

Lecture 19: Abstract Interpretation Idea. Lattices

Viktor Kunčák

Abstract Interpretation

A Method for Constructing Inductive Invariants

Basic idea of abstract interpretation

Abstract interpretation is a way to infer properties of program computations.
Consider the assignment: $z = x + y$.

Interpreter:

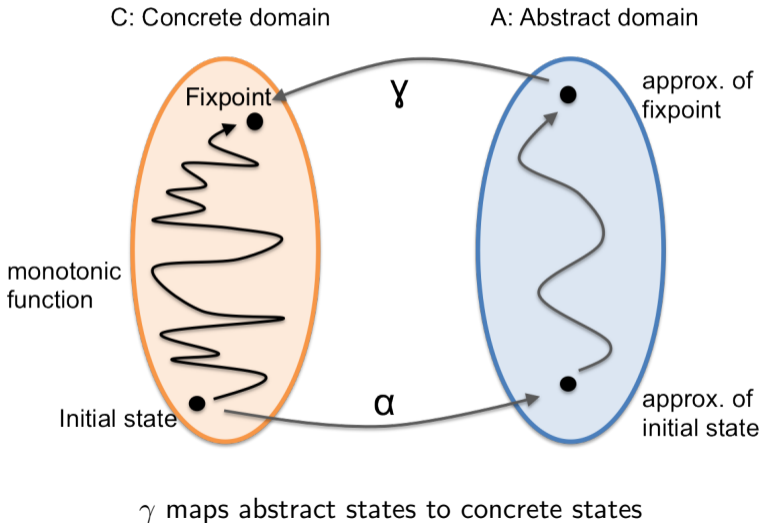
$$\begin{pmatrix} x : 10 \\ y : -2 \\ z : 3 \end{pmatrix} \xrightarrow{z=x+y} \begin{pmatrix} x : 10 \\ y : -2 \\ z : 8 \end{pmatrix}$$

Abstract interpreter:

$$\begin{pmatrix} x \in [0, 10] \\ y \in [-5, 5] \\ z \in [0, 10] \end{pmatrix} \xrightarrow{z=x+y} \begin{pmatrix} x \in [0, 10] \\ y \in [-5, 5] \\ z \in [-5, 15] \end{pmatrix}$$

Each abstract state represents a set of concrete states

Program Meaning is a Fixpoint. We Approximate It.



Proving through Fixpoints of Approximate Functions

Meaning of a program (e.g. a relation) is a least fixpoint of F .

Given specification s , the goal is to prove $\mathbf{lfp}(F) \subseteq s$

- ▶ if $F(s) \subseteq s$ then $\mathit{lfp}(F) \subseteq s$ and we are done
- ▶ $\mathit{lfp}(F) = \bigcup_{k \geq 0} F^k(\emptyset)$, but that is too hard to compute because it is infinite union unless, by some luck, $F^{n+1}(\emptyset) = F^n(\emptyset)$ for some n

Instead, we search for an inductive strengthening of s : find s' such that:

- ▶ $F(s') \subseteq s'$ (s' is inductive). If so, theorem says $\mathit{lfp}(F) \subseteq s'$
- ▶ $s' \subseteq s$ (s' implies the desired specification). Then $\mathit{lfp}(F) \subseteq s' \subseteq s$

How to find s' ? Iterating F is hard, so we try some simpler function $F_{\#}$:

- ▶ suppose $F_{\#}$ is *approximation*: $F(r) \subseteq F_{\#}(r)$ for all r
- ▶ we can find s' such that: $F_{\#}(s') \subseteq s'$ (e.g. $s' = F_{\#}^{n+1}(\emptyset) = F_{\#}^n(\emptyset)$)

Then: $F(s') \subseteq F_{\#}(s') \subseteq s'$. So, if $s' \subseteq s$, we have know $\mathit{lfp}(F) \subseteq s'$.

Abstract interpretation: automatically construct $F_{\#}$ using F and s

Programs as control-flow graphs

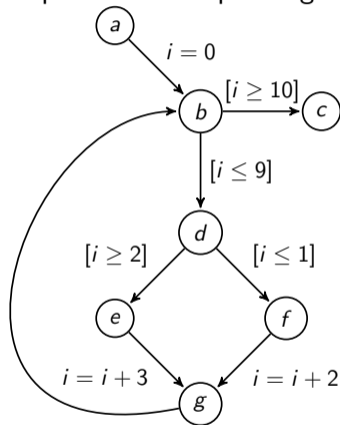
One possible corresponding control-flow graph is:

```
//a
i = 0;
    //b
while (i < 10) {
    //d
    if (i > 1)
        //e
        i = i + 3;
    else
        //f
        i = i + 2;
    //g
}
//c
```

Programs as control-flow graphs

```
//a  
i = 0;  
    //b  
while (i < 10) {  
    //d  
    if (i > 1)  
        //e  
        i = i + 3;  
    else  
        //f  
        i = i + 2;  
    //g  
}  
//c
```

One possible corresponding control-flow graph is:



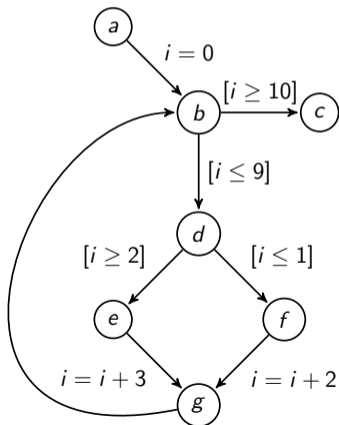
Sets of states at each program point

Suppose that

- ▶ program state is given by the value of the integer variable i
- ▶ initially, it is possible that i has any value

Compute the set of states at each vertex in the CFG.

```
//a
i = 0;
    //b
while (i < 10) {
    //d
    if (i > 1)
        //e
        i = i + 3;
    else
        //f
        i = i + 2;
    //g
}
//c
```



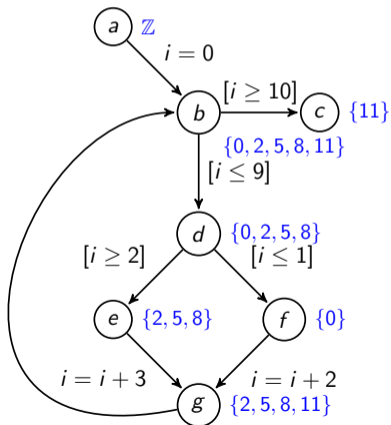
Sets of states at each program point

Suppose that

- ▶ program state is given by the value of the integer variable i
- ▶ initially, it is possible that i has any value

Compute the set of states at each vertex in the CFG.

```
//a  
i = 0;  
    //b  
while (i < 10) {  
    //d  
    if (i > 1)  
        //e  
        i = i + 3;  
    else  
        //f  
        i = i + 2;  
    //g  
}  
//c
```



Sets of states at each program point

Running the Program

One way to describe the set of states for each program point: for each initial state, run the CFG with this state and insert the modified states at appropriate points.

Reachable States as A Set of Recursive Equations

If c is the label on the edge of the graph, let $\rho(c)$ denotes the relation between initial and final state that describes the meaning of statement. For example,

$$\rho(i = 0) = \{(i, i') \mid i' = 0\}$$

$$\rho(i = i + 2) = \{(i, i') \mid i' = i + 2\}$$

$$\rho(i = i + 3) = \{(i, i') \mid i' = i + 3\}$$

$$\rho([i < 10]) = \{(i, i') \mid i' = i \wedge i < 10\}$$

Sets of states at each program point

We will write $T(S, c)$ (transfer function) for the image of set S under relation $\rho(c)$. For example,

$$T(\{10, 15, 20\}, i = i + 2) = \{12, 17, 22\}$$

General definition can be given using the notion of strongest postcondition

$$T(S, c) = sp(S, \rho(c))$$

If $[p]$ is a condition (assume(p), coming from 'if' or 'while') then

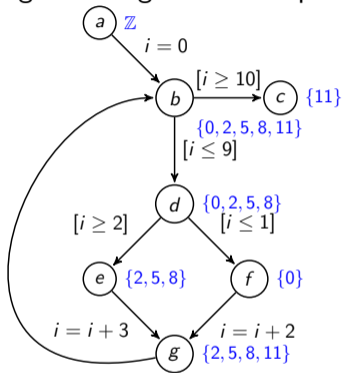
$$T(S, [p]) = \{x \in S \mid p\}$$

If an edge has no label, we denote it skip. So, $T(S, skip) = S$.

Reachable States as A Set of Recursive Equations

Now we can describe the meaning of our program using recursive equations:

$$\begin{aligned} S(a) &= \{\dots, -2, -1, 0, 1, 2, \dots\} \\ S(b) &= T(S(a), i = 0) \cup T(S(g), \text{skip}) \\ S(c) &= T(S(b), [\neg(i < 10)]) \\ S(d) &= T(S(b), [i < 10]) \\ S(e) &= T(S(d), [i > 1]) \\ S(f) &= T(S(d), [\neg(i > 1)]) \\ S(g) &= T(S(e), i = i + 3) \\ &\quad \cup T(S(f), i = i + 2) \end{aligned}$$



Our solution is the unique **least** solution of these equations. Can be computed by iterating starting from empty sets as initial solution.

The problem: These exact equations are as difficult to compute as running the program on all possible input states. Instead, we consider **approximate** descriptions of these sets of states.

A Large Analysis Domain: All Intervals of Integers

For every $L, U \in \mathbb{Z}$ interval:

$$\{x \mid L \leq x \wedge x \leq U\}$$

This domain has infinitely many elements, but is already an approximation of all possible sets of integers.

Smaller Domain: Finitely Many Intervals

We continue with the same example but instead of allowing to denote all possible sets, we will allow sets represented by expressions

$$[L, U]$$

which denote the set $\{x \mid L \leq x \wedge x \leq U\}$.

Example: $[0, 127]$ denotes integers between 0 and 127.

- ▶ L is the lower bound and U is the upper bound, with $L \leq U$.
- ▶ to ensure that we have only a few elements, we let

$$L, U \in \{\text{MININT}, -128, 1, 0, 1, 127, \text{MAXINT}\}$$

- ▶ $[\text{MININT}, \text{MAXINT}]$ denotes all possible integers, denote it \top
- ▶ instead of writing $[1, 0]$ and other empty sets, we will always write \perp

So, we only work with a finite number of sets $1 + \binom{7}{2} = 22$.

Denote the family of these sets by D (domain).

New Set of Recursive Equations

We want to write the same set of equations as before, but because we have only a finite number of sets, we must approximate. We approximate sets with possibly larger sets.

$$\begin{aligned}S^\#(a) &= \top \\S^\#(b) &= T^\#(S^\#(a), i = 0) \\&\sqcup T^\#(S^\#(g), skip) \\S^\#(c) &= T^\#(S^\#(b), [\neg(i < 10)]) \\S^\#(d) &= T^\#(S^\#(b), [i < 10]) \\S^\#(e) &= T^\#(S^\#(d), [i > 1]) \\S^\#(f) &= T^\#(S^\#(d), [\neg(i > 1)]) \\S^\#(g) &= T^\#(S^\#(e), i = i + 3) \\&\sqcup T^\#(S^\#(f), i = i + 2)\end{aligned}$$

- ▶ $S_1 \sqcup S_2$ denotes the approximation of $S_1 \cup S_2$: it is the set that contains both S_1 and S_2 , that belongs to D , and is otherwise as small as possible. Here $[a, b] \sqcup [c, d] = [\min(a, c), \max(b, d)]$
- ▶ We use approximate functions $T^\#(S, c)$ that give a result in D .

Updating Sets

We solve the equations by starting in the initial state and repeatedly applying them.

- ▶ in the 'entry' point, we put \top , in all others we put \perp .

$$S^\#(a) = \top$$

$$S^\#(b) = T^\#(S^\#(a), i = 0)$$

$$\sqcup T^\#(S^\#(g), \text{skip})$$

$$S^\#(c) = T^\#(S^\#(b), [\neg(i < 10)])$$

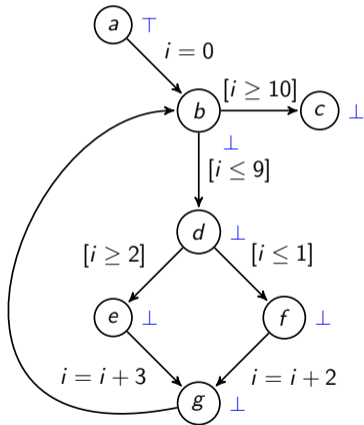
$$S^\#(d) = T^\#(S^\#(b), [i < 10])$$

$$S^\#(e) = T^\#(S^\#(d), [i > 1])$$

$$S^\#(f) = T^\#(S^\#(d), [\neg(i > 1)])$$

$$S^\#(g) = T^\#(S^\#(e), i = i + 3)$$

$$\sqcup T^\#(S^\#(f), i = i + 2)$$



Updating Sets

Sets after a few iterations:

$$S^\#(a) = \top$$

$$S^\#(b) = T^\#(S^\#(a), i = 0)$$

$$\sqcup T^\#(S^\#(g), skip)$$

$$S^\#(c) = T^\#(S^\#(b), [\neg(i < 10)])$$

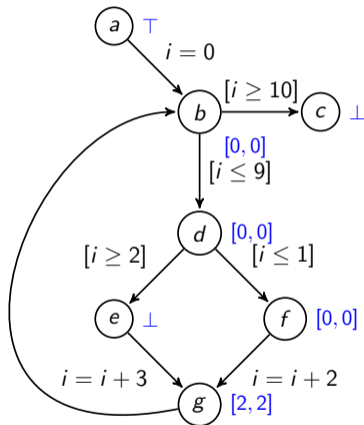
$$S^\#(d) = T^\#(S^\#(b), [i < 10])$$

$$S^\#(e) = T^\#(S^\#(d), [i > 1])$$

$$S^\#(f) = T^\#(S^\#(d), [\neg(i > 1)])$$

$$S^\#(g) = T^\#(S^\#(e), i = i + 3)$$

$$\sqcup T^\#(S^\#(f), i = i + 2)$$



Updating Sets

Sets after a few more iterations:

$$S^\#(a) = \top$$

$$S^\#(b) = T^\#(S^\#(a), i = 0)$$

$$\sqcup T^\#(S^\#(g), skip)$$

$$S^\#(c) = T^\#(S^\#(b), [\neg(i < 10)])$$

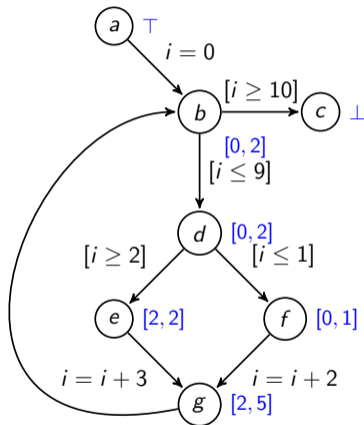
$$S^\#(d) = T^\#(S^\#(b), [i < 10])$$

$$S^\#(e) = T^\#(S^\#(d), [i > 1])$$

$$S^\#(f) = T^\#(S^\#(d), [\neg(i > 1)])$$

$$S^\#(g) = T^\#(S^\#(e), i = i + 3)$$

$$\sqcup T^\#(S^\#(f), i = i + 2)$$



Fixpoint Found

Final values of sets:

$$S^\#(a) = \top$$

$$S^\#(b) = T^\#(S^\#(a), i = 0)$$

$$\sqcup T^\#(S^\#(g), \text{skip})$$

$$S^\#(c) = T^\#(S^\#(b), [\neg(i < 10)])$$

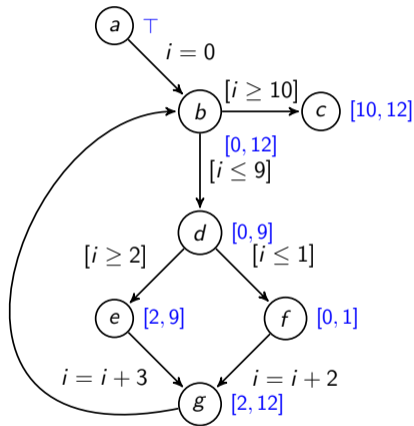
$$S^\#(d) = T^\#(S^\#(b), [i < 10])$$

$$S^\#(e) = T^\#(S^\#(d), [i > 1])$$

$$S^\#(f) = T^\#(S^\#(d), [\neg(i > 1)])$$

$$S^\#(g) = T^\#(S^\#(e), i = i + 3)$$

$$\sqcup T^\#(S^\#(f), i = i + 2)$$

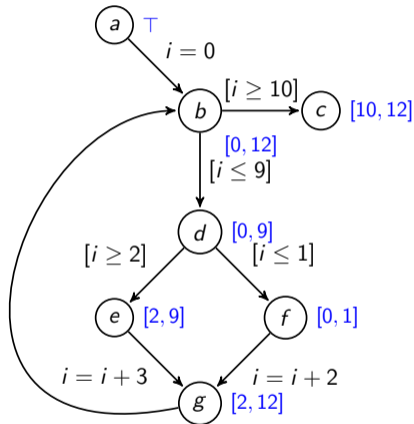


If we map intervals to sets, this is also solution of the original constraints.

Automatically Constructed Hoare Logic Proof

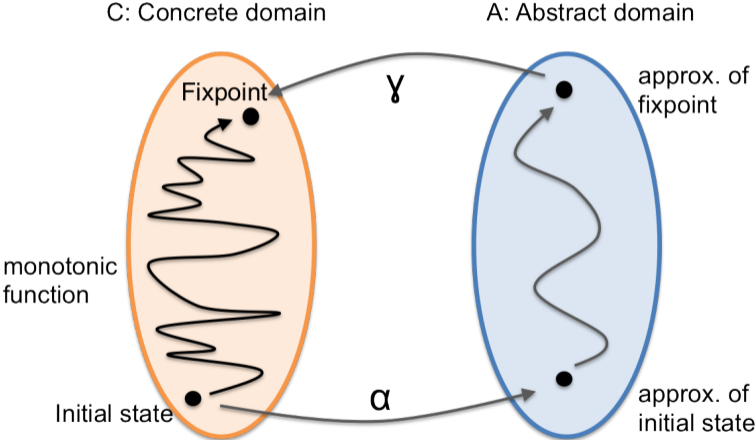
Final values of sets:

```
//a: true
i = 0;
    //b:  $0 \leq i \leq 12$ 
while (i < 10) {
    //d:  $0 \leq i \leq 9$ 
    if (i > 1)
        //e:  $2 \leq i \leq 9$ 
        i = i + 3;
    else
        //f:  $0 \leq i \leq 1$ 
        i = i + 2;
    //g:  $2 \leq i \leq 12$ 
}
//c:  $10 \leq i \leq 12$ 
```



This method constructed a sufficiently annotated program and ensured that all Hoare triples that were constructed hold

Abstract Interpretation Big Picture



Abstract Domains are Partial Orders

Program semantics is given by certain sets (e.g. sets of reachable states).

- ▶ subset relation \subseteq : used to compare sets
- ▶ union of states: used to combine sets coming from different executions (e.g. if statement)

Our goal is to approximate such sets. We introduce a domain of elements $d \in D$ where each d represents a set.

- ▶ $\gamma(d)$ is a set of states. γ is called **concretization function**
- ▶ given d_1 and d_2 , it could happen that there is **no element** d representing union

$$\gamma(d_1) \cup \gamma(d_2) = \gamma(d)$$

Instead, we use a set d that approximates union, and denote it $d_1 \sqcup d_2$

This leads us to review the theory of **partial orders** and **(semi)lattices**.

Partial Orders

Partial ordering relation is a binary relation \leq that is reflexive, antisymmetric, and transitive, that is, the following properties hold for all x, y, z :

- ▶ $x \leq x$ (reflexivity)
- ▶ $x \leq y \wedge y \leq x \rightarrow x = y$ (antisymmetry)
- ▶ $x \leq y \wedge y \leq z \rightarrow x \leq z$ (transitivity)

If A is a set and \leq a binary relation on A , we call the pair (A, \leq) a **partial order**.

Given a partial ordering relation \leq , the corresponding **strict ordering relation** $x < y$ is defined by $x \leq y \wedge x \neq y$ and can be viewed as a shorthand for this conjunction.

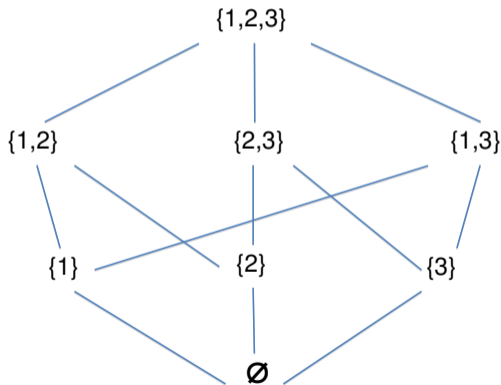
- ▶ Orders on integers, rationals, reals are all special cases of partial orders called *linear orders*.
- ▶ Given a set U , let A be any set of subsets of U , that is $A \subseteq 2^U$. Then (A, \subseteq) is a partial order.

Example: Let $U = \{1, 2, 3\}$ and let $A = \{\emptyset, \{1\}, \{2\}, \{3\}, \{2, 3\}, \{1, 2, 3\}\}$. Then (A, \subseteq) is a partial order. We can draw it as a *Hasse diagram*.

Hasse diagram

presents the relation as a directed graph in a plane, such that

- ▶ the direction of edge is given by which nodes is drawn above
- ▶ transitive and reflexive edges are not represented (they can be derived)



Extreme Elements in Partial Orders

Given a partial order (A, \leq) and a set $S \subseteq A$, we call an element $a \in A$

- ▶ **upper bound** of S if for all $a' \in S$ we have $a' \leq a$
- ▶ **lower bound** of S if for all $a' \in S$ we have $a \leq a'$
- ▶ **minimal element** of S if $a \in S$ and there is no element $a' \in S$ such that $a' < a$
- ▶ **maximal element** of S if $a \in S$ and there is no element $a' \in S$ such that $a < a'$
- ▶ **greatest element** of S if $a \in S$ and for all $a' \in S$ we have $a' \leq a$
- ▶ **least element** of S if $a \in S$ and for all $a' \in S$ we have $a \leq a'$
- ▶ **least upper bound** (lub, supremum, join, \sqcup) of S if a is the least element in the set of all upper bounds of S
- ▶ **greatest lower bound** (glb, infimum, meet, \sqcap) of S if a is the greatest element in the set of all lower bounds of S

Taking $S = A$ we obtain minimal, maximal, greatest, least elements for the entire partial order.

Extreme Elements in Partial Orders

Notes

- ▶ minimal element need not exist: $(0, 1)$ interval of rationals
- ▶ there may be multiple minimal elements: $\{\{a\}, \{b\}, \{a, b\}\}$
- ▶ if minimal element exists, it need not be least: above example
- ▶ there are no two distinct least elements for the same set
- ▶ least element is always glb and minimal
- ▶ if glb belongs to the set, then it is always least and minimal
- ▶ on a family of relations closed under \cap and \cup , *glb* is \cap and *lub* is \cup for the partial order \subseteq ; not all families of sets are closed; these are:
 - ▶ the set of all subsets
 - ▶ the family of open sets from topology

Least upper bound (lub, supremum, join, \sqcup)

Denoted $\text{lub}(S)$, least upper bound of S is an element M , if it exists, such that M is the least element of the set

$$U = \{x \mid x \text{ is upper bound on } S\}$$

In other words:

- ▶ M is an upper bound on S
- ▶ for every other upper bound M' on S , we have that $M \leq M'$

Note: this is the same definition as supremum in real analysis.

Least upper bound (glb, infimum, meet, \sqcap)

$a_1 \sqcup a_2$ denotes $\text{lub}(\{a_1, a_2\})$

$(\dots (a_1 \sqcup a_2) \dots) \sqcup a_n$ is in fact $\text{lub}(\{a_1, \dots, a_n\})$

So the operation is

- ▶ associative
- ▶ commutative
- ▶ idempotent

Real Analysis

Take as S the open interval of reals $(0, 1) = \{x \mid 0 < x < 1\}$

Then

- ▶ S has no maximal element
- ▶ S thus has no greatest element
- ▶ 2, 2.5, 3, ... are all upper bounds on S
- ▶ $\text{lub}(S) = 1$

Exercise: subsets of U

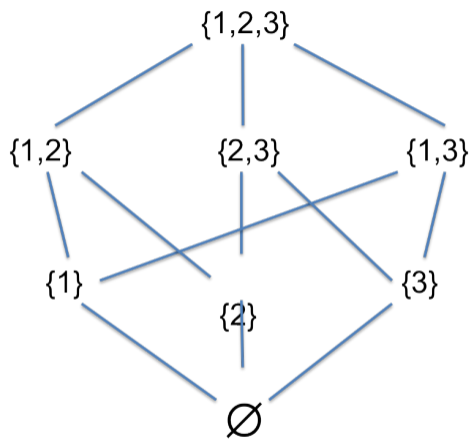
Consider

$$A = 2^U = \{S \mid S \subseteq U\} \quad \text{and} \quad (A, \subseteq)$$

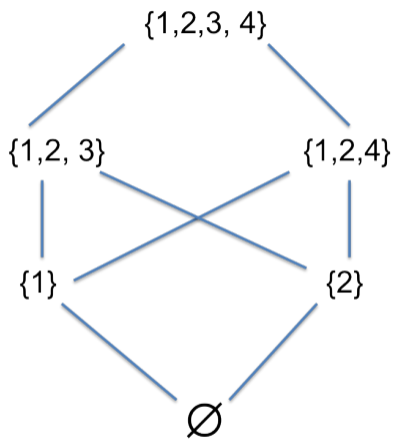
Do these exist, and if so, what are they?

- ▶ $s_1 \subseteq S, s_2 \subseteq S, \text{lub}(\{s_1, s_2\}) = ?$
- ▶ $\text{lub}(S) = ?$

Exercise: find the lub

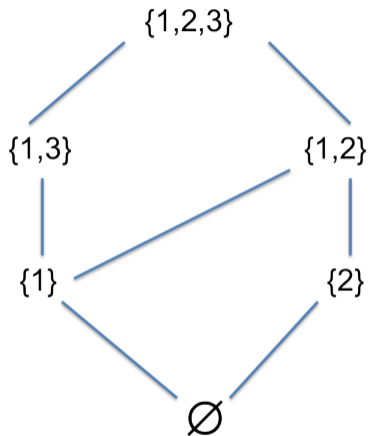


$$\{1\} \sqcup \{2\} =$$

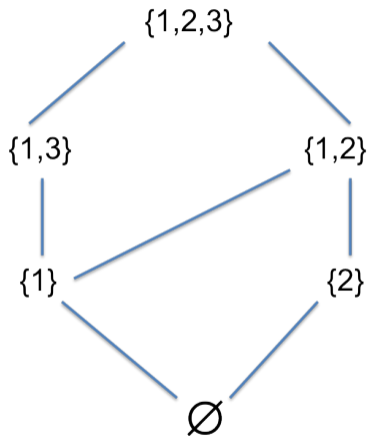


$$\{1\} \sqcup \{2\} =$$

Does every pair of elements in this order have a least upper bound?



Does every pair of elements in this order have a least upper bound?



Dually, does it have a **greatest lower bound**?

Partial order for the domain of intervals

Domain: $D = \{\perp\} \cup \{(L, U) \mid L \in \{-\infty\} \cup \mathbb{Z}, U \in \{+\infty\} \cup \mathbb{Z}\}$
such that $L \leq U$.

The associated set of elements is given by the function γ :

$$\gamma : D \rightarrow 2^{\mathbb{Z}}, \quad \gamma((L, U)) = \{x \mid L \leq x \wedge x \leq U\}$$

Lub: for $d_1, d_2 \in D$, $d_1 \sqsubseteq d_2 \iff \gamma(d_1) \subseteq \gamma(d_2)$

hence

$$(L_1, U_1) \sqsubseteq (L_2, U_2) \iff L_2 \leq L_1 \wedge U_1 \leq U_2$$

$$\perp \sqsubseteq d \quad \forall d \in D$$

$$(L_1, U_1) \sqcup (L_2, U_2) = (\min(L_1, L_2), \max(U_1, U_2))$$

Remark on constructing orders using inverse images

Suppose $\gamma : D \rightarrow C$ where C is some collection of sets.

If we define relation \sqsubseteq by:

$$d_1 \sqsubseteq d_2 \iff \gamma(d_1) \subseteq \gamma(d_2)$$

then

1. \sqsubseteq is reflexive
2. \sqsubseteq is transitive
3. \sqsubseteq is antisymmetric if and only iff γ is injective

If \sqsubseteq is not antisymmetric then we can define equivalence relation

$$d_1 \sim d_2 \iff \gamma(d_1) = \gamma(d_2)$$

and then take D' to be equivalence classes of such new set.

Example: suppose we defined intervals as all possible pairs of integers (L, U) . Then there would be many representations of the empty set, all those intervals where $L > U$.

Lattices

Definition: A lattice is a partial order in which every two-element set has a least upper bound and a greatest lower bound.

Lemma: In a lattice every non-empty finite set has a lub (\sqcup) and glb (\sqcap).

Lattices

Definition: A lattice is a partial order in which every two-element set has a least upper bound and a greatest lower bound.

Lemma: In a lattice every non-empty finite set has a lub (\sqcup) and glb (\sqcap).

Proof: is by induction!

Case where the set S has three elements x, y and z :

Let $a = (x \sqcup y) \sqcup z$.

By definition of \sqcup we have $z \sqsubseteq a$ and $x \sqcup y \sqsubseteq a$.

Then we have again by definition of \sqcup , $x \sqsubseteq x \sqcup y$ and $y \sqsubseteq x \sqcup y$. Thus by transitivity we have $x \sqsubseteq a$ and $y \sqsubseteq a$.

Thus we have $S \sqsubseteq a$ and a is an upper bound.

Now suppose that there exists a' such that $S \sqsubseteq a'$. We want $a \sqsubseteq a'$ (a least upper bound):

We have $x \sqsubseteq a'$ and $y \sqsubseteq a'$, thus $x \sqcup y \sqsubseteq a'$. But $z \sqsubseteq a'$, thus $((x \sqcup y) \sqcup z) \sqsubseteq a'$.

Thus a is the lub of our 3 elements set.

Examples of Lattices

Lemma: Every linear order is a lattice.

Example: Every bounded subset of the set of real numbers has a lub. This is an axiom of real numbers, the way they are defined (or constructed from rationals).

- ▶ If a lattice has least and greatest element, then every finite set (including empty set) has a lub and glb.
- ▶ This does not imply there are lub and glb for infinite sets.

Example: In the order $([0, 1], \leq)$ with standard ordering on reals is a lattice, the entire set has no lub. The set of all rationals of interval $[0, 10]$ is a lattice, but the set $\{x \mid 0 \leq x \wedge x^2 < 2\}$ has no lub.

Exercises

Prove the following:

1. $(x \sqcup y) \sqcup z = x \sqcup (y \sqcup z)$
2. $\sqcup A \sqsubseteq \sqcap B \Leftrightarrow \forall x \in A. \forall y \in B. x \sqsubseteq y$
3. Let (A, \sqsubseteq) be a partial order such that every set $S \subseteq A$ has the greatest lower bound. Prove that then every set $S \subseteq A$ has the least upper bound.

Solution

1. Let $(x \sqcup y) \sqcup z = a$, then $z \sqsubseteq a$ and $x \sqcup y \sqsubseteq a$ and hence $x \sqsubseteq a$ and $y \sqsubseteq a$. From $z \sqsubseteq a$ and $y \sqsubseteq a$ it follows $z \sqcup y \sqsubseteq a$ and finally $x \sqcup (y \sqcup z) \sqsubseteq a = (x \sqcup y) \sqcup z$.
Symmetrically, we also get that $(x \sqcup y) \sqcup z \sqsubseteq x \sqcup (y \sqcup z)$ and hence $(x \sqcup y) \sqcup z = x \sqcup (y \sqcup z)$.
2. Let $a = \sqcup A$ and $b = \sqcap B$. Then

$$a = \sqcup A \Leftrightarrow \forall x \in A. x \sqsubseteq a \wedge \forall a'. \forall x \in A. x \sqsubseteq a' \rightarrow a \sqsubseteq a'$$

and

$$b = \sqcap B \Leftrightarrow \forall y \in B. b \sqsubseteq y \wedge \forall b'. \forall y \in B. b' \sqsubseteq y \rightarrow b' \sqsubseteq b$$

\Rightarrow

Suppose $a \sqsubseteq b$, then by transitivity it follows that $\forall x \in A. \forall y \in B. x \sqsubseteq y$.

\Leftarrow

Now suppose $\forall x \in A. \forall y \in B. x \sqsubseteq y$. This implies that the set A is a set of lower bounds to B , and B is a set of upper bounds on A . Hence $a \sqsubseteq b$ (intuitively).

Solution

3. Suppose we have some arbitrary set $S \subseteq A$, then we know that $\sqcap S$ exists.

Let U be the set of all its upper bounds, i.e. $U = \{x \mid \forall y \in S. y \sqsubseteq x\}$. Since every subset of A has a greatest lower bound, we know that $a = \sqcap U$ exists. Now we want to show that a is an upper bound on S and that it is the least one.

We want to show that $\forall y \in S. y \sqsubseteq a$.

Let L be the set of lower bounds on U , i.e. $L = \{z \mid \forall x \in U. z \sqsubseteq x\}$.

Clearly, $S \subseteq L$, since U are the upper bounds on S .

Then a is the greatest element in L , and thus $\forall y \in S. y \sqsubseteq a$ and so a is an upper bound on S .

Now take any upper bound $x \in U$. Since $a = \sqcap U$, we have $a \sqsubseteq x$ and so a is the least upper bound.

Constructing Partial Orders using Maps

Example: Let A be the set of all propositional formulas containing only variables p, q . For a formula $F \in A$ define

$$[F] = \{(u, v). u, v \in \{0, 1\} \wedge F \text{ is true for } p \mapsto u, q \mapsto v\}$$

i.e. $[F]$ denotes the set of assignments for which F is true. Note that $F \implies G$ is a tautology iff $[F] \subseteq [G]$. Define ordering on formulas A by

$$F \leq G \iff [F] \subseteq [G]$$

Is \leq a partial order? Which laws does \leq satisfy?

Constructing Partial Orders using Maps

Lemma: Let (C, \leq) be a lattice and A a set. Let $\gamma : A \rightarrow C$ be an injective function. Define order $x \sqsubseteq y$ on A by $\gamma(x) \leq \gamma(y)$. Then (A, \sqsubseteq) is a partial order.

Note: even if (C, \leq) had top and bottom element and was a lattice, the constructed order need not have top and bottom or be a lattice. For example, we take A to be a subset of A and define γ to be identity.

Lattices

Definition: A lattice is a partial order in which every two-element set has a least upper bound and a greatest lower bound (so, we have \sqcap and \sqcup as well-defined binary operations).

Lemma: In every lattice, $x \sqcup (x \sqcap y) = x$.

Lattices

Definition: A lattice is a partial order in which every two-element set has a least upper bound and a greatest lower bound (so, we have \cap and \sqcup as well-defined binary operations).

Lemma: In every lattice, $x \sqcup (x \cap y) = x$.

Proof:

We trivially have $x \sqsubseteq x \sqcup (x \cap y)$.

Let's prove that $x \sqcup (x \cap y) \sqsubseteq x$:

x is an upper bound of x and $x \cap y$, $x \sqcup (x \cap y)$ is the least upper bound of x and $x \cap y$, thus $x \sqcup (x \cap y) \sqsubseteq x$.

Definition: A lattice is *distributive* iff

$$x \cap (y \sqcup z) = (x \cap y) \sqcup (x \cap z)$$

$$x \sqcup (y \cap z) = (x \sqcup y) \cap (x \sqcup z)$$

Lattice of all subsets of a set is distributive. Linear order is a distributive lattice.

Products of Lattices

Note: for $n = 2$ a function $f : \{1, 2\} \rightarrow (L_1 \cup L_2)$ with $f(1) \in L_1$, $f(2) \in L_2$ is isomorphic to an ordered pair $(f(1), f(2))$. We denote the product by $(L_1, \leq_1) \times (L_2, \leq_2)$.

Example: Let $R = \{a, b, c, d\}$ denote set of values. Let $A_1 = A_2 = 2^R$. Let

$$s_1 \leq_1 s_2 \iff s_1 \subseteq s_2$$

and let

$$t_1 \leq_2 t_2 \iff t_1 \supseteq t_2$$

Then we can define the product $(A_1, \leq_1) \times (A_2, \leq_2)$. In this product, $(s_1, t_1) \leq (s_2, t_2)$ iff: $s_1 \subseteq s_2$ and $t_1 \supseteq t_2$. The original partial orders were lattices, so the product is also a lattice. For example, we have

$$(\{a, b, c\}, \{a, b, d\}) \sqcap (\{b, c, d\}, \{c, d\}) = (\{b, c\}, \{a, b, c, d\})$$

Products of Lattices

Lattice elements can be combined into finite or infinite-dimensional vectors, and the result is again a lattice.

Lemma: Let $(A_1, \leq_1), \dots, (A_n, \leq_n)$ be partial orders. Define (A, \leq) by

$$A = \{f \mid f : \{1, \dots, n\} \rightarrow (A_1 \cup \dots \cup A_n) \text{ where } \forall i. f(i) \in A_i\}$$

For $f, g \in A$ define

$$f \leq g \iff \forall i. f(i) \leq_i g(i)$$

Then (A, \leq) is a partial order. We denote (A, \leq) by

$$\prod_{i=1}^n (L_i, \leq_i)$$

Moreover, if for each i , (A_i, \leq_i) is a lattice, then (A, \leq) is also a lattice.

Properties of $\sqcap S$ and $\sqcup S$

Consider a partial order (A, \sqsubseteq) .

- ▶ Suppose $S_1 \subseteq S_2 \subseteq A$ and $\sqcup S_1$ and $\sqcup S_2$ exist. In what relationship are these two elements?
- ▶ Suppose $S_1 \subseteq S_2 \subseteq A$ and $\sqcap S_1$ and $\sqcap S_2$ exist. In what relationship are these two elements?
- ▶ Suppose $\sqcup \emptyset$ exists. Describe this element.
- ▶ Suppose $\sqcap \emptyset$ exists. Describe this element.

Properties of $\sqcap S$ and $\sqcup S$

Consider a partial order (A, \sqsubseteq) .

- ▶ Suppose $S_1 \subseteq S_2 \subseteq A$ and $\sqcup S_1$ and $\sqcup S_2$ exist. In what relationship are these two elements?
- ▶ Suppose $S_1 \subseteq S_2 \subseteq A$ and $\sqcap S_1$ and $\sqcap S_2$ exist. In what relationship are these two elements?
- ▶ Suppose $\sqcup \emptyset$ exists. Describe this element.
- ▶ Suppose $\sqcap \emptyset$ exists. Describe this element.

$\sqcup \emptyset = \perp$ and $\sqcap \emptyset = \top$. This is because every element is an upper bound and a lower bound of \emptyset : $\forall x. \forall y \in \emptyset. y \sqsubseteq x$ is valid, as well as $\forall x. \forall y \in \emptyset. y \sqsupseteq x$.

Complete Semilattice is a Complete Lattice

If we have all \sqcap -s we then also have all \sqcup -s:

Theorem: Let (A, \sqsubseteq) be a partial order such that every set $S \subseteq A$ has the greatest lower bound (\sqcap). Prove that then every set $S \subseteq A$ has the least upper bound (\sqcup).

Example: Application of the Previous Theorem

Let U be a set and $A \subseteq U \times U$ the set of all **equivalence relations** on this set. Consider the partial order (A, \subseteq) .

Lemma

If $I \subseteq A$ is a set of equivalence relations, then $\cap I$ is also an equivalence relation.

Consequence: Given $I \subseteq A$ there exists the least equivalence relation containing every relation from I (equivalence closure of relations in I).

Note: **congruence** is equivalence relation that agrees with some operations. For example, $x \sim x'$ and $y \sim y'$ implies $(x + y) \sim (x' + y')$. The analogous properties hold for congruence relations.

Complete Lattices

Definition: A **complete** lattice is a lattice where for every set S (including empty set and infinite sets) there exist $\sqcup S$ and $\sqcap S$.

Monotonic functions

Given two partial orders (C, \leq) and (A, \sqsubseteq) , we call a function $\alpha : C \rightarrow A$ *monotonic* iff for all $x, y \in C$,

$$x \leq y \rightarrow \alpha(x) \sqsubseteq \alpha(y)$$