Example Language

Syntax of While-programs

$$
\begin{array}{lcl}
a & ::= & x \mid n \mid a_1 \; op_a \; a_2 \\
b & ::= & \texttt{true} \mid \texttt{false} \mid \texttt{not}\ b \mid b_1 \; op_b \; b_2 \mid a_1 \; op_r \; a_2 \\
S & ::= & [x := a]^\ell \mid [\texttt{skip}]^\ell \mid S_1; S_2 \mid \\
 & & \texttt{if}\ [b]^\ell\ \texttt{then}\ S_1\ \texttt{else}\ S_2 \mid \texttt{while}\ [b]^\ell\ \texttt{do}\ S
\end{array}
$$

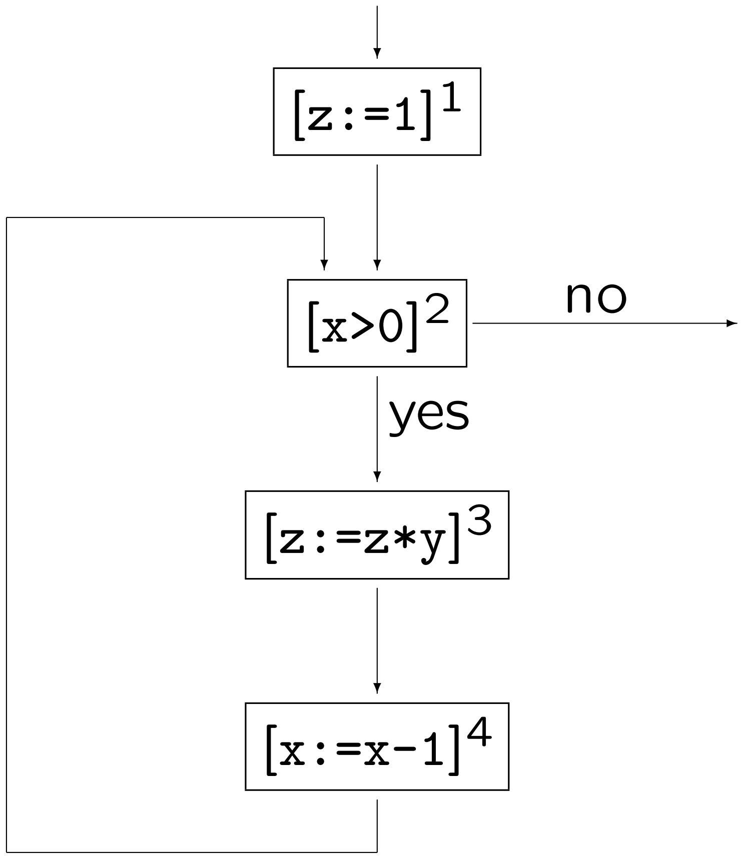
Example: $[\texttt{z:=1}]^1; \texttt{while}\ [\texttt{x>0}]^2\ \texttt{do}\ ([\texttt{z:=z*y}]^3; [\texttt{x:=x-1}]^4)$

Abstract syntax – parentheses are inserted to disambiguate the syntax

Building an “Abstract Flowchart”

Example: `[z:=1]`[1]; **while** `[x>0]`[2] **do** (`[z:=z*y]`[3]; `[x:=x-1]`[4])

$$
\begin{aligned}
\textit{init}(\cdots) &= 1 \\
\textit{final}(\cdots) &= \{2\} \\
\textit{labels}(\cdots) &= \{1,2,3,4\} \\
\textit{flow}(\cdots) &= \{(1,2),(2,3), \\
& (3,4),(4,2)\} \\
\textit{flow}^R(\cdots) &= \{(2,1),(2,4), \\
& (3,2),(4,3)\}
\end{aligned}
$$



Initial labels

init(S) is the label of the first elementary block of S:

$$\mathit{init} : \mathbf{Stmt} \to \mathbf{Lab}$$

$$\begin{array}{rcl}
\mathit{init}([x := a]^\ell) & = & \ell \\
\mathit{init}([\mathtt{skip}]^\ell) & = & \ell \\
\mathit{init}(S_1; S_2) & = & \mathit{init}(S_1) \\
\mathit{init}(\mathtt{if}\ [b]^\ell\ \mathtt{then}\ S_1\ \mathtt{else}\ S_2) & = & \ell \\
\mathit{init}(\mathtt{while}\ [b]^\ell\ \mathtt{do}\ S) & = & \ell
\end{array}$$

Example:

$$\mathit{init}([\mathtt{z:=1}]^1; \mathtt{while}\ [\mathtt{x>0}]^2\ \mathtt{do}\ ([\mathtt{z:=z*y}]^3; [\mathtt{x:=x-1}]^4)) = 1$$

Final labels

final(S) is the set of labels of the last elementary blocks of S:

$$\mathit{final} : \mathbf{Stmt} \to \mathcal{P}(\mathbf{Lab})$$

$$\begin{array}{rcl}
\mathit{final}([x := a]^\ell) & = & \{\ell\} \\
\mathit{final}([\mathtt{skip}]^\ell) & = & \{\ell\} \\
\mathit{final}(S_1; S_2) & = & \mathit{final}(S_2) \\
\mathit{final}(\mathtt{if}\ [b]^\ell\ \mathtt{then}\ S_1\ \mathtt{else}\ S_2) & = & \mathit{final}(S_1) \cup \mathit{final}(S_2) \\
\mathit{final}(\mathtt{while}\ [b]^\ell\ \mathtt{do}\ S) & = & \{\ell\}
\end{array}$$

Example:

$$\mathit{final}([\mathtt{z:=1}]^1; \mathtt{while}\ [\mathtt{x>0}]^2\ \mathtt{do}\ ([\mathtt{z:=z*y}]^3; [\mathtt{x:=x-1}]^4)) = \{2\}$$

Labels

labels(S) is the entire set of labels in the statement S:

$$\textit{labels} : \mathbf{Stmt} \rightarrow \mathcal{P}(\mathbf{Lab})$$

$$\begin{array}{rcl}
\textit{labels}([x := a]^\ell) & = & \{\ell\} \\
\textit{labels}([\texttt{skip}]^\ell) & = & \{\ell\} \\
\textit{labels}(S_1; S_2) & = & \textit{labels}(S_1) \cup \textit{labels}(S_2) \\
\textit{labels}(\texttt{if } [b]^\ell \texttt{ then } S_1 \texttt{ else } S_2) & = & \{\ell\} \cup \textit{labels}(S_1) \cup \textit{labels}(S_2) \\
\textit{labels}(\texttt{while } [b]^\ell \texttt{ do } S) & = & \{\ell\} \cup \textit{labels}(S)
\end{array}$$

Example

$$\textit{labels}([\texttt{z:=1}]^1; \texttt{while } [\texttt{x>0}]^2 \texttt{ do } ([\texttt{z:=z*y}]^3; [\texttt{x:=x-1}]^4)) = \{1, 2, 3, 4\}$$

Flows and reverse flows

flow(S) and *flow*$^R(S)$ are representations of how control flows in S:

$$\mathit{flow}, \mathit{flow}^R : \mathbf{Stmt} \to \mathcal{P}(\mathbf{Lab} \times \mathbf{Lab})$$

$$\begin{array}{rcl}
\mathit{flow}([x := a]^\ell) & = & \emptyset \\
\mathit{flow}([\mathtt{skip}]^\ell) & = & \emptyset \\
\mathit{flow}(S_1; S_2) & = & \mathit{flow}(S_1) \cup \mathit{flow}(S_2) \\
 & & \cup\ \{(\ell, \mathit{init}(S_2)) \mid \ell \in \mathit{final}(S_1)\} \\
\mathit{flow}(\mathtt{if}\ [b]^\ell\ \mathtt{then}\ S_1\ \mathtt{else}\ S_2) & = & \mathit{flow}(S_1) \cup \mathit{flow}(S_2) \\
 & & \cup\ \{(\ell, \mathit{init}(S_1)), (\ell, \mathit{init}(S_2))\} \\
\mathit{flow}(\mathtt{while}\ [b]^\ell\ \mathtt{do}\ S) & = & \mathit{flow}(S) \cup \{(\ell, \mathit{init}(S))\} \\
 & & \cup\ \{(\ell', \ell) \mid \ell' \in \mathit{final}(S)\}
\end{array}$$

$$\mathit{flow}^R(S) \;=\; \{(\ell, \ell') \mid (\ell', \ell) \in \mathit{flow}(S)\}$$

Elementary blocks

A statement consists of a set of *elementary blocks*

$$\mathit{blocks} : \mathbf{Stmt} \to \mathcal{P}(\mathbf{Blocks})$$

$$\begin{array}{rcl}
\mathit{blocks}([\mathtt{x} := a]^\ell) & = & \{[\mathtt{x} := a]^\ell\} \\
\mathit{blocks}([\mathtt{skip}]^\ell) & = & \{[\mathtt{skip}]^\ell\} \\
\mathit{blocks}(S_1; S_2) & = & \mathit{blocks}(S_1) \ \cup \ \mathit{blocks}(S_2) \\
\mathit{blocks}(\mathtt{if}\ [b]^\ell\ \mathtt{then}\ S_1\ \mathtt{else}\ S_2) & = & \{[b]^\ell\} \cup \mathit{blocks}(S_1) \cup \mathit{blocks}(S_2) \\
\mathit{blocks}(\mathtt{while}\ [b]^\ell\ \mathtt{do}\ S) & = & \{[b]^\ell\} \ \cup \ \mathit{blocks}(S)
\end{array}$$

A statement S is *label consistent* if and only if any two elementary statements $[S_1]^\ell$ and $[S_2]^\ell$ with the same label in S are equal: $S_1 = S_2$

A statement *where all labels are unique* is automatically label consistent

Intraprocedural Analysis

Classical analyses:

- Available Expressions Analysis
- Reaching Definitions Analysis
- Very Busy Expressions Analysis
- Live Variables Analysis

Derived analysis:

- Use-Definition and Definition-Use Analysis

Available Expressions Analysis

The aim of the *Available Expressions Analysis* is to determine

> For each program point, which expressions must have already been computed, and not later modified, on all paths to the program point.

Example:

point of interest

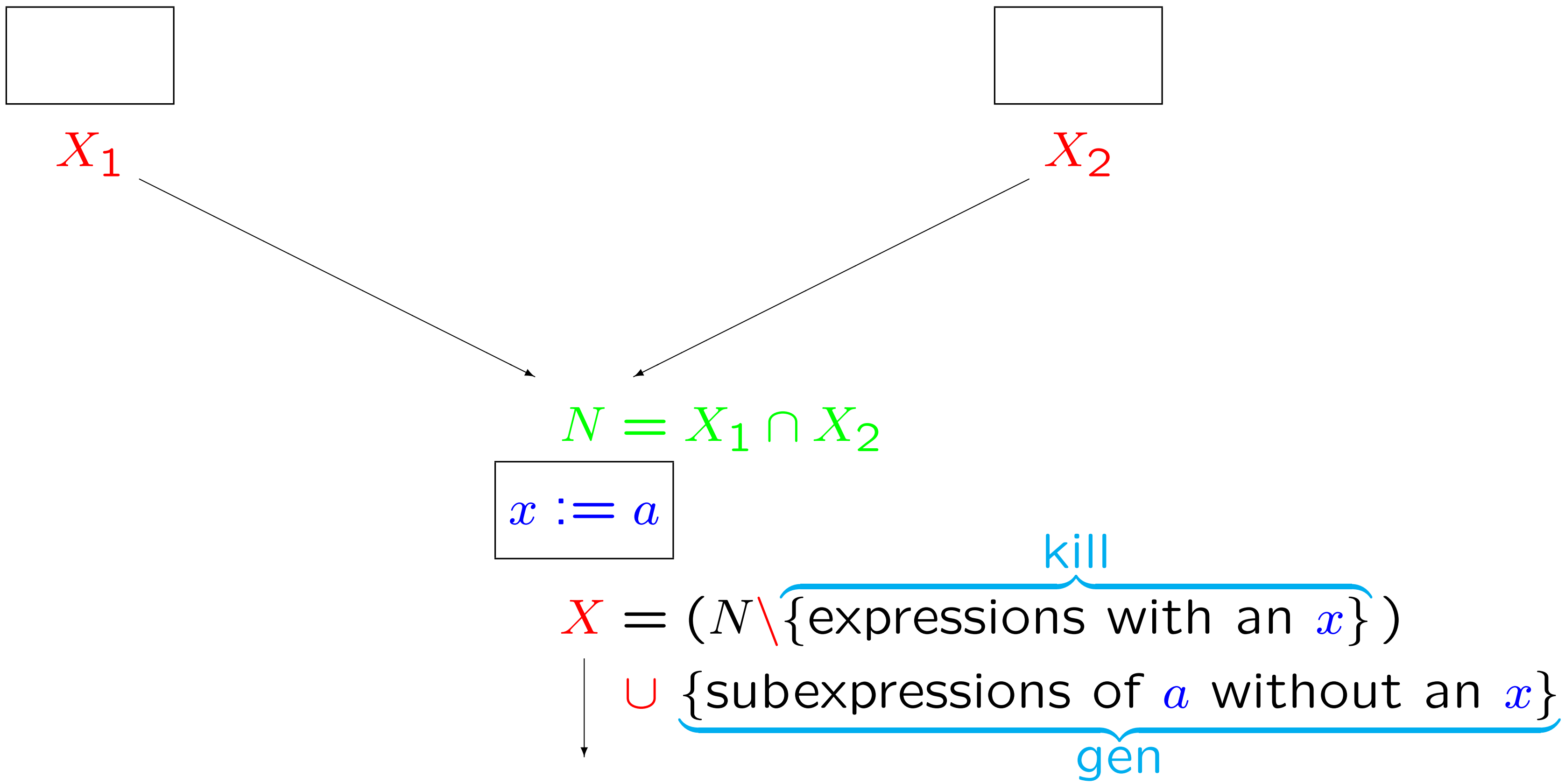
$$\Downarrow$$

$$[\mathtt{x}{:=}\,\mathtt{a+b}]^1;\,[\mathtt{y}{:=}\mathtt{a*b}]^2;\,\mathtt{while}\ [\mathtt{y}{>}\,\mathtt{a+b}]^3\ \mathtt{do}\ ([\mathtt{a}{:=}\mathtt{a+1}]^4;\,[\mathtt{x}{:=}\,\mathtt{a+b}]^5)$$

The analysis enables a transformation into

$$[\mathtt{x}{:=}\ \mathtt{a+b}]^1;\,[\mathtt{y}{:=}\mathtt{a*b}]^2;\,\mathtt{while}\ [\mathtt{y}{>}\,\mathtt{x}]^3\ \mathtt{do}\ ([\mathtt{a}{:=}\mathtt{a+1}]^4;\,[\mathtt{x}{:=}\ \mathtt{a+b}]^5)$$

Available Expressions Analysis – the basic idea



Available Expressions Analysis

kill and *gen* functions

$$
\begin{aligned}
\mathit{kill}_{\mathsf{AE}}([x := a]^{\ell}) &= \{a' \in \mathbf{AExp}_{\star} \mid x \in \mathit{FV}(a')\} \\
\mathit{kill}_{\mathsf{AE}}([\mathtt{skip}]^{\ell}) &= \emptyset \\
\mathit{kill}_{\mathsf{AE}}([b]^{\ell}) &= \emptyset \\
\\
\mathit{gen}_{\mathsf{AE}}([x := a]^{\ell}) &= \{a' \in \mathbf{AExp}(a) \mid x \notin \mathit{FV}(a')\} \\
\mathit{gen}_{\mathsf{AE}}([\mathtt{skip}]^{\ell}) &= \emptyset \\
\mathit{gen}_{\mathsf{AE}}([b]^{\ell}) &= \mathbf{AExp}(b)
\end{aligned}
$$

data flow equations: $\boxed{\mathsf{AE}^{=}}$

$$
\mathsf{AE}_{entry}(\ell) = \begin{cases} \emptyset & \text{if } \ell = \mathit{init}(S_{\star}) \\ \bigcap\{\mathsf{AE}_{exit}(\ell') \mid (\ell',\ell) \in \mathit{flow}(S_{\star})\} & \text{otherwise} \end{cases}
$$

$$
\begin{aligned}
\mathsf{AE}_{exit}(\ell) &= (\mathsf{AE}_{entry}(\ell) \backslash \mathit{kill}_{\mathsf{AE}}(B^{\ell})) \cup \mathit{gen}_{\mathsf{AE}}(B^{\ell}) \\
&\ \text{where } B^{\ell} \in \mathit{blocks}(S_{\star})
\end{aligned}
$$

Example:

$$[\texttt{x:=a+b}]^1; [\texttt{y:=a*b}]^2; \texttt{while } [\texttt{y>a+b}]^3 \texttt{ do } ([\texttt{a:=a+1}]^4; [\texttt{x:=a+b}]^5)$$

kill and *gen* functions:

ℓ	$\mathit{kill}_{\mathsf{AE}}(\ell)$	$\mathit{gen}_{\mathsf{AE}}(\ell)$
1	$\emptyset$	$\{\texttt{a+b}\}$
2	$\emptyset$	$\{\texttt{a*b}\}$
3	$\emptyset$	$\{\texttt{a+b}\}$
4	$\{\texttt{a+b}, \texttt{a*b}, \texttt{a+1}\}$	$\emptyset$
5	$\emptyset$	$\{\texttt{a+b}\}$

Example (cont.):

$$[\mathtt{x:=a+b}]^1; [\mathtt{y:=a*b}]^2; \mathtt{while}\ [\mathtt{y>a+b}]^3\ \mathtt{do}\ ([\mathtt{a:=a+1}]^4; [\mathtt{x:=a+b}]^5)$$

Equations:

$$\begin{aligned}
\mathsf{AE}_{entry}(1) &= \emptyset \\
\mathsf{AE}_{entry}(2) &= \mathsf{AE}_{exit}(1) \\
\mathsf{AE}_{entry}(3) &= \mathsf{AE}_{exit}(2) \cap \mathsf{AE}_{exit}(5) \\
\mathsf{AE}_{entry}(4) &= \mathsf{AE}_{exit}(3) \\
\mathsf{AE}_{entry}(5) &= \mathsf{AE}_{exit}(4)
\end{aligned}$$

$$\begin{aligned}
\mathsf{AE}_{exit}(1) &= \mathsf{AE}_{entry}(1) \cup \{\mathtt{a+b}\} \\
\mathsf{AE}_{exit}(2) &= \mathsf{AE}_{entry}(2) \cup \{\mathtt{a*b}\} \\
\mathsf{AE}_{exit}(3) &= \mathsf{AE}_{entry}(3) \cup \{\mathtt{a+b}\} \\
\mathsf{AE}_{exit}(4) &= \mathsf{AE}_{entry}(4) \backslash \{\mathtt{a+b}, \mathtt{a*b}, \mathtt{a+1}\} \\
\mathsf{AE}_{exit}(5) &= \mathsf{AE}_{entry}(5) \cup \{\mathtt{a+b}\}
\end{aligned}$$

Example (cont.):

$[\mathtt{x:=a+b}]^1; [\mathtt{y:=a*b}]^2; \mathtt{while}\ [\mathtt{y>a+b}]^3\ \mathtt{do}\ ([\mathtt{a:=a+1}]^4; [\mathtt{x:=a+b}]^5)$

Largest solution:

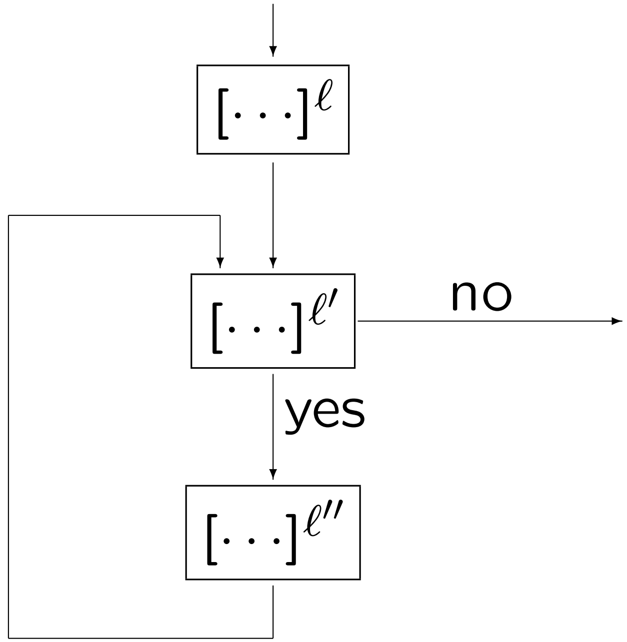
ℓ	$\mathsf{AE}_{entry}(\ell)$	$\mathsf{AE}_{exit}(\ell)$
1	$\emptyset$	$\{\mathtt{a+b}\}$
2	$\{\mathtt{a+b}\}$	$\{\mathtt{a+b},\ \mathtt{a*b}\}$
3	$\{\mathtt{a+b}\}$	$\{\mathtt{a+b}\}$
4	$\{\mathtt{a+b}\}$	$\emptyset$
5	$\emptyset$	$\{\mathtt{a+b}\}$

Why largest solution?

$$[\mathtt{z:=x+y}]^{\ell};\, \mathtt{while}\ [\mathtt{true}]^{\ell'}\ \mathtt{do}\ [\mathtt{skip}]^{\ell''}$$

Equations:

$$\begin{array}{rcl}
\mathsf{AE}_{entry}(\ell) & = & \emptyset \\
\mathsf{AE}_{entry}(\ell') & = & \mathsf{AE}_{exit}(\ell) \ \cap \ \mathsf{AE}_{exit}(\ell'') \\
\mathsf{AE}_{entry}(\ell'') & = & \mathsf{AE}_{exit}(\ell') \\
\mathsf{AE}_{exit}(\ell) & = & \mathsf{AE}_{entry}(\ell) \cup \{\mathtt{x+y}\} \\
\mathsf{AE}_{exit}(\ell') & = & \mathsf{AE}_{entry}(\ell') \\
\mathsf{AE}_{exit}(\ell'') & = & \mathsf{AE}_{entry}(\ell'')
\end{array}$$



After some simplification: $\mathsf{AE}_{entry}(\ell') = \{\mathtt{x+y}\} \cap \mathsf{AE}_{entry}(\ell')$

Two solutions to this equation: $\{\mathtt{x+y}\}$ and $\emptyset$

Reaching Definitions Analysis

The aim of the *Reaching Definitions Analysis* is to determine

> For each program point, which assignments may have been made and not overwritten, when program execution reaches this point along some path.

Example:

point of interest

$$\Downarrow$$

$$[\mathtt{x:=5}]^1;[\mathtt{y:=1}]^2;\mathtt{while}\ [\mathtt{x>1}]^3\ \mathtt{do}\ ([\mathtt{y:=x*y}]^4;[\mathtt{x:=x-1}]^5)$$

useful for definition-use chains and use-definition chains

Reaching Definitions Analysis – the basic idea

X_1 $\qquad\qquad\qquad\qquad$ X_2

$$N = X_1 \cup X_2$$

$$\boxed{[x := a]^\ell}$$

$$X = (N \setminus \overbrace{\{(x,?),(x,1),\cdots\}}^{\text{kill}}) \cup \underbrace{\{(x,\ell)\}}_{\text{gen}}$$

Reaching Definitions Analysis

kill and *gen* functions

$$
\begin{aligned}
\mathit{kill}_{\mathsf{RD}}([x := a]^\ell) &= \{(x, ?)\} \\
&\quad \cup \{(x, \ell') \mid B^{\ell'} \text{ is an assignment to } x \text{ in } S_\star\} \\
\mathit{kill}_{\mathsf{RD}}([\mathtt{skip}]^\ell) &= \emptyset \\
\mathit{kill}_{\mathsf{RD}}([b]^\ell) &= \emptyset \\
\\
\mathit{gen}_{\mathsf{RD}}([x := a]^\ell) &= \{(x, \ell)\} \\
\mathit{gen}_{\mathsf{RD}}([\mathtt{skip}]^\ell) &= \emptyset \\
\mathit{gen}_{\mathsf{RD}}([b]^\ell) &= \emptyset
\end{aligned}
$$

data flow equations: $\mathsf{RD}^=$

$$
\begin{aligned}
\mathsf{RD}_{entry}(\ell) &= \begin{cases} \{(x, ?) \mid x \in \mathit{FV}(S_\star)\} & \text{if } \ell = \mathit{init}(S_\star) \\ \bigcup\{\mathsf{RD}_{exit}(\ell') \mid (\ell', \ell) \in \mathit{flow}(S_\star)\} & \text{otherwise} \end{cases} \\
\\
\mathsf{RD}_{exit}(\ell) &= (\mathsf{RD}_{entry}(\ell) \backslash \mathit{kill}_{\mathsf{RD}}(B^\ell)) \cup \mathit{gen}_{\mathsf{RD}}(B^\ell) \\
&\quad \text{where } B^\ell \in \mathit{blocks}(S_\star)
\end{aligned}
$$

Example:

$$[\texttt{x:=5}]^1; [\texttt{y:=1}]^2; \texttt{while } [\texttt{x>1}]^3 \texttt{ do } ([\texttt{y:=x*y}]^4; [\texttt{x:=x-1}]^5)$$

kill and *gen* functions:

ℓ	$kill_{\mathsf{RD}}(\ell)$	$gen_{\mathsf{RD}}(\ell)$
1	$\{(x,?),(x,1),(x,5)\}$	$\{(x,1)\}$
2	$\{(y,?),(y,2),(y,4)\}$	$\{(y,2)\}$
3	$\emptyset$	$\emptyset$
4	$\{(y,?),(y,2),(y,4)\}$	$\{(y,4)\}$
5	$\{(x,?),(x,1),(x,5)\}$	$\{(x,5)\}$

Example (cont.):

$$[\texttt{x:=5}]^1; [\texttt{y:=1}]^2; \texttt{while } [\texttt{x>1}]^3 \texttt{ do } ([\texttt{y:=x*y}]^4; [\texttt{x:=x-1}]^5)$$

Equations:

$$\begin{aligned}
\mathsf{RD}_{entry}(1) &= \{(\mathrm{x},?),(\mathrm{y},?)\} \\
\mathsf{RD}_{entry}(2) &= \mathsf{RD}_{exit}(1) \\
\mathsf{RD}_{entry}(3) &= \mathsf{RD}_{exit}(2) \cup \mathsf{RD}_{exit}(5) \\
\mathsf{RD}_{entry}(4) &= \mathsf{RD}_{exit}(3) \\
\mathsf{RD}_{entry}(5) &= \mathsf{RD}_{exit}(4)
\end{aligned}$$

$$\begin{aligned}
\mathsf{RD}_{exit}(1) &= (\mathsf{RD}_{entry}(1)\backslash\{(\mathrm{x},?),(\mathrm{x},1),(\mathrm{x},5)\}) \cup \{(\mathrm{x},1)\} \\
\mathsf{RD}_{exit}(2) &= (\mathsf{RD}_{entry}(2)\backslash\{(\mathrm{y},?),(\mathrm{y},2),(\mathrm{y},4)\}) \cup \{(\mathrm{y},2)\} \\
\mathsf{RD}_{exit}(3) &= \mathsf{RD}_{entry}(3) \\
\mathsf{RD}_{exit}(4) &= (\mathsf{RD}_{entry}(4)\backslash\{(\mathrm{y},?),(\mathrm{y},2),(\mathrm{y},4)\}) \cup \{(\mathrm{y},4)\} \\
\mathsf{RD}_{exit}(5) &= (\mathsf{RD}_{entry}(5)\backslash\{(\mathrm{x},?),(\mathrm{x},1),(\mathrm{x},5)\}) \cup \{(\mathrm{x},5)\}
\end{aligned}$$

Example (cont.):

$$[\mathtt{x:=5}]^1; [\mathtt{y:=1}]^2; \mathtt{while}\ [\mathtt{x>1}]^3\ \mathtt{do}\ ([\mathtt{y:=}\ \mathtt{x*y}\ ]^4; [\mathtt{x:=x-1}]^5)$$

Smallest solution:

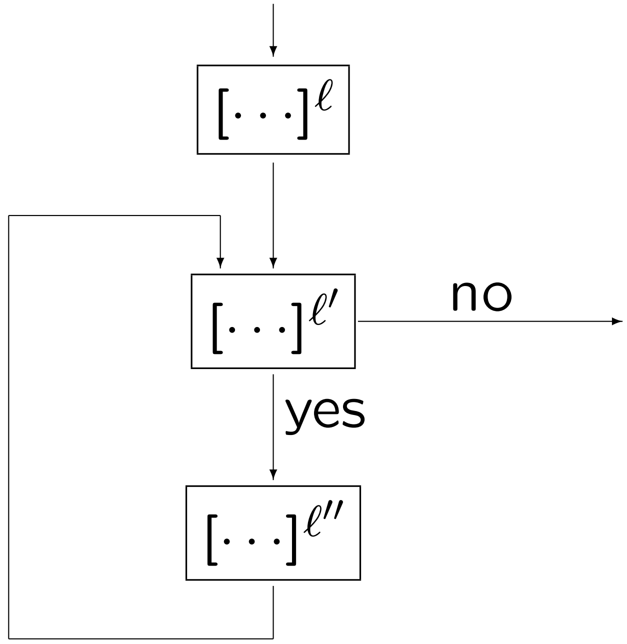
ℓ	$\mathsf{RD}_{entry}(\ell)$	$\mathsf{RD}_{exit}(\ell)$
1	$\{(\mathrm{x},?),(\mathrm{y},?)\}$	$\{(\mathrm{y},?),(\mathrm{x},1)\}$
2	$\{(\mathrm{y},?),(\mathrm{x},1)\}$	$\{(\mathrm{x},1),(\mathrm{y},2)\}$
3	$\{(\mathrm{x},1),(\mathrm{y},2),(\mathrm{y},4),(\mathrm{x},5)\}$	$\{(\mathrm{x},1),(\mathrm{y},2),(\mathrm{y},4),(\mathrm{x},5)\}$
4	$\{(\mathrm{x},1),(\mathrm{y},2),(\mathrm{y},4),(\mathrm{x},5)\}$	$\{(\mathrm{x},1),(\mathrm{y},4),(\mathrm{x},5)\}$
5	$\{(\mathrm{x},1),(\mathrm{y},4),(\mathrm{x},5)\}$	$\{(\mathrm{y},4),(\mathrm{x},5)\}$

Why smallest solution?

$$[\texttt{z:=x+y}]^{\ell}; \texttt{while } [\texttt{true}]^{\ell'} \texttt{ do } [\texttt{skip}]^{\ell''}$$

Equations:

$$\begin{array}{rcl}
\mathsf{RD}_{entry}(\ell) & = & \{(\mathtt{x},?),(\mathtt{y},?),(\mathtt{z},?)\} \\
\mathsf{RD}_{entry}(\ell') & = & \mathsf{RD}_{exit}(\ell) \cup \mathsf{RD}_{exit}(\ell'') \\
\mathsf{RD}_{entry}(\ell'') & = & \mathsf{RD}_{exit}(\ell') \\
\mathsf{RD}_{exit}(\ell) & = & (\mathsf{RD}_{entry}(\ell) \setminus \{(\mathtt{z},?)\}) \cup \{(\mathtt{z},\ell)\} \\
\mathsf{RD}_{exit}(\ell') & = & \mathsf{RD}_{entry}(\ell') \\
\mathsf{RD}_{exit}(\ell'') & = & \mathsf{RD}_{entry}(\ell'')
\end{array}$$



After some simplification: $\mathsf{RD}_{entry}(\ell') = \{(\mathtt{x},?),(\mathtt{y},?),(\mathtt{z},\ell)\} \cup \mathsf{RD}_{entry}(\ell')$

Many solutions to this equation: any superset of $\{(\mathtt{x},?),(\mathtt{y},?),(\mathtt{z},\ell)\}$

Very Busy Expressions Analysis

An expression is *very busy* at the exit from a label if, no matter what path is taken from the label, the expression is always used before any of the variables occurring in it are redefined.

The aim of the *Very Busy Expressions Analysis* is to determine

> For each program point, which expressions must be very busy at the exit from the point.

Example:

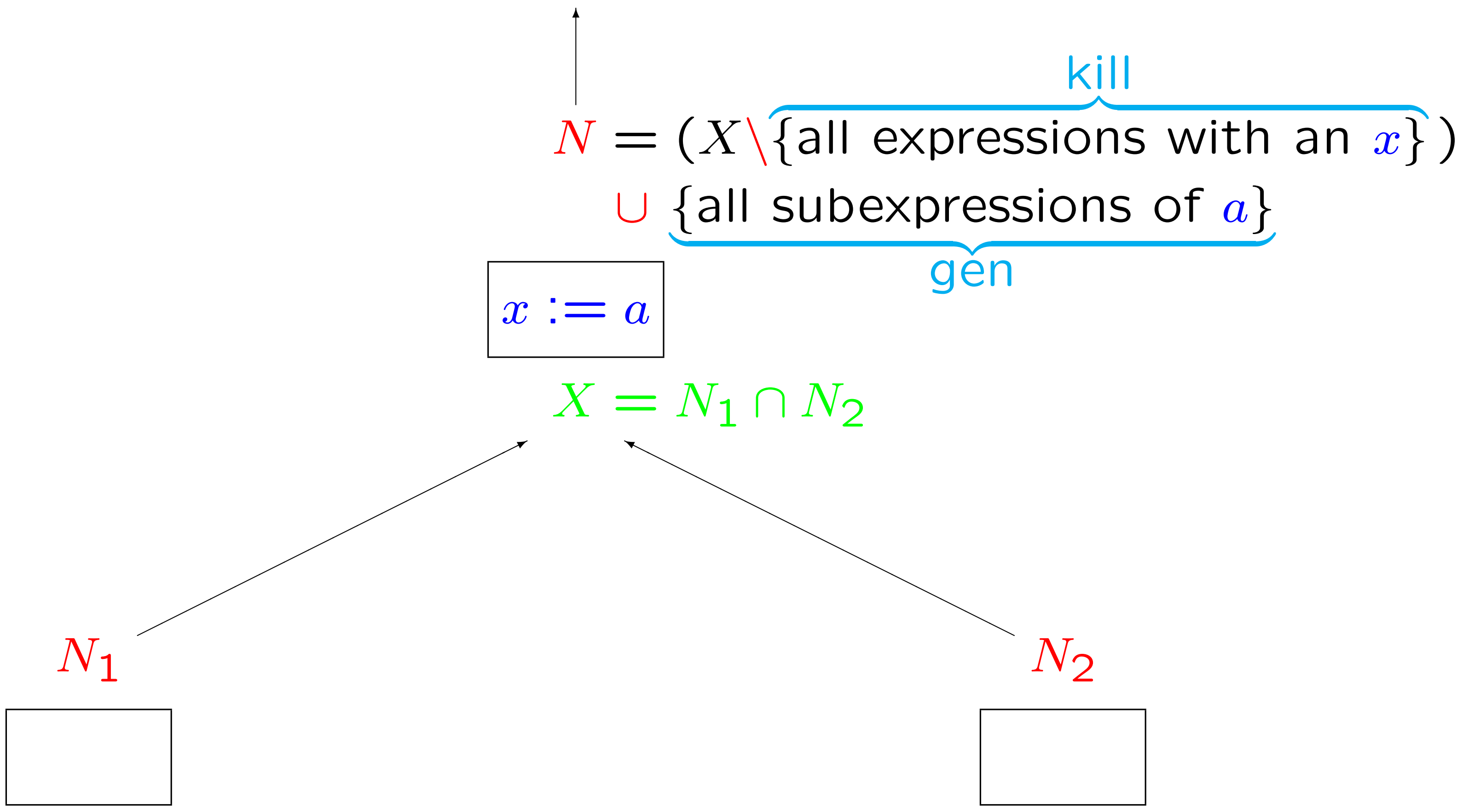
point of interest

$\Downarrow\texttt{if } [\texttt{a>b}]^1 \texttt{ then } ([\texttt{x:=b-a}]^2; [\texttt{y:=a-b}]^3) \texttt{ else } ([\texttt{y:=b-a}]^4; [\texttt{x:=a-b}]^5)$

The analysis enables a transformation into

$$[\texttt{t1:=b-a}]^A; [\texttt{t2:=b-a}]^B;$$
$$\texttt{if } [\texttt{a>b}]^1 \texttt{ then } ([\texttt{x:=t1}]^2; [\texttt{y:=t2}]^3) \texttt{ else } ([\texttt{y:=t1}]^4; [\texttt{x:=t2}]^5)$$

Very Busy Expressions Analysis – the basic idea



Very Busy Expressions Analysis

kill and *gen* functions

$$\begin{aligned}
\mathit{kill}_{\mathsf{VB}}([x := a]^\ell) &= \{a' \in \mathbf{AExp}_\star \mid x \in \mathit{FV}(a')\} \\
\mathit{kill}_{\mathsf{VB}}([\mathtt{skip}]^\ell) &= \emptyset \\
\mathit{kill}_{\mathsf{VB}}([b]^\ell) &= \emptyset \\
\\
\mathit{gen}_{\mathsf{VB}}([\mathtt{x} := a]^\ell) &= \mathbf{AExp}(a) \\
\mathit{gen}_{\mathsf{VB}}([\mathtt{skip}]^\ell) &= \emptyset \\
\mathit{gen}_{\mathsf{VB}}([b]^\ell) &= \mathbf{AExp}(b)
\end{aligned}$$

data flow equations: $\mathsf{VB}^=$

$$\mathsf{VB}_{exit}(\ell) = \begin{cases} \emptyset & \text{if } \ell \in \mathit{final}(S_\star) \\ \bigcap\{\mathsf{VB}_{entry}(\ell') \mid (\ell', \ell) \in \mathit{flow}^R(S_\star)\} & \text{otherwise} \end{cases}$$

$$\begin{aligned}
\mathsf{VB}_{entry}(\ell) = {} & (\mathsf{VB}_{exit}(\ell) \backslash \mathit{kill}_{\mathsf{VB}}(B^\ell)) \cup \mathit{gen}_{\mathsf{VB}}(B^\ell) \\
& \text{where } B^\ell \in \mathit{blocks}(S_\star)
\end{aligned}$$

Example:

$$\texttt{if } [\texttt{a>b}]^1 \texttt{ then } ([\texttt{x:=b-a}]^2; [\texttt{y:=a-b}]^3) \texttt{ else } ([\texttt{y:=b-a}]^4; [\texttt{x:=a-b}]^5)$$

kill and *gen* function:

ℓ	$\mathit{kill}_{\mathsf{VB}}(\ell)$	$\mathit{gen}_{\mathsf{VB}}(\ell)$
1	$\emptyset$	$\emptyset$
2	$\emptyset$	$\{\texttt{b-a}\}$
3	$\emptyset$	$\{\texttt{a-b}\}$
4	$\emptyset$	$\{\texttt{b-a}\}$
5	$\emptyset$	$\{\texttt{a-b}\}$

Example (cont.):

`if [a>b]`^{1} `then ([x:=b-a]`^{2}`; [y:=a-b]`^{3}`) else ([y:=b-a]`^{4}`; [x:=a-b]`^{5}`)`

Equations:

$$
\begin{aligned}
\mathsf{VB}_{entry}(1) &= \mathsf{VB}_{exit}(1) \\
\mathsf{VB}_{entry}(2) &= \mathsf{VB}_{exit}(2) \cup \{\texttt{b-a}\} \\
\mathsf{VB}_{entry}(3) &= \{\texttt{a-b}\} \\
\mathsf{VB}_{entry}(4) &= \mathsf{VB}_{exit}(4) \cup \{\texttt{b-a}\} \\
\mathsf{VB}_{entry}(5) &= \{\texttt{a-b}\}
\end{aligned}
$$

$$
\begin{aligned}
\mathsf{VB}_{exit}(1) &= \mathsf{VB}_{entry}(2) \cap \mathsf{VB}_{entry}(4) \\
\mathsf{VB}_{exit}(2) &= \mathsf{VB}_{entry}(3) \\
\mathsf{VB}_{exit}(3) &= \emptyset \\
\mathsf{VB}_{exit}(4) &= \mathsf{VB}_{entry}(5) \\
\mathsf{VB}_{exit}(5) &= \emptyset
\end{aligned}
$$

Example (cont.):

$\texttt{if [a>b]}^1\ \texttt{then ([x:=b-a]}^2\texttt{; [y:=a-b]}^3\texttt{) else ([y:=b-a]}^4\texttt{; [x:=a-b]}^5\texttt{)}$

Largest solution:

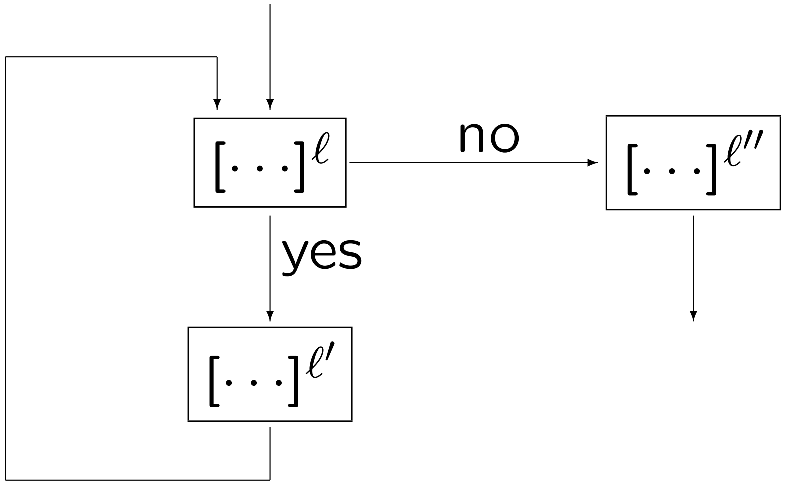
ℓ	$\mathsf{VB}_{entry}(\ell)$	$\mathsf{VB}_{exit}(\ell)$
1	$\{\texttt{a-b}, \texttt{b-a}\}$	$\{\texttt{a-b}, \texttt{b-a}\}$
2	$\{\texttt{a-b}, \texttt{b-a}\}$	$\{\texttt{a-b}\}$
3	$\{\texttt{a-b}\}$	$\emptyset$
4	$\{\texttt{a-b}, \texttt{b-a}\}$	$\{\texttt{a-b}\}$
5	$\{\texttt{a-b}\}$	$\emptyset$

Why largest solution?

$$(\texttt{while [x>1]}^{\ell} \texttt{ do [skip]}^{\ell'}); \texttt{[x:=x+1]}^{\ell''}$$

Equations:

$$
\begin{aligned}
\mathsf{VB}_{entry}(\ell) &= \mathsf{VB}_{exit}(\ell)\\
\mathsf{VB}_{entry}(\ell') &= \mathsf{VB}_{exit}(\ell')\\
\mathsf{VB}_{entry}(\ell'') &= \{\texttt{x+1}\}\\
\mathsf{VB}_{exit}(\ell) &= \mathsf{VB}_{entry}(\ell') \cap \mathsf{VB}_{entry}(\ell'')\\
\mathsf{VB}_{exit}(\ell') &= \mathsf{VB}_{entry}(\ell)\\
\mathsf{VB}_{exit}(\ell'') &= \emptyset
\end{aligned}
$$



After some simplifications: $\mathsf{VB}_{exit}(\ell) = \mathsf{VB}_{exit}(\ell) \cap \{\texttt{x+1}\}$

Two solutions to this equation: $\{\texttt{x+1}\}$ and $\emptyset$

Live Variables Analysis

A variable is *live* at the exit from a label if there is a path from the label to a use of the variable that does not re-define the variable.

The aim of the *Live Variables Analysis* is to determine

> For each program point, which variables may be live at the exit from the point.

Example:

point of interest

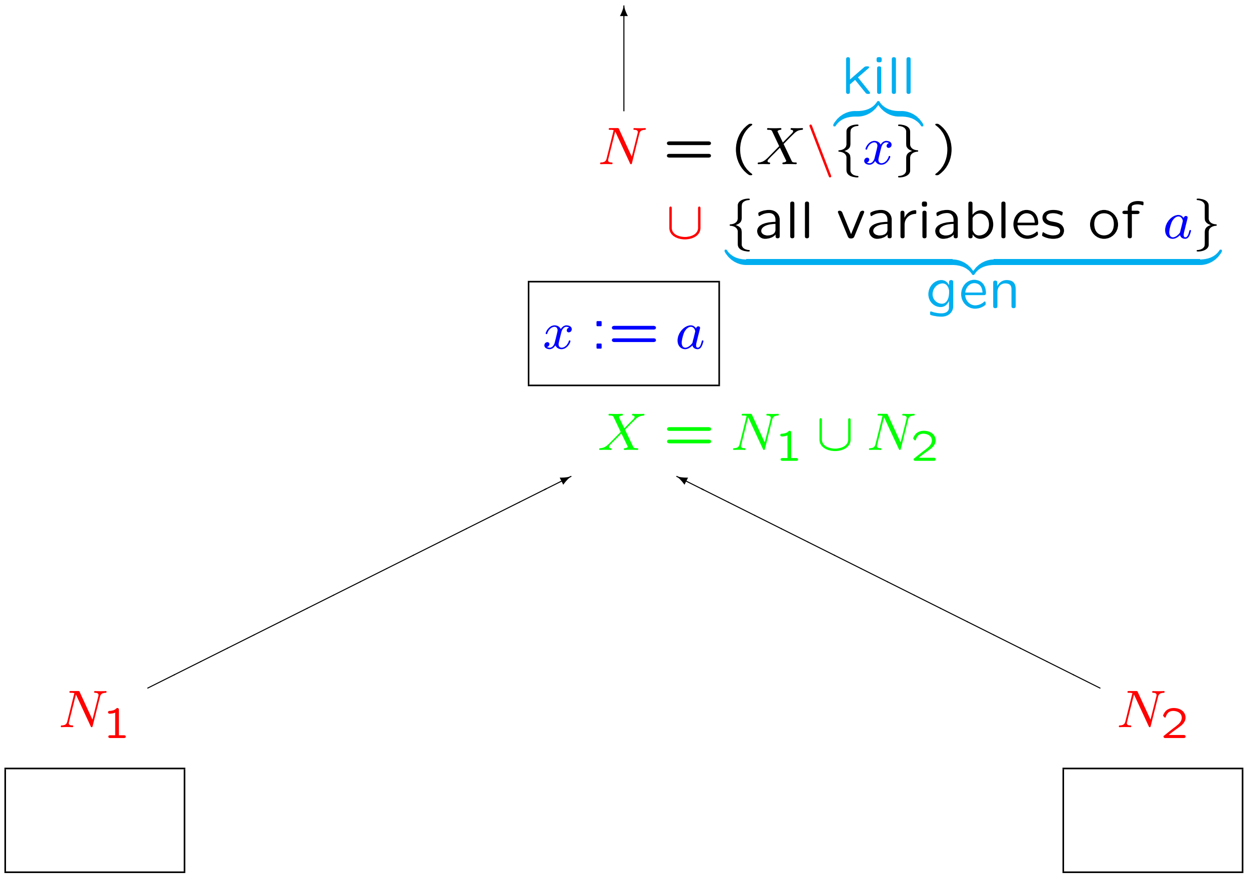
$\Downarrow$

$[\mathtt{x}:=2]^1; [\mathtt{y}:=4]^2; [\mathtt{x}:=1]^3; (\mathtt{if}\ [\mathtt{y}>\mathtt{x}]^4\ \mathtt{then}\ [\mathtt{z}:=\mathtt{y}]^5\ \mathtt{else}\ [\mathtt{z}:=\mathtt{y}*\mathtt{y}]^6); [\mathtt{x}:=\mathtt{z}]^7$

The analysis enables a transformation into

$$[\mathtt{y}:=4]^2; [\mathtt{x}:=1]^3; (\mathtt{if}\ [\mathtt{y}>\mathtt{x}]^4\ \mathtt{then}\ [\mathtt{z}:=\mathtt{y}]^5\ \mathtt{else}\ [\mathtt{z}:=\mathtt{y}*\mathtt{y}]^6); [\mathtt{x}:=\mathtt{z}]^7$$

Live Variables Analysis – the basic idea



Live Variables Analysis

kill and *gen* functions

$$
\begin{array}{rcl}
kill_{\mathsf{LV}}([x := a]^\ell) & = & \{x\} \\
kill_{\mathsf{LV}}([\texttt{skip}]^\ell) & = & \emptyset \\
kill_{\mathsf{LV}}([b]^\ell) & = & \emptyset \\
\\
gen_{\mathsf{LV}}([x := a]^\ell) & = & FV(a) \\
gen_{\mathsf{LV}}([\texttt{skip}]^\ell) & = & \emptyset \\
gen_{\mathsf{LV}}([b]^\ell) & = & FV(b)
\end{array}
$$

data flow equations: $\boxed{\mathsf{LV}^=}$

$$
\mathsf{LV}_{exit}(\ell) = \begin{cases} \emptyset & \text{if } \ell \in final(S_\star) \\ \bigcup\{\mathsf{LV}_{entry}(\ell') \mid (\ell', \ell) \in flow^R(S_\star)\} & \text{otherwise} \end{cases}
$$

$$
\mathsf{LV}_{entry}(\ell) = (\mathsf{LV}_{exit}(\ell) \backslash kill_{\mathsf{LV}}(B^\ell)) \cup gen_{\mathsf{LV}}(B^\ell)
$$

where $B^\ell \in blocks(S_\star)$

Example:

$[\texttt{x:=2}]^1; [\texttt{y:=4}]^2; [\texttt{x:=1}]^3; (\texttt{if } [\texttt{y>x}]^4 \texttt{ then } [\texttt{z:=y}]^5 \texttt{ else } [\texttt{z:=y*y}]^6); [\texttt{x:=z}]^7$

kill and *gen* functions:

ℓ	$\mathit{kill}_{\mathsf{LV}}(\ell)$	$\mathit{gen}_{\mathsf{LV}}(\ell)$
1	$\{\texttt{x}\}$	$\emptyset$
2	$\{\texttt{y}\}$	$\emptyset$
3	$\{\texttt{x}\}$	$\emptyset$
4	$\emptyset$	$\{\texttt{x}, \texttt{y}\}$
5	$\{\texttt{z}\}$	$\{\texttt{y}\}$
6	$\{\texttt{z}\}$	$\{\texttt{y}\}$
7	$\{\texttt{x}\}$	$\{\texttt{z}\}$

Example (cont.):

$$[\mathtt{x:=2}]^1; [\mathtt{y:=4}]^2; [\mathtt{x:=1}]^3; (\mathtt{if}\ [\mathtt{y>x}]^4\ \mathtt{then}\ [\mathtt{z:=y}]^5\ \mathtt{else}\ [\mathtt{z:=y*y}]^6); [\mathtt{x:=z}]^7$$

Equations:

$$\begin{array}{rcl}
\mathsf{LV}_{entry}(1) & = & \mathsf{LV}_{exit}(1) \backslash \{\mathtt{x}\} \\
\mathsf{LV}_{entry}(2) & = & \mathsf{LV}_{exit}(2) \backslash \{\mathtt{y}\} \\
\mathsf{LV}_{entry}(3) & = & \mathsf{LV}_{exit}(3) \backslash \{\mathtt{x}\} \\
\mathsf{LV}_{entry}(4) & = & \mathsf{LV}_{exit}(4) \cup \{\mathtt{x}, \mathtt{y}\} \\
\mathsf{LV}_{entry}(5) & = & (\mathsf{LV}_{exit}(5) \backslash \{\mathtt{z}\}) \cup \{\mathtt{y}\} \\
\mathsf{LV}_{entry}(6) & = & (\mathsf{LV}_{exit}(6) \backslash \{\mathtt{z}\}) \cup \{\mathtt{y}\} \\
\mathsf{LV}_{entry}(7) & = & \{\mathtt{z}\}
\end{array}
\qquad
\begin{array}{rcl}
\mathsf{LV}_{exit}(1) & = & \mathsf{LV}_{entry}(2) \\
\mathsf{LV}_{exit}(2) & = & \mathsf{LV}_{entry}(3) \\
\mathsf{LV}_{exit}(3) & = & \mathsf{LV}_{entry}(4) \\
\mathsf{LV}_{exit}(4) & = & \mathsf{LV}_{entry}(5) \cup \mathsf{LV}_{entry}(6) \\
\mathsf{LV}_{exit}(5) & = & \mathsf{LV}_{entry}(7) \\
\mathsf{LV}_{exit}(6) & = & \mathsf{LV}_{entry}(7) \\
\mathsf{LV}_{exit}(7) & = & \emptyset
\end{array}$$

Example (cont.):

$[\mathtt{x:=2}]^1; [\mathtt{y:=4}]^2; [\mathtt{x:=1}]^3; (\mathtt{if}\ [\mathtt{y>x}]^4\ \mathtt{then}\ [\mathtt{z:=y}]^5\ \mathtt{else}\ [\mathtt{z:=y*y}]^6); [\mathtt{x:=z}]^7$

Smallest solution:

ℓ	$\mathsf{LV}_{entry}(\ell)$	$\mathsf{LV}_{exit}(\ell)$
1	$\emptyset$	$\emptyset$
2	$\emptyset$	$\{\mathtt{y}\}$
3	$\{\mathtt{y}\}$	$\{\mathtt{x}, \mathtt{y}\}$
4	$\{\mathtt{x}, \mathtt{y}\}$	$\{\mathtt{y}\}$
5	$\{\mathtt{y}\}$	$\{\mathtt{z}\}$
6	$\{\mathtt{y}\}$	$\{\mathtt{z}\}$
7	$\{\mathtt{z}\}$	$\emptyset$

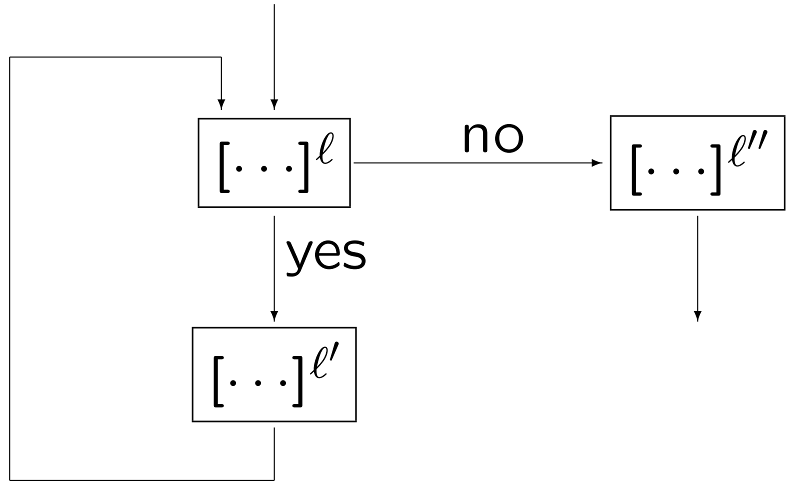
Why smallest solution?

$$(\texttt{while } [\texttt{x>1}]^{\ell} \texttt{ do } [\texttt{skip}]^{\ell'}); [\texttt{x:=x+1}]^{\ell''}$$

Equations:

$$\begin{aligned}
\mathsf{LV}_{entry}(\ell) &= \mathsf{LV}_{exit}(\ell) \cup \{\texttt{x}\} \\
\mathsf{LV}_{entry}(\ell') &= \mathsf{LV}_{exit}(\ell') \\
\mathsf{LV}_{entry}(\ell'') &= \{\texttt{x}\}
\end{aligned}$$

$$\begin{aligned}
\mathsf{LV}_{exit}(\ell) &= \mathsf{LV}_{entry}(\ell') \cup \mathsf{LV}_{entry}(\ell'') \\
\mathsf{LV}_{exit}(\ell') &= \mathsf{LV}_{entry}(\ell) \\
\mathsf{LV}_{exit}(\ell'') &= \emptyset
\end{aligned}$$



After some calculations: $\mathsf{LV}_{exit}(\ell) = \mathsf{LV}_{exit}(\ell) \cup \{\texttt{x}\}$

Many solutions to this equation: any superset of $\{\texttt{x}\}$