Program Analysis & Verification

Sriram K. Rajamani
MSR India
sriram@microsoft.com

Goal:

* In-depth introduction to theory & practice of software verification.
  - Small number of topics
  - in depth study

* Basic principles

* Appreciation for trade-offs

Plan:

- Today: "Abstract interpretation", Cousot - Cousot, POPL 77
- Various kinds of program analyses
  - flow insensitive Vs flow sensitive
  - context insensitive Vs context sensitive
- Predicate abstraction
- Automatic refinement of abstractions a/c SLAM

Duality

Testing
"run time analysis"

Some paths

under-approximation

unsound
missed errors

Vs

Verification
"static analysis"

all paths

over approximation

incomplete
false errors

Abstract interpretation is a unified theory for all paths analysis!

General approach

- Behaviors of program modelled as a semilattice
- Over-approx of program semantics $\triangleq$ fixpoint computed over the semilattice

Semilattice $\triangleq$ poset in which
* either every pair of elements has a supremum, or
* every pair of element has an infimum

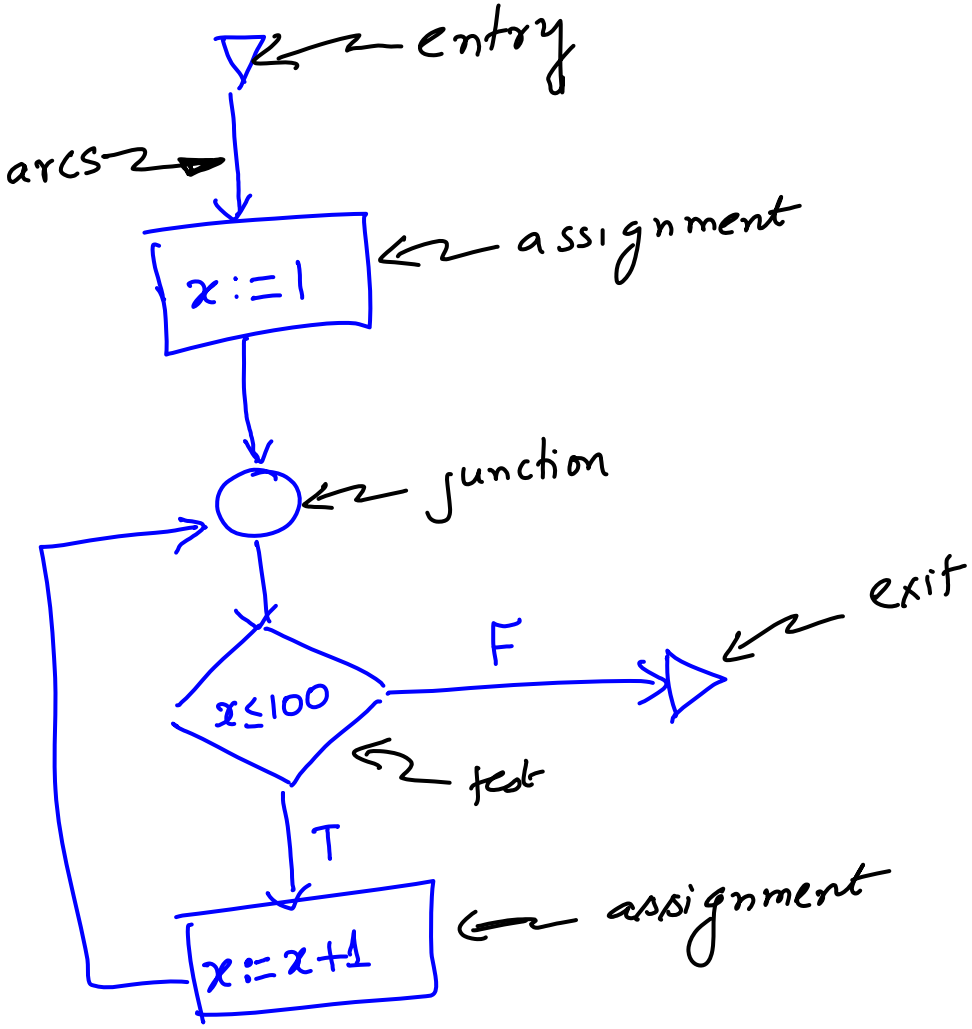
poset : set with a reflexive, anti-symmetric, transitive relation

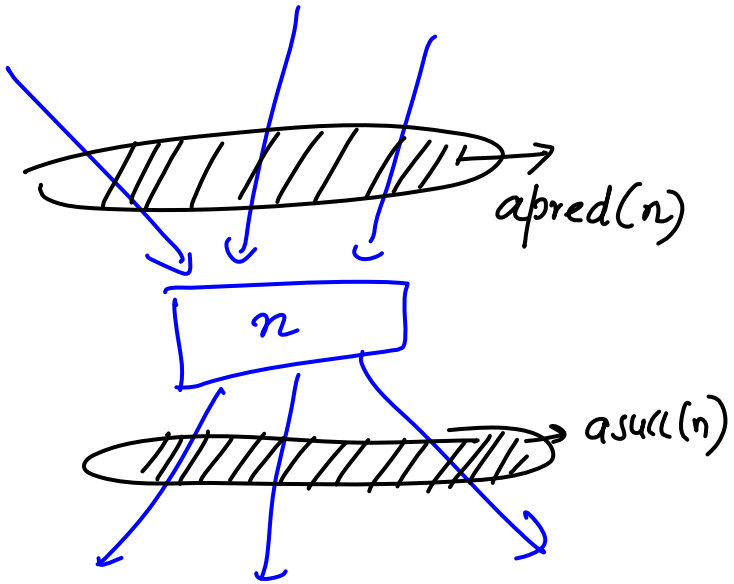
Program $P = \langle Nodes, n\text{-}succ, n\text{-}pred \rangle$

$n\text{-}succ : Nodes \rightarrow 2^{Nodes}$

$n\text{-}pred : Nodes \rightarrow 2^{Nodes}$

$Nodes =$ Entries $\oplus$ Assignments $\oplus$ Tests $\oplus$ Junctions $\oplus$ Exits


entry
arcs
x := 1
assignment
junction
exit
F
x≤100
test
T
x := x+1
assignment


apred(n)
n
asucc(n)

Concrete Semantics

What does a program mean?

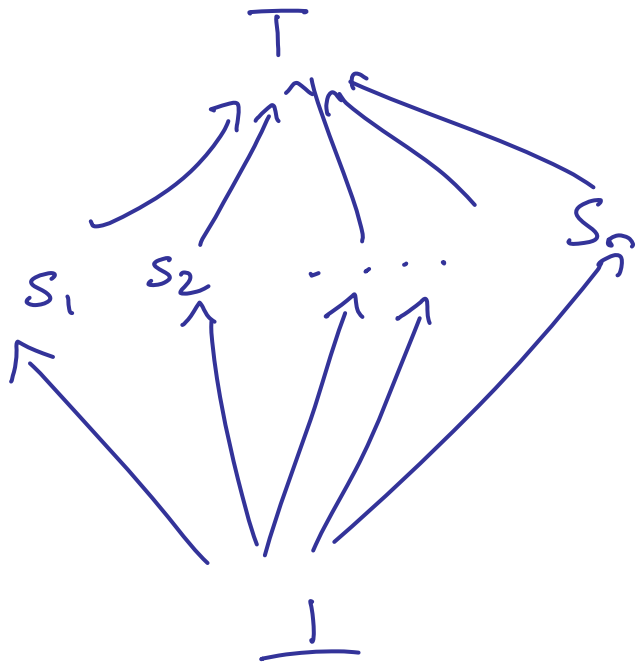
Approaches

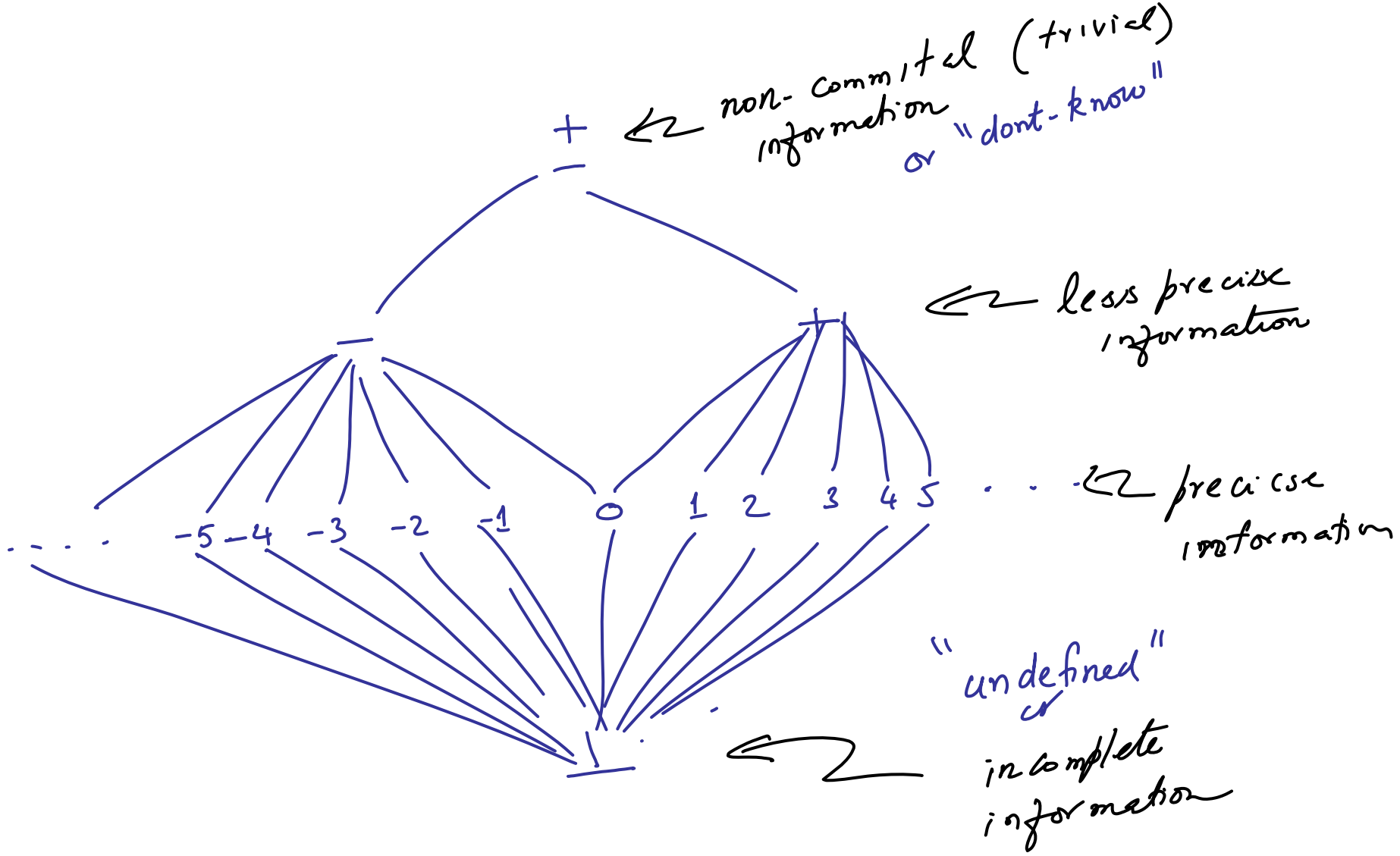
- denotational
  describe program as a mathematical function
- Operational
  Set of reachable states
- Axiomatic
  Annotate assertions and say when they are consistent with the program

Set & Lattices

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

S° :




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

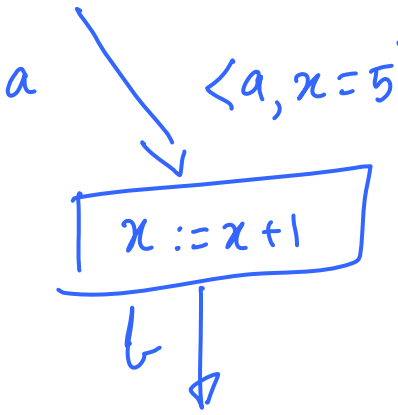
Denotational Semantics

$$States = Arcs \times Env$$

$$Env = Ident \to Values$$

$$\underline{val} : Expr \to [Env \to Values]$$

$$n\text{-}state : States \longrightarrow States$$



$$n\text{-}state(\langle a, x=5 \rangle) = \langle b, x=6 \rangle$$

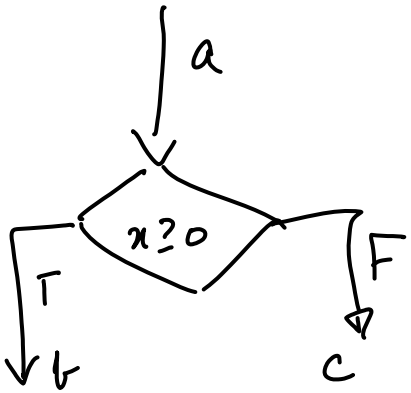
Trick: Each partial function f on set S is converted to a total function on S°

with:

$$f(\bot) = \bot,$$

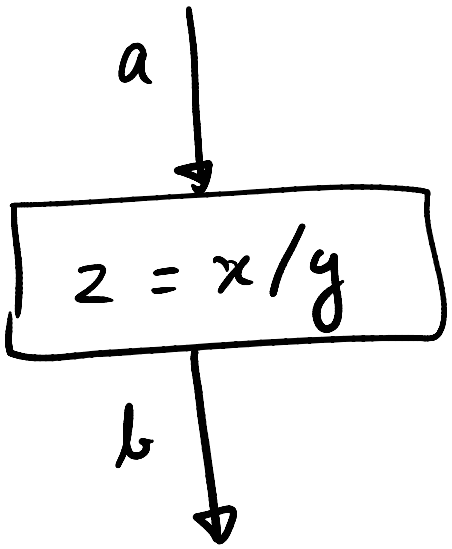
$$f(\top) = \top,$$

& $f(x) = \bot$ if x is undefined on x



n-state $(\langle a, x = \bot \rangle) =$

$\langle \bot, x = \bot \rangle$



n-state (<a, <x=5, y=0, z=35>>)

= <b, <x=5, y=0, z=1>>

Denotational Semantics

$$n\text{-}state^{\alpha} : States \to States$$

$$n\text{-}state^{\alpha} = LFP\cdot\left(\lambda F.\ (n\text{-}state \circ F)\right)$$

In general, this fix point may be un-computable, i.e., it may not converge but it exists, nevertheless, due to Knaster-Tarski fixpoint Theorem

Initial states

$$\underline{I}\text{-states} = \left\{ \langle a\text{-succ}(m),\ \underline{\perp}_{env} \rangle \;\middle|\; m \in Entries \right\}$$

$$\forall x \in Ident \left(Env(x) = \underline{\perp}_{values} \right)$$

Reachability

For $a, b \in States$

$$a \rightsquigarrow b \triangleq \exists n.\ b = n\text{-state}^{n}(a)$$

Reachable states

$$Reachable\ states = \{ s \mid \exists i \in \underline{I}\text{-states}.\ i \rightsquigarrow s \}$$

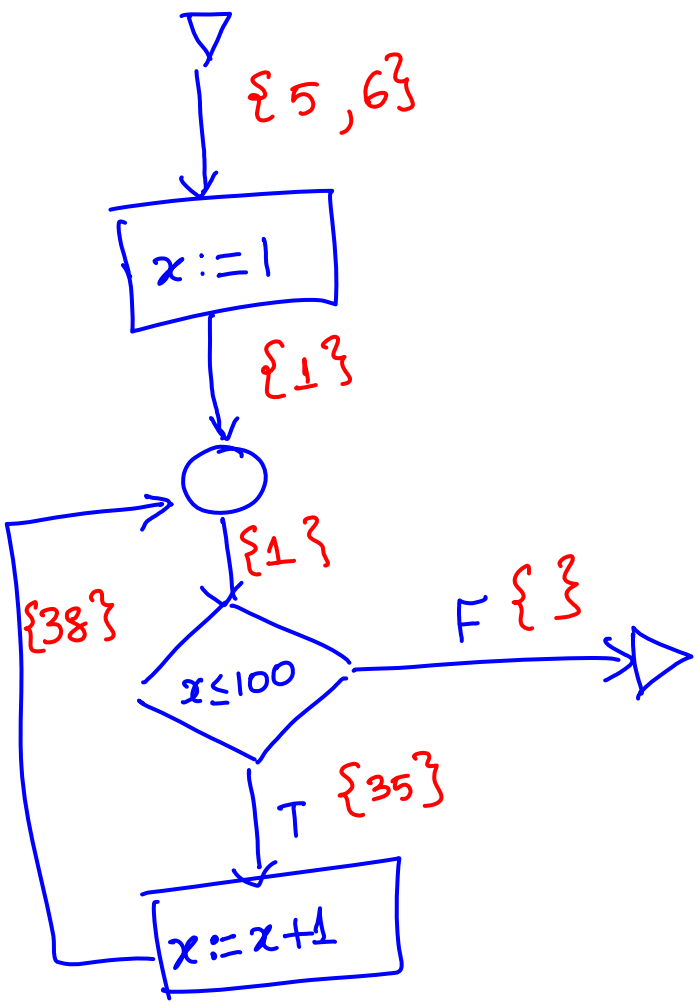
Operational Semantics

$$C_q \in Contexts = 2^{Env}$$

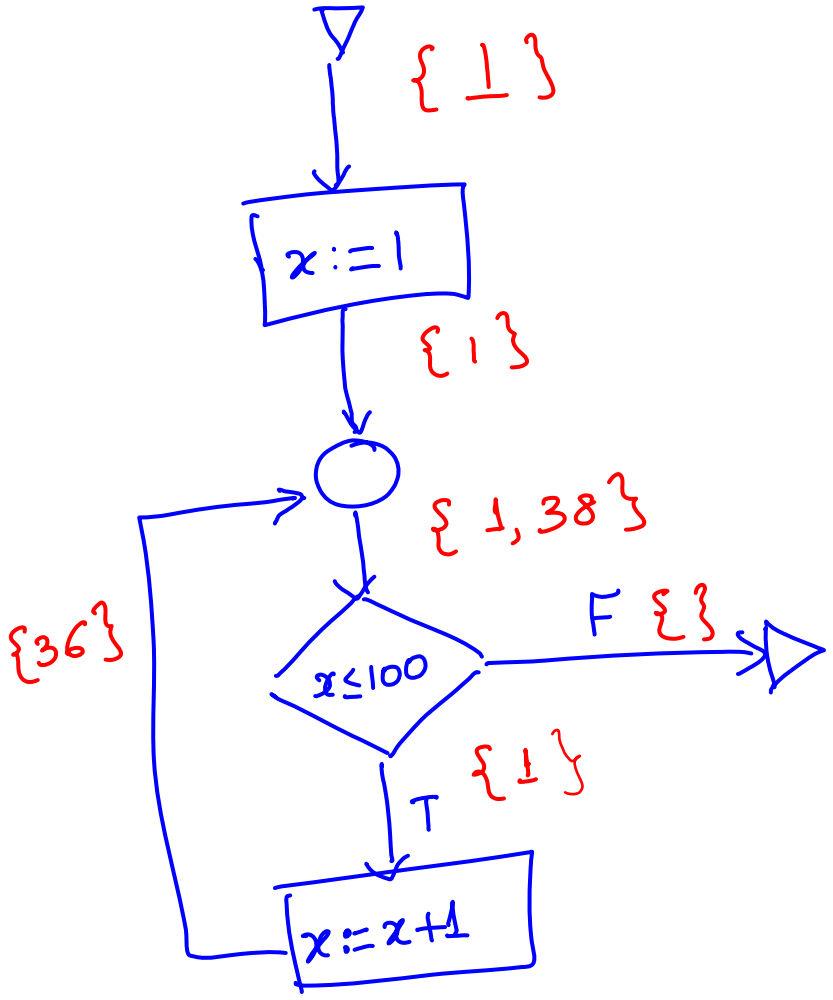
$$C_q = \{ e \mid \exists i \in Init\text{-}states : i \rightarrow \langle q, e \rangle \}$$

$$ContextVectors = Arcs \rightarrow Contexts$$

$$F\_cont : ContextVectors \rightarrow ContextVectors$$



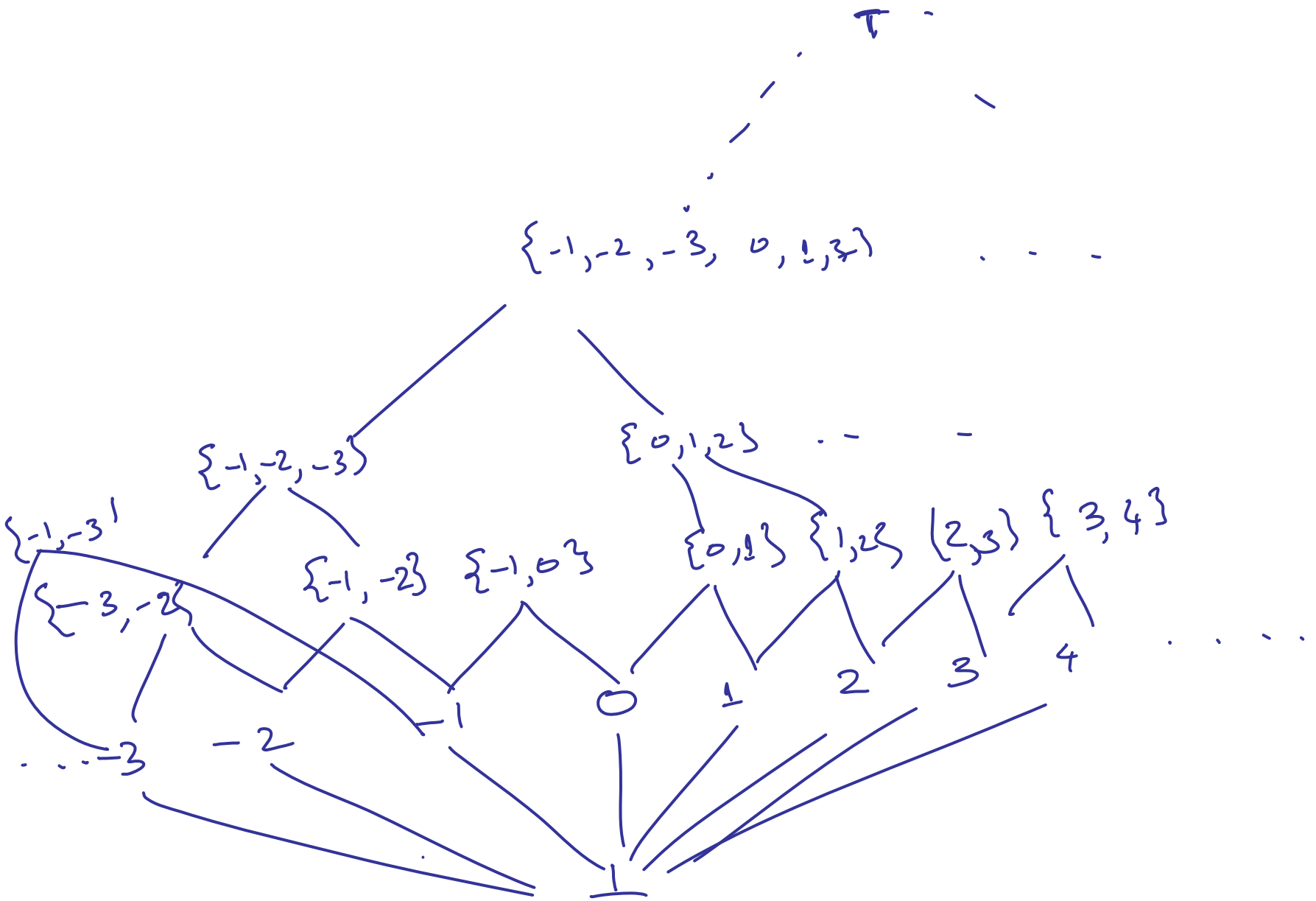
F_Cont



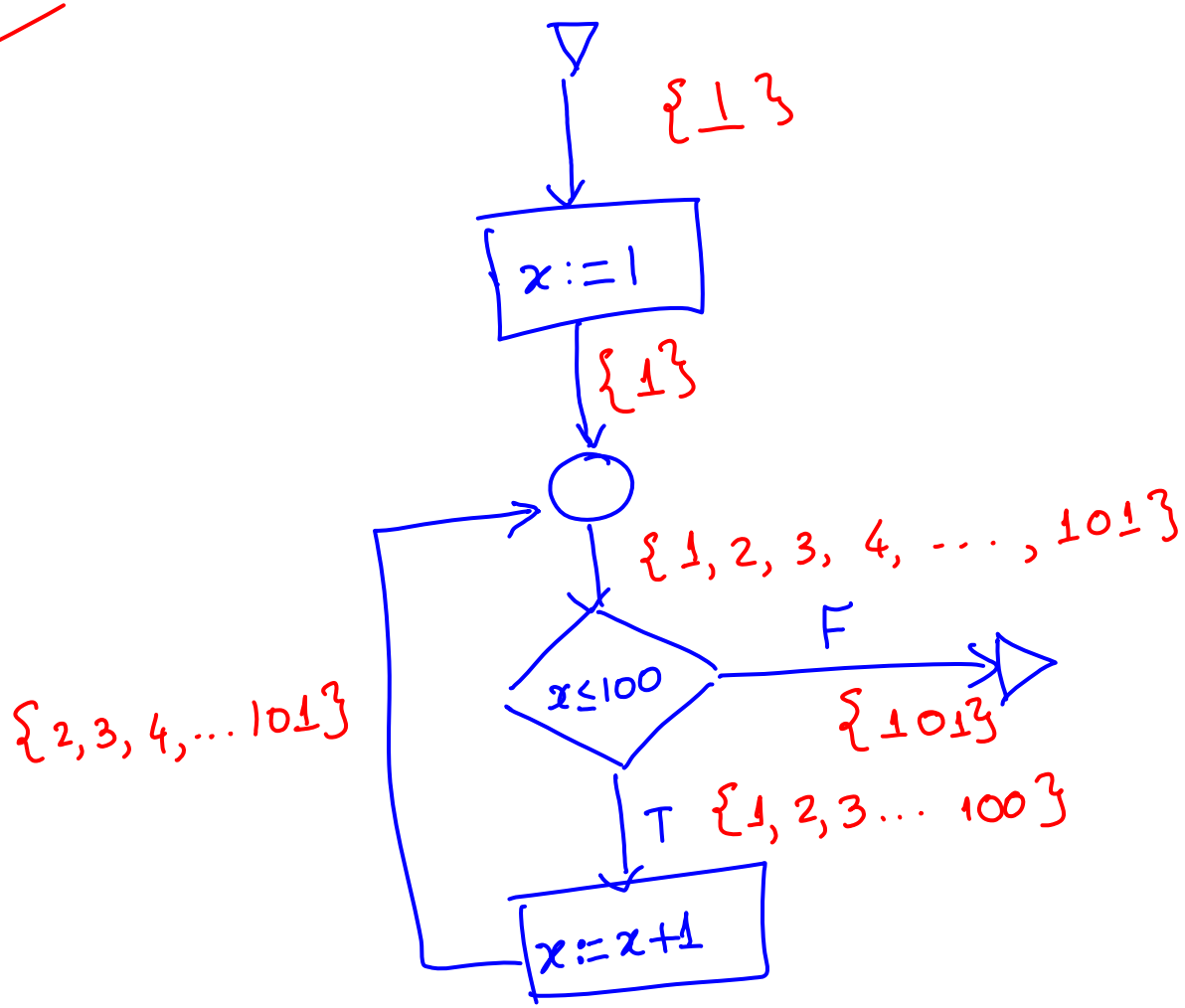
Equivalence

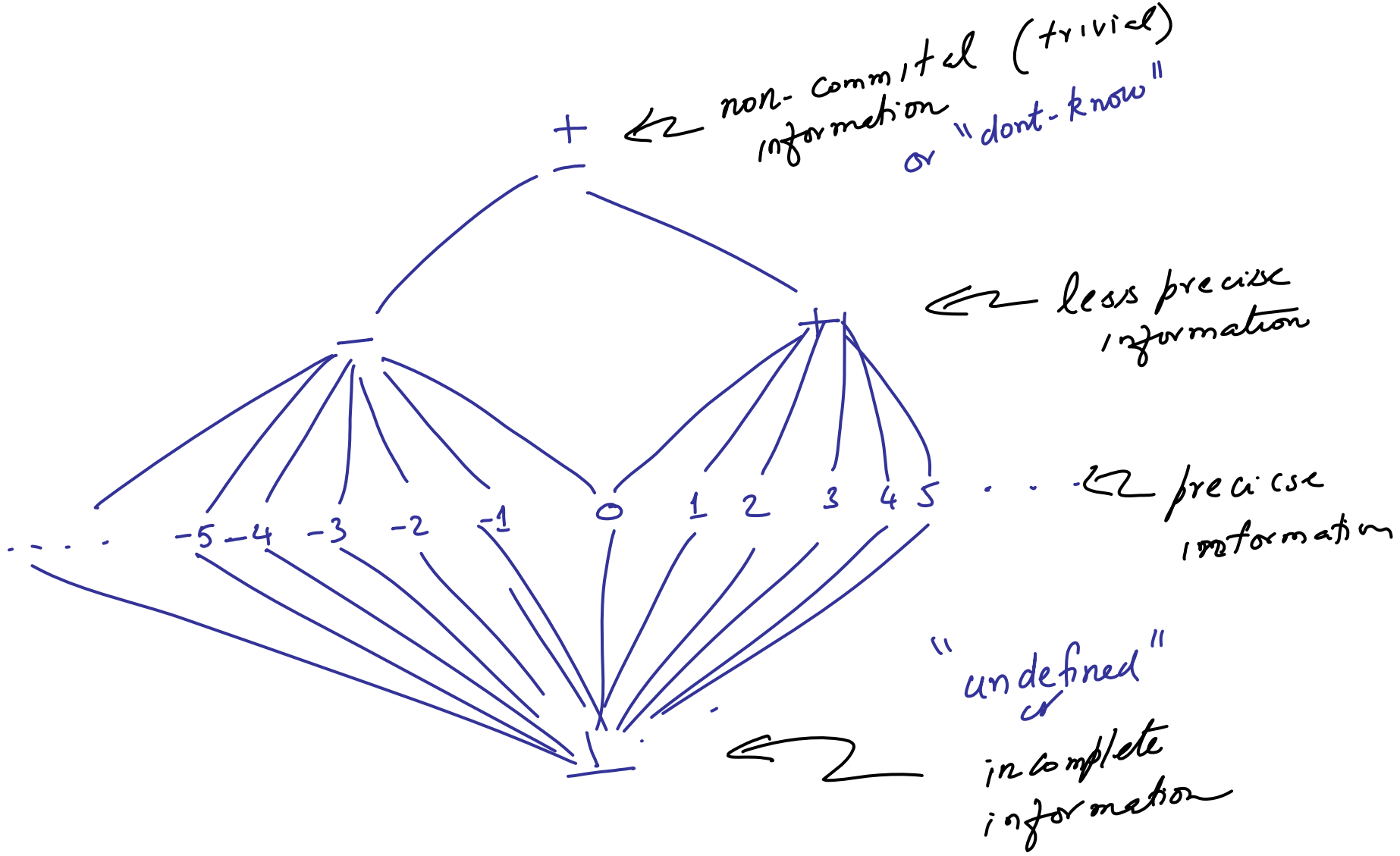
$$C_v = \lambda q.\ \{ e \mid \exists i \in \text{Init\_states}.\ i \rightarrow_0 \langle q, e \rangle \}$$

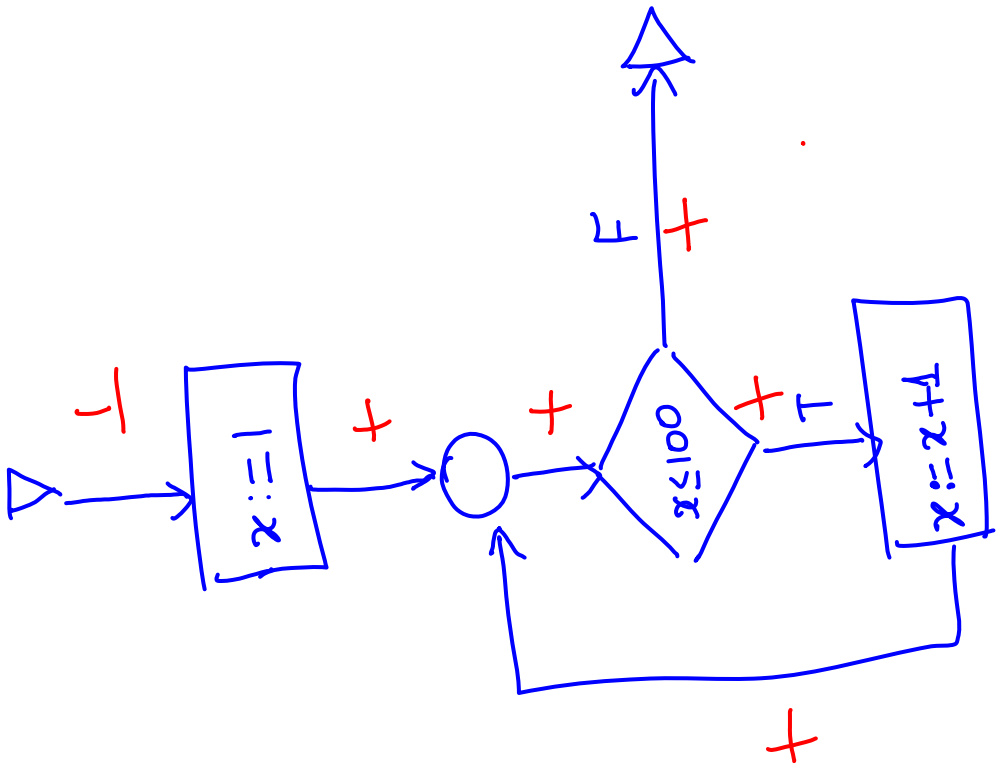
$$C_v = \text{LFP}(\text{F\_cont})$$

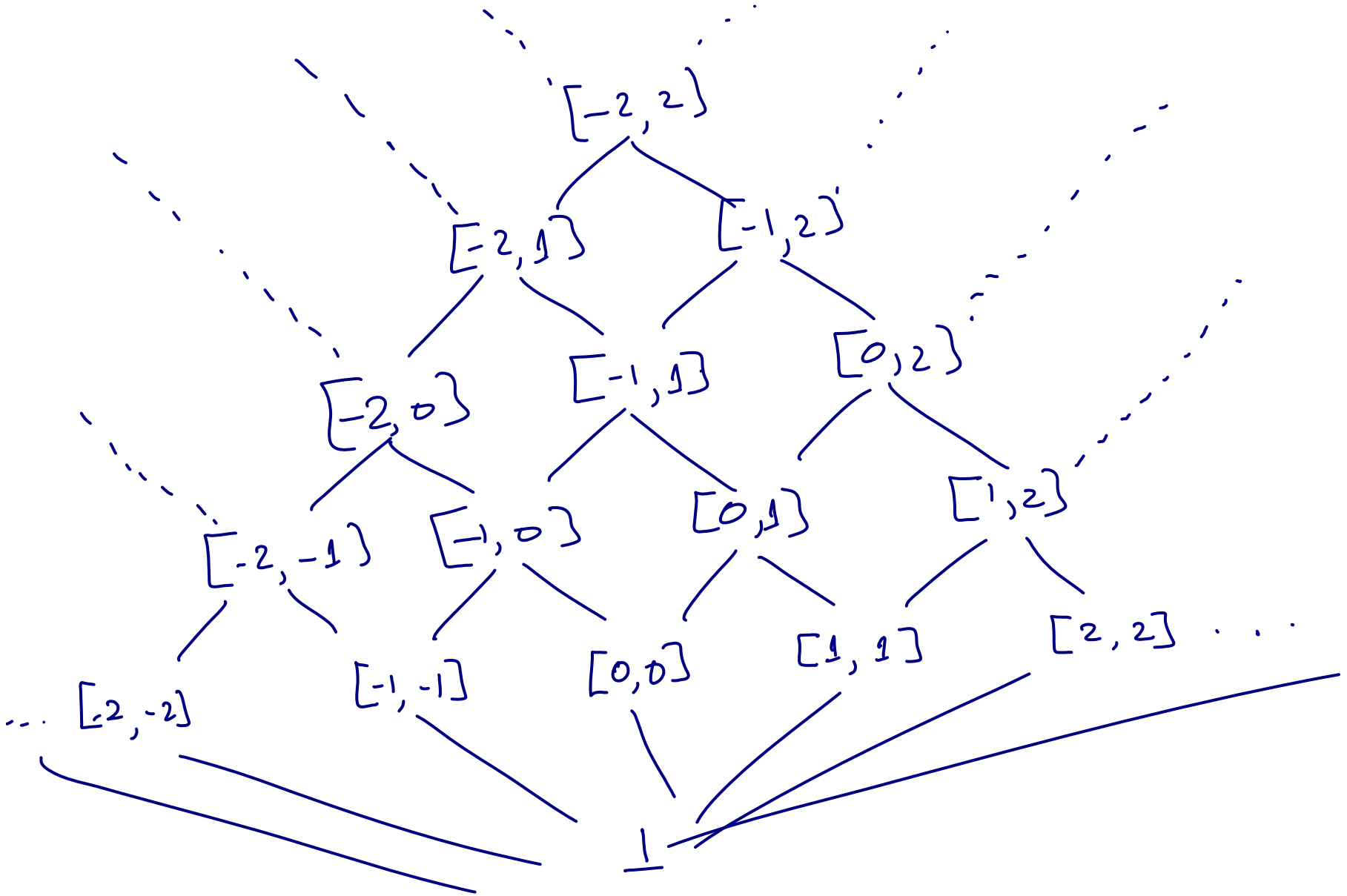

T
{-1,-2,-3, 0, 1, 2}
{-1,-2,-3}
{0,1,2}
{-1,-3}
{-3,-2}
{-1,-2}
{-1,0}
{0,1}
{1,2}
{2,3}
{3,4}
-3
-2
-1
0
1
2
3
4
1

Fixpoint

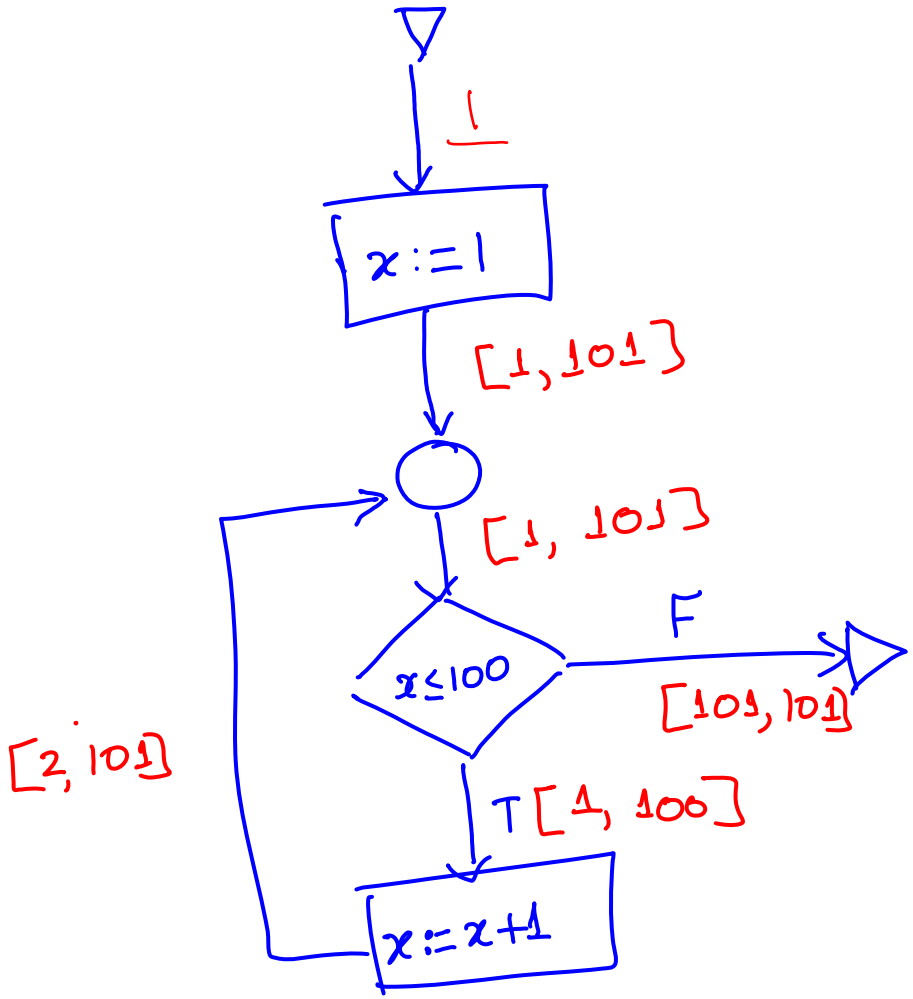



non-committal (trivial) information
or "dont-know"
less precise information
precise information
"undefined"
or
incomplete information
-5 -4 -3 -2 -1 0 1 2 3 4 5


x := 1
x<=100
x := x+1

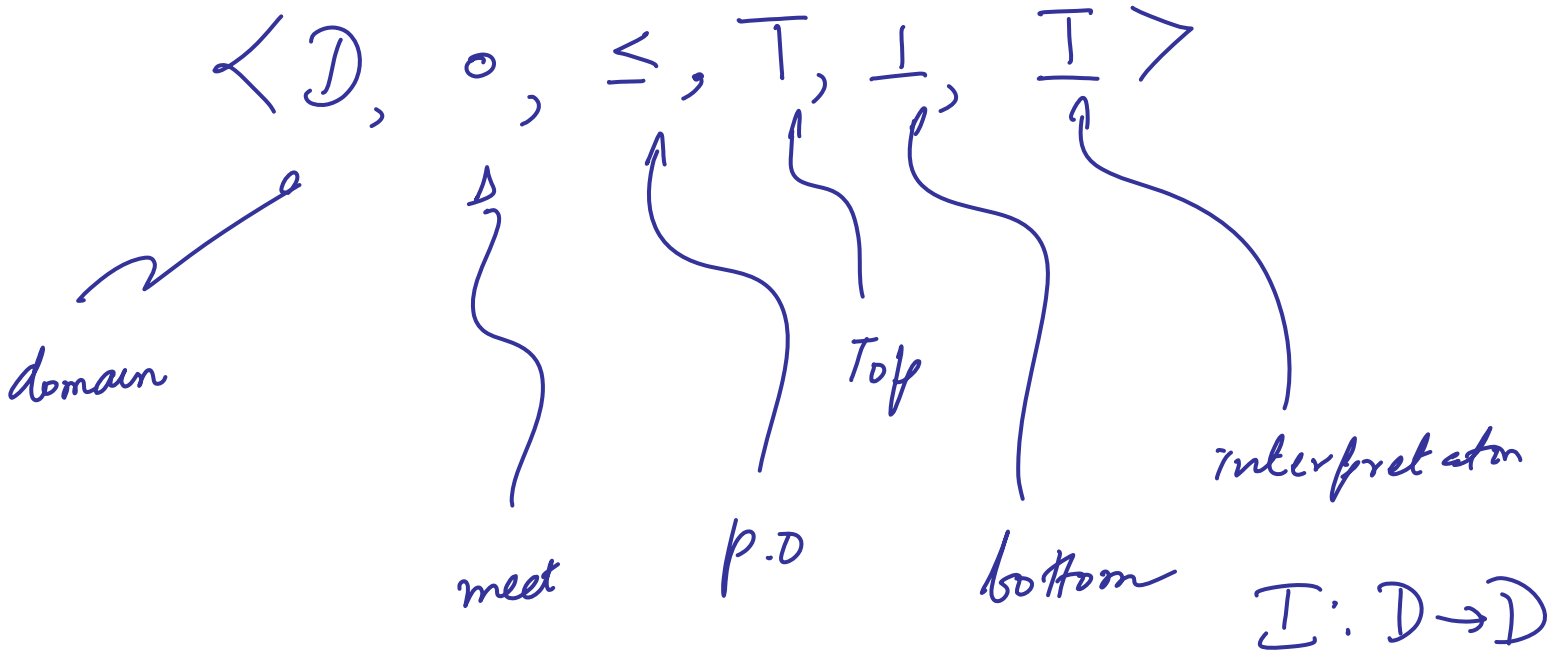

[-2, 2]
[-2, 1]
[-1, 2]
[-2, 0]
[-1, 1]
[0, 2]
[-2, -1]
[-1, 0]
[0, 1]
[1, 2]
[-2, -2]
[-1, -1]
[0, 0]
[1, 1]
[2, 2]
...
...
1

Fixpoint

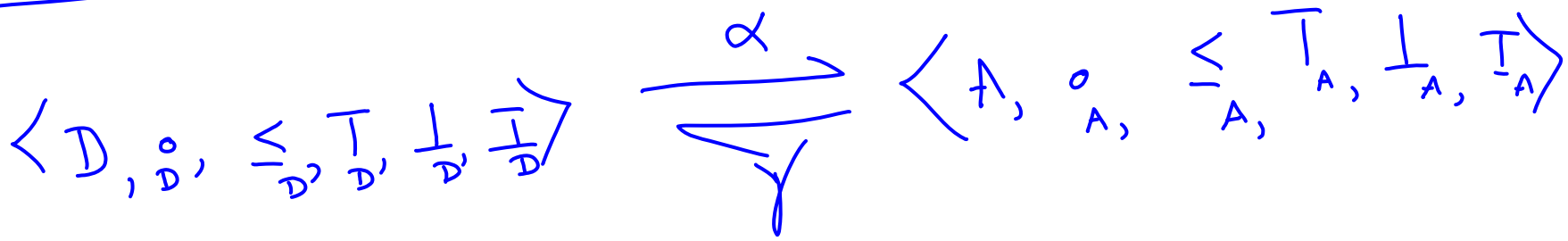


So...

an abstract interpretation is really



Science of Sound Abstract Interpretations



eg:

Sets of Integers $\rightleftharpoons$ Signs

Sets of Integers $\rightleftharpoons$ Intervals

Abstract interpretations Theorem 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. $\forall a \in A \; a = \alpha(\gamma(a))$

From Galois connection to abstract state transition fn.

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

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

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

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

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

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

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

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

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