Abstract Interpretation, cont.

Announcements

- HW3 and HW4?
  - I will extend deadline and adjust schedule
  - Office hours tomorrow:
    - Linh: 1-3pm in GREENE 120
    - Ana: 4-5pm on Webex
- HW5
  - Abstract interpretation and Haskell
  - Download and get started with Haskell

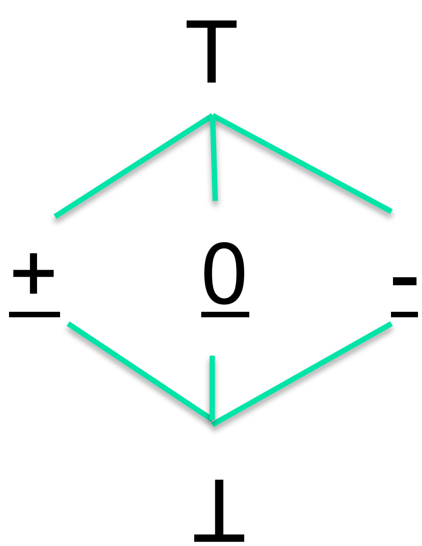
Outline

- Overview
- Semantics
- Notion of abstraction
- Concretization and abstraction functions
- Galois Connections
- Applications of abstract interpretation

Abstraction Example 1: signs

- Concrete values: sets of integers
- Abstract values: signs

Lattice of signs:



- ⊥ represents the empty set
- + represents any set of positive integers
- 0 represents set **{ 0 }**
- - represents any set of negative integers
- T represents any set of integers

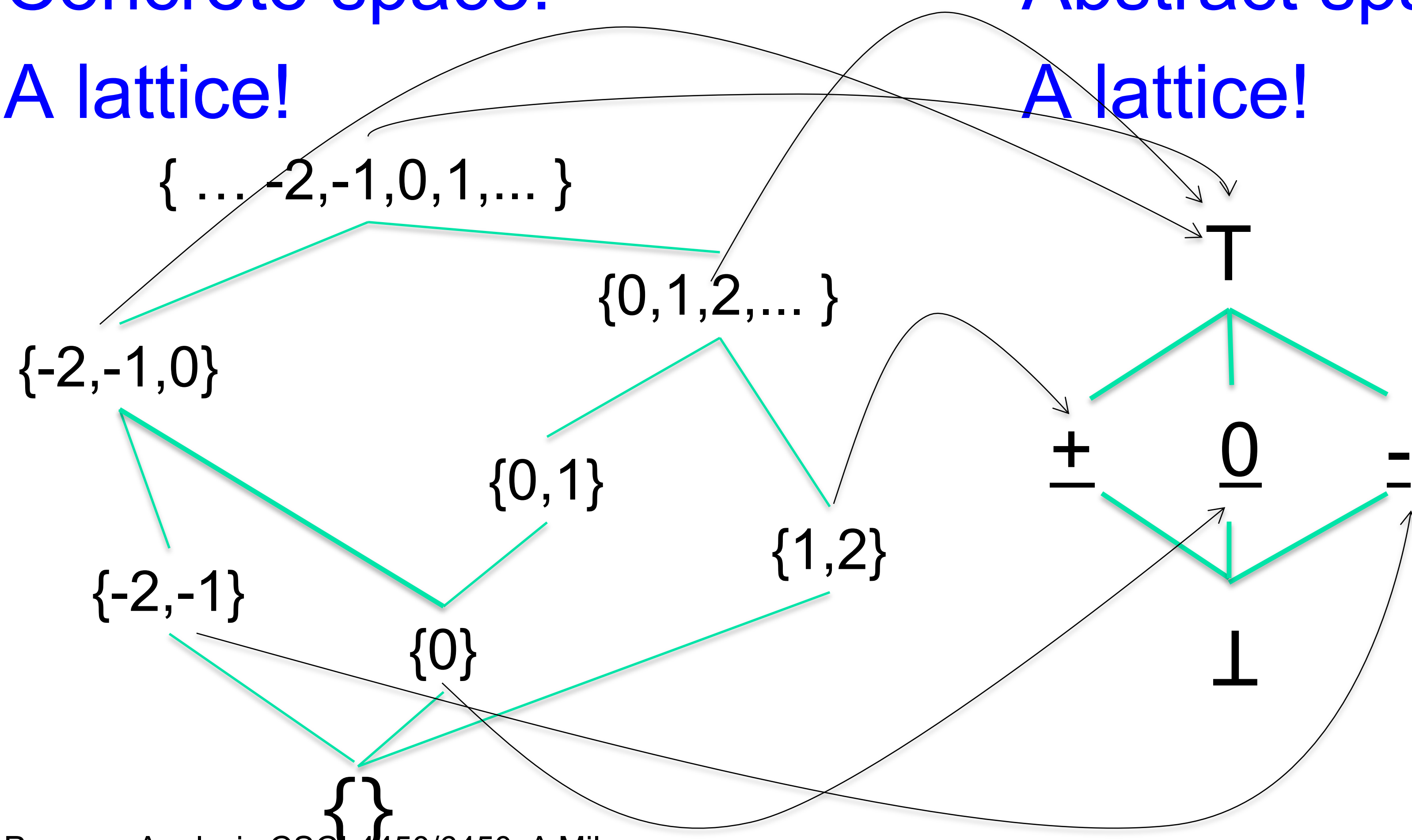
Abstraction Example 1: signs

Concrete space:

A lattice!

Abstract space:

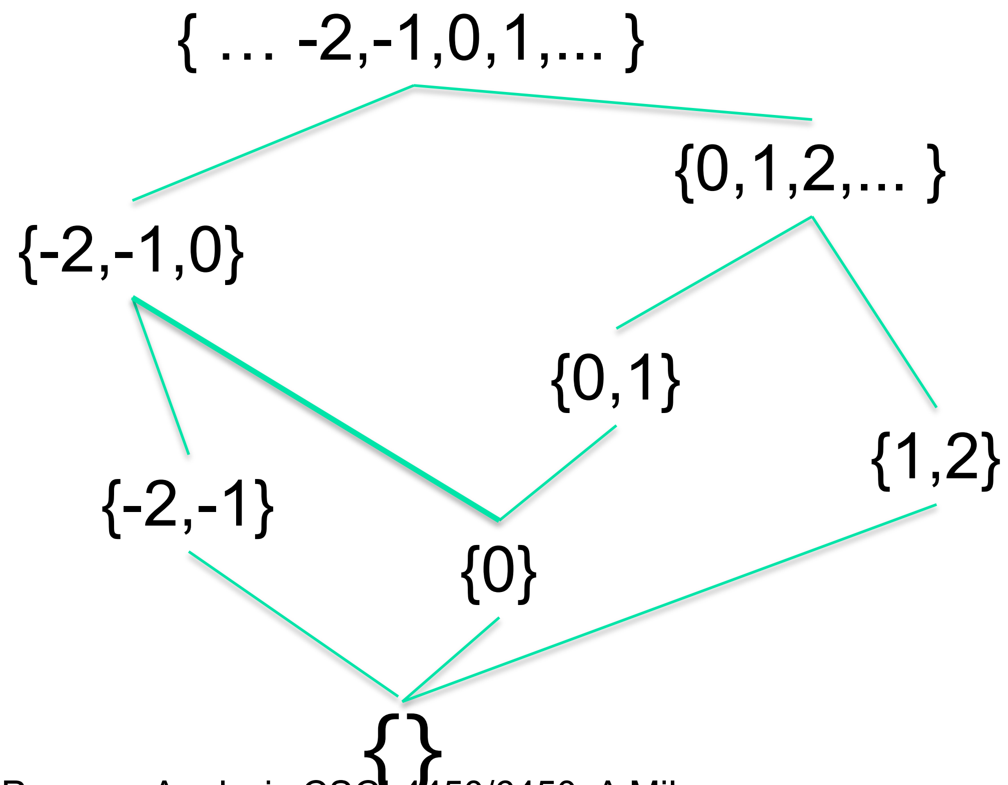
A lattice!



Abstraction Example 1: signs

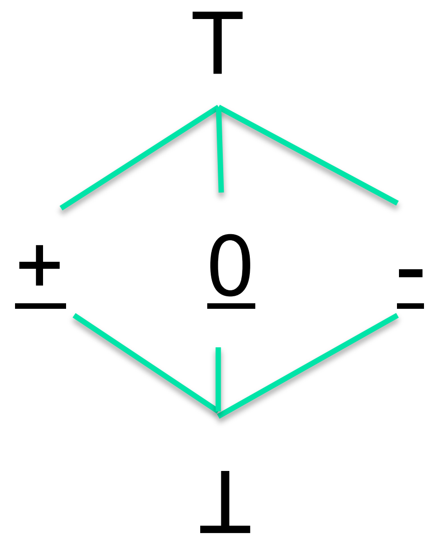
Concrete space:

A lattice!



Abstract space:

A lattice!



Abstraction Example 1: signs

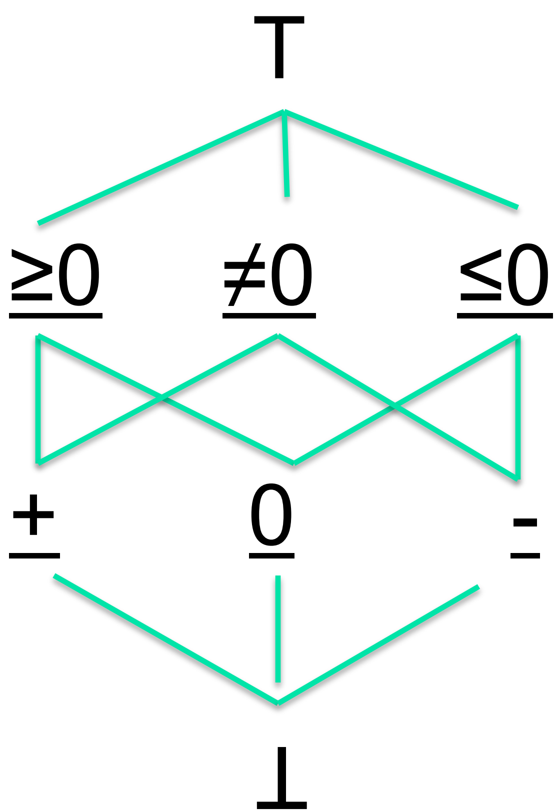
- Concrete elements: elements of the concrete lattice $\mathbf{c} \in \mathbf{2^Z}$
- Abstract elements: elements of abstract lattice of signs
- Abstraction relation relates concrete elements to abstract ones: $\mathbf{c} \vdash_S \mathbf{a}$ (i.e., **a** represents **c**, or conversely **c** is represented by **a**)

$$\{1,2,3\} \vdash_S \underline{+}$$

$$\{1,2,3\} \vdash_S \mathrm{T}$$

Abstraction Example 1: signs

- We can refine the abstract space

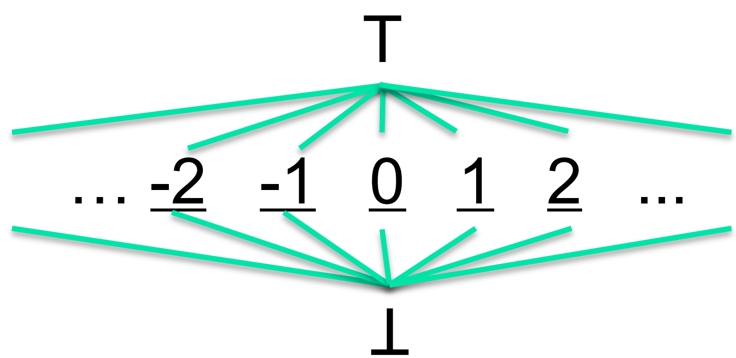


- ⊥ represents the empty set
- + represents any set of positive integers
- 0 represents set { 0 }
- - represents any set of negative integers
- T represents any set of integers

- ≥0 represents any set of non-negative integers
- ≤0 represents any set of non-positive integers
- ≠0 represents any set of non-zero integers

Abstraction Example 2: constants

- Concrete elements: elements of concrete lattice, $\mathbf{c} \in \mathbf{2^Z}$
- Abstract elements: $\perp$, T, $\underline{\mathbf{n}}$ **,** where $\mathbf{n} \in \mathbf{Z}$
  - Flat lattice:



- Abstraction relation:
  - **{n}** is represented by $\underline{\mathbf{n}}$ and by T
  - empty set is represented by $\perp$, any $\underline{\mathbf{n}}$, and by T
  - an arbitrary set of integers is represented by T

Abstraction Example 2: constants

- Abstract semantics, works on abstract elements (the elements of the flat lattice)
- If **x** is $\underline{\mathbf{n_1}}$ and **y** is $\underline{\mathbf{n_2}}$ then **x + y** is $\underline{\mathbf{n_1+n_2}}$
  - $\underline{\mathbf{n_1}}$ represents integer $\mathbf{n_1}$,
  - $\underline{\mathbf{n_2}}$ represents $\mathbf{n_2}$,
  - then $\underline{\mathbf{n_1+n_2}}$ $\mathbf{n_1} + \mathbf{n_2}$
- If **x** is $\underline{\mathbf{n_1}}$ and **y** is T, then what is **x + y**? T

Abstraction Example 3: intervals $\mathcal{I}$

- Concrete elements: S $\in \mathbf{2^Z}$
- Abstract elements: $\perp$, T, intervals [**a**,**b**] where **a** $\in$ **Z**U**{−∞}** and **b** $\in$ **Z**U**{∞}** and **a ≤ b**
  - Is it a lattice?
  - Yes!

$T = [-\infty, \infty]$

$\ldots \quad [-1,0] \quad [0,1] \quad \ldots$

$\ldots \quad [-1,-1] \quad [0,0] \quad [1,1] \quad [2,2] \quad \ldots$

$\perp$

$\{5\} \vdash_{\mathcal{I}} [5,5]$

$\vdash_{\mathcal{I}} [-10,10]$

- Abstraction relation:

$\{\} \vdash_{\mathcal{I}} \perp$

$S \vdash_{\mathcal{I}} T$ (any S is represented by Top)

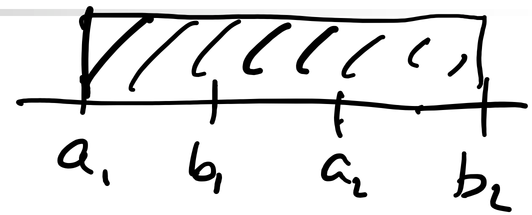
$S \vdash_{\mathcal{I}} [a,b]$ when $u \in S \quad a \le u \le b$.

Abstraction Example 3: intervals

- Concrete elements: elements of $S \in \mathbf{2^Z}$
- Abstract elements: $\perp$, T, intervals [**a**,**b**] where **a** $\in$ **Z**U**{−∞}** and **b** $\in$ **Z**U**{∞}** and **a ≤ b**
  - Is it a lattice?
  - Yes!
- Abstraction relation:
  - $\emptyset \vdash_I \perp$
  - $\mathbf{S} \vdash_I \mathbf{T}$
  - $\mathbf{S} \vdash_I [\mathbf{a},\mathbf{b}]$ iff for every $\mathbf{n} \in \mathbf{S}$, $\mathbf{a \leq n \leq b}$

Abstraction Example 3: intervals

- Abstract semantics

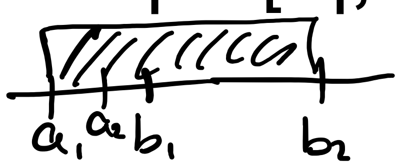


  - If $\mathbf{x_1}$ is $[\mathbf{a_1},\mathbf{b_1}]$ and $\mathbf{x_2}$ is $[\mathbf{a_2},\mathbf{b_2}]$ then $\mathbf{x_1} + \mathbf{x_2}$ is?

    $[a_1+a_2,\ b_1+b_2]$

  - If $\mathbf{x_1}$ is $[\mathbf{a_1},\mathbf{b_1}]$ and $\mathbf{x_2}$ is $[\mathbf{a_2},\mathbf{b_2}]$ then $\mathbf{x_1} \cup \mathbf{x_2}$ is?

    $[\min(a_1,a_2),\ \max(b_1,b_2)]$



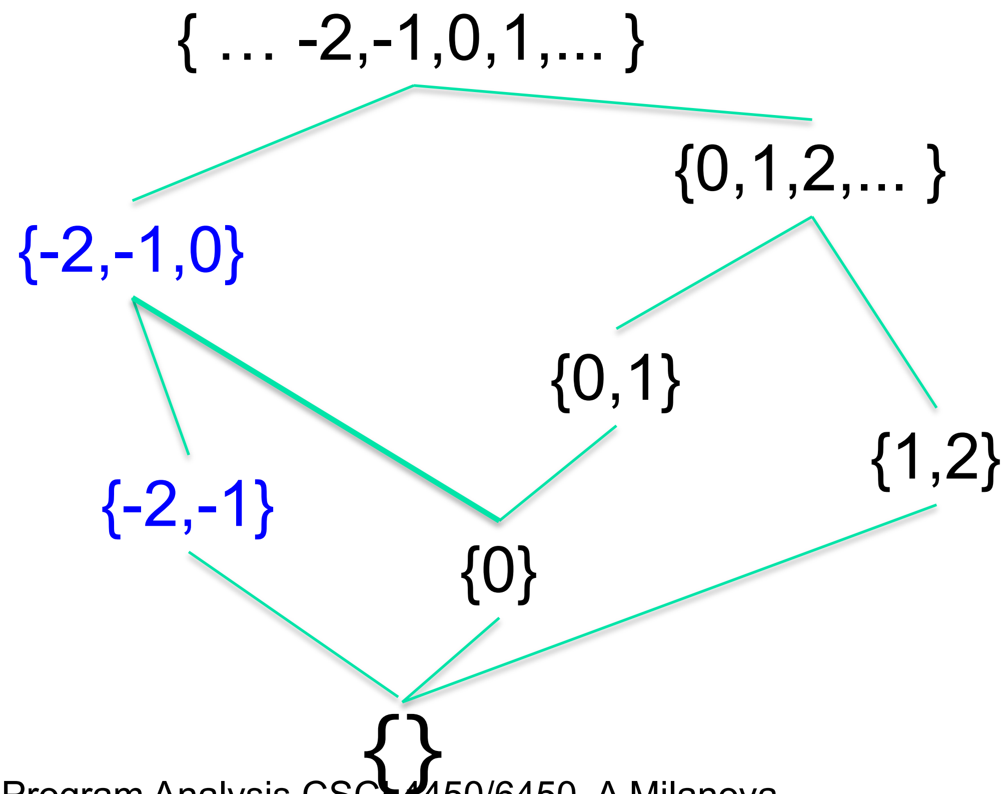
  - If $\mathbf{x_1}$ is $[\mathbf{a_1},\mathbf{b_1}]$ and $\mathbf{x_2}$ is $[\mathbf{a_2},\mathbf{b_2}]$ then $\mathbf{x_1} \cap \mathbf{x_2}$ is?

    $\perp$ if $b_1 \le a_2$ or $b_2 < a_1$

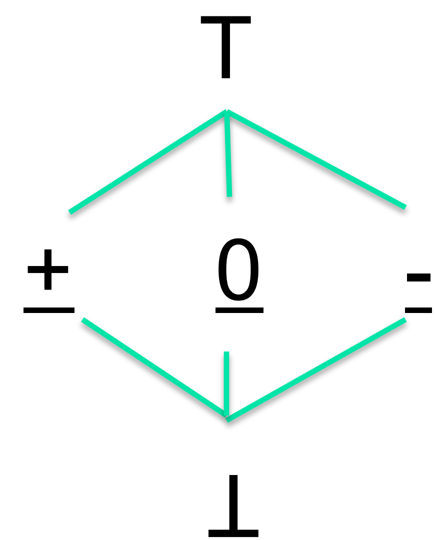
    $[\max(a_1,a_2),\ \min(b_1,b_2)]$ otherwise

Abstraction Relation

Concrete lattice:

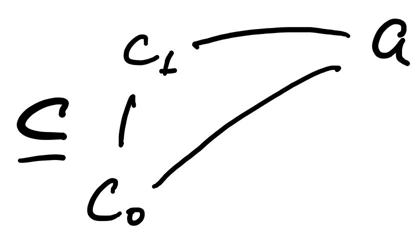


Abstract lattice:

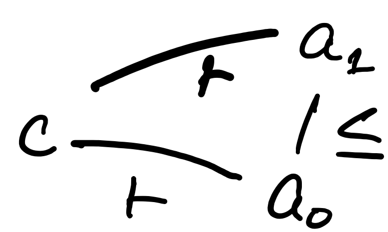


Towards Concretization and Abstraction Functions

- Abstraction relation is consistent with order!
- Concrete order:
  - If $\mathbf{c_0} \subseteq \mathbf{c_1}$ and $\mathbf{c_1}$ is represented by **a**, then $\mathbf{c_0}$ is represented by **a**

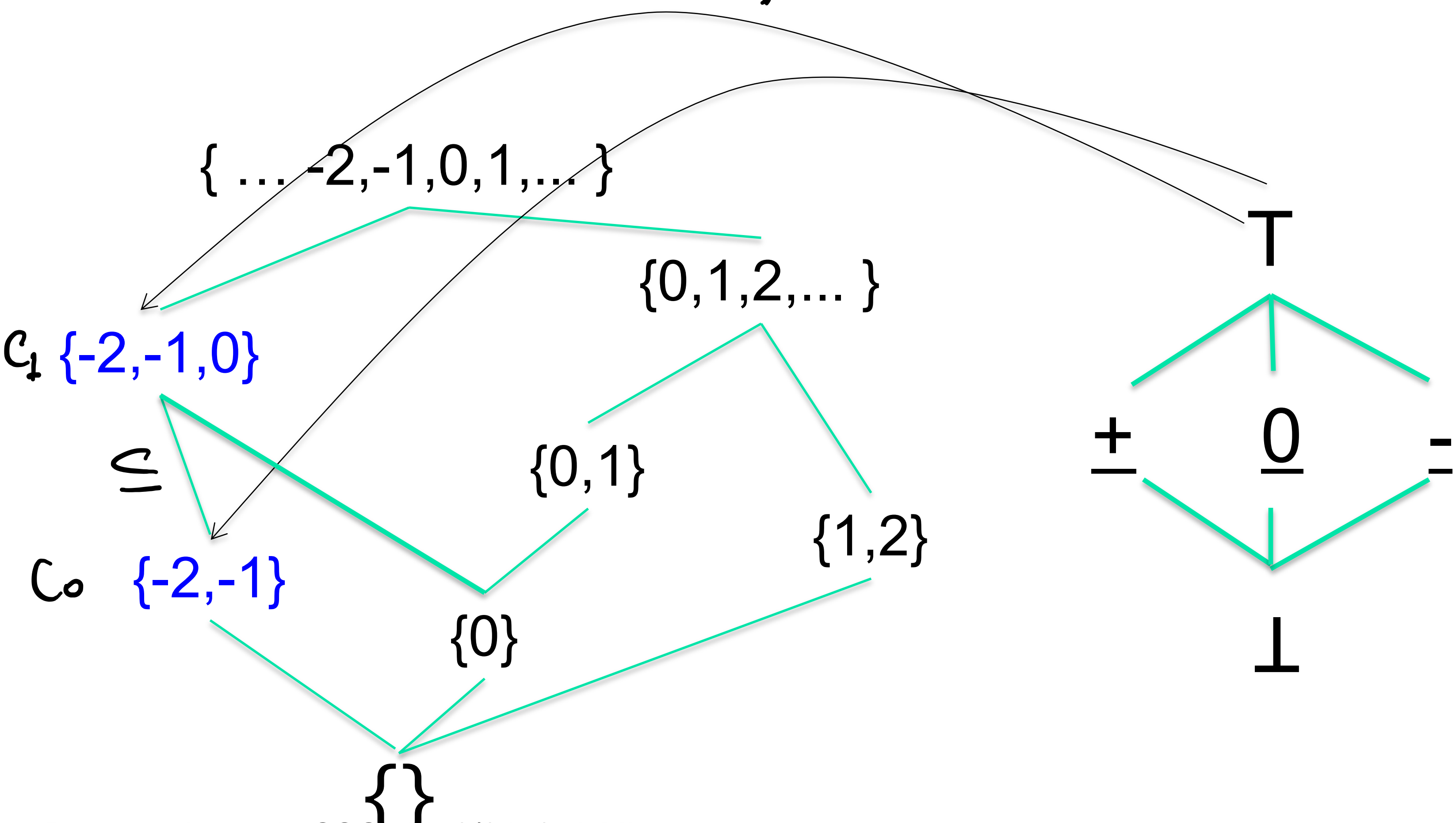


- Abstract order:
  - If $\mathbf{a_0} \leq \mathbf{a_1}$ and **c** is represented by $\mathbf{a_0}$, then **c** is represented by $\mathbf{a_1}$



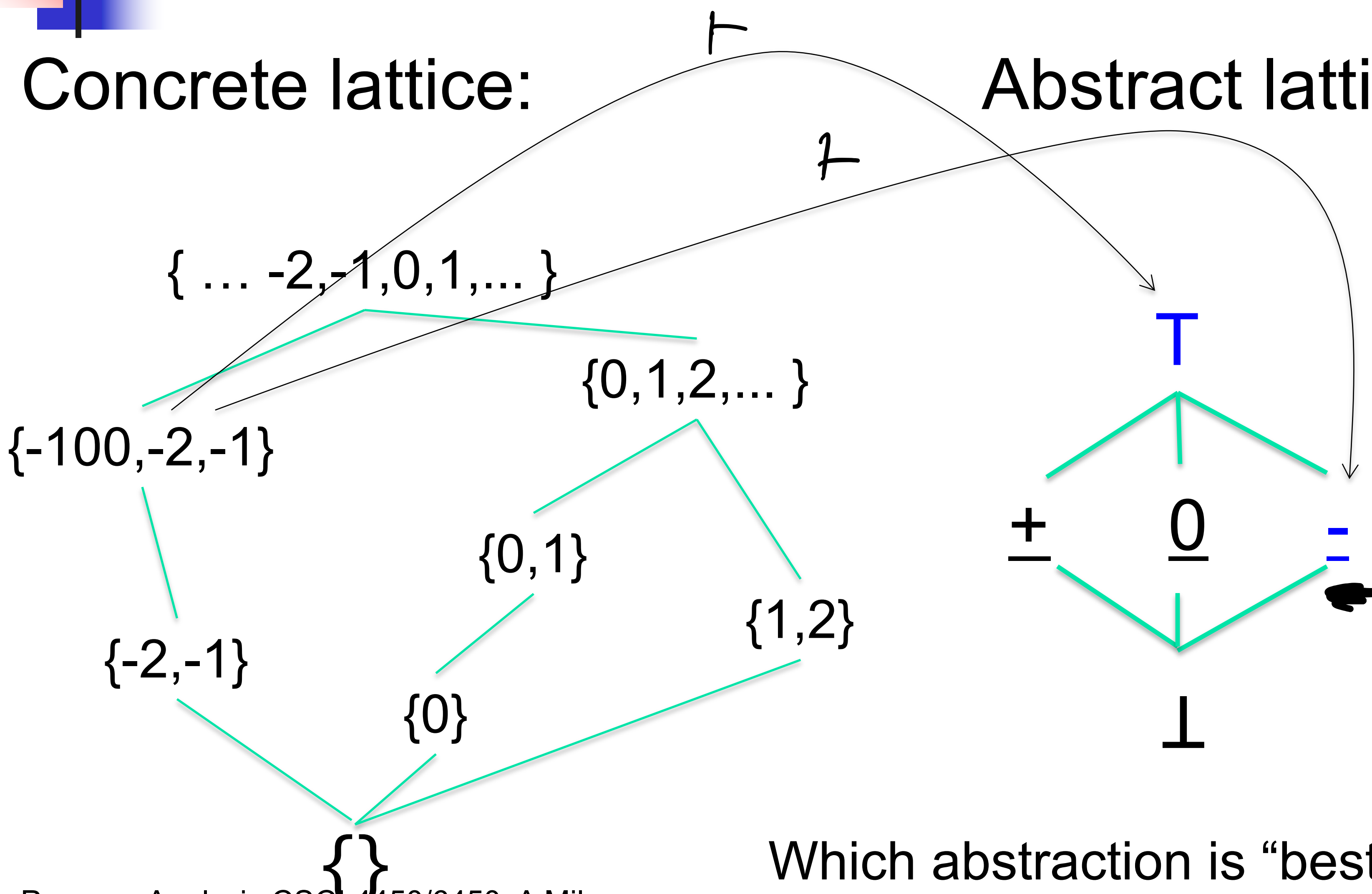
Abstraction Relation is Consistent with Partial Orders!

Concrete lattice: α Abstract lattice:



Abstraction Relation is Consistent with Partial Orders!

Concrete lattice: Abstract lattice:



Which abstraction is "best"?

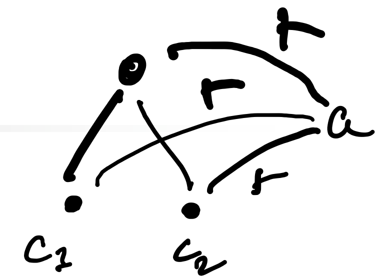
Towards Concretization and Abstraction Functions

- Previous slides, more formally
- Concrete lattice **C**, $\subseteq$ and abstract lattice **A**, $\leq$
- Abstraction relation is consistent with ordering:
  - For every $\mathbf{c_0}, \mathbf{c_1} \in \mathbf{C}$ and every $\mathbf{a} \in \mathbf{A}$,

    $\mathbf{c_0} \subseteq \mathbf{c_1}$ and $\mathbf{c_1} \vdash \mathbf{a} \Rightarrow \mathbf{c_0} \vdash \mathbf{a}$
  - For every $\mathbf{a_0}, \mathbf{a_1} \in \mathbf{A}$ and every $\mathbf{c} \in \mathbf{C}$,

    $\mathbf{a_0} \leq \mathbf{a_1}$ and $\mathbf{c} \vdash \mathbf{a_0} \Rightarrow \mathbf{c} \vdash \mathbf{a_1}$
- The abstraction relation makes sense but easier to have functions
  - Concretization function: A → C
  - Abstraction function: C → A

Concretization Function



- Definition:

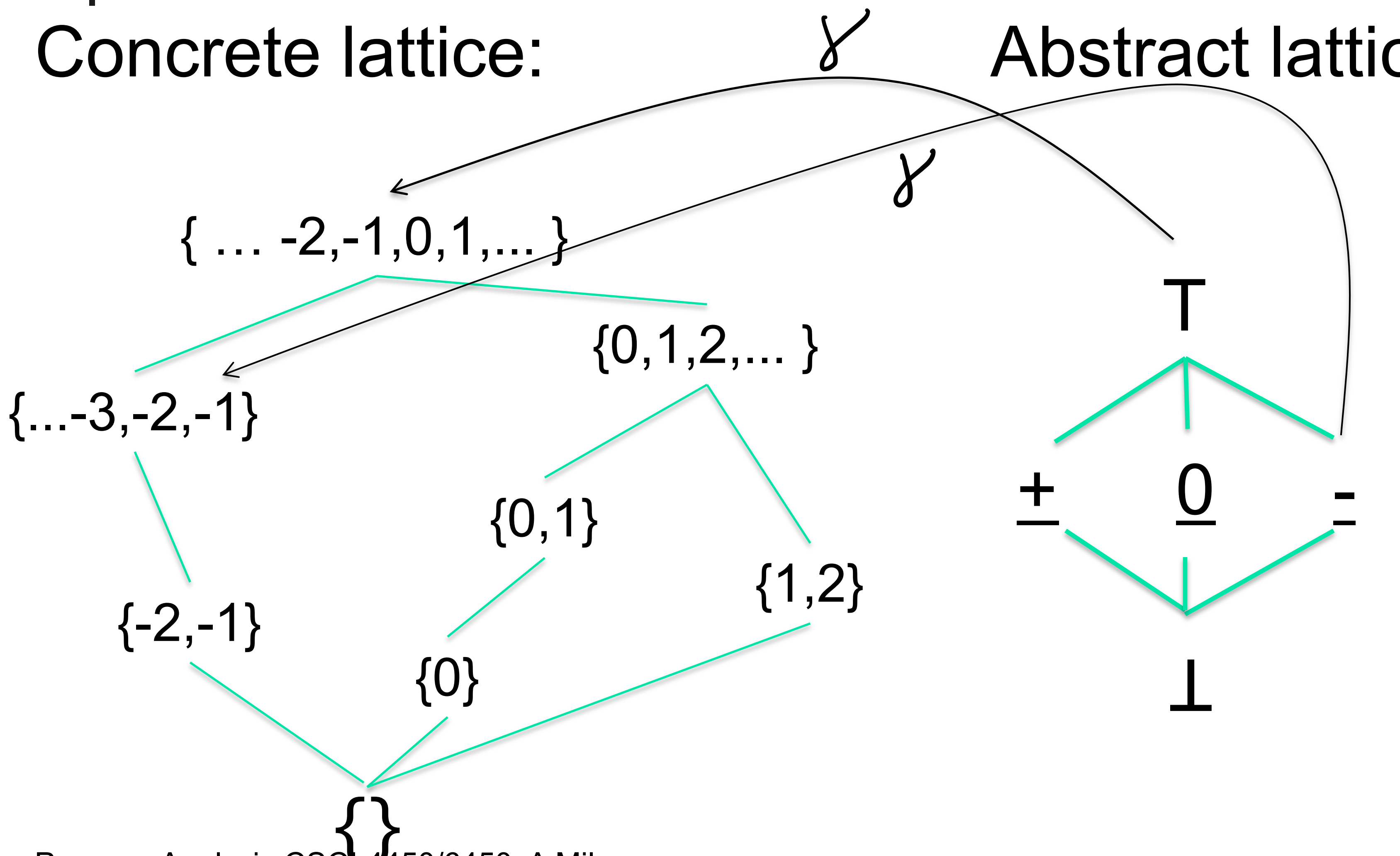
Concretization function γ : **A→C** (if it exists) maps **a**$\in$**A** to the largest (most general) element **c**$\in$**C** such that **c** $\vdash$ **a**

Note: γ(**a**) "covers" all concrete elements that are represented by **a**

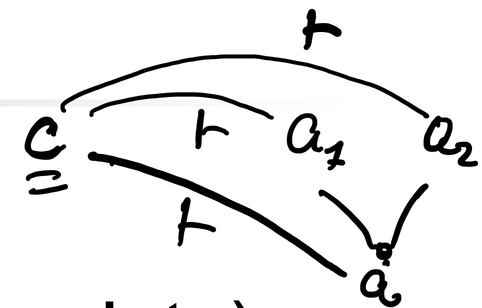
- γ(**a**) returns the most general element **c** such that **c** is represented by **a**. This is called concretization

Gamma Examples

Concrete lattice: Abstract lattice:



Abstraction Function



- Definition:

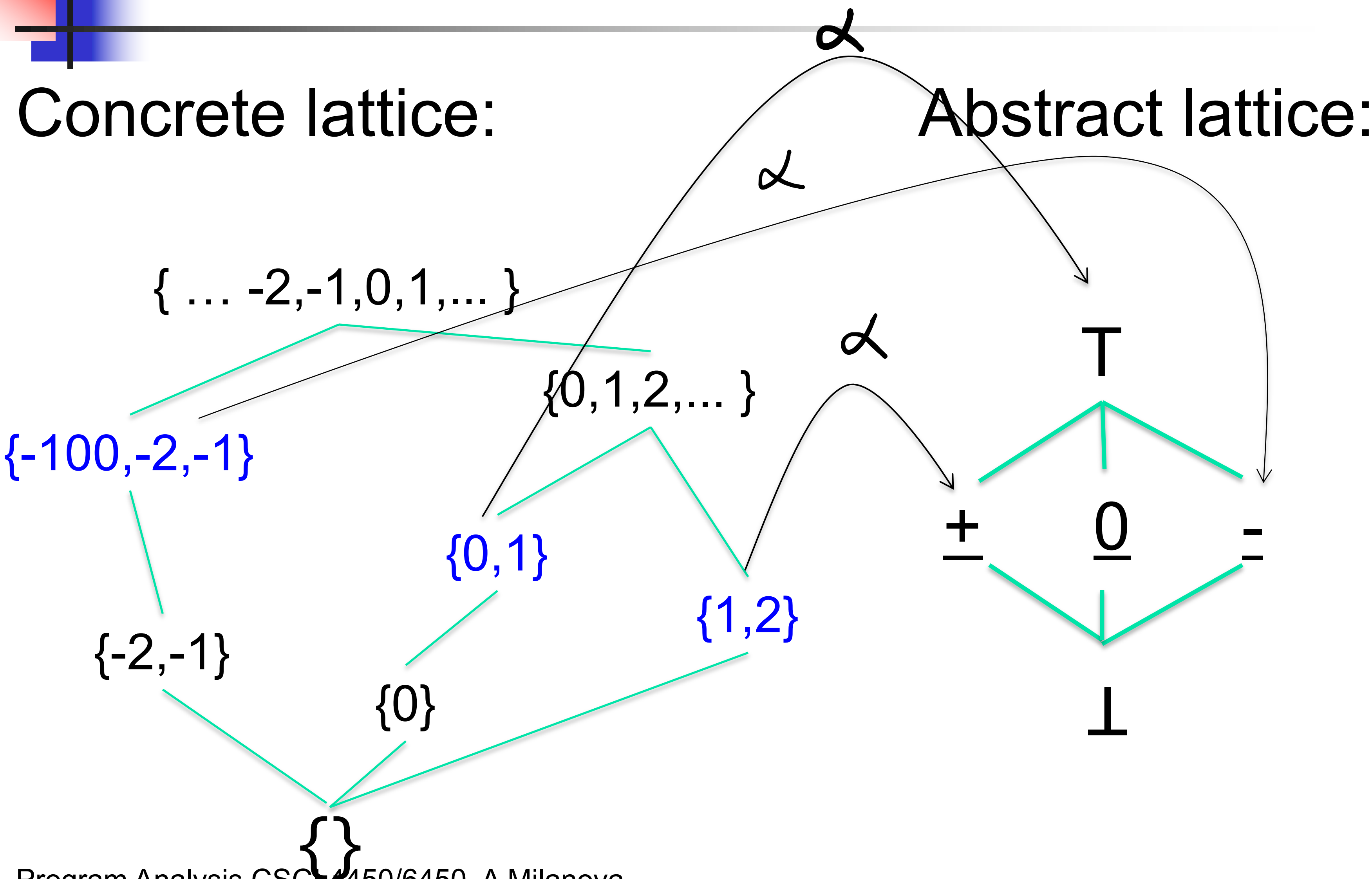
Abstraction function α : **C→A** (if it exists) maps **c**$\in$**C** to the smallest (most precise) element **a**$\in$**A** such that **c** $\vdash$ **a**

$c \vdash a' \Rightarrow$

$\alpha(c) \leq a'$

- α maps **c** to the most precise **a** such that **a** represents **c.** This is called best abstraction

Alpha Examples



Concretization Function Examples

- Concretization of lattice of signs

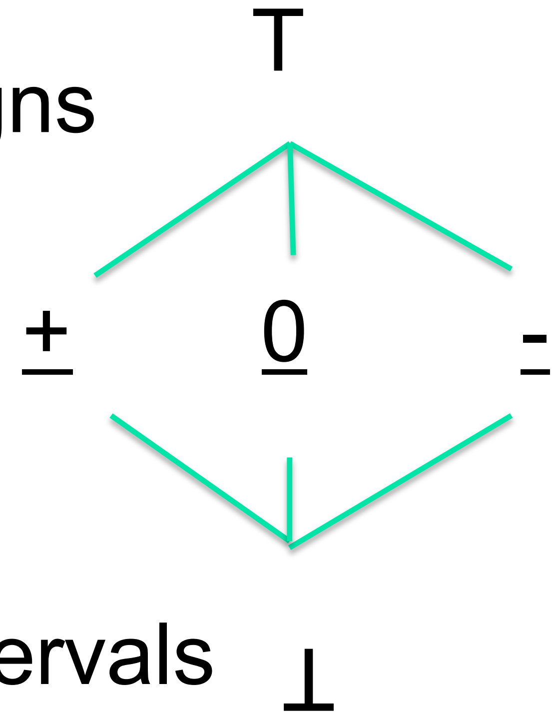
$\gamma(\top) = \mathbb{Z}$

$\gamma(+) = \{1, 2, 3, \ldots\}$

$\gamma(-) = \{\ldots -3, -2, -1\}$

$\gamma(0) = \{0\}$

$\gamma(\bot) = \{\}$



- Concretization of lattice of intervals

$\gamma_I(\top) = \mathbb{Z}$

$\gamma_I(\bot) = \{\}$

$\gamma_I([a,b]) = \{a, a+1, \ldots b-1, b\}$

Abstraction Function Examples

- Signs abstraction $\alpha_S(\mathbf{c})$

$\alpha(\{\}) = \bot$

$\alpha(\{0\}) = \underline{0}$

$\alpha(c) = \underline{+}$ if c contains only positive ints

$\alpha(c) = \underline{-}$ if c contains only negative ints

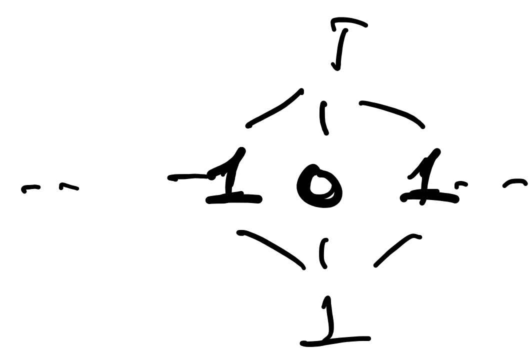
$\alpha(c) = \top$ otherwise

- Constants abstraction $\alpha_C(\mathbf{c})$

$\alpha(\{\}) = \bot$

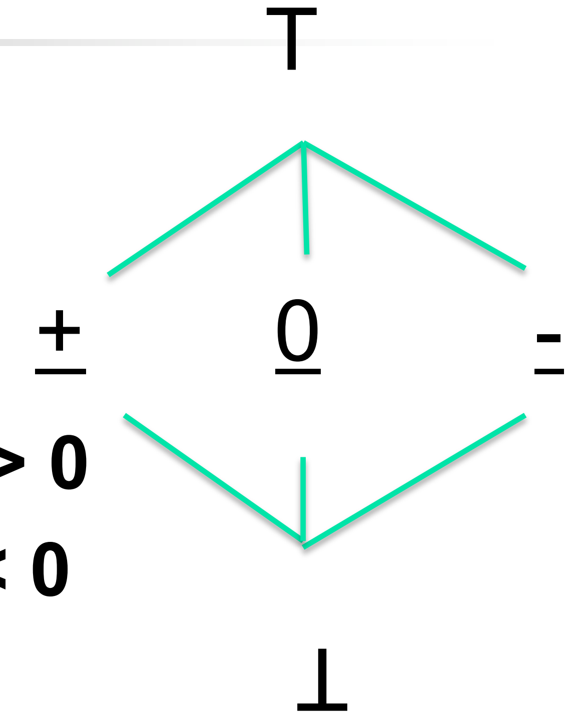
$\alpha(\{n\}) = \underline{n}$

$\alpha(c) = \top$ otherwise



Abstraction Function Examples

- Signs abstraction
  - α_S**(c)** → ⊥ if **c = {}**
  - α_S**(c)** → $\underline{0}$ if **c = {0}**
  - α_S**(c)** → $\underline{+}$ if for every **n** ∈ **c, n > 0**
  - α_S**(c)** → $\underline{-}$ if for every **n** ∈ **c, n < 0**
  - α_S**(c)** → **T** otherwise



- Constants abstraction
  - α_C**(c)** → ⊥ if **c = {}**
  - α_C**(c)** → $\underline{n}$ if **c = {n}**
  - α_C**(c)** → **T** otherwise

Outline

- Overview of Abstract interpretation

- Semantics
- Notion of abstraction
- Concretization and abstraction functions
- Galois Connections

- Applications of abstract interpretation

Galois Connection

- A Galois Connection links α and γ. It captures that they represent the abstraction relation $\vdash$!
- Definition

A Galois connection is defined by concrete lattice ($\mathbf{C},\subseteq$), abstract lattice ($\mathbf{A},\leq$), an abstraction function α : $\mathbf{C}\rightarrow\mathbf{A}$ and concretization function γ : $\mathbf{A}\rightarrow\mathbf{C}$ such that

for every $\mathbf{a} \in \mathbf{A}$ and every $\mathbf{c} \in \mathbf{C}$

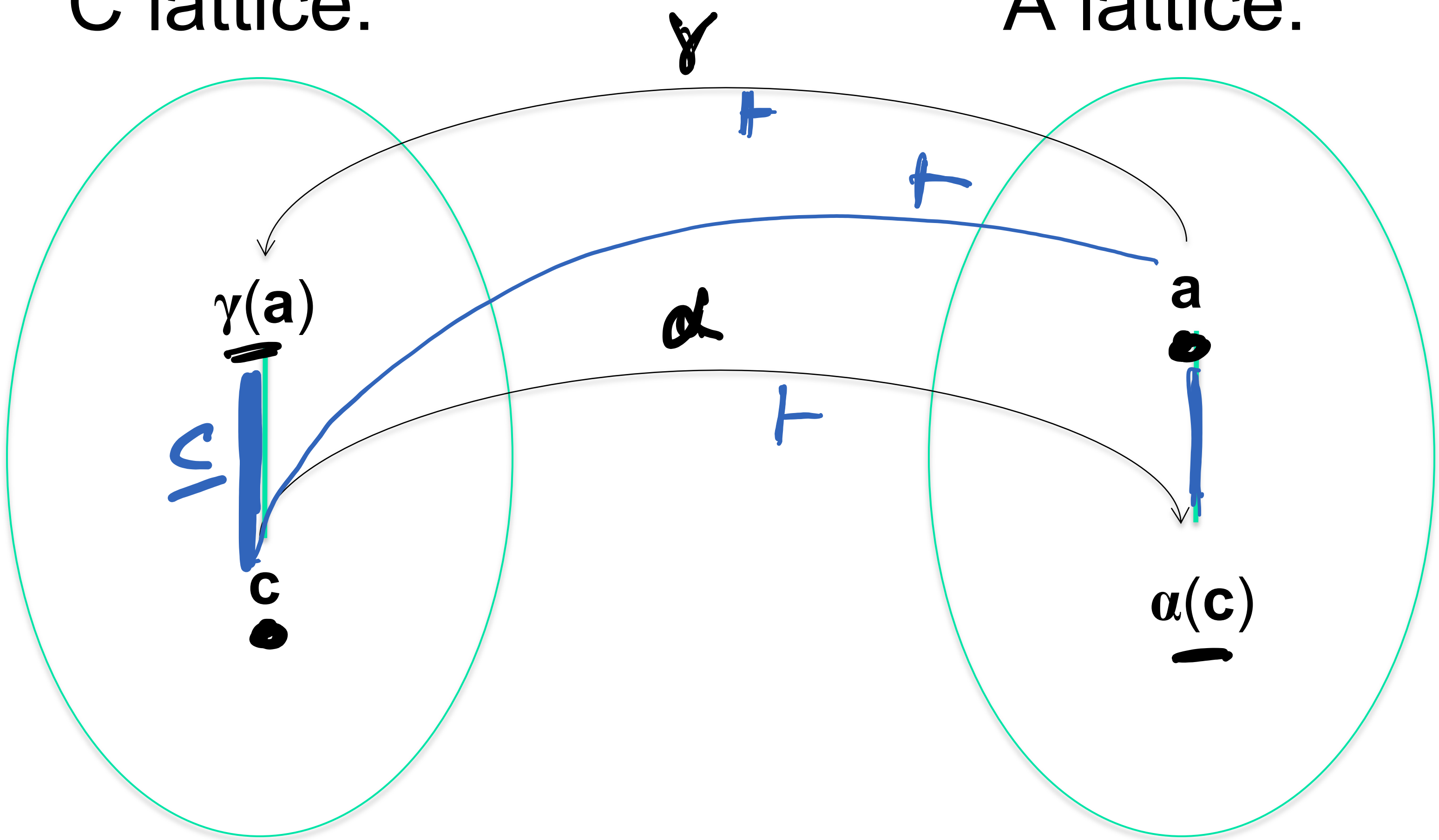
$\mathbf{c} \subseteq \gamma(\mathbf{a})$ if and only if $\alpha(\mathbf{c}) \leq \mathbf{a}$

Galois Connection

$c \subseteq \gamma(a) \Rightarrow \alpha(c) \leq a$

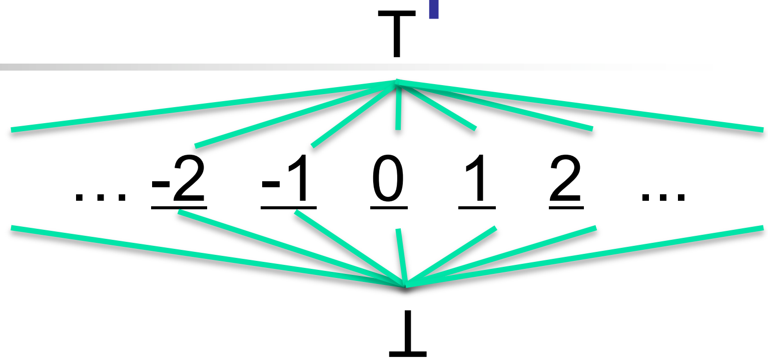
$\alpha(c) \leq a \Rightarrow c \subseteq \gamma(a)$

C lattice: A lattice:



Galois Connection Example

- Constants lattice



$\alpha_C(\mathbf{c}) \rightarrow \perp$ if $\mathbf{c} = \{\}$

$\alpha_C(\mathbf{c}) \rightarrow \underline{\mathbf{n}}$ if $\mathbf{c} = \{\mathbf{n}\}$

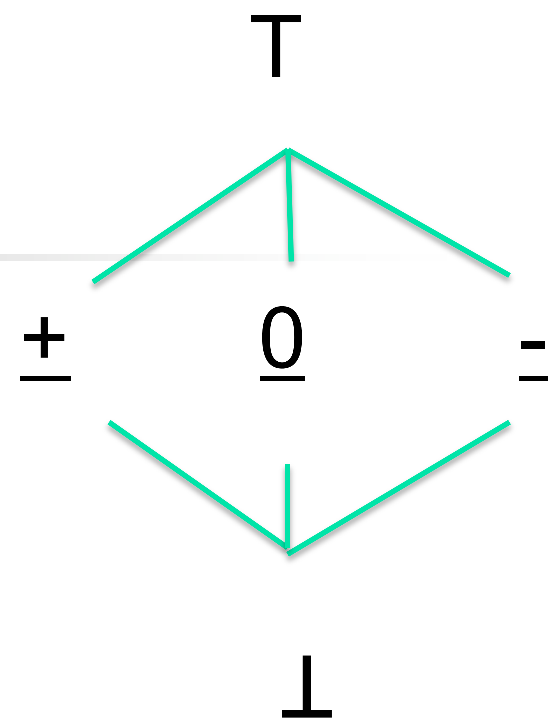
$\alpha_C(\mathbf{c}) \rightarrow$ T otherwise

$\gamma_C(\mathrm{T}) \rightarrow \mathbf{Z}$

$\gamma_C(\underline{\mathbf{n}}) \rightarrow \{\mathbf{n}\}$

$\gamma_C(\perp) \rightarrow \{\}$

Galois Connection Example



- Signs lattice

  $\alpha_C(\mathbf{c}) \rightarrow \underline{+}$ if for every **n** in **c, n>0**

  $\alpha_C(\mathbf{c}) \rightarrow \underline{-}$ if for every **n** in **c, n<0**

  $\alpha_C(\mathbf{c}) \rightarrow \underline{\mathbf{0}}$ if **c = {0}**

  $\alpha_C(\mathbf{c}) \rightarrow \perp$ if **c = {}**

  $\alpha_C(\mathbf{c}) \rightarrow$ T otherwise

$\gamma_C(\mathrm{T}) \rightarrow \mathbf{Z}$

$\gamma_C(\underline{+}) \rightarrow \{1,2,\ldots\}$

$\gamma_C(\underline{0}) \rightarrow \{0\}$

$\gamma_C(\underline{-}) \rightarrow \{\ldots,-2,-1\}$

$\gamma_C(\perp) \rightarrow \{\}$