Solutions to Problem Set 3

Problem 1. **(a)** List all the different binary relations on the set $\{0, 1\}$.

Solution. There are altogether 16 binary relations.

1. $\emptyset$
2. $\{(0, 0)\}$
3. $\{(0, 1)\}$
4. $\{(1, 0)\}$
5. $\{(1, 1)\}$
6. $\{(0, 0), (0, 1)\}$
7. $\{(0, 0), (1, 0)\}$
8. $\{(0, 0), (1, 1)\}$
9. $\{(0, 1), (1, 0)\}$
10. $\{(0, 1), (1, 1)\}$
11. $\{(1, 0), (1, 1)\}$
12. $\{(0, 0), (0, 1), (1, 0)\}$
13. $\{(0, 0), (0, 1), (1, 1)\}$
14. $\{(0, 0), (1, 0), (1, 1)\}$
15. $\{(0, 1), (1, 0), (1, 1)\}$
16. $\{(0, 0), (0, 1), (1, 0), (1, 1)\}$

■

(b) Over the domain $\{0, 1\}$, which of these relations are weak partial orders? strict partial orders? equivalence relations?

Solution. We first list the relations that satisfy each of the following properties:

- reflexive: $8, 13, 14, 16$
- symmetric: $1, 2, 5, 8, 9, 12, 15, 16$
- antisymmetric: $1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 13, 14$
- transitive: $1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 13, 14, 16$

From this we can see that the weak partial orders are $\{8, 13, 14\}$, the strict partial orders are $\{1, 3, 4\}$, and the equivalence relations are $\{8, 16\}$. ■

Problem 2. We partially order the power set, $\mathcal{P}(\{1, 2, \ldots, n\})$, by the subset relation, $\subseteq$.

(a) Describe a maximum length chain in $\mathcal{P}(\{1, 2, \ldots, n\})$. Briefly explain why there can't be a longer chain than the one you described.

Solution. The length $n + 1$ chain

$$\emptyset, \{1\}, \{1, 2\}, \{1, 2, 3\}, \ldots, \{1, 2, \ldots, n\}$$

is a maximum length chain. There can't be a longer one: any longer chain would have to contain two subsets of the same size (think about why!), and no finite set is contained in any other set of the same size. ■

(b) Describe a topological sort of $\mathcal{P}(\{1, 2, \ldots, n\})$, with a brief justification that your sort is correct.

Solution. All sets of size 0, followed by all sets of size 1 (in any order), followed by all sets of size 2 (again, in any order), $\ldots$, followed by all sets of size n. See Figure 1 for the case of $n = 3$. $\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}$ is a topological sort of this relation.

Note that not all chains and antichains are labelled. A chain is any group that is connected in some manner with arrows. Antichains are groups of elements where no two members are connected. ■

(c) Use Dilworth's Lemma to show that there must be an antichain of size $\geq 2^n/(n + 1)$. Describe the biggest antichain that you can find.

Solution. Since a maximum length chain is of size at most $n + 1$, and the powerset has 2^n elements, Dilworth's Lemma tells us that there must be an antichain of size at least $2^n/(n + 1)$.

A maximum length antichain is the set of all subsets containing exactly $\lceil n/2 \rceil$ elements; the only other one is the set of all subsets containing exactly $\lfloor n/2 \rfloor$ elements (of course these are the same if n is even). A proof of this is actually tricky. The size of this antichain is about $2^n/\sqrt{2\pi n}$. We can't present proofs of either of these facts yet, because they depend on concepts that won't be introduced for another month. ■

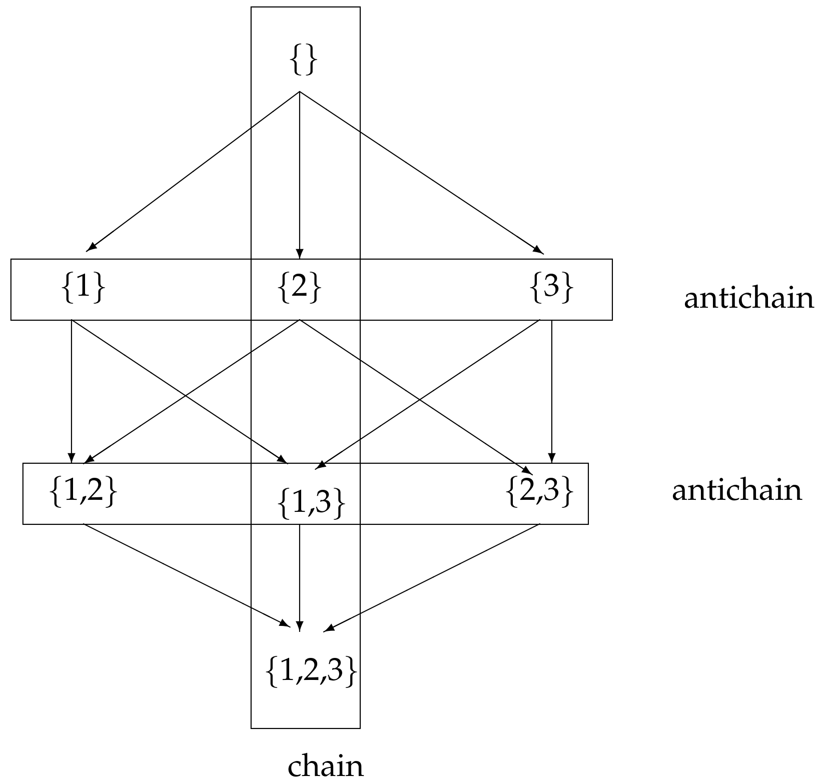


Figure 1: DAG for Problem 4(c) with $n = 3$.

Problem 3. Consider the natural numbers partially ordered by divisibility.

(a) Prove that this partial order has an infinite chain.

Solution. 1 2 4 8 16 . . . is a chain with infinite length. ∎

(b) Prove that this partial order has an infinite antichain.

Solution. The set of prime numbers is infinite. Since no prime divides another, any two primes are incomparable. So the set of prime numbers is an antichain. ∎

(c) Now restrict the domain to the natural numbers $\leq n$. Consider the chain $1 \preceq_R 2 \preceq_R 4 \preceq_R \dots \preceq_R 2^{\lfloor \log_2 n \rfloor}$. Prove that it is maximal.

Solution. Suppose there is a longer chain $a_0 \preceq_R a_1 \preceq_R a_2 \preceq_R \dots \preceq_R a_m$. Since this chain is longer, $m \geq \lfloor \log_2 n \rfloor + 1$. For $i \in \{1, 2, 3, \dots, m\}$, let $a_i = p_i a_{i-1}$, where p_i is an integer greater than 1. Then $a_m = p_1 p_2 \dots p_m a_0$. Since each $p_i \geq 2$ and $a_0 \geq 1$, we have

$$\begin{aligned} a_m &= p_1 p_2 \ldots p_m a_0 \\ &\geq 2^m a_0 \\ &\geq 2^m \\ &\geq 2^{\lfloor \log_2 n \rfloor + 1} \\ &> 2^{\log_2 n} \\ &= n, \end{aligned}$$

which is a contradiction since $a_m \leq n$ as $a_m \in A$. ■

(d) Let c be the length of the power of 2 chain. By Dilworth's Lemma there is an antichain of length n/c. Describe one.

Solution. The set $\{\lfloor \frac{n}{2} \rfloor + 1, \lfloor \frac{n}{2} \rfloor + 2, \ldots, n\}$ is an antichain with size $\lceil \frac{n}{2} \rceil$, which is no less than the lower bound. ■

Problem 4. We consider DAG's where each vertex represents a task to be completed. If there is a path from one vertex, v, to another vertex, w, then the v task must be completed before the w task. Assuming all tasks take unit time to complete, we showed in the Notes that the minimum time schedule to complete all the tasks is the size (number of vertices), t, of the longest path (chain) in the DAG.

Formally, a *schedule* for a DAG is a partition of the vertices. Each block of the partition is supposed to correspond to a set of tasks that are to be performed simultaneously. The *number of processors required* by a schedule is the maximum number of tasks that are scheduled to be performed simultaneously.

(a) Describe purely in terms of graph, partition, and partial order properties (no informal descriptions in terms of "jobs," "parallel processing," etc.):

- exactly the properties a vertex partition of a DAG must satisfy in order to represent a possible schedule for the vertex tasks,
- the total time required to complete a schedule,
- the number of processors required by a schedule.

Solution.

- A schedule for a DAG, G, is a partition of the edges of G into a sequence of blocks, $B_1, B_2, \ldots, B_k$ such that if $a \in B_i$, $b \in B_j$, and $a < b$ (that is, there is a path of positive length from vertex a to vertex b), then $i < j$. Another way to say this is that the blocks are anti-chains, and the sequence consisting of the elements in B_1 in any order, followed by the elements of B_2 in any order, through the elements of B_k, is a topological sort of the partial order defined by G.
- The total time required to complete a schedule is the number, k, of blocks it has.
- The number of processors required by a schedule is the size of the largest block.

■

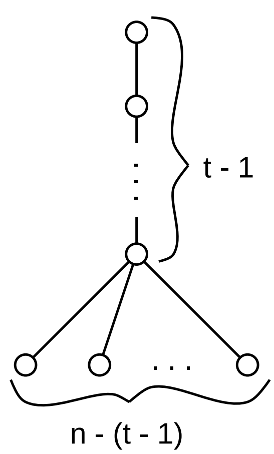
(b) Give a small example of a DAG with more than one minimum time schedule.

Solution. $V = \{1, 2, 3\}, E = \{1 \longrightarrow 2\}$. There are two minimum time schedules: $\{\{1, 3\}\{2\}\}$ and $\{\{1\}\{2, 3\}\}$. ■

(c) Explain why any schedule that requires only p processors to complete n tasks must take time at least $\lceil n/p \rceil$.

Solution. If there are $k < \lceil n/p \rceil$, then the integer k is less than n/p. So if there are k blocks and each block has at most p vertices, the total number of vertices is $\leq kp < (n/p) \cdot p = n$, a contradiction. ■

(d) Let $D_{n,t}$ be the DAG with n vertices that consists of a directed path of $t - 1$ vertices ending with edges from the final, $(t - 1)$st, vertex on the path directly to each of the remaining $n - (t - 1)$ vertices, as in the following figure:



What is the minimum time schedule for $D_{n,t}$? Explain why it is unique. How many processors does it require?

Solution. There's no choice but to schedule each of the $t-1$ vertices on the path one at a time in order. A minimum time schedule then does all the remaining $n-(t-1)$ vertices at the tth time interval. The number of processors required is therefore $n-t+1$. The time is t, the number of vertices on the longest chain in the graph. ■

(e) Describe a minimum time p-processor schedule for $D_{n,t}$. Write a simple formula for this minimum time, $M(n,t,p)$.

Solution. As in part (**??**), there's no choice but to schedule each of the $t-1$ vertices on the path one at a time in order. A minimum time schedule then does all the remaining $n-(t-1)$ vertices p at a time, for a total time of

$$M(n,t,p) ::= (t-1) + \left\lceil \frac{n-(t-1)}{p} \right\rceil . \tag{1}$$

■

(f) Show that *every* DAG with n vertices and maximum chain size, t, has a p-processor schedule that runs in time $M(n,t,p)$.

Hint: Induction – you decide on what variable. You may find it helpful to use the fact that if $a \geq b \geq 0$, then

$$\lceil a-b \rceil \leq 1 + \lceil a \rceil - \lceil b \rceil \tag{2}$$

for all real numbers a, b.

Solution. *Proof.* Induction on t. Induction hypothesis:

> $P(t) ::= \forall$ DAGs $G, \forall n, p \in \mathbb{N}^+$, if G has n vertices and maximum chain size t, then there is a p-processor schedule for G that takes time $M(n,t,p)$.

Base case $t=1$: In this case there are n vertices and no edges between them. So any partition of the vertices into $\lceil n/p \rceil$ blocks of size at most p will be a p-processor schedule taking time $\lceil n/p \rceil = 0 + \lceil (n-0)/p \rceil = M(n,1,p)$.

Inductive step: Assume $P(t)$ and conclude $P(t+1)$ where $t \geq 1$.

Let G be any DAG with n vertices and maximum chain size $t+1$. Suppose k vertices are endpoints of maximum-size chains in G. Note that no edge can leave any of these endpoint vertices, for otherwise there would be a chain of length one more than the maximum chain size. Let H be the subgraph of G obtained by removing these k vertices.

Now H is a DAG with $n-k$ vertices and maximum chain size t, so by Induction Hypothesis, there is a p-processor schedule for H taking time $M(n-k,t,p)$.

This p-processor schedule for H can be extended to one for G by adding $\lceil k/p \rceil$ disjoint blocks of the endpoints, all of size $\leq p$. So the time for this schedule for G is

$$\begin{aligned}
&M(n-k,t,p) + \left\lceil \frac{k}{p} \right\rceil \\
&= (t-1) + \left\lceil \frac{n-k-(t-1)}{p} \right\rceil + \left\lceil \frac{k}{p} \right\rceil && \text{(def of } M\text{)} \\
&= (t-1) + \left\lceil \frac{n-t}{p} - \frac{k-1}{p} \right\rceil + \left\lceil \frac{k}{p} \right\rceil && (3)
\end{aligned}$$

We complete the proof by showing that the expression (**??**) is $\leq M(n,t+1,p)$. To do this, we consider two cases:

- **Case 1** ($k-1$ is not a multiple of p): We have

  $$\left\lceil \frac{k-1}{p} \right\rceil = \left\lceil \frac{k}{p} \right\rceil, \tag{4}$$

  so

  $$\begin{aligned}
  (\textbf{??}) &\leq (t-1) + \left(1 + \left\lceil \frac{n-t}{p} \right\rceil - \left\lceil \frac{k-1}{p} \right\rceil\right) + \left\lceil \frac{k}{p} \right\rceil && \text{(by (\textbf{??}))} \\
  &= (t-1) + \left(1 + \left\lceil \frac{n-t}{p} \right\rceil - \left\lceil \frac{k}{p} \right\rceil\right) + \left\lceil \frac{k}{p} \right\rceil && \text{(by (\textbf{??}))} \\
  &= t + \left\lceil \frac{n-t}{p} \right\rceil \\
  &= M(n,t+1,p). && \text{(def of } M\text{)}
  \end{aligned}$$

- **Case 2** ($k-1$ is a multiple of p): Now we have

  $$\left\lceil \frac{k}{p} \right\rceil = 1 + \frac{k-1}{p}, \tag{5}$$

  so

  $$\begin{aligned}
  (\textbf{??}) &= (t-1) + \left(\left\lceil \frac{n-t}{p} \right\rceil - \frac{k-1}{p}\right) + \left\lceil \frac{k}{p} \right\rceil && \text{(since } (k-1)/p \in \mathbb{Z}\text{)} \\
  &= (t-1) + \left\lceil \frac{n-t}{p} \right\rceil - \frac{k-1}{p} + \left(1 + \frac{k-1}{p}\right) && \text{(by (\textbf{??}))} \\
  &= t + \left\lceil \frac{n-t}{p} \right\rceil \\
  &= M(n,t+1,p). && \text{(def of } M\text{)}
  \end{aligned}$$

$\square$

$\blacksquare$