>
<td>${y}$</td>
</tr>
<tr>
<td>$\text{LV}_{\text{entry}}(2)$</td>
<td>${y}$</td>
</tr>
<tr>
<td>$\text{LV}_{\text{entry}}(3)$</td>
<td>${y}$</td>
</tr>
<tr>
<td>$\text{LV}_{\text{entry}}(4)$</td>
<td>${x,y}$</td>
</tr>
<tr>
<td>$\text{LV}_{\text{entry}}(5)$</td>
<td>${x,y}$</td>
</tr>
<tr>
<td>$\text{LV}_{\text{entry}}(6)$</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>$\text{LV}_{\text{exit}}(1)$</td>
<td>${x,y}$</td>
</tr>
<tr>
<td>$\text{LV}_{\text{exit}}(2)$</td>
<td>${y}$</td>
</tr>
<tr>
<td>$\text{LV}_{\text{exit}}(3)$</td>
<td>${x,y}$</td>
</tr>
<tr>
<td>$\text{LV}_{\text{exit}}(4)$</td>
<td>${x,y}$</td>
</tr>
<tr>
<td>$\text{LV}_{\text{exit}}(5)$</td>
<td>${x,y}$</td>
</tr>
<tr>
<td>$\text{LV}_{\text{exit}}(6)$</td>
<td>$\emptyset$</td>
</tr>
</tbody>
</table>

with which we can eliminate all of the dead code in the program:

$$
\begin{align*}
[x & := 4]^3 \\
\text{while } [y < x]^4 \text{ do} \\
[y & := y+x]^5 \\
\text{end}
\end{align*}
$$