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

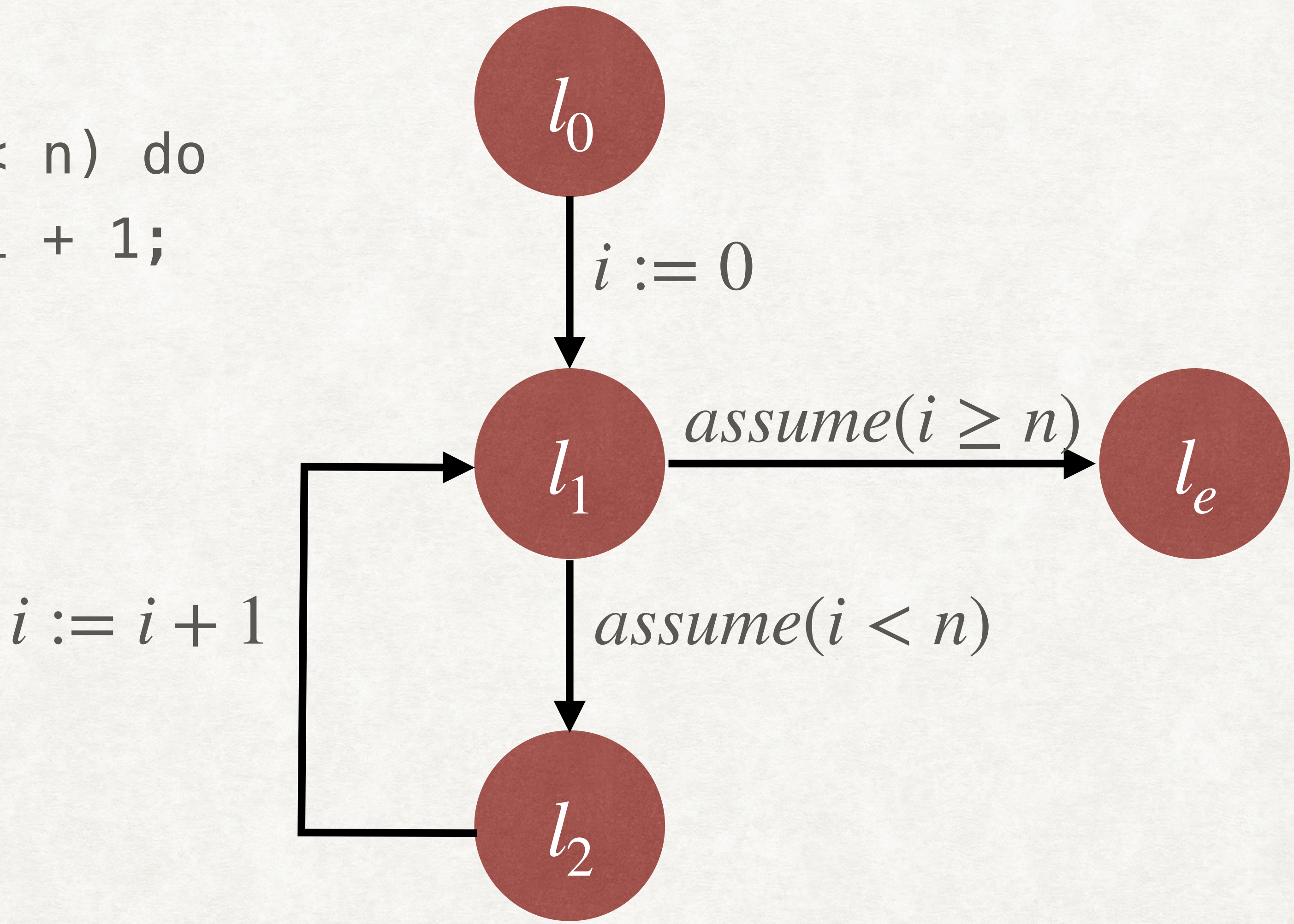
ABSTRACT INTERPRETATION

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 (σ, l) consists of the values of the variables $(\sigma : V \to \mathbb{R})$ and the location.
- An execution is a sequence of program states, $(\sigma_0, l_0), (\sigma_1, l_1), \ldots, (\sigma_n, l_n)$, such that for all i, $0 \leq i \leq n - 1$, $(l_i, c, l_{i+1}) \in T$ and $(\sigma_i, c) \hookrightarrow^* (\sigma_{i+1}, skip)$.
- A program satisfies its specification $\{P\}c\{Q\}$ if $\forall \sigma \in P$, for all executions $(\sigma, l_0), (\sigma_1, l_1), \ldots, (\sigma', l_e)$ of Γ_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 μ is an inductive assertion map on Γ_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

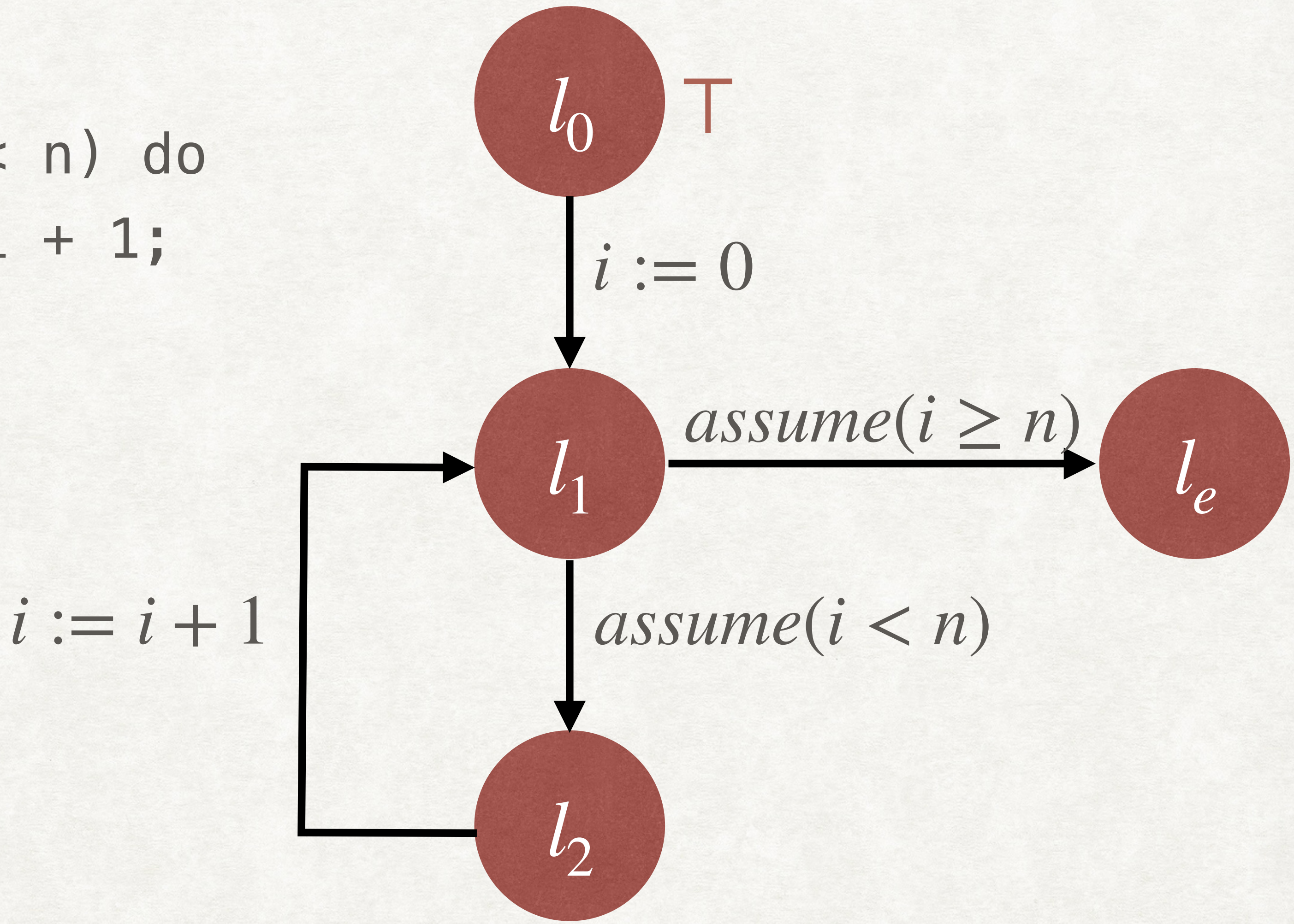
- We can express the inductive assertion map as a solution of a system of equations:
  - $X_{l_0} = P$
  - For all other locations $l \in L \backslash \{l_0\},\ X_l = \bigvee_{(l',c,l) \in T} sp(X_{l'}, c)$

GENERATING THE INDUCTIVE ASSERTION MAP

```
ForwardPropagate(Γ_c,P)
  S := {l_0};
  μ(l_0) := P;
  μ(l) := ⊥, for l ∈ L\{l_0};
  while S ≠ ∅ do{
     l := Choose S;
     S := S \ {l};
     foreach (l, c, l′) ∈ T do{
        F := sp(μ(l), c);
        if ¬(F → μ(l′)) then{
           μ(l′) := μ(l′) ∨ F;
           S := S ∪ {l′};
        }
     }
  }
```

EXAMPLE

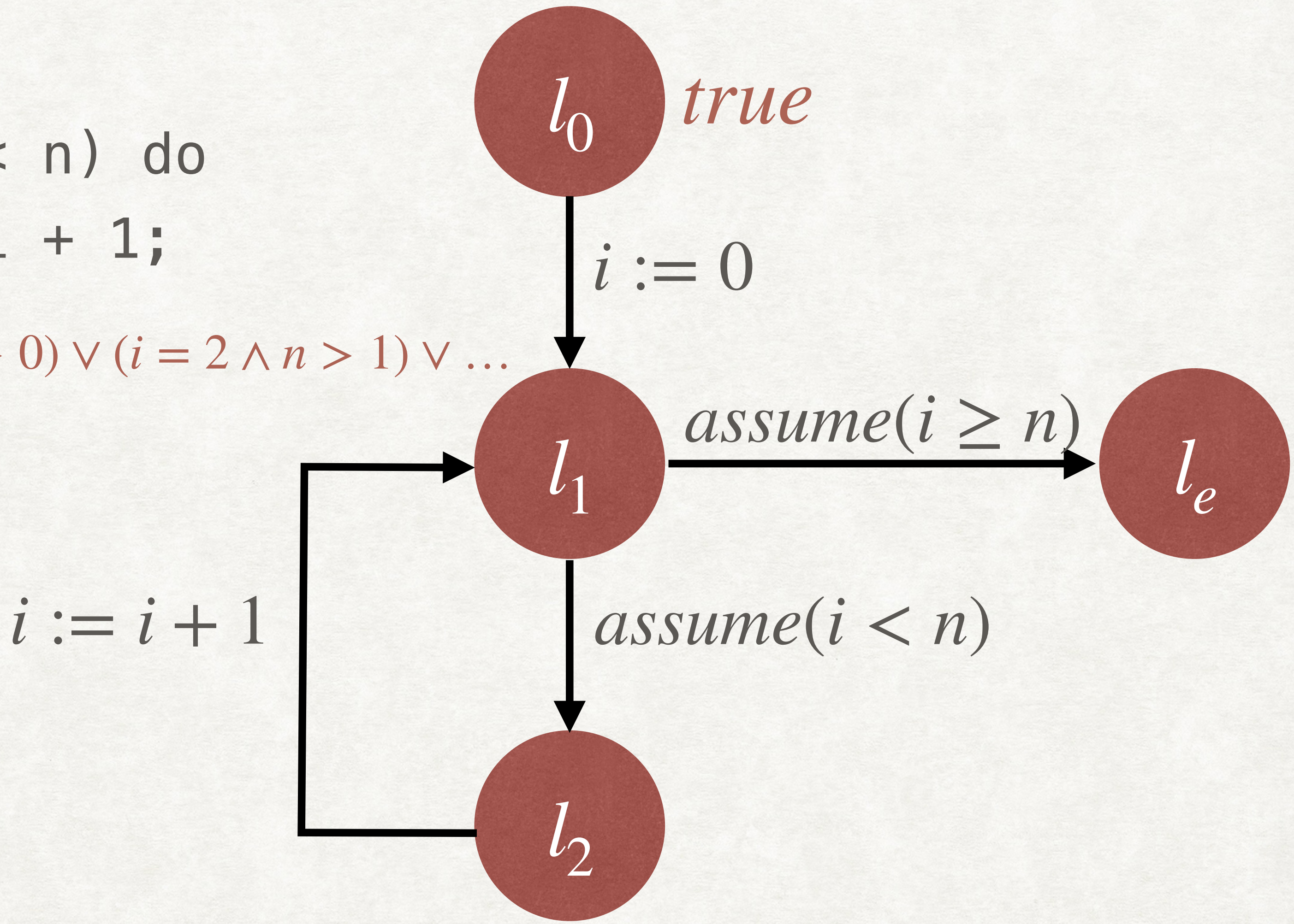
```
i := 0;
while(i < n) do
    i := i + 1;
```



EXAMPLE

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

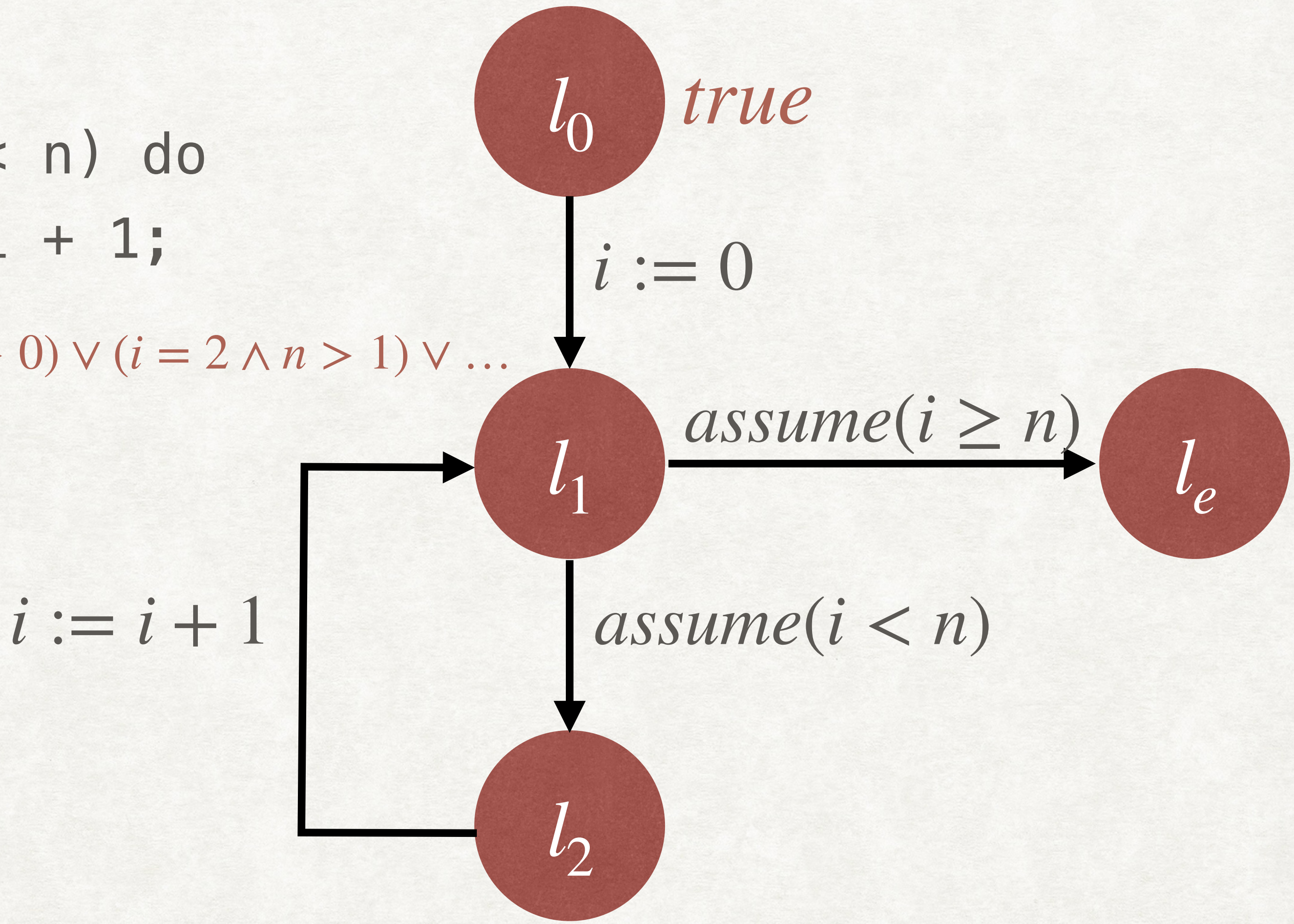
$(i = 0) \vee (i = 1 \wedge n > 0) \vee (i = 2 \wedge n > 1) \vee \ldots$



EXAMPLE

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

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



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(Γc,P)
  S := {l0};
  μ(l0) := P;
  μ(l) := ⊥, for l ∈ L\{l0};
  while S ≠ ∅ do{
     l := Choose S;
     S := S \ {l};
     foreach (l,c,l′) ∈ T do{
        F := sp(μ(l),c);
        if ¬(F → μ(l′)) then{
           μ(l′) := μ(l′) ∨ F;
           S := S ∪ {l′};
        }
     }
  }
```

ABSTRACT FORWARD PROPAGATE

```
AbstractForwardPropagate(Γ_c,P)
  S := {l_0};
  μ̂(l_0) := α(P);
  μ̂(l) := ⊥, for l ∈ L\{l_0};
  while S ≠ ∅ do{
     l := Choose S;
     S := S \ {l};
     foreach (l, c, l′) ∈ T do{
        F := sp̂(μ̂(l), c);
        if ¬(F ≤ μ̂(l′)) then{
           μ̂(l′) := μ̂(l′) ⊔ F;
           S := S ∪ {l′};
        }
     }
  }
```

ABSTRACT FORWARD PROPAGATE

```
AbstractForwardPropagate(Γ_c,P)
  S := {l_0};
  μ̂(l_0) := α(P);
  μ̂(l) := ⊥, for l ∈ L\{l_0};
  while S ≠ ∅ do{
      l := Choose S;
      S := S \ {l};
      foreach (l,c,l') ∈ T do{
          F := ŝp(μ̂(l),c);
          if ¬(F ≤ μ̂(l')) then{
              μ̂(l') := μ̂(l') ⊔ F;
              S := S ∪ {l'};
          }
      }
  }
```

Abstract Domain D

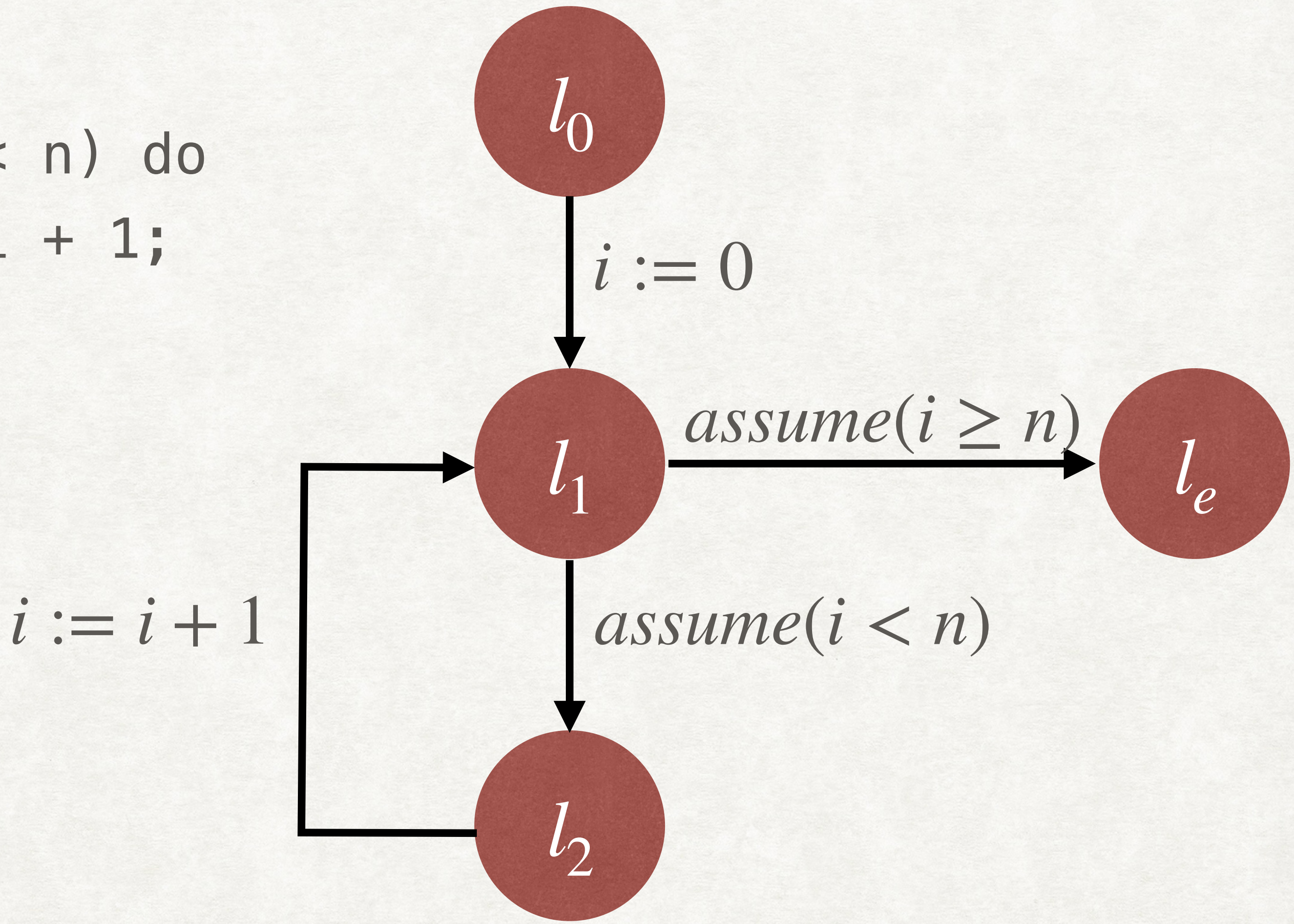
is a lattice $(D, \leq, \sqcup)$

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

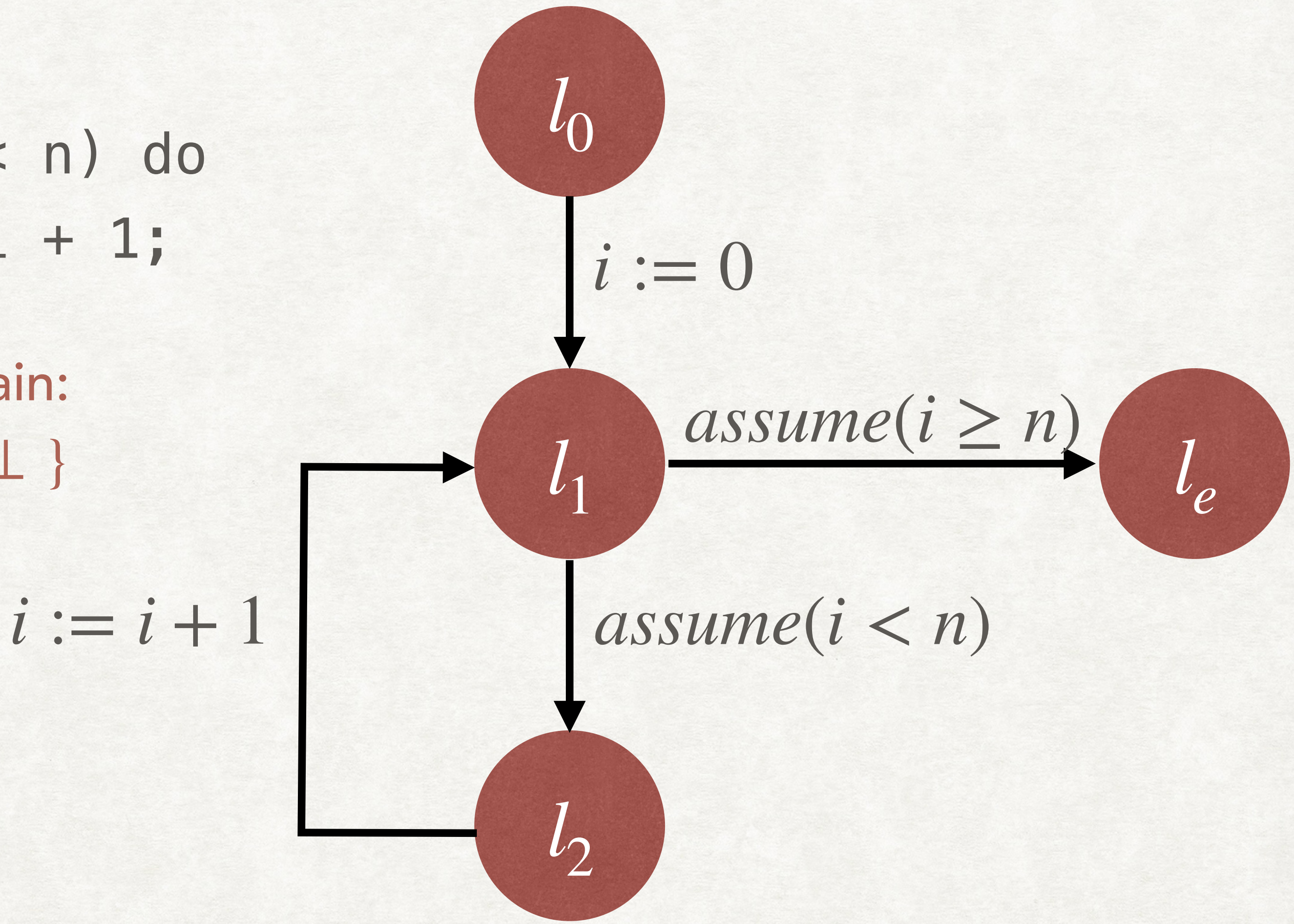
```
i := 0;
while(i < n) do
    i := i + 1;
```



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:

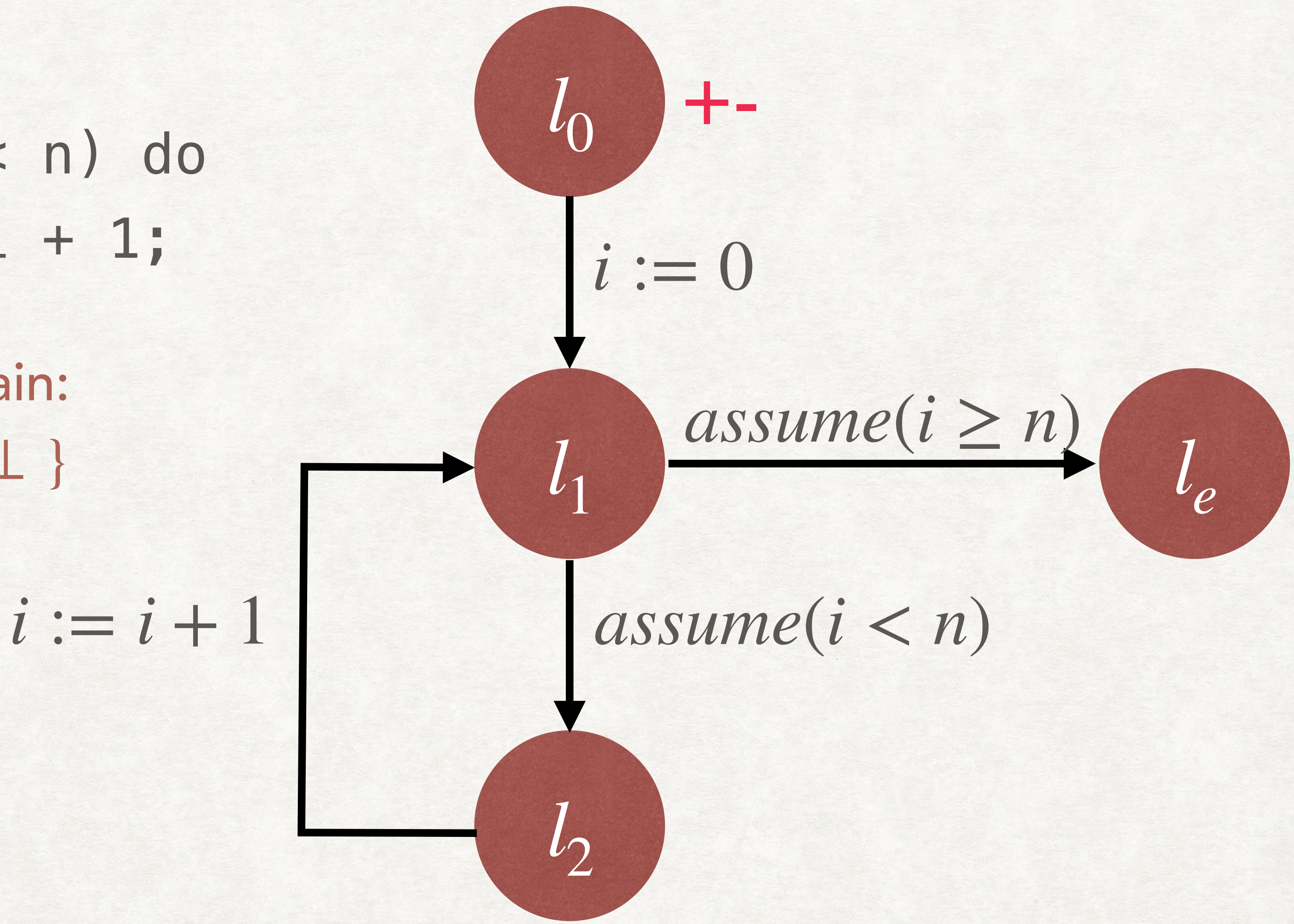
$D = \{ +-, +, -, \bot \}$

$\gamma(+-) = \top$

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

$\gamma(-) = i < 0$

$\gamma(\bot) = \bot$



EXAMPLE

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

Sign Abstract Domain:

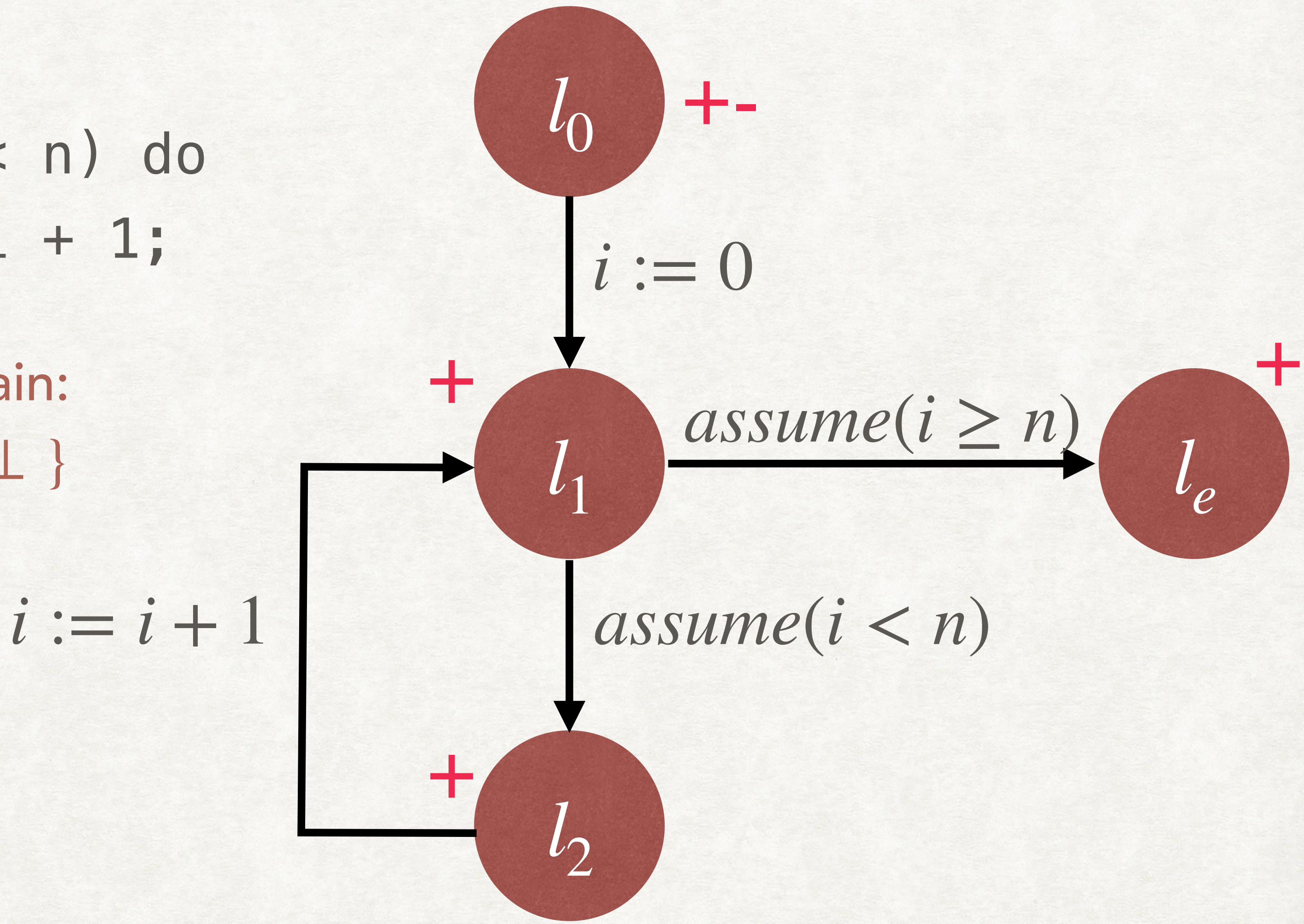
$D = \{ +-, +, -, \perp \}$

$\gamma(+-) = \top$

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

$\gamma(-) = i < 0$

$\gamma(\perp) = \perp$



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.
  - $(\mathbb{P}(State), \subseteq) \underset{\gamma}{\overset{\alpha}{\rightleftarrows}} (D, \leq)$ 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.

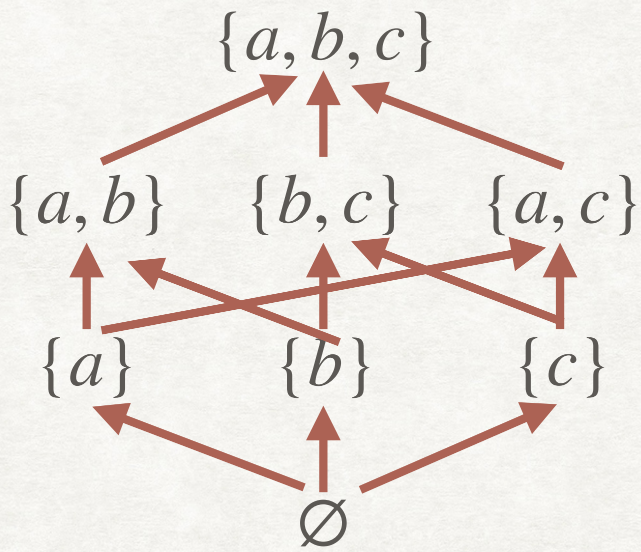
PARTIAL ORDER

- 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

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

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

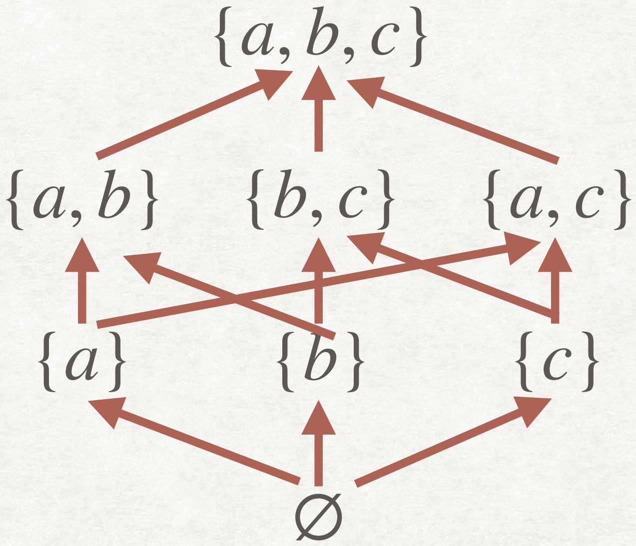


Partially Ordered Set: $(\mathbb{P}(S), \subseteq)$

PARTIAL ORDER - EXAMPLES

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

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



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.

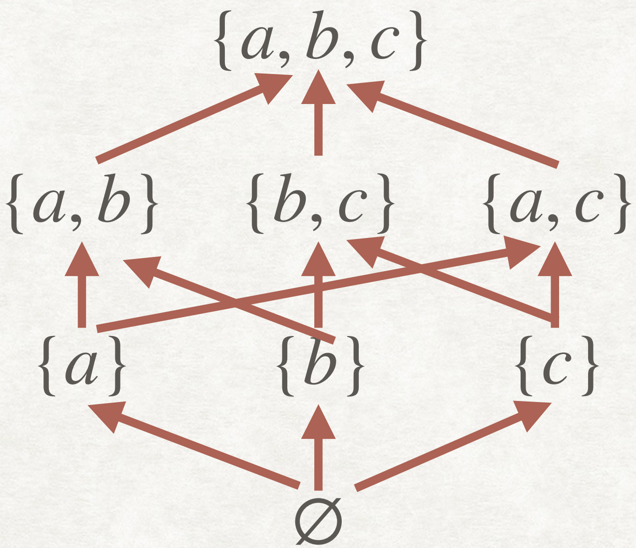
GREATEST LOWER BOUND

- 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.

LUB - EXAMPLE

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

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



- Consider $X = \{\{a\}, \{b\}\}$
- $\{a, b\}, \{a, b, c\}$ are both upper bounds of X
- $\{a, b\}$ is the least upper bound.

LATTICE

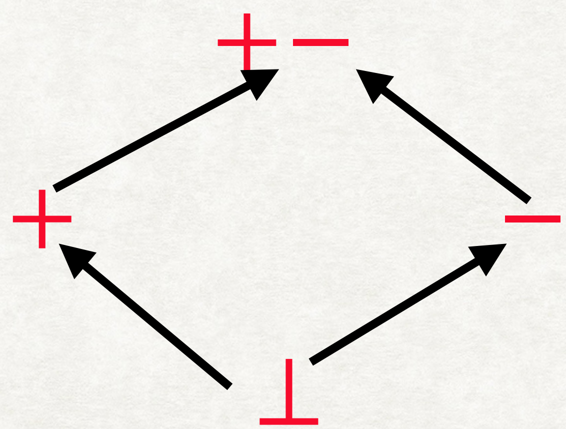
- A lattice is a poset $(D, \leq)$ such that $\forall x, y \in D$, $x \sqcup y$ and $x \sqcap y$ exist.
- A complete lattice is a lattice such that $\forall X \subseteq D$, $\sqcup X$ and $\sqcap X$ exists.
- Example: $(\mathbb{P}(S), \subseteq)$ is a complete lattice.

LATTICE - MORE EXAMPLES

- What is the simplest example of a poset that is not a lattice?
  - $(\{a, b\}, \{(a, a), (b, b)\})$
- What is an example of a lattice which is not a complete lattice?
  - $(\mathbb{N}, \leq)$

LATTICE - MORE EXAMPLES

- What is the simplest example of a poset that is not a lattice?
  - $(\{a, b\}, \{(a, a), (b, b)\})$
- What is an example of a lattice which is not a complete lattice?
  - $(\mathbb{N}, \leq)$
- Sign Lattice:



SOME PROPERTIES OF LATTICES

- $(D, \leq)$ is a lattice, $x, y, z \in D$
  - If $x \leq y$, then $x \sqcup y = y$ and $x \sqcap y = x$.
  - $x \sqcup x = x$ and $x \sqcap x = x$
  - $(x \sqcup y) \sqcup z = x \sqcup (y \sqcup z) = \sqcup \{x, y, z\}$
  - If D is finite, then D is also a complete lattice.

MINIMUM AND MAXIMUM

- Given a poset $(D, \leq)$, $x \in D$ is called the minimum element if $\forall y \in D \,.\, x \leq y$.
    - Also called the bottom element. Denoted by $\perp$.
- Given a poset $(D, \leq)$, $x \in D$ is called the maximum element if $\forall y \in D \,.\, y \leq x$.
    - Also called the top element. Denoted by $\top$.
- Complete lattices are guaranteed to have top and bottom elements.
    - $\sqcup D = \top \,, \sqcap D = \perp$
    - $\sqcup \varnothing = \perp \,, \sqcap \varnothing = \top$

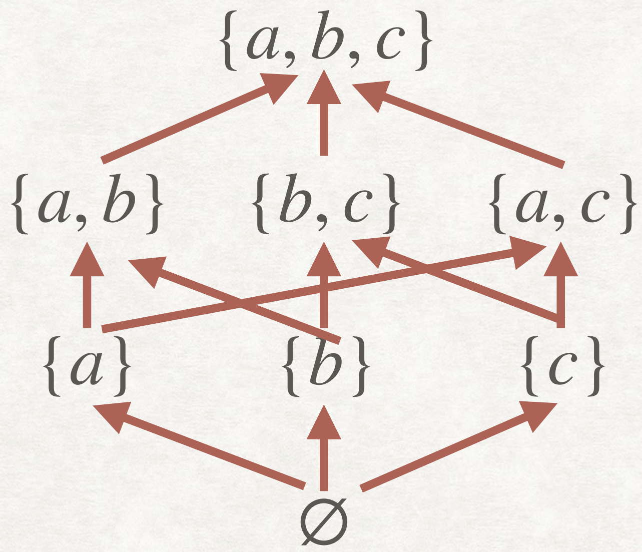
MONOTONIC FUNCTIONS

- Given two posets $(D_1, \leq_1)$ and $(D_2, \leq_2)$, function $f : D_1 \rightarrow D_2$ is called monotonic (or order-preserving) if
  - $\forall x, y \in D_1 \,.\, x \leq_1 y \rightarrow f(x) \leq_2 f(y)$
- In the special case when $D_1 = D_2 = D$, $f : D \rightarrow D$ is monotonic if
  - $\forall x, y \in D \,.\, x \leq y \rightarrow f(x) \leq f(y)$

MONOTONIC FUNCTIONS - EXAMPLE

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

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



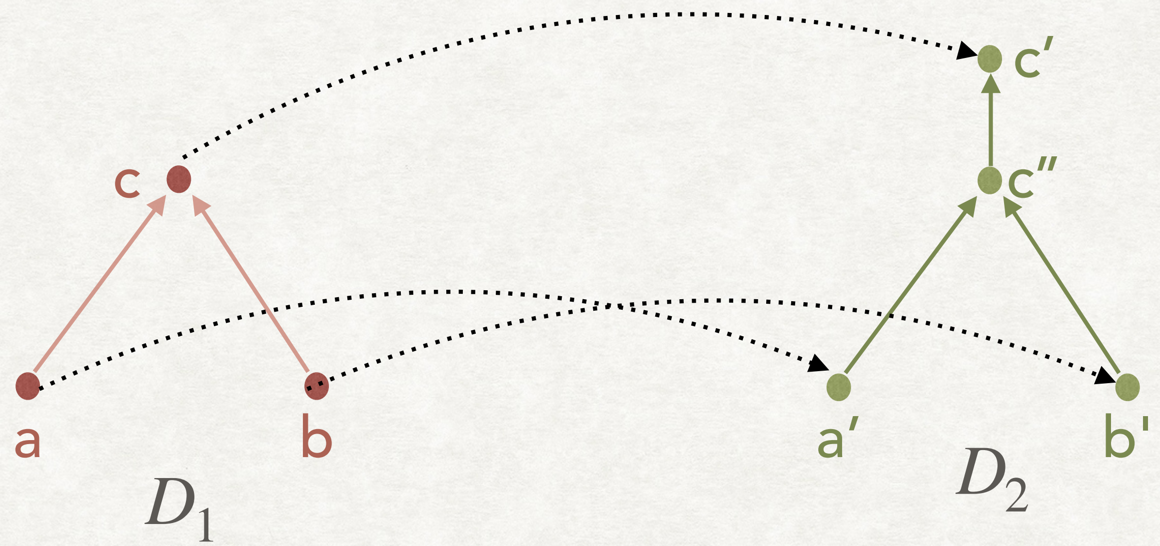
- Consider $f : \mathbb{P}(S) \to \mathbb{P}(S)$, $f(X) = X \cup \{a\}$.
  - f is monotonic.
- What about $f(X) = X \cap \{a\}$?
- Example of a non-monotonic function on $\mathbb{P}(S)$?

JOIN PRESERVING

- Given posets $(D_1, \leq_1)$ and $(D_2, \leq_2)$, a monotonic function $f: D_1 \to D_2$, and $S \subseteq D_1$, if $\sqcup_1 S$ and $\sqcup_2 f(S)$ exist, then $\sqcup_2 f(S) \leq_2 f(\sqcup_1 S)$.

JOIN PRESERVING

- Given posets $(D_1, \leq_1)$ and $(D_2, \leq_2)$, a monotonic function $f : D_1 \to D_2$, and $S \subseteq D_1$, if $\sqcup_1 S$ and $\sqcup_2 f(S)$ exist, then $\sqcup_2 f(S) \leq_2 f(\sqcup_1 S)$.



JOIN PRESERVING

- Given posets $(D_1, \leq_1)$ and $(D_2, \leq_2)$, a monotonic function $f: D_1 \to D_2$, and $S \subseteq D_1$, if $\sqcup_1 S$ and $\sqcup_2 f(S)$ exist, then $\sqcup_2 f(S) \leq_2 f(\sqcup_1 S)$.

Proof:

JOIN PRESERVING

- Given posets $(D_1, \leq_1)$ and $(D_2, \leq_2)$, a monotonic function $f : D_1 \rightarrow D_2$, and $S \subseteq D_1$, if $\sqcup_1 S$ and $\sqcup_2 f(S)$ exist, then $\sqcup_2 f(S) \leq_2 f(\sqcup_1 S)$.

Proof: Let $u = \sqcup_1 S$.

Then $\forall x \in S . x \leq_1 u$. This implies that $\forall x \in S . f(x) \leq_2 f(u)$.

Thus $f(u)$ is an upper bound of $f(S)$.

Hence, $\sqcup_2 f(S) \leq_2 f(u)$.