Abstract Interpretation

Lecture (1)

Sriram Rajamani
Microsoft Research

Two approaches to analysing a program

Testing

- Exercise some behaviours
- “Underapproximation”
- Can miss errors
- Unsound

Verification

- Exercise all behaviours (even infeasible ones)
- “overapproximation”
- Can generate false errors
- Incomplete

Abstract interpretation is a unified theory for verification of programs.
Proposed by Cousot-Cousot in a classic POPL 1977 paper.

Concrete vs abstract interpretation

- Concrete interpretation of a program is how we normally imagine how a program executes
  - We give it inputs, it runs and produces an output

- Abstract interpretation models "all possible" execution over "all possible inputs".
  - For this, we need do understand some special domains (which are sets with orderings) which are "semi-lattices"

Partially ordered sets (or Po-sets)

S is a po-set or a partially ordered set, if it has a binary relation $\leq$ which is:

- Reflexive: for all $x \in S$, $x \leq x$
- Antisymmetric: for all $x, y \in S$, $x \leq y \wedge y \leq x \Rightarrow x = y$
- Transitive: for all $x, y, z \in S$, $x \leq y \wedge y \leq z \Rightarrow x \leq z$

Lower bounds

Let $\langle S, \leq \rangle$ be a po-set

The lower bound of a set $A \subseteq S$ is an element ℓ such that

for all $a \in A$, $\ell \leq a$

Note1 : lower bound need not be unique

Note 2: if there is a lower bound $\ell \uparrow *$ such that for every lower bound ℓ of A we have that $\ell \leq \ell \uparrow *$, then such an $\ell \uparrow *$ is called a “greatest lower bound” or “GLB” of A

Upper bounds

Let $\langle S, \leq \rangle$ be a po-set

The upper bound of a set $A \subseteq S$ is an element u such that

for all $a \in A$, $a \leq u$

Note1 : upper bound need not be unique

Note 2: if there is a lower bound $u\uparrow *$ such that for every upper bound u of A we have that $u\uparrow * \leq u$, then such a $u\uparrow *$ is called a “least upper bound” or “LUB” of A

Lattice

Let $\langle S, \leq \rangle$ be a po-set

$\langle S, \leq \rangle$ is a lattice if every non-empty subset of elements in S has a GLB and LUB

Join Semi-Lattice

Let $\langle S, \leq \rangle$ be a po-set

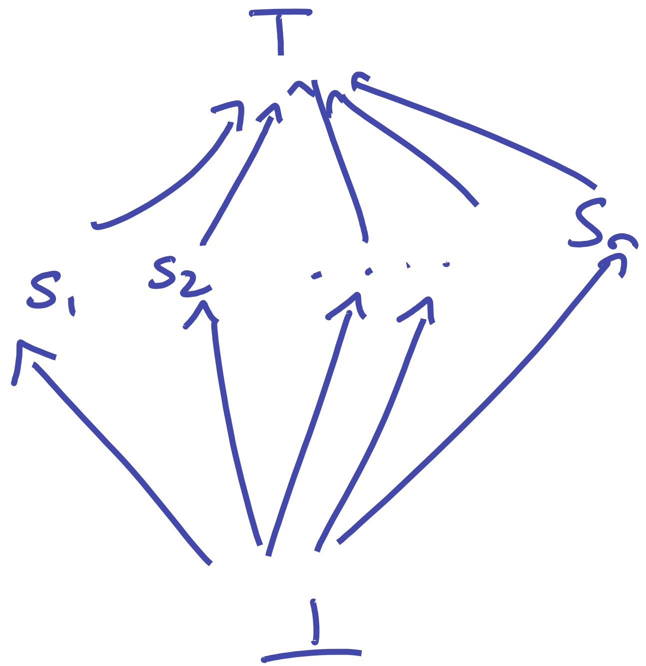
$\langle S, \leq \rangle$ is a join semi-lattice if every non-empty subset of elements in S has a LUB in S

Note: we can similarly define a meet semi-lattice, but we won't bother!

Sets & Lattices

Any set $S = \{s_1, s_2, \ldots s_n\}$ can be made into a lattice S°

S° :

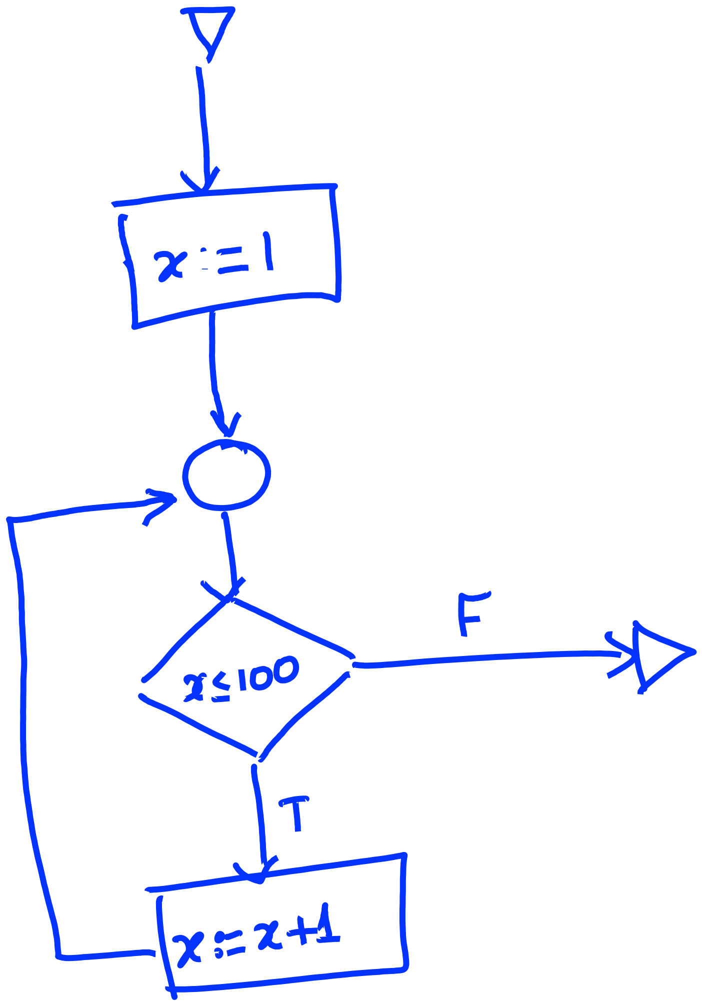


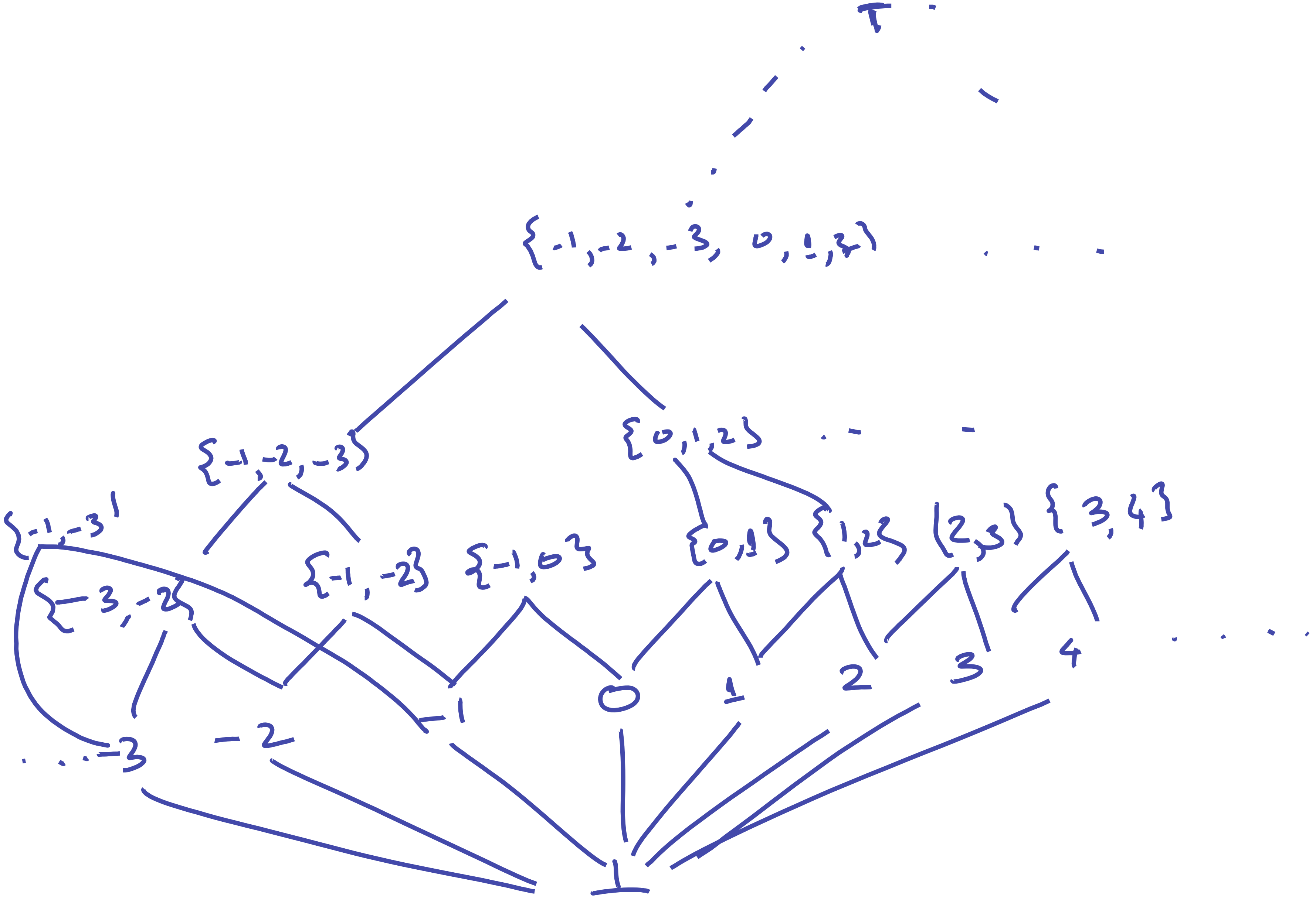
Why did we do all this semi-lattice stuff?

In order to do verification ☺

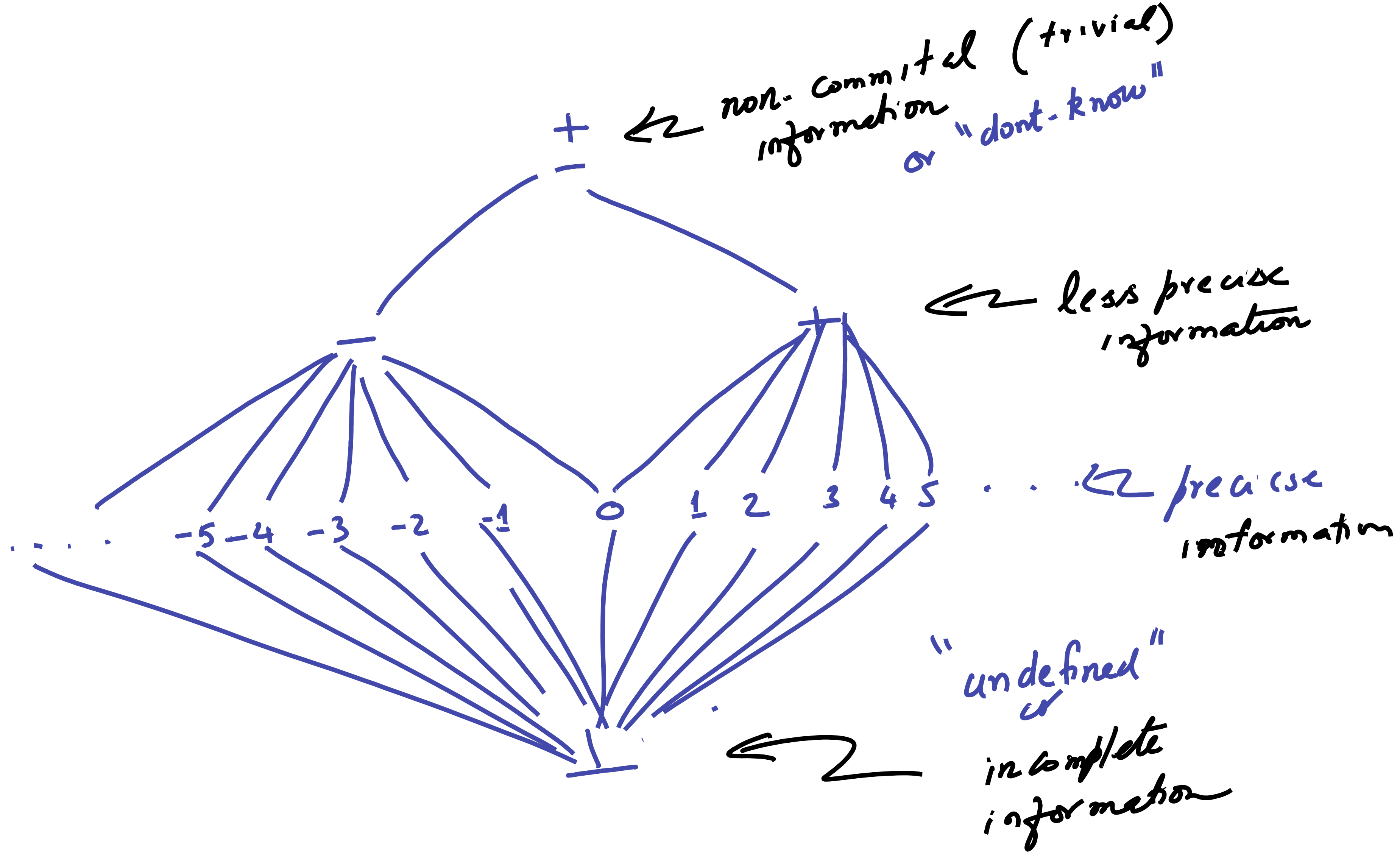
We can give meaning to a program (over all behaviours) by a fix-point computed over a semi-lattice!!!!

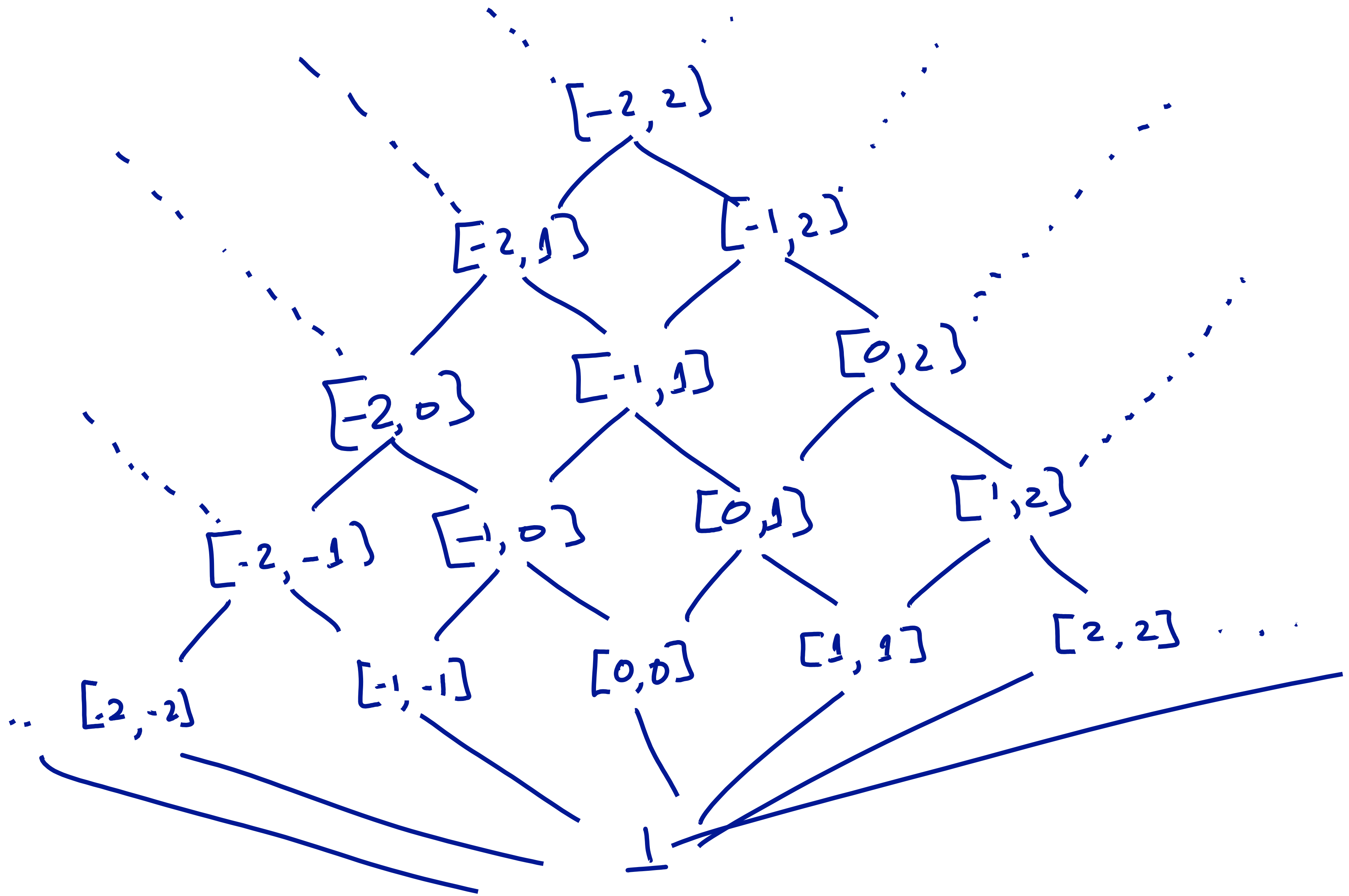
Example: we will use the program on the right as a case study to illustrate and explain abstract interpretation



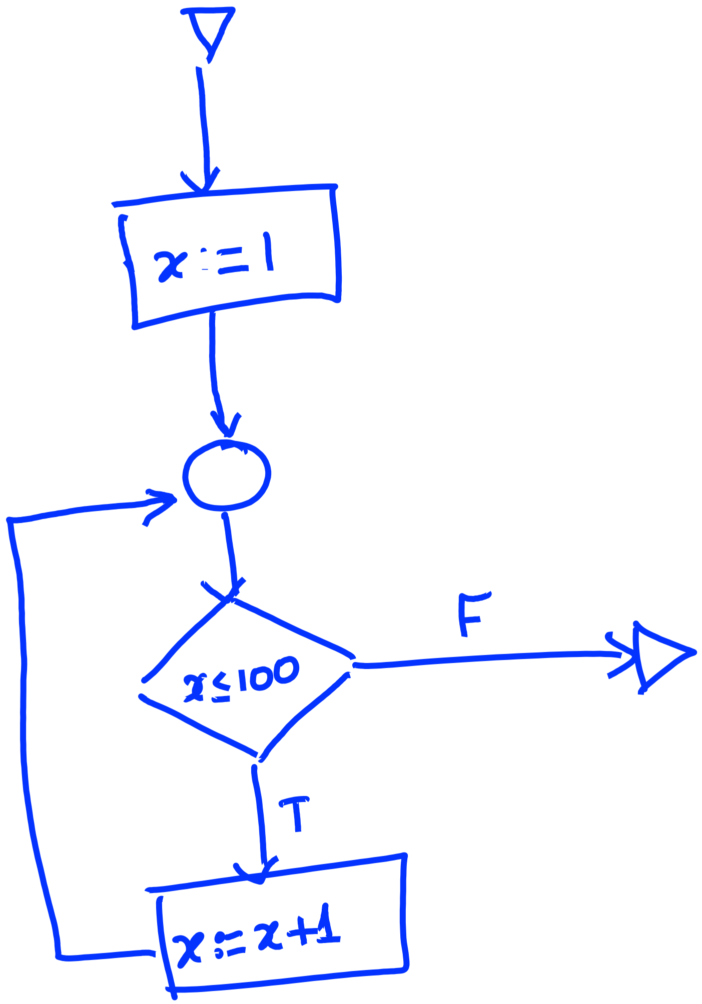


⊤
{-1, -2, -3, 0, 1, 2}
{-1, -2, -3}
{0, 1, 2}
{-1, -3}
{-3, -2}
{-1, -2}
{-1, 0}
{0, 1}
{1, 2}
{2, 3}
{3, 4}
-3
-2
-1
0
1
2
3
4
⊥

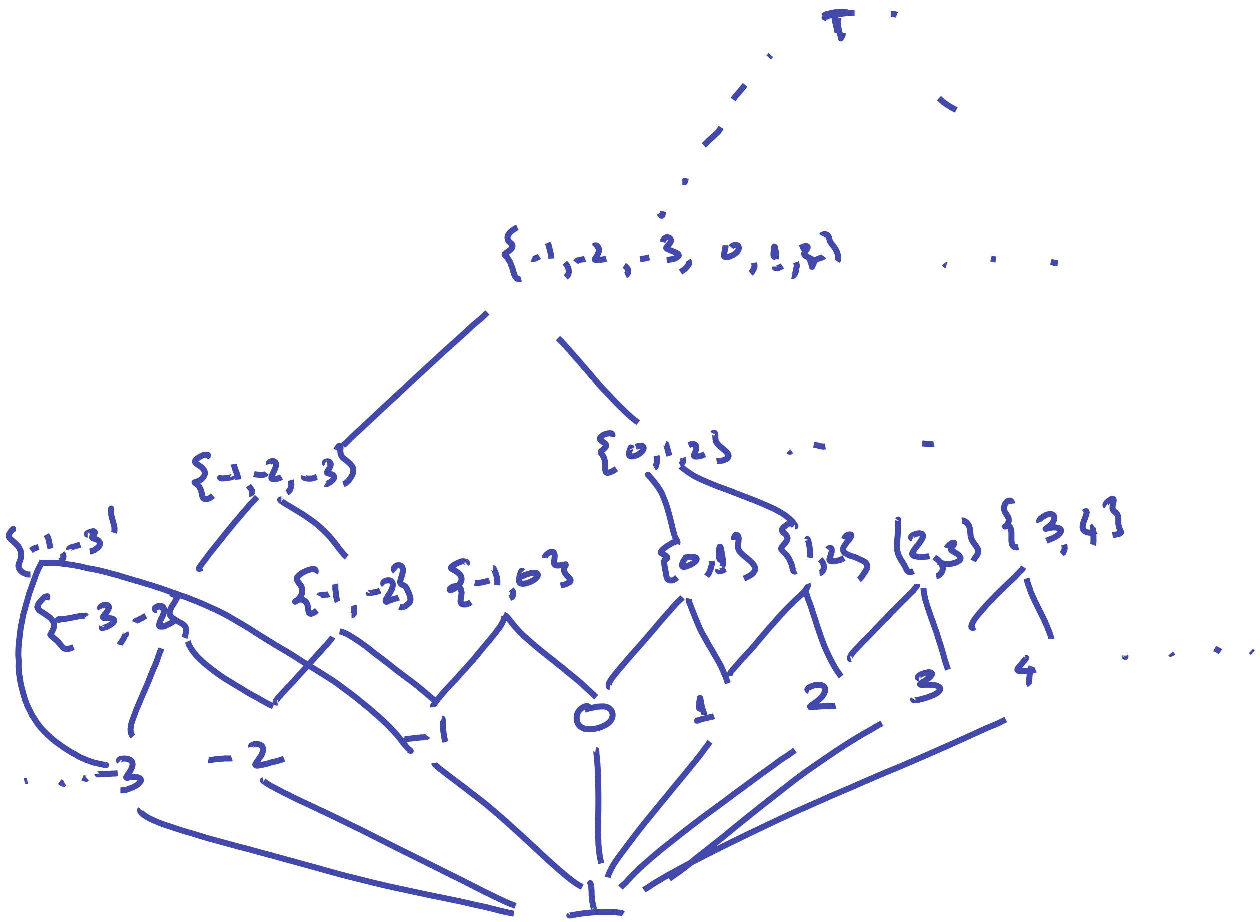

⊤
non-commited (trivial) information
or "dont-know"
⊤
less precise information
...
-5
-4
-3
-2
-1
0
1
2
3
4
5
...
precise information
"undefined"
or
incomplete information
⊥



[-2, 2]
[-2, 1]
[-1, 2]
[-2, 0]
[-1, 1]
[0, 2]
[-2, -1]
[-1, 0]
[0, 1]
[1, 2]
[-2, -2]
[-1, -1]
[0, 0]
[1, 1]
[2, 2]
⊥

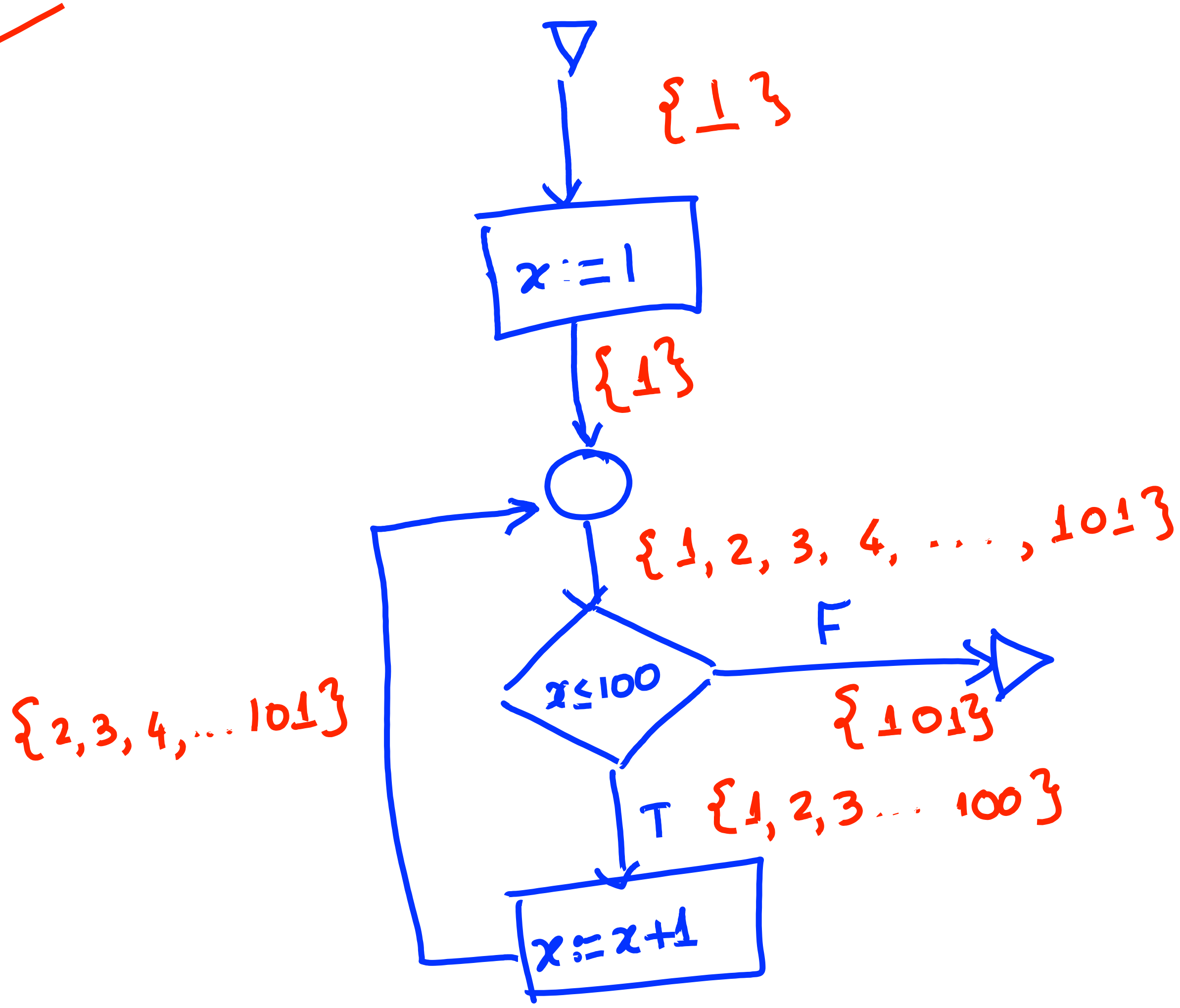


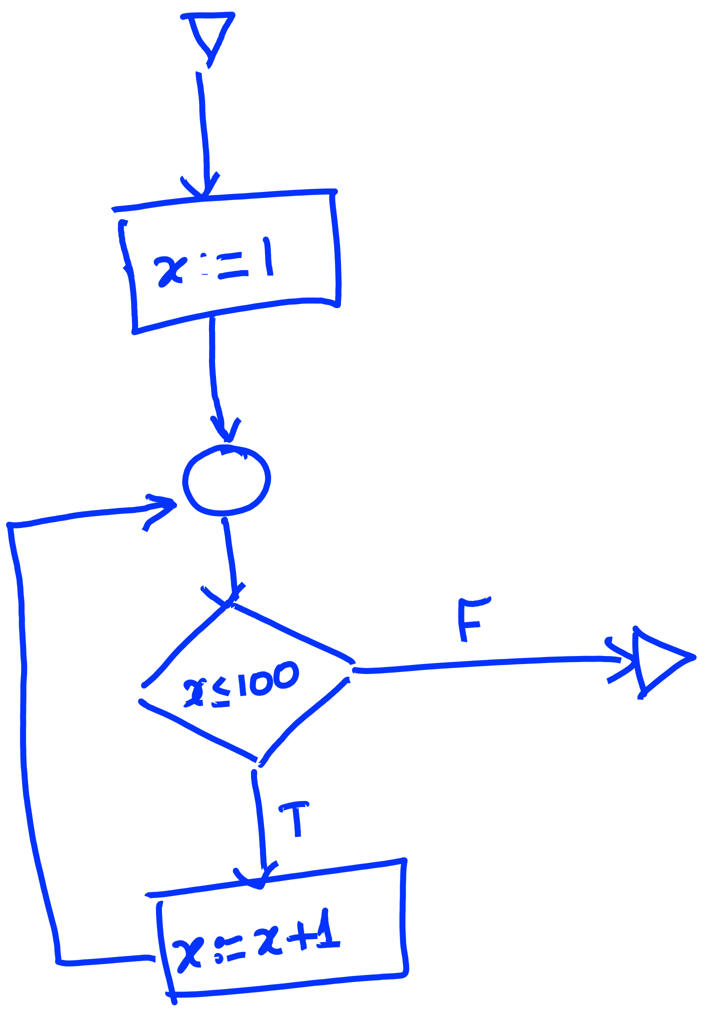
x := 1
x ≤ 100
F
T
x := x+1



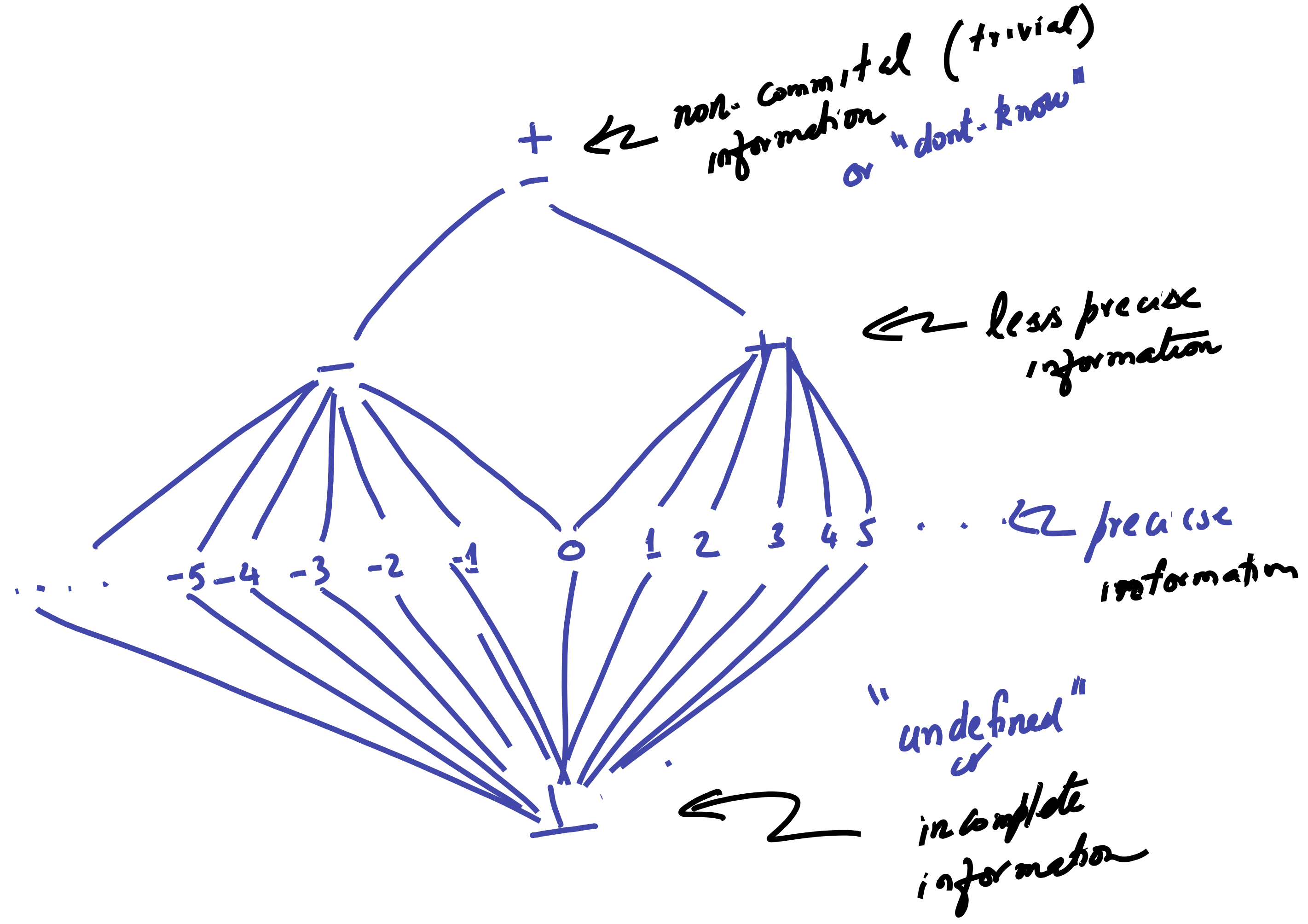
⊤
{-1,-2,-3, 0,1,2}
{-1,-2,-3}
{0,1,2}
{-1,-3}
{-3,-2}
{-1,-2}
{-1,0}
{0,1}
{1,2}
{2,3}
{3,4}
-3
-2
-1
0
1
2
3
4
⊥

Fixpoint



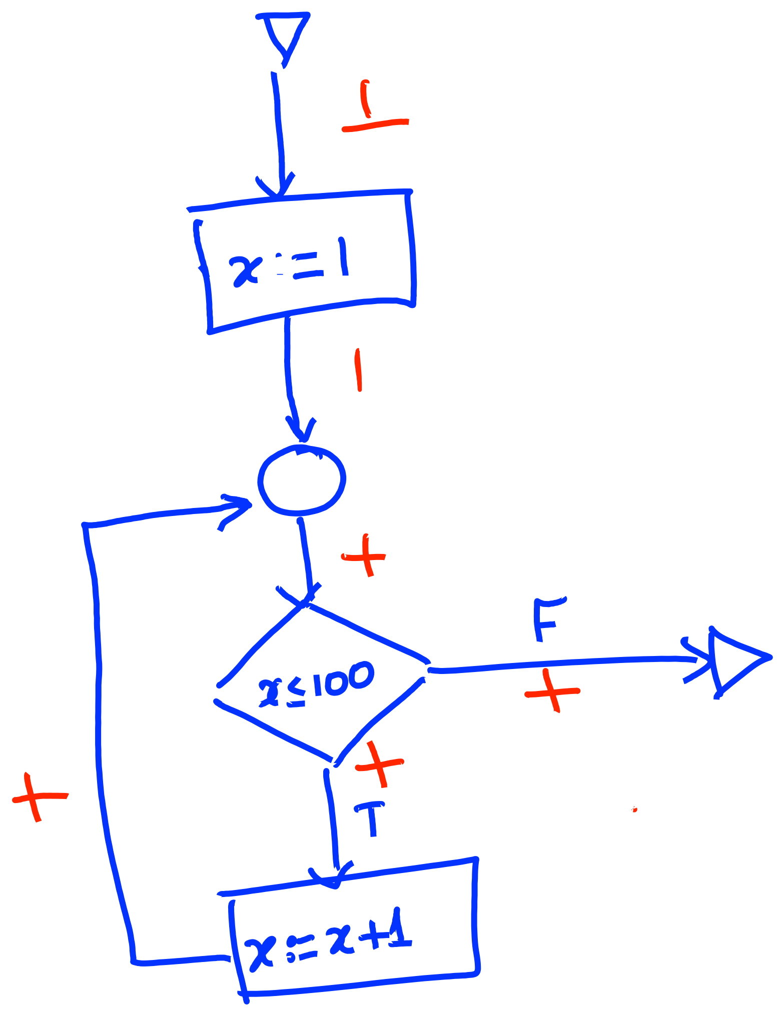


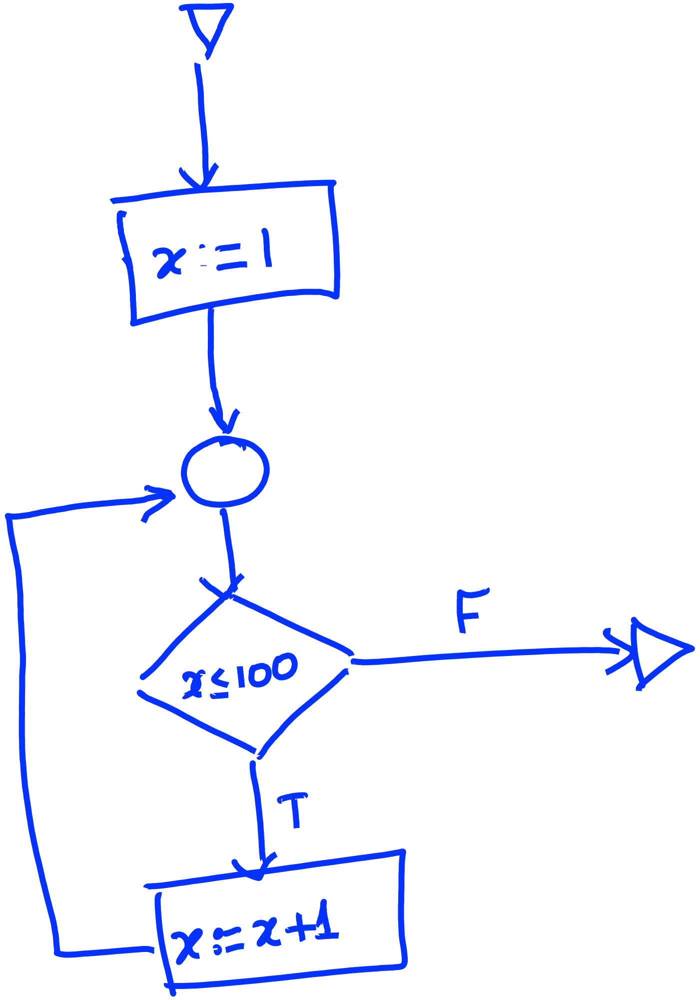
x := 1
x ≤ 100
F
T
x := x+1

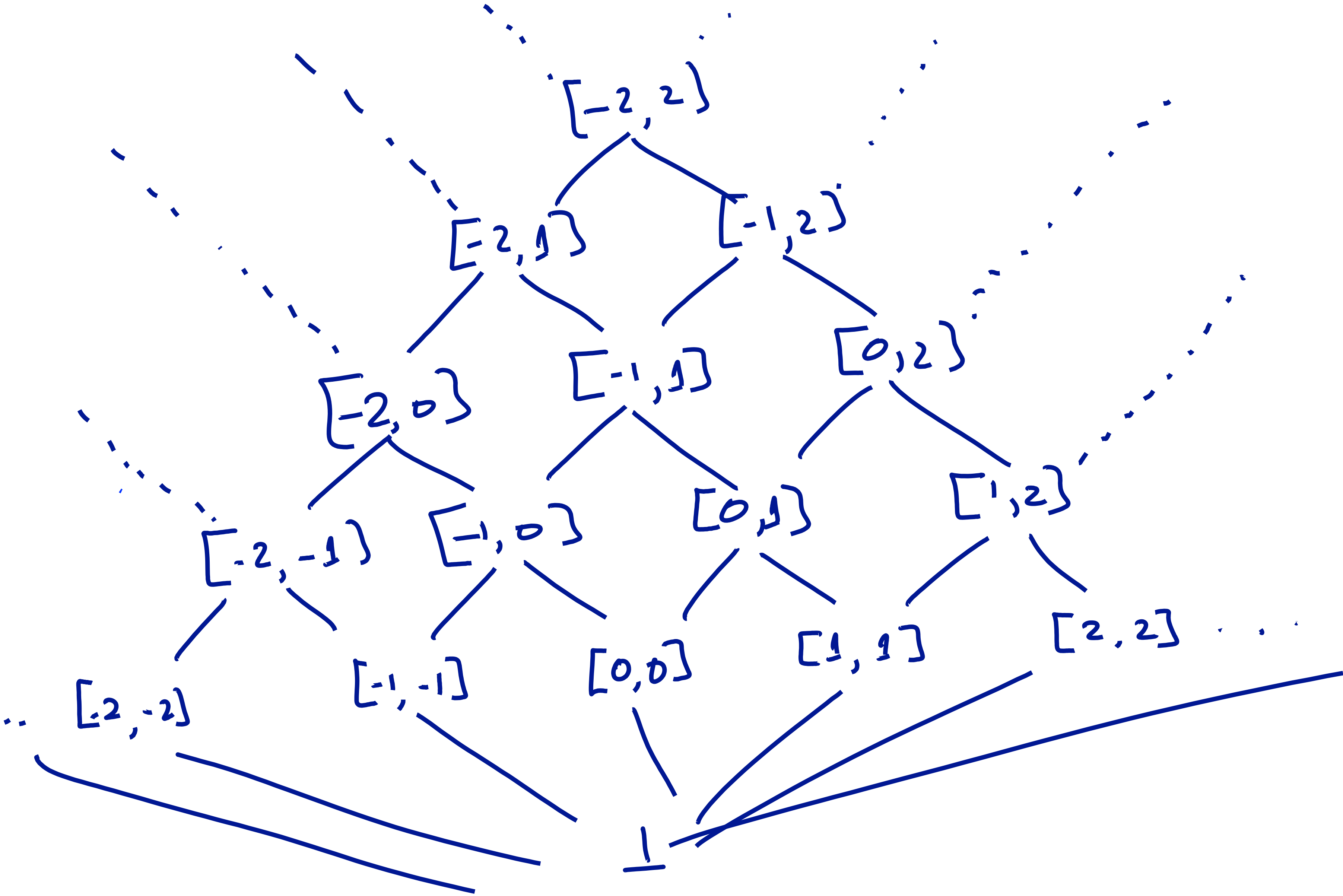


non-commited (trivial) information
or "dont-know"
less precise information
... -5 -4 -3 -2 -1 0 1 2 3 4 5 ...
precise information
"undefined" or incomplete information

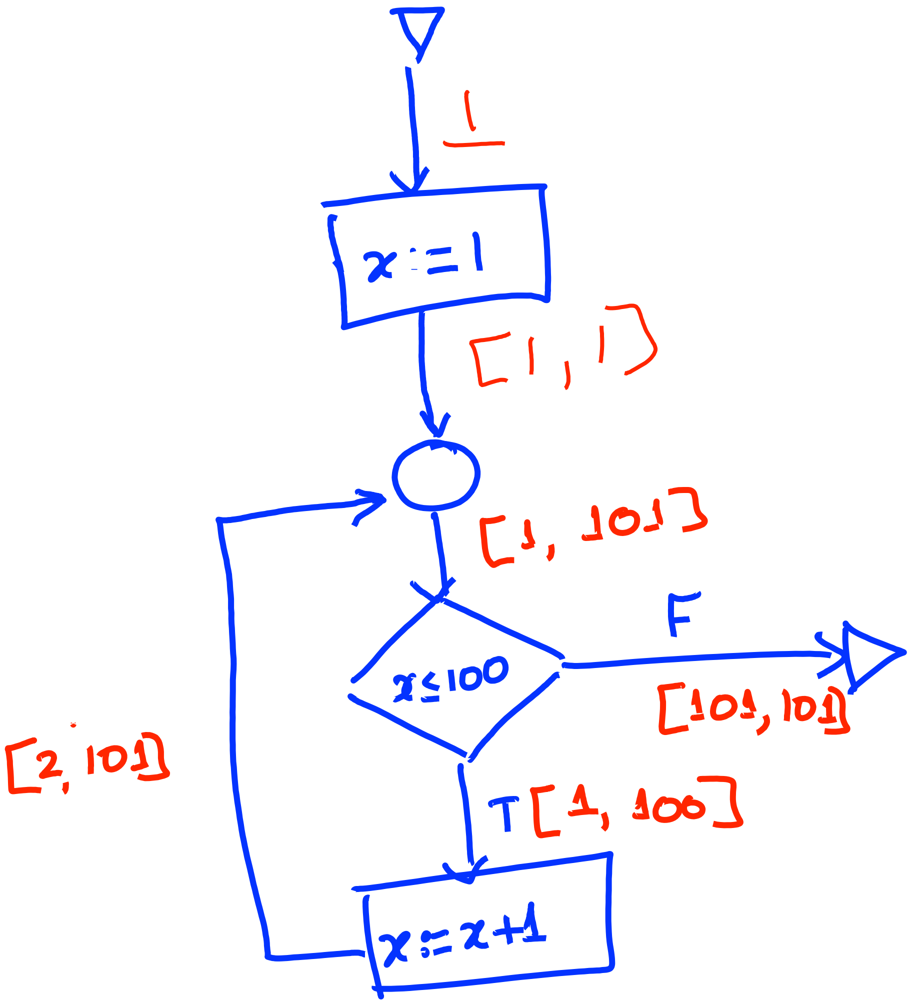
Fixpoint




x := 1
x ≤ 100
F
T
x := x+1


[-2, 2]
[-2, 1]
[-1, 2]
[-2, 0]
[-1, 1]
[0, 2]
[-2, -1]
[-1, 0]
[0, 1]
[1, 2]
[-2, -2]
[-1, -1]
[0, 0]
[1, 1]
[2, 2]
⊥

Fixpoint



So...

an abstract interpretation is really

$$\langle D,\ \circ,\ \leq,\ \top,\ \bot,\ I \rangle$$

domain

lub

p.o

Top

bottom

interpretation

$I : D \to D$

Science of Sound Abstract Interpretations

$$\langle D, \circ_D, \le_D, \top_D, \bot_D, \underline{\top}_D \rangle \underset{\gamma}{\overset{\alpha}{\rightleftarrows}} \langle A, \circ_A, \le_A, \top_A, \bot_A, \underline{\top}_A \rangle$$

eg:

Sets of Integers $\rightleftarrows$ Signs

Sets of Integers $\rightleftarrows$ Intervals

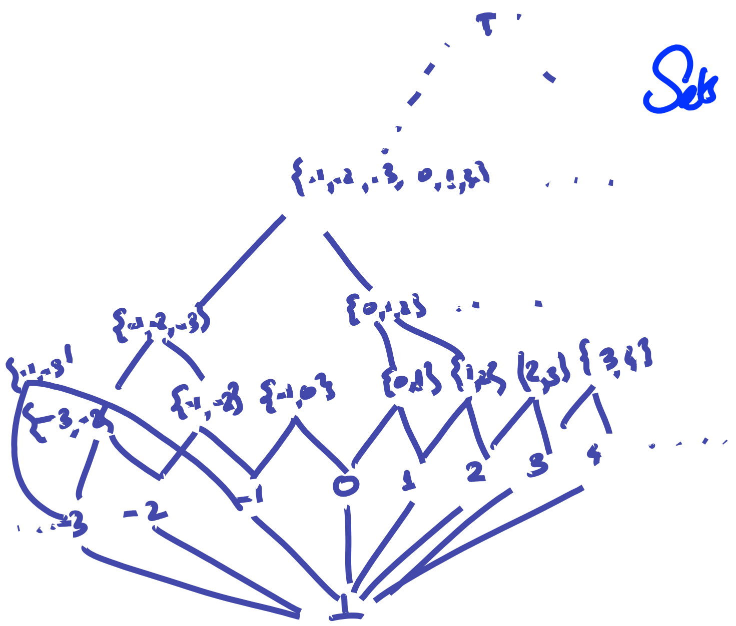
Abstract interpretations themselves form a lattice!

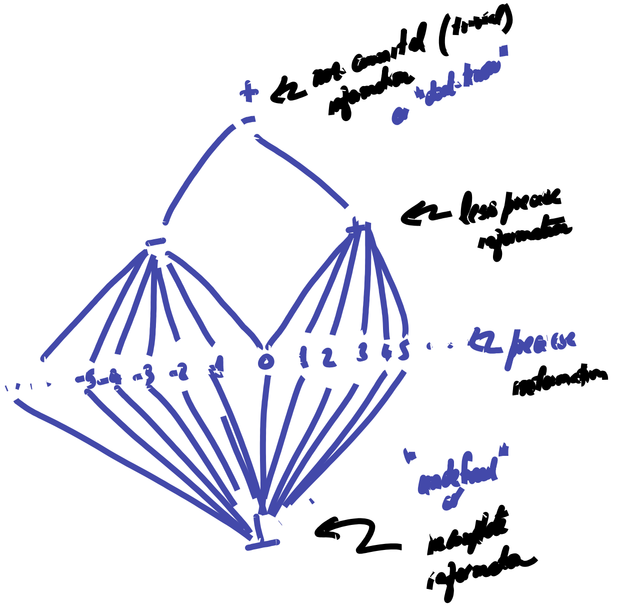
Science of Sound Abstract Interpretations

$$\langle D, \circ_D, \leq_D, \top_D, \perp_D, \bar{\top}_D \rangle \underset{\gamma}{\overset{\alpha}{\rightleftarrows}} \langle A, \circ_A, \leq_A, \top_A, \perp_A, \bar{\top}_A \rangle$$

eg: Sets of Integers $\rightleftarrows$ Signs

Sets of Integers $\rightleftarrows$ Intervals





Abstract interpretations themselves form a lattice!

Specifying an abstract interpretation

$$\langle D, \circ_D, \leq_D, \top_D, \bot_D, I_D \rangle \underset{\gamma}{\overset{\alpha}{\rightleftarrows}} \langle A, \circ_A, \leq_A, \top_A, \bot_A, I_A \rangle$$

$$\alpha : D \longrightarrow A$$

$$\gamma : A \longrightarrow D$$

α, γ form a Galois connection iff

1. α, γ are order preserving

   $\forall d_1, d_2 \in D \quad d_1 \leq_D d_2 \Rightarrow \alpha(d_1) \leq_A \alpha(d_2)$

   $\forall a_1, a_2 \in A \quad a_1 \leq_A a_2 \Rightarrow \gamma(a_1) \leq_D \gamma(a_2)$

2. $\forall d \in D . \; d \leq \gamma(\alpha(d))$

3. $\forall a \in A \quad a = \alpha(\gamma(a))$

From Galois connection to abstract state transition fn.

$$\langle D, \circ_D, \le_D, \top_D, \bot_D, I_D \rangle \underset{\gamma}{\overset{\alpha}{\rightleftarrows}} \langle A, \circ_A, \le_A, \top_A, \bot_A, I_A \rangle$$

Sp. $\langle \alpha, \gamma \rangle$ form a Galois connection

Can define $\underline{\underline{I_A}}$ in terms of I_D, α, γ.

$$I_A(a) = \alpha(I_D(\gamma(a)))$$

ie.. $I_A = \alpha \circ I_D \circ \gamma$

$$\langle D, \circ_D, \leq_D, T_D, \perp_D, I_D \rangle \underset{\gamma}{\overset{\alpha}{\rightleftarrows}} \langle A, \circ_A, \leq_A, T_A, \perp_A, I_A \rangle$$

$$I_A = \alpha \circ I_D \circ \gamma$$

$$\boxed{\underline{\text{Theorem:}} \quad Reach(I_D) \leq_D \gamma(Reach(I_A))}$$

Thus, any property proved on I_A carries over to I_D !!

Recipe for analysis:

Program's concrete interpretation : $\mathcal{C} = \langle D, \circ_D, \leq_D, \top_D, \bot_D, I_D \rangle$

Concrete semantics : Least Fix Point (I_D)

Difficulty : Least Fix Point (I_D) may be expensive to compute, or may not converge.

Solution: Come up with an abstract domain A and a Galois connection $D \underset{\gamma}{\overset{\alpha}{\rightleftarrows}} A$

Immediately get : $\mathcal{A} = \langle A, \circ_A, \leq_A, \top_A, \bot_A, I_A \rangle$ $I_A = \gamma \circ I_D \circ \alpha$

Abstract Semantics : Least Fix Point (I_A)

Hopefully, easier to compute!

Homework

- Review and understand these slides

- Start looking at the Cousot-Cousot 77 paper:

http://www.di.ens.fr/~cousot/COUSOTpapers/publications.www/CousotCousot-POPL-77-ACM-p238--252-1977.pdf

- Think about: under what circumstances does the "fixpoint computation" terminate? When might it not terminate? What could we do to make it always terminate?

End of Lecture 1