Static Analyses

Summary of analyses and transformations

Analysis	Transformation
Available expressions analysis	Common subexpression elimination
Reaching Definition	Invariant code motion
Live variables analysis	Dead code elimination
Detection of induction variables	Strength reduction
Equivalent expression analysis	Copy propagation

The essence of program analysis

Program analysis offers techniques for predicting statically

at compile-time

safe & efficient approximations
to the set of configurations or behaviours arising dynamically at run-time

Safe: faithful to the semantics

Efficient: implementation with

- good time performance and

- low space consumption

Why approximations?

read(x); (if x>0 then y=1 else (y=2;S)); z=y

Assume S does not contain any assignment to y

Which values of y can reach the assignment z=y??

It depends if S diverges

Correct (Safe) approximations :

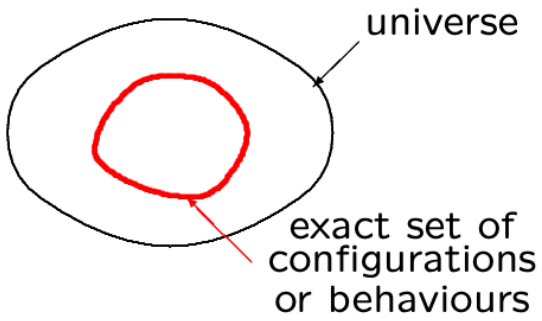
- {1,17} if S diverges
- {1,2,5, 27} otherwise

Best (Precise) approximations :

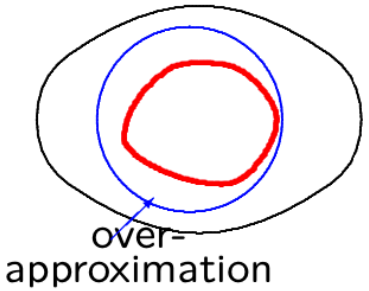
- {1} in the first case
- {1,2} otherwise

The nature of approximations

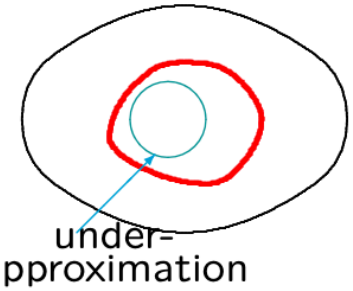
The exact world



Over-approximation



Under-approximation



Trade precision for efficiency!

but approximation have to be on the right side!

Approaches to Program Analysis

A family of techniques . . .

- data flow analysis
- constraint based analysis
- type and effect systems
- abstract interpretation

that differs in their focus

- algorithmic methods
- semantic foundations
- language paradigms
  - imperative/procedural
  - object oriented
  - logical
  - functional
  - concurrent/distributive
  - mobile

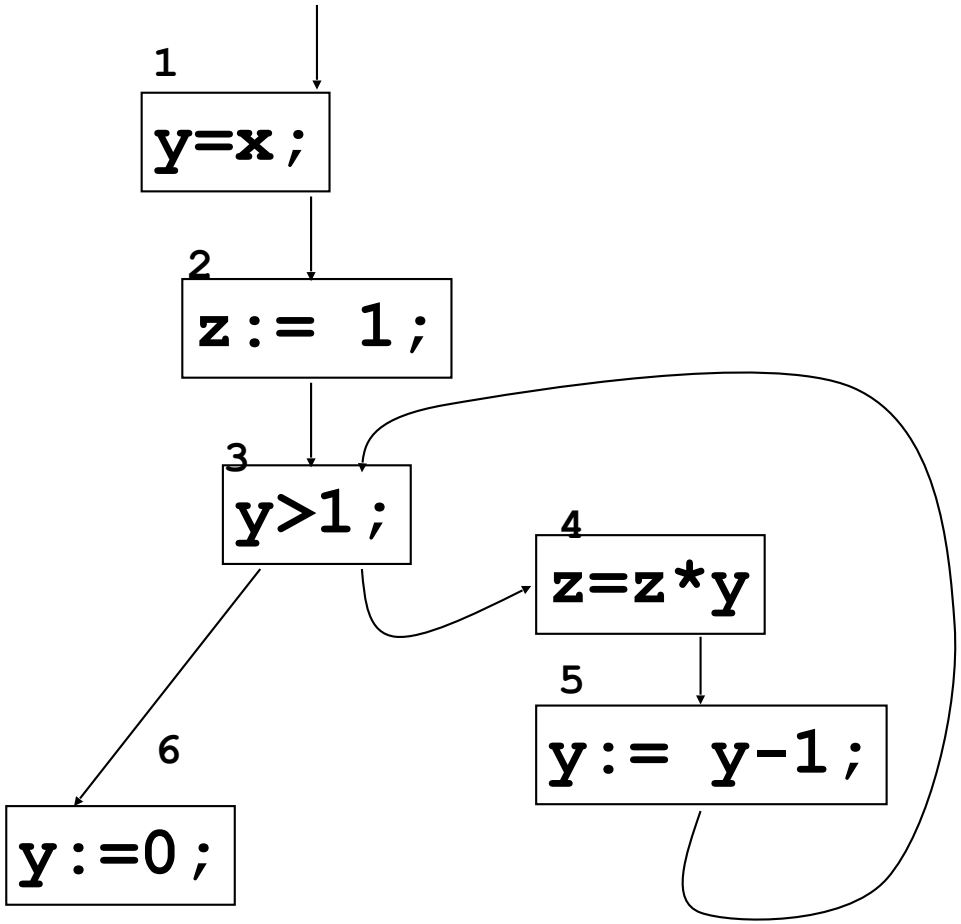
Dataflow Analysis

The ideas :

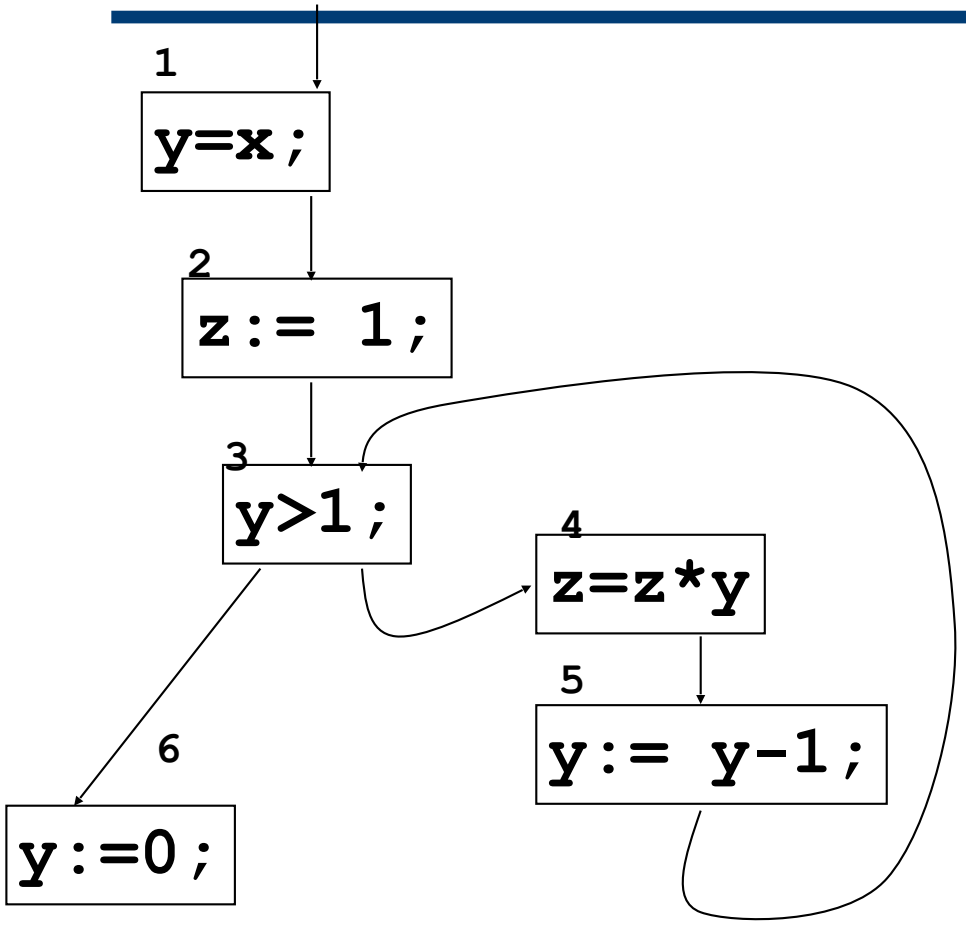
- Based on the CFG
- Constructs an equation system in terms of the property at the entry and exit of a node in the CFG
- Looks for solution of the system of equations as a fix point
- Compute either a least fixpont or a greatest fix point depending on the direction of the approximation

An Example Reaching Definitions

$$[y := x]^1; [z := 1]^2; while [y > 1]^3 do [z = z * y]^4; [y := y - 1]^5 od; [y := 0]^6$$



An example : RD



	Kill	Gen
1	(y,?),(y,1)(y,5)(y,6)	(y,1)
2	(z,?)(z,2)(z,4)	(z,2)
4	(z,?)(z,2)(z,4)	(z,4)
5	(y,?),(y,1)(y,5)(y,6)	(y,5)
6	(y,?),(y,1)(y,5)(y,6)	(y,6)

Second iteration fix point!

$RD_{entry}(1)= \{(x,?)(y,?)(z,?)\}$

$RD_{exit}(1) = \{(x,?)(y,1)(z,?)\}$

$RD_{entry}(2)= \{(x,?)(y,1)(z,?)\}$

$RD_{exit}(2)= \{(x,?)(y,1)(z,2)\}$

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

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

$RD_{entry}(4)= \{(x,?),(y,1)(z,2)(z,4)\ (y,5)\}$

$RD_{exit}(4)= \{(x,?),(y,1)(z,4)\ (y,5)\}$

$RD_{entry}(5)= \{(x,?),(y,1)(z,4)\ (y,5)\}$

$RD_{exit}(5)= \{(x,?),(y,5)(z,4)\}$

$RD_{entry}(6)= \{(x,?),(y,1)(z,2)\ (z,4)(y,5)\}$

$RD_{exit}(6)= \{(x,?),(y,6)(z,2)\ \ (z,4)\}$

$RD_{entry}(p) = \{(x,?) |\ x \text{ in } Vars\}$, if p is initial

$RD_{entry}(p) = \cup\{RD_{exit}(q) \mid q \text{ in } pre[p]\}$, otherwise

$RD_{exit}(p) = (RD_{entry}(p) \setminus kill_{RD}[p]) \cup gen_{RD}[p]$

Control Flow Analysis

- Constructs a constraint system
  - Looks for a solution of the constraint system as a fix point

Control Flow Analysis

- An alternative to equational approach is the constraint based approach.
- The idea is to extract a number of inclusions (equation or constraints) out of the program
- Once again we consider the property at the entry and exit of a node
- We encode with constraints the information of the flow of the CFG

Constraints for effects of elementary blocks

$$[y = 1]^1; [z = 1]^2; \text{ while } [y > 1]^3 \text{ do } ([z = z * y]^4; [y = y - 1]^5); [y = 0]^6$$

- We have constraints that express the effects for elementary blocks

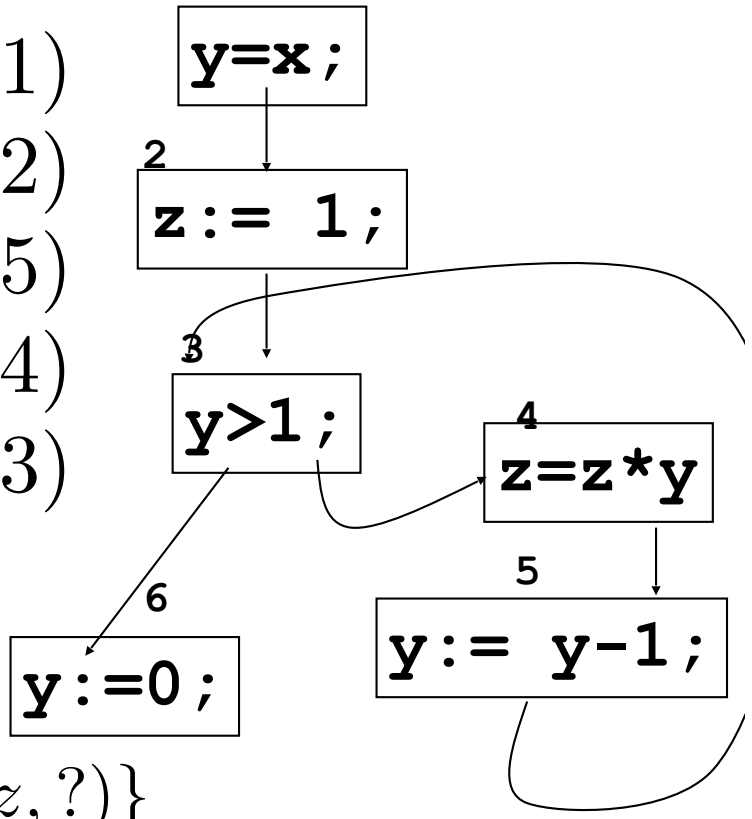
$$RD_{exit}(1) \supseteq RD_{entry}(1)/\{(y, l) \text{ such that } l \in \mathbf{Lab}\}$$
$$RD_{exit}(1) \supseteq \{(y, 1)\}$$
$$RD_{exit}(2) \supseteq RD_{entry}(2)/\{(z, l) \text{ such that } l \in \mathbf{Lab}\}$$
$$RD_{exit}(2) \supseteq \{(z, 2)\}$$
$$RD_{exit}(3) \supseteq RD_{entry}(3)$$
$$RD_{exit}(4) \supseteq RD_{entry}(4)/\{(z, l) \text{ such that } l \in \mathbf{Lab}\}$$
$$RD_{exit}(4) \supseteq \{(z, 4)\}$$
$$RD_{exit}(5) \supseteq RD_{entry}(5)/\{(y, l) \text{ such that } l \in \mathbf{Lab}\}$$
$$RD_{exit}(5) \supseteq \{(y, 5)\}$$
$$RD_{exit}(6) \supseteq RD_{entry}(6)/\{(y, l) \text{ such that } l \in \mathbf{Lab}\}$$
$$RD_{exit}(6) \supseteq \{(y, 6)\}$$

More constraints

- We have constraints that expression control may flow through the program

$$[y = 1]^1; [z = 1]^2; \text{while } [y > 1]^3 \text{ do } ([z = z * y]^4; [y = y - 1]^5); [y = 0]^6$$

$$\begin{aligned}
RD_{entry}(2) &\supseteq RD_{exit}(1) \\
RD_{entry}(3) &\supseteq RD_{exit}(2) \\
RD_{entry}(3) &\supseteq RD_{exit}(5) \\
RD_{entry}(5) &\supseteq RD_{exit}(4) \\
RD_{entry}(6) &\supseteq RD_{exit}(3)
\end{aligned}$$



Finally the constraint

$$RD_{entry}(1) \supseteq \{(x, ?), (y, ?), (z, ?)\}$$

Summary of the constraints system

$$RD_{exit}(1) \supseteq RD_{entry}(1)/\{(y,l) \text{ such that } l \in \mathbf{Lab}\}$$
$$RD_{exit}(1) \supseteq \{(y,1)\}$$
$$RD_{exit}(2) \supseteq RD_{entry}(2)/\{(z,l) \text{ such that } l \in \mathbf{Lab}\}$$
$$RD_{exit}(2) \supseteq \{(z,2)\}$$
$$RD_{exit}(3) \supseteq RD_{entry}(3)$$
$$RD_{exit}(4) \supseteq RD_{entry}(4)/\{(z,l) \text{ such that } l \in \mathbf{Lab}\}$$
$$RD_{exit}(4) \supseteq \{(z,4)\}$$
$$RD_{exit}(5) \supseteq RD_{entry}(5)/\{(y,l) \text{ such that } l \in \mathbf{Lab}\}$$
$$RD_{exit}(5) \supseteq \{(y,5)\}$$
$$RD_{exit}(6) \supseteq RD_{entry}(6)/\{(y,l) \text{ such that } l \in \mathbf{Lab}\}$$
$$RD_{exit}(6) \supseteq \{(y,6)\}$$
$$RD_{entry}(2) \supseteq RD_{exit}(1)$$
$$RD_{entry}(3) \supseteq RD_{exit}(2)$$
$$RD_{entry}(3) \supseteq RD_{exit}(5)$$
$$RD_{entry}(5) \supseteq RD_{exit}(4)$$
$$RD_{entry}(6) \supseteq RD_{exit}(3)$$
$$RD_{entry}(1) \supseteq \{(x,?),(y,?),(z,?)\}$$

12 sets

17 constraints

We look for $\overleftarrow{RD}$

$$\overleftarrow{RD} \sqsupseteq F(\overleftarrow{RD})$$

the solution is a prefix point

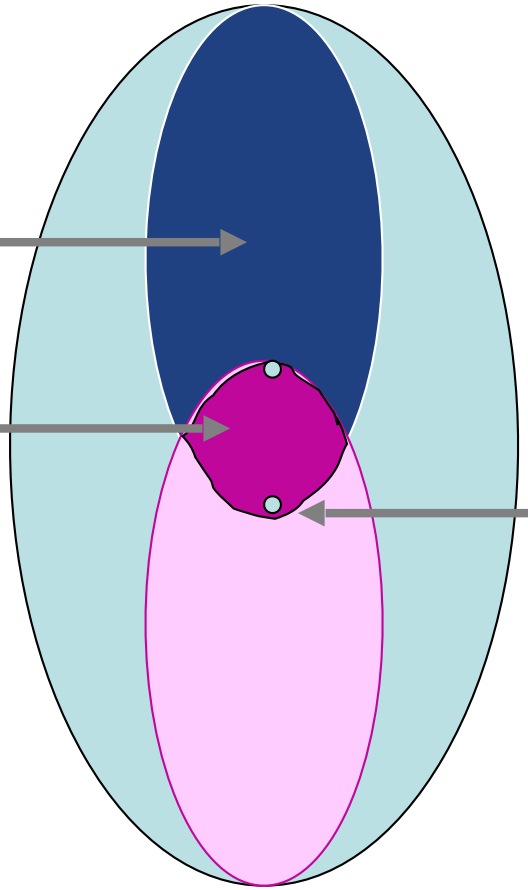
Remember…Fixpoints on Complete Lattices

Prefixpoints

$\{ l \in L \mid f(l) \leq_P l \}$

$Fix(f) = \{ l \in L \mid f(l) = l \}$

$lfp(f) = glb\{ l \in L \mid f(l) \leq l \}$



The solution can be computed as a least fix point!

The same solution!

$RD_{entry}(1)= \{(x,?)(y,?)(z,?)\}$

$RD_{exit}(1) = \{(x,?)(y,1)(z,?)\}$

$RD_{entry}(2)= \{(x,?)(y,1)(z,?)\}$

$RD_{exit}(2)= \{(x,?)(y,1)(z,2)\}$

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

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

$RD_{entry}(4)= \{(x,?),(y,1)(z,2)(z,4)\ (y,5)\}$

$RD_{exit}(4)= \{(x,?),(y,1)(z,4)(y,5)\}$

$RD_{entry}(5)= \{(x,?),(y,1)(z,4)(y,5)\}$

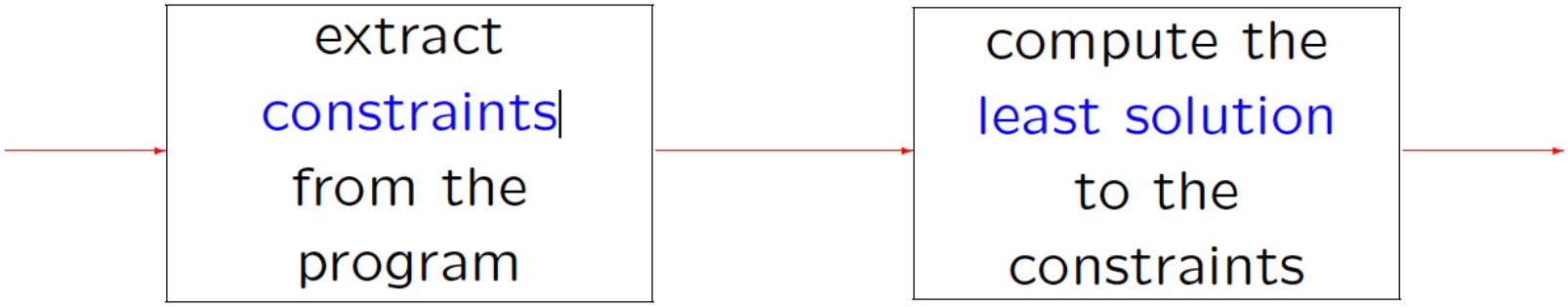
$RD_{exit}(5)= \{(x,?),(y,5)(z,4)\}$

$RD_{entry}(6)= \{(x,?),(y,1)(z,2)(z,4)(y,5)\}$

$RD_{exit}(6)= \{(x,?),(y,6)(z,2)\ (z,4)\}$

Constraint based analysis

How to automate the analysis



Type and Effect Systems

idea:

-annotated base types

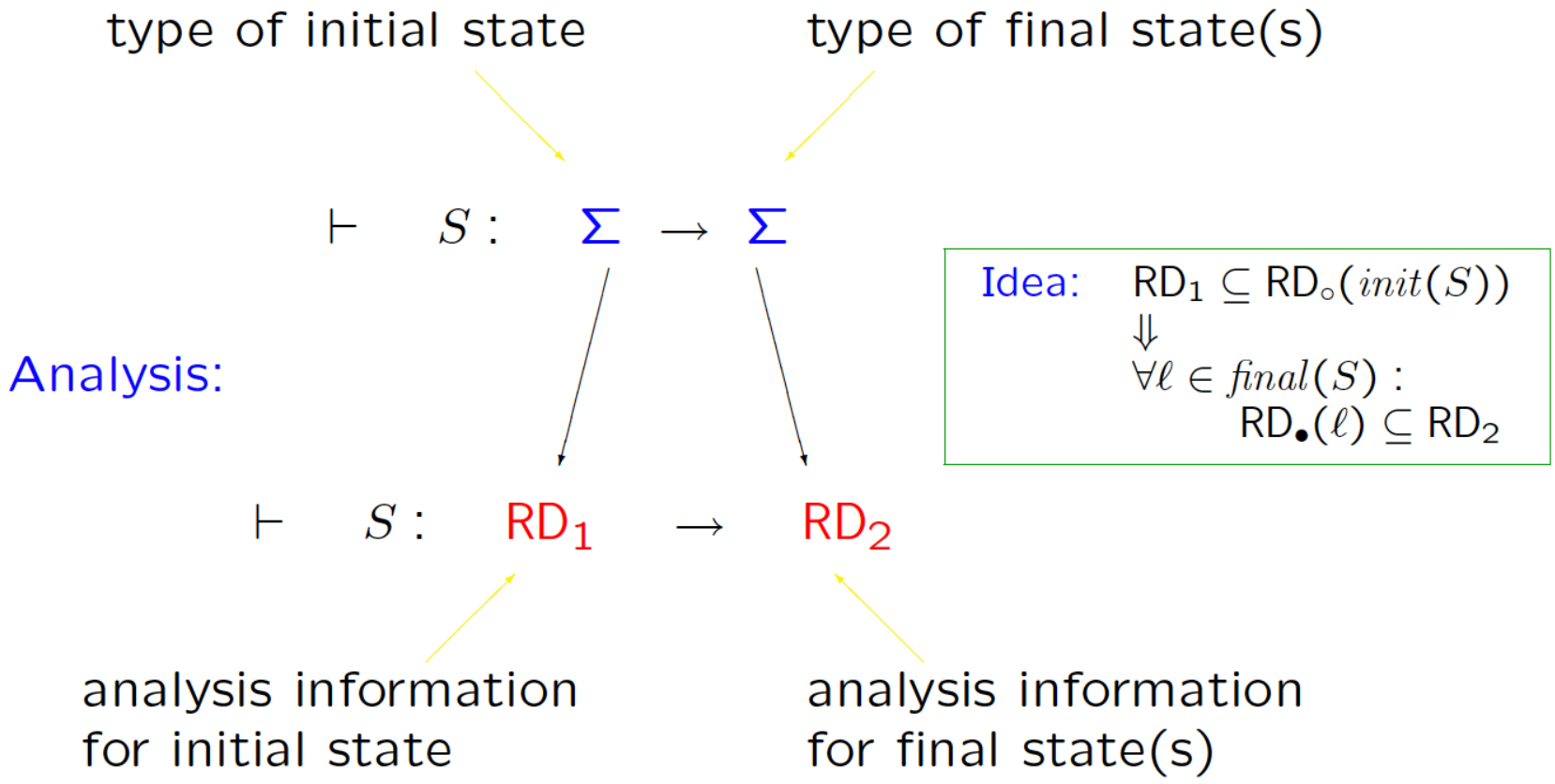
- annotated type constructors

• types:

Σ is the type of states;

all statements S have type $\Sigma \to \Sigma$ written $\vdash$ S : $\Sigma \to \Sigma$

Annotated data types



Annotated type system I

$$\vdash [x := a]^{\ell} : \underbrace{\mathsf{RD}}_{\text{before}} \rightarrow \underbrace{(\mathsf{RD} \setminus \{(x, \ell') \mid \ell' \in \mathbf{Lab}\}) \cup \{(x, \ell)\}}_{\text{after}}$$

$$\frac{\vdash S_1 : \mathsf{RD}_1 \rightarrow \mathsf{RD}_2 \qquad \vdash S_2 : \mathsf{RD}_2 \mid\!\rightarrow \mathsf{RD}_3}{\vdash S_1; S_2 : \underbrace{\mathsf{RD}_1}_{\text{before}} \rightarrow \underbrace{\mathsf{RD}_3}_{\text{after}}}$$

assumptions / conclusion

Implicit: the analysis information at the exit of S_1 equals the analysis information at the entry of S_2

Annotated type system II

$$\frac{\vdash S_1 : \mathsf{RD}_1 \to \mathsf{RD}_2 \qquad \vdash S_2 : \mathsf{RD}_1 \to \mathsf{RD}_2}{\vdash \texttt{if } [b]^\ell \texttt{ then } S_1 \texttt{ else } S_2 \texttt{ fi} : \mathsf{RD}_1 \to \mathsf{RD}_2}$$

Implicit: the two branches have the same analysis information at their respective entry and exit points

$$\frac{\vdash S : \mathsf{RD} \to \mathsf{RD}}{\vdash \texttt{while } [b]^\ell \texttt{ do } S \texttt{ od} : \mathsf{RD} \to \mathsf{RD}}$$

Implicit: the occurrences of RD express an invariance i.e. a fixed point property!

Annotated type system III

The subsumption rule:

$$\frac{\vdash S : \mathsf{RD}'_1 \to \mathsf{RD}'_2}{\vdash S : \mathsf{RD}_1 \to \mathsf{RD}_2} \quad \text{if } \mathsf{RD}_1 \subseteq \mathsf{RD}'_1 \text{ and } \mathsf{RD}'_2 \subseteq \mathsf{RD}_2$$

The rule is essential for the rules for conditional and iteration to work

- $\mathsf{RD}_1 \subseteq \mathsf{RD}'_1$: strengthen the analysis information for the initial state
- $\mathsf{RD}'_2 \subseteq \mathsf{RD}_2$: weaken the analysis information for the final states

We want to prove that

$$\frac{\vdash S_1 : \mathsf{RD}_1 \to \mathsf{RD}_2 \qquad \vdash S_2 : \mathsf{RD}_2 \mapsto \mathsf{RD}_3}{\vdash S_1; S_2 : \mathsf{RD}_1 \to \mathsf{RD}_3}$$

$$\{(x,?),(y,?),(z,?)\} \to \{(x,?),(y,6),(z,2),(z,4)\}$$

$\vdash [z:=1]^2$; while $[y>1]^3$ do $[z:=z*y]^4$; $[y:=y-1]^5$ od; $[y=0]^6$

$\{(x,?),(y,1),(z,?)\} \to \{(x,?),(y,6),(z,2),(z,4)\}$

$\vdash [y:=x]^1 : \{(x,?),(y,?),(z,?)\} \to \{(x,?),(y,1),(z,?)\}$

SEQ

$\vdash [y:=x]^1$; $[z:=1]^2$; while $[y>1]^3$ od $[z:=z*y]^4$; $[y:=y-1]^5$ od; $[y:=0]^6$:

$\{(x,?),(y,?),(z,?)\} \to \{(x,?),(y,6),(z,2),(z,4)\}$

We now need to prove that

$$\frac{\vdash S_1 : \mathsf{RD}_1 \to \mathsf{RD}_2 \qquad \vdash S_2 : \mathsf{RD}_2 \mapsto \mathsf{RD}_3}{\vdash S_1; S_2 : \mathsf{RD}_1 \to \mathsf{RD}_3}$$

$$[z := 1]^2; while\,[y := 1]^3\, do\, [z = z * y]^4; [y := y - 1]^5\, od; [y := 0]^6 :$$

$$\{(x, ?), (y, 1), (z, ?)\} \to \{(x, ?), (y, 6), (z, 2), (z, 4)\}$$

$\vdash$ while $[y>1]^3$ do $[z:=z*y]^4$; $[y:=y-1]$ od; $[y:=0]^6$:

$\{(x,?),(y,1),(z,2)\} \to \{(x,?),(y,6)(z,2),(z,4)\}$

$\vdash [z:=1]^2 : \{(x,?),(y,1),(z,?)\} \to \{(x,?),(y,1),(z,2)\}$

SEQ

$\vdash [z:=1]^2$; while $[y>1]^3$ do $[z:=z*y]^4$; $[y:=y-1]^5$ od; $[y:=0]^6$:

$\{(x,?),(y,1),(z,?)\} \to \{(x,?),(y,6),(z,2),(z,4)\}$

We now need to prove that

$$while[y := 1]^3 do[z = z * y]^4; [y := y - 1]^5 od; [y := 0]^6 :$$

$$\{(x, ?), (y, 1), (z, 2)\} \rightarrow \{(x, ?), (y, 6), (z, 2), (z, 4)\}$$

$$\frac{\vdash S_1 : RD_1 \rightarrow RD_2 \quad \vdash S_2 : RD_2 \mapsto RD_3}{\vdash S_1; S_2 : RD_1 \rightarrow RD_3}$$

$$\frac{\vdash S : RD'_1 \rightarrow RD'_2}{\vdash S : RD_1 \rightarrow RD_2} \quad \text{if } RD_1 \subseteq RD'_1 \text{ and } RD'_2 \subseteq RD_2$$

$\vdash [y:=0]^6$

RD → {(x,?),(y,6),(z,2),(z,4)}

$\vdash$ while $[y>1]^3$ do $[z:=z*y]^4; [y:=y-1]^5$ od; $[y:=0]^6$: RD → RD

SUB

while $[y>1]^3$ do $[z:=z*y]^4; [y:=y-1]^5$ od; $[y:=0]^6$:

{(x,?),(y,1),(z,2)} → {(x,?),(y,1),(y,5),(z,2),(z,4)} = RD

while $[y>1]^3$ do $[z:=z*y]^4; [y:=y-1]^5$ od; $[y:=0]^6$:

{(x,?),(y,1),(z,2)} → {(x,?),(y,6),(z,2),(z,4)}

$$\frac{\vdash S : \mathsf{RD} \to \mathsf{RD}}{\vdash \texttt{while}\ [b]^{\ell}\ \texttt{do}\ S\ \texttt{od} : \mathsf{RD} \to \mathsf{RD}}$$

We are left to prove that

$$while[y := 1]^3 do[z = z * y]^4; [y := y - 1]^5 od : RD \to RD$$

Abbreviation: $\mathsf{RD} = \{(x, ?), (y, 1), (y, 5), (z, 2), (z, 4)\}$

$$\frac{\dfrac{\dfrac{\vdash [z:=z*y]^4 : \mathsf{RD} \to \{(x,?),(y,1),(y,5),(z,4)\}}{\vdash [y:=y-1]^5 : \{(x,?),(y,1),(y,5),(z,4)\} \to \{(x,?),(y,5),(z,4)\}}}{\vdash [z:=z*y]^4;\ [y:=y-1]^5 : \mathsf{RD} \to \{(x,?),(y,5),(z,4)\}}}{\vdash [z:=z*y]^4;\ [y:=y-1]^5 : \mathsf{RD} \to \mathsf{RD} \quad \text{using } \{(x,?),(y,5),(z,4)\} \subseteq \mathsf{RD}}$$

$$\frac{}{\vdash \texttt{while}\ [y>1]^3\ \texttt{do}\ [z:=z*y]^4;\ [y:=y-1]^5\ \texttt{od} : \mathsf{RD} \to \mathsf{RD}}$$

How to automate the analysis