Lecture 11: Relations and Partial Orders

Reading: 7.1.1, 7.4–7.8

Theme: Model dependencies as directed graphs, how do we schedule?

Key definition:

def: A relation R between a set A and itself is $R \subseteq A \times A$

e.g. $\{(x,y) \mid x, y \in \mathbb{N}^+ \text{ and } x \text{ divides } y\}$

$\{(x,y) \mid x, y \in \mathbb{Z} \text{ and } x \equiv y \pmod{k}\}$

$\{(x,y) \mid x, y \in \mathbb{Z} \text{ and } x \leq y\}$

We've seen relations many times, sometimes useful to study their general properties

transitive: xRy and $yRz \Rightarrow xRz$

e.g. $x \mid y$ and $y \mid z \Rightarrow x \mid z$, which other ones of examples? all

symmetric: $xRy \Rightarrow yRx$

e.g. $x \equiv y \pmod{k} \Rightarrow y \equiv x \pmod{k}$

any others? no

antisymmetric: xRy and $yRx \Rightarrow x = y$

e.g. $x \mid y$ and $y \mid x \Rightarrow x = y$

<u>reflexive</u>: $\forall x,\ xRx$

We will be interested is special relations

<u>def</u>: A <u>partially ordered set</u> (poset) is a relation that is transitive, antisymmetric and reflexive

which example is a poset? $x \leq y$

More interesting example:

$$\begin{tikzcd}
& 11 & \\
01 \arrow[ur] & & 10 \arrow[ul] \\
& 00 \arrow[ul] \arrow[ur] &
\end{tikzcd}
\qquad A = \{00, 01, 10, 11\}$$

$x \leq y$ iff $x_1 \leq y_1$ and $x_2 \leq y_2$

This is called a <u>Hasse diagram</u>: arrows from smaller to larger elements

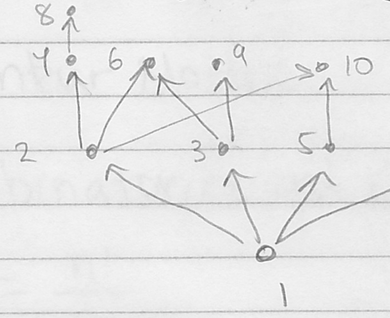
<u>def'</u>: Given an acyclic directed graph on nodes A, the corresponding poset is

$x \leq y$ iff ~~x=y or~~ there is a directed path from x to y could be length zero

can omit arrows →

Let's get some practice:

Example: $A = \{1, 2, \dots 10\}$ with $x \preceq y$ iff $x | y$

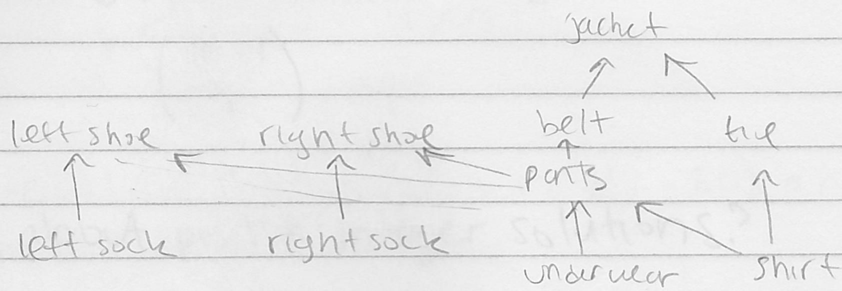


Notice any path from x to y implies $x | y$ (e.g. $2 \to 4 \to 8$)

Posets naturally arise in scheduling

Example: Getting dressed: {left sock, right sock, left shoe, right shoe, underwear, pants, shirt, tie, belt, jacket}

$x \preceq y$ iff x needs to be put on before y



These are the dependencies, but where do we start?

Lemma: Every finite poset has a minimal element

e.g. $x \in A$ so that $\forall y \in A \; x \preceq y$

First a useful definition

<u>def:</u> A <u>chain</u> is a sequence of distinct elements with $a_1 \preceq a_2 \preceq \ldots \preceq a_k$ (length k)
end

<u>Proof:</u> Let $a_1 \preceq a_2 \ldots \preceq a_k$ be the longest chain (or tied for longest).

claim ~~then~~ a_1 is minimal. <u>Proof:</u> Suppose not, then there is an a_0 that does not appear in the chain with $a_0 \preceq a_1$. But then $a_0 \preceq a_1 \ldots \preceq a_k$ is a longer chain, contradicting our assumption ☒

☒

Is the lemma true for infinite posets?

Where does the proof break down?

Now that we know where to start, how do we schedule?

(Do example)

This is called a topological sort:

<u>def:</u> Given a poset $\preceq$, a topological sort $\preceq_T$ is a <u>total ordering</u> (every pair of elements, either $x \preceq_T y$ or $y \preceq_T x$) with

$$x \preceq y \Rightarrow x \preceq_T y$$

e.g. (do clothes example)

Theorem: Every finite poset has a topological sort

Proof: (By induction)

$P(n) \triangleq$ Theorem is true for posets on n elements

Inductive Step: Let $(A, \preceq)$ be the poset on $n+1$ elements, let x be a minimal element

By induction poset on $A \setminus \{x\}$ has a topological sort

$$(x \preceq_T)\, a_1 \preceq_T a_2 \preceq_T \dots\ a_n$$

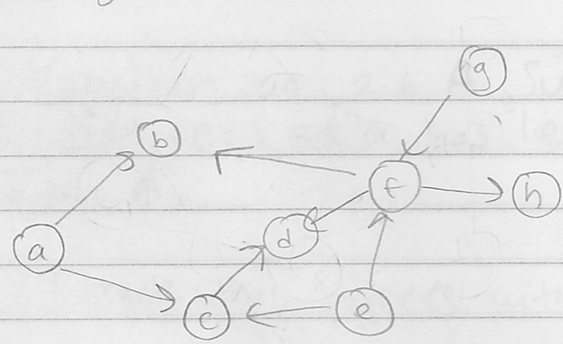
Then $y \preceq z \Rightarrow y \preceq_T z$

Check by cases: case #1: $y, z \in A \setminus \{x\}$

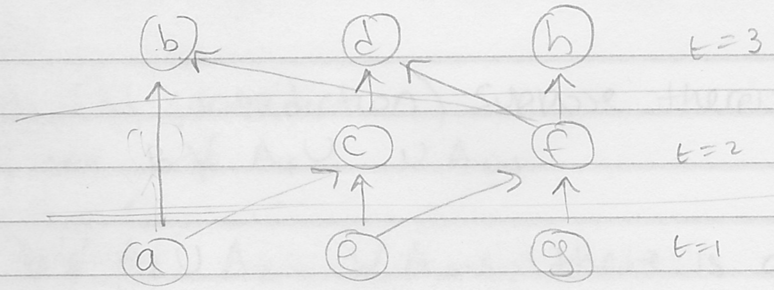
case #2: $y = x$, $z \in A \setminus \{x\}$ $\boxtimes$

Now we know how to get dressed in the morning!

What if you can do some tasks in parallel?



If you have many parallel processors, how many steps do you need?



Why can't we shedule it better? Because there is a length three chain

def: Given a poset $(A, \preceq)$, a valid parallel schedule of height k is a partition

$$A = A_1 \dot\cup A_2 \dot\cup \dots \dot\cup A_k$$

and $\forall i \in \{1, 2 \dots k\}$, $\forall a \in A_i$, $\forall b \preceq a$ then $b \in A_1 \cup A_2 \dots \cup A_{i-1}$

Fundamental theorem of parallel scheduling:

Theorem: Given a poset $(A, \preceq)$ with longest chain k, there is a valid parallel schedule of height k

Proof: Consider an $a \in A$. Suppose longest chain that ends at a has length i. then put a in A_i.

(check this agrees with scheduling above)

Why is this height at most k? Because the longest chain has length k

claim: $A_1, A_2 \dots A_k$ is a valid parallel schedule

Proof: (By contradiction) suppose there is $a \in A_i$, $b \preceq a$ and $b \notin A_1 \cup \dots \cup A_{i-1}$.

$b \to a$

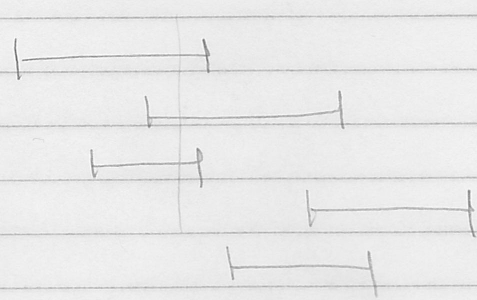
Since $b \notin A_1 \cup A_2 \dots \cup A_{i-1}$, there is a chain of length i that ends at b

$$a_1 \preceq a_2 \preceq \dots \preceq a_{i-1} \preceq b \ (\preceq a)$$

But this is a chain of length $i+1$ that ends at $a \Rightarrow a \notin A_i \Rightarrow\Leftarrow$ $\boxed{}$

Many other examples relating posets and scheduling

Example: Hotel reservations



Intervals I_1, I_2 have $I_1 \preceq I_2$ iff I_1 ends before I_2 starts

def: An antichain of size k is a collection $a_1, a_2 \dots a_k$ where $a_i \not\preceq a_j$ $\forall i \neq j$ (no pair is comparable

How many rooms do you need? The size of the largest anti-chain, and no switching needed.