Formalization of Data Flow Analysis in Coq

Alison Smith

Vikas Shivashankar

Motivation

- Coq provides a natural machinery for assisting with proofs that are traditionally difficult.
- Proving correctness of DFA techniques is important because of its many applications.
  - Optimizing compilers
  - Error checking

Review of Data Flow Analysis

- DFA operates on control flow graphs (CFGs)
  - Directed graphs
  - Nodes represent statements
  - Edges represent control flow

A Simple Language

```
Skip                --> do nothing
Copy x a            -->  x := a
Plus x a1 a2        -->  x := a1 + a2
Minus x a1 a2       -->  x := a1 – a2
CondEq a1 a2 n'     -->  if a1 = a2
                         then goto n'
```

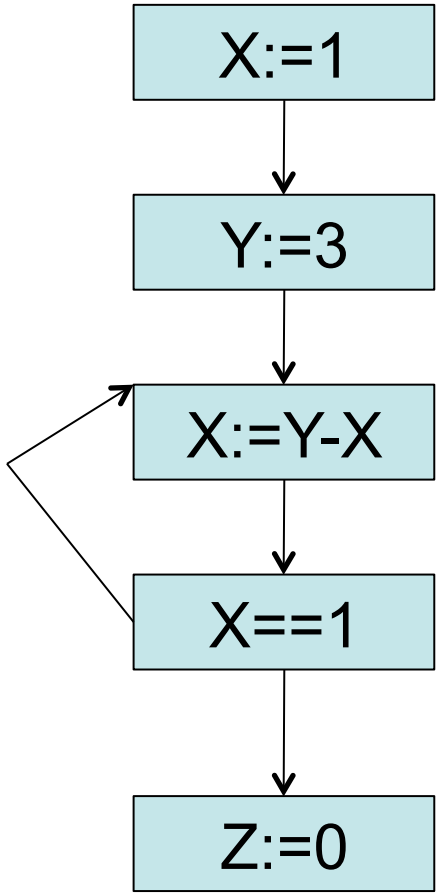
```
Inductive arg : Type :=
  | AId : id -> arg
  | ANat : nat -> arg.

Inductive label : Type :=
  | Label : nat -> label.

Inductive stmt : Type :=
  | Skip : stmt
  | Copy : id -> arg -> stmt
  | Plus : id -> arg -> arg -> stmt
  | Minus : id -> arg -> arg -> stmt
  | CondEq : arg -> arg -> label -> stmt.
```

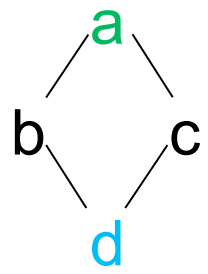
Control Flow Graph Example

0: X:=1
1: Y:=3
2: X:= Y-X
3: if x==1 then 2
4: Z:= 0



Lattices

- A partial order is a binary relation over a set which is reflexive, antisymmetric, and transitive.
- A lattice is a partial order over which meet (greatest lower bound) and join (least upper bound) are defined.



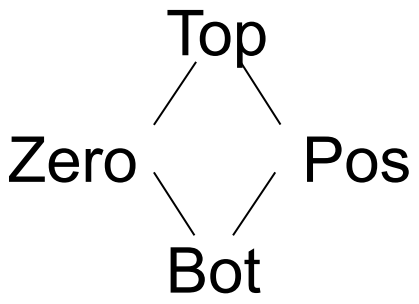
- Data flow facts form a lattice.

Data Flow Analysis – Signs

- Analysis maps each program label to a mapping from identifiers to their sign at the beginning of that statement
  - abstract state : id -> sign
  - analysis result : label -> abstract state
- Analysis is sound for a given cfg if it holds of all possible program runs.

Data Flow Analysis - Signs

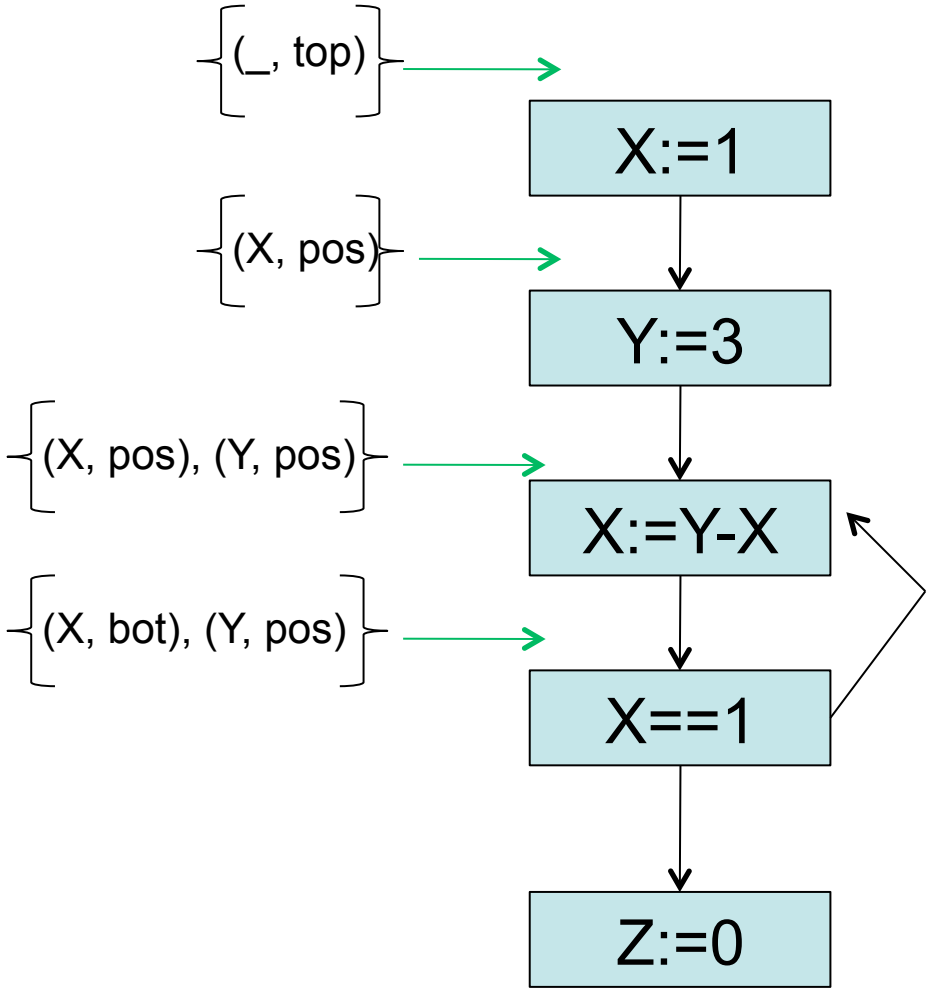
Sign := Bot | Zero | Pos | Top



- Bot is the least informative abstract value of a sign, whereas Top is the most informative.
- At the beginning of analysis of a program, all ids are mapped to top and then successively refined with each statement.

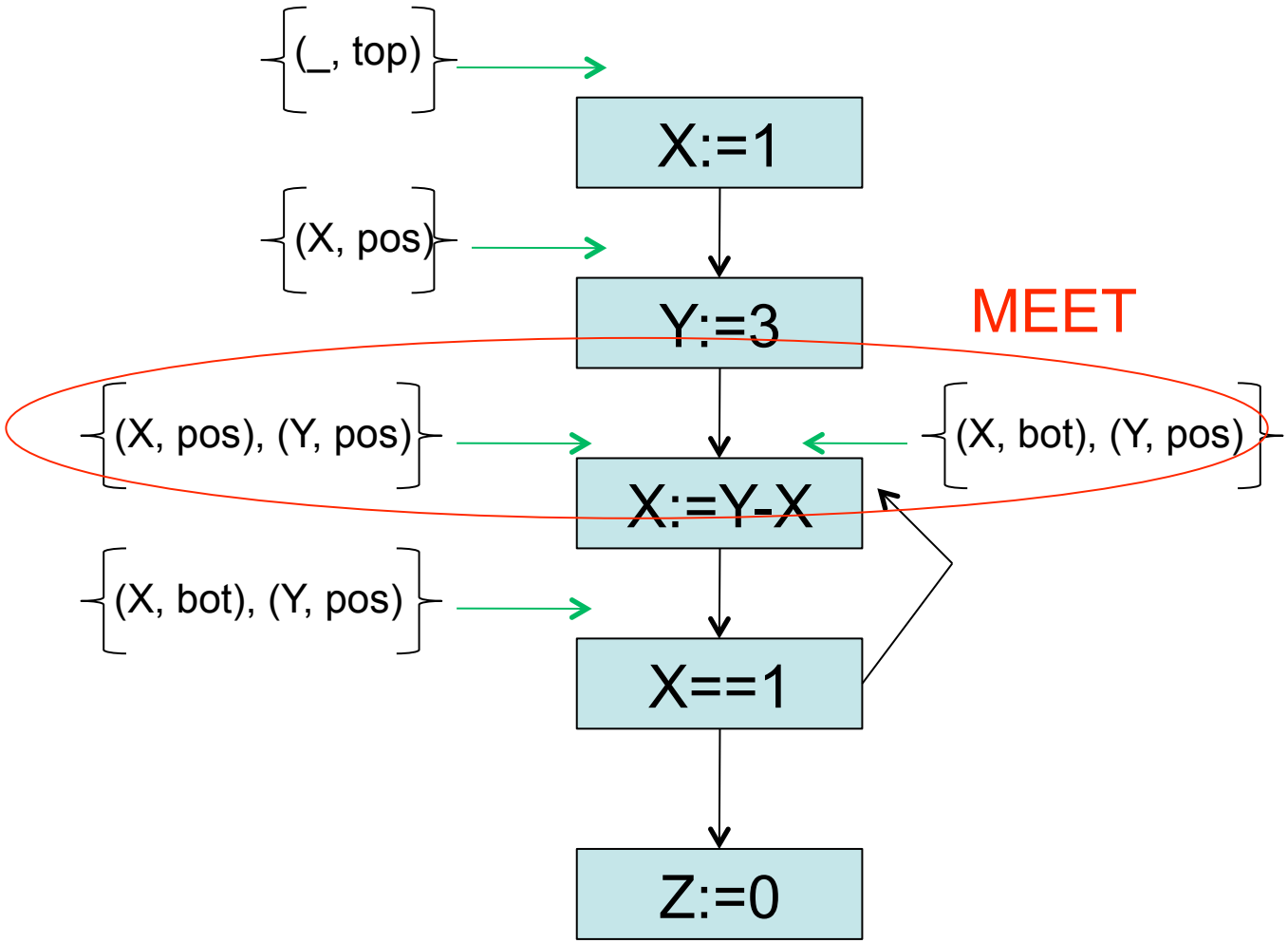
Data Flow Analysis - Signs

0: X:=1
1: Y:=3
2: X:= Y-X
3: if x==1 then 2
4: Z:= 0



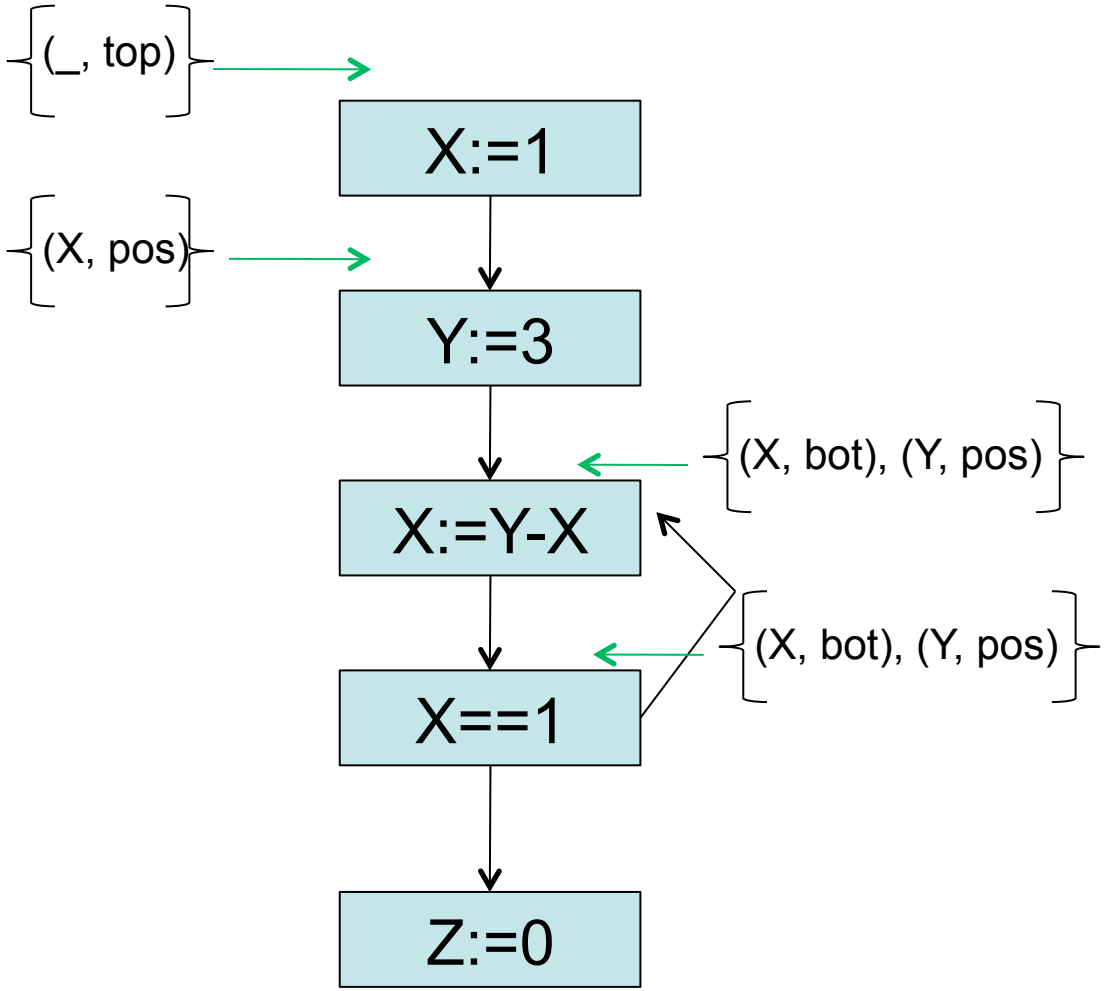
Data Flow Analysis - Signs

0: X:=1
1: Y:=3
2: X:= Y-X
3: if x==1 then 2
4: Z:= 0



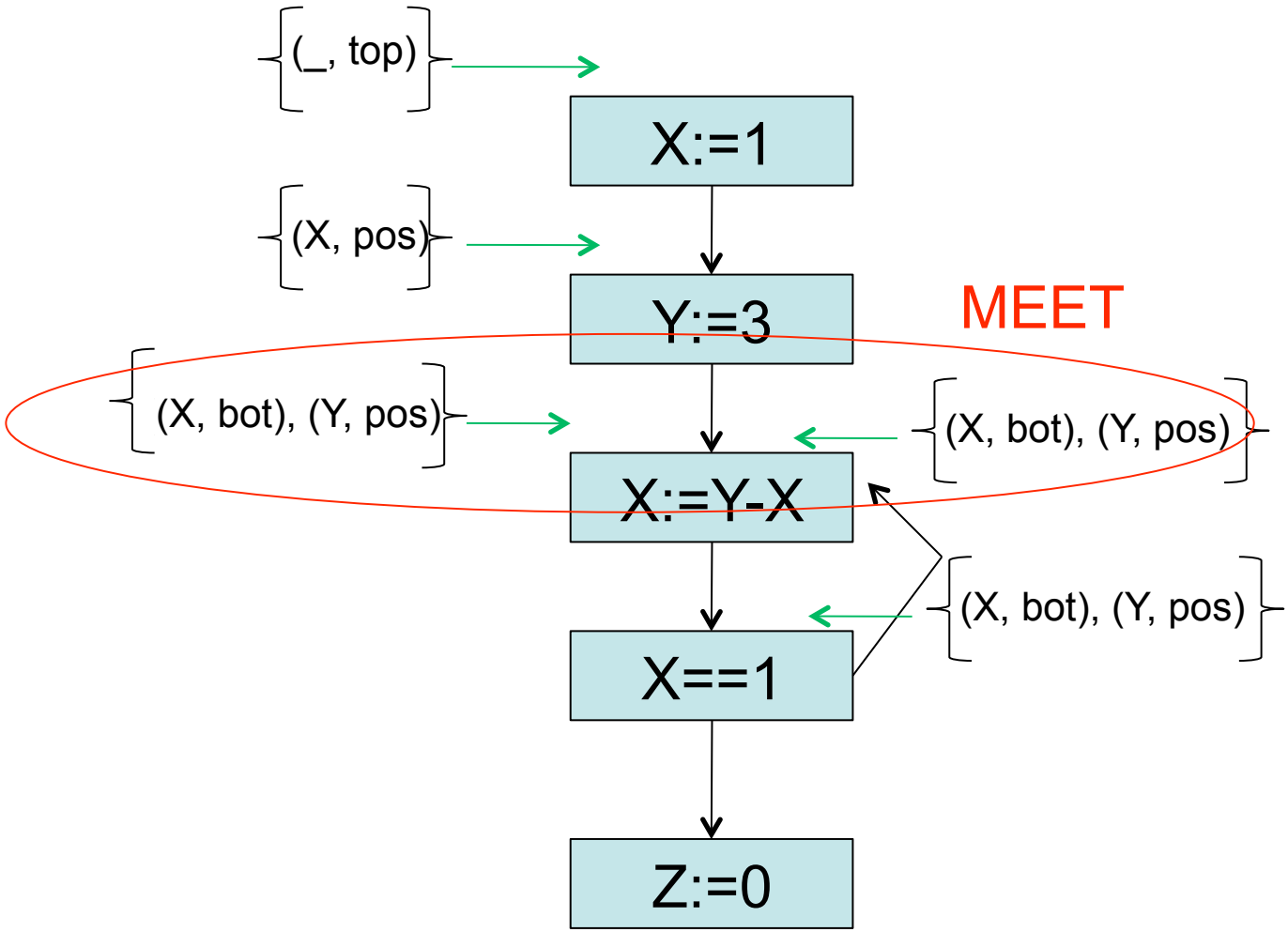
Data Flow Analysis - Signs

0: X:=1
1: Y:=3
2: X:= Y-X
3: if x==1 then 2
4: Z:= 0



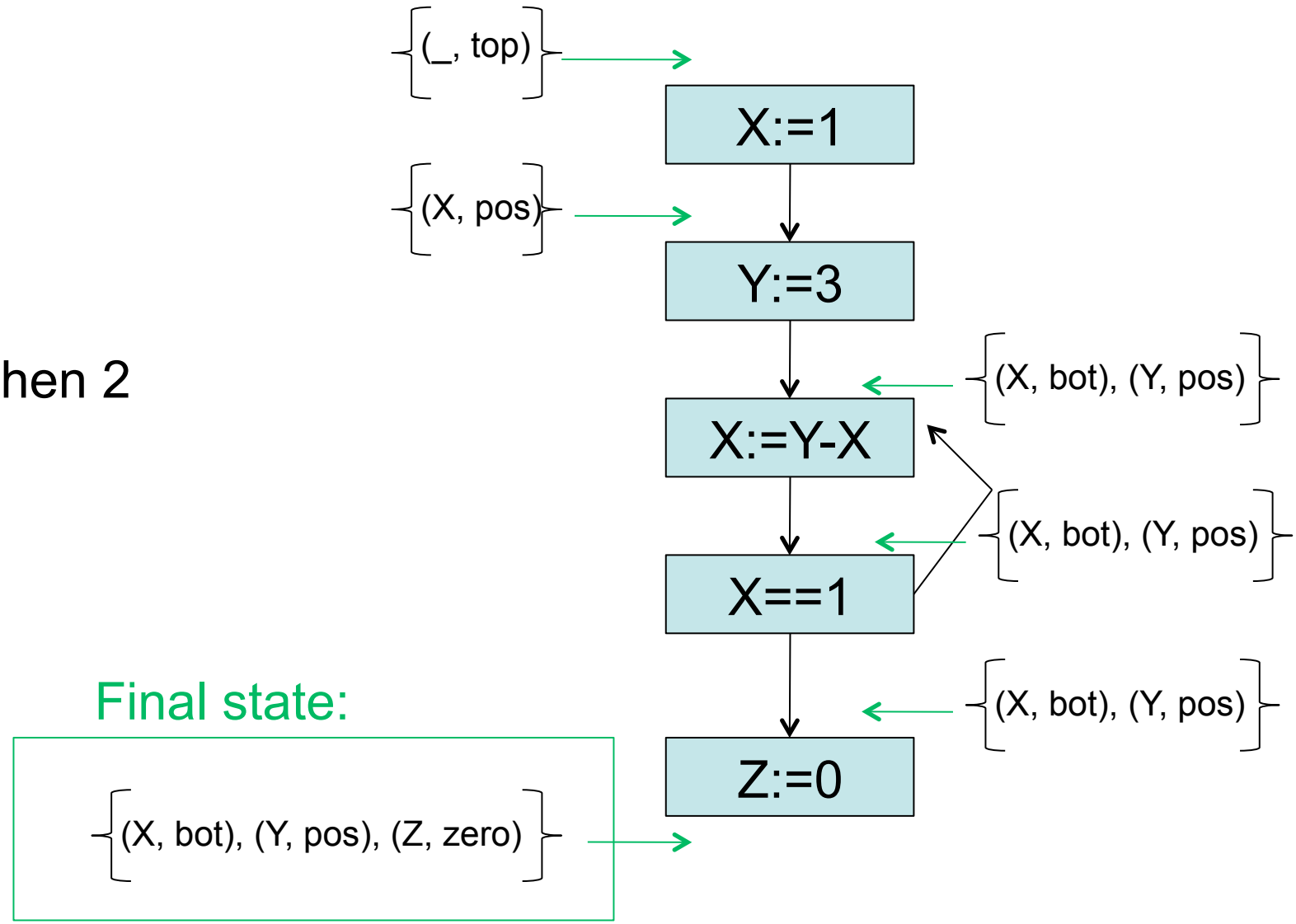
Data Flow Analysis - Signs

0: X:=1
1: Y:=3
2: X:= Y-X
3: if x==1 then 2
4: Z:= 0



Data Flow Analysis - Signs

0: X:=1
1: Y:=3
2: X:= Y-X
3: if x==1 then 2
4: Z:= 0



Dataflow analysis in Coq

- Defining base language in Coq
- CFG definition
  - Examples of CFG evaluation
- DFA definition
  - Abstract state
  - Lattice
  - Sound Analysis

Coq demo

Sound Dataflow Analysis

- step : abstract_state -> stmt -> abstract_state
- meet : $2^{\{abstract\_state\}}$ -> abstract_state
- Theorem : ∀ c : cfg, ar : analysis_result,
  - c ~ ar
  - ar satisfies the above dataflow equations

    => ar is a fixpoint