Consider the recent renovation of Avery Hall. In this process several things had to be done.

- Remove Asbestos
- Replace Windows
- Paint Walls
- Refinish Floors
- Assign Offices
- Move in Office-Furniture.

On the other hand, several things could have been done concurrently—painting could be done while replacing the windows and assigning office could have been done at anytime.

Such a scenario can be nicely modeled using partial orderings.

We use the notation $a \preccurlyeq b$ to indicate that $(a, b) \in R$ is a partial order and $a \prec b$ when $a \neq b$.

The notation $\prec$ is not to be mistaken for “less than equal to.” Rather, $\prec$ is used to denote any partial ordering.

Latex notation: \preccurlyeq, \prec.
**Total Orders**

**Definition**
If \((S, \preceq)\) is a poset and every two elements of \(S\) are comparable, \(S\) is called a **totally ordered set**. The relation \(\preceq\) is said to be a **total order**.

**Example**
The set of integers over the relation “less than equal to” is a total order; \((\mathbb{Z}, \leq)\) since for every \(a, b \in \mathbb{Z}\), it must be the case that \(a \leq b\) or \(b \leq a\).

What happens if we replace \(\leq\) with \(<\)?

---

**Well-Orderings**

**Definition**
\((S, \preceq)\) is a **well-ordered set** if it is a poset such that \(\preceq\) is a total ordering and such that every nonempty subset of \(S\) has a **least element**.

**Example**
The natural numbers along with \(\leq\), \((\mathbb{N}, \leq)\) is a well-ordered set since any subset of \(\mathbb{N}\) will have a least element and \(\leq\) is a total ordering on \(\mathbb{N}\) as before.

However, \((\mathbb{Z}, \leq)\) is not a well-ordered set. Why? Is it totally ordered?

---

**Principle of Well-Ordered Induction**

**Definition**
Well-ordered sets are the basis of the proof technique known as induction (more when we cover Chapter 3).

**Theorem (Principle of Well-Ordered Induction)**

Suppose that \(S\) is a well ordered set. Then \(P(x)\) is true for all \(x \in S\) if

- **Basis Step**: \(P(x_0)\) is true for the least element of \(S\) and
- **Induction Step**: For every \(y \in S\) if \(P(x)\) is true for all \(x < y\) then \(P(y)\) is true.

---

**Lexicographic Orderings I**

Lexicographic ordering is the same as any dictionary or phone book—we use alphabetical order starting with the first character in the string, then the next character (if the first was equal) etc. (you can consider “no character” for shorter words to be less than “a”).

---

**Lexicographic Orderings II**

Formally, lexicographic ordering is defined by combining two other orderings.

**Definition**
Let \((A_1, \preceq_1)\) and \((A_2, \preceq_2)\) be two posets. The **lexicographic ordering** \(\preceq\) on the Cartesian product \(A_1 \times A_2\) is defined by

\[(a_1, a_2) \preceq (a'_1, a'_2)\]

if \(a_1 \preceq_1 a'_1\) or if \(a_1 = a'_1\) and \(a_2 \preceq_2 a'_2\).

If we add equality to the lexicographic ordering \(<\) on \(A_1 \times A_2\), we obtain a partial ordering \(\preceq\).
Lexicographic Orderings III

Lexicographic ordering generalizes to the Cartesian product of \( n \) sets in the natural way.

Define \( \preceq \) on \( A_1 \times A_2 \times \cdots \times A_n \) by
\[
(a_1, a_2, \ldots, a_n) \preceq (b_1, b_2, \ldots, b_n)
\]
if \( a_1 < b_1 \) or if there is an integer \( i > 0 \) such that
\[
a_1 = b_1, a_2 = b_2, \ldots, a_i = b_i
\]
and \( a_{i+1} < b_{i+1} \)

Hasse Diagrams

As with relations and functions, there is a convenient graphical representation for partial orders—Hasse Diagrams.

Consider the digraph representation of a partial order—since we know we are dealing with a partial order, we implicitly know that the relation must be reflexive and transitive. Thus we can simplify the graph as follows:

- Remove all self-loops.
- Remove all transitive edges.
- Make the graph direction-less—that is, we can assume that the orientations are upwards.

The resulting diagram is far simpler.

Lexicographic Orderings I

Strings

Consider the two non-equal strings \( a_1a_2\cdots a_m \) and \( b_1b_2\cdots b_n \) on a poset \( S \).

Let \( t = \min(m, n) \) and \( \prec \) is the lexicographic ordering on \( S^t \).
a_1a_2\cdots a_m \) is less than \( b_1b_2\cdots b_n \) if and only if

- \((a_1, a_2, \ldots, a_t) \prec (b_1, b_2, \ldots, b_t)\), or
- \((a_1, a_2, \ldots, a_t) = (b_1, b_2, \ldots, b_t)\) and \( m < n \)
Hasse Diagram
Example

Of course, you need not always start with the complete relation in the partial order and then trim everything. Rather, you can build a Hasse directly from the partial order.

Example

Draw a Hasse diagram for the partial ordering
\{(a, b) \mid a \mid b\}
on \{1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60\} (these are the divisors of 60 which form the basis of the ancient Babylonian base-60 numeral system)

Extremal Elements I
Summary

We will define the following terms:

- A maximal/minimal element in a poset \((S, \preceq)\).
- The maximum (greatest)/minimum (least) element of a poset \((S, \preceq)\).
- An upper/lower bound element of a subset \(A\) of a poset \((S, \preceq)\).
- The greatest lower/least upper bound element of a subset \(A\) of a poset \((S, \preceq)\).
- Lattice

Definition

An element \(a\) in a poset \((S, \preceq)\) is called maximal if it is not less than any other element in \(S\). That is,
\[ \not\exists b \in S (a \prec b) \]

If there is one unique maximal element \(a\), we call it the maximum element (or the greatest element).

Extremal Elements II
Definition

An element \(a\) in a poset \((S, \preceq)\) is called minimal if it is not greater than any other element in \(S\). That is,
\[ \not\exists b \in S (b \prec a) \]

If there is one unique minimal element \(a\), we call it the minimum element (or the least element).
**Extremal Elements**

**Definition**

Let $(S, \preceq)$ be a poset and let $A \subseteq S$. If $u$ is an element of $S$ such that $a \preceq u$ for all elements $a \in A$ then $u$ is an upper bound of $A$.

An element $a$ that is an upper bound on a subset $A$ and is less than all other upper bounds on $A$ is called the least upper bound on $A$. We abbreviate “lub”.

**Extremal Elements**

**Example I**

What are the minimal, maximal, minimum, maximum elements?

- Minimal: $\{a, b\}$
- Maximal: $\{c, d\}$
- There are no unique minimal or maximal elements.

**Example II**

What are the lower/upper bounds and glb/lub of the sets $\{d, e, f\}$, $\{a, c\}$ and $\{b, d\}$?

- $\{d, e, f\}$
  - Lower Bounds: $\emptyset$, thus no glb either.
  - Upper Bounds: $\emptyset$, thus no lub either.

- $\{a, c\}$
  - Lower Bounds: $\emptyset$, thus no glb either.
  - Upper Bounds: $\{h\}$, since its unique, lub is also $h$.

- $\{b, d\}$
  - Lower Bounds: $\{b\}$ and so also glb.
  - Upper Bounds: $\{d, g\}$ and since $d \prec g$, the lub is $d$.

**Example III**

Minimal/Maximal elements?

- Maximal Elements: $b, d, i, j$.
- Bounds, glb, lub of $\{c, e\}$?
  - Lower Bounds: $\{a, c\}$, thus glb is $c$.
  - Upper Bounds: $\{d, e, f, g, h, i, j\}$ thus lub is $e$.
- Bounds, glb, lub of $\{b, i\}$?
  - Lower Bounds: $\{a\}$, thus glb is $a$.
  - Upper Bounds: $\emptyset$, thus lub DNE.
Lattices

A special structure arises when every pair of elements in a poset has a lub and glb.

**Definition**
A partially ordered set in which every pair of elements has both a least upper bound and a greatest lower bound is called a *lattice*.

---

**Example**

Is the example from before a lattice?

```
      i   j
     / \  /\
    f   g h
   /   / /
  e   c d
 /   /  /
 b   a b
```

No, since the pair \((b, c)\) do not have a least upper bound.

---

**Example**

What if we modified it as follows?

```
      i   j
     / \  /\
    f   g h
   /   / /
  e   c d
 /   /  /
 b   a b
```

Yes, it is now a lattice, since for any pair, there is a lub & glb.

---

Topological Sorting

**Introduction**

Let us return to the introductory example of the Avery renovation. Now that we have got a partial order model, it would be nice to actually create a concrete schedule.

That is, given a partial order, we would like to transform it into a total order that is compatible with the partial order.

A total order is compatible if it doesn’t violate any of the original relations in the partial ordering.

Essentially, we are simply imposing an order on incomparable elements in the partial order.

---

Preliminaries

Before we give the algorithm, we need some tools to justify its correctness.

**Fact**
Every finite, nonempty poset \((S, \preceq)\) has a minimal element.

We will prove by a form of reductio ad absurdum.
Preliminaries

Proof

Assume to the contrary that a nonempty, finite (WLOG, assume $|S| = n$) poset $(S \preceq)$ has no minimal element. In particular, $a_1$ is not a minimal element.

If $a_1$ is not minimal, then there exists $a_2$ such that $a_2 \prec a_1$. But also, $a_2$ is not minimal by the assumption.

Therefore, there exists $a_3$ such that $a_3 \prec a_2$. This process proceeds until we have the last element, $a_n$ thus,

$$a_n \prec a_{n-1} \prec \cdots \prec a_2 \prec a_1$$

thus by definition $a_n$ is the minimal element.

### Topological Sorting

Intuition

The idea to topological sorting is that we start with a poset $(S, \preceq)$ and remove a minimal element (choosing arbitrarily if there are more than one). Such an element is guaranteed to exist by the previous fact.

As we remove each minimal element, the set shrinks. Thus, we are guaranteed the algorithm will halt in a finite number of steps.

Furthermore, the order in which elements are removed is a total order;

$$a_1 \prec a_2 \prec \cdots \prec a_n$$

We now present the algorithm itself.

#### Algorithm

**TOPOLLOGICAL SORT**

| INPUT   | $(S, \preceq)$ a poset with $|S| = n$ |
|---------|-----------------------------------|
| OUTPUT  | A total ordering $(a_1, a_2, \ldots, a_n)$ |
| $k = 1$ |                                     |
| **while** $S \neq \emptyset$ **do** | |
| $a_k \leftarrow$ a minimal element in $S$ | |
| $S = S \setminus \{a_k\}$ | |
| $k = k + 1$ | |
| **end** | |
| **return** $(a_1, a_2, \ldots, a_n)$ | |

#### Example

Find a compatible ordering (topological ordering) of the poset represented by the diagram below.

![Diag_1.png](attachment:Diag_1.png)

Questions?