

# COURSE STRUCTURE

## CONSTRAINT SOLVERS

- Propositional Logic, SAT solving, DPLL
- First-Order Logic, SMT
- First-Order Theories

## DEDUCTIVE VERIFICATION

- Operational Semantics
- Strongest Post-condition, Weakest Pre-condition
- Hoare Logic

## MODEL CHECKING AND OTHER VERIFICATION TECHNIQUES

- Abstract Interpretation
- Predicate Abstraction, CEGAR
- Property-directed Reachability



# ABSTRACT INTERPRETATION



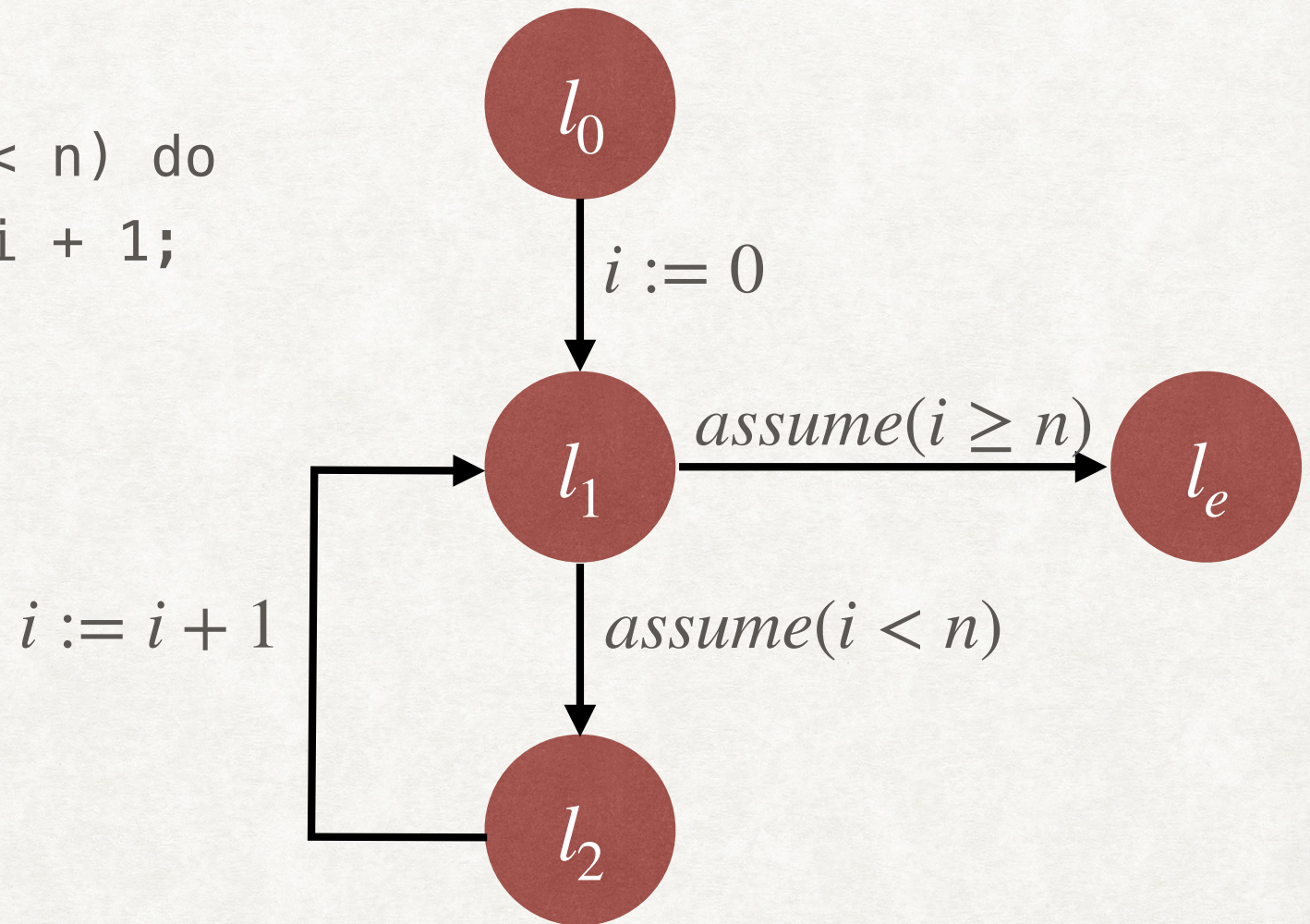
# LABELLED TRANSITION SYSTEM

- We express the program  $c$  as a labelled transition system  $\Gamma_c \equiv (V, L, l_0, l_e, T)$ 
  - $V$  is the set of program variables
  - $L$  is the set of program locations
  - $l_0$  is the start location
  - $l_e$  is the end location
  - $T \subseteq L \times c \times L$  is the set of labelled transitions between locations.



# EXAMPLE

```
i := 0;  
while(i < n) do  
  i := i + 1;
```





# PROGRAMS AS LTS

- There are various ways to construct the LTS of a program
  - We can use control flow graph
  - We can use basic paths as defined by the book (BM Chapter 5). A basic path is a sequence of instructions that begins at the start of the program or a loop head, and ends at a loop head or the end of the program.
- Program State  $(\sigma, l)$  consists of the values of the variables  $(\sigma : V \rightarrow \mathbb{R})$  and the location.
- An execution is a sequence of program states,  $(\sigma_0, l_0), (\sigma_1, l_1), \dots, (\sigma_n, l_n)$ , such that for all  $i$ ,  $0 \leq i \leq n - 1$ ,  $(l_i, c, l_{i+1}) \in T$  and  $(\sigma_i, c) \hookrightarrow^* (\sigma_{i+1}, \text{skip})$ .
- A program satisfies its specification  $\{P\}c\{Q\}$  if  $\forall \sigma \in P$ , for all executions  $(\sigma, l_0), (\sigma_1, l_1), \dots, (\sigma', l_e)$  of  $\Gamma_c$ ,  $\sigma' \in Q$ .



# INDUCTIVE ASSERTION MAP

- With each location, we associate a set of states which are reachable at that location in any execution.
  - $\mu : L \rightarrow \Sigma(V)$
- To express that such a map is an inductive assertion map, we will use Strongest Post-condition.
  - $\forall (l, c, l') \in T. sp(\mu(l), c) \rightarrow \mu(l')$
- Then, if  $\mu$  is an inductive assertion map on  $\Gamma_c$ , the Hoare triple  $\{P\}c\{Q\}$  is valid if  $P \rightarrow \mu(l_0)$  and  $\mu(l_e) \rightarrow Q$ .



# GENERATING THE INDUCTIVE ASSERTION MAP

- We can express the inductive assertion map as a solution of a system of equations:
  - $X_{l_0} = P$
  - For all other locations  $l \in L \setminus \{l_0\}$ ,  $X_l = \bigvee_{(l',c,l) \in T} sp(X_{l'}, c)$



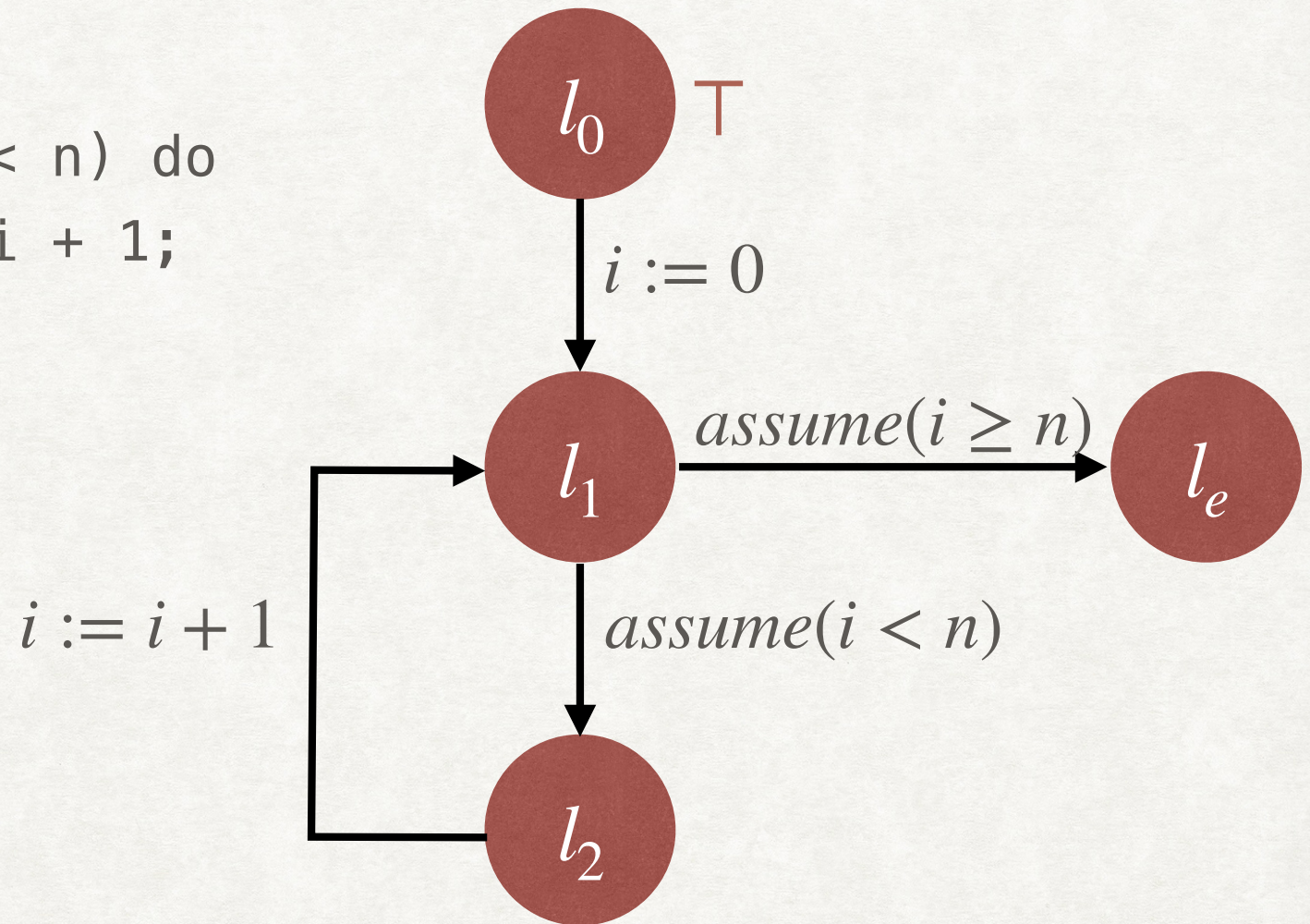
# GENERATING THE INDUCTIVE ASSERTION MAP

```
ForwardPropagate( $\Gamma_c, P$ )
   $S := \{l_0\};$ 
   $\mu(l_0) := P;$ 
   $\mu(l) := \perp$ , for  $l \in L \setminus \{l_0\};$ 
  while  $S \neq \emptyset$  do{
     $l := \text{Choose } S;$ 
     $S := S \setminus \{l\};$ 
    foreach  $(l, c, l') \in T$  do{
       $F := sp(\mu(l), c);$ 
      if  $\neg(F \rightarrow \mu(l'))$  then{
         $\mu(l') := \mu(l') \vee F;$ 
         $S := S \cup \{l'\};$ 
      }
    }
  }
```



# EXAMPLE

```
i := 0;  
while(i < n) do  
  i := i + 1;
```

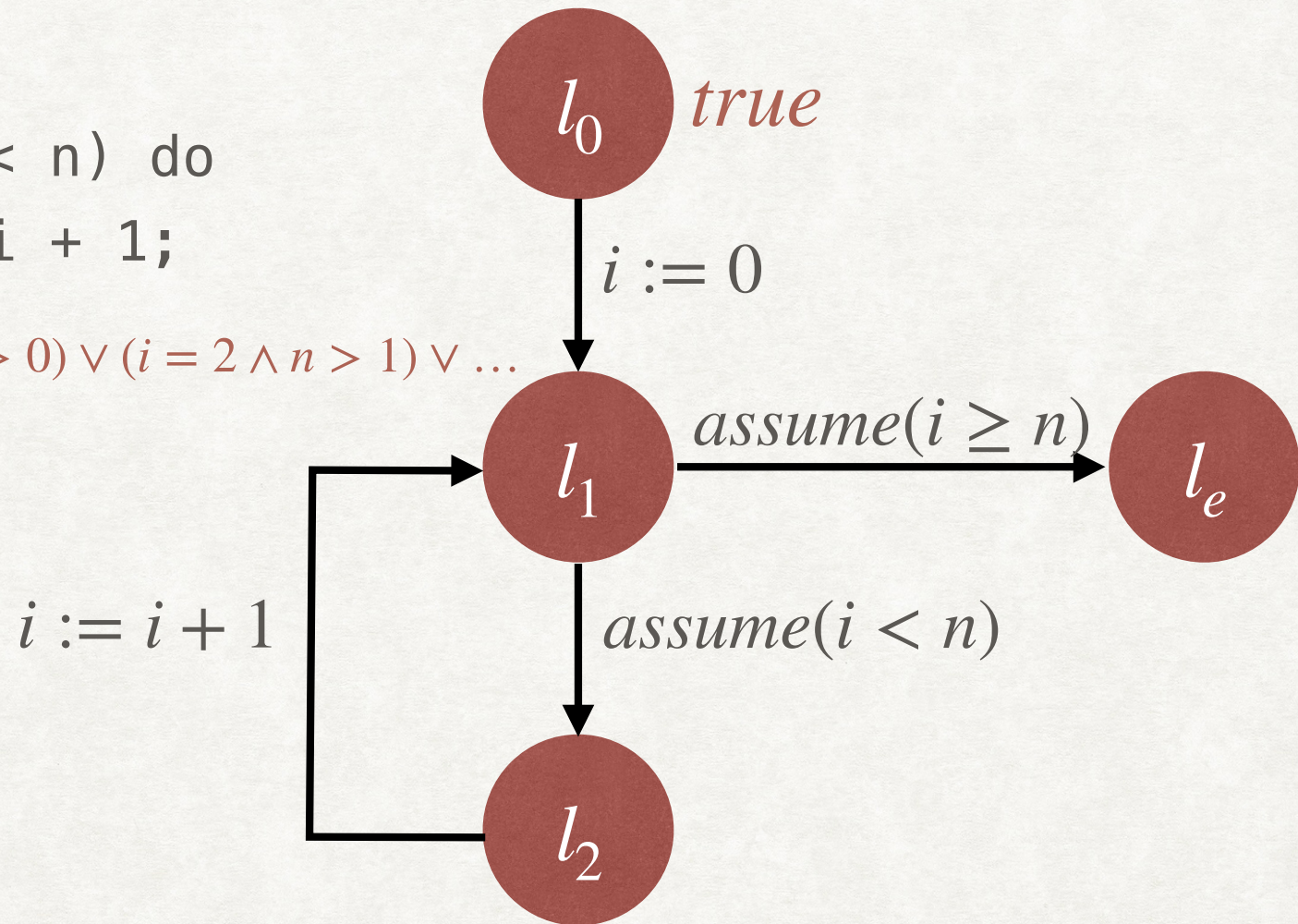




# EXAMPLE

```
i := 0;  
while(i < n) do  
  i := i + 1;
```

$(i = 0) \vee (i = 1 \wedge n > 0) \vee (i = 2 \wedge n > 1) \vee \dots$

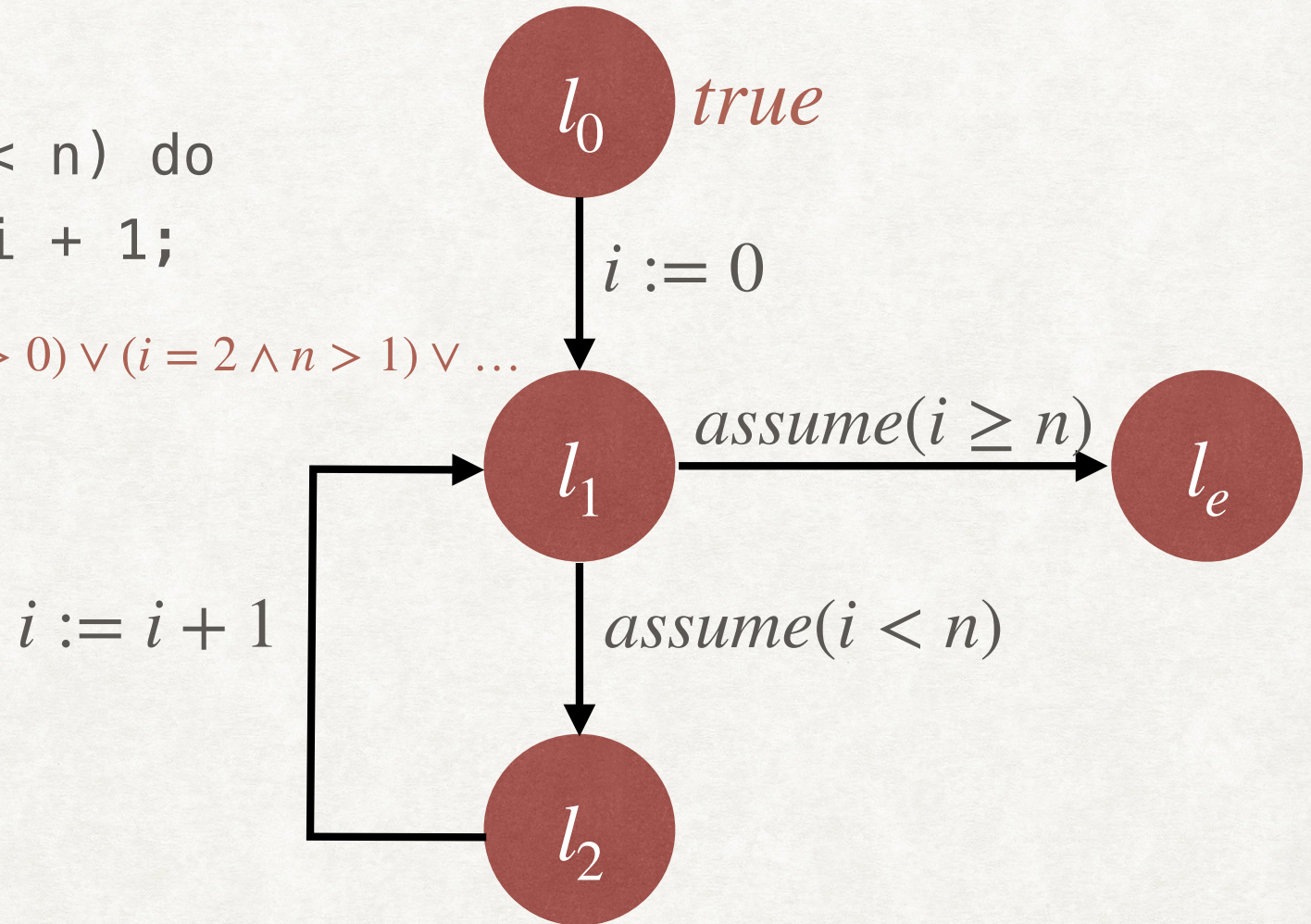




# EXAMPLE

```
i := 0;  
while(i < n) do  
  i := i + 1;
```

$(i = 0) \vee (i = 1 \wedge n > 0) \vee (i = 2 \wedge n > 1) \vee \dots$



FORWARDPROPAGATE WILL NOT TERMINATE



# ABSTRACT INTERPRETATION: OVERVIEW

- Instead of maintaining an arbitrary set of states at each location, maintain an artificially constrained set of states, coming from an abstract domain  $D$ .
  - $\hat{\mu} : L \rightarrow D$
- Let  $States \triangleq V \rightarrow \mathbb{R}$  be the set of all possible concrete states.
  - Abstraction function,  $\alpha : \mathbb{P}(States) \rightarrow D$
  - Concretization function,  $\gamma : D \rightarrow \mathbb{P}(States)$
- $\hat{\mu}$  over approximates the set of states at every location.
  - For all locations  $l$ ,  $\gamma(\hat{\mu}(l)) \supseteq \mu(l)$
- Use abstract strongest post-condition operator  $\hat{sp} : D \times c \rightarrow D$ 
  - $\gamma(\hat{sp}(d, c)) \supseteq sp(\gamma(d), c)$



# GENERATING THE INDUCTIVE ASSERTION MAP

```
ForwardPropagate( $\Gamma_c, P$ )
   $S := \{l_0\};$ 
   $\mu(l_0) := P;$ 
   $\mu(l) := \perp$ , for  $l \in L \setminus \{l_0\};$ 
  while  $S \neq \emptyset$  do{
     $l := \text{Choose } S;$ 
     $S := S \setminus \{l\};$ 
    foreach  $(l, c, l') \in T$  do{
       $F := sp(\mu(l), c);$ 
      if  $\neg(F \rightarrow \mu(l'))$  then{
         $\mu(l') := \mu(l') \vee F;$ 
         $S := S \cup \{l'\};$ 
      }
    }
  }
```



# ABSTRACT FORWARD PROPAGATE

AbstractForwardPropagate( $\Gamma_c, P$ )

$S := \{l_0\};$

$\hat{\mu}(l_0) := \alpha(P);$

$\hat{\mu}(l) := \perp, \text{ for } l \in L \setminus \{l_0\};$

while  $S \neq \emptyset$  do{

$l := \text{Choose } S;$

$S := S \setminus \{l\};$

    foreach  $(l, c, l') \in T$  do{

$F := \hat{sp}(\hat{\mu}(l), c);$

        if  $\neg(F \leq \hat{\mu}(l'))$  then{

$\hat{\mu}(l') := \hat{\mu}(l') \sqcup F;$

$S := S \cup \{l'\};$

        }

    }

}



# ABSTRACT FORWARD PROPAGATE

AbstractForwardPropagate( $\Gamma_c, P$ )

$S := \{l_0\};$

$\hat{\mu}(l_0) := \alpha(P);$

$\hat{\mu}(l) := \perp, \text{ for } l \in L \setminus \{l_0\};$

while  $S \neq \emptyset$  do{

$l := \text{Choose } S;$

$S := S \setminus \{l\};$

    foreach  $(l, c, l') \in T$  do{

$F := \hat{sp}(\hat{\mu}(l), c);$

        if  $\neg(F \leq \hat{\mu}(l'))$  then{

$\hat{\mu}(l') := \hat{\mu}(l') \sqcup F;$

$S := S \cup \{l'\};$

        }

    }

}

Abstract Domain D  
is a lattice  $(D, \leq, \sqcup)$



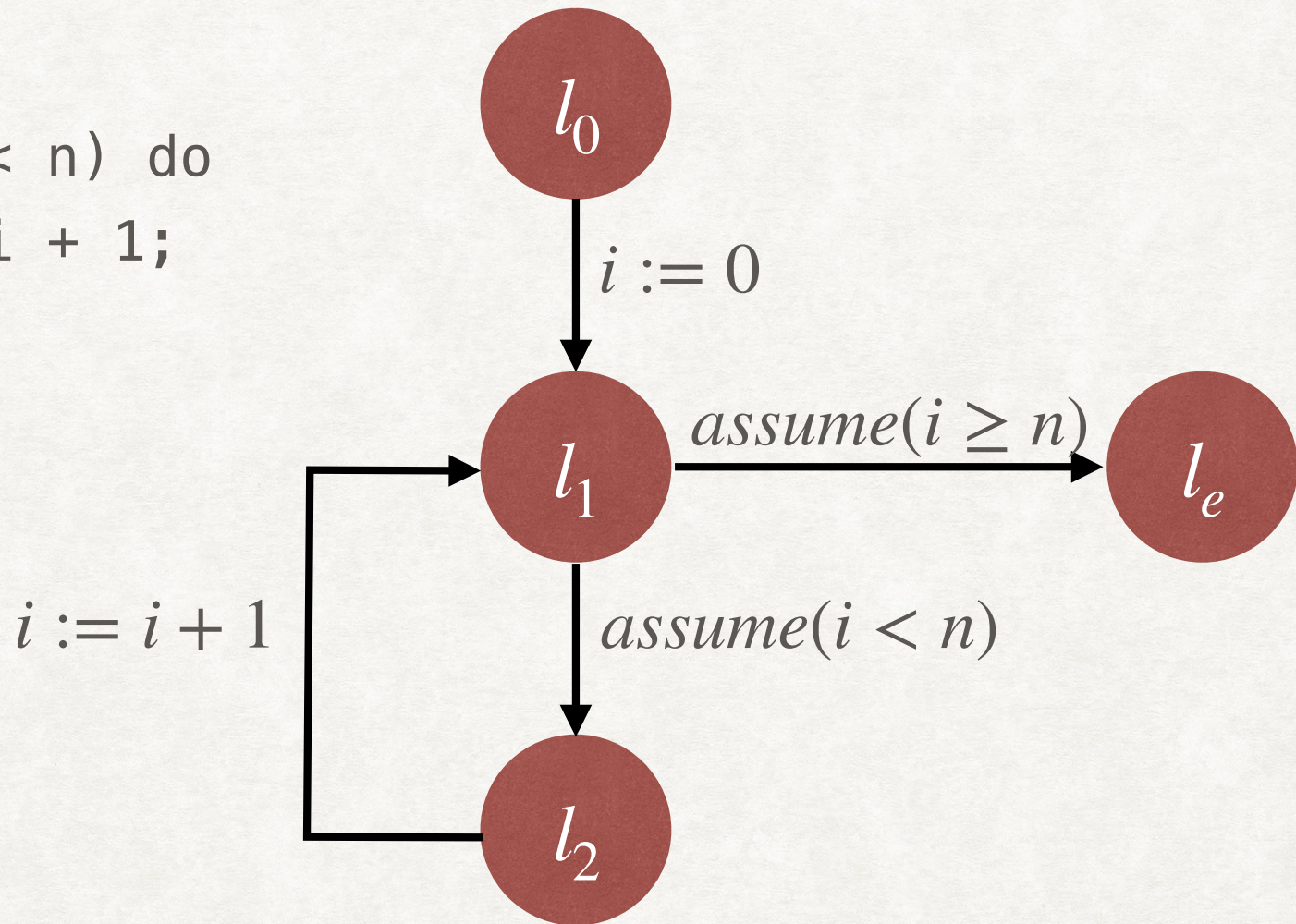
# ABSTRACT INTERPRETATION: OVERVIEW

- At the end, we will check whether  $\hat{\mu}(l_e) \leq \alpha(Q)$ .
  - Equivalently,  $\gamma(\hat{\mu}(l_e)) \subseteq Q$



## EXAMPLE

```
i := 0;  
while(i < n) do  
  i := i + 1;
```



Suppose we want to prove the post-condition :  $i \geq 0$



# EXAMPLE

```
i := 0;  
while(i < n) do  
  i := i + 1;
```

Sign Abstract Domain:

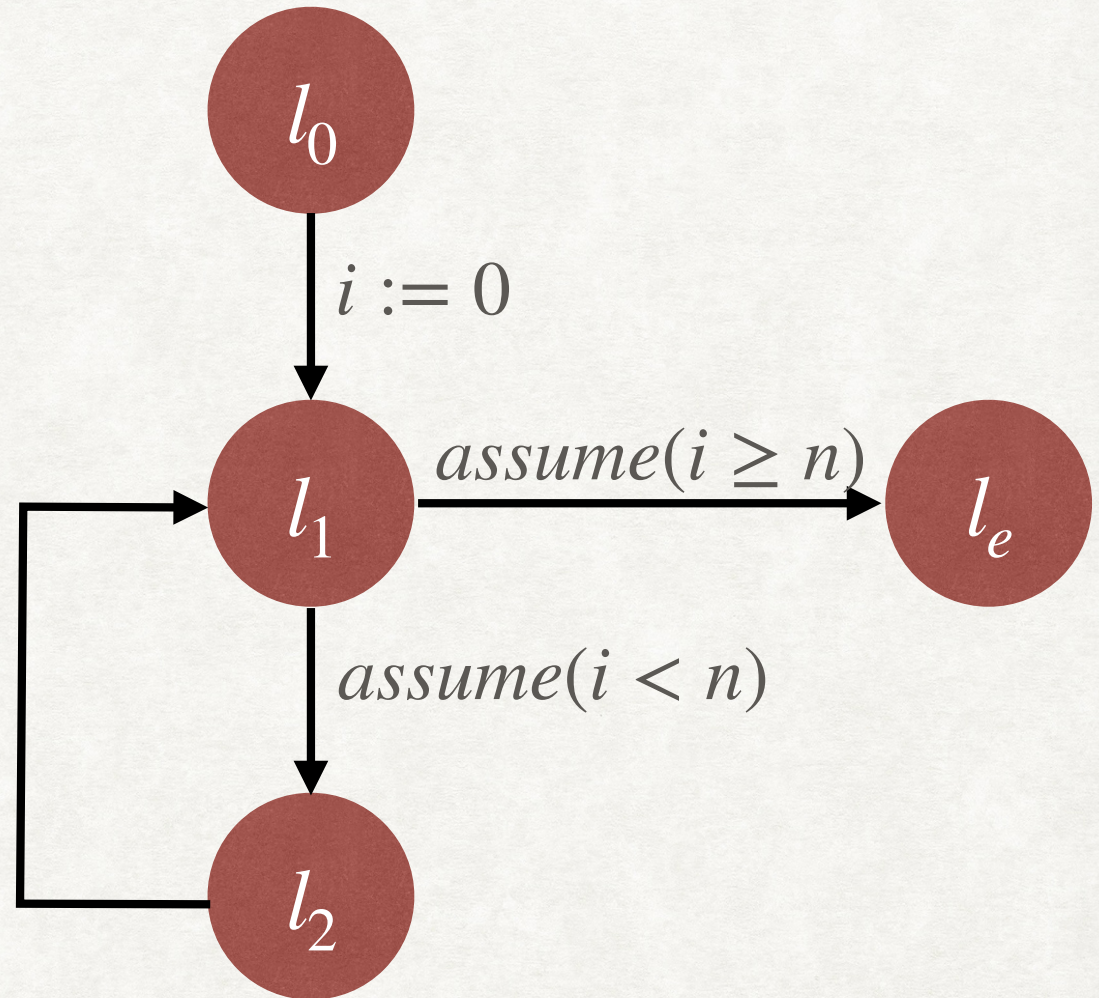
$D = \{ +, -, \perp \}$

$\gamma(+, -) = \top$

$\gamma(+ ) = i \geq 0$

$\gamma(- ) = i < 0$

$\gamma(\perp ) = \perp$





# EXAMPLE

```
i := 0;  
while(i < n) do  
  i := i + 1;
```

Sign Abstract Domain:

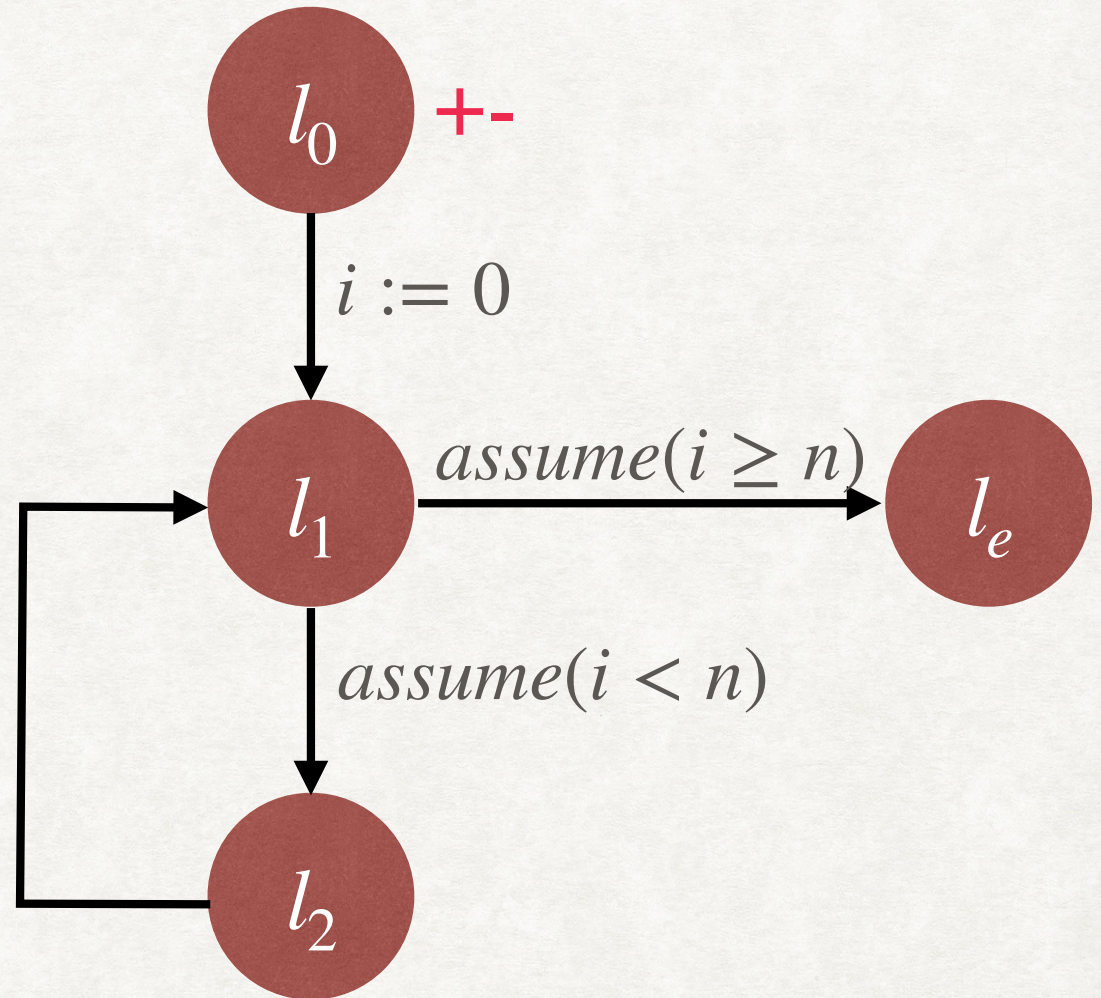
$D = \{ +-, +, -, \perp \}$

$\gamma(+ -) = \top$

$\gamma(+ ) = i \geq 0$

$\gamma(- ) = i < 0$

$\gamma(\perp ) = \perp$





# EXAMPLE

```
i := 0;  
while(i < n) do  
  i := i + 1;
```

Sign Abstract Domain:

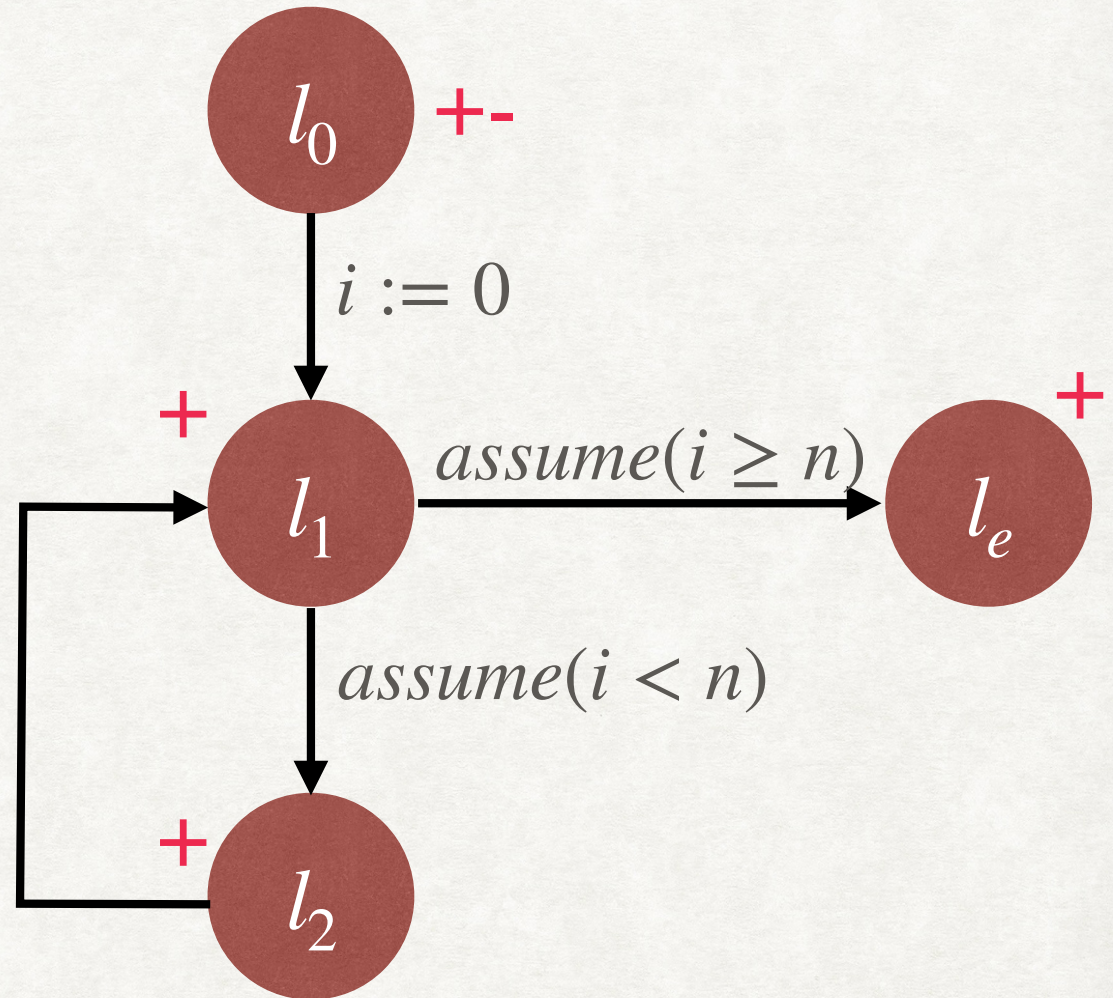
$D = \{ +-, +, -, \perp \}$

$\gamma(+ -) = \top$

$\gamma(+ ) = i \geq 0$

$\gamma(- ) = i < 0$

$\gamma(\perp ) = \perp$





# ABSTRACT INTERPRETATION: OVERVIEW

- Desirable properties of Abstract Interpretation
  - Soundness:  $\hat{\mu}$  over approximates the set of states at every location.
  - Guaranteed termination of AbstractForwardPropagate
- We will use concepts from lattice theory to characterise the conditions required for these properties.



# SNEAK PEEK

## SOUNDNESS OF ABSTRACT INTERPRETATION

- An abstract interpretation  $(D, \leq, \alpha, \gamma)$  is sound if:
  - $(D, \leq)$  is **complete lattice**.
  - $(\mathbb{P}(\text{State}), \subseteq) \begin{smallmatrix} \xrightarrow{\alpha} \\ \xleftarrow{\gamma} \end{smallmatrix} (D, \leq)$  is a **Galois Connection**.
  - $\hat{sp}$  is a **consistent abstraction** of  $sp$ .



# SNEAK PEEK

## GUARANTEED TERMINATION OF ABSTRACT FORWARD PROPAGATE

- AbstractForwardPropagate on abstract domain  $(D, \leq)$  is guaranteed to terminate if:
  - $(D, \leq)$  is a **complete lattice**.
  - $\hat{sp}$  is **monotonic**.
  - $(D, \leq)$  satisfies the **ascending chain condition**.



# PARTIAL ORDER

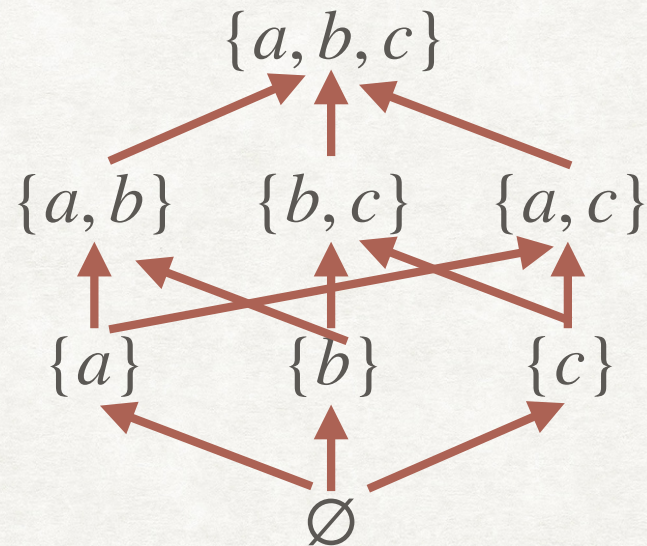
- Given a set  $D$ , a binary relation  $\leq \subseteq D \times D$  is a partial order on  $D$  if
  - $\leq$  is reflexive:  $\forall d \in D. d \leq d$
  - $\leq$  is anti-symmetric:  $\forall d, d' \in D. d \leq d' \wedge d' \leq d \rightarrow d = d'$
  - $\leq$  is transitive:  $\forall d_1, d_2, d_3 \in D, d_1 \leq d_2 \wedge d_2 \leq d_3 \rightarrow d_1 \leq d_3$
- Examples
  - $\leq$  on  $\mathbb{N}$  is a partial order.
  - Given a set  $S$ ,  $\subseteq$  on  $\mathbb{P}(S)$  is a partial order.



# PARTIAL ORDER - EXAMPLES

$$S = \{a, b, c\}$$

$$\mathbb{P}(S) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{b, c\}, \{a, c\}, \{a, b, c\}\}$$



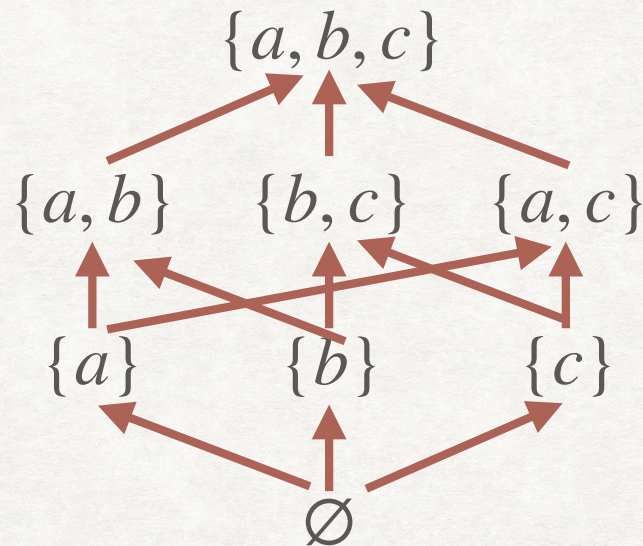
Partially Ordered Set:  $(\mathbb{P}(S), \subseteq)$



# PARTIAL ORDER - EXAMPLES

$$S = \{a, b, c\}$$

$$\mathbb{P}(S) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{b, c\}, \{a, c\}, \{a, b, c\}\}$$



Hasse diagram:

- Doesn't show reflexive edges (self-loops)
- Doesn't show transitive edges

Partially Ordered Set:  $(\mathbb{P}(S), \subseteq)$



# PARTIAL ORDER - MORE EXAMPLES

- Which of the following are partially ordered sets (posets)?
  - $(\mathbb{N} \times \mathbb{N}, \{(a, b), (c, d) \mid a \leq c\})$
  - $(\mathbb{N} \times \mathbb{N}, \{(a, b), (c, d) \mid a \leq c \wedge b \leq d\})$
  - $(\mathbb{N} \times \mathbb{N}, \{(a, b), (c, d) \mid a \leq c \vee b \leq d\})$



# LEAST UPPER BOUND

- Given a poset  $(D, \leq)$  and  $X \subseteq D$ ,  $u \in D$  is called an **upper bound** on  $X$  if  $\forall x \in X. x \leq u$ .
- $u \in D$  is called the **least upper bound (lub) of  $X$** , if  $u$  is an upper bound of  $X$ , and for every other upper bound  $u'$  of  $X$ ,  $u \leq u'$ .
- We use the notation  $\sqcup X$  to denote the least upper bound of  $X$ . Also called the join of  $X$ .
- **Exercise:** Prove that the least upper bound, if it exists, is unique.



# GREATEST LOWER BOUND

- Given a poset  $(D, \leq)$  and  $X \subseteq D$ ,  $l \in D$  is called a **lower bound** on  $X$  if  $\forall x \in X. l \leq x$ .
- $l \in D$  is called the **greatest lower bound (glb) of  $X$** , if  $l$  is a lower bound of  $X$ , and for every other lower bound  $l'$ ,  $l' \leq l$ .
- We use the notation  $\sqcap X$  to denote the greatest lower bound of  $X$ . Also called the meet of  $X$ .
- **Homework**: Prove that the greatest lower bound, if it exists, is unique.