# Data flow analysis

Thomas Jensen

INRIA

Ecole Jeunes Chercheurs en Programmation
May 2013

# Program analysis

Goal: deduce mechanically properties about the program behaviour without executing it.

Application area: compilers, code optimisation, program verification, debugging...

3 rules:

1. The analyser must terminate;
2. The computed information must be correct;
3. It is allowed to return an approximative description of the program behaviour.

# Static vs. dynamic

Static analysis:

- Work done at compile-time
- Characterizes all executions
- Conservative: approximates concrete program states

Dynamic analysis:

- Run-time overhead
- Characterizes one or a few executions
- Precise: knows the concrete program state
- Can't "look into the future"

# Why abstraction?

The bad news: Rice's theorem:

*For a Turing-complete programming language, for any non-trivial property, the question of whether the computation of a given program satisfies this property is undecidable.*

Solutions:

- verify a model of the program (model checking)
- verify the program interactively with the help of the user (deductive methods)
- computes only an approximation of the behavior of the program
  - Rice's theorem for static analyses:

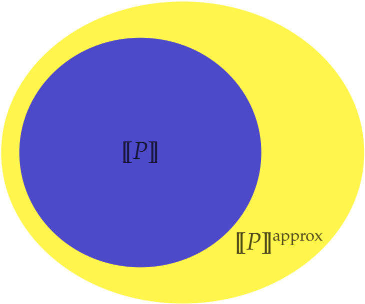
    *No static analysis can prove a non-trivial property for any programs in a finite time.*

  - It does not mean that it is impossible for *some* programs!

    ASTRÉE[1] analyses electric flight control codes of Airbus (~ 1 M loc)

[1] `http://www.astree.ens.fr/`

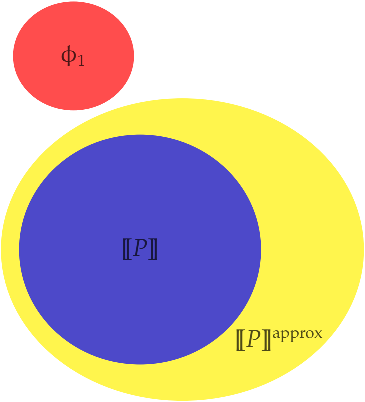
# A static analysis computes an approximation[2]




| | | |
|---|---|---|
| $[\![P]\!]$: | concrete semantics (e.g. set of reachable states) | (not computable) |
| $[\![P]\!]^{\text{approx}}$: | analyser result (here over-approximation) | (computable) |

[2] cf `http://www.astree.ens.fr/IntroAbsInt.html`

# A static analysis computes an approximation[2]



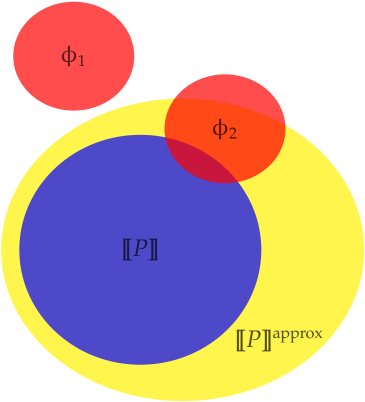

- $P$ is safe w.r.t. $\phi_1$ and the analyser proves it

$$[\![P]\!] \cap \phi_1 = \emptyset \qquad [\![P]\!]^{\text{approx}} \cap \phi_1 = \emptyset$$

| | | |
|---|---|---|
| $[\![P]\!]$: | concrete semantics (e.g. set of reachable states) | (not computable) |
| $\phi_1$: | erroneous/dangerous set of states | (computable) |
| $[\![P]\!]^{\text{approx}}$: | analyser result (here over-approximation) | (computable) |

[2] cf `http://www.astree.ens.fr/IntroAbsInt.html`

# A static analysis computes an approximation[2]




- $P$ is safe w.r.t. $\phi_1$ and the analyser proves it

$$[\![P]\!] \cap \phi_1 = \emptyset \qquad [\![P]\!]^{\text{approx}} \cap \phi_1 = \emptyset$$

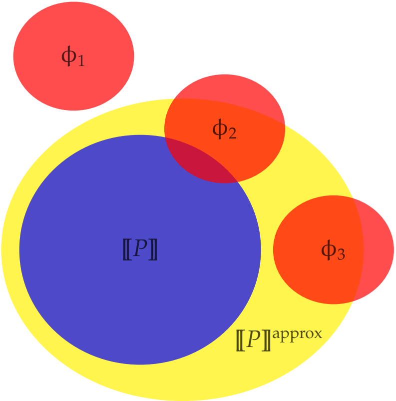
- $P$ is unsafe w.r.t. $\phi_2$ and the analyser warns about it

$$[\![P]\!] \cap \phi_2 \neq \emptyset \qquad [\![P]\!]^{\text{approx}} \cap \phi_2 \neq \emptyset$$

| | | |
|---|---|---|
| $[\![P]\!]$: | concrete semantics (e.g. set of reachable states) | (not computable) |
| $\phi_1, \phi_2$: | erroneous/dangerous set of states | (computable) |
| $[\![P]\!]^{\text{approx}}$: | analyser result (here over-approximation) | (computable) |

[2] cf `http://www.astree.ens.fr/IntroAbsInt.html`

# A static analysis computes an approximation[2]




- $P$ is safe w.r.t. $\phi_1$ and the analyser proves it

$$[\![P]\!] \cap \phi_1 = \emptyset \qquad [\![P]\!]^{\text{approx}} \cap \phi_1 = \emptyset$$

- $P$ is unsafe w.r.t. $\phi_2$ and the analyser warns about it

$$[\![P]\!] \cap \phi_2 \neq \emptyset \qquad [\![P]\!]^{\text{approx}} \cap \phi_2 \neq \emptyset$$

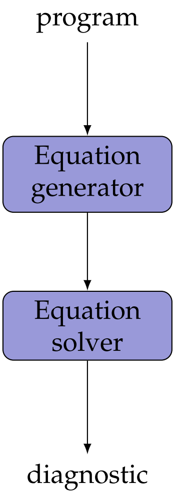
- but $P$ is safe w.r.t. $\phi_3$ and the analyser can't prove it (this is called a *false alarm*)

$$[\![P]\!] \cap \phi_3 = \emptyset \qquad [\![P]\!]^{\text{approx}} \cap \phi_3 \neq \emptyset$$

| | | |
|---|---|---|
| $[\![P]\!]$: | concrete semantics (e.g. set of reachable states) | (not computable) |
| $\phi_1, \phi_2, \phi_3$: | erroneous/dangerous set of states | (computable) |
| $[\![P]\!]^{\text{approx}}$: | analyser result (here over-approximation) | (computable) |

[2] cf `http://www.astree.ens.fr/IntroAbsInt.html`

# Common structure of analyses




An analysis can be separated into two parts:

1. From a program description, producing an equation system (analysis specification)
   - the solutions of the system must be proved correct w.r.t. the program semantics
2. Solving the system
   - *fixpoint* iterations in *lattice* structures

# Dataflow analysis: examples

Reachable definitions : May a definition reach a given point ?
(Dependency analysis between instructions)

Available expressions : What are the expressions already computed at a given point ?
(Re-use of expression computations)

Live variables : Is a variable used in the future ?
(Assignments deletion, Register allocation)

# Reachable definition analysis

Determine the set of definitions (assignments) that may reach a program point

Factorial function :

```
1. y := x;
2. z := 1;
3. while y > 1 do
4.   z := z * y;
5.   y := y - 1;
   end
6. y := 0;
```

At point 4, the definition that occurs at labels 1, 2, 4 and 5 are reachable (not for label 6).

# Reachable definition analysis

A definition is represented by a couple $(v, l) \in Var \times Lab^?$ with $Lab^? = Lab \cup \{?\}$.

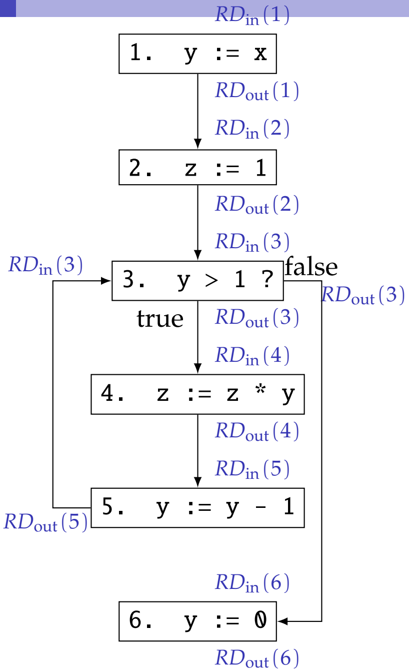
$$
\begin{array}{lcl}
(v, l) & : & \text{"the variable } v \text{ has been defined at program point } l \\
 & & \quad \text{and has not been modified since"} \\
(v, ?) & : & \text{"the variable } v \text{ is not initialised"}
\end{array}
$$

We compute two sets at each label (program point) $l$:

$$
\begin{array}{lcl}
RD_{\text{in}}(l) & = & \text{the definitions that enter in } l \text{ (}\textit{i.e.}\text{reachable)} \\
RD_{\text{out}}(l) & = & \text{the definitions that exit from } l \text{ (auxiliary set)}
\end{array}
$$

Each instruction define some relations between theses set of definitions

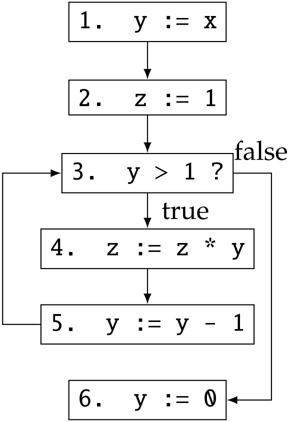
```
1. y := x;
2. z := 1;
3. while y > 1 do
4.     z := z * y;
5.     y := y - 1;
   end
6. y := 0;
```



# Reachable definition analysis: equations (1)

An assignment deletes the previous definitions of the assigned variable.

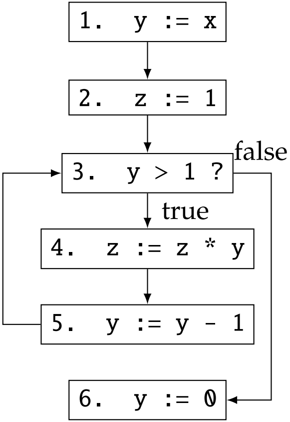
$$RD_{out}(1) = RD_{in}(1) \setminus \{(\mathbf{y}, l) \mid l \in Lab^?\} \cup \{(\mathbf{y}, 1)\}$$
$$RD_{out}(2) = RD_{in}(2) \setminus \{(\mathbf{z}, l) \mid l \in Lab^?\} \cup \{(\mathbf{z}, 2)\}$$
$$RD_{out}(3) = RD_{in}(3)$$
$$RD_{out}(4) = RD_{in}(4) \setminus \{(\mathbf{z}, l) \mid l \in Lab^?\} \cup \{(\mathbf{z}, 4)\}$$
$$RD_{out}(5) = RD_{in}(5) \setminus \{(\mathbf{y}, l) \mid l \in Lab^?\} \cup \{(\mathbf{y}, 5)\}$$
$$RD_{out}(6) = RD_{in}(6) \setminus \{(\mathbf{y}, l) \mid l \in Lab^?\} \cup \{(\mathbf{y}, 6)\}$$



# Reachable definition analysis: equations (2)

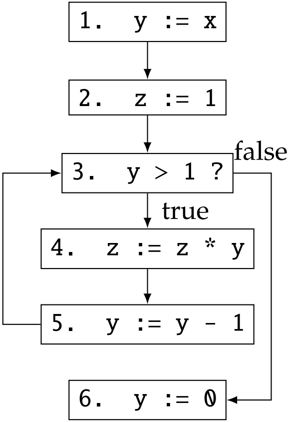
Definitions that are reachable after an instruction, are reachable before the next instruction.

$$RD_{in}(1) = \{(v, ?) \mid v \in Var\}$$
$$RD_{in}(2) = RD_{out}(1)$$
$$RD_{in}(3) = RD_{out}(2) \cup RD_{out}(5)$$
$$RD_{in}(4) = RD_{out}(3)$$
$$RD_{in}(5) = RD_{out}(4)$$
$$RD_{in}(6) = RD_{out}(3)$$



# Reachable definition analysis : a solution

$$
\begin{aligned}
RD_{\text{in}}(1) &= \{(x,?),(y,?),(z,?)\}\\
RD_{\text{in}}(2) &= \{(x,?),(y,1),(z,?)\}\\
RD_{\text{in}}(3) &= \{(x,?),(y,1),(y,5),(z,2),(z,4)\}\\
RD_{\text{in}}(4) &= \{(x,?),(y,1),(y,5),(z,2),(z,4)\}\\
RD_{\text{in}}(5) &= \{(x,?),(y,1),(y,5),(z,4)\}\\
RD_{\text{in}}(6) &= \{(x,?),(y,1),(y,5),(z,2),(z,4)\}\\
RD_{\text{out}}(1) &= \{(x,?),(y,1),(z,?)\}\\
RD_{\text{out}}(2) &= \{(x,?),(y,1),(z,2)\}\\
RD_{\text{out}}(3) &= \{(x,?),(y,1),(y,5),(z,2),(z,4)\}\\
RD_{\text{out}}(4) &= \{(x,?),(y,1),(y,5),(z,4)\}\\
RD_{\text{out}}(5) &= \{(x,?),(y,5),(z,4)\}\\
RD_{\text{out}}(6) &= \{(x,?),(y,6),(z,2),(z,4)\}
\end{aligned}
$$




We observe that $(y,1),(y,5) \in RD_{\text{in}}(6)$.

# Reachable definition analysis : iterative computation

The solution can be computed by iteration. $RD_{\text{in}}(l)$ and $RD_{\text{out}}(l)$ are initialised with $\emptyset$ and their values are recomputed until stabilisation.

Equations : $\vec{RD} = F(\vec{RD})$

$$
\begin{array}{llll}
RD_{in}(1) = \{(v, ?) \mid v \in Var\} & (e_1) & RD_{out}(1) = RD_{in}(1) \setminus \{(\mathbf{y}, l) \mid l \in Lab^?\} \cup \{(\mathbf{y}, 1)\} & (s_1) \\
RD_{in}(2) = RD_{out}(1) & (e_2) & RD_{out}(2) = RD_{in}(2) \setminus \{(\mathbf{z}, l) \mid l \in Lab^?\} \cup \{(\mathbf{z}, 2)\} & (s_2) \\
RD_{in}(3) = RD_{out}(2) \cup RD_{out}(5) & (e_3) & RD_{out}(3) = RD_{in}(3) & (s_3) \\
RD_{in}(4) = RD_{out}(3) & (e_4) & RD_{out}(4) = RD_{in}(4) \setminus \{(\mathbf{z}, l) \mid l \in Lab^?\} \cup \{(\mathbf{z}, 4)\} & (s_4) \\
RD_{in}(5) = RD_{out}(4) & (e_5) & RD_{out}(5) = RD_{in}(5) \setminus \{(\mathbf{y}, l) \mid l \in Lab^?\} \cup \{(\mathbf{y}, 5)\} & (s_5) \\
RD_{in}(6) = RD_{out}(3) & (e_6) & RD_{out}(6) = RD_{in}(6) \setminus \{(\mathbf{y}, l) \mid l \in Lab^?\} \cup \{(\mathbf{y}, 6)\} & (s_6)
\end{array}
$$

Iteration 0: $\vec{\emptyset}$

$$
\begin{array}{llll}
RD_{in}(1) = & \emptyset & RD_{out}(1) = & \emptyset \\
RD_{in}(2) = & \emptyset & RD_{out}(2) = & \emptyset \\
RD_{in}(3) = & \emptyset & RD_{out}(3) = & \emptyset \\
RD_{in}(4) = & \emptyset & RD_{out}(4) = & \emptyset \\
RD_{in}(5) = & \emptyset & RD_{out}(5) = & \emptyset \\
RD_{in}(6) = & \emptyset & RD_{out}(6) = & \emptyset
\end{array}
$$

# Reachable definition analysis : iterative computation

The solution can be computed by iteration. $RD_{\text{in}}(l)$ and $RD_{\text{out}}(l)$ are initialised with $\emptyset$ and their values are recomputed until stabilisation.

Equations : $\vec{RD} = F(\vec{RD})$

$$
\begin{array}{lll}
RD_{\text{in}}(1) = \{(v,?) \mid v \in Var\} & (e_1) & RD_{\text{out}}(1) = RD_{\text{in}}(1) \setminus \{(y,l) \mid l \in Lab^?\} \cup \{(y,1)\} \; (s_1) \\
RD_{\text{in}}(2) = RD_{\text{out}}(1) & (e_2) & RD_{\text{out}}(2) = RD_{\text{in}}(2) \setminus \{(z,l) \mid l \in Lab^?\} \cup \{(z,2)\} \; (s_2) \\
RD_{\text{in}}(3) = RD_{\text{out}}(2) \cup RD_{\text{out}}(5) & (e_3) & RD_{\text{out}}(3) = RD_{\text{in}}(3) \; (s_3) \\
RD_{\text{in}}(4) = RD_{\text{out}}(3) & (e_4) & RD_{\text{out}}(4) = RD_{\text{in}}(4) \setminus \{(z,l) \mid l \in Lab^?\} \cup \{(z,4)\} \; (s_4) \\
RD_{\text{in}}(5) = RD_{\text{out}}(4) & (e_5) & RD_{\text{out}}(5) = RD_{\text{in}}(5) \setminus \{(y,l) \mid l \in Lab^?\} \cup \{(y,5)\} \; (s_5) \\
RD_{\text{in}}(6) = RD_{\text{out}}(3) & (e_6) & RD_{\text{out}}(6) = RD_{\text{in}}(6) \setminus \{(y,l) \mid l \in Lab^?\} \cup \{(y,6)\} \; (s_6)
\end{array}
$$

Iteration 1: $F(\vec{\emptyset})$

$$
\begin{array}{llll}
RD_{\text{in}}(1) = & \{(x,?),(y,?),(z,?)\} & RD_{\text{out}}(1) = & \{(y,1)\} \\
RD_{\text{in}}(2) = & \emptyset & RD_{\text{out}}(2) = & \{(z,2)\} \\
RD_{\text{in}}(3) = & \emptyset & RD_{\text{out}}(3) = & \emptyset \\
RD_{\text{in}}(4) = & \emptyset & RD_{\text{out}}(4) = & \{(z,4)\} \\
RD_{\text{in}}(5) = & \emptyset & RD_{\text{out}}(5) = & \{(y,5)\} \\
RD_{\text{in}}(6) = & \emptyset & RD_{\text{out}}(6) = & \{(y,6)\}
\end{array}
$$

# Reachable definition analysis : iterative computation

The solution can be computed by iteration. $RD_{in}(l)$ and $RD_{out}(l)$ are initialised with $\emptyset$ and their values are recomputed until stabilisation.

Equations : $\vec{RD} = F(\vec{RD})$

$$
\begin{array}{llll}
RD_{in}(1) = \{(v,?) \mid v \in Var\} & (e_1) & RD_{out}(1) = RD_{in}(1) \setminus \{(y,l) \mid l \in Lab^?\} \cup \{(y,1)\} & (s_1) \\
RD_{in}(2) = RD_{out}(1) & (e_2) & RD_{out}(2) = RD_{in}(2) \setminus \{(z,l) \mid l \in Lab^?\} \cup \{(z,2)\} & (s_2) \\
RD_{in}(3) = RD_{out}(2) \cup RD_{out}(5) & (e_3) & RD_{out}(3) = RD_{in}(3) & (s_3) \\
RD_{in}(4) = RD_{out}(3) & (e_4) & RD_{out}(4) = RD_{in}(4) \setminus \{(z,l) \mid l \in Lab^?\} \cup \{(z,4)\} & (s_4) \\
RD_{in}(5) = RD_{out}(4) & (e_5) & RD_{out}(5) = RD_{in}(5) \setminus \{(y,l) \mid l \in Lab^?\} \cup \{(y,5)\} & (s_5) \\
RD_{in}(6) = RD_{out}(3) & (e_6) & RD_{out}(6) = RD_{in}(6) \setminus \{(y,l) \mid l \in Lab^?\} \cup \{(y,6)\} & (s_6)
\end{array}
$$

Iteration 2: $F^2(\vec{\emptyset})$

$$
\begin{array}{llll}
RD_{in}(1) = & \{(x,?),(y,?),(z,?)\} & RD_{out}(1) = & \{(x,?),(y,1),(z,?)\} \\
RD_{in}(2) = & \{(y,1)\} & RD_{out}(2) = & \{(z,2)\} \\
RD_{in}(3) = & \{(y,5),(z,2)\} & RD_{out}(3) = & \emptyset \\
RD_{in}(4) = & \emptyset & RD_{out}(4) = & \{(z,4)\} \\
RD_{in}(5) = & \{(z,4)\} & RD_{out}(5) = & \{(y,5)\} \\
RD_{in}(6) = & \emptyset & RD_{out}(6) = & \{(y,6)\}
\end{array}
$$

# Reachable definition analysis : iterative computation

The solution can be computed by iteration. $RD_{\text{in}}(l)$ and $RD_{\text{out}}(l)$ are initialised with $\emptyset$ and their values are recomputed until stabilisation.

Equations : $\vec{RD} = F(\vec{RD})$

| | | | |
|---|---|---|---|
| $RD_{in}(1) = \{(v,?) \mid v \in Var\}$ | $(e_1)$ | $RD_{out}(1) = RD_{in}(1) \setminus \{(y,l) \mid l \in Lab^?\} \cup \{(y,1)\}$ | $(s_1)$ |
| $RD_{in}(2) = RD_{out}(1)$ | $(e_2)$ | $RD_{out}(2) = RD_{in}(2) \setminus \{(z,l) \mid l \in Lab^?\} \cup \{(z,2)\}$ | $(s_2)$ |
| $RD_{in}(3) = RD_{out}(2) \cup RD_{out}(5)$ | $(e_3)$ | $RD_{out}(3) = RD_{in}(3)$ | $(s_3)$ |
| $RD_{in}(4) = RD_{out}(3)$ | $(e_4)$ | $RD_{out}(4) = RD_{in}(4) \setminus \{(z,l) \mid l \in Lab^?\} \cup \{(z,4)\}$ | $(s_4)$ |
| $RD_{in}(5) = RD_{out}(4)$ | $(e_5)$ | $RD_{out}(5) = RD_{in}(5) \setminus \{(y,l) \mid l \in Lab^?\} \cup \{(y,5)\}$ | $(s_5)$ |
| $RD_{in}(6) = RD_{out}(3)$ | $(e_6)$ | $RD_{out}(6) = RD_{in}(6) \setminus \{(y,l) \mid l \in Lab^?\} \cup \{(y,6)\}$ | $(s_6)$ |

Iteration 3: $F^3(\vec{\emptyset})$

| | | | |
|---|---|---|---|
| $RD_{in}(1) =$ | $\{(x,?),(y,?),(z,?)\}$ | $RD_{out}(1) =$ | $\{(x,?),(y,1),(z,?)\}$ |
| $RD_{in}(2) =$ | $\{(x,?),(y,1),(z,?)\}$ | $RD_{out}(2) =$ | $\{(y,1),(z,2)\}$ |
| $RD_{in}(3) =$ | $\{(y,5),(z,2)\}$ | $RD_{out}(3) =$ | $\{(y,5),(z,2)\}$ |
| $RD_{in}(4) =$ | $\emptyset$ | $RD_{out}(4) =$ | $\{(z,4)\}$ |
| $RD_{in}(5) =$ | $\{(z,4)\}$ | $RD_{out}(5) =$ | $\{(y,5),(z,4)\}$ |
| $RD_{in}(6) =$ | $\emptyset$ | $RD_{out}(6) =$ | $\{(y,6)\}$ |

# Reachable definition analysis : iterative computation

The solution can be computed by iteration. $RD_{\text{in}}(l)$ and $RD_{\text{out}}(l)$ are initialised with $\emptyset$ and their values are recomputed until stabilisation.

Equations : $\vec{RD} = F(\vec{RD})$

$$
\begin{array}{llll}
RD_{\text{in}}(1) = \{(v,?) \mid v \in Var\} & (e_1) & RD_{\text{out}}(1) = RD_{\text{in}}(1) \setminus \{(y,l) \mid l \in Lab^?\} \cup \{(y,1)\} & (s_1) \\
RD_{\text{in}}(2) = RD_{\text{out}}(1) & (e_2) & RD_{\text{out}}(2) = RD_{\text{in}}(2) \setminus \{(z,l) \mid l \in Lab^?\} \cup \{(z,2)\} & (s_2) \\
RD_{\text{in}}(3) = RD_{\text{out}}(2) \cup RD_{\text{out}}(5) & (e_3) & RD_{\text{out}}(3) = RD_{\text{in}}(3) & (s_3) \\
RD_{\text{in}}(4) = RD_{\text{out}}(3) & (e_4) & RD_{\text{out}}(4) = RD_{\text{in}}(4) \setminus \{(z,l) \mid l \in Lab^?\} \cup \{(z,4)\} & (s_4) \\
RD_{\text{in}}(5) = RD_{\text{out}}(4) & (e_5) & RD_{\text{out}}(5) = RD_{\text{in}}(5) \setminus \{(y,l) \mid l \in Lab^?\} \cup \{(y,5)\} & (s_5) \\
RD_{\text{in}}(6) = RD_{\text{out}}(3) & (e_6) & RD_{\text{out}}(6) = RD_{\text{in}}(6) \setminus \{(y,l) \mid l \in Lab^?\} \cup \{(y,6)\} & (s_6)
\end{array}
$$

Iteration 4: $F^4(\vec{\emptyset})$

$$
\begin{array}{llll}
RD_{\text{in}}(1) = & \{(x,?),(y,?),(z,?)\} & RD_{\text{out}}(1) = & \{(x,?),(y,1),(z,?)\} \\
RD_{\text{in}}(2) = & \{(x,?),(y,1),(z,?)\} & RD_{\text{out}}(2) = & \{(x,?),(y,1),(z,2)\} \\
RD_{\text{in}}(3) = & \{(y,1),(y,5),(z,2),(z,4)\} & RD_{\text{out}}(3) = & \{(y,5),(z,2)\} \\
RD_{\text{in}}(4) = & \{(y,5),(z,2)\} & RD_{\text{out}}(4) = & \{(z,4)\} \\
RD_{\text{in}}(5) = & \{(z,4)\} & RD_{\text{out}}(5) = & \{(y,5),(z,4)\} \\
RD_{\text{in}}(6) = & \{(y,5),(z,2)\} & RD_{\text{out}}(6) = & \{(y,6)\}
\end{array}
$$

# Reachable definition analysis : iterative computation

The solution can be computed by iteration. $RD_{\text{in}}(l)$ and $RD_{\text{out}}(l)$ are initialised with $\emptyset$ and their values are recomputed until stabilisation.

Equations : $\vec{RD} = F(\vec{RD})$

$$
\begin{array}{llll}
RD_{in}(1) = \{(v, ?) \mid v \in Var\} & (e_1) & RD_{out}(1) = RD_{in}(1) \setminus \{(y, l) \mid l \in Lab^?\} \cup \{(y, 1)\} & (s_1) \\
RD_{in}(2) = RD_{out}(1) & (e_2) & RD_{out}(2) = RD_{in}(2) \setminus \{(z, l) \mid l \in Lab^?\} \cup \{(z, 2)\} & (s_2) \\
RD_{in}(3) = RD_{out}(2) \cup RD_{out}(5) & (e_3) & RD_{out}(3) = RD_{in}(3) & (s_3) \\
RD_{in}(4) = RD_{out}(3) & (e_4) & RD_{out}(4) = RD_{in}(4) \setminus \{(z, l) \mid l \in Lab^?\} \cup \{(z, 4)\} & (s_4) \\
RD_{in}(5) = RD_{out}(4) & (e_5) & RD_{out}(5) = RD_{in}(5) \setminus \{(y, l) \mid l \in Lab^?\} \cup \{(y, 5)\} & (s_5) \\
RD_{in}(6) = RD_{out}(3) & (e_6) & RD_{out}(6) = RD_{in}(6) \setminus \{(y, l) \mid l \in Lab^?\} \cup \{(y, 6)\} & (s_6)
\end{array}
$$

Iteration 5: $F^5(\vec{\emptyset})$

$$
\begin{array}{llll}
RD_{in}(1) = & \{(x, ?), (y, ?), (z, ?)\} & RD_{out}(1) = & \{(x, ?), (y, 1), (z, ?)\} \\
RD_{in}(2) = & \{(x, ?), (y, 1), (z, ?)\} & RD_{out}(2) = & \{(x, ?), (y, 1), (z, 2)\} \\
RD_{in}(3) = & \{(x, ?), (y, 1), (y, 5), (z, 2), (z, 4)\} & RD_{out}(3) = & \{(y, 1), (y, 5), (z, 2), (z, 4)\} \\
RD_{in}(4) = & \{(y, 5), (z, 2)\} & RD_{out}(4) = & \{(y, 5), (z, 4)\} \\
RD_{in}(5) = & \{(z, 4)\} & RD_{out}(5) = & \{(y, 5), (z, 4)\} \\
RD_{in}(6) = & \{(y, 5), (z, 2)\} & RD_{out}(6) = & \{(y, 6), (z, 2)\}
\end{array}
$$

# Reachable definition analysis : iterative computation

The solution can be computed by iteration. $RD_{\text{in}}(l)$ and $RD_{\text{out}}(l)$ are initialised with $\emptyset$ and their values are recomputed until stabilisation.

Equations : $\vec{RD} = F(\vec{RD})$

$$
\begin{array}{llll}
RD_{\text{in}}(1) = \{(v,?) \mid v \in Var\} & (e_1) & RD_{\text{out}}(1) = RD_{\text{in}}(1) \setminus \{(y,l) \mid l \in Lab^?\} \cup \{(y,1)\} & (s_1) \\
RD_{\text{in}}(2) = RD_{\text{out}}(1) & (e_2) & RD_{\text{out}}(2) = RD_{\text{in}}(2) \setminus \{(z,l) \mid l \in Lab^?\} \cup \{(z,2)\} & (s_2) \\
RD_{\text{in}}(3) = RD_{\text{out}}(2) \cup RD_{\text{out}}(5) & (e_3) & RD_{\text{out}}(3) = RD_{\text{in}}(3) & (s_3) \\
RD_{\text{in}}(4) = RD_{\text{out}}(3) & (e_4) & RD_{\text{out}}(4) = RD_{\text{in}}(4) \setminus \{(z,l) \mid l \in Lab^?\} \cup \{(z,4)\} & (s_4) \\
RD_{\text{in}}(5) = RD_{\text{out}}(4) & (e_5) & RD_{\text{out}}(5) = RD_{\text{in}}(5) \setminus \{(y,l) \mid l \in Lab^?\} \cup \{(y,5)\} & (s_5) \\
RD_{\text{in}}(6) = RD_{\text{out}}(3) & (e_6) & RD_{\text{out}}(6) = RD_{\text{in}}(6) \setminus \{(y,l) \mid l \in Lab^?\} \cup \{(y,6)\} & (s_6)
\end{array}
$$

Iteration 6: $F^6(\vec{\emptyset})$

$$
\begin{array}{ll}
RD_{\text{in}}(1) = \{(x,?),(y,?),(z,?)\} & RD_{\text{out}}(1) = \{(x,?),(y,1),(z,?)\} \\
RD_{\text{in}}(2) = \{(x,?),(y,1),(z,?)\} & RD_{\text{out}}(2) = \{(x,?),(y,1),(z,2)\} \\
RD_{\text{in}}(3) = \{(x,?),(y,1),(y,5),(z,2),(z,4)\} & RD_{\text{out}}(3) = \{(x,?),(y,1),(y,5),(z,2),(z,4)\} \\
RD_{\text{in}}(4) = \{(y,1),(y,5),(z,2),(z,4)\} & RD_{\text{out}}(4) = \{(y,5),(z,4)\} \\
RD_{\text{in}}(5) = \{(y,5),(z,4)\} & RD_{\text{out}}(5) = \{(y,5),(z,4)\} \\
RD_{\text{in}}(6) = \{(y,1),(y,5),(z,2),(z,4)\} & RD_{\text{out}}(6) = \{(y,6),(z,2)\}
\end{array}
$$

# Reachable definition analysis : iterative computation

The solution can be computed by iteration. $RD_{in}(l)$ and $RD_{out}(l)$ are initialised with $\emptyset$ and their values are recomputed until stabilisation.

Equations : $\vec{RD} = F(\vec{RD})$

$$
\begin{array}{lll}
RD_{in}(1) = \{(v,?) \mid v \in Var\} & (e_1) & RD_{out}(1) = RD_{in}(1) \setminus \{(y,l) \mid l \in Lab^?\} \cup \{(y,1)\} \; (s_1) \\
RD_{in}(2) = RD_{out}(1) & (e_2) & RD_{out}(2) = RD_{in}(2) \setminus \{(z,l) \mid l \in Lab^?\} \cup \{(z,2)\} \; (s_2) \\
RD_{in}(3) = RD_{out}(2) \cup RD_{out}(5) & (e_3) & RD_{out}(3) = RD_{in}(3) \; (s_3) \\
RD_{in}(4) = RD_{out}(3) & (e_4) & RD_{out}(4) = RD_{in}(4) \setminus \{(z,l) \mid l \in Lab^?\} \cup \{(z,4)\} \; (s_4) \\
RD_{in}(5) = RD_{out}(4) & (e_5) & RD_{out}(5) = RD_{in}(5) \setminus \{(y,l) \mid l \in Lab^?\} \cup \{(y,5)\} \; (s_5) \\
RD_{in}(6) = RD_{out}(3) & (e_6) & RD_{out}(6) = RD_{in}(6) \setminus \{(y,l) \mid l \in Lab^?\} \cup \{(y,6)\} \; (s_6)
\end{array}
$$

Iteration 7: $F^7(\vec{\emptyset})$

$$
\begin{array}{ll}
RD_{in}(1) = \{(x,?),(y,?),(z,?)\} & RD_{out}(1) = \{(x,?),(y,1),(z,?)\} \\
RD_{in}(2) = \{(x,?),(y,1),(z,?)\} & RD_{out}(2) = \{(x,?),(y,1),(z,2)\} \\
RD_{in}(3) = \{(x,?),(y,1),(y,5),(z,2),(z,4)\} & RD_{out}(3) = \{(x,?),(y,1),(y,5),(z,2),(z,4)\} \\
RD_{in}(4) = \{(x,?),(y,1),(y,5),(z,2),(z,4)\} & RD_{out}(4) = \{(y,1),(y,5),(z,4)\} \\
RD_{in}(5) = \{(y,5),(z,4)\} & RD_{out}(5) = \{(y,5),(z,4)\} \\
RD_{in}(6) = \{(x,?),(y,1),(y,5),(z,2),(z,4)\} & RD_{out}(6) = \{(y,6),(z,2),(z,4)\}
\end{array}
$$

# Reachable definition analysis : iterative computation

The solution can be computed by iteration. $RD_{\text{in}}(l)$ and $RD_{\text{out}}(l)$ are initialised with $\emptyset$ and their values are recomputed until stabilisation.

Equations : $\vec{RD} = F(\vec{RD})$

$$
\begin{array}{llll}
RD_{\text{in}}(1) = \{(v,?) \mid v \in Var\} & (e_1) & RD_{\text{out}}(1) = RD_{\text{in}}(1) \setminus \{(y,l) \mid l \in Lab^?\} \cup \{(y,1)\} & (s_1) \\
RD_{\text{in}}(2) = RD_{\text{out}}(1) & (e_2) & RD_{\text{out}}(2) = RD_{\text{in}}(2) \setminus \{(z,l) \mid l \in Lab^?\} \cup \{(z,2)\} & (s_2) \\
RD_{\text{in}}(3) = RD_{\text{out}}(2) \cup RD_{\text{out}}(5) & (e_3) & RD_{\text{out}}(3) = RD_{\text{in}}(3) & (s_3) \\
RD_{\text{in}}(4) = RD_{\text{out}}(3) & (e_4) & RD_{\text{out}}(4) = RD_{\text{in}}(4) \setminus \{(z,l) \mid l \in Lab^?\} \cup \{(z,4)\} & (s_4) \\
RD_{\text{in}}(5) = RD_{\text{out}}(4) & (e_5) & RD_{\text{out}}(5) = RD_{\text{in}}(5) \setminus \{(y,l) \mid l \in Lab^?\} \cup \{(y,5)\} & (s_5) \\
RD_{\text{in}}(6) = RD_{\text{out}}(3) & (e_6) & RD_{\text{out}}(6) = RD_{\text{in}}(6) \setminus \{(y,l) \mid l \in Lab^?\} \cup \{(y,6)\} & (s_6)
\end{array}
$$

Iteration 8: $F^8(\vec{\emptyset})$

$$
\begin{array}{ll}
RD_{\text{in}}(1) = \{(x,?),(y,?),(z,?)\} & RD_{\text{out}}(1) = \{(x,?),(y,1),(z,?)\} \\
RD_{\text{in}}(2) = \{(x,?),(y,1),(z,?)\} & RD_{\text{out}}(2) = \{(x,?),(y,1),(z,2)\} \\
RD_{\text{in}}(3) = \{(x,?),(y,1),(y,5),(z,2),(z,4)\} & RD_{\text{out}}(3) = \{(x,?),(y,1),(y,5),(z,2),(z,4)\} \\
RD_{\text{in}}(4) = \{(x,?),(y,1),(y,5),(z,2),(z,4)\} & RD_{\text{out}}(4) = \{(x,?),(y,1),(y,5),(z,4)\} \\
RD_{\text{in}}(5) = \{(y,1),(y,5),(z,4)\} & RD_{\text{out}}(5) = \{(y,5),(z,4)\} \\
RD_{\text{in}}(6) = \{(x,?),(y,1),(y,5),(z,2),(z,4)\} & RD_{\text{out}}(6) = \{(x,?),(y,6),(z,2),(z,4)\}
\end{array}
$$

# Reachable definition analysis : iterative computation

The solution can be computed by iteration. $RD_{\text{in}}(l)$ and $RD_{\text{out}}(l)$ are initialised with $\emptyset$ and their values are recomputed until stabilisation.

Equations : $\vec{RD} = F(\vec{RD})$

$$
\begin{array}{lll}
RD_{\text{in}}(1) = \{(v,?) \mid v \in Var\} & (e_1) & RD_{\text{out}}(1) = RD_{\text{in}}(1) \setminus \{(y,l) \mid l \in Lab^?\} \cup \{(y,1)\} \ (s_1) \\
RD_{\text{in}}(2) = RD_{\text{out}}(1) & (e_2) & RD_{\text{out}}(2) = RD_{\text{in}}(2) \setminus \{(z,l) \mid l \in Lab^?\} \cup \{(z,2)\} \ (s_2) \\
RD_{\text{in}}(3) = RD_{\text{out}}(2) \cup RD_{\text{out}}(5) & (e_3) & RD_{\text{out}}(3) = RD_{\text{in}}(3) \ (s_3) \\
RD_{\text{in}}(4) = RD_{\text{out}}(3) & (e_4) & RD_{\text{out}}(4) = RD_{\text{in}}(4) \setminus \{(z,l) \mid l \in Lab^?\} \cup \{(z,4)\} \ (s_4) \\
RD_{\text{in}}(5) = RD_{\text{out}}(4) & (e_5) & RD_{\text{out}}(5) = RD_{\text{in}}(5) \setminus \{(y,l) \mid l \in Lab^?\} \cup \{(y,5)\} \ (s_5) \\
RD_{\text{in}}(6) = RD_{\text{out}}(3) & (e_6) & RD_{\text{out}}(6) = RD_{\text{in}}(6) \setminus \{(y,l) \mid l \in Lab^?\} \cup \{(y,6)\} \ (s_6)
\end{array}
$$

Iteration

$$
\begin{array}{ll}
RD_{\text{in}}(1) = \{(x,?),(y,?),(z,?)\} & RD_{\text{out}}(1) = \{(x,?),(y,1),(z,?)\} \\
RD_{\text{in}}(2) = \{(x,?),(y,1),(z,?)\} & RD_{\text{out}}(2) = \{(x,?),(y,1),(z,2)\} \\
RD_{\text{in}}(3) = \{(x,?),(y,1),(y,5),(z,2),(z,4)\} & RD_{\text{out}}(3) = \{(x,?),(y,1),(y,5),(z,2),(z,4)\} \\
RD_{\text{in}}(4) = \{(x,?),(y,1),(y,5),(z,2),(z,4)\} & RD_{\text{out}}(4) = \{(x,?),(y,1),(y,5),(z,4)\} \\
RD_{\text{in}}(5) = \{(x,?),(y,1),(y,5),(z,4)\} & RD_{\text{out}}(5) = \{(y,5),(z,4)\} \\
RD_{\text{in}}(6) = \{(x,?),(y,1),(y,5),(z,2),(z,4)\} & RD_{\text{out}}(6) = \{(x,?),(y,6),(z,2),(z,4)\}
\end{array}
$$

# Reachable definition analysis : iterative computation

The solution can be computed by iteration. $RD_{\text{in}}(l)$ and $RD_{\text{out}}(l)$ are initialised with $\emptyset$ and their values are recomputed until stabilisation.

Equations : $\vec{RD} = F(\vec{RD})$

$$
\begin{array}{lllr}
RD_{\text{in}}(1) = \{(v,?) \mid v \in Var\} & (e_1) & RD_{\text{out}}(1) = RD_{\text{in}}(1) \setminus \{(y,l) \mid l \in Lab^?\} \cup \{(y,1)\} & (s_1) \\
RD_{\text{in}}(2) = RD_{\text{out}}(1) & (e_2) & RD_{\text{out}}(2) = RD_{\text{in}}(2) \setminus \{(z,l) \mid l \in Lab^?\} \cup \{(z,2)\} & (s_2) \\
RD_{\text{in}}(3) = RD_{\text{out}}(2) \cup RD_{\text{out}}(5) & (e_3) & RD_{\text{out}}(3) = RD_{\text{in}}(3) & (s_3) \\
RD_{\text{in}}(4) = RD_{\text{out}}(3) & (e_4) & RD_{\text{out}}(4) = RD_{\text{in}}(4) \setminus \{(z,l) \mid l \in Lab^?\} \cup \{(z,4)\} & (s_4) \\
RD_{\text{in}}(5) = RD_{\text{out}}(4) & (e_5) & RD_{\text{out}}(5) = RD_{\text{in}}(5) \setminus \{(y,l) \mid l \in Lab^?\} \cup \{(y,5)\} & (s_5) \\
RD_{\text{in}}(6) = RD_{\text{out}}(3) & (e_6) & RD_{\text{out}}(6) = RD_{\text{in}}(6) \setminus \{(y,l) \mid l \in Lab^?\} \cup \{(y,6)\} & (s_6)
\end{array}
$$

Iteration

$$
\begin{array}{ll}
RD_{\text{in}}(1) = \{(x,?),(y,?),(z,?)\} & RD_{\text{out}}(1) = \{(x,?),(y,1),(z,?)\} \\
RD_{\text{in}}(2) = \{(x,?),(y,1),(z,?)\} & RD_{\text{out}}(2) = \{(x,?),(y,1),(z,2)\} \\
RD_{\text{in}}(3) = \{(x,?),(y,1),(y,5),(z,2),(z,4)\} & RD_{\text{out}}(3) = \{(x,?),(y,1),(y,5),(z,2),(z,4)\} \\
RD_{\text{in}}(4) = \{(x,?),(y,1),(y,5),(z,2),(z,4)\} & RD_{\text{out}}(4) = \{(x,?),(y,1),(y,5),(z,4)\} \\
RD_{\text{in}}(5) = \{(x,?),(y,1),(y,5),(z,4)\} & RD_{\text{out}}(5) = \{(x,?),(y,5),(z,4)\} \\
RD_{\text{in}}(6) = \{(x,?),(y,1),(y,5),(z,2),(z,4)\} & RD_{\text{out}}(6) = \{(x,?),(y,6),(z,2),(z,4)\}
\end{array}
$$

# Reachable definition analysis : iterative computation

The solution can be computed by iteration. $RD_{\text{in}}(l)$ and $RD_{\text{out}}(l)$ are initialised with $\emptyset$ and their values are recomputed until stabilisation.

Equations : $\vec{RD} = F(\vec{RD})$

| | | | |
|---|---|---|---|
| $RD_{\text{in}}(1) = \{(v,?) \mid v \in Var\}$ | $(e_1)$ | $RD_{\text{out}}(1) = RD_{\text{in}}(1) \setminus \{(y,l) \mid l \in Lab^?\} \cup \{(y,1)\}$ | $(s_1)$ |
| $RD_{\text{in}}(2) = RD_{\text{out}}(1)$ | $(e_2)$ | $RD_{\text{out}}(2) = RD_{\text{in}}(2) \setminus \{(z,l) \mid l \in Lab^?\} \cup \{(z,2)\}$ | $(s_2)$ |
| $RD_{\text{in}}(3) = RD_{\text{out}}(2) \cup RD_{\text{out}}(5)$ | $(e_3)$ | $RD_{\text{out}}(3) = RD_{\text{in}}(3)$ | $(s_3)$ |
| $RD_{\text{in}}(4) = RD_{\text{out}}(3)$ | $(e_4)$ | $RD_{\text{out}}(4) = RD_{\text{in}}(4) \setminus \{(z,l) \mid l \in Lab^?\} \cup \{(z,4)\}$ | $(s_4)$ |
| $RD_{\text{in}}(5) = RD_{\text{out}}(4)$ | $(e_5)$ | $RD_{\text{out}}(5) = RD_{\text{in}}(5) \setminus \{(y,l) \mid l \in Lab^?\} \cup \{(y,5)\}$ | $(s_5)$ |
| $RD_{\text{in}}(6) = RD_{\text{out}}(3)$ | $(e_6)$ | $RD_{\text{out}}(6) = RD_{\text{in}}(6) \setminus \{(y,l) \mid l \in Lab^?\} \cup \{(y,6)\}$ | $(s_6)$ |

Iteration

| | |
|---|---|
| $RD_{\text{in}}(1) = \{(x,?),(y,?),(z,?)\}$ | $RD_{\text{out}}(1) = \{(x,?),(y,1),(z,?)\}$ |
| $RD_{\text{in}}(2) = \{(x,?),(y,1),(z,?)\}$ | $RD_{\text{out}}(2) = \{(x,?),(y,1),(z,2)\}$ |
| $RD_{\text{in}}(3) = \{(x,?),(y,1),(y,5),(z,2),(z,4)\}$ | $RD_{\text{out}}(3) = \{(x,?),(y,1),(y,5),(z,2),(z,4)\}$ |
| $RD_{\text{in}}(4) = \{(x,?),(y,1),(y,5),(z,2),(z,4)\}$ | $RD_{\text{out}}(4) = \{(x,?),(y,1),(y,5),(z,4)\}$ |
| $RD_{\text{in}}(5) = \{(x,?),(y,1),(y,5),(z,4)\}$ | $RD_{\text{out}}(5) = \{(x,?),(y,5),(z,4)\}$ |
| $RD_{\text{in}}(6) = \{(x,?),(y,1),(y,5),(z,2),(z,4)\}$ | $RD_{\text{out}}(6) = \{(x,?),(y,6),(z,2),(z,4)\}$ |

# Reachable definition analysis : several solutions ?

The equation system admits several solutions.

Equations :

$$
\begin{array}{ll}
RD_{in}(1) = \{(v,?) \mid v \in Var\} & RD_{out}(1) = RD_{in}(1) \setminus \{(\mathbf{y},l) \mid l \in Lab^?\} \cup \{(\mathbf{y},1)\} \\
RD_{in}(2) = RD_{out}(1) & RD_{out}(2) = RD_{in}(2) \setminus \{(\mathbf{z},l) \mid l \in Lab^?\} \cup \{(\mathbf{z},2)\} \\
RD_{in}(3) = RD_{out}(2) \cup RD_{out}(5) & RD_{out}(3) = RD_{in}(3) \\
RD_{in}(4) = RD_{out}(3) & RD_{out}(4) = RD_{in}(4) \setminus \{(\mathbf{z},l) \mid l \in Lab^?\} \cup \{(\mathbf{z},4)\} \\
RD_{in}(5) = RD_{out}(4) & RD_{out}(5) = RD_{in}(5) \setminus \{(\mathbf{y},l) \mid l \in Lab^?\} \cup \{(\mathbf{y},5)\} \\
RD_{in}(6) = RD_{out}(3) & RD_{out}(6) = RD_{in}(6) \setminus \{(\mathbf{y},l) \mid l \in Lab^?\} \cup \{(\mathbf{y},6)\}
\end{array}
$$

Previous solution:

$$
\begin{array}{llll}
RD_{in}(1) = & \{(x,?),(y,?),(z,?)\} & RD_{out}(1) = & \{(x,?),(y,1),(z,?)\} \\
RD_{in}(2) = & \{(x,?),(y,1),(z,?)\} & RD_{out}(2) = & \{(x,?),(y,1),(z,2)\} \\
RD_{in}(3) = & \{(x,?),(y,1),(y,5),(z,2),(z,4)\} & RD_{out}(3) = & \{(x,?),(y,1),(y,5),(z,2),(z,4)\} \\
RD_{in}(4) = & \{(x,?),(y,1),(y,5),(z,2),(z,4)\} & RD_{out}(4) = & \{(x,?),(y,1),(y,5),(z,4)\} \\
RD_{in}(5) = & \{(x,?),(y,1),(y,5),(z,4)\} & RD_{out}(5) = & \{(x,?),(y,5),(z,4)\} \\
RD_{in}(6) = & \{(x,?),(y,1),(y,5),(z,2),(z,4)\} & RD_{out}(6) = & \{(x,?),(y,6),(z,2),(z,4)\}
\end{array}
$$

# Reachable definition analysis : several solutions ?

The equation system admits several solutions.

Equations :

$$
\begin{array}{ll}
RD_{\text{in}}(1) = \{(v,?) \mid v \in Var\} & RD_{\text{out}}(1) = RD_{\text{in}}(1) \setminus \{(\mathbf{y},l) \mid l \in Lab^?\} \cup \{(\mathbf{y},1)\} \\
RD_{\text{in}}(2) = RD_{\text{out}}(1) & RD_{\text{out}}(2) = RD_{\text{in}}(2) \setminus \{(\mathbf{z},l) \mid l \in Lab^?\} \cup \{(\mathbf{z},2)\} \\
RD_{\text{in}}(3) = RD_{\text{out}}(2) \cup RD_{\text{out}}(5) & RD_{\text{out}}(3) = RD_{\text{in}}(3) \\
RD_{\text{in}}(4) = RD_{\text{out}}(3) & RD_{\text{out}}(4) = RD_{\text{in}}(4) \setminus \{(\mathbf{z},l) \mid l \in Lab^?\} \cup \{(\mathbf{z},4)\} \\
RD_{\text{in}}(5) = RD_{\text{out}}(4) & RD_{\text{out}}(5) = RD_{\text{in}}(5) \setminus \{(\mathbf{y},l) \mid l \in Lab^?\} \cup \{(\mathbf{y},5)\} \\
RD_{\text{in}}(6) = RD_{\text{out}}(3) & RD_{\text{out}}(6) = RD_{\text{in}}(6) \setminus \{(\mathbf{y},l) \mid l \in Lab^?\} \cup \{(\mathbf{y},6)\}
\end{array}
$$

Another solution:

$$
\begin{array}{ll}
RD'_{\text{in}}(1) = \{(x,?),(y,?),(z,?)\} & RD'_{\text{out}}(1) = \{(x,?),(y,1),(z,?)\} \\
RD'_{\text{in}}(2) = \{(x,?),(y,1),(z,?)\} & RD'_{\text{out}}(2) = \{(x,?),(y,1),(z,2)\} \\
RD'_{\text{in}}(3) = \{(x,?),(x,1),(y,1),(y,5),(z,2),(z,4)\} & RD'_{\text{out}}(3) = \{(x,?),(x,1),(y,1),(y,5),(z,2),(z,4)\} \\
RD'_{\text{in}}(4) = \{(x,?),(x,1),(y,1),(y,5),(z,2),(z,4)\} & RD'_{\text{out}}(4) = \{(x,?),(x,1),(y,1),(y,5),(z,4)\} \\
RD'_{\text{in}}(5) = \{(x,?),(x,1),(y,1),(y,5),(z,4)\} & RD'_{\text{out}}(5) = \{(x,?),(x,1),(y,5),(z,4)\} \\
RD'_{\text{in}}(6) = \{(x,?),(x,1),(y,1),(y,5),(z,2),(z,4)\} & RD'_{\text{out}}(6) = \{(x,?),(x,1),(y,6),(z,2),(z,4)\}
\end{array}
$$

# Choosing the best solution

Remark :

$$RD_{\text{in}}(1) \subseteq RD'_{\text{in}}(1),\;\; RD_{\text{out}}(1) \subseteq RD'_{\text{out}}(1),\;\; \ldots,\;\; RD_{\text{out}}(6) \subseteq RD'_{\text{out}}(6)$$

$RD$ gives an information more precise than $RD'$ :

- $RD_{\text{in}}(3) = \{(x,?),(y,1),(y,5),(z,2),(z,4)\}$

  *"the value of* **x** *at point 3 is not initialised"*
- $RD'_{\text{in}}(3) = \{(x,?),(x,1),(y,1),(y,5),(z,2),(z,4)\}$

  *"the value of* **x** *at point 3 is not initialised, or has been defined at point 1"*

Between two comparables solutions (e.g. $\vec{RD} \subseteq \vec{RD}'$), we prefer the smallest.

Theoretical result: there always exists a smallest solution

# Equation resolution

The previous analysis is a solution of an equation system of the form

$$\begin{cases} x_1 & = & f_1(x_1, \ldots, x_n) \\ & \vdots & \\ x_n & = & f_n(x_1, \ldots, x_n) \end{cases} \qquad \text{or} \qquad \vec{x} = \vec{f}(\vec{x})$$

called fixpoint equations.

It is a common mathematical problem that raises two questions:

1. Existence and uniqueness (in what sense ?) of the solution ?
2. Effective computation method ?

A few observations about the previous analysis:

- The $x_i$ are sets, that can be ordered by set inclusion $\subseteq$
- The functions $f_i$ are monotone (*croissantes*) for the partial order $\subseteq$

# Poset

## Definition

A *partially ordered set (poset)* is a couple $(A, \sqsubseteq)$ with $A$ a set, and $\sqsubseteq$ a partial order relation, *i.e.*:

$$\begin{array}{ll} \forall x \in A,\ x \sqsubseteq x & \text{(reflexivity)} \\ \forall x,y \in A,\ x \sqsubseteq y \ \wedge\ y \sqsubseteq x \ \Rightarrow\ x = y & \text{(antisymmetry)} \\ \forall x,y,z \in A,\ x \sqsubseteq y \ \wedge\ y \sqsubseteq z \ \Rightarrow\ x \sqsubseteq z & \text{(transitivity)} \end{array}$$

## Examples

- $(\mathbb{N}, \leqslant)$ ( total : $\forall x, y,\ x \sqsubseteq y \ \vee\ y \sqsubseteq x$)
- $(\mathbb{N}$, *"is a divisor of"*$)$ written $(\mathbb{N}, |)$
- $(\mathcal{P}(X), \subseteq)$ with $X$ any set
- $(A^*$, *"to be a prefix of"*$)$ with $A$ an alphabet

# Exercice

Show that $(\mathbb{N}, \textit{"is a divisor of"})$ is a poset.

# Hasse diagram

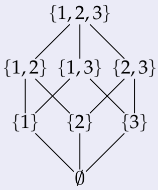
## Graphical representation of a poset

### Definition

A 2D-drawing (set of points and segments) is *a Hasse diagram* of a poset $(A, \sqsubseteq)$ iff

- each element of $A$ is associated with a point
- if $x \sqsubseteq y$ with $x \neq y$ and $\neg\exists z, x \sqsubseteq z \sqsubset y$ then
  - a segment connects the points $p_x$ and $p_y$ that are associated respectively with $x$ and $y$
  - the ordinate (vertical scale) of $p_x$ is lower than the ordinate of $p_y$

### Example




is an Hasse diagram of the poset $(\mathcal{P}(\{1,2,3\}), \subseteq)$

# Exercice

Give a Hasse diagram of the poset $(\{1, 2, 3, 4, 6, 8, 12\}, |)$

# Lattice

## Definition

A *lattice* is a 4-tuple $(A, \sqsubseteq, \sqcup, \sqcap)$ with

- $(A, \sqsubseteq)$ a poset,
- $\sqcup$ a binary least upper bound:

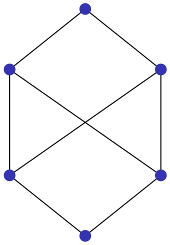
$$\forall x, y \in A,\ x \sqsubseteq x \sqcup y \ \wedge \ y \sqsubseteq x \sqcup y$$
$$\forall x, y, z \in A,\ x \sqsubseteq z \ \wedge \ y \sqsubseteq z \ \Rightarrow \ x \sqcup y \sqsubseteq z$$

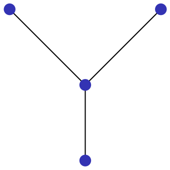
- $\sqcap$ a binary greatest lower bound:

$$\forall x, y \in A,\ x \sqcap y \sqsubseteq x \ \wedge \ x \sqcap y \sqsubseteq y$$
$$\forall x, y, z \in A,\ z \sqsubseteq x \ \wedge \ z \sqsubseteq y \ \Rightarrow \ z \sqsubseteq x \sqcap y$$

# Exercice

Between the following diagrams, which represent lattices ?





# Exercice

Give the lattice structure of $(\mathbb{N}, \leqslant)$ and $(\mathbb{N}, |)$.

# Complete lattice

## Definition

A complete lattice is a triple $(A, \sqsubseteq, \bigsqcup)$ with

- $(A, \sqsubseteq)$ a poset,
- $\bigsqcup$ a least upper bound : for all subsets $S$ of $A$,
  - $\forall a \in S,\ a \sqsubseteq \bigsqcup S$
  - $\forall b \in A, (\forall a \in S,\ a \sqsubseteq b) \Rightarrow \bigsqcup S \sqsubseteq b$

A complete lattice necessarily possesses *a greatest lower bound* $\bigsqcap$ operator, *i.e. :* for all subsets $S$ of $A$,

- $\forall a \in S,\ \bigsqcap S \sqsubseteq a$
- $\forall b \in A, (\forall a \in S,\ b \sqsubseteq a) \Rightarrow b \sqsubseteq \bigsqcap S$

Just consider

$$\bigsqcap S = \bigsqcup \{ y \mid \forall x \in S, y \sqsubseteq x \}.$$

# Examples

1. For all set $X$, $(\mathcal{P}(X), \subseteq, \bigcup)$ is a complete lattice for which $\bigcap$ is a greatest lower bound.
2. Every finite lattice is complete.

# Exercice

Show that any complete lattice admits

- *a greatest element* $\top$ $(\forall x, x \sqsubseteq \top)$
- *a least element* $\bot$ $(\forall x, \bot \sqsubseteq x)$

# Fixpoints, post-fixpoints and pre-fixpoints

## Definition

Consider $f \in A \to A$ with $(A, \sqsubseteq)$ a poset, an element $x \in A$

- is a *fixpoint of f* iff $f(x) = x$
- is a *greatest fixpoint of f* iff $f(x) = x$ and $\forall y,\ f(y) = y \ \Rightarrow y \sqsubseteq x$
- is a *least fixpoint of f* iff $f(x) = x$ and $\forall y,\ f(y) = y \ \Rightarrow x \sqsubseteq y$
- is a *post-fixpoint of f* iff $f(x) \sqsubseteq x$
- is a *pre-fixpoint of f* iff $x \sqsubseteq f(x)$

## Definition

Let $f \in A \to A$, $f$ is *monotone* iff

$$\forall x, y \in,\ x \sqsubseteq y \ \Rightarrow \ f(x) \sqsubseteq f(y)$$

# Fixpoints, post-fixpoints and pre-fixpoints

## Theorem (Knaster-Tarski)

*In a complete lattice* $(A, \sqsubseteq, \bigsqcup)$*, for all monotone functions* $f \in A \to A$*,*

- *the least fixpoint* $\mathrm{lfp}(f)$ *of* $f$ *exists and is* $\bigsqcap\{x \in A \mid f(x) \sqsubseteq x\}$*,*
- *the greatest fixpoint* $\mathrm{gfp}(f)$ *of* $f$ *exists and is* $\bigsqcup\{x \in A \mid x \sqsubseteq f(x)\}$*,*

# Proof of the Knaster-Tarski theorem

- Let us define $a = \bigsqcap P$ with $P = PostFix(f) = \{x \mid f(x) \sqsubseteq x\}$.

# Proof of the Knaster-Tarski theorem

- Let us define $a = \bigsqcap P$ with $P = \mathit{PostFix}(f) = \{x \mid f(x) \sqsubseteq x\}$.
- For all $x \in P$, we have
  –

# Proof of the Knaster-Tarski theorem

- Let us define $a = \bigsqcap P$ with $P = PostFix(f) = \{x \mid f(x) \sqsubseteq x\}$.
- For all $x \in P$, we have
  - $a \sqsubseteq x$ $\qquad$ $a$ greatest lower bound of $P$
  -

# Proof of the Knaster-Tarski theorem

- Let us define $a = \bigsqcap P$ with $P = \mathit{PostFix}(f) = \{x \mid f(x) \sqsubseteq x\}$.
- For all $x \in P$, we have
  - $a \sqsubseteq x$ — $a$ greatest lower bound of $P$
  - $f(a) \sqsubseteq f(x)$ — $f$ monotone
  -

# Proof of the Knaster-Tarski theorem

- Let us define $a = \bigsqcap P$ with $P = \mathit{PostFix}(f) = \{x \mid f(x) \sqsubseteq x\}$.
- For all $x \in P$, we have
  - $a \sqsubseteq x$ — $a$ greatest lower bound of $P$
  - $f(a) \sqsubseteq f(x)$ — $f$ monotone
  - $f(a) \sqsubseteq x$ — def. $P$ and transitivity

# Proof of the Knaster-Tarski theorem

- Let us define $a = \bigsqcap P$ with $P = PostFix(f) = \{x \mid f(x) \sqsubseteq x\}$.
- For all $x \in P$, we have
  - $a \sqsubseteq x$ — $a$ greatest lower bound of $P$
  - $f(a) \sqsubseteq f(x)$ — $f$ monotone
  - $f(a) \sqsubseteq x$ — def. $P$ and transitivity

  hence $f(a) \sqsubseteq x$ for all $x \in P$

# Proof of the Knaster-Tarski theorem

- Let us define $a = \bigsqcap P$ with $P = PostFix(f) = \{x \mid f(x) \sqsubseteq x\}$.
- For all $x \in P$, we have
  - $a \sqsubseteq x$ — $a$ greatest lower bound of $P$
  - $f(a) \sqsubseteq f(x)$ — $f$ monotone
  - $f(a) \sqsubseteq x$ — def. $P$ and transitivity

  hence $f(a) \sqsubseteq x$ for all $x \in P$
- $f(a) \sqsubseteq a$ — previous result and $a$ greatest lower bound of $P$

# Proof of the Knaster-Tarski theorem

- Let us define $a = \bigsqcap P$ with $P = \mathit{PostFix}(f) = \{x \mid f(x) \sqsubseteq x\}$.
- For all $x \in P$, we have
  - $a \sqsubseteq x$ — $a$ greatest lower bound of $P$
  - $f(a) \sqsubseteq f(x)$ — $f$ monotone
  - $f(a) \sqsubseteq x$ — def. $P$ and transitivity

  hence $f(a) \sqsubseteq x$ for all $x \in P$
- $f(a) \sqsubseteq a$ — previous result and $a$ greatest lower bound of $P$

  $\Rightarrow f(f(a)) \sqsubseteq f(a)$ — $f$ monotone

# Proof of the Knaster-Tarski theorem

- Let us define $a = \bigsqcap P$ with $P = \mathit{PostFix}(f) = \{x \mid f(x) \sqsubseteq x\}$.
- For all $x \in P$, we have
  - $a \sqsubseteq x$ — $a$ greatest lower bound of $P$
  - $f(a) \sqsubseteq f(x)$ — $f$ monotone
  - $f(a) \sqsubseteq x$ — def. $P$ and transitivity

  hence $f(a) \sqsubseteq x$ for all $x \in P$
- $f(a) \sqsubseteq a$ — previous result and $a$ greatest lower bound of $P$
  $\Rightarrow f(f(a)) \sqsubseteq f(a)$ — $f$ monotone
  $\Rightarrow f(a) \in P$ — def. $P$

# Proof of the Knaster-Tarski theorem

- Let us define $a = \bigsqcap P$ with $P = \mathit{PostFix}(f) = \{x \mid f(x) \sqsubseteq x\}$.
- For all $x \in P$, we have
  - $a \sqsubseteq x$ — $a$ greatest lower bound of $P$
  - $f(a) \sqsubseteq f(x)$ — $f$ monotone
  - $f(a) \sqsubseteq x$ — def. $P$ and transitivity

  hence $f(a) \sqsubseteq x$ for all $x \in P$
- $f(a) \sqsubseteq a$ — previous result and $a$ greatest lower bound of $P$
  $\Rightarrow f(f(a)) \sqsubseteq f(a)$ — $f$ monotone
  $\Rightarrow f(a) \in P$ — def. $P$
  $\Rightarrow a \sqsubseteq f(a)$ — $a$ lower bound of $P$

# Proof of the Knaster-Tarski theorem

- Let us define $a = \bigsqcap P$ with $P = \mathit{PostFix}(f) = \{x \mid f(x) \sqsubseteq x\}$.
- For all $x \in P$, we have
  - $a \sqsubseteq x$ — $a$ greatest lower bound of $P$
  - $f(a) \sqsubseteq f(x)$ — $f$ monotone
  - $f(a) \sqsubseteq x$ — def. $P$ and transitivity

  hence $f(a) \sqsubseteq x$ for all $x \in P$
- $f(a) \sqsubseteq a$ — previous result and $a$ greatest lower bound of $P$

  $\Rightarrow f(f(a)) \sqsubseteq f(a)$ — $f$ monotone

  $\Rightarrow f(a) \in P$ — def. $P$

  $\Rightarrow a \sqsubseteq f(a)$ — $a$ lower bound of $P$

  $\Rightarrow f(a) = a$ — antisymmetry

# Proof of the Knaster-Tarski theorem

- Let us define $a = \bigsqcap P$ with $P = PostFix(f) = \{x \mid f(x) \sqsubseteq x\}$.
- For all $x \in P$, we have
  - $a \sqsubseteq x$ — $a$ greatest lower bound of $P$
  - $f(a) \sqsubseteq f(x)$ — $f$ monotone
  - $f(a) \sqsubseteq x$ — def. $P$ and transitivity

  hence $f(a) \sqsubseteq x$ for all $x \in P$
- $f(a) \sqsubseteq a$ — previous result and $a$ greatest lower bound of $P$
  $\Rightarrow f(f(a)) \sqsubseteq f(a)$ — $f$ monotone
  $\Rightarrow f(a) \in P$ — def. $P$
  $\Rightarrow a \sqsubseteq f(a)$ — $a$ lower bound of $P$
  $\Rightarrow f(a) = a$ — antisymmetry

  hence $a \in Fix(f)$

# Proof of the Knaster-Tarski theorem

- Let us define $a = \bigsqcap P$ with $P = PostFix(f) = \{x \mid f(x) \sqsubseteq x\}$.
- For all $x \in P$, we have
  - $a \sqsubseteq x$ — $a$ greatest lower bound of $P$
  - $f(a) \sqsubseteq f(x)$ — $f$ monotone
  - $f(a) \sqsubseteq x$ — def. $P$ and transitivity

  hence $f(a) \sqsubseteq x$ for all $x \in P$
- $f(a) \sqsubseteq a$ — previous result and $a$ greatest lower bound of $P$
  $\Rightarrow f(f(a)) \sqsubseteq f(a)$ — $f$ monotone
  $\Rightarrow f(a) \in P$ — def. $P$
  $\Rightarrow a \sqsubseteq f(a)$ — $a$ lower bound of $P$
  $\Rightarrow f(a) = a$ — antisymmetry

  hence $a \in Fix(f)$
- If $x \in Fix(f)$ then $x \in P$, hence $a \sqsubseteq x$ because $a$ is a lower bound of $P$. Hence $a = \text{lfp}(f)$.

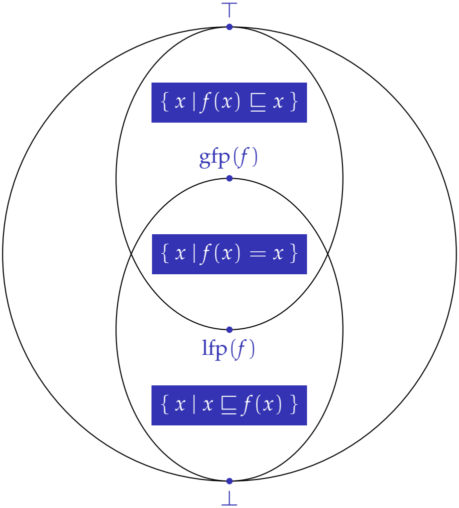
# Proof of the Knaster-Tarski theorem

- Let us define $a = \bigsqcap P$ with $P = \mathit{PostFix}(f) = \{x \mid f(x) \sqsubseteq x\}$.
- For all $x \in P$, we have
  - $a \sqsubseteq x$ — $a$ greatest lower bound of $P$
  - $f(a) \sqsubseteq f(x)$ — $f$ monotone
  - $f(a) \sqsubseteq x$ — def. $P$ and transitivity

  hence $f(a) \sqsubseteq x$ for all $x \in P$
- $f(a) \sqsubseteq a$ — previous result and $a$ greatest lower bound of $P$
  $\Rightarrow f(f(a)) \sqsubseteq f(a)$ — $f$ monotone
  $\Rightarrow f(a) \in P$ — def. $P$
  $\Rightarrow a \sqsubseteq f(a)$ — $a$ lower bound of $P$
  $\Rightarrow f(a) = a$ — antisymmetry

  hence $a \in \mathit{Fix}(f)$
- If $x \in \mathit{Fix}(f)$ then $x \in P$, hence $a \sqsubseteq x$ because $a$ is a lower bound of $P$. Hence $a = \mathrm{lfp}(f)$.
- Proof of $\mathrm{gfp}(f) = \bigsqcup \mathit{PreFix}(f)$ by duality.

# Fixpoints, post-fixpoints and pre-fixpoints




$$
\begin{aligned}
\top &= \bigsqcup \{\, x \mid f(x) \sqsubseteq x \,\} \\
\mathrm{gfp}(f) &= \bigsqcup \{\, x \mid x \sqsubseteq f(x) \,\} \\
\mathrm{lfp}(f) &= \bigsqcap \{\, x \mid f(x) \sqsubseteq x \,\} \\
\bot &= \bigsqcap \{\, x \mid x \sqsubseteq f(x) \,\}
\end{aligned}
$$

# Fixpoint computation

## Theorem

*Let $(A, \sqsubseteq)$ be a poset with a least element $\bot$. Let $f$ a monotone function. If the sequence $\bot, f(\bot), \ldots, f^n(\bot), \ldots$ stabilises from a given rank $k$ (i.e. $f^k(\bot) = f^{k+1}(\bot)$, then $f^k(\bot)$ is the least fixpoint of $f$.*

**Proof:** Since $\bot \sqsubseteq f(\bot)$ and $f$ is monotone, we can show by induction on $\mathbb{N}$ that $\bot, f(\bot), \ldots, f^n(\bot), \ldots$ is an increasing sequence.

Let $k$ such that $f^k(\bot) = f^{k+1}(\bot)$.

- Hence $f^k(\bot)$ is a fixpoint of $f$.
- If $x$ is a fixpoint of $f$, we show by induction on $\mathbb{N}$ that $f^n(\bot) \sqsubseteq x \; \forall n \in \mathbb{N}$. It shows in particular that $f^k(\bot) \sqsubseteq x$.

Remark : $\top, f(\top), \ldots, f^n(\top), \ldots$ allows to compute the greatest fixpoint of $f$.

# Fixpoint computation: ascending chain condition

## Definition

A poset $(A, \sqsubseteq)$ verifies the ascending chain condition if for all ascending (increasing) sequence $x_0 \sqsubseteq x_1 \sqsubseteq \cdots \sqsubseteq x_n \sqsubseteq \cdots$ there exists an index $k$ from which the sequence is stationary ($\forall n \geq k,\ x_k = x_n$) (*i.e.* the sequence eventually stabilises).

## Corollary

*Let $(A, \sqsubseteq)$ a poset that verifies the ascending chain condition and $f$ a monotone function. The sequence $\bot, f(\bot), \ldots, f^n(\bot), \ldots$ eventually stabilises. Its limit is the least fixpoint of $f$.*

Remark : A finite poset verifies the ascending chain condition.

# Fixpoint computation: Kleene fixpoint theorem

## Definition

Let $(A, \sqsubseteq, \bigsqcup)$ a complete lattice. A function $f \in A \to A$ is continuous iff

$$\forall S \subseteq A,\ \sqcup f(S) = f(\sqcup S)$$

Remark : a continuous function is necessarily monotone.

## Theorem (Kleene fixpoint theorem)

*In a complete lattice $(A, \sqsubseteq, \bigsqcup)$, for all continuous function $f \in A \to A$, the least fixpoint $\mathbf{lfp}(f)$ of $f$ is equal to $\bigsqcup\{ f^n(\bot) \mid n \in \mathbb{N} \}$.*

Remark : the original theorem is stated for *complete partial order (CPO)*.

# Proof of the Kleene fixpoint theorem

- We have already shown that $f^n(\bot) \sqsubseteq f^{n+1}(\bot)$
- $\bigsqcup_{n\geq 0} f^n(\bot)$ is a fixpoint of $f$:

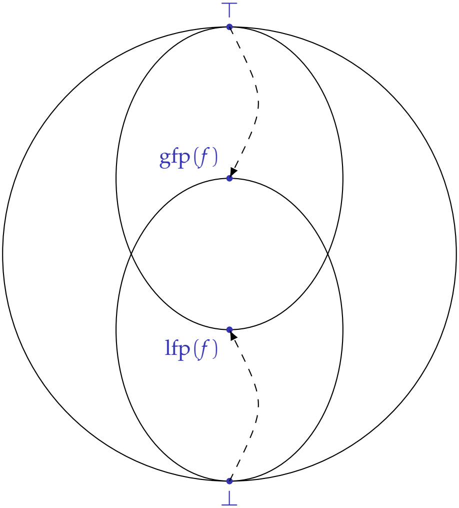
  $f(\bigsqcup_{n\geq 0} f^n(\bot))$

  $= \bigsqcup_{n\geq 0} f(f^n(\bot))$ — $f$ is continuous

  $= f^0(\bot) \sqcup \bigsqcup_{n\geq 0} f^{n+1}(\bot)$ — $(\bot \sqcup x = x)$ and def. $f^n$

  $= \bigsqcup_{n\geq 0} f^n(\bot)$
- It is the least fixpoint: consider $x \in Fix(f)$
  - $f^0(\bot) = \bot \sqsubseteq x$
  - $\forall n \geq 0 : f^n(\bot) \sqsubseteq x$ — induction on $n$, because $f$ monotone and $f(x) = x$
  - $\bigsqcup_{n\geq 0} f^n(\bot) \sqsubseteq x$ — greater bound

# Fixpoint computation




$\top, f(\top), \ldots, f^n(\top), \ldots, \mathbf{gfp}\, f$

$\bot, f(\bot), \ldots, f^n(\bot), \ldots, \mathbf{lfp}\, f$

# The underlying lattice structure of the Reaching definitions analysis

$\left(\mathcal{P}(Var \times Lab^?), \subseteq, \bigcup\right)$ is a complete lattice.

Lattice product: if $(L_1, \sqsubseteq_1, \bigsqcup_1)$ and $(L_2, \sqsubseteq_2), \bigsqcup_1$ are complete lattices, their *product* $L_1 \times L_2$ is the complete lattice $(L_1 \times L_2, \sqsubseteq_{L_1 \times L_2}, \bigsqcup_{L_1 \times L_2})$ defined par:

$$
\begin{array}{rcl}
(x_1, x_2) \sqsubseteq (y_1, y_2) & \Leftrightarrow & x_1 \sqsubseteq_1 y_1 \wedge x_2 \sqsubseteq_2 y_2 \\
\bigsqcup_{L_1 \times L_2} S & = & (\bigsqcup_1 proj_1(S), \bigsqcup_2 proj_2(S)),\ \forall S \subseteq L_1 \times L_2
\end{array}
$$

Conclusion :

$$(RD_s(1), RD_e(1), \ldots, RD_s(6), RD_e(6)) \in \mathcal{P}(Var \times Lab^?)^{12}$$

and $\left(\mathcal{P}(Var \times Lab^?)^{12}, \subseteq^{12}, \bigcup^{12}\right)$ is a complete lattice.

Exercise: Justify the termination of the analysis.

# Accelerated iterations

Consider the system

$$\begin{cases} x_1 & = & f_1(x_1, \ldots, x_n) \\ & \vdots & \\ x_n & = & f_n(x_1, \ldots, x_n) \end{cases}$$

Standard iterations:
$$\begin{array}{lcl} x_1^{i+1} & = & f_1(x_1^i, \ldots, x_n^i) \\ x_2^{i+1} & = & f_2(x_1^i, \ldots, x_n^i) \\ & \vdots & \\ x_n^{i+1} & = & f_n(x_1^i, \ldots, x_n^i) \end{array}$$

Chaotic iterations: at each step, we only use selected equations, without forgetting any equation infinitely often. $L \in \mathbb{N} \rightarrow \mathcal{P}(\{1, \ldots, n\})$ gives the iteration strategy (*i.e.* at the $i^{\text{th}}$ iteration, equations in $L_i$ are used).

$$\begin{array}{rl} x_j^{i+1} = f_j(x_1^i, \ldots, x_n^i) & \text{if } j \in L_{i+1} \\ x_j^{i+1} = x_j^i & \text{if } j \notin L_{i+1} \end{array}$$

# Example

Remark: the equation system can be simplified (at least by hand).

$$
\begin{array}{llll}
RD_{\text{in}}(1) = \{(v,?) \mid v \in Var\} & (e_1) & RD_{\text{out}}(1) = RD_{\text{in}}(1) \setminus \{(\mathbf{y},l) \mid l \in Lab^?\} \cup \{(\mathbf{y},1)\} & (s_1) \\
RD_{\text{in}}(2) = RD_{\text{out}}(1) & (e_2) & RD_{\text{out}}(2) = RD_{\text{in}}(2) \setminus \{(\mathbf{z},l) \mid l \in Lab^?\} \cup \{(\mathbf{z},2)\} & (s_2) \\
RD_{\text{in}}(3) = RD_{\text{out}}(2) \cup RD_{\text{out}}(5) & (e_3) & RD_{\text{out}}(3) = RD_{\text{in}}(3) & (s_3) \\
RD_{\text{in}}(4) = RD_{\text{out}}(3) & (e_4) & RD_{\text{out}}(4) = RD_{\text{in}}(4) \setminus \{(\mathbf{z},l) \mid l \in Lab^?\} \cup \{(\mathbf{z},4)\} & (s_4) \\
RD_{\text{in}}(5) = RD_{\text{out}}(4) & (e_5) & RD_{\text{out}}(5) = RD_{\text{in}}(5) \setminus \{(\mathbf{y},l) \mid l \in Lab^?\} \cup \{(\mathbf{y},5)\} & (s_5) \\
RD_{\text{in}}(6) = RD_{\text{out}}(3) & (e_6) & RD_{\text{out}}(6) = RD_{\text{in}}(6) \setminus \{(\mathbf{y},l) \mid l \in Lab^?\} \cup \{(\mathbf{y},6)\} & (s_6)
\end{array}
$$

# Example

Remark: the equation system can be simplified (at least by hand).

$$
\begin{array}{ll}
RD_{\text{in}}(1) = \{(\mathbf{x},?),(\mathbf{y},?),(\mathbf{z},?)\} \quad (e_1) & RD_{\text{out}}(1) = RD_{\text{in}}(1) \setminus \{(\mathbf{y},l) \mid l \in Lab^?\} \cup \{(\mathbf{y},1)\} \quad (s_1) \\
RD_{\text{in}}(2) = RD_{\text{out}}(1) \quad (e_2) & RD_{\text{out}}(2) = RD_{\text{in}}(2) \setminus \{(\mathbf{z},l) \mid l \in Lab^?\} \cup \{(\mathbf{z},2)\} \quad (s_2) \\
RD_{\text{in}}(3) = RD_{\text{out}}(2) \cup RD_{\text{out}}(5) \quad (e_3) & RD_{\text{out}}(3) = RD_{\text{in}}(3) \quad (s_3) \\
RD_{\text{in}}(4) = RD_{\text{out}}(3) \quad (e_4) & RD_{\text{out}}(4) = RD_{\text{in}}(4) \setminus \{(\mathbf{z},l) \mid l \in Lab^?\} \cup \{(\mathbf{z},4)\} \quad (s_4) \\
RD_{\text{in}}(5) = RD_{\text{out}}(4) \quad (e_5) & RD_{\text{out}}(5) = RD_{\text{in}}(5) \setminus \{(\mathbf{y},l) \mid l \in Lab^?\} \cup \{(\mathbf{y},5)\} \quad (s_5) \\
RD_{\text{in}}(6) = RD_{\text{out}}(3) \quad (e_6) & RD_{\text{out}}(6) = RD_{\text{in}}(6) \setminus \{(\mathbf{y},l) \mid l \in Lab^?\} \cup \{(\mathbf{y},6)\} \quad (s_6)
\end{array}
$$

# Example

Remark: the equation system can be simplified (at least by hand).

$$
\begin{array}{llll}
RD_{\text{in}}(1) = \{(\mathbf{x},?),(\mathbf{y},?),(\mathbf{z},?)\} & (e_1) & RD_{\text{out}}(1) = \{(\mathbf{x},?),(\mathbf{y},1),(\mathbf{z},?)\} & (s_1) \\
RD_{\text{in}}(2) = RD_{\text{out}}(1) & (e_2) & RD_{\text{out}}(2) = RD_{\text{in}}(2) \setminus \{(\mathbf{z},l) \mid l \in Lab^?\} \cup \{(\mathbf{z},2)\} & (s_2) \\
RD_{\text{in}}(3) = RD_{\text{out}}(2) \cup RD_{\text{out}}(5) & (e_3) & RD_{\text{out}}(3) = RD_{\text{in}}(3) & (s_3) \\
RD_{\text{in}}(4) = RD_{\text{out}}(3) & (e_4) & RD_{\text{out}}(4) = RD_{\text{in}}(4) \setminus \{(\mathbf{z},l) \mid l \in Lab^?\} \cup \{(\mathbf{z},4)\} & (s_4) \\
RD_{\text{in}}(5) = RD_{\text{out}}(4) & (e_5) & RD_{\text{out}}(5) = RD_{\text{in}}(5) \setminus \{(\mathbf{y},l) \mid l \in Lab^?\} \cup \{(\mathbf{y},5)\} & (s_5) \\
RD_{\text{in}}(6) = RD_{\text{out}}(3) & (e_6) & RD_{\text{out}}(6) = RD_{\text{in}}(6) \setminus \{(\mathbf{y},l) \mid l \in Lab^?\} \cup \{(\mathbf{y},6)\} & (s_6)
\end{array}
$$

# Example

Remark: the equation system can be simplified (at least by hand).

$$RD_{in}(1) = \{(\mathbf{x},?), (\mathbf{y},?), (\mathbf{z},?)\} \quad (e_1)$$
$$RD_{in}(2) = RD_{out}(1) \quad (e_2)$$
$$RD_{in}(3) = RD_{out}(2) \cup RD_{out}(5) \quad (e_3)$$
$$RD_{in}(4) = RD_{out}(3) \quad (e_4)$$
$$RD_{in}(5) = RD_{out}(4) \quad (e_5)$$
$$RD_{in}(6) = RD_{out}(3) \quad (e_6)$$

$$RD_{out}(1) = \{(\mathbf{x},?), (\mathbf{y},1), (\mathbf{z},?)\} \quad (s_1)$$
$$RD_{out}(2) = RD_{out}(1) \setminus \{(\mathbf{z},l) \mid l \in Lab^?\} \cup \{(\mathbf{z},2)\} \quad (s_2)$$
$$RD_{out}(3) = RD_{out}(2) \cup RD_{out}(5) \quad (s_3)$$
$$RD_{out}(4) = RD_{out}(3) \setminus \{(\mathbf{z},l) \mid l \in Lab^?\} \cup \{(\mathbf{z},4)\} \quad (s_4)$$
$$RD_{out}(5) = RD_{out}(4) \setminus \{(\mathbf{y},l) \mid l \in Lab^?\} \cup \{(\mathbf{y},5)\} \quad (s_5)$$
$$RD_{out}(6) = RD_{out}(3) \setminus \{(\mathbf{y},l) \mid l \in Lab^?\} \cup \{(\mathbf{y},6)\} \quad (s_6)$$

# Example

Remark: the equation system can be simplified (at least by hand).

$$
\begin{array}{llll}
RD_{\text{in}}(1) = \{(\mathbf{x},?),(\mathbf{y},?),(\mathbf{z},?)\} & (e_1) & RD_{\text{out}}(1) = \{(\mathbf{x},?),(\mathbf{y},1),(\mathbf{z},?)\} & (s_1) \\
RD_{\text{in}}(2) = RD_{\text{out}}(1) & (e_2) & RD_{\text{out}}(2) = \{(\mathbf{x},?),(\mathbf{y},1),(\mathbf{z},2)\} & (s_2) \\
RD_{\text{in}}(3) = RD_{\text{out}}(2) \cup RD_{\text{out}}(5) & (e_3) & RD_{\text{out}}(3) = RD_{\text{out}}(2) \cup RD_{\text{out}}(5) & (s_3) \\
RD_{\text{in}}(4) = RD_{\text{out}}(3) & (e_4) & RD_{\text{out}}(4) = RD_{\text{out}}(3) \setminus \{(\mathbf{z},l) \mid l \in Lab^?\} \cup \{(\mathbf{z},4)\} & (s_4) \\
RD_{\text{in}}(5) = RD_{\text{out}}(4) & (e_5) & RD_{\text{out}}(5) = RD_{\text{out}}(4) \setminus \{(\mathbf{y},l) \mid l \in Lab^?\} \cup \{(\mathbf{y},5)\} & (s_5) \\
RD_{\text{in}}(6) = RD_{\text{out}}(3) & (e_6) & RD_{\text{out}}(6) = RD_{\text{out}}(3) \setminus \{(\mathbf{y},l) \mid l \in Lab^?\} \cup \{(\mathbf{y},6)\} & (s_6)
\end{array}
$$

# Example

Remark: the equation system can be simplified (at least by hand).

$$
\begin{array}{llll}
RD_{\text{in}}(1) = \{(\mathbf{x},?),(\mathbf{y},?),(\mathbf{z},?)\} & (e_1) & RD_{\text{out}}(1) = \{(\mathbf{x},?),(\mathbf{y},1),(\mathbf{z},?)\} & (s_1) \\
RD_{\text{in}}(2) = RD_{\text{out}}(1) & (e_2) & RD_{\text{out}}(2) = \{(\mathbf{x},?),(\mathbf{y},1),(\mathbf{z},2)\} & (s_2) \\
RD_{\text{in}}(3) = RD_{\text{out}}(2) \cup RD_{\text{out}}(5) & (e_3) & RD_{\text{out}}(3) = \{(\mathbf{x},?),(\mathbf{y},1),(\mathbf{z},2)\} \cup RD_{\text{out}}(5) & (s_3) \\
RD_{\text{in}}(4) = RD_{\text{out}}(3) & (e_4) & RD_{\text{out}}(4) = RD_{\text{out}}(3) \setminus \{(\mathbf{z},l) \mid l \in Lab^{?}\} \cup \{(\mathbf{z},4)\} & (s_4) \\
RD_{\text{in}}(5) = RD_{\text{out}}(4) & (e_5) & RD_{\text{out}}(5) = RD_{\text{out}}(4) \setminus \{(\mathbf{y},l) \mid l \in Lab^{?}\} \cup \{(\mathbf{y},5)\} & (s_5) \\
RD_{\text{in}}(6) = RD_{\text{out}}(3) & (e_6) & RD_{\text{out}}(6) = RD_{\text{out}}(3) \setminus \{(\mathbf{y},l) \mid l \in Lab^{?}\} \cup \{(\mathbf{y},6)\} & (s_6)
\end{array}
$$

# Example

It is hence sufficient to solve the following system:

$$
\begin{aligned}
RD_{\text{out}}(3) &= \{(\mathtt{x},?), (\mathbf{y},1), (\mathbf{z},2)\} \cup RD_{\text{out}}(5) && (s_3)\\
RD_{\text{out}}(4) &= RD_{\text{out}}(3) \setminus \{(\mathbf{z},l) \mid l \in Lab^?\} \cup \{(\mathbf{z},4)\} && (s_4)\\
RD_{\text{out}}(5) &= RD_{\text{out}}(4) \setminus \{(\mathbf{y},l) \mid l \in Lab^?\} \cup \{(\mathbf{y},5)\} && (s_5)\\
RD_{\text{out}}(6) &= RD_{\text{out}}(3) \setminus \{(\mathbf{y},l) \mid l \in Lab^?\} \cup \{(\mathbf{y},6)\} && (s_6)
\end{aligned}
$$

# Example

It is hence sufficient to solve the following system:

$$RD_{out}(3) = \{(\mathbf{x},?), (\mathbf{y},1), (\mathbf{z},2)\} \cup RD_{out}(5) \qquad (s_3)$$
$$RD_{out}(4) = RD_{out}(3) \setminus \{(\mathbf{z},l) \mid l \in Lab^?\} \cup \{(\mathbf{z},4)\} \quad (s_4)$$
$$RD_{out}(5) = RD_{out}(4) \setminus \{(\mathbf{y},l) \mid l \in Lab^?\} \cup \{(\mathbf{y},5)\} \quad (s_5)$$
$$RD_{out}(6) = RD_{out}(3) \setminus \{(\mathbf{y},l) \mid l \in Lab^?\} \cup \{(\mathbf{y},6)\} \quad (s_6)$$

| $L_i$ | | $\{s_3\}$ | $\{s_4\}$ | $\{s_5\}$ |
|---|---|---|---|---|
| $RD_{out}(3)$ | $\emptyset$ | $\{(x,?),(y,1),(z,2)\}$ | $\{(x,?),(y,1),(z,2)\}$ | $\{(x,?),(y,1),(z,2)\}$ |
| $RD_{out}(4)$ | $\emptyset$ | $\emptyset$ | $\{(x,?),(y,1),(z,4)\}$ | $\{(x,?),(y,1),(z,4)\}$ |
| $RD_{out}(5)$ | $\emptyset$ | $\emptyset$ | $\emptyset$ | $\{(x,?),(y,5),(z,4)\}$ |
| $RD_{out}(6)$ | $\emptyset$ | $\emptyset$ | $\emptyset$ | $\emptyset$ |

| $\{s_3\}$ | $\{s_4, s_6\}$ |
|---|---|
| $\{(x,?),(y,1),(y,5),(z,2),(z,4)\}$ | $\{(x,?),(y,1),(y,5),(z,2),(z,4)\}$ |
| $\{(x,?),(y,1),(z,4)\}$ | $\{(x,?),(y,1),(y,5),(z,4)\}$ |
| $\{(x,?),(y,5),(z,4)\}$ | $\{(x,?),(y,5),(z,4)\}$ |
| $\emptyset$ | $\{(x,?),(y,6),(z,2),(z,4)\}$ |

# The Work-set Algorithm

**forall** $i \in \{1, \ldots, n\}$ **do** $x_i := \perp$;
$W := \{1, \ldots, n\}$
**repeat**
  $i := choose(W)$;
  $tmp := f_i(x_1, \ldots, x_n)$ ;
  **if** $tmp \neq x_i$ **then begin**  /* *the value of* $x_i$ *has changed* */
    $x_i := tmp$ ;
    $W := W \cup dependencies(x_i)$
**until** $W = \emptyset$

$choose(S)$: removes one element from a set $S$.
$dependencies(x)$: returns the set of variables that depends on a variable $x$.

The chaotic iteration (*i.e.* the choice of a good iteration order ) can be combined with the “work-set” technique (re-computation only when necessary).

# Control flow graph

- Definition of the While language with labels. Labels designate the program points where we stock some data flow information.
- Control flow graph of program and extraction from labeled programs.
- Generating data flow equations from control flow graphs.

# The While language with labels

We re-use the syntax of While that has been presented during the lecture on operational semantics.

$$S ::= \mathtt{x} := a \mid \mathtt{skip} \mid S_1; S_2 \mid \mathtt{if}\ b\ \mathtt{then}\ S_1\ \mathtt{else}\ S_2 \mid \mathtt{while}\ b\ \mathtt{do}\ S$$

$n \in Num,\ \mathtt{x} \in Var,\ a \in Aexp,\ b \in Bexp,\ S \in Comm$

# The While language with labels

We re-use the syntax of While that has been presented during the lecture on operational semantics.

$$S ::= [\mathtt{x} := a]^l \mid [\mathtt{skip}]^l \mid S_1; S_2 \mid \mathtt{if}\ [b]^l\ \mathtt{then}\ S_1\ \mathtt{else}\ S_2 \mid \mathtt{while}\ [b]^l\ \mathtt{do}\ S$$

$n \in Num$, $\mathtt{x} \in Var$, $a \in Aexp$, $b \in Bexp$, $S \in Comm$, $l \in Lab$

enriched with labels.

The labels allow to attach the analysis results.

# Control flow graph

We associate at each instruction $S \in Comm$:

$$
\begin{array}{lll}
init(S) & \in Lab & : \text{ entry point label of } S \\
final(S) & \subseteq Lab & : \text{ exit point labels of } S \\
labels(S) & \subseteq Lab & : \text{ labels which appear in } S \\
flow(S) & \subseteq Lab \times Lab & : \text{ edges of the control flow graph}
\end{array}
$$

Example :

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

$$
\begin{array}{rcl}
init(\texttt{power}) & = & 1 \\
final(\texttt{power}) & = & \{2\} \\
labels(\texttt{power}) & = & \{1,2,3,4\} \\
flow(\texttt{power}) & = & \{(1,2),\ (2,3),\ (3,4),\ (4,2)\}
\end{array}
$$

# Control flow graph

Each function is defined by induction on the While syntax.

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

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

# Control flow graph

Each function is defined by induction on the While syntax.

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

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

# Control flow graph

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

# Control flow graph

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

# Control flow graph

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

We only consider programs with distinct labels

- for all $l \in labels(S)$, $[B]^l \in S$ identifies, without ambiguity, the elementary block ($[\mathbf{x} := a]^l$ or $[\mathbf{skip}]^l$) or the test ($[b]^l$) which appears in $S$.

We suppose also that no flow reaches the initial point of the program.

# Control flow graph

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

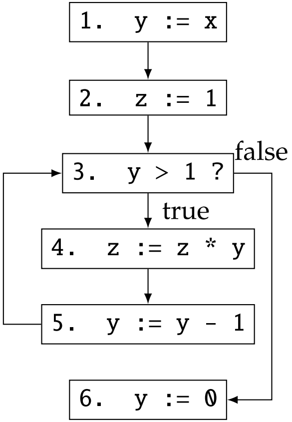
We only consider programs with distinct labels

- for all $l \in labels(S)$, $[B]^l \in S$ identifies, without ambiguity, the elementary block ($[\mathbf{x} := a]^l$ or $[\mathbf{skip}]^l$) or the test ($[b]^l$) which appears in $S$.

We suppose also that no flow reaches the initial point of the program.

# Remember : reachable definition analysis

$$
\begin{aligned}
RD_{\text{in}}(1) &= \{(v,?) \mid v \in Var\} \\
RD_{\text{out}}(1) &= RD_{\text{in}}(1) \setminus \{(\mathbf{y},l) \mid l \in Lab^?\} \cup \{(\mathbf{y},1)\} \\
RD_{\text{in}}(2) &= RD_{\text{out}}(1) \\
RD_{\text{out}}(2) &= RD_{\text{in}}(2) \setminus \{(\mathbf{z},l) \mid l \in Lab^?\} \cup \{(\mathbf{z},2)\} \\
RD_{\text{in}}(3) &= RD_{\text{out}}(2) \cup RD_{\text{out}}(5) \\
RD_{\text{out}}(3) &= RD_{\text{in}}(3) \\
RD_{\text{in}}(4) &= RD_{\text{out}}(3) \\
RD_{\text{out}}(4) &= RD_{\text{in}}(4) \setminus \{(\mathbf{z},l) \mid l \in Lab^?\} \cup \{(\mathbf{z},4)\} \\
RD_{\text{in}}(5) &= RD_{\text{out}}(4) \\
RD_{\text{out}}(5) &= RD_{\text{in}}(5) \setminus \{(\mathbf{y},l) \mid l \in Lab^?\} \cup \{(\mathbf{y},5)\} \\
RD_{\text{in}}(6) &= RD_{\text{out}}(3) \\
RD_{\text{out}}(6) &= RD_{\text{in}}(6) \setminus \{(\mathbf{y},l) \mid l \in Lab^?\} \cup \{(\mathbf{y},6)\}
\end{aligned}
$$



# Data flow equations of reachable definitions

Domain of data flow properties:

$$RD_{\text{in}}(l), RD_{\text{out}}(l) \in \wp(Var \times Lab^?)$$

"Gen-kill" rules

$$\begin{aligned}
kill([\mathtt{x} := a]^l) &= \{(\mathtt{x}, l') \mid l' \in Lab^?\} \\
kill([\mathtt{skip}]^l) &= \emptyset \\
kill([b]^l) &= \emptyset \\
\\
gen([\mathtt{x} := a]^l) &= \{(x, l)\} \\
gen([\mathtt{skip}]^l) &= \emptyset \\
gen([b]^l) &= \emptyset
\end{aligned}$$

For all program points $[\ ]^l \in P$,

$$\begin{aligned}
RD_{\text{in}}(l) &= \begin{cases} \{(\mathtt{x}, ?) \mid \mathtt{x} \in Var\} & \text{if } l = init(P) \\ \displaystyle\bigcup_{(l', l) \in flow(P)} RD_{\text{out}}(l') & \end{cases} \\
RD_{\text{out}}(l) &= RD_{\text{in}}(l) \setminus kill([b]^l) \cup gen([b]^l)
\end{aligned}$$

# Available expressions

Determine the expressions whose value is already available in a variable.

Domain of values: $AE_{\text{in}}(l), AE_{\text{out}}(l) \in \mathcal{P}(Var \times Aexp)$

$$
\begin{aligned}
kill([\mathtt{x} := a]^l) &= \\
kill([\mathtt{skip}]^l) &= \\
kill([b]^l) &=
\end{aligned}
$$

$$
\begin{aligned}
gen([\mathtt{x} := a]^l) &= \\
gen([\mathtt{skip}]^l) &= \\
gen([b]^l) &=
\end{aligned}
$$

For all program points $[\ ]^l \in P$,

$$AE_{\text{in}}(l) =$$

$$AE_{\text{out}}(l) =$$

# Available expressions

Determine the expressions whose value is already available in a variable.

Domain of values: $AE_{\text{in}}(l), AE_{\text{out}}(l) \in \mathcal{P}(Var \times Aexp)$

$$\begin{array}{lcl}
kill([x := a]^l) & = & \{(v, e) \mid x \in Var(e)\} \cup \{(x, e) \mid e \in Aexp\} \\
kill([\texttt{skip}]^l) & = & \emptyset \\
kill([b]^l) & = & \emptyset
\end{array}$$

$$\begin{array}{lcl}
gen([x := a]^l) & = & \{(x, a) \mid x \notin Var(a)\} \\
gen([\texttt{skip}]^l) & = & \emptyset \\
gen([b]^l) & = & \emptyset
\end{array}$$

For all program points $[\ ]^l \in P$,

$$AE_{\text{in}}(l) = \begin{cases} \emptyset & \text{if } l = init(P) \\ \bigcap_{(l', l) \in flow(P)} AE_{\text{out}}(l') & \end{cases}$$

$$AE_{\text{out}}(l) = AE_{\text{in}}(l) \setminus kill([b]^l) \cup gen([b]^l)$$

# Exercise : available expressions

Build and solve the equation system for the available expression analysis of the following program:

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

# Exercise : available expressions

Build and solve the equation system for the available expression analysis of the following program:

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

$$
\begin{aligned}
AE_{\text{in}}(1) &= \emptyset \\
AE_{\text{in}}(2) &= AE_{\text{out}}(1) \\
AE_{\text{in}}(3) &= AE_{\text{out}}(2) \cap AE_{\text{out}}(5) \\
AE_{\text{in}}(4) &= AE_{\text{out}}(3) \\
AE_{\text{in}}(5) &= AE_{\text{out}}(4) \\
AE_{\text{out}}(1) &= AE_{\text{in}}(1) \setminus \{(\mathtt{x}, \mathtt{a}+\mathtt{b}), (\mathtt{x}, \mathtt{a}*\mathtt{b}), (\mathtt{x}, \mathtt{a}+\mathtt{1})\} \cup \{(\mathtt{x}, \mathtt{a}+\mathtt{b})\} \\
AE_{\text{out}}(2) &= AE_{\text{in}}(2) \setminus \{(\mathtt{y}, \mathtt{a}+\mathtt{b}), (\mathtt{y}, \mathtt{a}*\mathtt{b}), (\mathtt{y}, \mathtt{a}+\mathtt{1})\} \cup \{(\mathtt{y}, \mathtt{a}*\mathtt{b})\} \\
AE_{\text{out}}(3) &= AE_{\text{in}}(3) \\
AE_{\text{out}}(4) &= AE_{\text{in}}(4) \setminus (\{\mathtt{x}, \mathtt{y}\} \times \{\mathtt{a}+\mathtt{b}, \mathtt{a}*\mathtt{b}, \mathtt{a}+\mathtt{1}\}) \\
AE_{\text{out}}(5) &= AE_{\text{in}}(5) \setminus \{(\mathtt{x}, \mathtt{a}+\mathtt{b}), (\mathtt{x}, \mathtt{a}*\mathtt{b}), (\mathtt{x}, \mathtt{a}+\mathtt{1})\} \cup \{(\mathtt{x}, \mathtt{a}+\mathtt{b})\}
\end{aligned}
$$

# Exercise : available expressions

Build and solve the equation system for the available expression analysis of the following program:

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

# Exercise : available expressions

Build and solve the equation system for the available expression analysis of the following program:

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

$$
\begin{aligned}
AE_{in}(1) &= \emptyset \\
AE_{out}(1) &= \{(x, a + b)\} \\
AE_{in}(2) &= \{(x, a + b)\} \\
AE_{out}(2) &= \{(x, a + b), (y, a * b)\}
\end{aligned}
$$

$$
\begin{aligned}
AE_{in}(3) &= \{(x, a + b), (y, a * b)\} \cap AE_{out}(5) \\
AE_{in}(4) &= AE_{out}(3) \\
AE_{in}(5) &= AE_{out}(4)
\end{aligned}
$$

$$
\begin{aligned}
AE_{out}(3) &= \{(x, a + b), (y, a * b)\} \cap AE_{out}(5) \\
AE_{out}(4) &= AE_{out}(3) \setminus (\{x, y\} \times \{a + b, a * b, a + 1\}) \\
AE_{out}(5) &= AE_{out}(4) \setminus \{(x, a + b), (x, a * b), (x, a + 1)\} \cup \{(x, a + b)\}
\end{aligned}
$$

# Exercise : available expressions

Build and solve the equation system for the available expression analysis of the following program:
$[\mathtt{x} := \mathtt{a} + \mathtt{b}]^1; [\mathtt{y} := \mathtt{a} * \mathtt{b}]^2; \mathbf{while}\ [\mathtt{y} > \mathtt{a} + \mathtt{b}]^3\ \mathbf{do}\ ([\mathtt{a} := \mathtt{a} + 1]^4; [\mathtt{x} := \mathtt{a} + \mathtt{b}]^5)$

$$
\begin{aligned}
AE_{\text{out}}(3) &= \{(\mathtt{x}, \mathtt{a}+\mathtt{b}), (\mathtt{y}, \mathtt{a}*\mathtt{b})\} \cap AE_{\text{out}}(5) && (1)\\
AE_{\text{out}}(4) &= AE_{\text{out}}(3) \setminus (\{\mathtt{x}, \mathtt{y}\} \times \{\mathtt{a}+\mathtt{b}, \mathtt{a}*\mathtt{b}, \mathtt{a}+1\}) && (2)\\
AE_{\text{out}}(5) &= AE_{\text{out}}(4) \setminus \{(\mathtt{x}, \mathtt{a}+\mathtt{b}), (\mathtt{x}, \mathtt{a}*\mathtt{b}), (\mathtt{x}, \mathtt{a}+1)\} \cup \{(\mathtt{x}, \mathtt{a}+\mathtt{b})\} && (3)
\end{aligned}
$$

We are looking for greatest fixpoint, hence we start from
$\top = \{\mathtt{x}, \mathtt{y}, \mathtt{a}\} \times \{\mathtt{a}+\mathtt{b}, \mathtt{a}*\mathtt{b}, \mathtt{a}+1\}$

| | | $\{1\}$ | $\{2\}$ | $\{3\}$ | $\{1\}$ | |
|---|---|---|---|---|---|---|
| $AE_{\text{out}}(3)$ | $\top$ | $\{(\mathtt{x}, \mathtt{a}+\mathtt{b}), (\mathtt{y}, \mathtt{a}*\mathtt{b})\}$ | — | — | $\{(\mathtt{x}, \mathtt{a}+\mathtt{b})\}$ | stable |
| $AE_{\text{out}}(4)$ | $\top$ | — | $\emptyset$ | — | — | stable |
| $AE_{\text{out}}(5)$ | $\top$ | — | — | $\{(\mathtt{x}, \mathtt{a}+\mathtt{b})$ | — | stable |

# Exercise : available expressions

Build and solve the equation system for the available expression analysis of the following program:

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

$$
\begin{aligned}
AE_{in}(1) &= \emptyset \\
AE_{out}(1) &= \{(\mathtt{x}, \mathtt{a} + \mathtt{b})\} \\
AE_{in}(2) &= \{(\mathtt{x}, \mathtt{a} + \mathtt{b})\} \\
AE_{out}(2) &= \{(\mathtt{x}, \mathtt{a} + \mathtt{b}), (\mathtt{y}, \mathtt{a} * \mathtt{b})\} \\
\\
AE_{in}(3) &= \{(\mathtt{x}, \mathtt{a} + \mathtt{b})\} \\
AE_{in}(4) &= \{(\mathtt{x}, \mathtt{a} + \mathtt{b})\} \\
AE_{in}(5) &= \emptyset \\
\\
AE_{out}(3) &= \{(\mathtt{x}, \mathtt{a} + \mathtt{b})\} \\
AE_{out}(4) &= \emptyset \\
AE_{out}(5) &= \{(\mathtt{x}, \mathtt{a} + \mathtt{b})\}
\end{aligned}
$$

# Live variables

A variable is *live* if it is used before being redefined.

Domain of values: $LV_{\text{in}}(l), LV_{\text{out}}(l) \in \mathcal{P}(Var)$

$$
\begin{aligned}
kill([\mathtt{x} := a]^l) &= \\
kill([\mathtt{skip}]^l) &= \\
kill([b]^l) &=
\end{aligned}
$$

$$
\begin{aligned}
gen([\mathtt{x} := a]^l) &= \\
gen([\mathtt{skip}]^l) &= \\
gen([b]^l) &=
\end{aligned}
$$

For all $[b]^l \in P$,

$$LV_{\text{in}}(l) =$$

$$LV_{\text{out}}(l) =$$

# Live variables

A variable is *live* if it is used before being redefined.

Domain of values: $LV_{in}(l), LV_{out}(l) \in \mathcal{P}(Var)$

$$
\begin{array}{lcl}
kill([\mathtt{x} := a]^l) & = & \{\mathtt{x}\} \\
kill([\mathtt{skip}]^l) & = & \emptyset \\
kill([b]^l) & = & \emptyset \\
 & & \\
gen([\mathtt{x} := a]^l) & = & \{\mathtt{y} \mid \mathtt{y} \in Var(a)\} \\
gen([\mathtt{skip}]^l) & = & \emptyset \\
gen([b]^l) & = & \{\mathtt{y} \mid \mathtt{y} \in Var(b)\}
\end{array}
$$

For all $[b]^l \in P$,

$$
\begin{array}{lcl}
LV_{in}(l) & = & LV_{out}(l) \setminus \mathbf{kill}([b]^l) \cup \mathbf{gen}([b]^l) \\
LV_{out}(l) & = & \begin{cases} \emptyset & \text{if } l \in final(P) \\ \displaystyle\bigcup_{(l,l') \in \mathbf{flow}(P)} LV_{in}(l') & \end{cases}
\end{array}
$$

# Exercise : live variables

Build and solve the equation system for the live variables analysis of the following program:

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

# Exercise : live variables

Build and solve the equation system for the live variables analysis of the following program:

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

We are looking for the least fixpoint, but thanks to symbolic simplification, we show that there is only one solution here.

$$
\begin{array}{lcl@{\qquad}lcl}
LV_{out}(1) & = & LV_{in}(2) & LV_{in}(1) & = & LV_{out}(1) \setminus \{\mathtt{x}\} \\
LV_{out}(2) & = & LV_{in}(3) & LV_{in}(2) & = & LV_{out}(2) \setminus \{\mathtt{y}\} \\
LV_{out}(3) & = & LV_{in}(4) & LV_{in}(3) & = & LV_{out}(3) \setminus \{\mathtt{x}\} \\
LV_{out}(4) & = & LV_{in}(5) \cup LV_{in}(6) & LV_{in}(4) & = & LV_{out}(4) \cup \{\mathtt{x}, \mathtt{y}\} \\
LV_{out}(5) & = & LV_{in}(7) & LV_{in}(5) & = & LV_{out}(5) \setminus \{\mathtt{z}\} \cup \{\mathtt{y}\} \\
LV_{out}(6) & = & LV_{in}(7) & LV_{in}(6) & = & LV_{out}(6) \setminus \{\mathtt{z}\} \cup \{\mathtt{y}\} \\
LV_{out}(7) & = & \emptyset & LV_{in}(7) & = & LV_{out}(7) \setminus \{\mathtt{x}\} \cup \{\mathtt{z}\}
\end{array}
$$

# Exercise : live variables

Build and solve the equation system for the live variables analysis of the following program:

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

$$
\begin{array}{lcl}
LV_{out}(1) & = & LV_{in}(2) \\
LV_{out}(2) & = & LV_{in}(3) \\
LV_{out}(3) & = & LV_{in}(4) \\
LV_{out}(4) & = & LV_{in}(5) \cup LV_{in}(6) \\
LV_{out}(5) & = & LV_{in}(7) \\
LV_{out}(6) & = & LV_{in}(7) \\
LV_{out}(7) & = & \emptyset
\end{array}
\qquad
\begin{array}{lcl}
LV_{in}(1) & = & LV_{out}(1) \setminus \{\mathtt{x}\} \\
LV_{in}(2) & = & LV_{out}(2) \setminus \{\mathtt{y}\} \\
LV_{in}(3) & = & LV_{out}(3) \setminus \{\mathtt{x}\} \\
LV_{in}(4) & = & LV_{out}(4) \cup \{\mathtt{x}, \mathtt{y}\} \\
LV_{in}(5) & = & LV_{out}(5) \setminus \{\mathtt{z}\} \cup \{\mathtt{y}\} \\
LV_{in}(6) & = & LV_{out}(6) \setminus \{\mathtt{z}\} \cup \{\mathtt{y}\} \\
LV_{in}(7) & = & \{\mathtt{z}\}
\end{array}
$$

# Exercise : live variables

Build and solve the equation system for the live variables analysis of the following program:

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

$$
\begin{array}{lcllcl}
LV_{\text{out}}(1) & = & LV_{\text{in}}(2) & LV_{\text{in}}(1) & = & LV_{\text{out}}(1) \setminus \{\mathtt{x}\} \\
LV_{\text{out}}(2) & = & LV_{\text{in}}(3) & LV_{\text{in}}(2) & = & LV_{\text{out}}(2) \setminus \{\mathtt{y}\} \\
LV_{\text{out}}(3) & = & LV_{\text{in}}(4) & LV_{\text{in}}(3) & = & LV_{\text{out}}(3) \setminus \{\mathtt{x}\} \\
LV_{\text{out}}(4) & = & LV_{\text{in}}(5) \cup LV_{\text{in}}(6) & LV_{\text{in}}(4) & = & LV_{\text{out}}(4) \cup \{\mathtt{x}, \mathtt{y}\} \\
LV_{\text{out}}(5) & = & \{\mathtt{z}\} & LV_{\text{in}}(5) & = & LV_{\text{out}}(5) \setminus \{\mathtt{z}\} \cup \{\mathtt{y}\} \\
LV_{\text{out}}(6) & = & \{\mathtt{z}\} & LV_{\text{in}}(6) & = & LV_{\text{out}}(6) \setminus \{\mathtt{z}\} \cup \{\mathtt{y}\} \\
LV_{\text{out}}(7) & = & \emptyset & LV_{\text{in}}(7) & = & \{\mathtt{z}\}
\end{array}
$$

# Exercise : live variables

Build and solve the equation system for the live variables analysis of the following program:

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

$$
\begin{array}{lcl}
LV_{\text{out}}(1) & = & LV_{\text{in}}(2) \\
LV_{\text{out}}(2) & = & LV_{\text{in}}(3) \\
LV_{\text{out}}(3) & = & LV_{\text{in}}(4) \\
LV_{\text{out}}(4) & = & LV_{\text{in}}(5) \cup LV_{\text{in}}(6) \\
LV_{\text{out}}(5) & = & \{\mathtt{z}\} \\
LV_{\text{out}}(6) & = & \{\mathtt{z}\} \\
LV_{\text{out}}(7) & = & \emptyset
\end{array}
\qquad
\begin{array}{lcl}
LV_{\text{in}}(1) & = & LV_{\text{out}}(1) \setminus \{\mathtt{x}\} \\
LV_{\text{in}}(2) & = & LV_{\text{out}}(2) \setminus \{\mathtt{y}\} \\
LV_{\text{in}}(3) & = & LV_{\text{out}}(3) \setminus \{\mathtt{x}\} \\
LV_{\text{in}}(4) & = & LV_{\text{out}}(4) \cup \{\mathtt{x}, \mathtt{y}\} \\
LV_{\text{in}}(5) & = & \{\mathtt{y}\} \\
LV_{\text{in}}(6) & = & \{\mathtt{y}\} \\
LV_{\text{in}}(7) & = & \{\mathtt{z}\}
\end{array}
$$

# Exercise : live variables

Build and solve the equation system for the live variables analysis of the following program:

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

$$
\begin{array}{lcllcl}
LV_{out}(1) & = & LV_{in}(2) & LV_{in}(1) & = & LV_{out}(1) \setminus \{\mathtt{x}\} \\
LV_{out}(2) & = & LV_{in}(3) & LV_{in}(2) & = & LV_{out}(2) \setminus \{\mathtt{y}\} \\
LV_{out}(3) & = & LV_{in}(4) & LV_{in}(3) & = & LV_{out}(3) \setminus \{\mathtt{x}\} \\
LV_{out}(4) & = & \{\mathtt{y}\} & LV_{in}(4) & = & LV_{out}(4) \cup \{\mathtt{x}, \mathtt{y}\} \\
LV_{out}(5) & = & \{\mathtt{z}\} & LV_{in}(5) & = & \{\mathtt{y}\} \\
LV_{out}(6) & = & \{\mathtt{z}\} & LV_{in}(6) & = & \{\mathtt{y}\} \\
LV_{out}(7) & = & \emptyset & LV_{in}(7) & = & \{\mathtt{z}\}
\end{array}
$$

# Exercise : live variables

Build and solve the equation system for the live variables analysis of the following program:

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

$$
\begin{array}{lcllcl}
LV_{\text{out}}(1) & = & LV_{\text{in}}(2) & LV_{\text{in}}(1) & = & LV_{\text{out}}(1) \setminus \{\mathtt{x}\} \\
LV_{\text{out}}(2) & = & LV_{\text{in}}(3) & LV_{\text{in}}(2) & = & LV_{\text{out}}(2) \setminus \{\mathtt{y}\} \\
LV_{\text{out}}(3) & = & LV_{\text{in}}(4) & LV_{\text{in}}(3) & = & LV_{\text{out}}(3) \setminus \{\mathtt{x}\} \\
LV_{\text{out}}(4) & = & \{\mathtt{y}\} & LV_{\text{in}}(4) & = & \{\mathtt{x}, \mathtt{y}\} \\
LV_{\text{out}}(5) & = & \{\mathtt{z}\} & LV_{\text{in}}(5) & = & \{\mathtt{y}\} \\
LV_{\text{out}}(6) & = & \{\mathtt{z}\} & LV_{\text{in}}(6) & = & \{\mathtt{y}\} \\
LV_{\text{out}}(7) & = & \emptyset & LV_{\text{in}}(7) & = & \{\mathtt{z}\}
\end{array}
$$

# Exercise : live variables

Build and solve the equation system for the live variables analysis of the following program:

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

$$
\begin{array}{lcllcl}
LV_{out}(1) & = & LV_{in}(2) & LV_{in}(1) & = & LV_{out}(1) \setminus \{\mathtt{x}\} \\
LV_{out}(2) & = & LV_{in}(3) & LV_{in}(2) & = & LV_{out}(2) \setminus \{\mathtt{y}\} \\
LV_{out}(3) & = & \{\mathtt{x}, \mathtt{y}\} & LV_{in}(3) & = & LV_{out}(3) \setminus \{\mathtt{x}\} \\
LV_{out}(4) & = & \{\mathtt{y}\} & LV_{in}(4) & = & \{\mathtt{x}, \mathtt{y}\} \\
LV_{out}(5) & = & \{\mathtt{z}\} & LV_{in}(5) & = & \{\mathtt{y}\} \\
LV_{out}(6) & = & \{\mathtt{z}\} & LV_{in}(6) & = & \{\mathtt{y}\} \\
LV_{out}(7) & = & \emptyset & LV_{in}(7) & = & \{\mathtt{z}\}
\end{array}
$$

# Exercise : live variables

Build and solve the equation system for the live variables analysis of the following program:

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

$$
\begin{array}{lcllcl}
LV_{out}(1) & = & LV_{in}(2) & LV_{in}(1) & = & LV_{out}(1) \setminus \{\mathtt{x}\} \\
LV_{out}(2) & = & LV_{in}(3) & LV_{in}(2) & = & LV_{out}(2) \setminus \{\mathtt{y}\} \\
LV_{out}(3) & = & \{\mathtt{x}, \mathtt{y}\} & LV_{in}(3) & = & \{\mathtt{y}\} \\
LV_{out}(4) & = & \{\mathtt{y}\} & LV_{in}(4) & = & \{\mathtt{x}, \mathtt{y}\} \\
LV_{out}(5) & = & \{\mathtt{z}\} & LV_{in}(5) & = & \{\mathtt{y}\} \\
LV_{out}(6) & = & \{\mathtt{z}\} & LV_{in}(6) & = & \{\mathtt{y}\} \\
LV_{out}(7) & = & \emptyset & LV_{in}(7) & = & \{\mathtt{z}\}
\end{array}
$$

# Exercise : live variables

Build and solve the equation system for the live variables analysis of the following program:

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

$$
\begin{array}{lcl lcl}
LV_{out}(1) & = & LV_{in}(2) & LV_{in}(1) & = & LV_{out}(1) \setminus \{\mathtt{x}\} \\
LV_{out}(2) & = & \{\mathtt{y}\} & LV_{in}(2) & = & LV_{out}(2) \setminus \{\mathtt{y}\} \\
LV_{out}(3) & = & \{\mathtt{x}, \mathtt{y}\} & LV_{in}(3) & = & \{\mathtt{y}\} \\
LV_{out}(4) & = & \{\mathtt{y}\} & LV_{in}(4) & = & \{\mathtt{x}, \mathtt{y}\} \\
LV_{out}(5) & = & \{\mathtt{z}\} & LV_{in}(5) & = & \{\mathtt{y}\} \\
LV_{out}(6) & = & \{\mathtt{z}\} & LV_{in}(6) & = & \{\mathtt{y}\} \\
LV_{out}(7) & = & \emptyset & LV_{in}(7) & = & \{\mathtt{z}\}
\end{array}
$$

# Exercise : live variables

Build and solve the equation system for the live variables analysis of the following program:

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

$$
\begin{array}{lcllcl}
LV_{out}(1) & = & LV_{in}(2) & LV_{in}(1) & = & LV_{out}(1) \setminus \{\mathtt{x}\} \\
LV_{out}(2) & = & \{\mathtt{y}\} & LV_{in}(2) & = & \emptyset \\
LV_{out}(3) & = & \{\mathtt{x}, \mathtt{y}\} & LV_{in}(3) & = & \{\mathtt{y}\} \\
LV_{out}(4) & = & \{\mathtt{y}\} & LV_{in}(4) & = & \{\mathtt{x}, \mathtt{y}\} \\
LV_{out}(5) & = & \{\mathtt{z}\} & LV_{in}(5) & = & \{\mathtt{y}\} \\
LV_{out}(6) & = & \{\mathtt{z}\} & LV_{in}(6) & = & \{\mathtt{y}\} \\
LV_{out}(7) & = & \emptyset & LV_{in}(7) & = & \{\mathtt{z}\}
\end{array}
$$

# Exercise : live variables

Build and solve the equation system for the live variables analysis of the following program:

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

$$
\begin{array}{lcllcl}
LV_{out}(1) & = & \emptyset & LV_{in}(1) & = & \emptyset \\
LV_{out}(2) & = & \{\mathtt{y}\} & LV_{in}(2) & = & \emptyset \\
LV_{out}(3) & = & \{\mathtt{x}, \mathtt{y}\} & LV_{in}(3) & = & \{\mathtt{y}\} \\
LV_{out}(4) & = & \{\mathtt{y}\} & LV_{in}(4) & = & \{\mathtt{x}, \mathtt{y}\} \\
LV_{out}(5) & = & \{\mathtt{z}\} & LV_{in}(5) & = & \{\mathtt{y}\} \\
LV_{out}(6) & = & \{\mathtt{z}\} & LV_{in}(6) & = & \{\mathtt{y}\} \\
LV_{out}(7) & = & \emptyset & LV_{in}(7) & = & \{\mathtt{z}\}
\end{array}
$$

# Analysis classification

The analyses we have presented deal with program execution path.

2 classification criteria:

1. The information is propagated forward or backward on paths;
2. The property deal with
   - at least one execution (something may happen, use of $\bigcup$, least fixpoint)
   - or all execution (something must happen, use of $\bigcap$, greatest fixpoint);

| | Forward analysis | Backward analysis |
|---|---|---|
| may | Reachable definitions | Live variables |
| must | Available expressions | Very busy expressions |

Remark: In Abstraction Interpretation (see PAS), we always focus on least fixpoint for a well chosen partial order ($\subseteq, \supseteq, \ldots$).

# Cooking a dataflow analysis

1. Formalize the property you want to track.
2. Describe the equation system attached to each program.
   - forward / backward ?
   - may / must information ?
     - ($\bigcup$, least fixpoint) or ($\bigcap$, greatest fixpoint)
3. Explain why the least/greatest fixpoint exist.
4. Explain why Kleene fixpoint iteration will terminate.