JOIN OVER PATHS

- Recall: Given a program as a LTS $\Gamma_c \equiv (V, L, l_0, l_e, T)$, the assertion map $\mu : L \rightarrow \mathbb{P}(State)$ associates a set of states with every location.
  - $\mu(l)$ is the set of states reachable at l during any execution.
  - μ is also called the Concrete Join Over Paths (JOP) or the collecting semantics.
- Instead of operating over concrete states, we can also consider JOP over abstract states.

ABSTRACT TRANSFER FUNCTION

- Given a Galois Connection $(\mathbb{P}(State), \subseteq) \underset{\gamma}{\overset{\alpha}{\rightleftarrows}} (D, \leq)$, for every program command p, we can define the abstract transfer function $\hat{f}_p$ (previously called the abstract strongest post-condition operator)
  - $\hat{f}_p : D \rightarrow D$.
- We can define the concrete transfer function as follows: $f_p(\sigma) = \{\sigma' \mid (\sigma, p) \hookrightarrow (\sigma', skip)\}$.
  - $f_p(c) = \bigcup_{\sigma \in c} f_p(\sigma)$
- Then, the abstract transfer function must be a consistent abstraction of the concrete transfer function:
  - $\forall d \in D \,.\, f_p(\gamma(d)) \subseteq \gamma(\hat{f}_p(d))$
  - Equivalently, $\forall c \in \mathbb{P}(State) \,.\, \hat{f}_p(\alpha(c)) \leq \alpha(f(c))$

ABSTRACT TRANSFER FUNCTION

EXAMPLE

- Consider the sign abstract domain, and the program command p : x := x+1.
  - $\hat{f}_p(+) = ???$

ABSTRACT TRANSFER FUNCTION

EXAMPLE

- Consider the sign abstract domain, and the program command p : x := x+1.
    - $\hat{f}_p(+) = +$

ABSTRACT TRANSFER FUNCTION

EXAMPLE

- Consider the sign abstract domain, and the program command p : x := x+1.
  - $\hat{f}_p(+) = +$
  - $\hat{f}_p(-) = ???$

ABSTRACT TRANSFER FUNCTION

EXAMPLE

- Consider the sign abstract domain, and the program command p : x := x+1.
  - $\hat{f}_p(+) = +$
  - $\hat{f}_p(-) = + -$

ABSTRACT TRANSFER FUNCTION

EXAMPLE

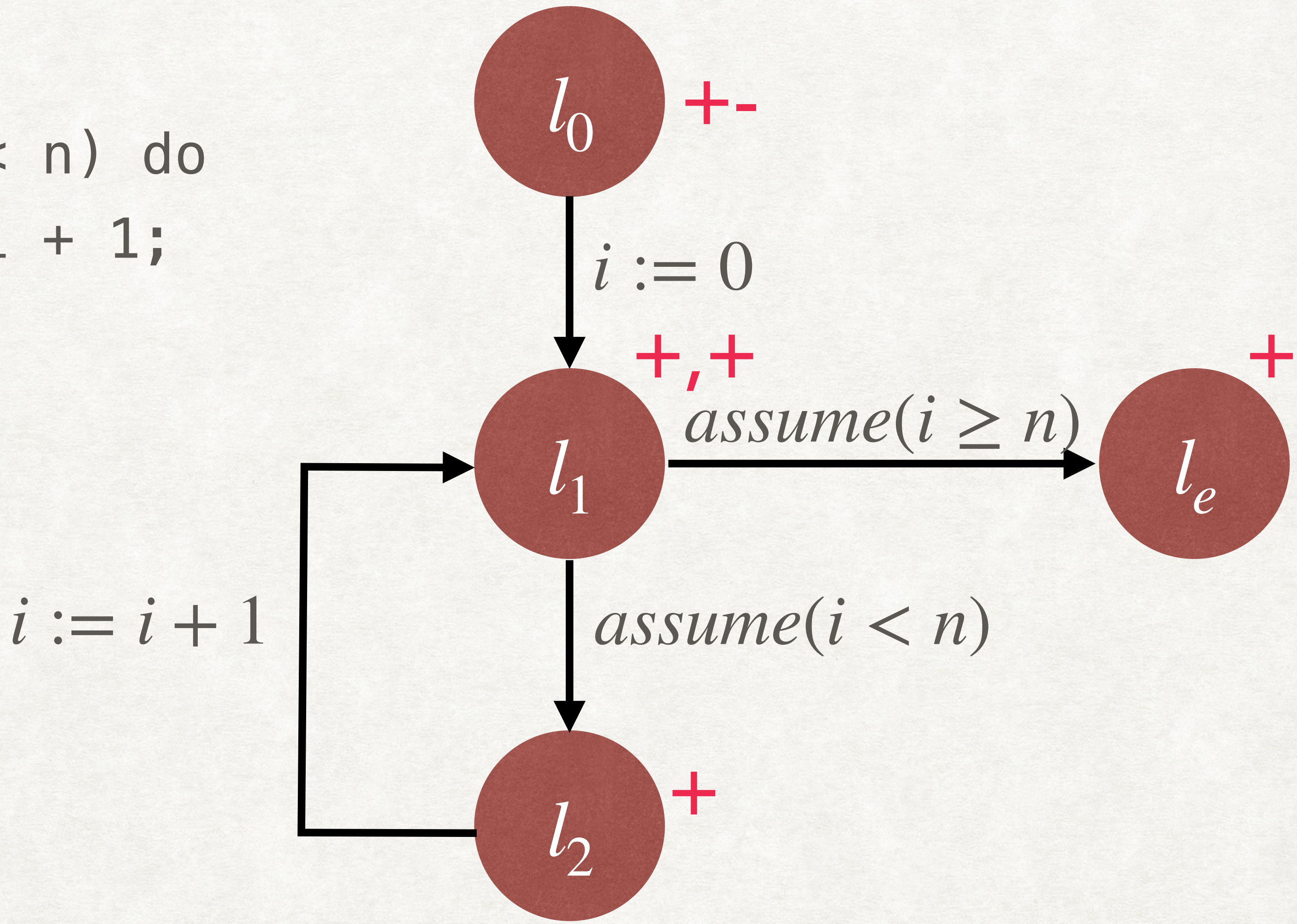
- Consider the sign abstract domain, and the program command p : x := x+1.
  - $\hat{f}_p(+) = +$
  - $\hat{f}_p(-) = +-$
  - $\hat{f}_p(+-) = +-$
  - $\hat{f}_p(\bot) = \bot$
- See whether the condition $\forall d \in D \,.\, f_p(\gamma(d)) \subseteq \gamma(\hat{f}_p(d))$ is satisfied.

ABSTRACT JOP

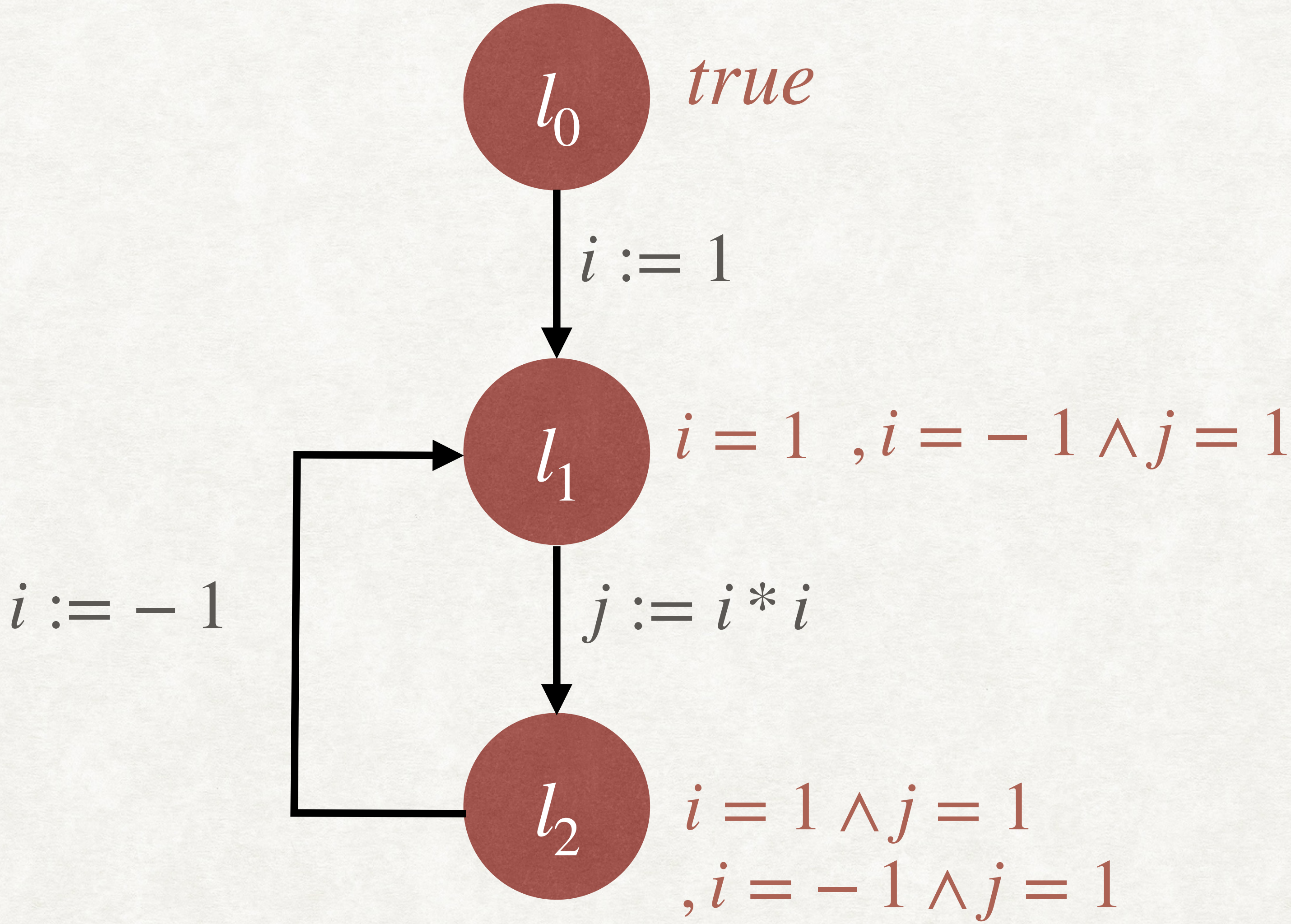
- Instead of executing the program with concrete states, we execute the program with abstract state, and the abstract transfer function for each program command.
- Collect all the abstract states at each location, for every possible execution
    - Their join is the abstract JOP map, $\hat{\mu} : L \to D$.

EXAMPLE

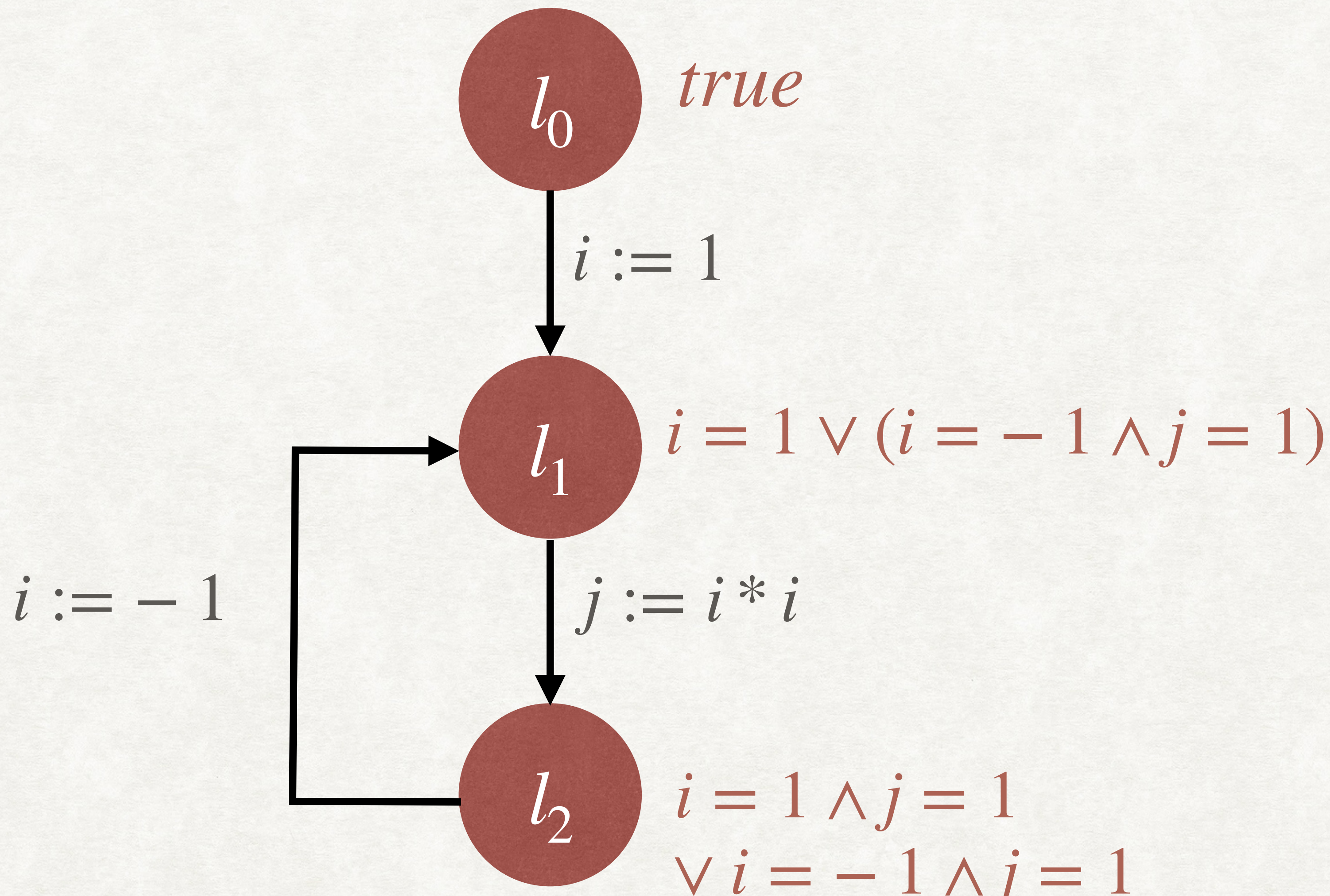
```
i := 0;
while(i < n) do
    i := i + 1;
```



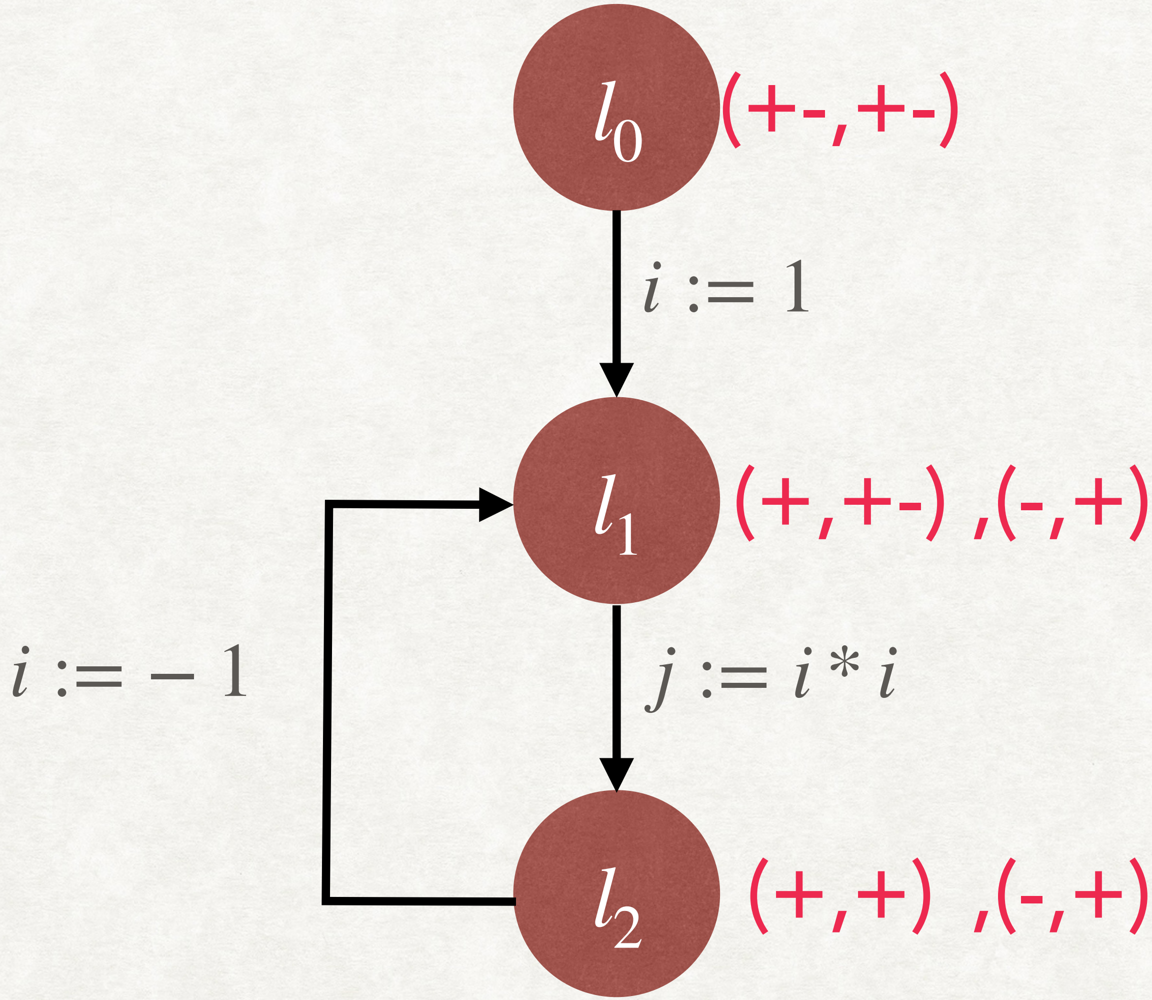
EXAMPLE - COLLECTING SEMANTICS



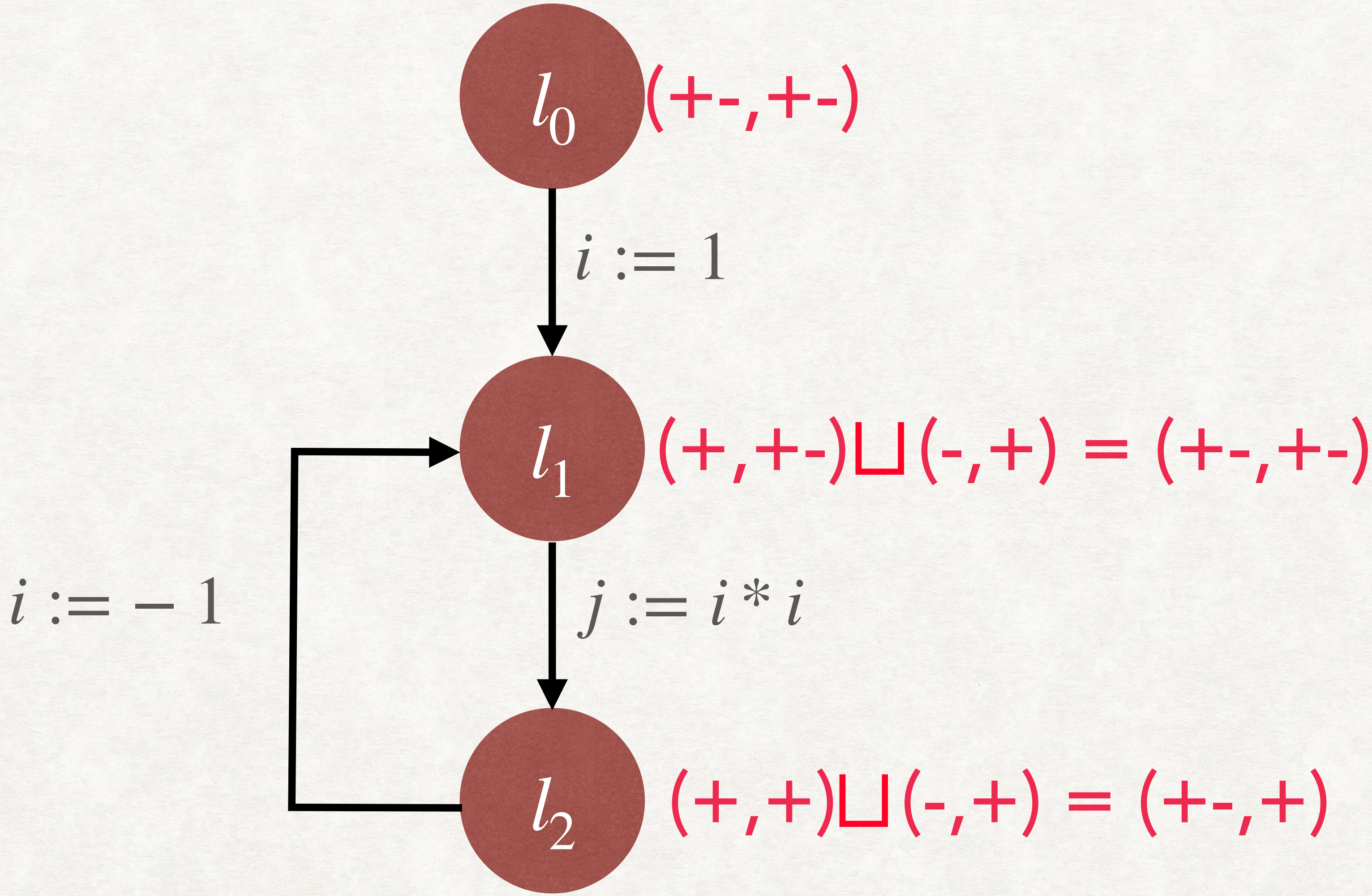
EXAMPLE - COLLECTING SEMANTICS



EXAMPLE - ABSTRACT JOP



EXAMPLE - ABSTRACT JOP



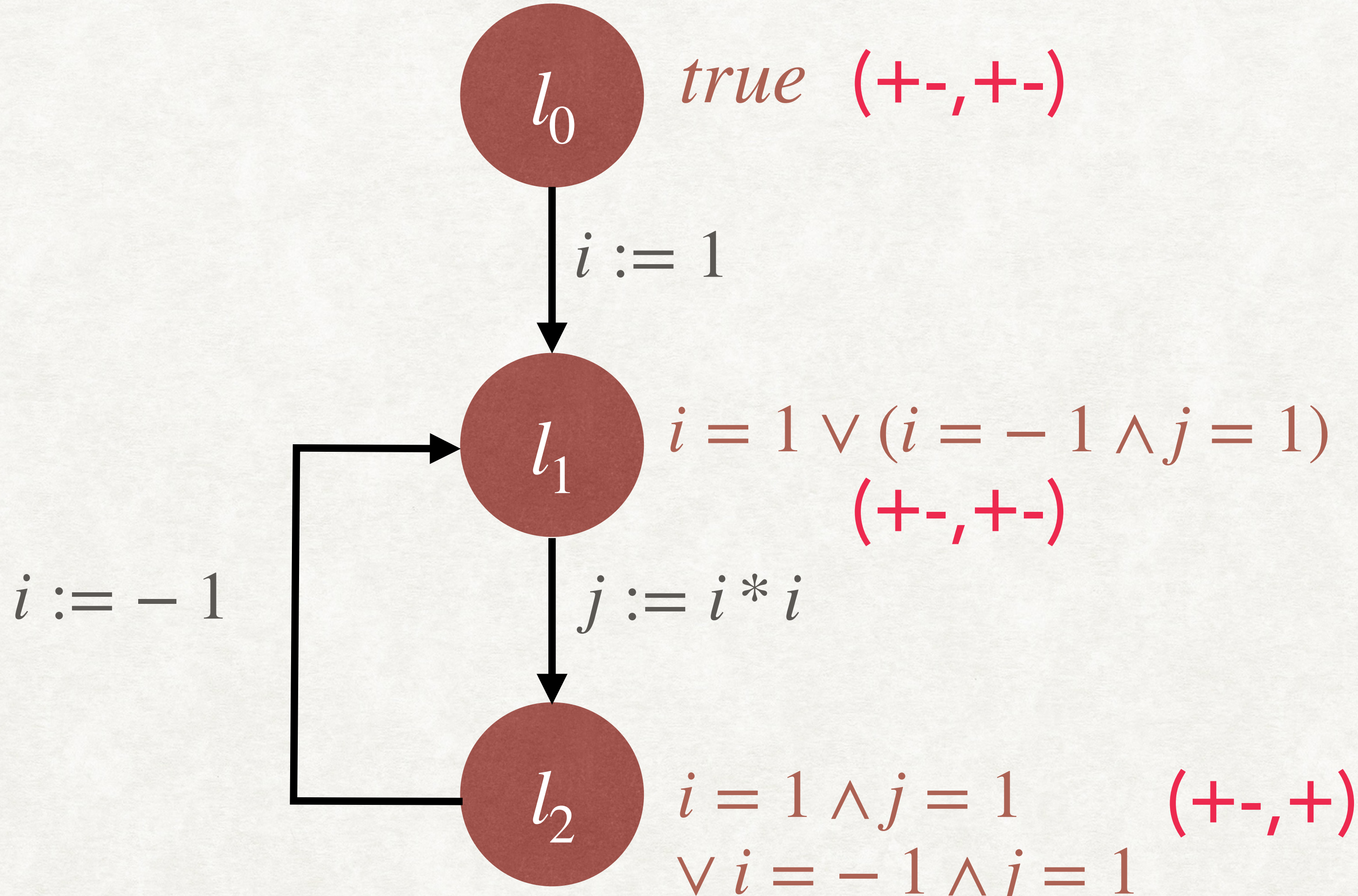
SOUNDNESS OF ABSTRACT INTERPRETATION

DEFINITION

- A given abstract interpretation (consisting of the abstract domain $(D, \leq)$, (α, γ), and abstract transfer functions $\hat{F}_D$) is sound, if for all $d_0 \in D$, assuming that $\hat{\mu}(l_0) = d_0$, the γ image of the abstract JOP $\hat{\mu}$ at all locations over approximates the collecting semantics μ, assuming that $\mu(l_0) = c_0$ where $c_0 \subseteq \gamma(d_0)$.
    - For all locations l, $\gamma(\hat{\mu}(l)) \supseteq \mu(l)$.

SOUNDNESS OF ABSTRACT INTERPRETATION

EXAMPLE



FROM ABSTRACT INTERPRETATION TO VERIFICATION

- In order to show the validity of the Hoare Triple $\{P\}c\{Q\}$, we instantiate a sound AI $(D, \leq, \alpha, \gamma, \hat{F}_D)$ with $\hat{\mu}(l_0) = d_0$, such that $\alpha(P) \leq d_0$ and compute the resulting JOP $\hat{\mu}$ at all locations.
- If $\gamma(\hat{\mu}(l_e)) \subseteq Q$, then the Hoare Triple is valid.
    - Since $\alpha(P) \leq d_0$, by definition of Galois connection, $P \subseteq \gamma(d_0)$.
    - Hence, by definition of soundness of AI, $\mu(l_e) \subseteq \gamma(\hat{\mu}(l_e))$, where μ is the collecting semantics assuming $\mu(l_0) = P$.

SOUNDNESS OF ABSTRACT INTERPRETATION

SUFFICIENT CONDITIONS

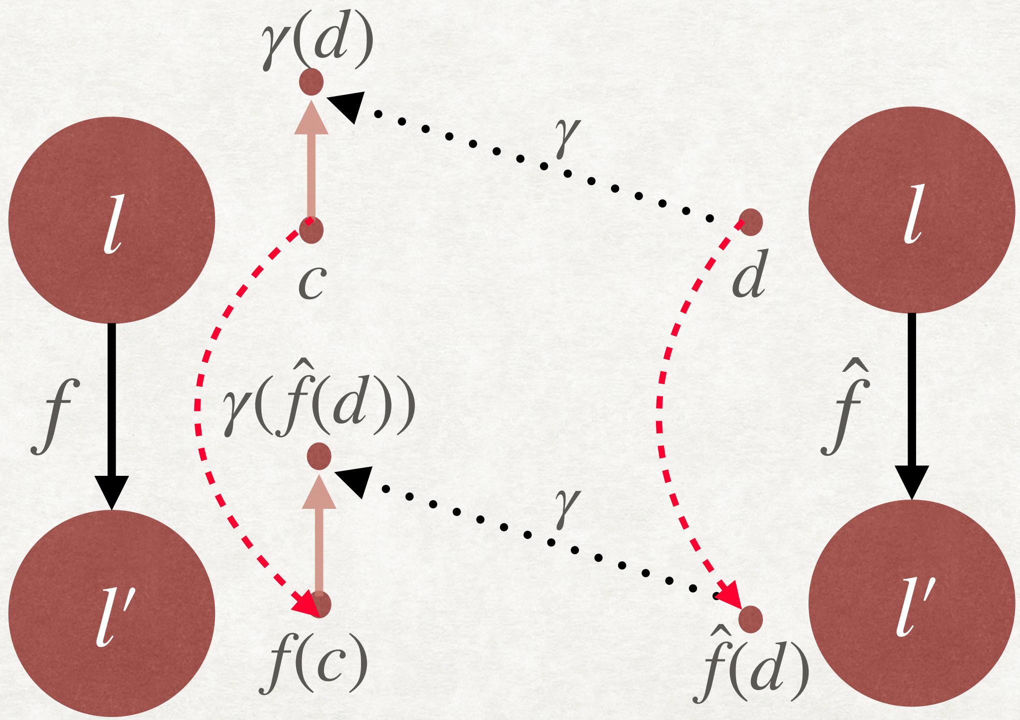
- An abstract interpretation $(D, \leq, \alpha, \gamma, \hat{F}_D)$ is sound if:
  - $(D, \leq)$ is complete lattice.
  - $(\mathbb{P}(State), \subseteq) \underset{\gamma}{\overset{\alpha}{\rightleftarrows}} (D, \leq)$
  - Every abstract transfer function in $\hat{F}_D$ is a consistent abstraction of the corresponding concrete transfer function.

PROOF OF SOUNDNESS OF AI

- Lemma-1: First, let us show that for any abstract transfer function $\hat{f} \in \hat{F}_D$ which is a consistent abstraction of concrete transfer function f, the following holds:
  - $\forall c \in \mathbb{P}(State) \,.\, \forall d \in D \,.\, c \subseteq \gamma(d) \Rightarrow f(c) \subseteq \gamma(\hat{f}(d))$

PROOF OF SOUNDNESS OF AI

- Lemma-1: First, let us show that for any abstract transfer function $\hat{f} \in \hat{F}_D$ which is a consistent abstraction of concrete transfer function f, the following holds:
  - $\forall c \in \mathbb{P}(State) \,.\, \forall d \in D \,.\, c \subseteq \gamma(d) \Rightarrow f(c) \subseteq \gamma(\hat{f}(d))$



PROOF OF SOUNDNESS OF AI

- Lemma-1: First, let us show that for any abstract transfer function $\hat{f} \in \hat{F}_D$ which is a consistent abstraction of concrete transfer function f, the following holds:
  - $\forall c \in \mathbb{P}(State) \,.\, \forall d \in D \,.\, c \subseteq \gamma(d) \Rightarrow f(c) \subseteq \gamma(\hat{f}(d))$

Proof: Consider $c \in \mathbb{P}(State), d \in D$ such that $c \subseteq \gamma(d)$.

Note that f is monotonic. (Why?)

Hence, $f(c) \subseteq f(\gamma(d))$.

Since $\hat{f}$ is a consistent abstraction of f, $f(\gamma(d)) \subseteq \gamma(\hat{f}(d))$.

Hence, $f(c) \subseteq \gamma(\hat{f}(d))$.

PROOF OF SOUNDNESS OF AI

CONCRETE AND ABSTRACT JOP

- Given a path $\pi : l_0 \xrightarrow{p_0} l_1 \xrightarrow{p_1} \dots \xrightarrow{p_{n-1}} l_n$ in the program LTS, the combined abstract transfer function $\hat{f}_\pi$ is the composition of the individual transfer functions: $\hat{f}_{p_{n-1}} \circ \dots \circ \hat{f}_{p_1} \circ \hat{f}_{p_0}$
  - Similarly, the concrete transfer function f_π is $f_{p_{n-1}} \circ \dots \circ f_{p_1} \circ f_{p_0}$
- Let Π_l be the set of all possible paths from l_0 to l.
- Assuming that $\hat{\mu}(l_0) = d_0$, the abstract JOP at a location l is given by:
  - $\hat{\mu}(l) = \bigsqcup_{\pi \in \Pi_l} \hat{f}_\pi(d_0)$
- Similarly, assuming $\mu(l_0) = c_0$ the concrete JOP, $\mu(l) = \bigsqcup_{\pi \in \Pi_l} f_\pi(c_0)$

PROOF OF SOUNDNESS OF AI

- Lemma-2: Assuming that $c_0 \subseteq \gamma(d_0)$, we will show that for any location l and path $\pi \in \Pi_l$, $f_\pi(c_0) \subseteq \gamma(\hat{f}_\pi(d_0))$.

Proof: We will use induction to show that for any $i \geq 0$, π_i which is the prefix of π of length i, $f_{\pi_i}(c_0) \subseteq \gamma(\hat{f}_{\pi_i}(d_0))$.

Base Case: For $i = 0$, we are already given that $c_0 \subseteq \gamma(d_0)$.

Inductive Case: The inductive hypothesis is that $f_{\pi_i}(c_0) \subseteq \gamma(\hat{f}_{\pi_i}(d_0))$.

Consider π_{i+1}. Let the $(i+1)$th edge in the path be labelled by program command p .

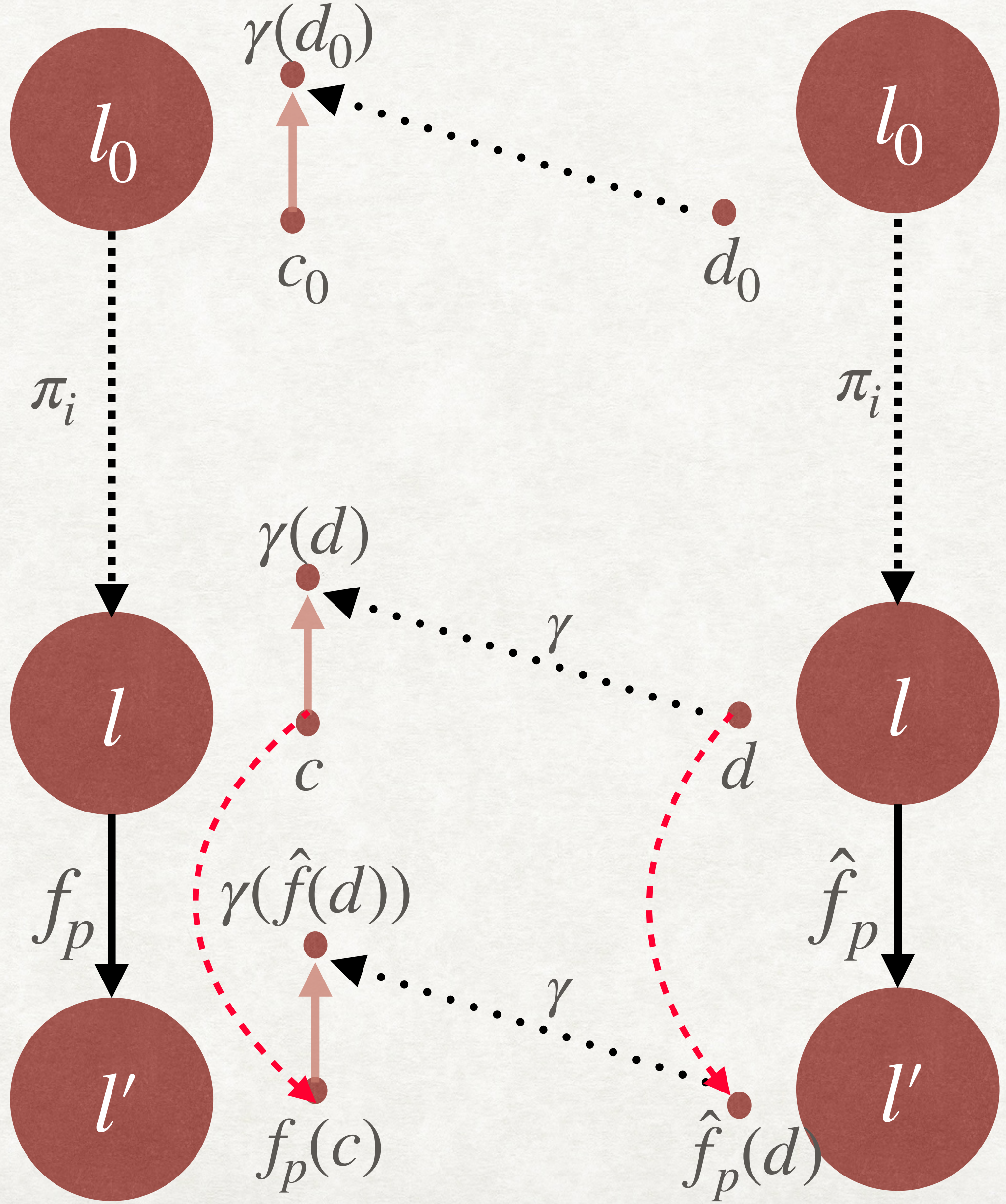
Then, $f_{\pi_{i+1}} = f_p \circ f_{\pi_i}$ and $\hat{f}_{\pi_{i+1}} = \hat{f}_p \circ \hat{f}_{\pi_i}$.

Let $f_{\pi_i}(c_0) = c$ and $\hat{f}_{\pi_i}(d_0) = d$. We have $c \subseteq \gamma(d)$ and $\hat{f}_p$ is a consistent abstraction of f_p. Hence, by Lemma-1, $f_p(c) \subseteq \gamma(\hat{f}_p(d))$.

This proves that $f_{\pi_{i+1}}(c_0) \subseteq \gamma(\hat{f}_{\pi_{i+1}}(d_0))$.

PROOF OF SOUNDNESS OF AI

LEMMA-2



PROOF OF SOUNDNESS OF AI

- Finally, we will show that for any location l, $\bigsqcup_{\pi \in \Pi_l} f_\pi(c_0) \subseteq \gamma(\bigsqcup_{\pi \in \Pi_l} \hat{f}_\pi(d_0))$, assuming that $c_0 \subseteq \gamma(d_0)$.

Proof: By Lemma-2, we know that $\forall \pi \in \Pi_l \,.\, f_\pi(c_0) \subseteq \gamma(\hat{f}_\pi(d_0))$.

Hence, $\bigsqcup_{\pi \in \Pi_l} f_\pi(c_0) \subseteq \bigsqcup_{\pi \in \Pi_l} \gamma(\hat{f}_\pi(d_0))$. Why?

[$\bigsqcup_{\pi \in \Pi_l} \gamma(\hat{f}_\pi(d_0)) \supseteq \gamma(\hat{f}_\pi(d_0)) \supseteq f_\pi(c_0)$. Hence, $\bigsqcup_{\pi \in \Pi_l} \gamma(\hat{f}_\pi(d_0))$ is an upper bound of $\{f_\pi(c_0) \,|\, \pi \in \Pi_l\}$.]

PROOF OF SOUNDNESS OF AI

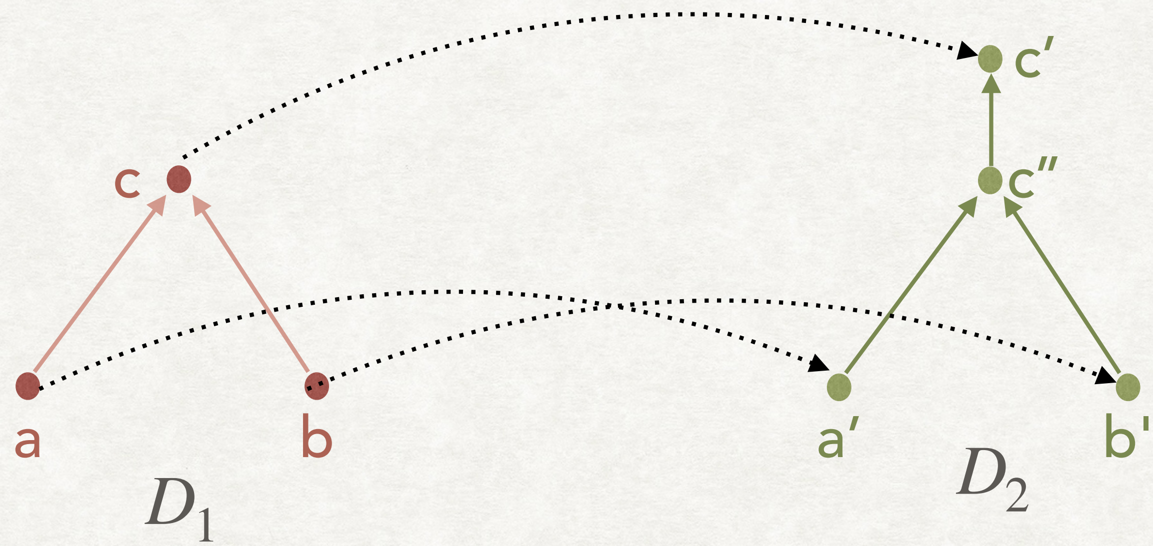
- Finally, we will show that for any location l, $\bigsqcup_{\pi \in \Pi_l} f_\pi(c_0) \subseteq \gamma(\bigsqcup_{\pi \in \Pi_l} \hat{f}_\pi(d_0))$, assuming that $c_0 \subseteq \gamma(d_0)$.

Proof: By Lemma-2, we know that $\forall \pi \in \Pi_l \,.\, f_\pi(c_0) \subseteq \gamma(\hat{f}_\pi(d_0))$.

Hence, $\bigsqcup_{\pi \in \Pi_l} f_\pi(c_0) \subseteq \bigsqcup_{\pi \in \Pi_l} \gamma(\hat{f}_\pi(d_0))$.

RECALL: 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 OF SOUNDNESS OF AI

- Finally, we will show that for any location l, $\bigsqcup_{\pi \in \Pi_l} f_\pi(c_0) \subseteq \gamma(\bigsqcup_{\pi \in \Pi_l} \hat{f}_\pi(d_0))$, assuming that $c_0 \subseteq \gamma(d_0)$.

Proof: By Lemma-2, we know that $\forall \pi \in \Pi_l \,.\, f_\pi(c_0) \subseteq \gamma(\hat{f}_\pi(d_0))$.

Hence, $\bigsqcup_{\pi \in \Pi_l} f_\pi(c_0) \subseteq \bigsqcup_{\pi \in \Pi_l} \gamma(\hat{f}_\pi(d_0))$.

We know that γ is monotonic and $(D, \leq)$ is a complete lattice, so that $\bigsqcup_{\pi \in \Pi_l} \hat{f}_\pi(d_0)$ exists. Hence, by the join-preserving property,

$\bigsqcup_{\pi \in \Pi_l} \gamma(\hat{f}_\pi(d_0)) \subseteq \gamma(\bigsqcup_{\pi \in \Pi_l} \hat{f}_\pi(d_0))$. Hence, $\bigsqcup_{\pi \in \Pi_l} f_\pi(c_0) \subseteq \gamma(\bigsqcup_{\pi \in \Pi_l} \hat{f}_\pi(d_0))$

ABSTRACT TRANSFER FUNCTION

SIGN ABSTRACT DOMAIN

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

p : x := e

$\hat{f}_p(d) \triangleq d[x \to g(d, e)]$

$$g(d, e_1 + e_2) = \begin{cases} + & \text{if } g(d, e_1) = + \text{ and } g(d, e_2) = + \\ - & \text{if } g(d, e_1) = - \text{ and } g(d, e_2) = - \\ +- & \text{otherwise} \end{cases}$$

$$g(d, e_1 - e_2) = \begin{cases} + & \text{if } g(d, e_1) = + \text{ and } g(d, e_2) = - \\ - & \text{if } g(d, e_1) = - \text{ and } g(d, e_2) = + \\ +- & \text{otherwise} \end{cases}$$

$g(d, \text{y}) = d(\text{y})$ if y is a program variable

$\hat{f}_p(\bot) = \bot$