Static Analysis with Abstract Interpretation

Presented by Guy Lev

06.04.2014

Outline

- Introduction
- Concrete & Abstract Semantics
- Abstract Domain
- Abstract Domains - examples
- Conclusion

Introduction

- **Static Analysis** – automatically get information about the possible executions of computer programs
- Main usages:
  - Compilers: decide whether certain optimizations are applicable
  - Certification of programs against classes of bugs

Introduction

- Last week: Splint – unsound static analysis (can miss errors)
- **Abstract Interpretation (AI)** - a theory of sound (conservative) approximation of the semantics of computer programs

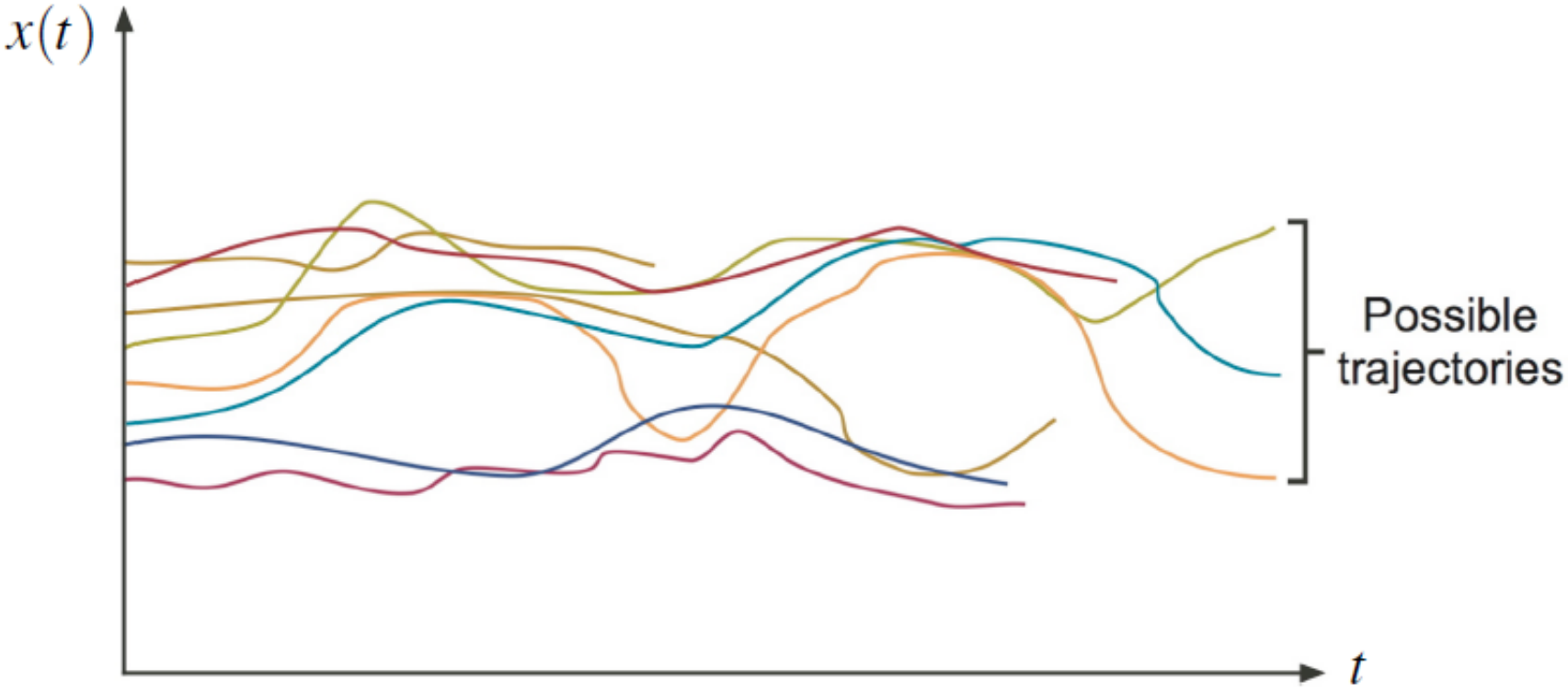
Introduction

- Soundness
  - If we proved some property: we are sure it is true for all possible executions of the program
  - If we were not able to prove a property: we cannot infer anything
- For example, if our analysis showed that:
  - No divisions by zero → it's for sure
  - A division by zero may occur → it might be a false alarm

Concrete semantics of programs

- Representation of the set of all possible executions of a program in all possible execution environments
- Environment: Input parameters, values of uninitialized variables, input from user, clock value, etc.
- Execution: a curve x(t)
  - A vector representing the state of the program
- State of the program: everything that interests us: values of variables, heap status, elapsed time, etc.

Concrete semantics: a set of curves



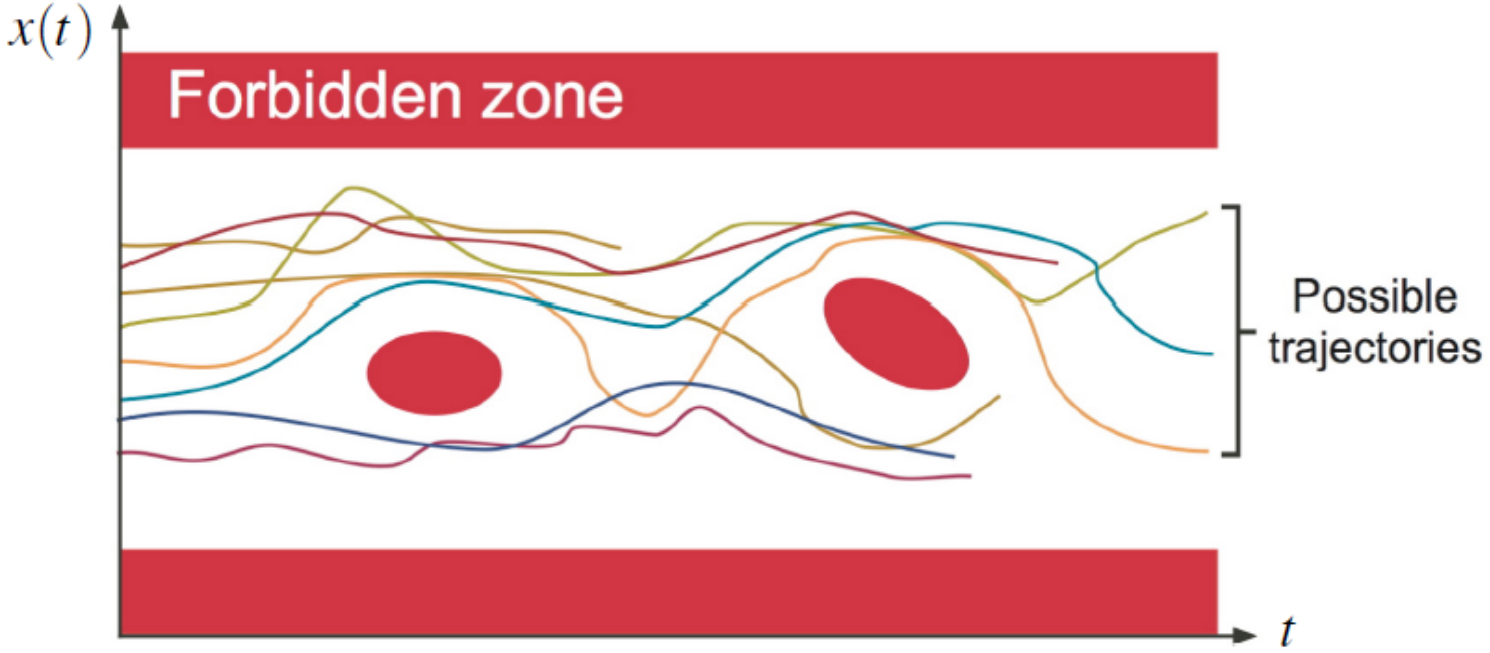
Undecidability

- The concrete semantics of a program is an infinite mathematical object which is *not computable*

⇒ All non trivial questions about the concrete semantics of a program are *undecidable*

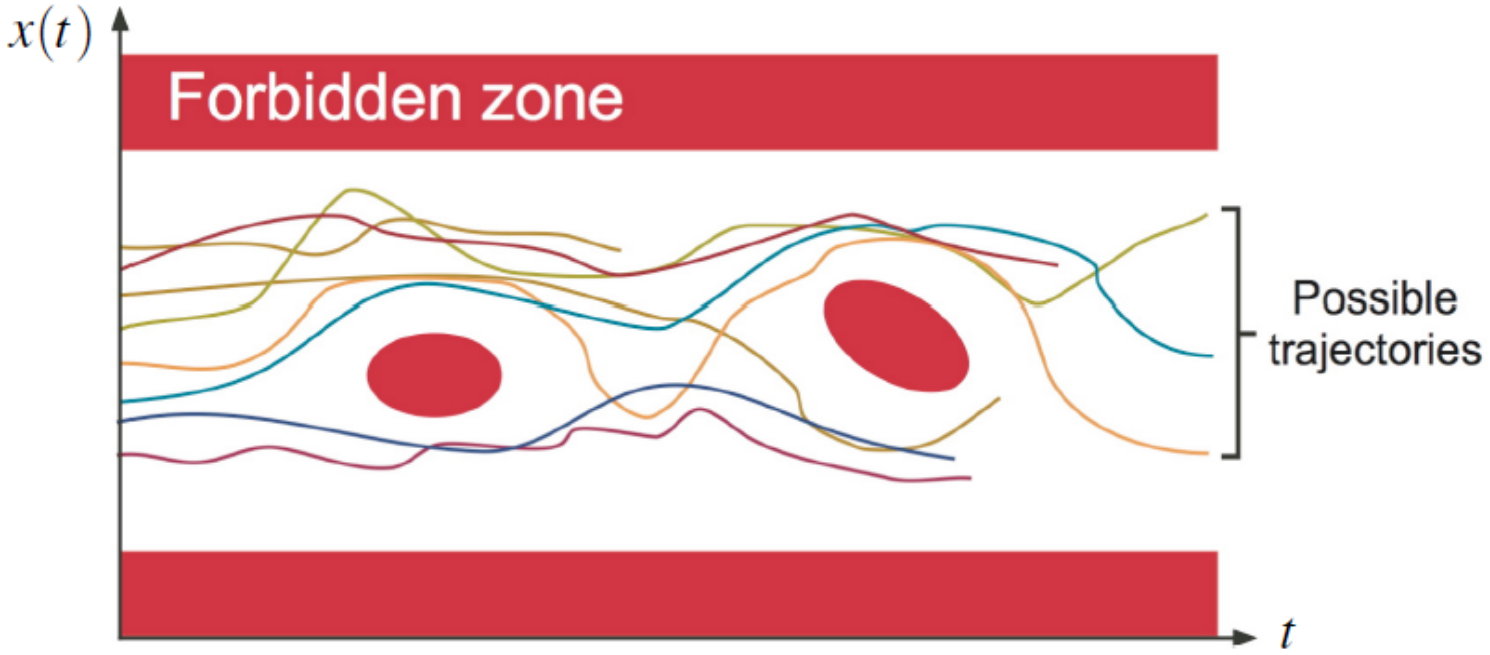
Safety Properties

- Given a program, we want to prove properties of it which express that no possible execution can reach an erroneous state



Safety Properties

- However, this verification problem is undecidable

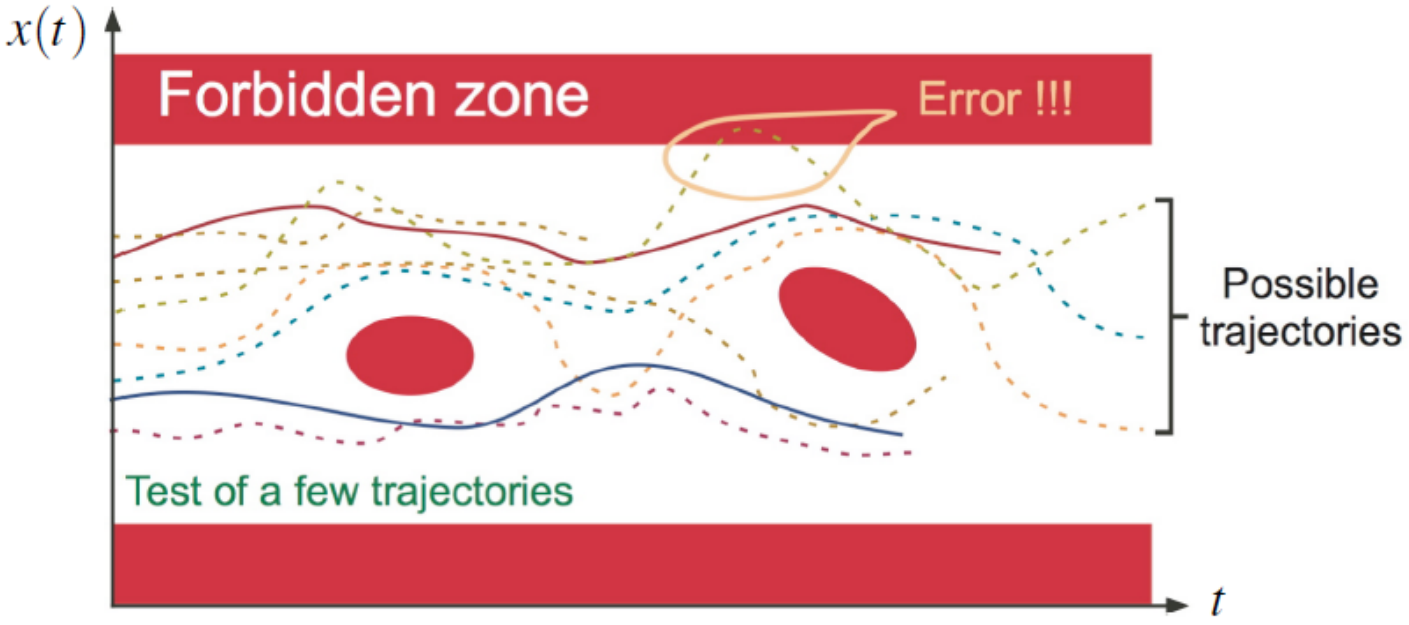


Testing

- Testing is an under-approximation of the program semantics:
  - Only part of executions are examined
  - Only the prefix of executions

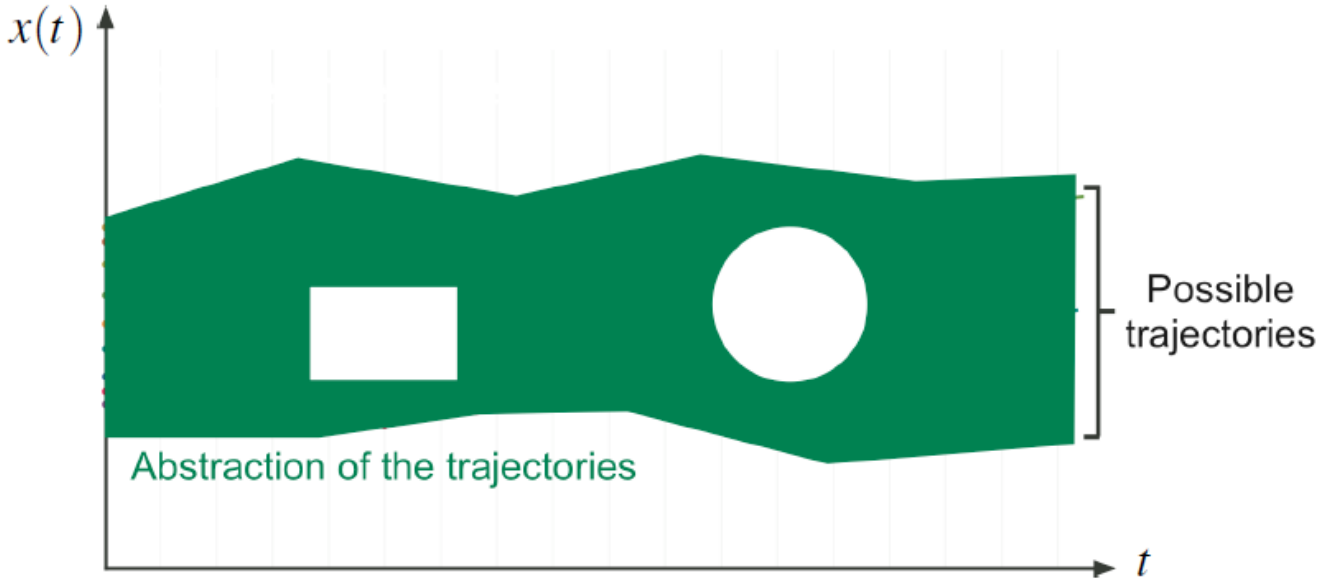
Testing

- Some erroneous executions might be forgotten:



Abstract Interpretation

- Considers an ***abstract semantics:*** a superset of the concrete semantics of the program
- An over-approximation of the possible executions



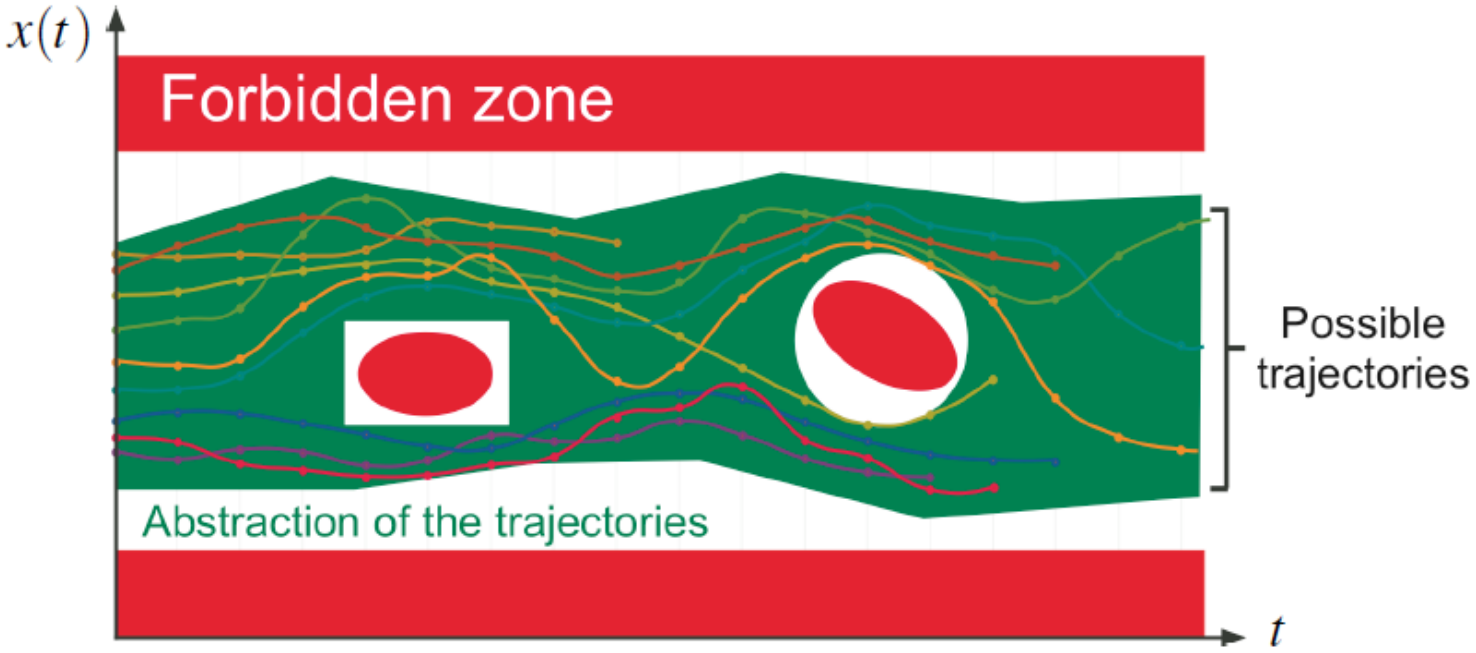
Abstract Semantics

- Abstract Semantics should be:
  - computer representable
  - effectively computable from the program text

Abstract Interpretation

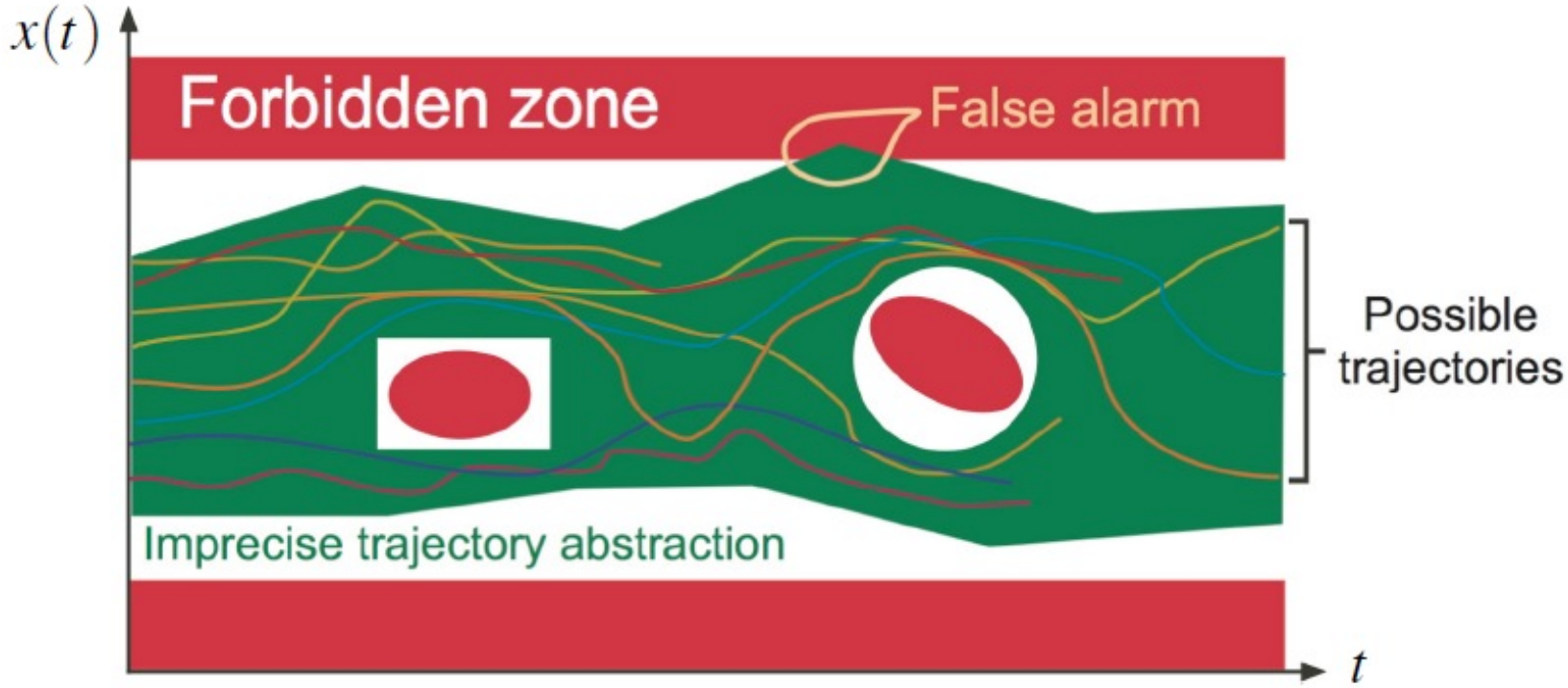
- If the abstract semantics is safe, then so is the concrete semantics

  ⇒ Soundness: no error can be missed



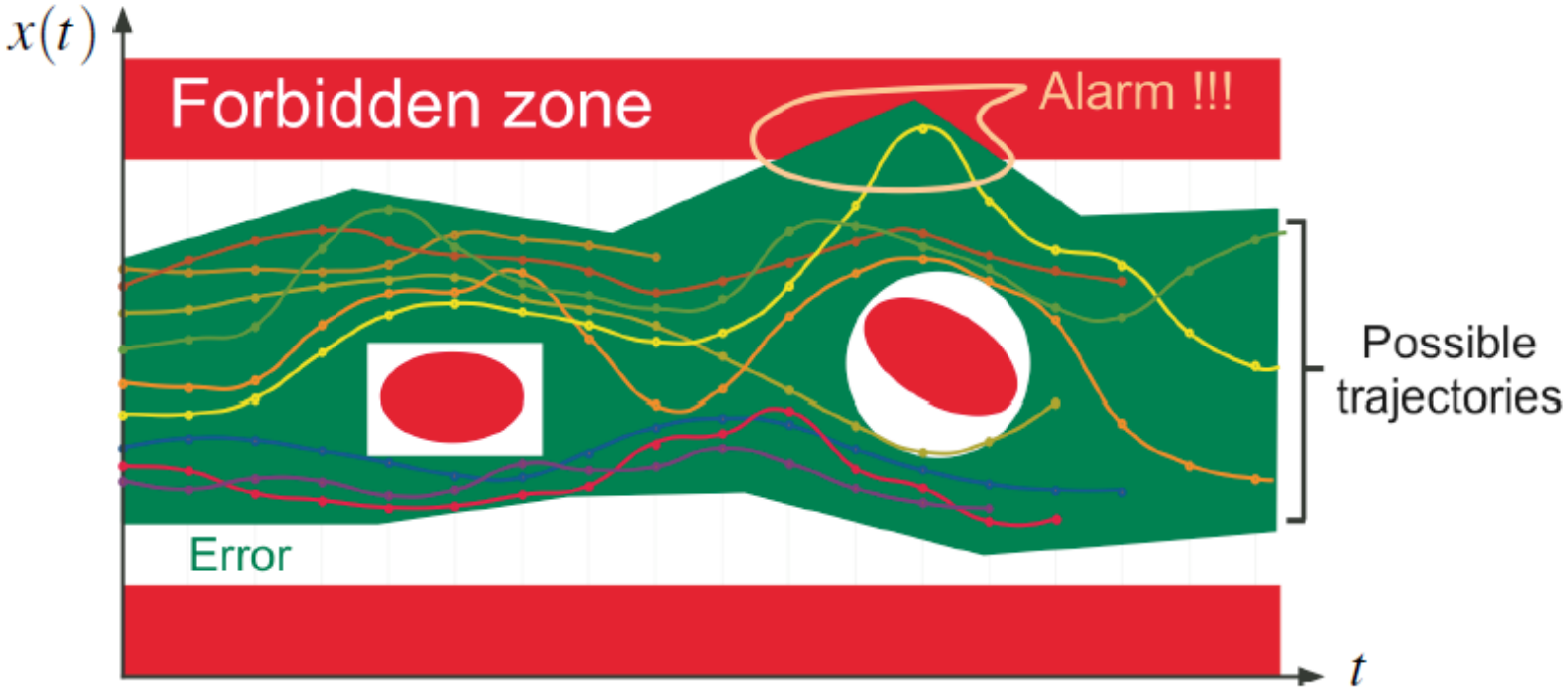
False Alarms

- If the over-approximation is too large, we might get false alarms:



Abstract Interpretation

- If no alarms: ensures safety
- In case of alarms: we don't know if they false or true:



To Summarize

- Testing: under-approximation, can miss errors
- Abstract Interpretation: over-approximation
  - Cannot miss any potential error
  - May yield false alarms
  - The objective: to get as precise abstraction as possible

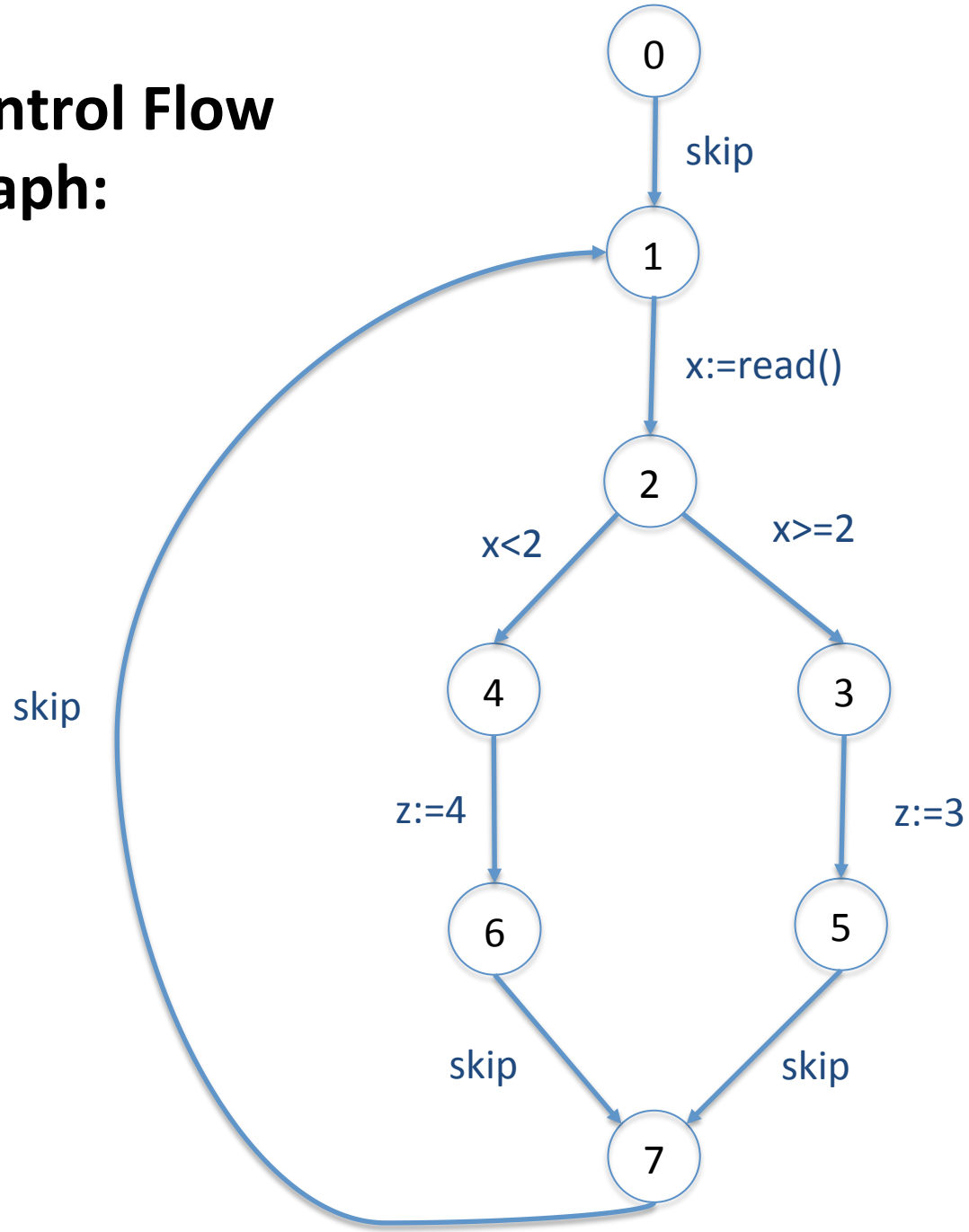
Example

- Let's analyze a program with 3 local variables: x, y, z
- Program Semantics: the values of these variables.
- Values are from $\mathcal{Z}$ (integers)

Example

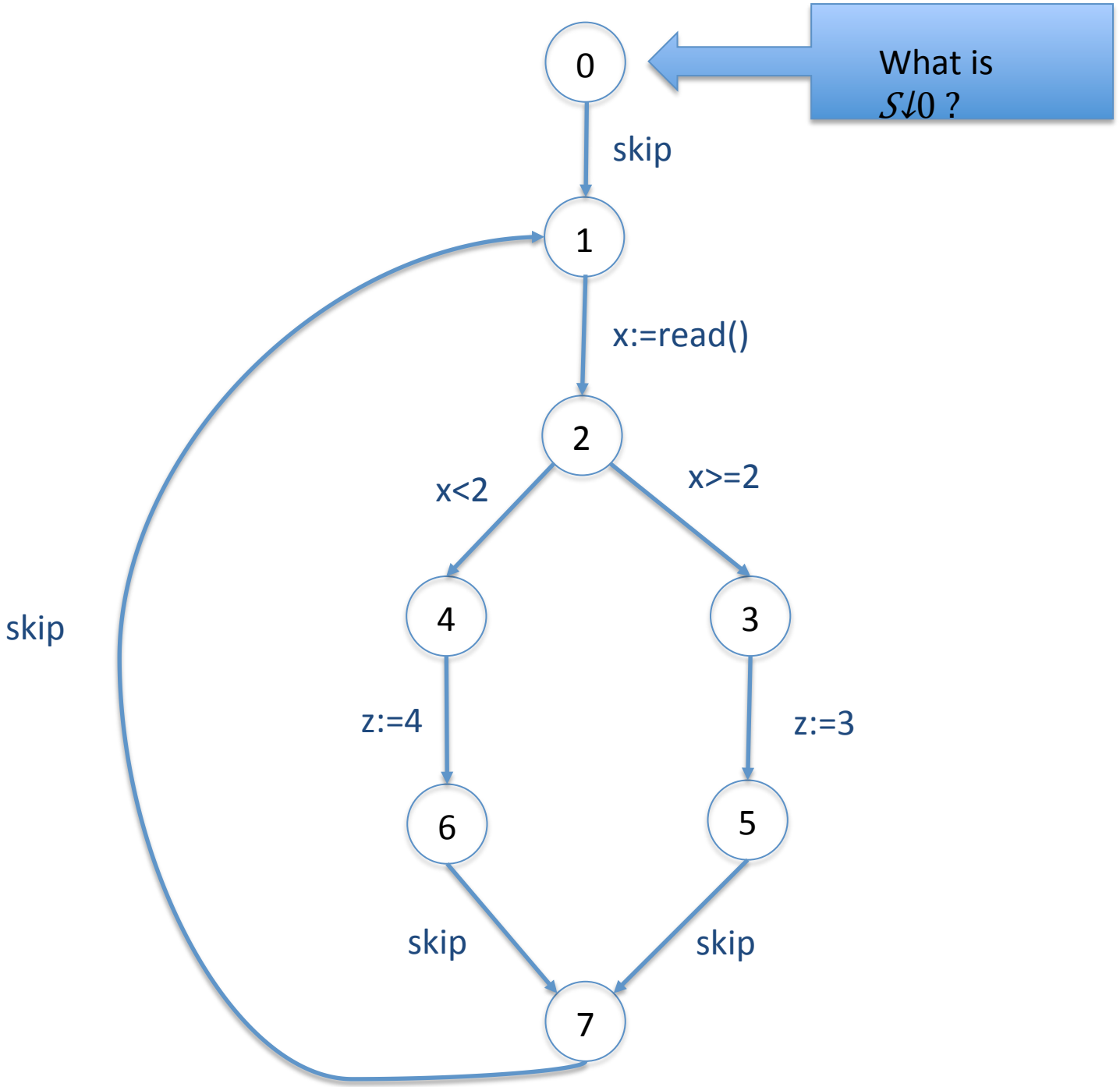
```
void f()
{
	int x,y,z;

	while (1)
	{
		x = read();

		if (x >= 2)
			z = 3;
		else
			z = 4;
	}
}
```

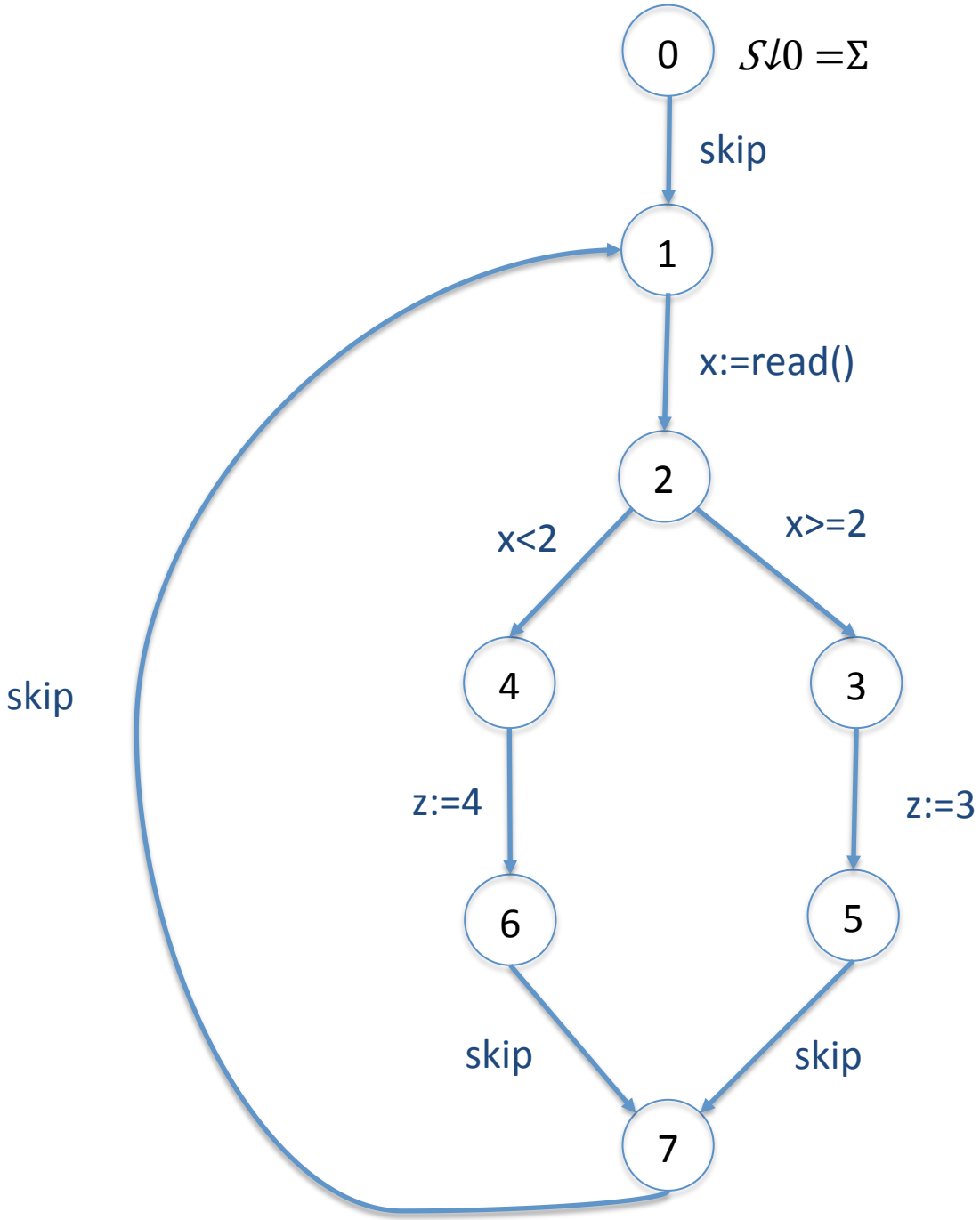
Control Flow Graph:

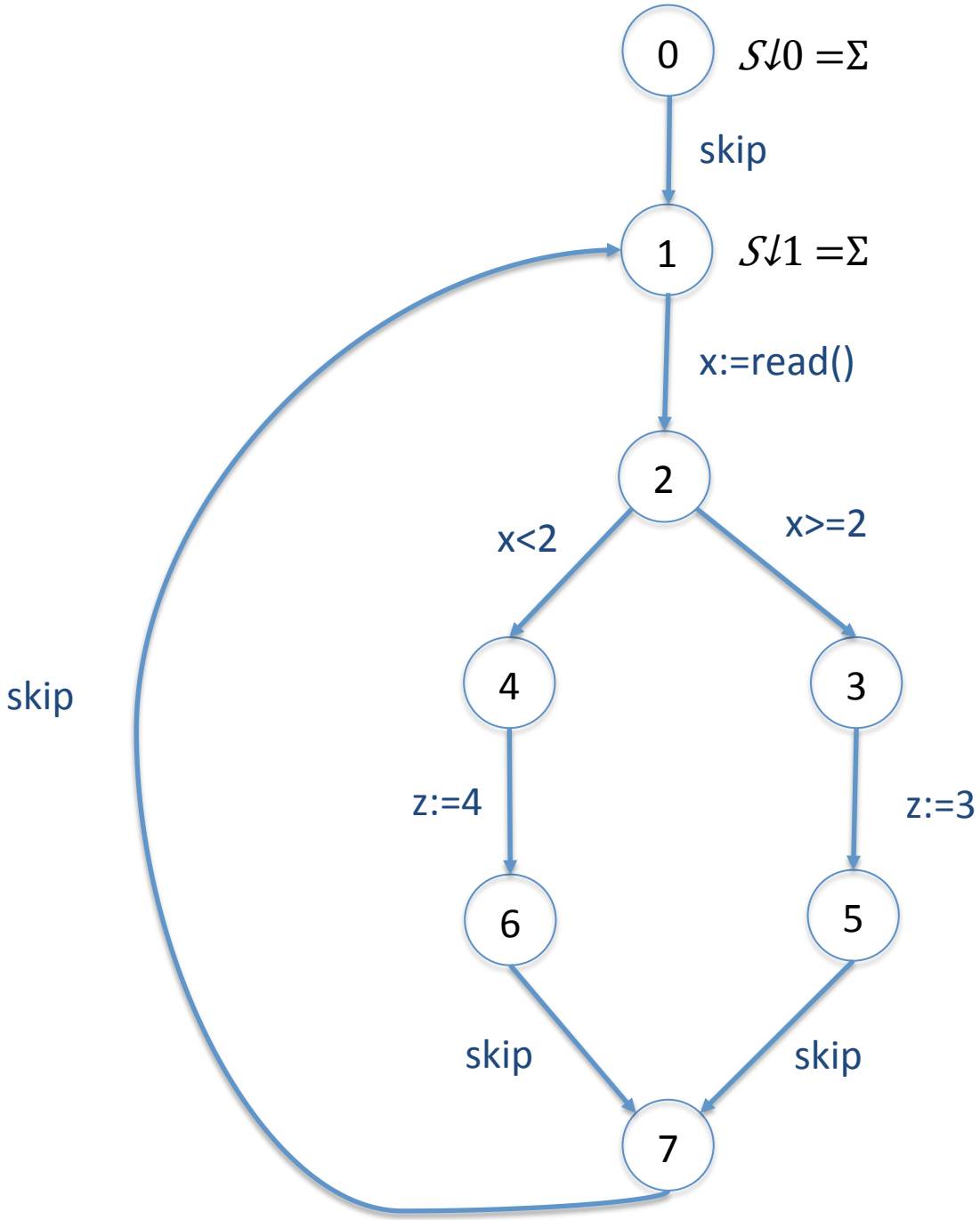


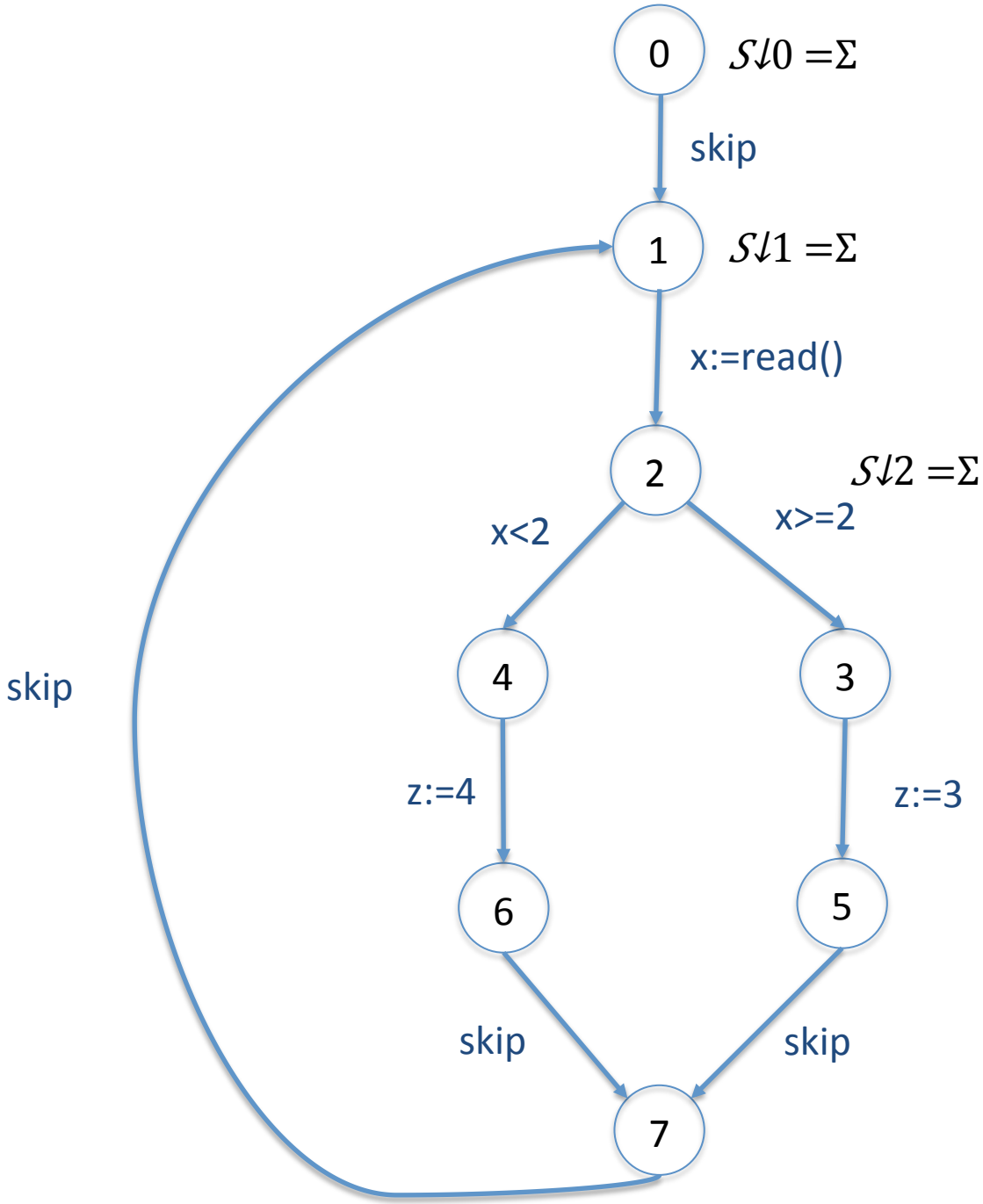
Concrete Semantics

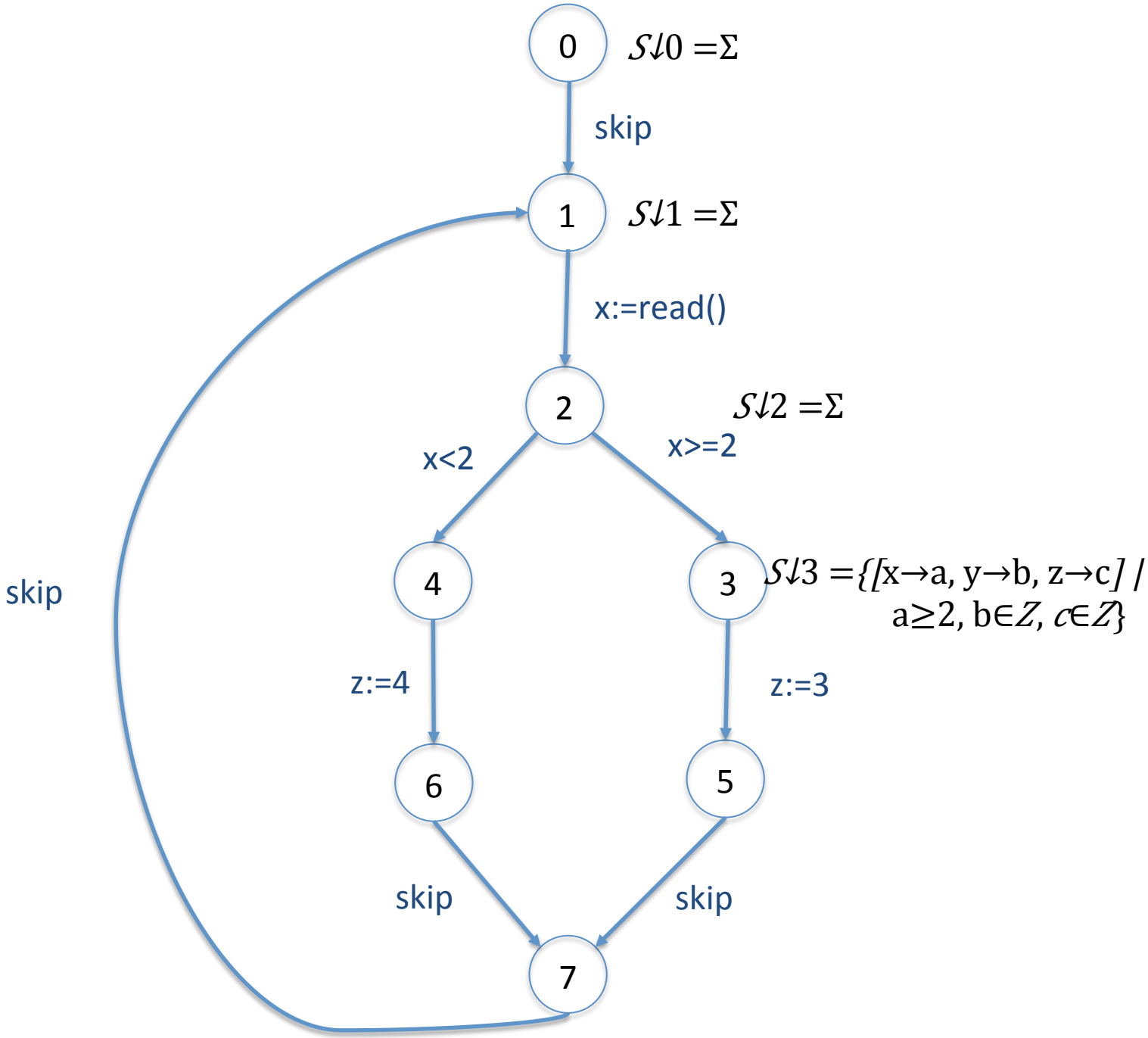
- We are interested in all possible states at each node
- Denote by Σ the set of all mappings $Var \rightarrow Z$
  - $Var = \{x, y, x\}$
- A state is a mapping $\sigma \in \Sigma$
- Each node has a subset $S \subseteq \Sigma$ of possible states

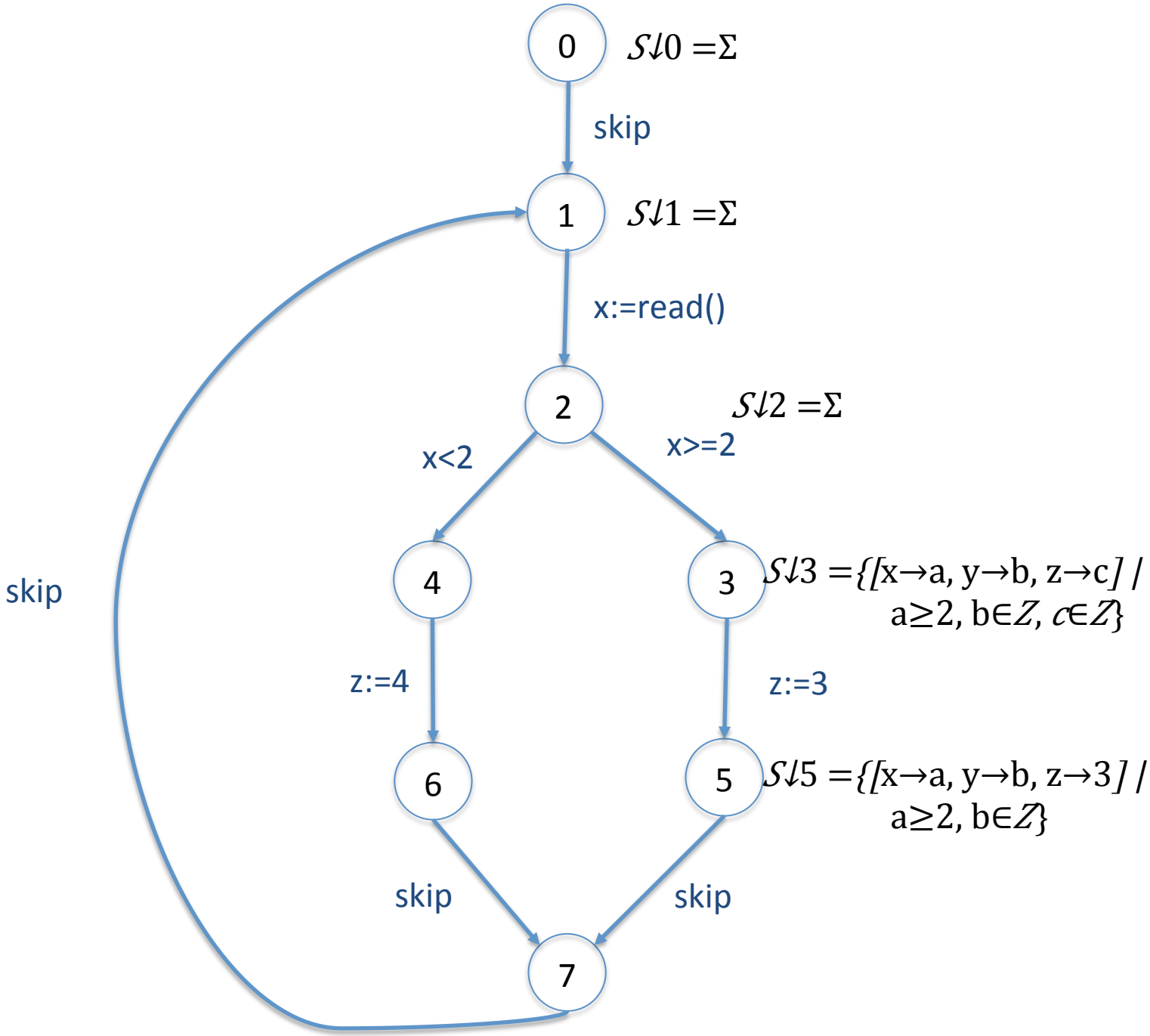

0
What is
$S\downarrow 0$?
skip
1
x:=read()
2
x<2
x>=2
4
3
skip
z:=4
z:=3
6
5
skip
skip
7

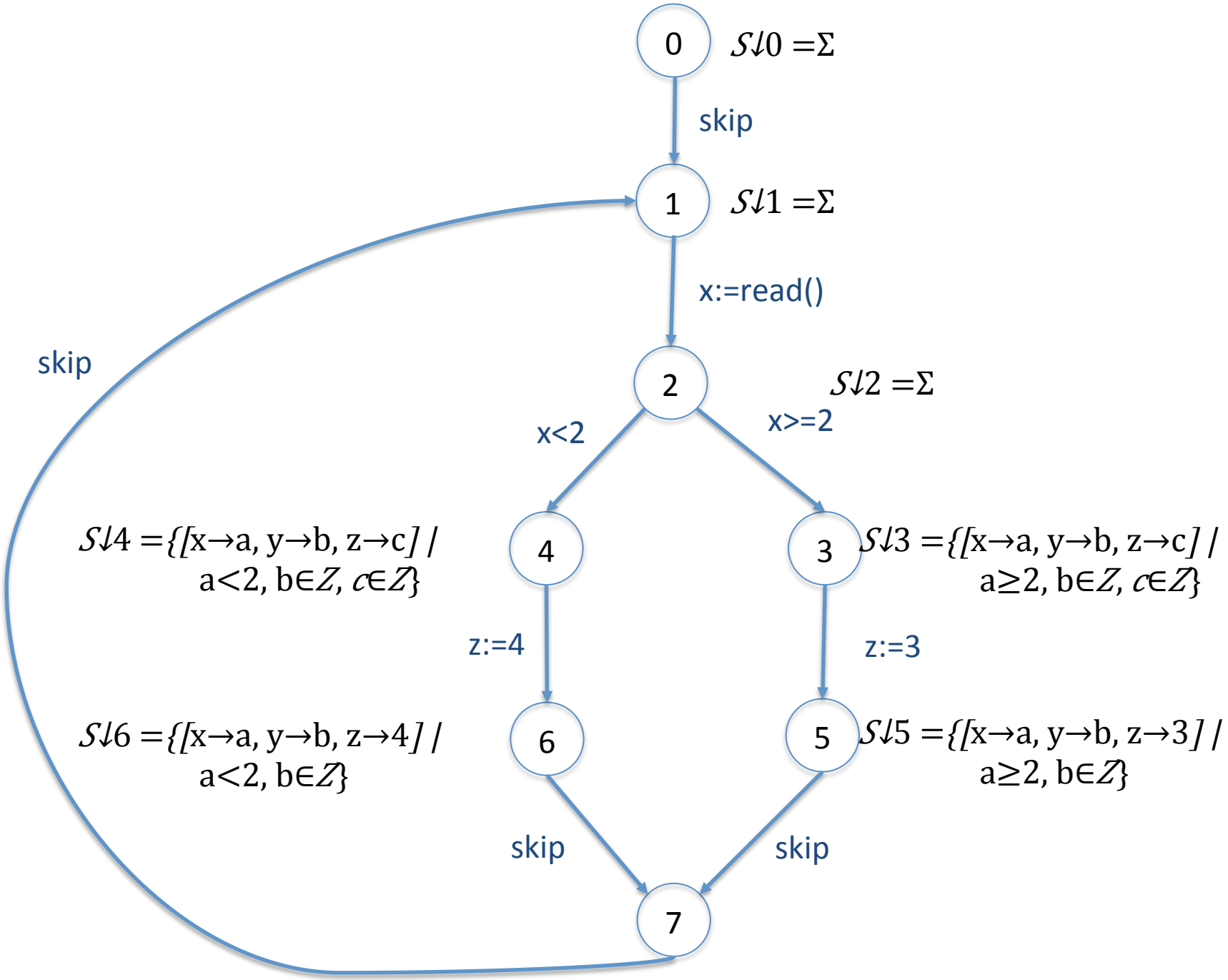

0
$S\downarrow 0 = \Sigma$
skip
1
x:=read()
2
x<2
x>=2
4
3
skip
z:=4
z:=3
6
5
skip
skip
7

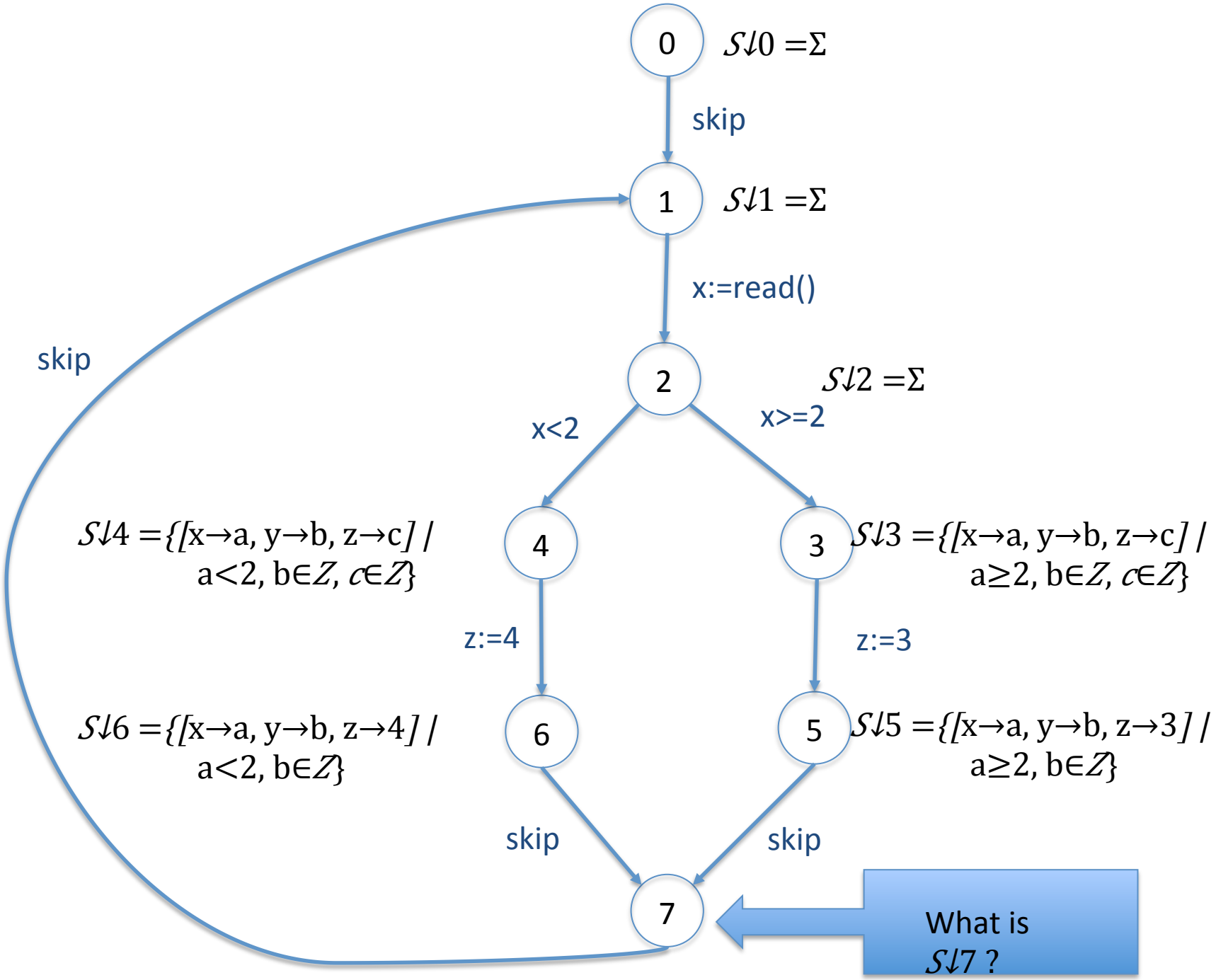

0
$\mathcal{S}\downarrow 0 = \Sigma$
skip
1
$\mathcal{S}\downarrow 1 = \Sigma$
x:=read()
2
x<2
x>=2
4
3
z:=4
z:=3
6
5
skip
skip
7
skip

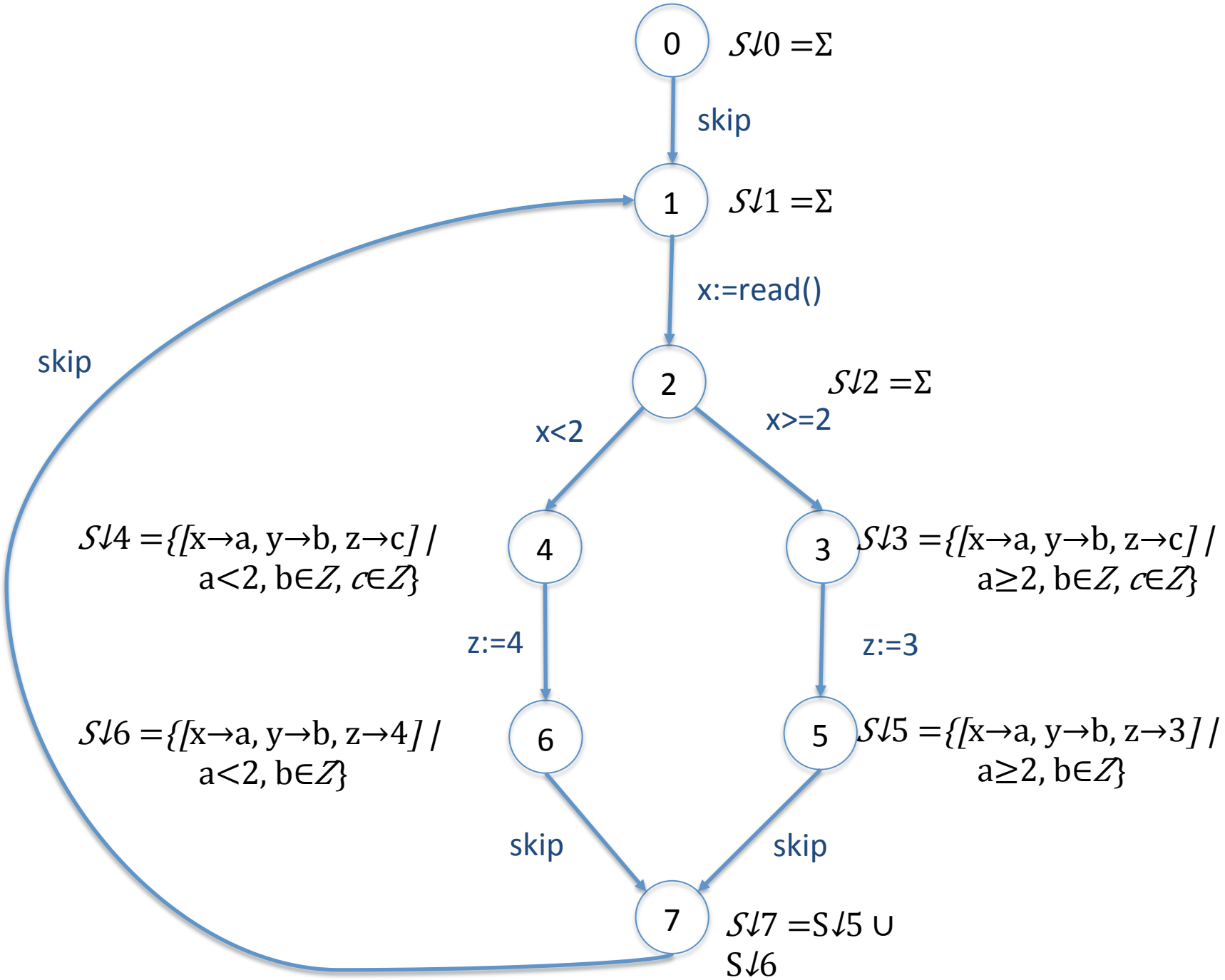

0
$\mathcal{S}\downarrow 0 = \Sigma$
skip
1
$\mathcal{S}\downarrow 1 = \Sigma$
x:=read()
2
$\mathcal{S}\downarrow 2 = \Sigma$
x<2
x>=2
4
3
skip
z:=4
z:=3
6
5
skip
skip
7

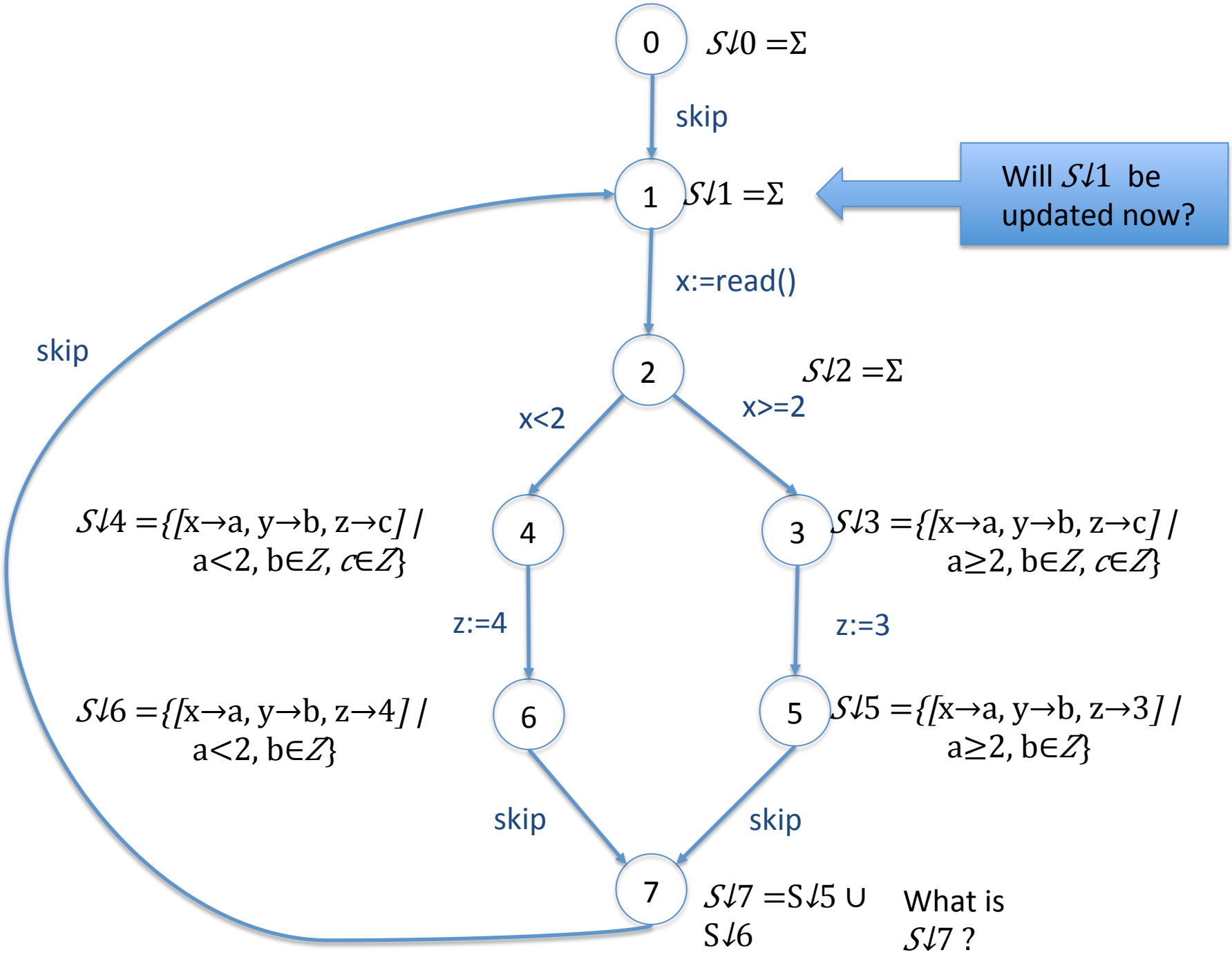

0
$\mathcal{S}\downarrow 0 = \Sigma$
skip
1
$\mathcal{S}\downarrow 1 = \Sigma$
x:=read()
2
$\mathcal{S}\downarrow 2 = \Sigma$
x<2
x>=2
4
3
$\mathcal{S}\downarrow 3 = \{[x \to a, y \to b, z \to c] \mid a \geq 2, b \in Z, c \in Z\}$
skip
z:=4
z:=3
6
5
skip
skip
7


0
$\mathcal{S}\downarrow 0 = \Sigma$
skip
1
$\mathcal{S}\downarrow 1 = \Sigma$
x:=read()
2
$\mathcal{S}\downarrow 2 = \Sigma$
x<2
x>=2
4
3
$\mathcal{S}\downarrow 3 = \{[x \to a, y \to b, z \to c] \mid a \geq 2, b \in Z, c \in Z\}$
skip
z:=4
z:=3
6
5
$\mathcal{S}\downarrow 5 = \{[x \to a, y \to b, z \to 3] \mid a \geq 2, b \in Z\}$
skip
skip
7


0
$S\downarrow 0 = \Sigma$
skip
1
$S\downarrow 1 = \Sigma$
x:=read()
skip
2
$S\downarrow 2 = \Sigma$
x<2
x>=2
$S\downarrow 4 = \{[x \to a, y \to b, z \to c] \mid a<2, b \in Z, c \in Z\}$
4
3
$S\downarrow 3 = \{[x \to a, y \to b, z \to c] \mid a \geq 2, b \in Z, c \in Z\}$
z:=4
z:=3
$S\downarrow 6 = \{[x \to a, y \to b, z \to 4] \mid a<2, b \in Z\}$
6
5
$S\downarrow 5 = \{[x \to a, y \to b, z \to 3] \mid a \geq 2, b \in Z\}$
skip
skip
7


0
$S\downarrow 0 = \Sigma$
skip
1
$S\downarrow 1 = \Sigma$
x:=read()
skip
2
$S\downarrow 2 = \Sigma$
x<2
x>=2
$S\downarrow 4 = \{[x \to a, y \to b, z \to c] \mid a<2, b \in Z, c \in Z\}$
4
3
$S\downarrow 3 = \{[x \to a, y \to b, z \to c] \mid a \geq 2, b \in Z, c \in Z\}$
z:=4
z:=3
$S\downarrow 6 = \{[x \to a, y \to b, z \to 4] \mid a<2, b \in Z\}$
6
5
$S\downarrow 5 = \{[x \to a, y \to b, z \to 3] \mid a \geq 2, b \in Z\}$
skip
skip
7
What is $S\downarrow 7$?


0
$S\downarrow 0 = \Sigma$
skip
1
$S\downarrow 1 = \Sigma$
x:=read()
skip
2
$S\downarrow 2 = \Sigma$
x<2
x>=2
$S\downarrow 4 = \{[x \to a, y \to b, z \to c] \mid a<2, b \in Z, c \in Z\}$
4
3
$S\downarrow 3 = \{[x \to a, y \to b, z \to c] \mid a \geq 2, b \in Z, c \in Z\}$
z:=4
z:=3
$S\downarrow 6 = \{[x \to a, y \to b, z \to 4] \mid a<2, b \in Z\}$
6
5
$S\downarrow 5 = \{[x \to a, y \to b, z \to 3] \mid a \geq 2, b \in Z\}$
skip
skip
7
$S\downarrow 7 = S\downarrow 5 \cup S\downarrow 6$


0
$S\downarrow 0 = \Sigma$
skip
1
$S\downarrow 1 = \Sigma$
Will $S\downarrow 1$ be updated now?
x:=read()
skip
2
$S\downarrow 2 = \Sigma$
x<2
x>=2
$S\downarrow 4 = \{[x \to a, y \to b, z \to c] \mid a<2, b \in Z, c \in Z\}$
4
3
$S\downarrow 3 = \{[x \to a, y \to b, z \to c] \mid a \geq 2, b \in Z, c \in Z\}$
z:=4
z:=3
$S\downarrow 6 = \{[x \to a, y \to b, z \to 4] \mid a<2, b \in Z\}$
6
5
$S\downarrow 5 = \{[x \to a, y \to b, z \to 3] \mid a \geq 2, b \in Z\}$
skip
skip
7
$S\downarrow 7 = S\downarrow 5 \cup S\downarrow 6$
What is $S\downarrow 7$?

Concrete Semantics

- When we go on from node 7 to 1 and update $S\downarrow 1$:

$$S\downarrow 1 := S\downarrow 1 \cup S\downarrow 7 = \Sigma \cup S\downarrow 7 = \Sigma$$

- $S\downarrow 1$ remained the same $\rightarrow$ we can stop the analysis
- So we computed concrete semantics of the program: the real possible states at each node
- We can infer, for example, that in node 7, the value of z is either 3 or 4.

Concrete Semantics

- The problem: in realistic programs:
  - The representation of the unions of states can explode
  - The analysis might not stop (we always discover new information)

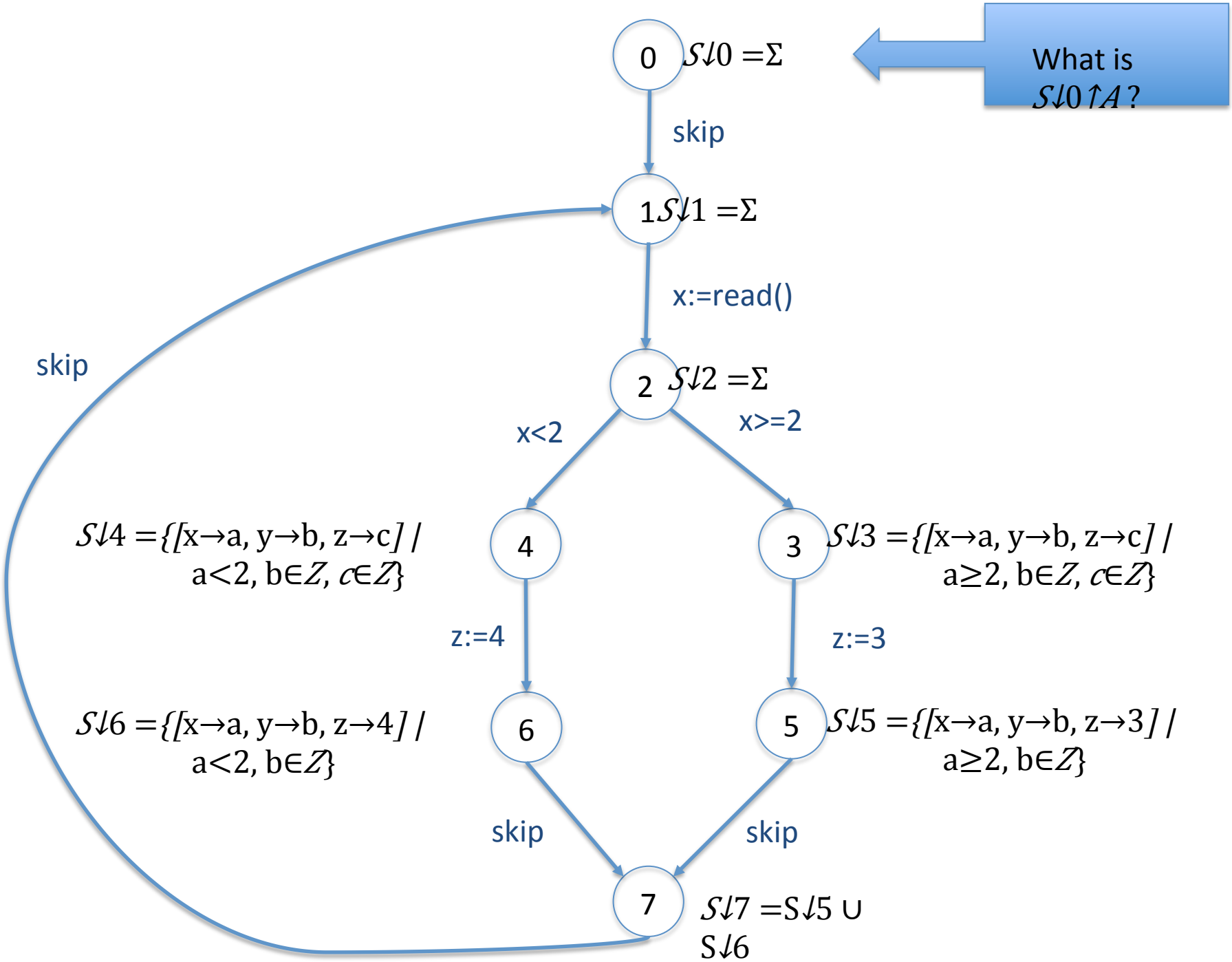
Abstract Semantics

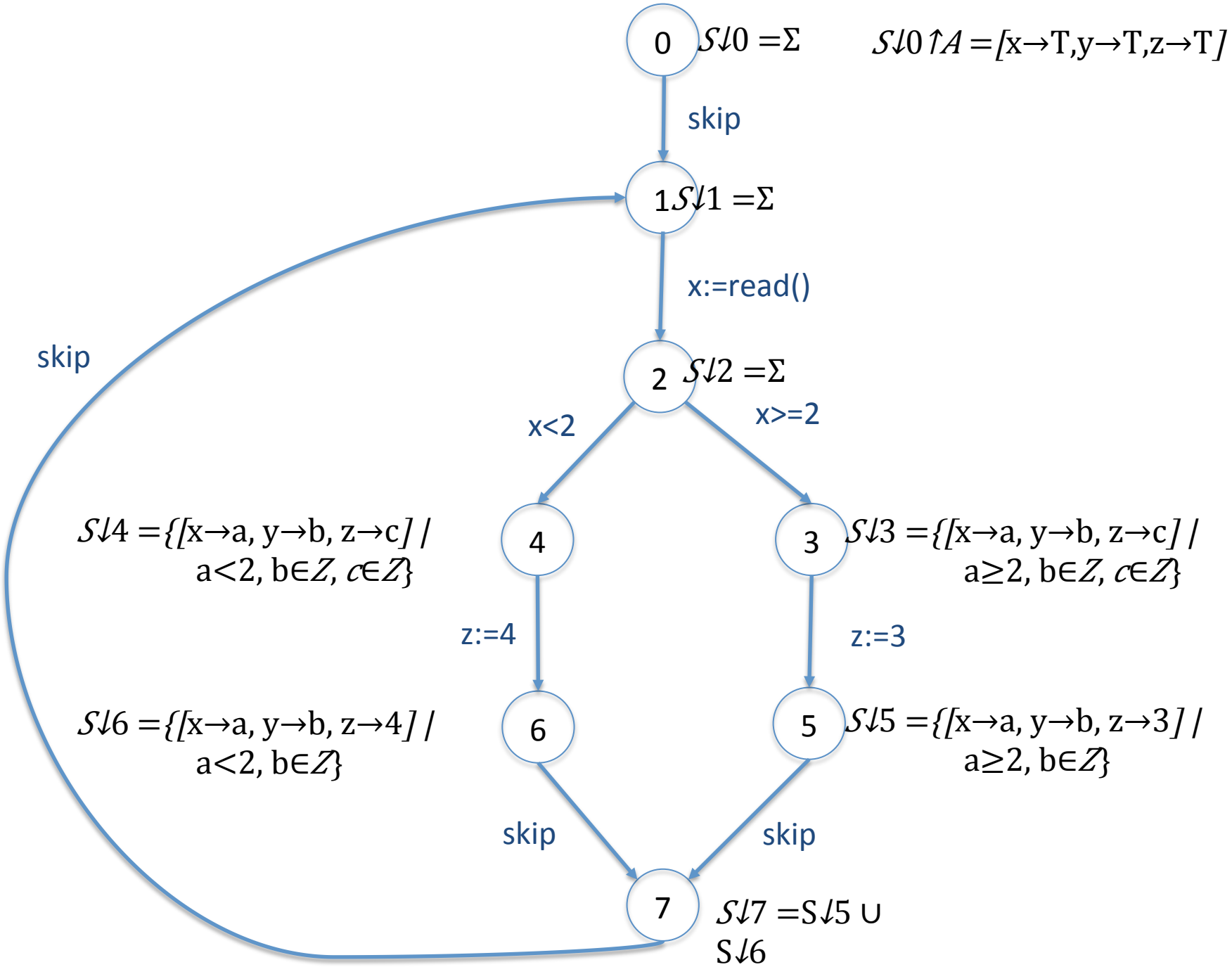
- Solution: we will use abstraction:
  - At each node: Instead of S, use $S\uparrow A \supseteq S$
  - By this we lose information
  - But we will be able to represent $S\uparrow A$
  - And our analysis will necessarily stop
  - If we prove a property for all $s \in S\uparrow A$, then this property holds for all $s \in S$

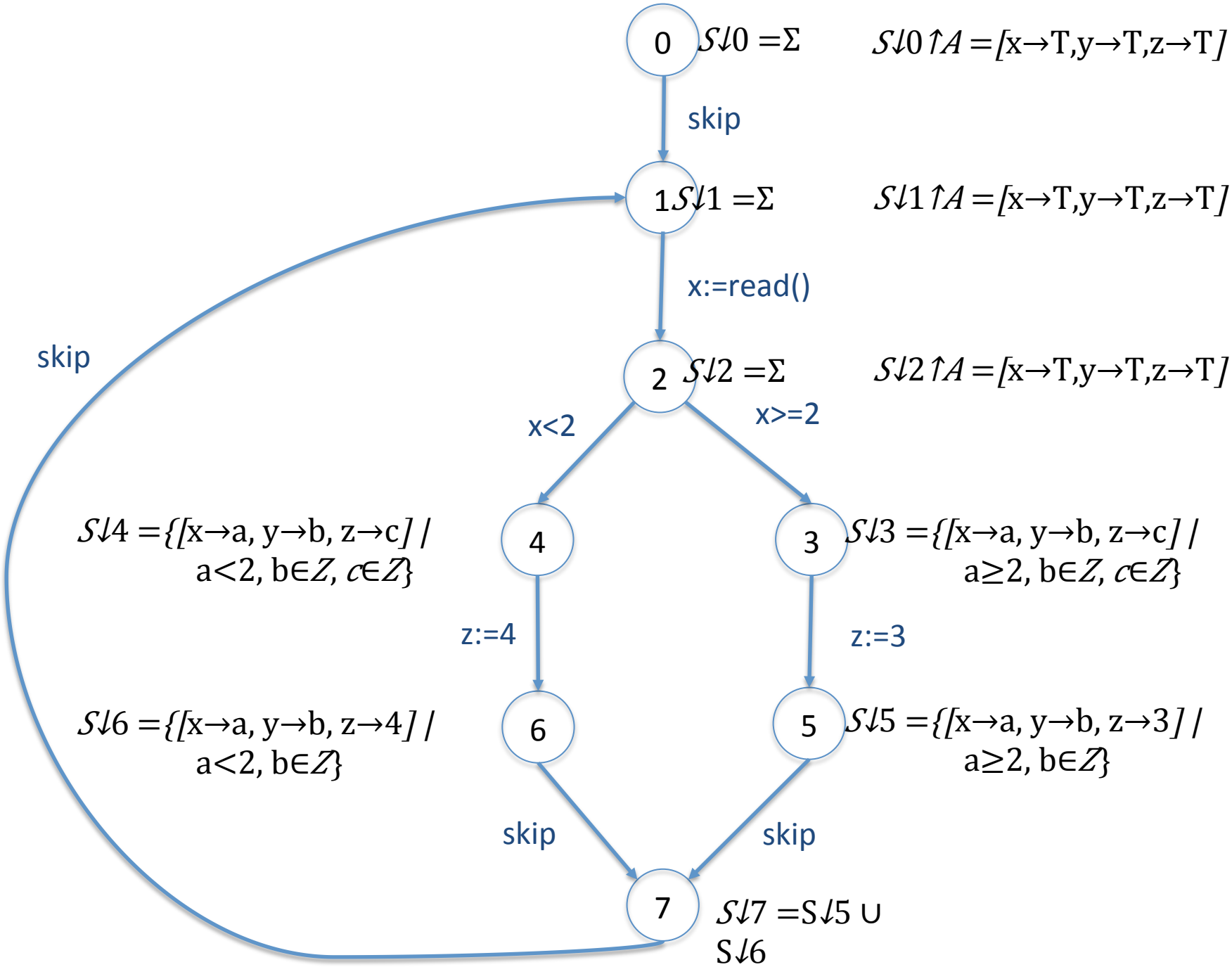
Abstract Semantics

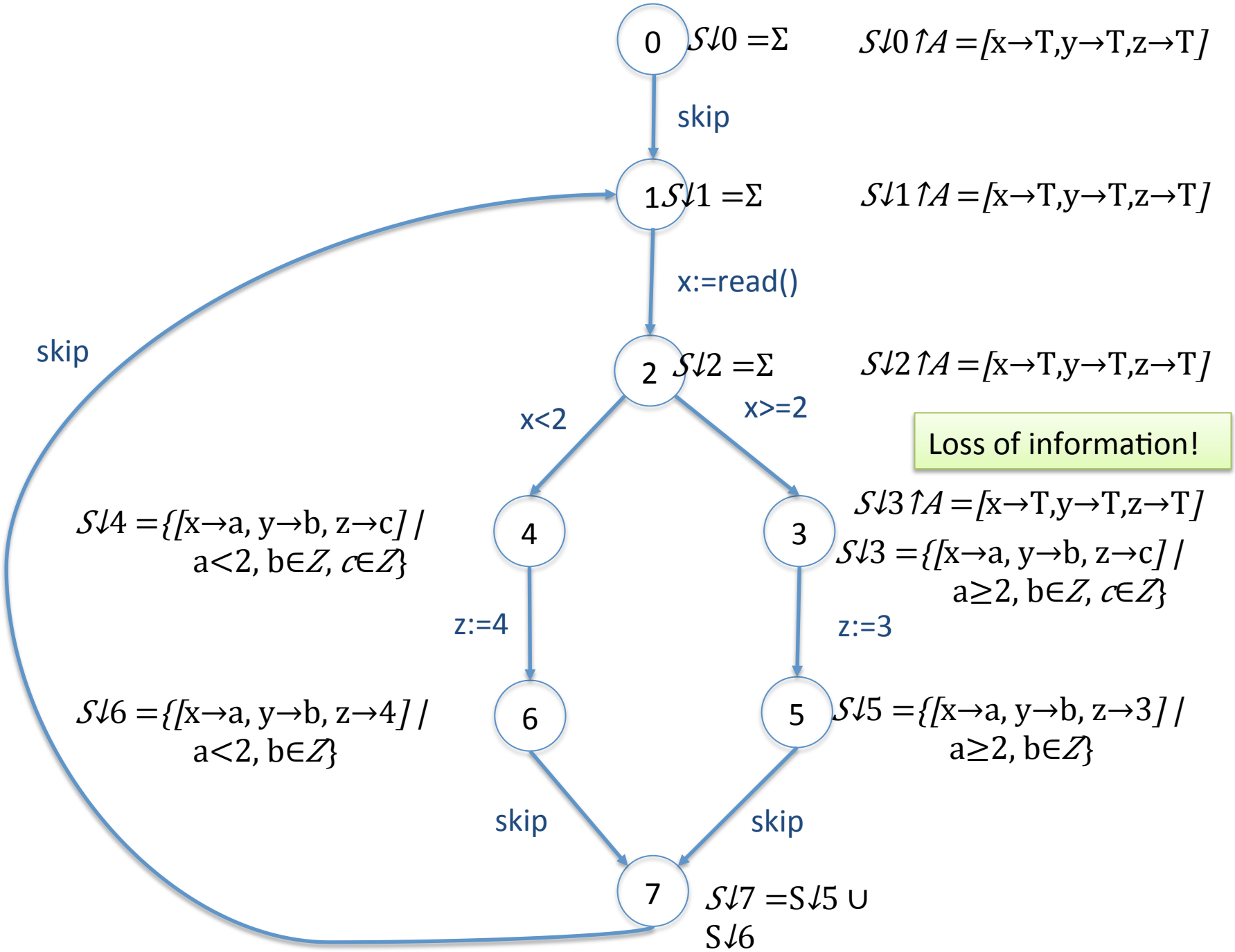
- Let's define the following abstract domain:
- $S\uparrow A : Var \rightarrow Z\uparrow T$
- $Z\uparrow T = Z \cup \{T\}$
- $S\uparrow A\ (v) = T$ (top) denotes that the variable v can have any value
- $S\uparrow A$ is an abstract mapping which represents a set of concrete states
  - E.g.: $S\uparrow A = [x \rightarrow T, y \rightarrow T, z \rightarrow 3]$ represents:

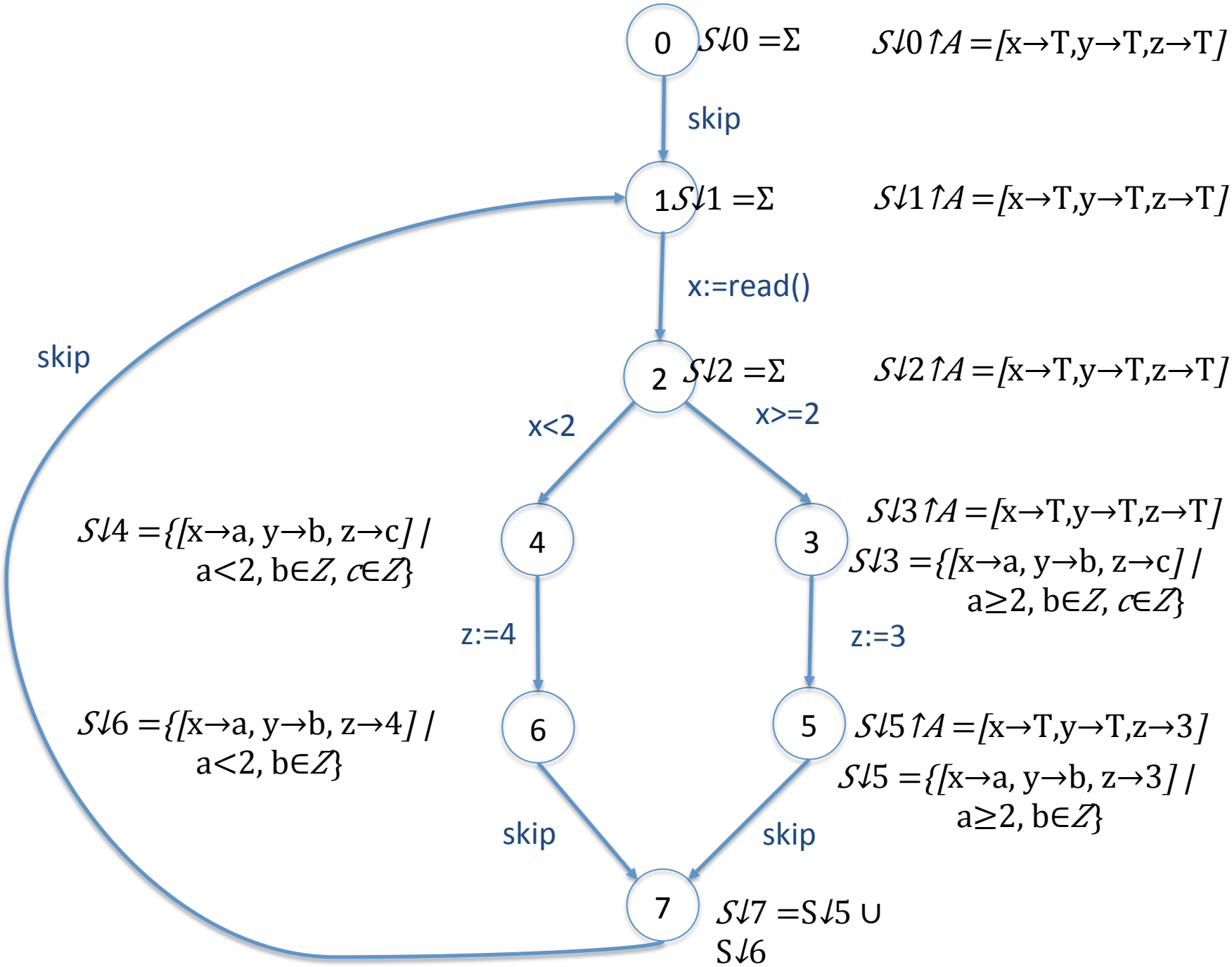
$\{[x \rightarrow a, y \rightarrow b, z \rightarrow 3] \mid a \in Z, b \in Z\}$

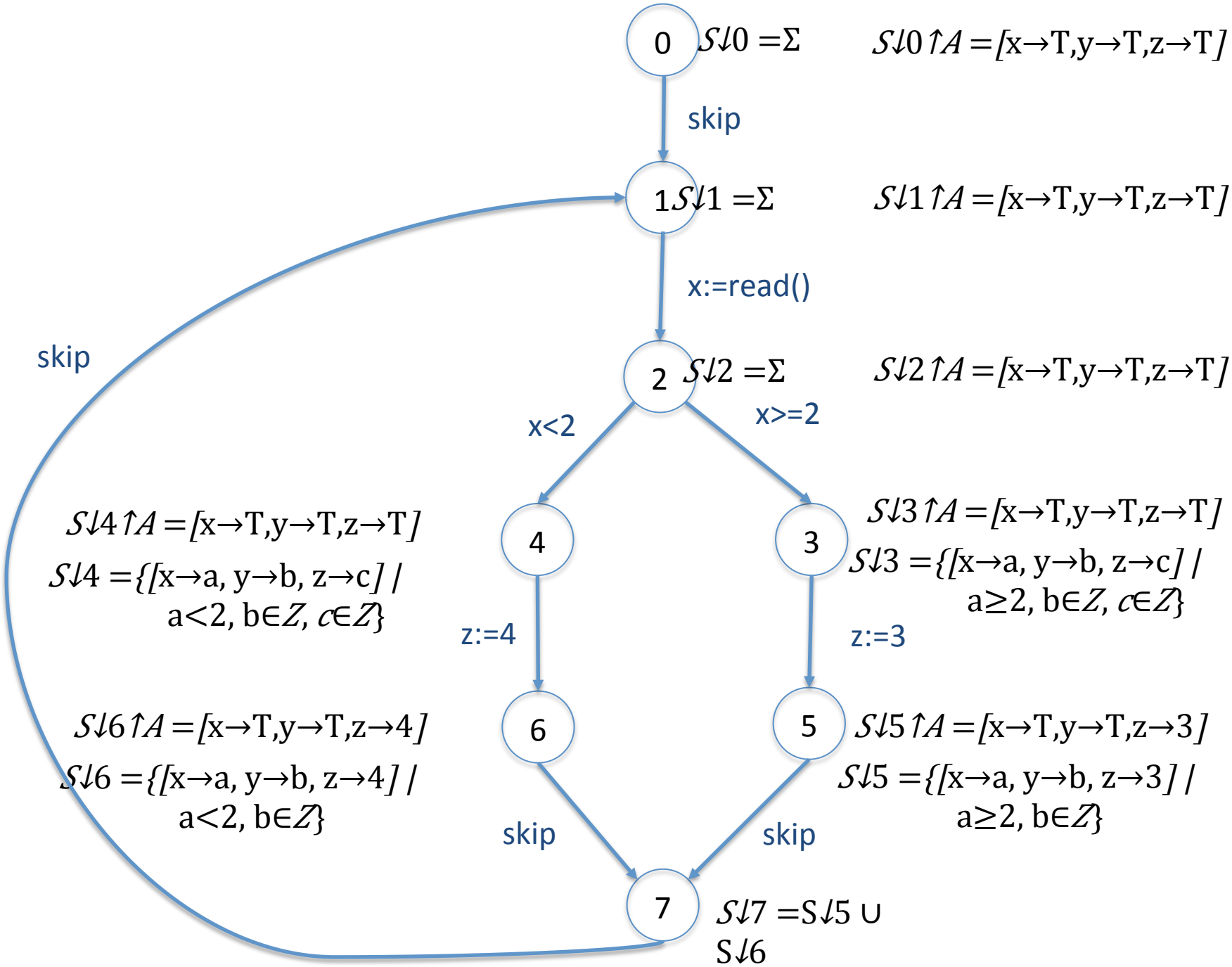

What is $S\downarrow 0 \uparrow A$?
0
$S\downarrow 0 = \Sigma$
skip
1
$S\downarrow 1 = \Sigma$
x:=read()
2
$S\downarrow 2 = \Sigma$
x<2
x>=2
4
$S\downarrow 4 = \{[x \to a, y \to b, z \to c] \mid a<2, b \in Z, c \in Z\}$
3
$S\downarrow 3 = \{[x \to a, y \to b, z \to c] \mid a \geq 2, b \in Z, c \in Z\}$
z:=4
z:=3
6
$S\downarrow 6 = \{[x \to a, y \to b, z \to 4] \mid a<2, b \in Z\}$
5
$S\downarrow 5 = \{[x \to a, y \to b, z \to 3] \mid a \geq 2, b \in Z\}$
skip
skip
7
$S\downarrow 7 = S\downarrow 5 \cup S\downarrow 6$
skip

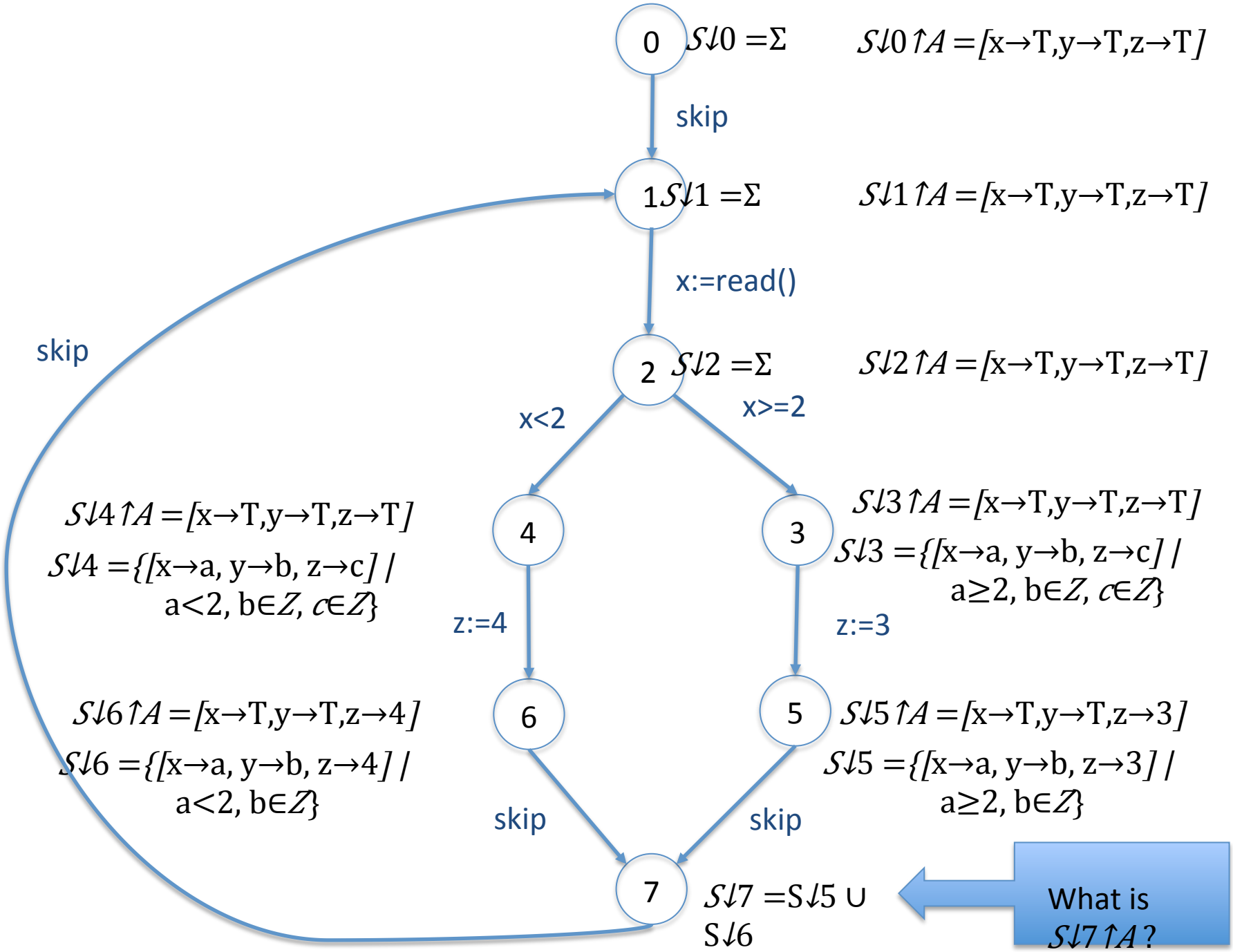

0
$S\downarrow 0 = \Sigma$
$S\downarrow 0 \uparrow A = [x \to T, y \to T, z \to T]$
skip
1
$S\downarrow 1 = \Sigma$
x:=read()
2
$S\downarrow 2 = \Sigma$
x<2
x>=2
skip
4
$S\downarrow 4 = \{[x \to a, y \to b, z \to c] \mid a<2, b \in Z, c \in Z\}$
3
$S\downarrow 3 = \{[x \to a, y \to b, z \to c] \mid a \geq 2, b \in Z, c \in Z\}$
z:=4
z:=3
6
$S\downarrow 6 = \{[x \to a, y \to b, z \to 4] \mid a<2, b \in Z\}$
5
$S\downarrow 5 = \{[x \to a, y \to b, z \to 3] \mid a \geq 2, b \in Z\}$
skip
skip
7
$S\downarrow 7 = S\downarrow 5 \cup S\downarrow 6$

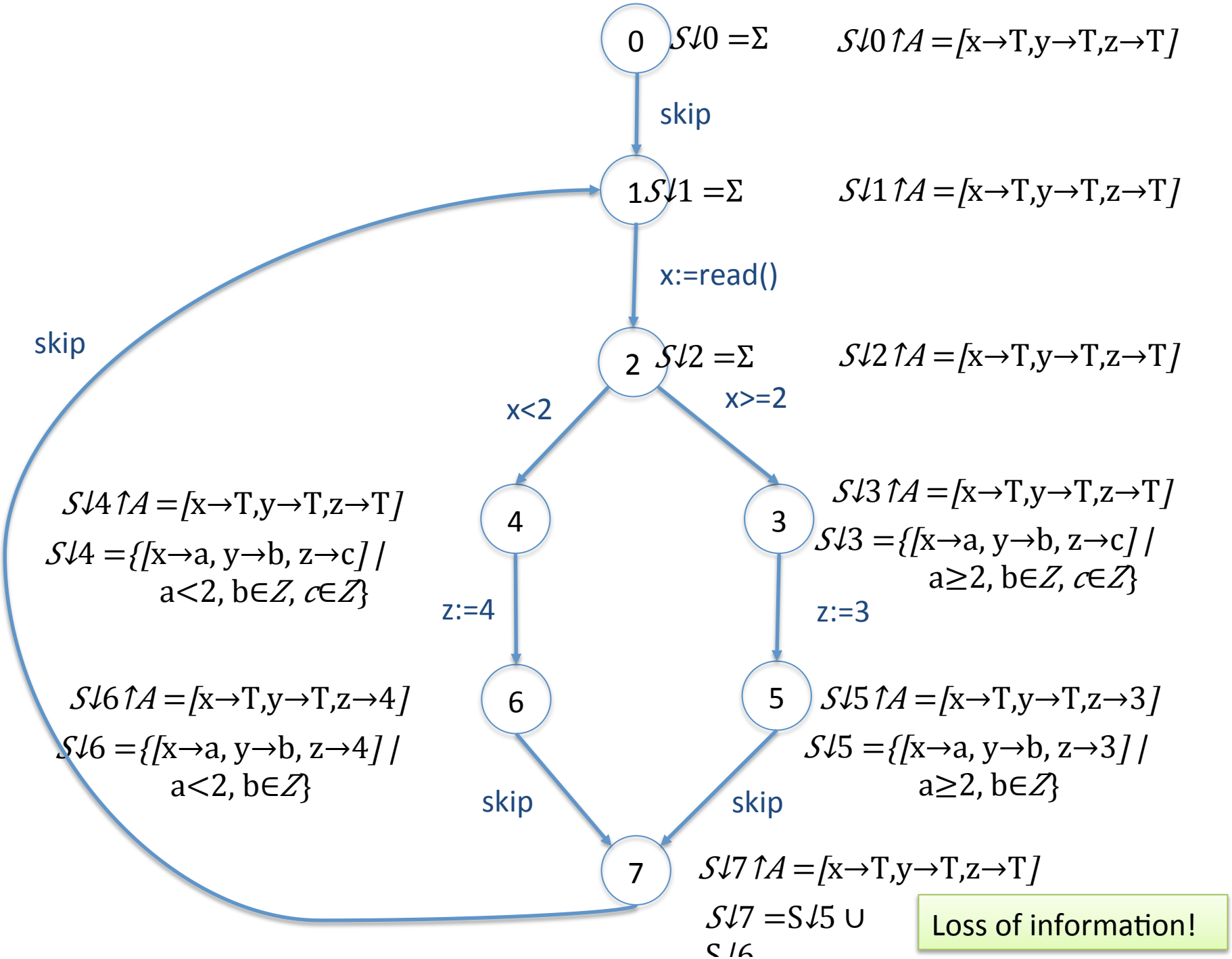

0
$S\downarrow 0 = \Sigma$
$S\downarrow 0 \uparrow A = [x \to T, y \to T, z \to T]$
skip
1
$S\downarrow 1 = \Sigma$
$S\downarrow 1 \uparrow A = [x \to T, y \to T, z \to T]$
x:=read()
skip
2
$S\downarrow 2 = \Sigma$
$S\downarrow 2 \uparrow A = [x \to T, y \to T, z \to T]$
x<2
x>=2
4
$S\downarrow 4 = \{[x \to a, y \to b, z \to c] \mid a<2, b \in Z, c \in Z\}$
3
$S\downarrow 3 = \{[x \to a, y \to b, z \to c] \mid a \geq 2, b \in Z, c \in Z\}$
z:=4
z:=3
6
$S\downarrow 6 = \{[x \to a, y \to b, z \to 4] \mid a<2, b \in Z\}$
5
$S\downarrow 5 = \{[x \to a, y \to b, z \to 3] \mid a \geq 2, b \in Z\}$
skip
skip
7
$S\downarrow 7 = S\downarrow 5 \cup S\downarrow 6$


0
$S\downarrow 0 = \Sigma$
$S\downarrow 0 \uparrow A = [x \to T, y \to T, z \to T]$
skip
1
$S\downarrow 1 = \Sigma$
$S\downarrow 1 \uparrow A = [x \to T, y \to T, z \to T]$
x:=read()
skip
2
$S\downarrow 2 = \Sigma$
$S\downarrow 2 \uparrow A = [x \to T, y \to T, z \to T]$
x<2
x>=2
Loss of information!
4
$S\downarrow 4 = \{[x \to a, y \to b, z \to c] \mid a < 2, b \in Z, c \in Z\}$
3
$S\downarrow 3 \uparrow A = [x \to T, y \to T, z \to T]$
$S\downarrow 3 = \{[x \to a, y \to b, z \to c] \mid a \geq 2, b \in Z, c \in Z\}$
z:=4
z:=3
6
$S\downarrow 6 = \{[x \to a, y \to b, z \to 4] \mid a < 2, b \in Z\}$
5
$S\downarrow 5 = \{[x \to a, y \to b, z \to 3] \mid a \geq 2, b \in Z\}$
skip
skip
7
$S\downarrow 7 = S\downarrow 5 \cup S\downarrow 6$

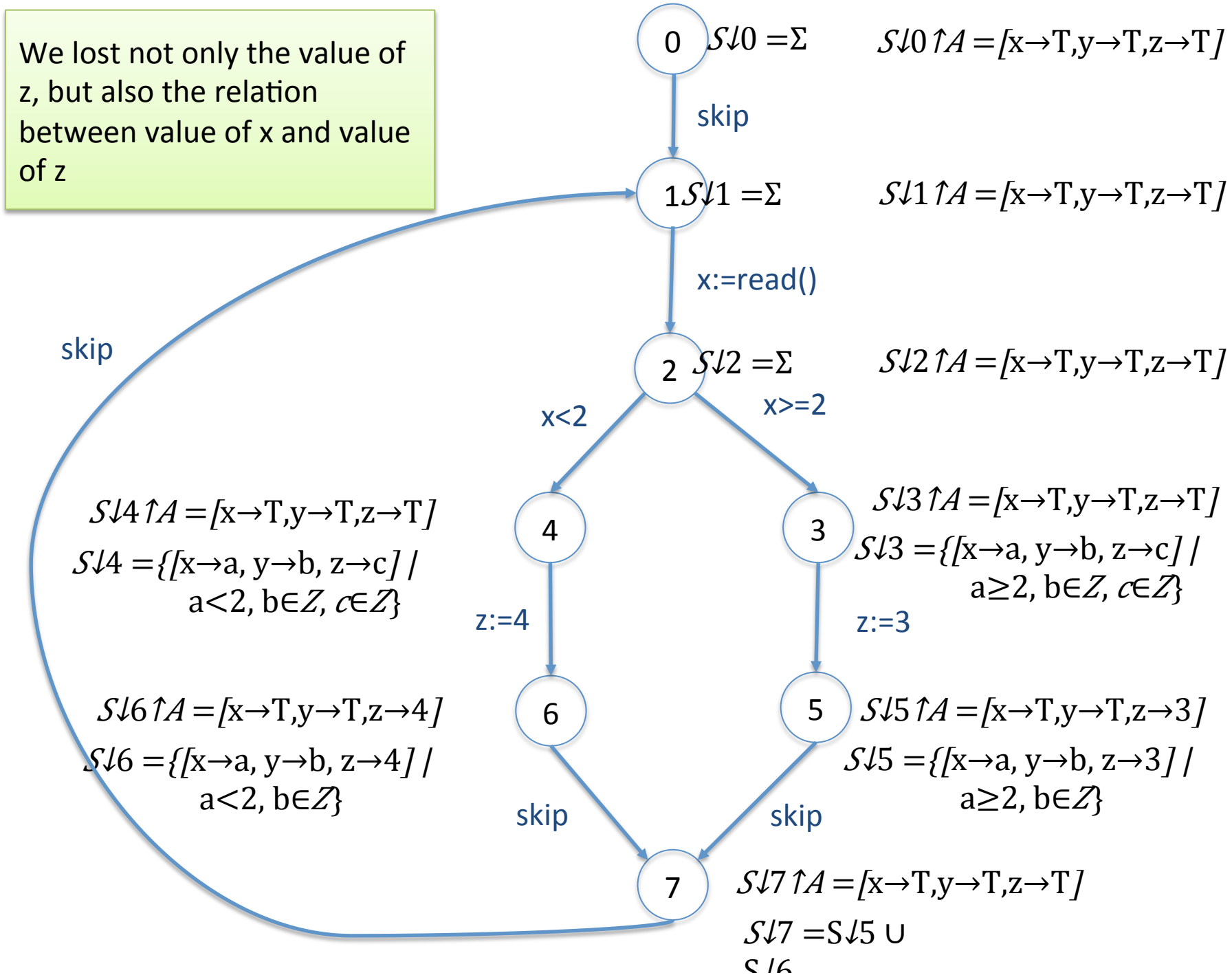

0
$S\downarrow 0 = \Sigma$
$S\downarrow 0 \uparrow A = [x \to T, y \to T, z \to T]$
skip
1
$S\downarrow 1 = \Sigma$
$S\downarrow 1 \uparrow A = [x \to T, y \to T, z \to T]$
x:=read()
skip
2
$S\downarrow 2 = \Sigma$
$S\downarrow 2 \uparrow A = [x \to T, y \to T, z \to T]$
x<2
x>=2
4
$S\downarrow 4 = \{[x \to a, y \to b, z \to c] \mid a < 2, b \in Z, c \in Z\}$
3
$S\downarrow 3 \uparrow A = [x \to T, y \to T, z \to T]$
$S\downarrow 3 = \{[x \to a, y \to b, z \to c] \mid a \geq 2, b \in Z, c \in Z\}$
z:=4
z:=3
6
$S\downarrow 6 = \{[x \to a, y \to b, z \to 4] \mid a < 2, b \in Z\}$
5
$S\downarrow 5 \uparrow A = [x \to T, y \to T, z \to 3]$
$S\downarrow 5 = \{[x \to a, y \to b, z \to 3] \mid a \geq 2, b \in Z\}$
skip
skip
7
$S\downarrow 7 = S\downarrow 5 \cup S\downarrow 6$

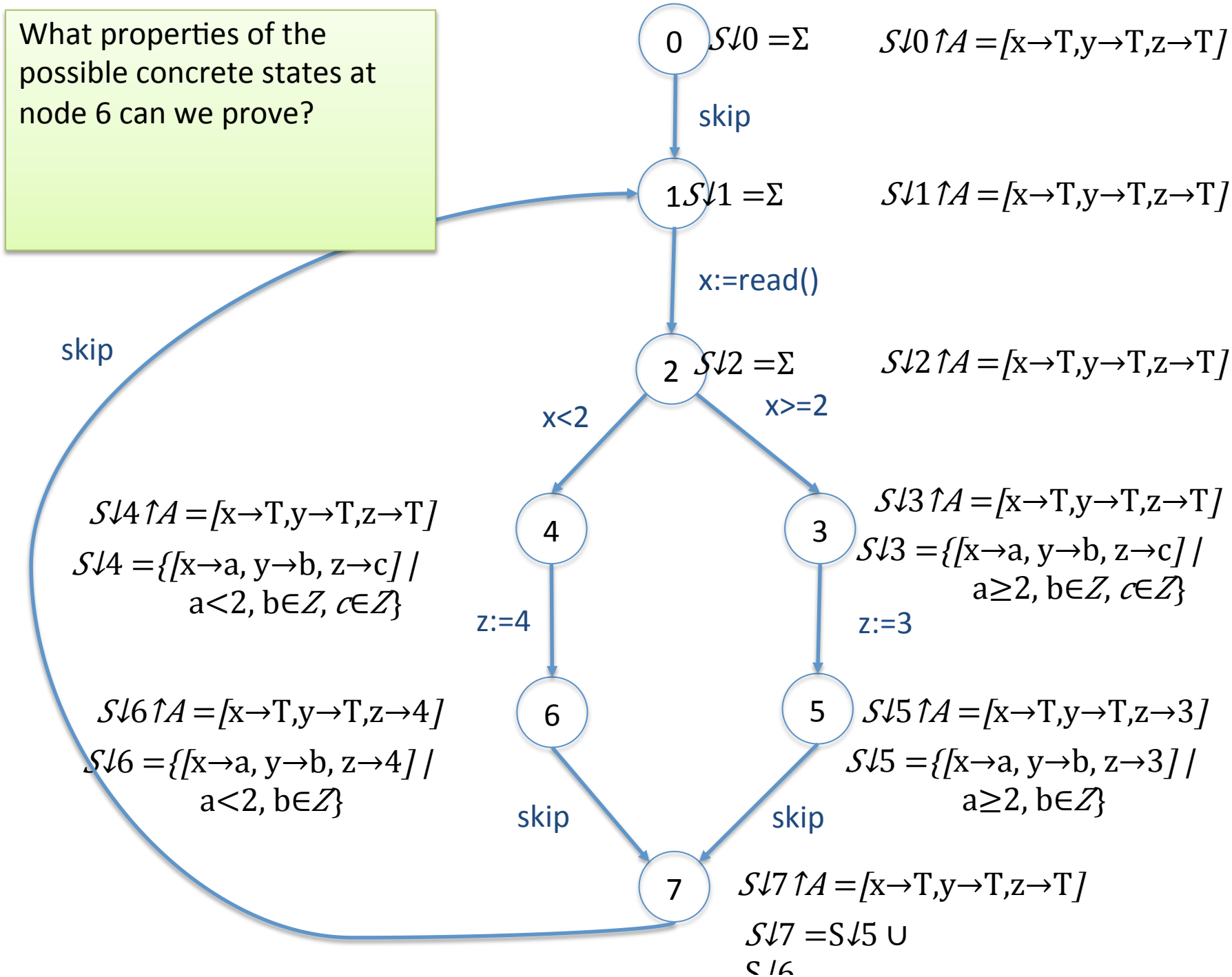

0
$S\downarrow 0 = \Sigma$
$S\downarrow 0 \uparrow A = [x \to T, y \to T, z \to T]$
skip
1
$S\downarrow 1 = \Sigma$
$S\downarrow 1 \uparrow A = [x \to T, y \to T, z \to T]$
x:=read()
skip
2
$S\downarrow 2 = \Sigma$
$S\downarrow 2 \uparrow A = [x \to T, y \to T, z \to T]$
x<2
x>=2
4
$S\downarrow 4 \uparrow A = [x \to T, y \to T, z \to T]$
$S\downarrow 4 = \{[x \to a, y \to b, z \to c] \mid a < 2, b \in Z, c \in Z\}$
3
$S\downarrow 3 \uparrow A = [x \to T, y \to T, z \to T]$
$S\downarrow 3 = \{[x \to a, y \to b, z \to c] \mid a \geq 2, b \in Z, c \in Z\}$
z:=4
z:=3
6
$S\downarrow 6 \uparrow A = [x \to T, y \to T, z \to 4]$
$S\downarrow 6 = \{[x \to a, y \to b, z \to 4] \mid a < 2, b \in Z\}$
5
$S\downarrow 5 \uparrow A = [x \to T, y \to T, z \to 3]$
$S\downarrow 5 = \{[x \to a, y \to b, z \to 3] \mid a \geq 2, b \in Z\}$
skip
skip
7
$S\downarrow 7 = S\downarrow 5 \cup S\downarrow 6$

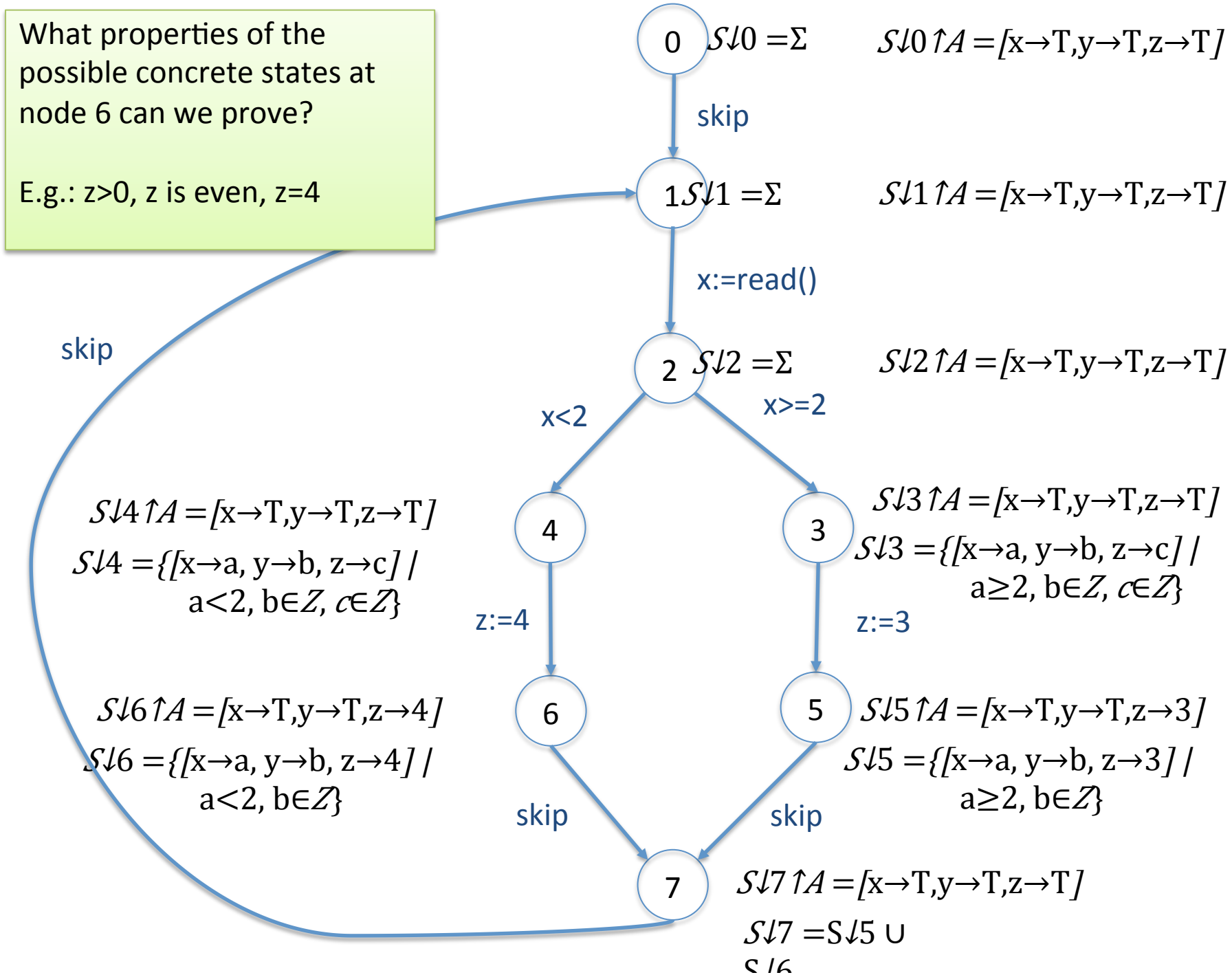

0
$S\downarrow 0 = \Sigma$
$S\downarrow 0 \uparrow A = [x \to T, y \to T, z \to T]$
skip
1
$S\downarrow 1 = \Sigma$
$S\downarrow 1 \uparrow A = [x \to T, y \to T, z \to T]$
x:=read()
2
$S\downarrow 2 = \Sigma$
$S\downarrow 2 \uparrow A = [x \to T, y \to T, z \to T]$
x<2
x>=2
skip
4
$S\downarrow 4 \uparrow A = [x \to T, y \to T, z \to T]$
$S\downarrow 4 = \{[x \to a, y \to b, z \to c] \mid a < 2, b \in Z, c \in Z\}$
3
$S\downarrow 3 \uparrow A = [x \to T, y \to T, z \to T]$
$S\downarrow 3 = \{[x \to a, y \to b, z \to c] \mid a \geq 2, b \in Z, c \in Z\}$
z:=4
z:=3
6
$S\downarrow 6 \uparrow A = [x \to T, y \to T, z \to 4]$
$S\downarrow 6 = \{[x \to a, y \to b, z \to 4] \mid a < 2, b \in Z\}$
5
$S\downarrow 5 \uparrow A = [x \to T, y \to T, z \to 3]$
$S\downarrow 5 = \{[x \to a, y \to b, z \to 3] \mid a \geq 2, b \in Z\}$
skip
skip
7
$S\downarrow 7 = S\downarrow 5 \cup S\downarrow 6$
What is $S\downarrow 7 \uparrow A$?

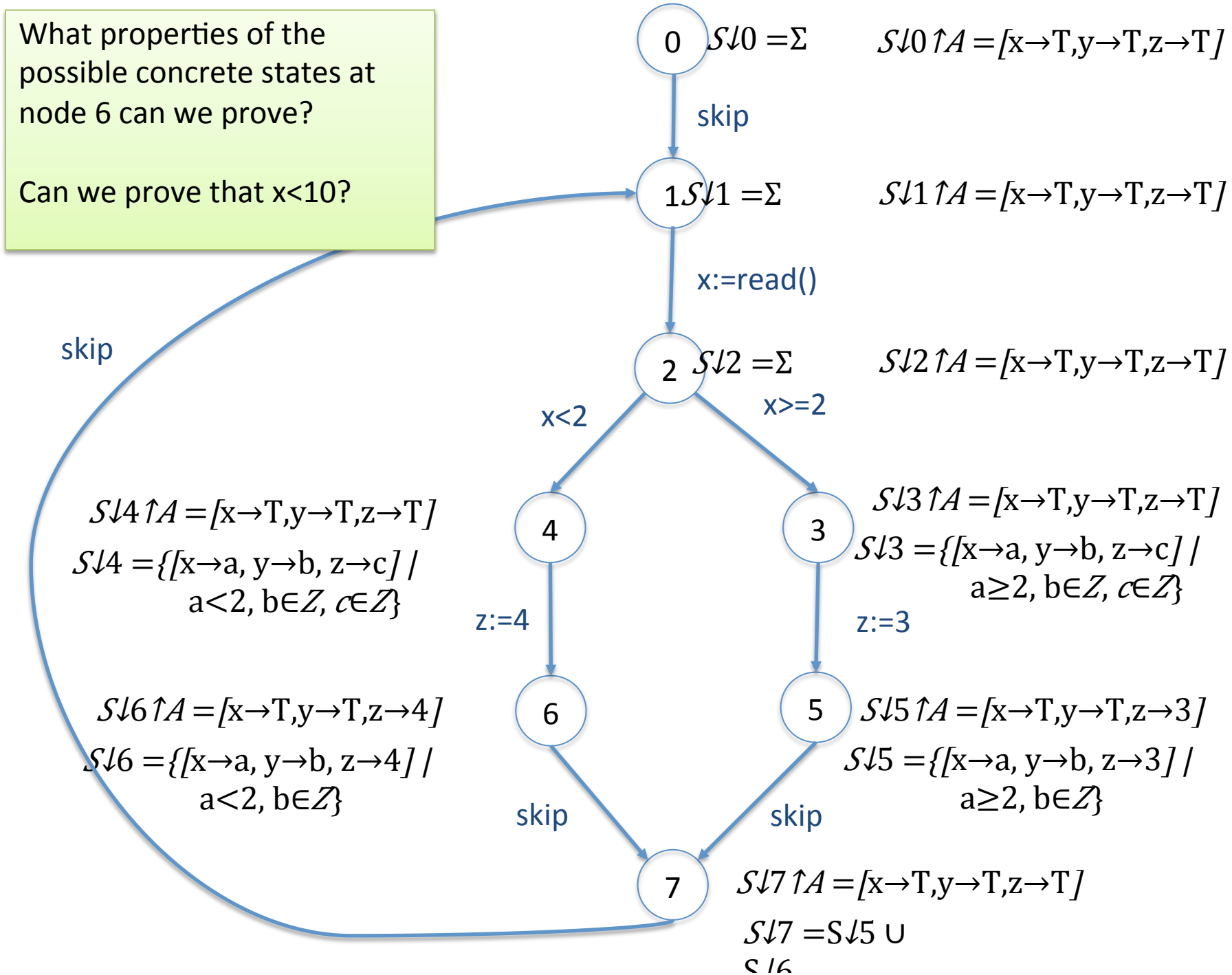

0
$S\downarrow 0 = \Sigma$
$S\downarrow 0 \uparrow A = [x \to T, y \to T, z \to T]$
skip
1
$S\downarrow 1 = \Sigma$
$S\downarrow 1 \uparrow A = [x \to T, y \to T, z \to T]$
x:=read()
2
$S\downarrow 2 = \Sigma$
$S\downarrow 2 \uparrow A = [x \to T, y \to T, z \to T]$
x<2
x>=2
skip
4
$S\downarrow 4 \uparrow A = [x \to T, y \to T, z \to T]$
$S\downarrow 4 = \{[x \to a, y \to b, z \to c] \mid a < 2, b \in Z, c \in Z\}$
3
$S\downarrow 3 \uparrow A = [x \to T, y \to T, z \to T]$
$S\downarrow 3 = \{[x \to a, y \to b, z \to c] \mid a \geq 2, b \in Z, c \in Z\}$
z:=4
z:=3
6
$S\downarrow 6 \uparrow A = [x \to T, y \to T, z \to 4]$
$S\downarrow 6 = \{[x \to a, y \to b, z \to 4] \mid a < 2, b \in Z\}$
5
$S\downarrow 5 \uparrow A = [x \to T, y \to T, z \to 3]$
$S\downarrow 5 = \{[x \to a, y \to b, z \to 3] \mid a \geq 2, b \in Z\}$
skip
skip
7
$S\downarrow 7 \uparrow A = [x \to T, y \to T, z \to T]$
$S\downarrow 7 = S\downarrow 5 \cup$
Loss of information!

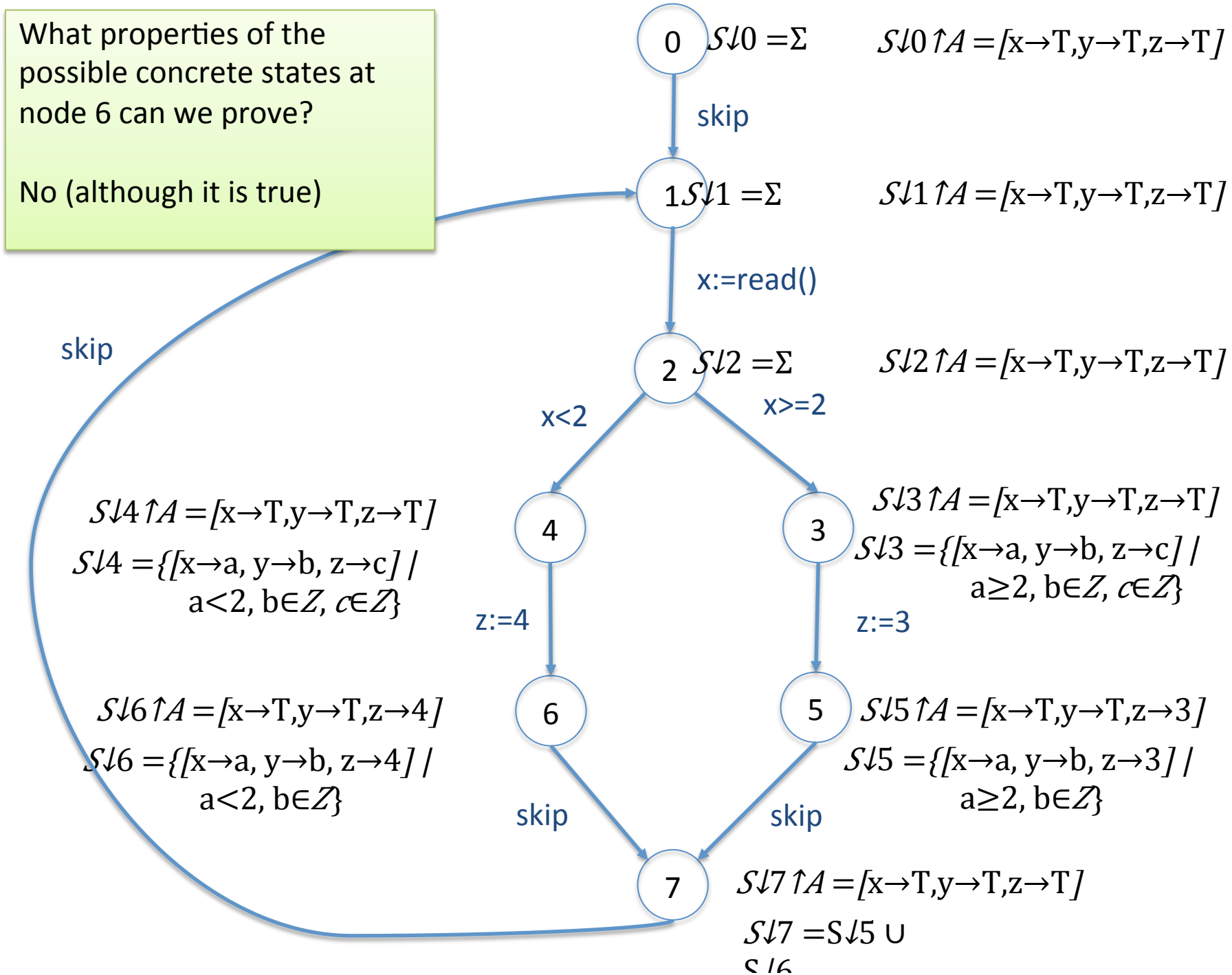
We lost not only the value of z, but also the relation between value of x and value of z




What properties of the possible concrete states at node 6 can we prove?
0
$S\downarrow 0 = \Sigma$
$S\downarrow 0 \uparrow A = [x \to T, y \to T, z \to T]$
skip
1
$S\downarrow 1 = \Sigma$
$S\downarrow 1 \uparrow A = [x \to T, y \to T, z \to T]$
x:=read()
2
$S\downarrow 2 = \Sigma$
$S\downarrow 2 \uparrow A = [x \to T, y \to T, z \to T]$
x<2
x>=2
4
$S\downarrow 4 \uparrow A = [x \to T, y \to T, z \to T]$
$S\downarrow 4 = \{[x \to a, y \to b, z \to c] \mid a < 2, b \in Z, c \in Z\}$
3
$S\downarrow 3 \uparrow A = [x \to T, y \to T, z \to T]$
$S\downarrow 3 = \{[x \to a, y \to b, z \to c] \mid a \geq 2, b \in Z, c \in Z\}$
z:=4
z:=3
6
$S\downarrow 6 \uparrow A = [x \to T, y \to T, z \to 4]$
$S\downarrow 6 = \{[x \to a, y \to b, z \to 4] \mid a < 2, b \in Z\}$
5
$S\downarrow 5 \uparrow A = [x \to T, y \to T, z \to 3]$
$S\downarrow 5 = \{[x \to a, y \to b, z \to 3] \mid a \geq 2, b \in Z\}$
skip
skip
skip
7
$S\downarrow 7 \uparrow A = [x \to T, y \to T, z \to T]$
$S\downarrow 7 = S\downarrow 5 \cup$
$S\downarrow 6$


What properties of the possible concrete states at node 6 can we prove?
E.g.: z>0, z is even, z=4
0
$S\downarrow 0 = \Sigma$
$S\downarrow 0 \uparrow A = [x \to T, y \to T, z \to T]$
skip
1
$S\downarrow 1 = \Sigma$
$S\downarrow 1 \uparrow A = [x \to T, y \to T, z \to T]$
x:=read()
2
$S\downarrow 2 = \Sigma$
$S\downarrow 2 \uparrow A = [x \to T, y \to T, z \to T]$
x<2
x>=2
skip
4
$S\downarrow 4 \uparrow A = [x \to T, y \to T, z \to T]$
$S\downarrow 4 = \{[x \to a, y \to b, z \to c] \mid a < 2, b \in Z, c \in Z\}$
3
$S\downarrow 3 \uparrow A = [x \to T, y \to T, z \to T]$
$S\downarrow 3 = \{[x \to a, y \to b, z \to c] \mid a \geq 2, b \in Z, c \in Z\}$
z:=4
z:=3
6
$S\downarrow 6 \uparrow A = [x \to T, y \to T, z \to 4]$
$S\downarrow 6 = \{[x \to a, y \to b, z \to 4] \mid a < 2, b \in Z\}$
5
$S\downarrow 5 \uparrow A = [x \to T, y \to T, z \to 3]$
$S\downarrow 5 = \{[x \to a, y \to b, z \to 3] \mid a \geq 2, b \in Z\}$
skip
skip
7
$S\downarrow 7 \uparrow A = [x \to T, y \to T, z \to T]$
$S\downarrow 7 = S\downarrow 5 \cup$
$S\downarrow 6$


What properties of the possible concrete states at node 6 can we prove?
Can we prove that x<10?
0
$S\downarrow 0 = \Sigma$
$S\downarrow 0 \uparrow A = [x \to T, y \to T, z \to T]$
skip
1
$S\downarrow 1 = \Sigma$
$S\downarrow 1 \uparrow A = [x \to T, y \to T, z \to T]$
x:=read()
2
$S\downarrow 2 = \Sigma$
$S\downarrow 2 \uparrow A = [x \to T, y \to T, z \to T]$
x<2
x>=2
4
$S\downarrow 4 \uparrow A = [x \to T, y \to T, z \to T]$
$S\downarrow 4 = \{[x \to a, y \to b, z \to c] \mid a < 2, b \in Z, c \in Z\}$
3
$S\downarrow 3 \uparrow A = [x \to T, y \to T, z \to T]$
$S\downarrow 3 = \{[x \to a, y \to b, z \to c] \mid a \geq 2, b \in Z, c \in Z\}$
z:=4
z:=3
6
$S\downarrow 6 \uparrow A = [x \to T, y \to T, z \to 4]$
$S\downarrow 6 = \{[x \to a, y \to b, z \to 4] \mid a < 2, b \in Z\}$
5
$S\downarrow 5 \uparrow A = [x \to T, y \to T, z \to 3]$
$S\downarrow 5 = \{[x \to a, y \to b, z \to 3] \mid a \geq 2, b \in Z\}$
skip
skip
skip
7
$S\downarrow 7 \uparrow A = [x \to T, y \to T, z \to T]$
$S\downarrow 7 = S\downarrow 5 \cup$
$S\downarrow 6$


What properties of the possible concrete states at node 6 can we prove?
No (although it is true)
0
$S\downarrow 0 = \Sigma$
$S\downarrow 0 \uparrow A = [x \to T, y \to T, z \to T]$
skip
1
$S\downarrow 1 = \Sigma$
$S\downarrow 1 \uparrow A = [x \to T, y \to T, z \to T]$
x:=read()
2
$S\downarrow 2 = \Sigma$
$S\downarrow 2 \uparrow A = [x \to T, y \to T, z \to T]$
x<2
x>=2
4
$S\downarrow 4 \uparrow A = [x \to T, y \to T, z \to T]$
$S\downarrow 4 = \{[x \to a, y \to b, z \to c] \mid a < 2, b \in Z, c \in Z\}$
3
$S\downarrow 3 \uparrow A = [x \to T, y \to T, z \to T]$
$S\downarrow 3 = \{[x \to a, y \to b, z \to c] \mid a \geq 2, b \in Z, c \in Z\}$
z:=4
z:=3
6
$S\downarrow 6 \uparrow A = [x \to T, y \to T, z \to 4]$
$S\downarrow 6 = \{[x \to a, y \to b, z \to 4] \mid a < 2, b \in Z\}$
5
$S\downarrow 5 \uparrow A = [x \to T, y \to T, z \to 3]$
$S\downarrow 5 = \{[x \to a, y \to b, z \to 3] \mid a \geq 2, b \in Z\}$
skip
skip
7
$S\downarrow 7 \uparrow A = [x \to T, y \to T, z \to T]$
$S\downarrow 7 = S\downarrow 5 \cup$
skip

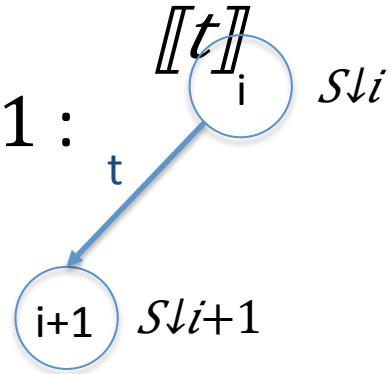
Abstract Interpretation

- Abstract Interpretation: inferring properties from an abstract state
- Abstract state is an over-approximation (superset of the concrete states)

  → cannot infer all properties of the concrete states

Definition of Semantics

- We defined Concrete Semantics:
  1. *Concrete Domain*: Σ (all possible states)
  2. *Transfer functions*: for each command t between 2 nodes i and $i+1$, we have a function $\downarrow c : 2^{\uparrow\Sigma} \rightarrow 2^{\uparrow\Sigma}$ which maps $S\downarrow i$ to $S\downarrow i+1$:



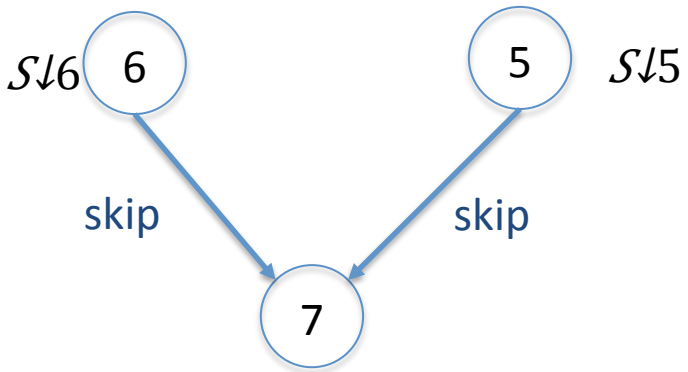
$$[\![v := 3]\!]\downarrow c\ (S) = \{\sigma[v \rightarrow 3] \mid \sigma \in S\}$$

$$[\![v := read()]\!]\downarrow c\ (S) = \{\sigma[v \rightarrow a] \mid \sigma \in S,\ a \in Z\}$$

$$[\![v \geq 2]\!]\downarrow c\ (S) = \{\sigma[v \rightarrow a] \mid \sigma \in S,\ a \geq 2\}$$

Definition of Semantics

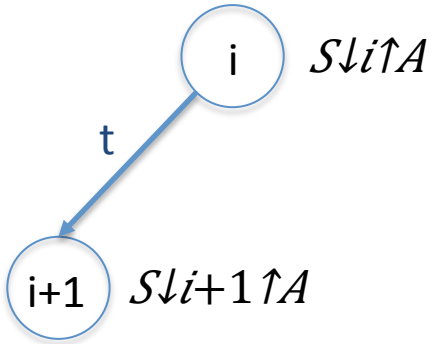
- We defined Concrete Semantics:
  3. *Join* operation $\sqcup_{\downarrow c}$:



$$S\downarrow 5 \sqcup_{\downarrow c} S\downarrow 6 := S\downarrow 5 \cup S\downarrow 6$$

Definition of Semantics

- In addition, we defined abstract semantics:
  1. *Abstract Domain*: $\Sigma \nearrow \mathrm{T} = \mathrm{Var} \rightarrow \mathrm{Z} \nearrow \mathrm{T}$
  2. *Transfer functions*: for each command t between 2 nodes i and $i+1$, we have a function $[\![t]\!]$ $\downarrow A$ which maps $S \downarrow i \uparrow A$ to $\mathrm{S} \downarrow \mathrm{i}+1 \uparrow A$



The Abstract Transfer Functions

$$[\![v=3]\!]\downarrow A\ (S\uparrow A\ )=S\uparrow A\ [v\rightarrow 3]$$

$$[\![v\leq 2]\!]\downarrow A\ (S\uparrow A\ )=\{\blacksquare S\uparrow A \qquad S\uparrow A\ (v)\leq 2 S\uparrow A$$

$$S\uparrow A\ (v)=T\bot \qquad S\uparrow A\ (v)>2$$

⊥ (bottom): the "undefined mapping" which represents the empty set of states.

Definition of Semantics

3. Join:

   Example:

$$S\downarrow 1\uparrow A = [x \to 2, y \to T, z \to 3]$$

$$S\downarrow 2\uparrow A = [x \to 2, y \to 5, z \to 4]$$

$$S\downarrow 1\uparrow A \sqcup S\downarrow 2\uparrow A = [x \to 2, y \to T, z \to T]$$

Join

- Formally:

$(S{\downarrow}1{\uparrow}A \sqcup S{\downarrow}2{\uparrow}A)(v) = \{\blacksquare T \quad S{\downarrow}1{\uparrow}A(v) = T\ T \quad S{\downarrow}2{\uparrow}A(v) = TT$

$S{\downarrow}1{\uparrow}A(v) \neq S{\downarrow}2{\uparrow}A(v) S{\downarrow}1{\uparrow}A(v) \quad S{\downarrow}1{\uparrow}A(v) = S{\downarrow}2{\uparrow}A(v)$

$S{\uparrow}A \sqcup \bot = S{\uparrow}A$

Stopping Problem?

- Is it possible that we discover new information forever?
- Define order relation:

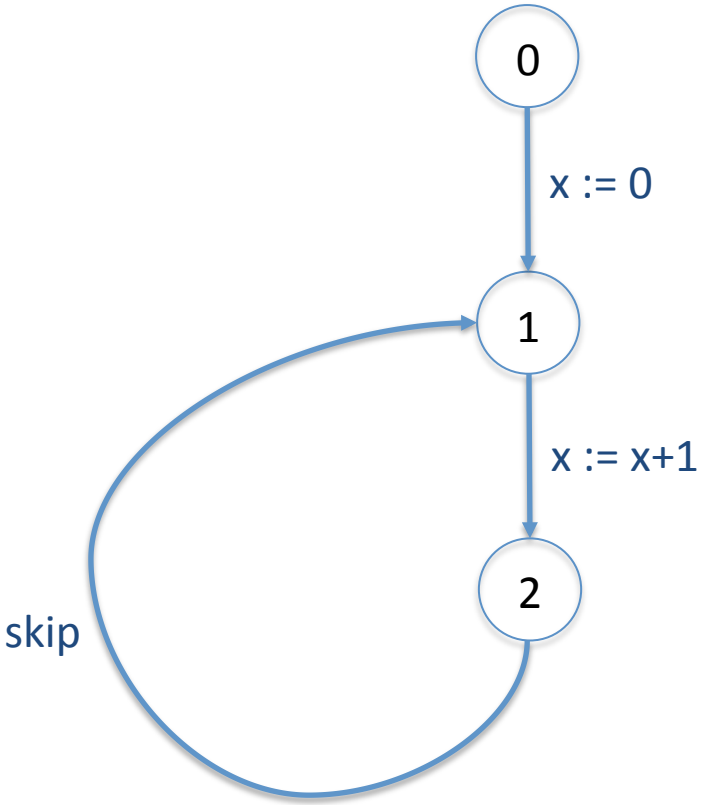
$$S\downarrow 1\uparrow A \sqsubseteq S\downarrow 2\uparrow A \quad if \quad S\downarrow 1\uparrow A \subseteq S\downarrow 2\uparrow A$$

- Notice that the join operation is monotonic
- At each node: each variable can go up at most 2 levels of abstraction:

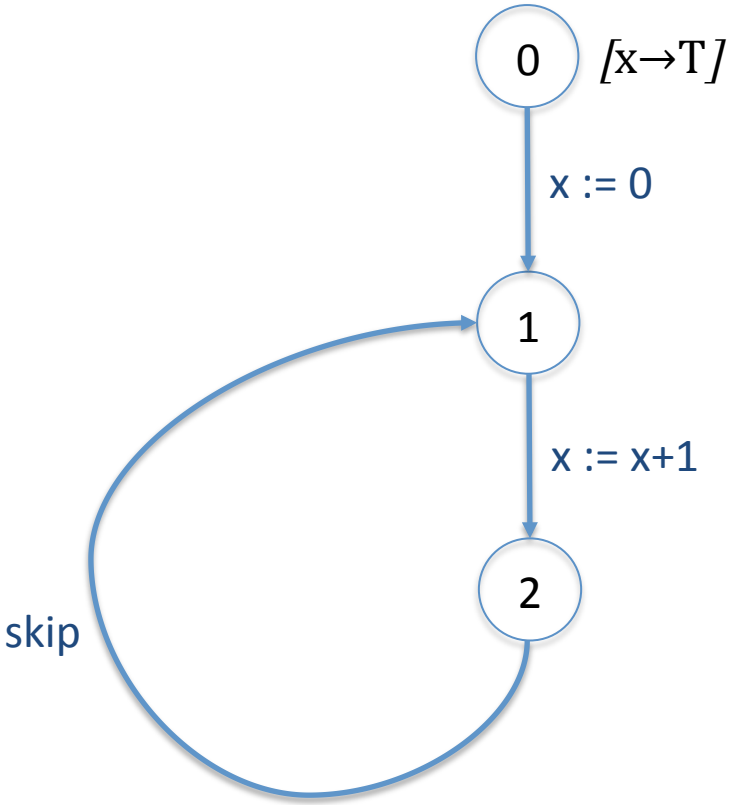
$$\perp \rightarrow 7 \rightarrow T$$

- Therefore: we will stop after finite number of steps.

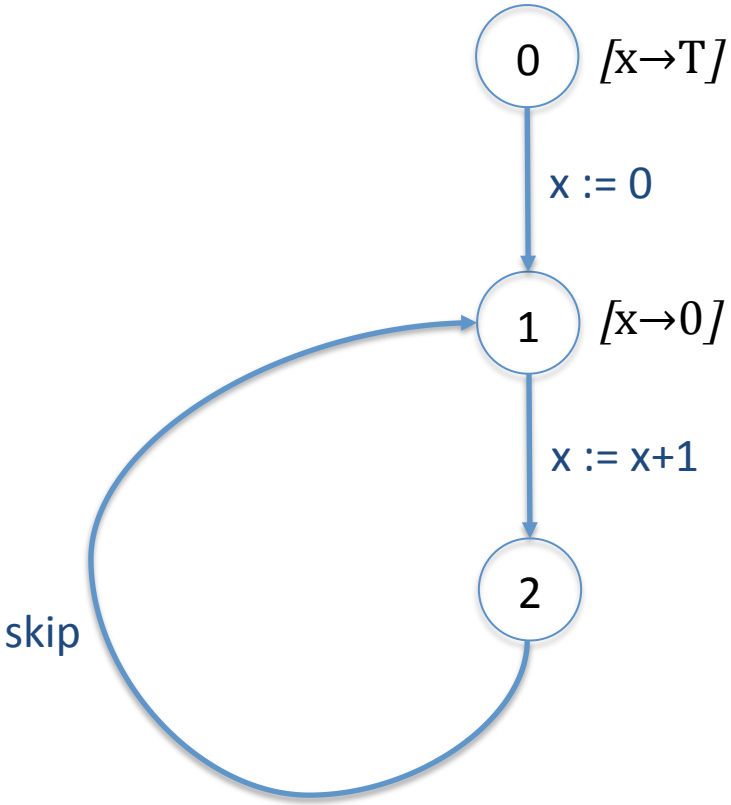
Example



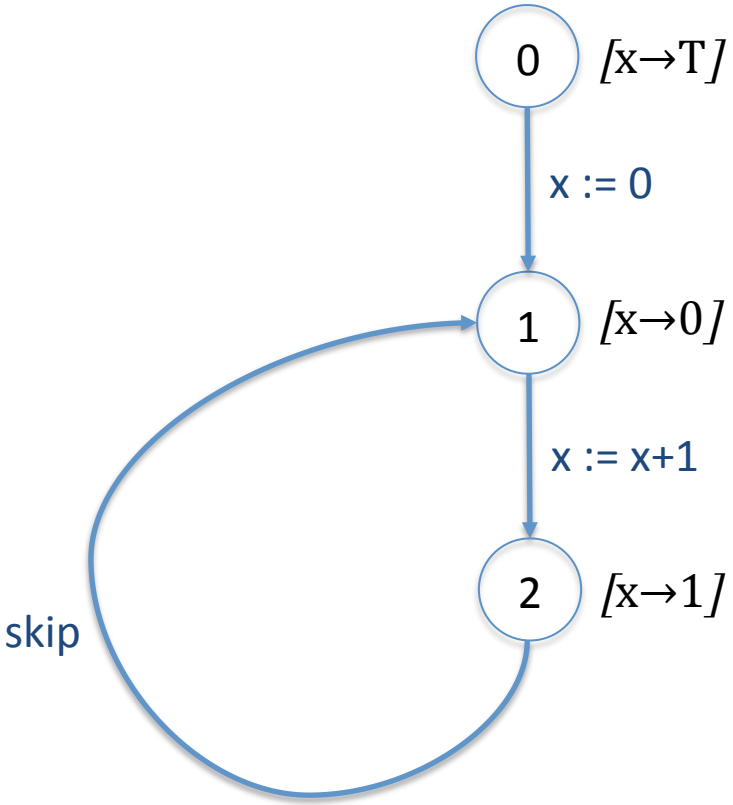
Example



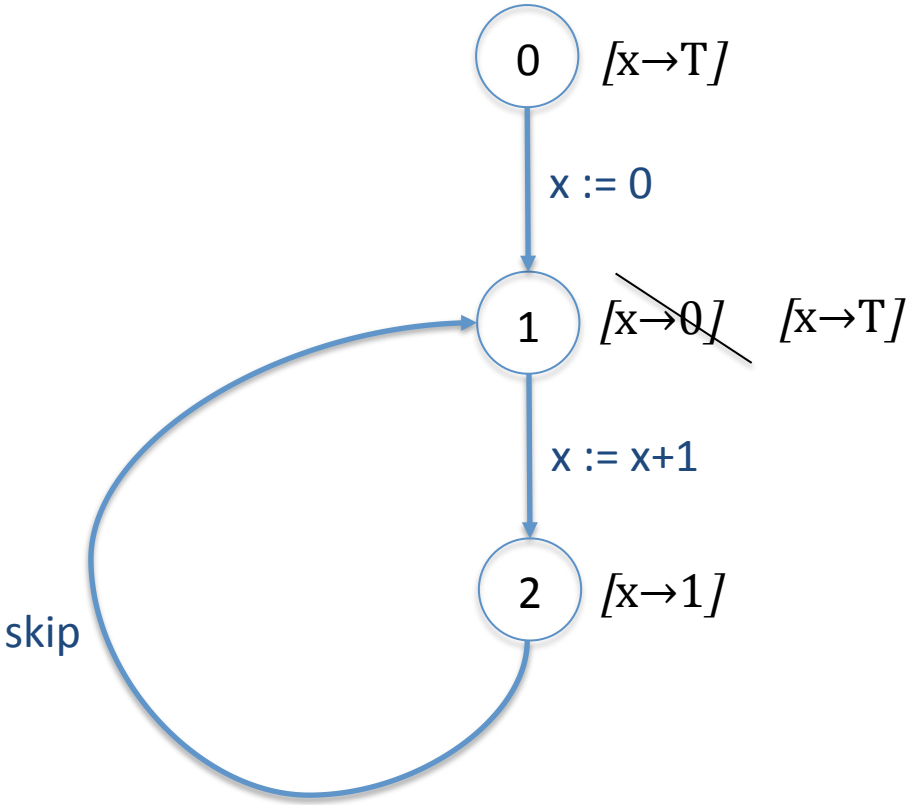
Example



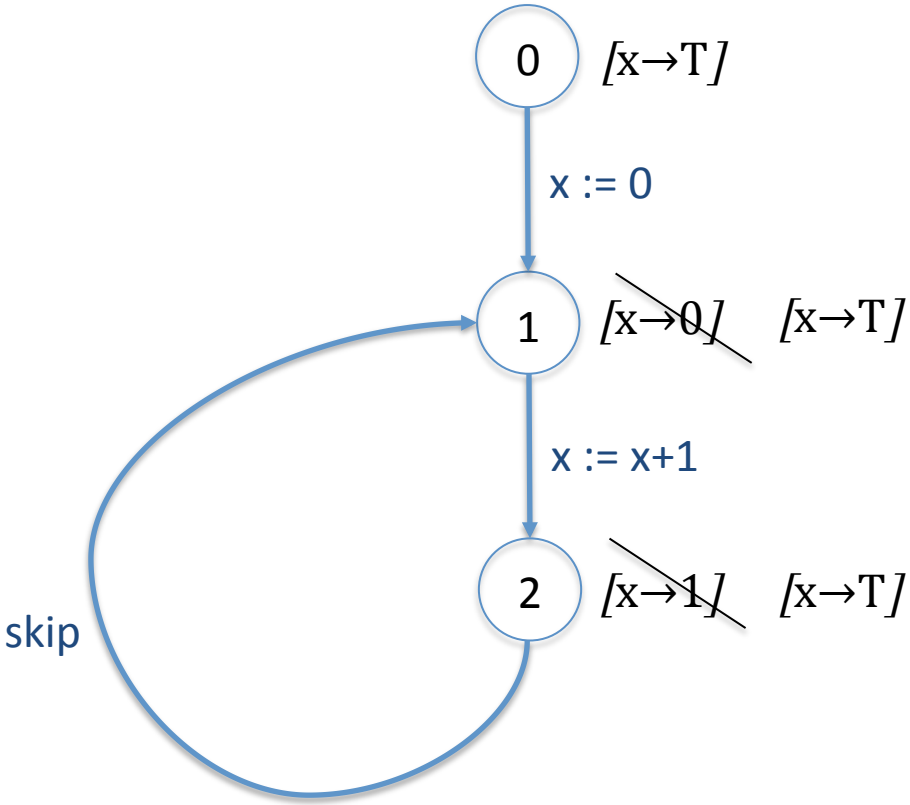
Example



Example

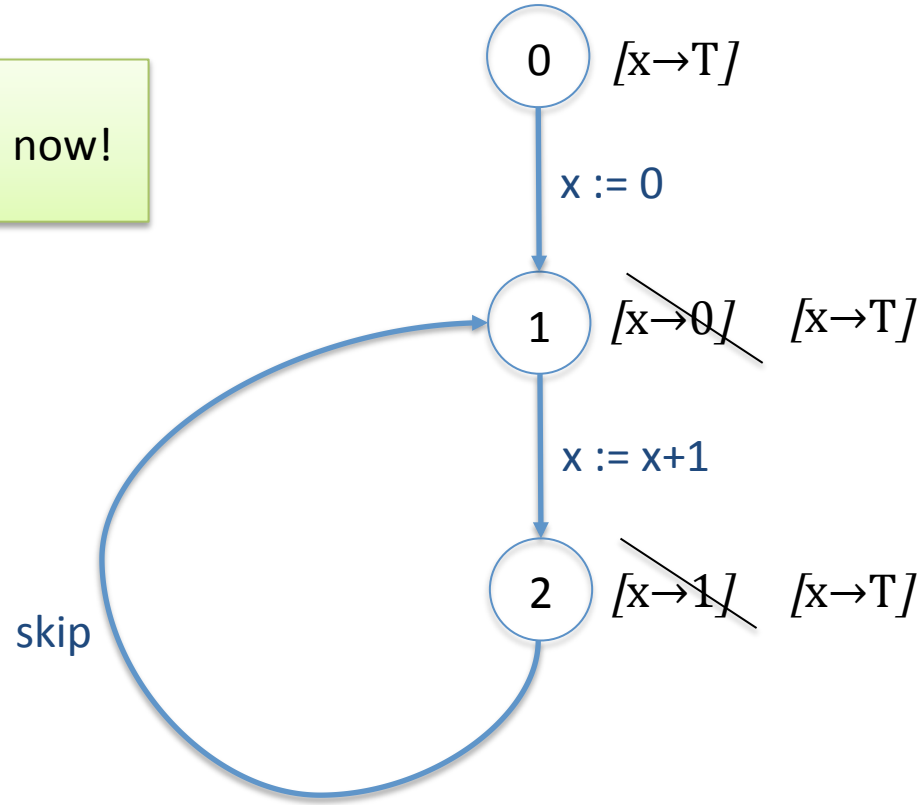


Example



Example

We can stop now!



To Summarize

- With our Abstract Semantic:
  - Abstract states are representable
  - No stopping problem
  - Soundness: each abstract state is a superset of the concrete states
    - If we prove a property of the abstract state, this is also true for the concrete states

Abstraction & Concretization functions

- Concretization function:

$\gamma(S\uparrow A)=S$

  - Maps each abstract state to the set of concrete states it represents

- Abstraction function:

$\alpha(S)=S\uparrow A$

  - Maps each set of concrete states to the "smallest" (most precise) abstract state which represents it

Abstract Domains

- In our example: very low precision
  - Because abstraction is coarse
- Better precision → more complexity
  - Representation of abstract states
  - Computation of the Transfer functions and Join
  - Takes more time to get to fixpoint (end of analysis)

Interval Abstraction

- Let's define a more precise abstract domain
- Possible values of a variable: an interval

$I=[a,b]$

$a \in Z \cup \{-\infty\},\ b \in Z \cup \{\infty\}$

- Transfer function: trivial

  $I,\ x \geq 2 \ \rightarrow\ I \cap [2,\infty]$

- Join:

$I=[a,b],\ J=[s,t]$

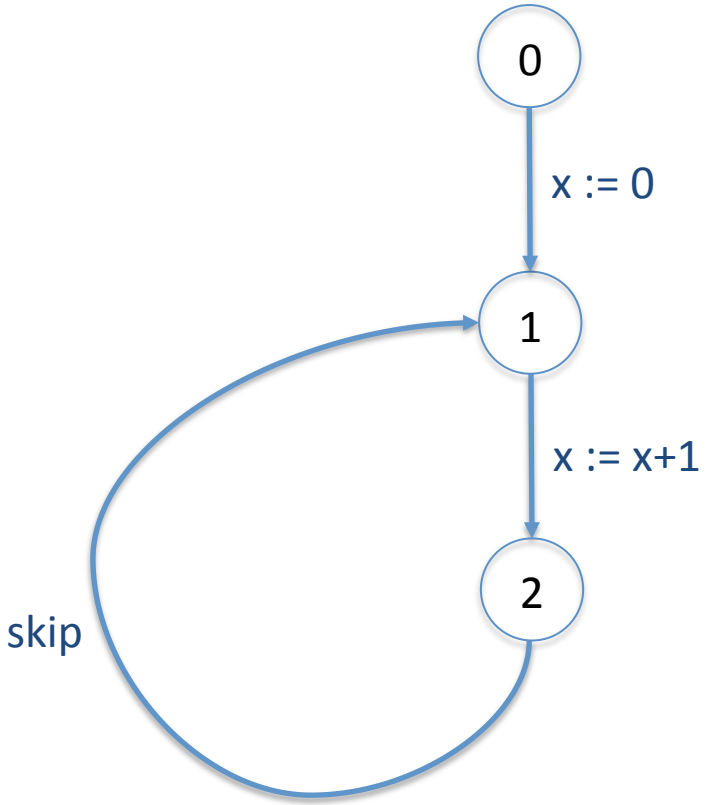
$I \sqcup J = [\inf(a,s)\ ,\sup(b,t)\ ]$

(the smallest interval which contains both I,J)

Interval Abstraction

- Is it guaranteed that analysis will reach a fixpoint and stop?

Interval Abstraction



Interval Abstraction

- The sequence of values assigned to x:

  $[0,0]$, $[1,1]$,$[2,2]$,$[3,3]$, …
- What would be the corresponding sequence of abstract states?

Interval Abstraction

- The sequence of values assigned to x:

  $[0,0]$, $[1,1]$,$[2,2]$,$[3,3]$, …
- What would be the corresponding sequence of abstract states?

  $[0,0]$, $[0,1]$,$[0,2]$,$[0,3]$,…
- Analysis will not stop!

Interval Abstraction

- Let's try a different *Join*
- Choose $L = [-2,2]$
- Define $\sqcup\!\uparrow_L$:

$$I \sqcup\!\uparrow_L J = \begin{cases} I \sqcup J & \text{if } I \sqsubseteq L \vee J \sqsubseteq I \\ [-\infty,\infty] & \text{otherwise} \end{cases}$$

Interval Abstraction

$$I \sqcup \uparrow_L J = \begin{cases} \blacksquare & \text{if } I \sqsubseteq L \vee J \sqsubseteq I[-\infty,\infty] \\ I \sqcup J & \text{otherwise} \end{cases} \qquad L = [-2,2]$$

- The sequence of values assigned to x:

  $[0,0]$, $[1,1]$,$[2,2]$,$[3,3]$,$[4,4]$, …
- What would be the corresponding sequence of abstract states?

Interval Abstraction

$$I \sqcup_{\uparrow L} J = \begin{cases} \blacksquare I \sqcup J & \text{if } I \sqsubseteq L \vee J \sqsubseteq I[-\infty,\infty] \\ \text{otherwise} & L = [-2,2] \end{cases}$$

- The sequence of values assigned to x:

  $[0,0]$, $[1,1]$,$[2,2]$,$[3,3]$,$[4,4]$, …
- What would be the corresponding sequence of abstract states?

  $[0,0]$, $[0,1]$, $[0,2]$, $[0,3]$, ?

Interval Abstraction

$$I \sqcup \uparrow_L J = \begin{cases} \blacksquare & \text{if } I \sqsubseteq L \vee J \sqsubseteq I[-\infty,\infty] \\ I \sqcup J & \text{otherwise} \end{cases} \qquad L = [-2,2]$$

- The sequence of values assigned to x:

  $[0,0]$, $[1,1]$,$[2,2]$,$[3,3]$,$[4,4]$, …
- What would be the corresponding sequence of abstract states?

  $[0,0]$, $[0,1]$, $[0,2]$,$[0,3]$, $[-\infty,\infty]$, $[-\infty,\infty]$, …

Interval Abstraction

$$I \sqcup\!\uparrow_L J = \begin{cases} \blacksquare & \text{if } I \sqsubseteq L \vee J \sqsubseteq I \\ I \sqcup J & \text{otherwise} \end{cases} \qquad [-\infty, \infty] \qquad L = [-2, 2]$$

- $I \sqcup\!\uparrow_L J$ can grow bigger than I only if $I \sqsubseteq L$

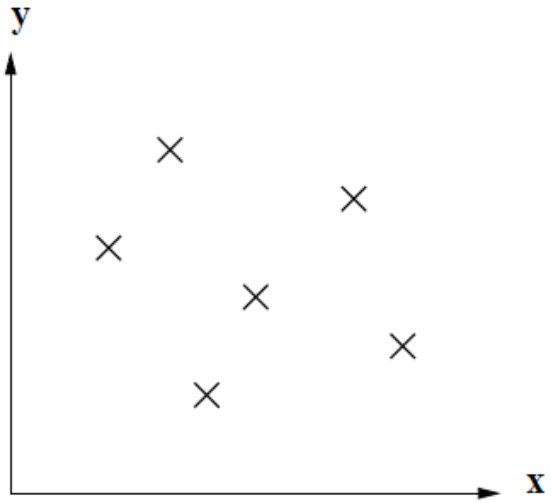
Interval Abstraction

$[0,0]$, $[0,1]$, $[0,2]$, $[0,3]$, $[-\infty,\infty]$, $[-\infty,\infty]$, …

- Going up from $[0,3]$ to $[-\infty,\infty]$ is called ***Widening***
- We forget information
- We do it conservatively (maintaining over-approximation)
- This loss of information ensures stopping

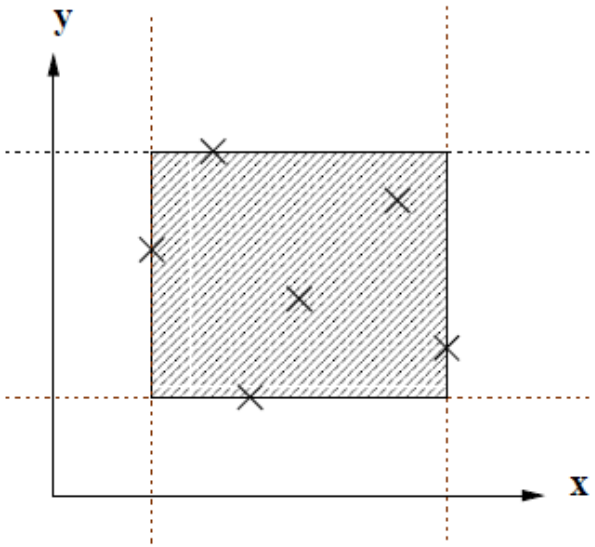
Interval Abstraction

- Interval abstraction is more precise, but …
- It doesn't maintain any relation between variables
- Consider 2 variables x, y. Suppose the relation between them is:



Interval Abstraction

- In the interval domain, the best over-approximation is a rectangle with sides parallel to the axis:

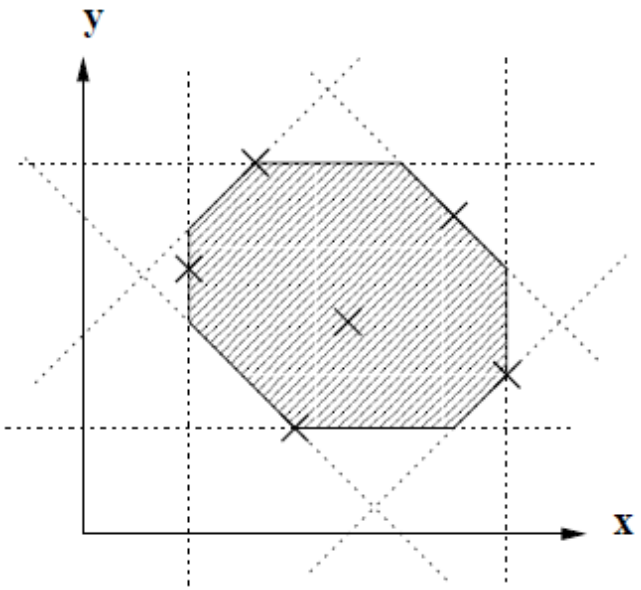


Octagon Abstraction

- Octagon Abstraction: a more complex domain with a better precision
- For each 2 variables, maintain inequalities of the form: $\pm x \pm y \leq c$
- Here we do maintain relations between variables

Octagon Abstraction

- Here, the best over-approximation is octagon – a polygon with at most eight edges:

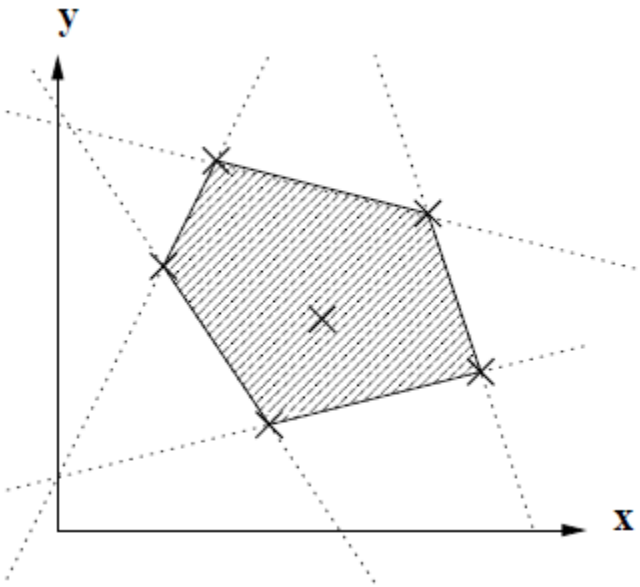


Polyhedron Abstraction

- And a more precise domain: Polyhedron
- For each 2 variables, maintain inequalities of the form: $ax+by\leq c$
- Here we maintain more informative relations between variables

Polyhedron Abstraction

- Here, the best over-approximation is the convex polygon defined by the inequalities:



Conclusion

- Non-trivial questions about a program: undecidable
- Abstract Interpretation: an over-approximation of the possible executions

→ Sound static analysis

- Abstract Domains
  - Tradeoff between precision and complexity

Questions?