7 Relations and Partial Orders

A relation is a mathematical tool for describing associations between elements of sets. Relations are widely used in computer science, especially in databases and scheduling applications. A relation can be defined across many items in many sets, but in this text, we will focus on *binary* relations, which represent an association between two items in one or two sets.

7.1 Binary Relations

7.1.1 Definitions and Examples

Definition 7.1.1. Given sets A and B, a *binary relation* $R : A \to B$ *from*[1] A to B is a subset of $A \times B$. The sets A and B are called the *domain* and *codomain* of R, respectively. We commonly use the notation aRb or $a \sim_R b$ to denote that $(a, b) \in R$.

A relation is similar to a function. In fact, every function $f : A \to B$ is a relation. In general, the difference between a function and a relation is that a relation might associate multiple elements of B with a single element of A, whereas a function can only associate at most one element of B (namely, $f(a)$) with each element $a \in A$.

We have already encountered examples of relations in earlier chapters. For example, in Section 5.2, we talked about a relation between the set of men and the set of women where mRw if man m likes woman w. In Section 5.3, we talked about a relation on the set of MIT courses where $c_1 R c_2$ if the exams for c_1 and c_2 cannot be given at the same time. In Section 6.3, we talked about a relation on the set of switches in a network where $s_1 R s_2$ if s_1 and s_2 are directly connected by a wire that can send a packet from s_1 to s_2. We did not use the formal definition of a relation in any of these cases, but they are all examples of relations.

As another example, we can define an "in-charge-of" relation T from the set of MIT faculty F to the set of subjects in the 2010 MIT course catalog. This relation contains pairs of the form

$$(\langle\text{instructor-name}\rangle, \langle\text{subject-num}\rangle)$$

[1] We also say that the relationship is *between* A and B, or *on* A if $B = A$.

(Meyer,	6.042),
(Meyer,	18.062),
(Meyer,	6.844),
(Leighton,	6.042),
(Leighton,	18.062),
(Freeman,	6.011),
(Freeman,	6.881)
(Freeman,	6.882)
(Freeman,	6.UAT)
(Eng,	6.UAT)
(Guttag,	6.00)

Figure 7.1 Some items in the "in-charge-of" relation T between faculty and subject numbers.

where the faculty member named ⟨instructor-name⟩ is in charge of the subject with number ⟨subject-num⟩. So T contains pairs like those shown in Figure 7.1.

This is a surprisingly complicated relation: Meyer is in charge of subjects with three numbers. Leighton is also in charge of subjects with two of these three numbers—because the same subject, Mathematics for Computer Science, has two numbers (6.042 and 18.062) and Meyer and Leighton are jointly in-charge-of the subject. Freeman is in-charge-of even more subjects numbers (around 20), since as Department Education Officer, he is in charge of whole blocks of special subject numbers. Some subjects, like 6.844 and 6.00 have only one person in-charge. Some faculty, like Guttag, are in-charge-of only one subject number, and no one else is jointly in-charge-of his subject, 6.00.

Some subjects in the codomain, N, do not appear in the list—that is, they are not an element of any of the pairs in the graph of T; these are the Fall term only subjects. Similarly, there are faculty in the domain, F, who do not appear in the list because all their in-charge-of subjects are Fall term only.

7.1.2 Representation as a Bipartite Graph

Every relation $R : A \to B$ can be easily represented as a bipartite graph $G = (V, E)$ by creating a "left" node for each element of A and a "right" node for each element of B. We then create an edge between a left node u and a right node v whenever aRb. Similarly, every bipartite graph (and every partition of the nodes into a "left" and "right" set for which no edge connects a pair of left nodes or a pair of right nodes) determines a relation between the nodes on the left and the nodes on the right.

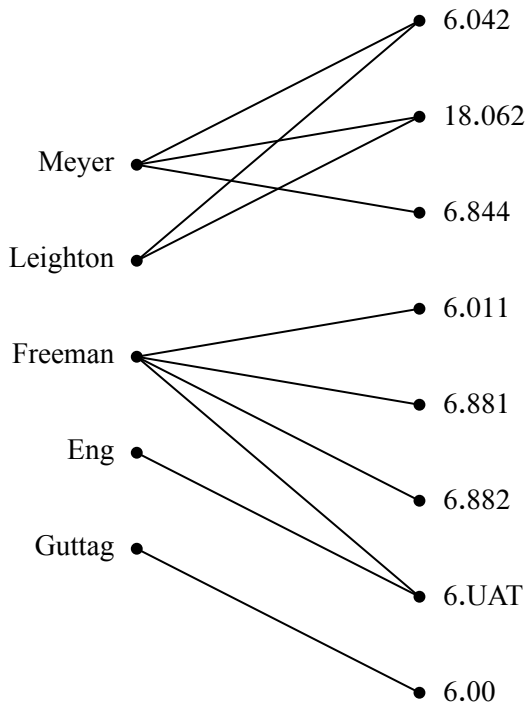

Figure 7.2 Part of the bipartite graph for the "in charge of" relation T from Figure 7.1.

For example, we have shown part of the bipartite graph for the in-charge-of relation from Figure 7.1 in Figure 7.2. In this case, there is an edge between ⟨instructor-name⟩ and ⟨subject-number⟩ if ⟨instructor-name⟩ is in charge of ⟨subject-number⟩.

A relation $R : A \to B$ between finite sets can also be represented as a matrix $A = \{a_{ij}\}$ where

$$a_{ij} = \begin{cases} 1 & \text{if the } i\text{th element of } A \text{ is related to the } j\text{th element of } B \\ 0 & \text{otherwise} \end{cases}$$

for $1 \leq i \leq |A|$ and $1 \leq j \leq |B|$. For example, the matrix for the relation in Figure 7.2 (but restricted to the five faculty and eight subject numbers shown in Figure 7.2, ordering them as they appear top-to-bottom in Figure 7.2) is shown in Figure 7.3.

7.1.3 Relational Images

The idea of the image of a set under a function extends directly to relations.

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Figure 7.3 The matrix for the "in charge of" relation T restricted to the five faculty and eight subject numbers shown in Figure 7.2. The $(3, 4)$ entry of this matrix is 1 since the third professor (Freeman) is in charge of the fourth subject number (6.011).

Definition 7.1.2. The *image* of a set Y under a relation $R : A \to B$, written $R(Y)$, is the set of elements that are related to some element in Y, namely,

$$R(Y) ::= \{ b \in B \mid yRb \text{ for some } y \in Y \}.$$

The image of the domain, $R(A)$, is called the *range* of R.

For example, to find the subject numbers that Meyer is in charge of, we can look for all the pairs of the form

$$(\text{Meyer}, \langle \text{subject-number} \rangle)$$

in the graph of the teaching relation T, and then just list the right-hand sides of these pairs. These right-hand sides are exactly the image $T(\text{Meyer})$, which happens to be $\{6.042, 18.062, 6.844\}$. Similarly, since the domain F is the set of all in-charge faculty, $T(F)$, the range of T, is exactly the set of *all* subjects being taught.

7.1.4 Inverse Relations and Images

Definition 7.1.3. The *inverse* R^{-1} of a relation $R : A \to B$ is the relation from B to A defined by the rule

$$bR^{-1}a \text{ if and only if } aRb.$$

The image of a set under the relation R^{-1} is called the *inverse image* of the set. That is, the inverse image of a set X under the relation R is $R^{-1}(X)$.

Continuing with the in-charge-of example above, we can find the faculty in charge of 6.UAT by taking the pairs of the form

$$(\langle \text{instructor-name} \rangle, 6.\text{UAT})$$

for the teaching relation T, and then just listing the left-hand sides of these pairs; these turn out to be just Eng and Freeman. These left-hand sides are exactly the inverse image of {6.UAT} under T.

7.1.5 Combining Relations

There are at least two natural ways to combine relations to form new relations. For example, given relations $R : B \to C$ and $S : A \to B$, the *composition* of R with S is the relation $(R \circ S) : A \to C$ defined by the rule

$$a(R \circ S)c \quad \text{IFF} \quad \exists b \in B.\ (bRc) \text{ AND } (aSb)$$

where $a \in A$ and $c \in C$.

As a special case, the composition of two functions $f : B \to C$ and $g : A \to B$ is the function $f \circ g : A \to C$ defined by

$$(f \circ g)(a) = f(g(a))$$

for all $a \in A$. For example, if $A = B = C = \mathbb{R}$, $g(x) = x + 1$ and $f(x) = x^2$, then

$$\begin{aligned}(f \circ g)(x) &= (x+1)^2 \\ &= x^2 + 2x + 1.\end{aligned}$$

One can also define the *product* of two relations $R_1 : A_1 \to B_1$ and $R_2 : A_2 \to B_2$ to be the relation $S = R_1 \times R_2$ where

$$S : A_1 \times A_2 \to B_1 \times B_2$$

and

$$(a_1, a_2)S(b_1, b_2) \quad \text{iff} \quad a_1 R_1 b_1 \text{ and } a_2 R_2 b_2.$$

7.2 Relations and Cardinality

7.2.1 Surjective and Injective Relations

There are some properties of relations that will be useful when we take up the topic of counting in Part III because they imply certain relations between the *sizes* of domains and codomains. In particular, we say that a binary relation $R : A \to B$ is

- *surjective* if every element of B is assigned to at least one element of A. More concisely, R is surjective iff $R(A) = B$ (that is, if the range of R is the codomain of R),

- *total* when every element of A is assigned to some element of B. More concisely, R is total iff $A = R^{-1}(B)$,
- *injective* if every element of B is mapped *at most once*, and
- *bijective* if R is total, surjective, injective, and a function[2].

We can illustrate these properties of a relation $R : A \to B$ in terms of the corresponding bipartite graph G for the relation, where nodes on the left side of G correspond to elements of A and nodes on the right side of G correspond to elements of B. For example:

- "R is a function" means that every node on the left is incident to *at most* one edge.
- "R is total" means that *every* node on the left is incident to *at least* one edge. So if R is a function, being total means that every node on the left is incident to exactly one edge.
- "R is surjective" means that *every* node on the right is incident to *at least* one edge.
- "R is injective" means that *every* node on the right is incident to *at most* one edge.
- "R is bijective" means that every node on both sides is incident to *precisely* one edge (that is, there is a perfect matching between A and B).

For example, consider the relations R_1 and R_2 shown in Figure 7.4. R_1 is a total surjective function (every node in the left column is incident to exactly one edge, and every node in the right column is incident to at least one edge), but not injective (node 3 is incident to 2 edges). R_2 is a total injective function (every node in the left column is incident to exactly one edge, and every node in the right column is incident to at most one edge), but not surjective (node 4 is not incident to any edges).

Notice that we need to know what the domain is to determine whether a relation is total, and we need to know the codomain to determine whether it's surjective. For example, the function defined by the formula $1/x^2$ is total if its domain is $\mathbb{R}^+$ but partial if its domain is some set of real numbers that includes 0. It is bijective if its domain and codomain are both $\mathbb{R}^+$, but neither injective nor surjective it is domain and codomain are both $\mathbb{R}$.

[2]These words *surjective*, *injective*, and *bijective* are not very memorable. Some authors use the possibly more memorable phrases *onto* for surjective, *one-to-one* for injective, and *exact correspondence* for bijective.

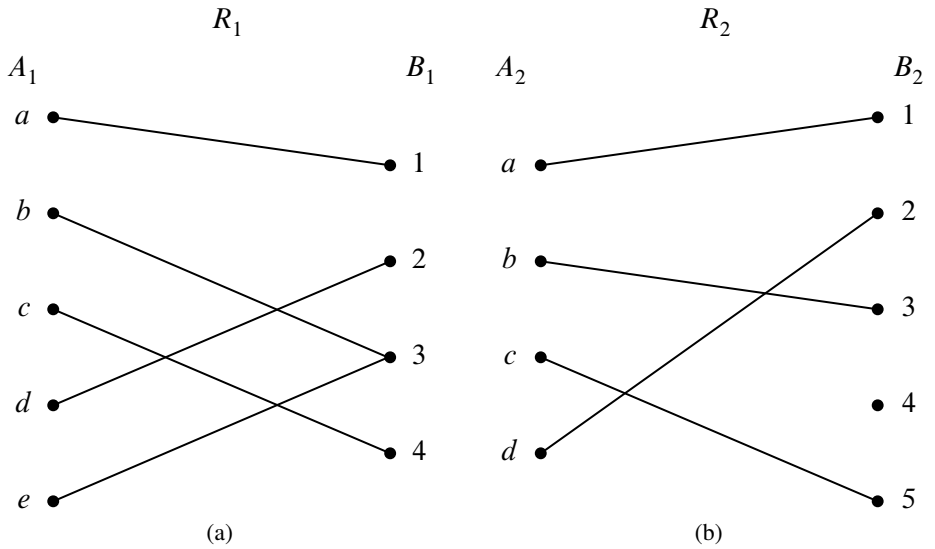

Figure 7.4 Relation $R_1 : A_1 \to B_1$ is shown in (a) and relation $R_2 : A_2 \to B_2$ is shown in (b).

7.2.2 Cardinality

The relational properties in Section 7.2.1 are useful in figuring out the relative sizes of domains and codomains.

If A is a finite set, we use $|A|$ to denote the number of elements in A. This is called the *cardinality* of A. In general, a finite set may have no elements (the empty set), or one element, or two elements, ..., or any nonnegative integer number of elements, so for any finite set, $|A| \in \mathbb{N}$.

Now suppose $R : A \to B$ is a function. Then every edge in the bipartite graph $G = (V, E)$ for R is incident to exactly one element of A, so the number of edges is at most the number of elements of A. That is, if R is a function, then

$$|E| \leq |A|.$$

Similarly, if R is surjective, then every element of B is incident to an edge, so there must be at least as many edges in the graph as the size of B. That is

$$|E| \geq |B|.$$

Combining these inequalities implies that $R : A \to B$ is a surjective function, then $|A| \geq |B|$. This fact and two similar rules relating domain and codomain size to relational properties are captured in the following theorem.

Theorem 7.2.1 (Mapping Rules). *Let A and B be finite sets.*

1. *If there is a surjection from A to B, then $|A| \geq |B|$.*
2. *If there is an injection from A to B, then $|A| \leq |B|$.*
3. *If there is a bijection between A and B, then $|A| = |B|$.*

Mapping rule 2 can be explained by the same kind of reasoning we used for rule 1. Rule 3 is an immediate consequence of the first two mapping rules.

We will see many examples where Theorem 7.2.1 is used to determine the cardinality of a finite set. Later, in Chapter 13, we will consider the case when the sets are infinite and we'll use surjective and injective relations to prove that some infinite sets are "bigger" than other infinite sets.

7.3 Relations on One Set

For the rest of this chapter, we are going to focus on relationships between elements of a single set; that is, relations from a set A to a set B where $A = B$. Thus, a relation on a set A is a subset $R \subseteq A \times A$. Here are some examples:

- Let A be a set of people and the relation R describe who likes whom: that is, $(x, y) \in R$ if and only if x likes y.
- Let A be a set of cities. Then we can define a relation R such that xRy if and only if there is a nonstop flight from city x to city y.
- Let $A = \mathbb{Z}$ and let xRy hold if and only if $x \equiv y \pmod 5$.
- Let $A = \mathbb{N}$ and let xRy if and only if $x \mid y$.
- Let $A = \mathbb{N}$ and let xRy if and only if $x \leq y$.

The last examples clarify the reason for using xRy or $x \sim_R y$ to indicate that the relation R holds between x and y: many common relations ($<$, $\leq$, $=$, $|$, $\equiv$) are expressed with the relational symbol in the middle.

7.3.1 Representation as a Digraph

Every relation on a single set A can be modeled as a directed graph (albeit one that may contain loops). For example, the graph in Figure 7.5 describes the "likes" relation for a particular set of 3 people.

In this case, we see that:

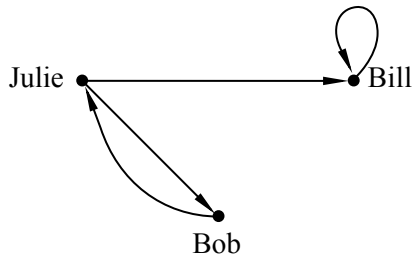

Figure 7.5 The directed graph for the "likes" relation on the set {Bill, Bob, Julie}.

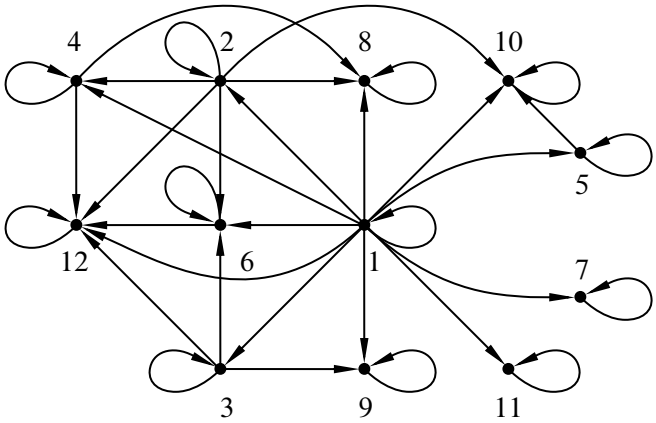

Figure 7.6 The digraph for divisibility on $\{1, 2, \ldots, 12\}$.

- Julie likes Bill and Bob, but not herself.
- Bill likes only himself.
- Bob likes Julie, but not Bill nor himself.

Everything about the relationship is conveyed by the directed graph and nothing more. This is no coincidence; a set A together with a relation R is precisely the same thing as directed graph $G = (V, E)$ with vertex set $V = A$ and edge set $E = R$ (where E may have loops).

As another example, we have illustrated the directed graph for the divisibility relationship on the set $\{1, 2, \ldots, 12\}$ in Figure 7.6. In this graph, every node has a loop (since every positive number divides itself) and the composite numbers are the nodes with indegree more than 1 (not counting the loop).

Relations on a single set can also be represented as a 0, 1-matrix. In this case, the matrix is identical to the adjacency matrix for the corresponding digraph. For

example, the matrix for the relation shown in Figure 7.5 is simply

$$\begin{pmatrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

where $v_1 =$ Julie, $v_2 =$ Bill, and $v_3 =$ Bob.

7.3.2 Symmetry, Transitivity, and Other Special Properties

Many relations on a single set that arise in practice possess one or more noteworthy properties. These properties are summarized in the box on the following page. In each case, we provide the formal of the definition of the property, explain what the property looks like in a digraph G for the relation, and give an example of what the property means for the "likes" relation.

For example, the congruence relation modulo 5 on $\mathbb{Z}$ is reflexive symmetric, and transitive, but not irreflexive, antisymmetric, or asymmetric. The same is true for the "connected" relation $R : V \to V$ on an undirected graph $G = (V, E)$ where uRv if u and v are in the same connected component of graph G. In fact, relations that have these three properties are so common that we give them a special name: *equivalence relations*. We will discuss them in greater detail in just a moment.

As another example, the "divides" relation on $\mathbb{Z}^+$ is reflexive, antisymmetric, and transitive, but not irreflexive, symmetric, or asymmetric. The same is true for the "$\leq$" relation on $\mathbb{R}$. Relations that have these three properties are also very common and they fall into a special case of relations called a *partial order*. We will discuss partial orders at length in Sections 7.5–7.9.

As a final example, consider the "likes" relation on the set {Julie, Bill, Bob} illustrated in Figure 7.5. This relation has *none* of the six properties described in the box.

7.4 Equivalence Relations

A relation is an *equivalence relation* if it is reflexive, symmetric, and transitive. Congruence modulo n is an excellent example of an equivalence relation:

- It is reflexive because $x \equiv x \pmod{n}$.
- It is symmetric because $x \equiv y \pmod{n}$ implies $y \equiv x \pmod{n}$.
- It is transitive because $x \equiv y \pmod{n}$ and $y \equiv z \pmod{n}$ imply that $x \equiv z \pmod{n}$.

Properties of a Relation $R : A \to A$

Reflexivity R is *reflexive* if

$\forall x \in A.\ xRX.$

"Everyone likes themselves."

Every node in G has a loop.

Irreflexivity R is *irreflexive* if

$\neg\exists x \in A.\ xRx.$

"No one likes themselves."

There are no loops in G.

Symmetry R is *symmetric* if

$\forall x, y \in A.\ xRy$ IMPLIES $yRx.$

"If x likes y, then y likes x."

If there is an edge from x to y in G, then there is an edge from y to x in G as well.

Antisymmetry R is *antisymmetric* if

$\forall x, y \in A\ (xRy$ AND $yRx)$ IMPLIES $x = y.$

"No pair of distinct people like each other."

There is at most one directed edge between any pair of distinct nodes.

Asymmetry R is *asymmetric* if

$\neg\exists x, y \in A.\ xRy$ AND $yRx.$

"No one likes themselves and no pair of people like each other."

There are no loops and there is at most one directed edge between any pair of nodes.

Transitivity R is *transitive* if

$\forall x, y, z \in A.\ (xRy$ AND $yRz)$ IMPLIES $xRz.$

"If x likes y and y likes z, then x likes z too."

For any walk $v_0, v_1, \ldots, v_k$ in G where $k \geq 2,\ v_0 \to v_k$ is in G (and, hence, $v_i \to v_j$ is also in G for all $i < j$.

There is an even more well-known example of an equivalence relation: equality itself. Thus, an equivalence relation is a relation that shares some key properties with "=".

7.4.1 Partitions

There is another way to think about equivalence relations, but we'll need a couple of definitions to understand this alternative perspective.

Definition 7.4.1. Given an equivalence relation $R : A \to A$, the *equivalence class* of an element $x \in A$ is the set of all elements of A related to x by R. The equivalence class of x is denoted $[x]$. Thus, in symbols:

$$[x] = \{ y \mid xRy \}.$$

For example, suppose that $A = \mathbb{Z}$ and xRy means that $x \equiv y \pmod 5$. Then

$$[7] = \{\ldots, -3, 2, 7, 12, 22, \ldots\}.$$

Notice that 7, 12, 17, etc., all have the same equivalence class; that is, $[7] = [12] = [17] = \cdots$.

Definition 7.4.2. A *partition* of a finite set A is a collection of disjoint, nonempty subsets $A_1, A_2, \ldots, A_n$ whose union is all of A. The subsets are usually called the *blocks* of the partition.[3] For example, one possible partition of $A = \{a, b, c, d, e\}$ is

$$A_1 = \{a, c\} \qquad A_2 = \{b, e\} \qquad A_3 = \{d\}.$$

Here's the connection between all this stuff: there is an exact correspondence between *equivalence relations on* A and *partitions of* A. We can state this as a theorem:

Theorem 7.4.3. *The equivalence classes of an equivalence relation on a set A form a partition of A.*

We won't prove this theorem (too dull even for us!), but let's look at an example.

[3] We think they should be called the *parts* of the partition. Don't you think that makes a lot more sense?

The congruent-mod-5 relation partitions the integers into five equivalence classes:

$$\{\ldots, -5, 0, 5, 10, 15, 20, \ldots\}$$
$$\{\ldots, -4, 1, 6, 11, 16, 21, \ldots\}$$
$$\{\ldots, -3, 2, 7, 12, 17, 22, \ldots\}$$
$$\{\ldots, -2, 3, 8, 13, 18, 23, \ldots\}$$
$$\{\ldots, -1, 4, 9, 14, 19, 24, \ldots\}$$

In these terms, $x \equiv y \pmod 5$ is equivalent to the assertion that x and y are both in the same block of this partition. For example, $6 \equiv 16 \pmod 5$, because they're both in the second block, but $2 \not\equiv 9 \pmod 5$ because 2 is in the third block while 9 is in the last block.

In social terms, if "likes" were an equivalence relation, then everyone would be partitioned into cliques of friends who all like each other and no one else.

7.5 Partial Orders

7.5.1 Strong and Weak Partial Orders

Definition 7.5.1. A relation R on a set A is a *weak partial order* if it is transitive, antisymmetric, and reflexive. The relation is said to be a *strong partial order* if it is transitive, antisymmetric, and irreflexive.[4]

Some authors defined partial orders to be what we call weak partial orders, but we'll use the phrase *partial order* to mean either a weak or a strong partial order. The difference between a weak partial order and a strong one has to do with the reflexivity property: in a weak partial order, *every* element is related to itself, but in a strong partial order, *no* element is related to itself. Otherwise, they are the same in that they are both transitive and antisymmetric.

Examples of weak partial orders include "$\leq$" on $\mathbb{R}$, "$\subseteq$" on the set of subsets of (say) $\mathbb{Z}$, and the "divides" relation on $\mathbb{N}^+$. Examples of strict partial orders include "$<$" on $\mathbb{R}$, and "$\subset$" on the set of subsets of $\mathbb{Z}$.[5]

[4] Equivalently, the relation is transitive and asymmetric, but stating it this way might have obscured the irreflexivity property.

[5] If you are not feeling comfortable with all the definitions that we've been throwing at you, it's probably a good idea to verify that each of these relations are indeed partial orders by checking that they have the transitivity and antisymmetry properties.

We often denote a weak partial order with a symbol such as $\preceq$ or $\sqsubseteq$ instead of a letter such as R. This makes sense from one perspective since the symbols call to mind $\leq$ and $\subseteq$, which define common partial orders. On the other hand, a partial order is really a set of related pairs of items, and so a letter like R would be more normal.

Likewise, we will often use a symbol like $\prec$ or $\sqsubset$ to denote a strong partial order.

7.5.2 Total Orders

A partial order is "partial" because there can be two elements with no relation between them. For example, in the "divides" partial order on $\{1, 2, \ldots, 12\}$, there is no relation between 3 and 5 (since neither divides the other).

In general, we say that two elements a and b are *incomparable* if neither $a \preceq b$ nor $b \preceq a$. Otherwise, if $a \preceq b$ or $b \preceq a$, then we say that a and b are *comparable*.

Definition 7.5.2. A *total order* is a partial order in which every pair of distinct elements is comparable.

For example, the "$\leq$" partial order on $\mathbb{R}$ is a total order because for any pair of real numbers x and y, either $x \leq y$ or $y \leq x$. The "divides" partial order on $\{1, 2, \ldots, 12\}$ is not a total order because $3 \nmid 5$ and $5 \nmid 3$.

7.6 Posets and DAGs

7.6.1 Partially Ordered Sets

Definition 7.6.1. Given a partial order $\preceq$ on a set A, the pair $(A, \preceq)$ is called a *partially ordered set* or *poset*.

In terms of graph theory, a poset is simply the directed graph $G = (A, \preceq)$ with vertex set A and edge set $\preceq$. For example, Figure 7.6 shows the graph form of the poset for the "divides" relation on $\{1, 2, \ldots, 12\}$. We have shown the graph form of the poset for the "$<$"-relation on $\{1, 2, 3, 4\}$ in Figure 7.7.

7.6.2 Posets Are Acyclic

Did you notice anything that is common to Figures 7.6 and 7.7? Of course, they both exhibit the transitivity and antisymmetry properties. And, except for the loops in Figure 7.6, they both do not contain any cycles. This is not a coincidence. In fact, the combination of the transitivity and asymmetry properties imply that the digraph

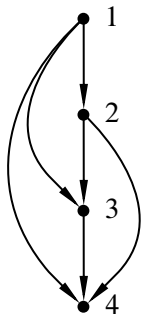

Figure 7.7 Representing the poset for the "<"-relation on $\{1, 2, 3, 4\}$ as a digraph.

for any poset is an acyclic graph (that is, a DAG), at least if you don't count loops as cycles. We prove this fact in the following theorem.

Theorem 7.6.2. *A poset has no directed cycles other than self-loops.*

Proof. We use proof by contradiction. Let $(A, \preceq)$ be a poset. Suppose that there exist $n \geq 2$ distinct elements $a_1, a_2, \ldots, a_n$ such that

$$a_1 \preceq a_2 \preceq a_3 \preceq \cdots \preceq a_{n-1} \preceq a_n \preceq a_1.$$

Since $a_1 \preceq a_2$ and $a_2 \preceq a_3$, transitivity implies $a_1 \preceq a_3$. Another application of transitivity shows that $a_1 \preceq a_4$ and a routine induction argument establishes that $a_1 \preceq a_n$. Since we know that $a_n \preceq a_1$, antisymmetry implies $a_1 = a_n$, contradicting the supposition that $a_1, \ldots, a_n$ are distinct and $n \geq 2$. Thus, there is no such directed cycle. ■

Thus, deleting the self-loops from a poset leaves a directed graph without cycles, which makes it a *directed acyclic graph* or *DAG*.

7.6.3 Transitive Closure

Theorem 7.6.2 tells us that every poset corresponds to a DAG. Is the reverse true? That is, does every DAG correspond to a poset? The answer is "Yes," but we need to modify the DAG to make sure that it satisfies the transitivity property. For example, consider the DAG shown in Figure 7.8. As any DAG must, this graph satisfies the antisymmetry property[6] but it does not satisfy the transitivity property because $v_1 \to v_2$ and $v_2 \to v_3$ are in the graph but $v_1 \to v_3$ is not in the graph.

[6] If $u \to v$ and $v \to u$ are in a digraph G, then G would have a cycle of length 2 and it could not be a DAG.

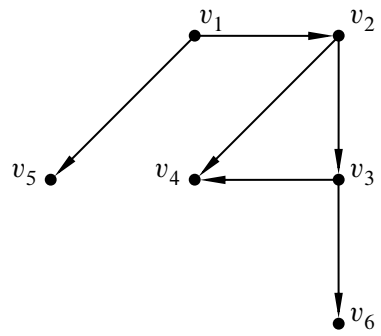

Figure 7.8 A 6-node digraph that does not satisfy the transitivity property.

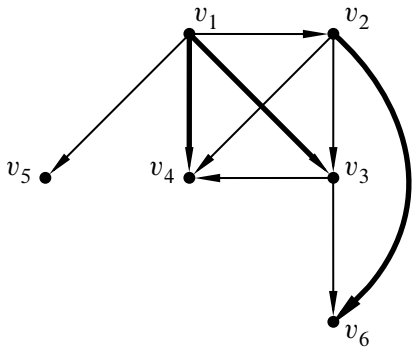

Figure 7.9 The transitive closure for the digraph in Figure 7.8. The edges that were added to form the transitive closure are shown in bold.

Definition 7.6.3. Given a digraph $G = (V, E)$, the *transitive closure* of G is the digraph $G^+ = (V, E^+)$ where

$$E^+ = \{\, u \to v \mid \text{there is a directed path of positive length from } u \text{ to } v \text{ in } G \,\}.$$

Similarly, if R is the relation corresponding to G, the *transitive closure* of R (denoted R^+) is the relation corresponding to G^+.

For example, the transitive closure for the graph in Figure 7.8 is shown in Figure 7.9.

If G is a DAG, then the transitive closure of G is a strong partial order. The proof of this fact is left as an exercise in the problem section.

7.6.4 The Hasse Diagram

One problem with viewing a poset as a digraph is that there tend to be lots of edges due to the transitivity property. Fortunately, we do not necessarily have to draw

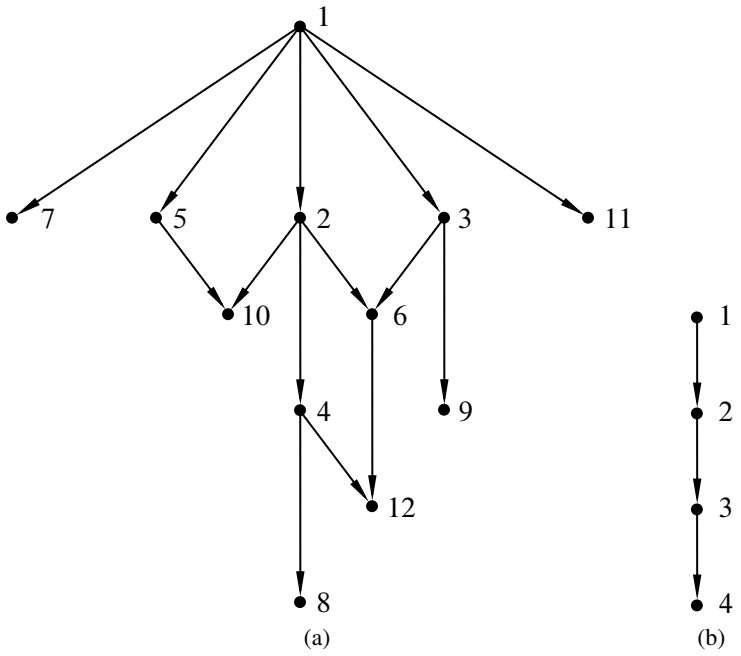

Figure 7.10 The Hasse diagrams for the posets in Figure 7.6 and 7.7.

all the edges if we know that the digraph corresponds to a poset. For example, we could choose not to draw any edge which would be implied by the transitivity property, knowing that it is really there by implication. In general, a *Hasse diagram* for a poset $(A, \preceq)$ is a digraph with vertex set A and edge set $\preceq$ minus all self-loops and edges implied by transitivity. For example, the Hasse diagrams of the posets shown in Figures 7.6 and 7.7 are shown in Figure 7.10.

7.7 Topological Sort

A total order that is consistent with a partial order is called a topological sort. More precisely,

Definition 7.7.1. A *topological sort* of a poset $(A, \preceq)$ is a total order $(A, \preceq_T)$ such that

$$x \preceq y \text{ IMPLIES } x \preceq_T y.$$

For example, consider the poset that describes how a guy might get dressed for a formal occasion. The Hasse diagram for such a poset is shown in Figure 7.11.

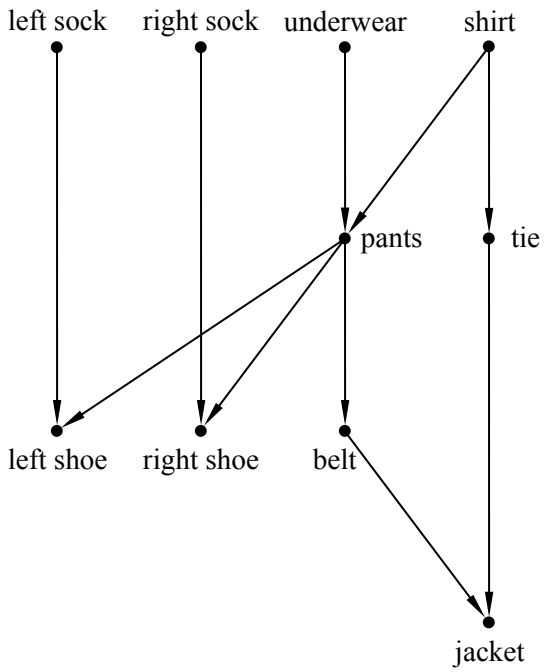

Figure 7.11 The Hasse diagram for a poset that describes which items much precede others when getting dressed.

In this poset, the *set* is all the garments and the *partial order* specifies which items much precede others when getting dressed.

There are several total orders that are consistent with the partial order shown in Figure 7.11. We have shown two of them in list form in Figure 7.12. Each such list is a topological sort for the partial order in Figure 7.11. In what follows, we will prove that every *finite* poset has a topological sort. You can think of this as a mathematical proof that you *can* get dressed in the morning (and then show up for math lecture).

Theorem 7.7.2. *Every finite poset has a topological sort.*

We'll prove the theorem constructively. The basic idea is to pull the "smallest" element a out of the poset, find a topological sort of the remainder recursively, and then add a back into the topological sort as an element smaller than all the others.

The first hurdle is that "smallest" is not such a simple concept in a set that is only partially ordered. In a poset $(A, \preceq)$, an element $x \in A$ is *minimal* if there is no other element $y \in A$ such that $y \preceq x$. For example, there are *four* minimal elements in the getting-dressed poset: left sock, right sock, underwear, and shirt. (It may seem

underwear	left sock
pants	shirt
belt	tie
shirt	underwear
tie	right sock
jacket	pants
left sock	right shoe
right sock	belt
left shoe	jacket
right shoe	left shoe
(a)	(b)

Figure 7.12 Two possible topological sorts of the poset shown in Figure 7.11. In each case, the elements are listed so that $x \preceq y$ iff x is above y in the list.

odd that the minimal elements are at the top of the Hasse diagram rather than the bottom. Some people adopt the opposite convention. If you're not sure whether minimal elements are on the top or bottom in a particular context, ask.) Similarly, an element $x \in A$ is *maximal* if there is no other element $y \in A$ such that $x \preceq y$.

Proving that every poset *has* a minimal element is extremely difficult, because it is not true. For example, the poset $(\mathbb{Z}, \leq)$ has no minimal element. However, there is at least one minimal element in every *finite* poset.

Lemma 7.7.3. *Every finite poset has a minimal element.*

Proof. Let $(A, \preceq)$ be an arbitrary poset. Let $a_1, a_2, \ldots, a_n$ be a maximum-length sequence of distinct elements in A such that

$$a_1 \preceq a_2 \preceq \cdots \preceq a_n.$$

The existence of such a maximum-length sequence follows from the Well Ordering Principle and the fact that A is finite. Now $a_0 \preceq a_1$ cannot hold for any element $a_0 \in A$ not in the chain, since the chain already has maximum length. And $a_i \preceq a_1$ cannot hold for any $i \geq 2$, since that would imply a cycle

$$a_i \preceq a_1 \preceq a_2 \preceq \cdots \preceq a_i$$

and no cycles exist in a poset by Theorem 7.6.2. Therefore a_1 is a minimal element. ■

Now we're ready to prove Theorem 7.7.2, which says that every finite poset has a topological sort. The proof is rather intricate; understanding the argument requires a clear grasp of all the mathematical machinery related to posets and relations!

Proof of Theorem 7.7.2. We use induction. Let $P(n)$ be the proposition that every n-element poset has a topological sort.

Base case: Every 1-element poset is already a total order and thus is its own topological sort. So $P(1)$ is true.

Inductive step: Now we assume $P(n)$ in order to prove $P(n+1)$ where $n \geq 1$. Let $(A, \preceq)$ be an $(n+1)$-element poset. By Lemma 7.7.3, there exists a minimal element in $a \in A$. Remove a and all pairs in $\preceq$ involving a to obtain an n-element poset $(A', \preceq')$. This has a topological sort $(A', \preceq'_T)$ by the assumption $P(n)$. Now we construct a total order $(A, \preceq_T)$ by adding a back as an element smaller than all the others. Formally, let

$$\preceq_T = \preceq'_T \cup \{\, (a, z) \mid z \in A \,\}.$$

All that remains is to check that this total order is consistent with the original partial order $(A, \preceq)$; that is, we must show that

$$x \preceq y \text{ IMPLIES } x \preceq_T y.$$

We assume that the left side is true and show that the right side follows. There are two cases.

Case 1 If $x = a$, then $a \preceq_T y$ holds, because $a \preceq_T z$ for all $z \in A$.

Case 2 if $x \neq a$, then y can not equal a either, since a is a minimal element in the partial order $\preceq$. Thus, both x and y are in A' and so $x \preceq' y$. This means $x \preceq'_T y$, since $\preceq'_T$ is a topological sort of the partial order $\preceq'$. And this implies $x \preceq_T y$ since $\preceq_T$ contains $\preceq'_T$.

Thus, $(A, \preceq_T)$ is a topological sort of $(A, \preceq)$. This shows that $P(n)$ implies $P(n+1)$ for all $n \geq 1$. Therefore $P(n)$ is true for all $n \geq 1$ by the principle of induction, which proves the theorem. ■

7.8 Parallel Task Scheduling

When items of a poset are tasks that need to be done and the partial order is a precedence constraint, topological sorting provides us with a way to execute the tasks sequentially without violating the precedence constraints.

But what if we have the ability to execute more than one task at the same time? For example, suppose that the tasks are programs, the partial order indicates data

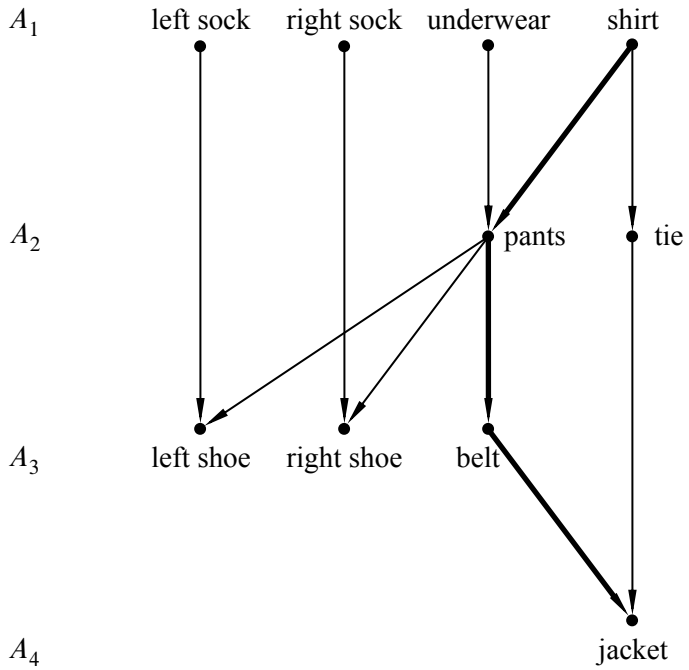

Figure 7.13 A parallel schedule for the tasks-in-getting-dressed poset in Figure 7.11. The tasks in A_i can be performed in step i for $1 \leq i \leq 4$. A chain of length 4 (the critical path in this example) is shown with bold edges.

dependence, and we have a parallel machine with lots of processors instead of a sequential machine with only one processor. How should we schedule the tasks so as to minimize the total time used?

For simplicity, assume all tasks take 1 unit of time and we have an unlimited number of identical processors. For example, in the clothes example in Figure 7.11, how long would it take to handle all the garments?

In the first unit of time, we should do all minimal items, so we would put on our left sock, our right sock, our underwear, and our shirt.[7] In the second unit of time, we should put on our pants and our tie. Note that we cannot put on our left or right shoe yet, since we have not yet put on our pants. In the third unit of time, we should put on our left shoe, our right shoe, and our belt. Finally, in the last unit of time, we can put on our jacket. This schedule is illustrated in Figure 7.13.

The total time to do these tasks is 4 units. We cannot do better than 4 units of

[7] Yes, we know that you can't actually put on both socks at once, but imagine you are being dressed by a bunch of robot processors and you are in a big hurry. Still not working for you? Ok, forget about the clothes and imagine they are programs with the precedence constraints shown in Figure 7.11.

time because there is a sequence of 4 tasks, each needing to be done before the next, of length 4. For example, we must put on our shirt before our pants, our pants before our belt, and our belt before our jacket. Such a sequence of items is known as a *chain*

Definition 7.8.1. A *chain* is a sequence $a_1 \preceq a_2 \preceq \cdots \preceq a_t$, where $a_i \neq a_j$ for all $i \neq j$, such that each item is comparable to the next in the chain, and it is smaller with respect to $\preceq$. The *length* of the chain is t, the number of elements in the chain.

Thus, the time it takes to schedule tasks, even with an unlimited number of processors, is at least the length of the longest chain. Indeed, if we used less time, then two items from a longest chain would have to be done at the same time, which contradicts the precedence constraints. For this reason, a longest chain is also known as a *critical path*. For example, Figure 7.13 shows the critical path for the getting-dressed poset.

In this example, we were in fact able to schedule all the tasks in t steps, where t is the length of the longest chain. The really nice thing about posets is that this is always possible! In other words, for any poset, there is a legal parallel schedule that runs in t steps, where t is the length of the longest chain.

There are lots of ways to prove this fact. Our proof will also give us the corresponding schedule in t time steps, and allow us to obtain some nice corollaries.

Theorem 7.8.2. *Given any finite poset $(A, \preceq)$ for which the longest chain has length t, it is possible to partition A into t subsets $A_1, A_2, \ldots, A_t$ such that for all $i \in \{1, 2, \ldots, t\}$ and for all $a \in A_i$, we have that all $b \preceq a$ appear in the set $A_1 \cup \ldots \cup A_{i-1}$.*

Before proving this theorem, first note that for each i, all items in A_i can be scheduled in time step i. This is because all preceding tasks are scheduled in preceding time steps, and thus are already completed. So the theorem implies that

Corollary 7.8.3. *The total amount of parallel time needed to complete the tasks is the same as the length of the longest chain.*

Proof of Theorem 7.8.2. For all $a \in A_i$, put a in A_i, where i is the length of the longest chain ending at a. For example, the A_i for the getting-dressed poset are shown in Figure 7.13. In what follows, we show that for all i, for all $a \in A_i$ and for all $b \preceq a$ with $b \neq a$, we have $b \in A_1 \cup A_2 \cup \ldots \cup A_{i-1}$.

We prove this by contradiction. Assume there is some i, $a \in A_i$, and $b \preceq a$ with $b \neq a$ and $b \notin A_1 \cup A_2 \cup \ldots \cup A_{i-1}$. By the way we defined A_i, this implies there is a chain of length at least i ending at b. Since $b \preceq a$ and $b \neq a$, we can extend this chain to a chain of length at least $i + 1$, ending at a. But then a could not be in A_i. This is a contradiction. ■

If we have an unlimited number of processors, then the time to complete all tasks is equal to the length of the longest chain of dependent tasks. The case where there are only a limited number of processors is very useful in practice and it is covered in the Problems section.

7.9 Dilworth's Lemma

Definition 7.9.1. An *antichain* in a poset is a set of elements such that any two elements in the set are incomparable.

For example, each A_i in the proof of Theorem 7.8.2 and in Figure 7.13 is an antichain since its elements have no dependencies between them (which is why they could be executed at the same time).

Our conclusions about scheduling also tell us something about antichains.

Corollary 7.9.2. *If the largest chain in a partial order on a set A is of size t, then A can be partitioned into t antichains.*

Proof. Let the antichains be the sets $A_1, A_2, \ldots, A_t$ defined in Theorem 7.8.2. ■

Corollary 7.9.2 implies a famous result[8] about partially ordered sets:

Lemma 7.9.3 (Dilworth)**.** *For all $t > 0$, every partially ordered set with n elements must have either a chain of size greater than t or an antichain of size at least n/t.*

Proof. By contradiction. Assume that the longest chain has length at most t and the longest antichain has size less than n/t. Then by Corollary 7.9.2, the n elements can be partitioned into at most t antichains. Hence, there are fewer than $t \cdot n/t = n$ elements in the poset, which is a contradiction. Hence there must be a chain longer than t or an antichain with at least n/t elements. ■

Corollary 7.9.4. *Every partially ordered set with n elements has a chain of size greater than $\sqrt{n}$ or an antichain of size at least $\sqrt{n}$.*

Proof. Set $t = \sqrt{n}$ in Lemma 7.9.3. ■

As an application, consider a permutation of the numbers from 1 to n arranged as a sequence from left to right on a line. Corollary 7.9.4 can be used to show that there must be a $\sqrt{n}$-length subsequence of these numbers that is completely

[8] Lemma 7.9.3 also follows from a more general result known as Dilworth's Theorem that we will not discuss.

increasing or completely decreasing as you move from left to right. For example, the sequence

$$7, 8, 9, 4, 5, 6, 1, 2, 3$$

has an increasing subsequence of length 3 (for example, 7, 8, 9) and a decreasing subsequence of length 3 (for example, 9, 6, 3). The proof of this result is left as an exercise that will test your ability to find the right partial order on the numbers in the sequence.

6.042J / 18.062J Mathematics for Computer Science
Fall 2010