1 Strongly Live Variables

The strongly live variable analysis is an extension to the live variables analysis. A variable is considered strongly live only, if it is used in an assignment of another strongly live variable, or if it is used within an statement other than an assignment. Therefore, the kill/gen-sets of strongly live variables analysis are those from the live-variables analysis:

\[ \text{kill}_{\text{SLV}}([x := a]) = \{ x \} \]
\[ \text{kill}_{\text{SLV}}([\text{skip}]) = \emptyset \]
\[ \text{kill}_{\text{SLV}}([b]) = \emptyset \]

\[ \text{gen}_{\text{SLV}}([x := a]) = \text{FV}(a) \]
\[ \text{gen}_{\text{SLV}}([\text{skip}]) = \emptyset \]
\[ \text{gen}_{\text{SLV}}([b]) = \text{FV}(b) \]

where \( \text{FV}(x) \) denotes the variables used in expression \( x \).

The difference between both analyses is caused by the computation of the entry-sets. In case of an assignment \([x := y]\), the data flow value is only updated if the variable \( x \) is a strongly live variable:

\[
\text{SLV}_{\text{Entry}}(l) = \begin{cases} 
\text{SLV}_{\text{exit}}(l) \setminus \text{kill}_{\text{SLV}}(B^l) \cup \text{gen}_{\text{SLV}}(B^l) & \text{if } \text{kill}_{\text{SLV}}(B^l) \subset \text{SLV}_{\text{exit}}(l) \\
\text{SLV}_{\text{exit}}(l) & \text{otherwise}
\end{cases}
\]

\[
\text{SLV}_{\text{Exit}}(l) = \begin{cases} 
\emptyset & \text{if } l \in \text{final}(S) \\
\bigcup \{ \text{SLV}_{\text{entry}}(l') \mid l', l \in \text{flow}^R(S) \} & \text{otherwise}
\end{cases}
\]
2 MOP versus MFP

a) The MOP solution is in general more precise or at least as precise as the MFP solution. MOP means *merge over all paths* and combines the data flow values of all possible paths at the end of the paths and not, at the points where the control flow converges. In general, however, the MOP solution is not computable, since there might be infinitely many possible paths (for instance, in case of an unbounded loop). The MFP solution, the maximal fixpoint safely approximates the MOP solution.

b) Both solutions are equal in case the transfer functions are distributive \((f(x \cup y) = f(x) \cup f(y))\). The intuition is rather simple: if the transfer function is distributive, it does not matter, if the data flow information is merged at the end or as the flow converges.

c) A simple analysis were MOP and MFP differ is the *constant propagation analysis*.

\[
\begin{align*}
    x &> 0; \uparrow^1 \\
    x &:= -2; \uparrow^2 \\
    x &:= 2; \uparrow^3 \\
    x &< 0; \uparrow^4 \\
    x &:= 2 \times x; \uparrow^5 \\
    x &:= -2 \times x; \uparrow^6 \\
    c &= x; \uparrow^7
\end{align*}
\]

Obviously, there are only two valid paths within the program: \((1, 3, 4, 5, 7)\) in case \(x\) is positive and \((1, 2, 4, 6, 7)\) otherwise. Both lead to the same result, namely \(c = x = 4\). The MFP solution merges the control flow at program point 4 and thus, loses all information, whereas the MOP solution merges both paths at the end and preserves the information.
3 Code Scheduling

The instruction on slide 31 are the following:

5: \( M[A1] = A1 \)
6: \( D2 = D2 + 1 \)
7: \( D3 = M[A1 + 12] \)
8: \( D3 = D3 + D1 \)

Since the algorithm traverses the instructions backwards, it starts with the last one. The initial sets \( \text{firstDefs} \) and \( \text{expUses} \) are:

\[
\text{firstDefs}_{\text{Init}} = \{(M, 9)\}
\]

\[
\text{expUses}_{\text{Init}} = \{(D3, 9), (A1, 9)\}
\]

1. Iteration:

\[
\text{conflict}(D3, 8, 9) \rightarrow (8, 9, \text{du})
\]

\[
\text{firstDefs} = \{(D3, 8), (M, 9)\}
\]

\[
\text{expUses} = \{(D1, 8), (D3, 8), (A1, 9)\}
\]

2. Iteration:

\[
\text{conflict}(A1, 7, 9)
\]

\[
\text{conflict}(D3, 7, 8) \rightarrow (7, 8, \text{du})
\]

\[
\text{conflict}(M, 7, 9) \rightarrow (7, 9, \text{ud})
\]

\[
\text{firstDefs} = \{(D3, 7), (M, 9)\}
\]

\[
\text{expUses} = \{(A1, 7), (A1, 9), (D1, 8), (M, 7)\}
\]

3. Iteration:

\[
\text{no conflicts checked}
\]

\[
\text{firstDefs} = \{(D2, 6), (D3, 7), (M, 9)\}
\]

\[
\text{expUses} = \{(A1, 7), (A1, 9), (D1, 8), (D2, 6), (M, 7)\}
\]

4. Iteration:

\[
\text{conflict}(A1, 5, 7)
\]

\[
\text{conflict}(A1, 5, 9)
\]

\[
\text{conflict}(M, 5, 7) \rightarrow (5, 7, \text{du})
\]

\[
\text{firstDefs} = \{(D2, 6), (D3, 7), (M, 5)\}
\]

\[
\text{expUses} = \{(A1, 5), (A1, 7), (A1, 9), (D1, 8), (D2, 6)\}
\]

The final sets \( \text{firstDefs} \) and \( \text{expUses} \) are:

\[
\text{firstDefs}_{\text{Final}} = \{(D2, 6), (D3, 7), (M, 5)\}
\]

\[
\text{expUses}_{\text{Final}} = \{(A1, 5), (A1, 7), (A1, 9), (D1, 8), (D2, 6)\}
\]