Goals of the lecture

- Relations
- Posets
- A run or a distributed computation
- Happened-before relation
Model of Distributed systems

- **events**
  - beginning of procedure foo
  - termination of bar
  - send of a message
  - receive of a message
  - termination of a process

- **happened-before relation**
Relation

- $X = \text{any set}$
  a binary relation $R$ is a subset of $X \times X$.

- Example: $X = \{a, b, c\}$, and
  $R = \{(a, c), (a, a), (b, c), (c, a)\}$. 

![Diagram](attachment:image.png)
Relation [Contd.]

**Reflexive:** If for each \( x \in X \), \( (x, x) \in R \).
- Example: \( X \) is the set of natural numbers, and \( R = \{(x, y) \mid x \text{ divides } y\} \).

**Irreflexive:** For each \( x \in X \), \( (x, x) \notin R \).
- Example: \( X \) is the set of natural numbers, and \( R = \{(x, y) \mid x \text{ less than } y\} \).

**Reflexive or irreflexive?**
Relation [Contd.]

**Symmetric**: \((x, y) \in R \) implies \((y, x) \in R\).
- Examples: is sibling of, \(x \mod k = y \mod k\).

**Anti-symmetric**: \((x, y) \in R, (y, x) \in R\) implies \(x = y\).
- Examples: \(\leq\), divides.

**Asymmetric**: \((x, y) \in R\) implies \((y, x) \not\in R\).
- Examples: is child of, \(<\).
Relation [Contd.]

Transitive:  \((x, y), (y, z) \in R\) implies \((x, z) \in R\).

- Examples: is reachable from, \(<\), divides.

Puzzle: Example of a symmetric and transitive but not reflexive relation.
Partially Ordered Sets [Posets]

Partial Order

Reflexive
Transitive
Anti-symmetric
Example: \( \leq \)

Irreflexive
Transitive
Anti-symmetric
Example: <

Examples:

- \( X \): Ground Set, \((2^X, \subseteq)\) is a irreflexive partial order
- \((\mathbb{N}, \text{divides})\) is a reflexive partial order
- \((\mathcal{R}, \leq)\) is a reflexive partial order (also a total order)
- causality in a distributed system (later ..)
Posets [Contd.]

Let $Y \subseteq X$, where $(X, \leq)$ is a poset.

**Infimum:** $m = \inf(Y)$ iff

- $\forall y \in Y : m \leq y$
- $\forall x \in X : (\forall y \in Y : x \leq y) \Rightarrow x \leq m$

$m$ is also called *glb* of the set $Y$.

**Supremum:** $s = \sup(Y)$ iff ($s$ is also called *lub*)

- $\forall y \in Y : y \leq s$
- $\forall x \in X : (\forall y \in Y : y \leq s) \Rightarrow s \leq x$

We denote the *glb* of $\{a, b\}$ by $a \sqcap b$, and *lub* by $a \sqcup b$.

$X = \{a, b, c, d, e, f\}$

$R = \{(a, b), (a, c), (b, d), (c, f), (c, e), (d, e)\}$
Lattices

- Let $S$ be any set, and $2^S$ be its power set. The poset $(2^S, \subseteq)$ is a lattice.
- Set of rationals with usual $\leq$.
- Set of global states

- A lattice is an algebraic system $(L, \sqcup, \sqcap)$ where $\sqcup$ and $\sqcap$ satisfy commutative, associative and absorption laws.
Monotone functions

A function \( f : X \rightarrow Y \) is monotone iff

\[
\forall x, y \in X : x \leq y \Rightarrow f(x) \leq f(y).
\]

- Examples
  - union, intersection
  - addition, multiplication with positive number
  - clocks in distributed systems
Down-Sets and Up-Sets

Let \((X, <)\) be any poset.

- We call a subset \(Y \subseteq X\) a down-set (alternatively, order ideal) if
  \[ f \in Y \land e < f \Rightarrow e \in Y. \]

- Similarly, we call \(Y \subseteq X\) an up-set (alternatively, order filter) if
  \[ e \in Y \land e < f \Rightarrow f \in Y. \]

- We use \(\mathcal{O}(X)\) to denote the set of all down-sets of \(X\).
  We now show a simple but important lemma.

Lemma 1 Let \((X, <)\) be any poset. Then, \((\mathcal{O}(X), \subseteq)\) is a lattice.
Run

- Each process $P_i$ in a run generates an execution trace $s_{i,0}e_{i,0}s_{i,1} \ldots e_{i,l-1}s_{i,l}$, which is a finite sequence of local states and events in the process $P_i$.
  - state = values of all variables, program counter
  - event = internal, send, receive

- A run $r$ is a vector of traces with $r[i]$ as the trace of the process $P_i$. 

\[ r[1] \quad 0,1 \quad 1,3 \quad \text{send}(x) \quad 2,3 \quad 3,2 \quad (pc, x) \]

\[ r[2] \quad 0,1 \quad 1,4 \quad \text{receive}(y) \quad 2,3 \quad 3,6 \quad (pc, y) \]
Relations

- $s \prec_1 t$ if and only if $s$ immediately precedes $t$ in the trace $r[i]$.
  - $s.next = t$ or $t.prev = s$ whenever $s \prec_1 t$.
  - $\prec = \text{irreflexive transitive closure of } \prec_1$.
  - $\preceq = \text{reflexive transitive closure of } \prec_1$.

- event $e$ in the trace $r[i] \rightsquigarrow$ event $f$ in the trace $r[j]$ if $e$ is the send of a message and $f$ is the receive event of the same message.
Relations [Contd.]

Causally precedes relation \( \equiv \) the transitive closure of union of \( \prec_1 \) and \( \sim \). That is, \( s \rightarrow t \) iff

1. \( (s \prec_1 t) \lor (s \sim t) \), or
2. \( \exists u : (s \rightarrow u) \land (u \rightarrow t) \)

\( s \) and \( t \) are concurrent (denoted by \( s \parallel t \)) if \( \neg(s \rightarrow t) \land \neg(t \rightarrow s) \).