

Exercise 01

2021/10/19

Reading Material

Please read the part about lattices in Sect. 16.3.1 of the [Handbook on Model Checking](#) (definition and example).

1 From Partial Orders to Lattices

PO Candidate	is a PO?	finite height?	height	\top <i>top</i>	\perp <i>bottom</i>
$(\mathbb{Z}, \leq) = \{(a, b) \in \mathbb{Z}^2 : b - a \in \mathbb{N}\}$	Yes	No	∞		
$([-10, 10], \leq)$	Yes	Yes	20	10	-10
$([-10, 10], <)$	No	Yes	20		
$(\mathbb{Z}, \geq) = \{(a, b) \in \mathbb{Z}^2 : a - b \in \mathbb{N}\}$	Yes	No	∞		
$(\mathbb{Z}, =) = \{(a, b) \in \mathbb{Z}^2 : a - b = 0\}$	No	Yes	0		
$(2^Q, \subseteq), Q \text{ finite}$	Yes	No	∞	Q	\emptyset
$(\Sigma^*, \text{lexicographic order})$	Yes	No	∞		\emptyset
$([a - z]^3, \text{lexicographic order})$	Yes	Yes	26		\emptyset
$(\bigcup_{0 \leq i \leq 3} [a - z]^i, \text{lexicographic order})$	Yes	Yes	78		\emptyset
$(\Sigma^*, \text{suffix})$	Yes	No	∞		\emptyset
$([a - z]^3, \text{suffix})$	Yes	Yes	26		\emptyset
$(\bigcup_{0 \leq i \leq 3} [a - z]^i, \text{suffix})$	Yes	Yes	78		\emptyset

(a) Which three properties need to be fulfilled by a set \mathbb{X} and subsumption relation \sqsubseteq to make a partial order? *Reflexive, transitive, antisymmetric*

(b) Decide for each of the listed partial order candidates whether it actually is a partial order. Note: $[-10, 10]$ should be treated as an interval of integers.

(c) The height n of a partial order $(\mathbb{X}, \sqsubseteq)$ is defined as the number of subsumption operations in the longest ascending chain $x_1 \sqsubseteq x_2 \cdots \sqsubseteq x_{n+1}$ that can be constructed from pairwise different elements x_i of the underlying set \mathbb{X} . (As such, the height is always one smaller than the length of the longest ascending chain).

Decide for each of the tuples $(\mathbb{X}, \sqsubseteq)$ identified in (b) as partial orders whether it has a finite height. If so, also write down a formula that describes the value of that finite height.

(d) Which properties need to be fulfilled by a lattice? *a partial order, top, bottom, LUB, GLB*

(e) Which of the identified partial orders can be extended into (semi-)lattices? For those who can, write down the \top and \perp element.

$(2^Q, \subseteq)$ finite, $([-10, 10], \leq)$, $([-10, 10], <)$

2 Upper and Lower Bounds

1. Consider the partial order $(\mathbb{N}, \sqsubseteq)$ with $x \sqsubseteq y$ if $x \leq y \wedge x \bmod 2 = y \bmod 2$. Compute for each of the following subsets the least upper bound and the greatest lower bound if they exist.

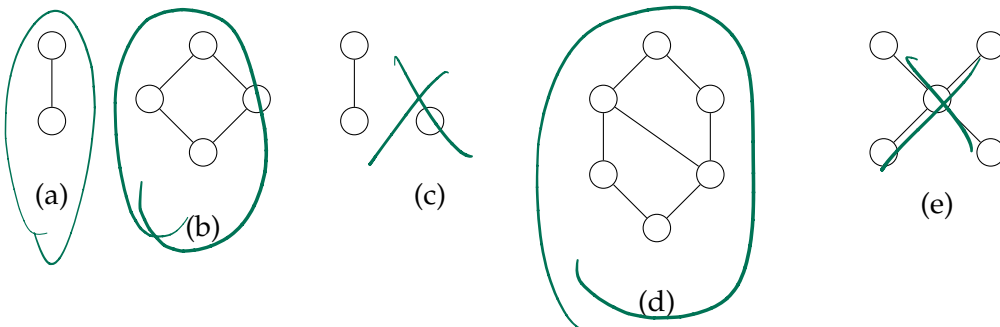
- a) $\{1, 2\}$ $LUB \hat{=} 1$ $\&LUB \hat{=} 2$
 b) $\{2, 4, 8\}$ $LUB \hat{=} 2$ $\&LUB \hat{=} 8$
 c) $\{x \mid x \bmod 2 = 0\}$ not exist
 d) \emptyset not exist
 e) \mathbb{N} not exist

2. Consider the following partial orders. What are the least upper bound and greatest lower bound operators for finite, non-empty subsets?

- a) $(2^Q, \supseteq)$ $LUB \hat{=} Q$ $\&LUB \hat{=} \emptyset$
 b) $(\mathbb{N}, \sqsubseteq_{\text{div}})$ with $x \sqsubseteq_{\text{div}} y$ if $y \bmod x = 0$. $LUB \hat{=} N$, $\&LUB \hat{=} \emptyset$

3 Lattices

Consider the following partial orders. Which of them are lattices?



4 Concretization

For describing the states of the system we are interested in, we model it with so-called concrete states from the set C of all concrete states.

A concrete state $c \in C$ is a pair of program counter and data state ($C = L \times \Sigma$). The second part of a concrete state is also sometimes referred to as the data state.

- Program counter $pc \in L$
 - Where am I?
 - Location in CFA
 - $c(pc)$ refers to program counter of concrete state
- Data state $\sigma \in \Sigma, \Sigma = V \rightarrow \mathbb{Z}$ (σ is a function that maps variables $v \in V$ to concrete values)
 - Fixes variable values

– $c(v)$ refers to the value that is assigned to variable v in the concrete state c

1. Consider the flat lattice \mathcal{L} derived from the set L of program locations. Define a concretization function for the lattice \mathcal{L} that maps each element to all concrete states with a program counter considered by this element. Consider also the \top and \perp element of the lattice.

5 Programming Exercise on Lattices: Lattice of Prime Factors

Go to the lecture's notebooks repository¹ and complete the programming tasks in the Jupyter Notebook named `01_Lattices.ipynb`.

You can either click on the badge labeled "launch binder" that will open the notebook in a clean environment on mybinder.org, or you can download the notebook and run it from your personal computer using **Jupyter** according to the installation instructions in the repository. Binder will sometimes take a while to create the environment, especially when a notebook recently changed or is not used that frequently.

All the necessary text is given in the notebook, here is a rough sketch of what awaits you:

We have a look at a lattice where the elements consist of numbers that are created by multiplying the numbers in each subset of the set $\{1,2,3,5\}$. For example, $\{3,5\}$ is a subset of $\{1,2,3,5\}$, so $3 \cdot 5 = 15$ is an element of the lattice.

The exercise contains three sub-exercises:

- (a) The subsumption relation:

In order to get a nice lattice, we have to define a better subsumption relation. Your task is to implement in the next cell the subsumption relation that where $x \sqsubseteq y$ iff the prime factors of x are a subset of the prime factors of y . Execute the next three cells and observe the output. If you are not satisfied with your implementation of 'subsumption', refine it until you are happy.

- (b) remove redundant edges to increase readability:

Some of the edges in the graph generated in a) are redundant, they could be left out because of the transitivity of the subsumption relation. For example, $1 \sqsubseteq 6$ could be left out because it follows from $1 \sqsubseteq 2$ and $2 \sqsubseteq 6$. To reflect this, let's add another function that returns *True* whenever y is a direct "successor" of x , i.e.: $successor(x, y) \Leftrightarrow x \sqsubset y \wedge \nexists x'. x \sqsubset x' \wedge x' \sqsubset y$

- (c) \top and \perp :

From the graph generated in b) it should be quite clear which element is \top and which element is \perp . Can you determine this programmatically? Your solution should work even if we modified the set s by adding other primes like 7,11,13.

¹<https://gitlab2.cip.ifi.lmu.de/sosy-lab/sv-notebooks-ws21>