Relations; Induction

1 Binary Relations

Relations are another fundamental Mathematical data type. Equality and "less than" are the most familiar examples of Mathematical relations. These are called *binary* relations because they relate two objects – are the objects equal? is the first less than the second? In this section of Notes we'll define some basic properties of binary relations and then focus on *partial orders*, which are a class of binary relations of particular importance in Computer Science, with direct applications that include task scheduling, database concurrency control, and proving that computations terminate.

1.1 Binary Relations and Functions

Binary relations are far more general than equality or less-than. Here's the official definition:

Definition 1.1. A *binary relation*, R, consists of a set, A, called the *domain* of R, a set, B, called the *codomain* of R, and a subset of $A \times B$ called the *graph* of R.

For example, we can define an "is teaching relation" for Spring '07 at MIT to have domain equal to the names of all the teaching staff (faculty, T.A.'s, *etc*.) and codomain equal to all the subject numbers in the current catalogue. Its graph would contain pairs like

$$\begin{array}{l}(\texttt{Albert R. Meyer}, 6.042),\\ (\texttt{Tina Nolte}, 18.062),\\ (\texttt{Chiyoun Park}, 6.042),\\ (\texttt{Albert R. Meyer}, 18.062),\\ (\texttt{Srini Devadas}, 6.046),\\ (\texttt{Donald Sadoway}, 3.091),\\ \quad\vdots\end{array}$$

Notice that Definition 1.1 is exactly the same as the definition of a *function*, except that it doesn't require the functional condition that, for each domain element, a, there is *at most* one pair in the graph whose first coordinate is a. So a function is a special case of a binary relation.

A relation whose domain is A and codomain is B is said to be "between A and B", or "from A to B." When the domain and codomain are the same set, A, we simply say the relation is "on A." It's common to use infix notation "$a\,R\,b$" to mean that the pair (a, b) is in the graph of R.

1.2 Images and Inverse Images

Before we go any further, it's worth introducing some notation that we'll get a lot of mileage out of. If R is a binary relation from A to B, and C is any set, define

$$\begin{aligned} CR &::= \{b \in B \mid cRb \text{ for some } c \in C\}, \\ RC &::= \{a \in A \mid aRc \text{ for some } c \in C\}. \end{aligned}$$

The set CR is called the *image* of C under R. With this notation, we could have defined the *range* of R simply as AR.

The set RC is called the *inverse image* of C under R. Notice the clash with pointwise application notation from Notes 2 when R happens to be a function: $\widehat{R}(C) = CR$, not RC. Sorry about that.

1.3 Surjective and like that

A relation with the property that every codomain element is related to some domain element is called a *surjective* (or *onto*) relation —again, the same definition as for functions. More concisely, a relation, R, between A and B is surjective iff $AR = B$. Likewise, a relation with the property that every domain element is related to some codomain element is called a *total* relation; more concisely, R is total iff $A = RB$.

2 Partial Orders

The prerequisite structure among MIT subjects provides a nice illustration of partial orders. Here is a table indicating some of the prerequisites of subjects in the Course 6 program:

Direct Prerequisites	Subject
18.01	6.042
18.01	18.02
18.01	18.03
8.01	8.02
6.001	6.034
6.042	6.046
18.03, 8.02	6.002
6.001, 6.002	6.004
6.001, 6.002	6.003
6.004	6.033
6.033	6.857
6.046	6.840

Since 18.01 is a direct prerequisite for 6.042, a student must take 18.01 before 6.042. Also, 6.042 is a direct prerequisite for 6.046, so in fact, a student has to take *both* 18.01 and 6.042 before taking 6.046. So 18.01 is also really a prerequisite for 6.046, though an implicit or indirect one; we'll indicate this by writing

$$18.01 \rightarrow 6.046.$$

This prerequisite relation has a basic property known as *transitivity*: if subject a is an indirect prerequisite of subject b, and b is an indirect prerequisite of subject c, then a is also an indirect prerequisite of c.

In this table, a longest sequence of prerequisites is

$$18.01 \to 18.03 \to 6.002 \to 6.004 \to 6.033 \to 6.857$$

so a student would need at least six terms to work through this sequence of courses. But it would take a lot longer to complete a Course 6 major if the direct prerequisites led to a situation[1] where two subjects turned out to be prerequisites of *each other*! So another crucial property of the prerequisite relation is that if $a \to b$, then it is not the case that $b \to a$. This property is called *asymmetry*.

Another basic example of a partial order is the subset relation, $\subseteq$, on sets. In fact, we'll see that every partial order can be represented by the subset relation.

2.1 Axioms for Partial Orders

Definition 2.1. A binary relation, R, on a set A is:

- *transitive* iff
$$[a\, R\, b \text{ and } b\, R\, c] \quad \text{implies} \quad a\, R\, c,$$
for every $a, b, c \in A$,
- *asymmetric* iff
$$a\, R\, b \quad \text{implies} \quad \neg(b\, R\, a)$$
for all $a, b \in A$,
- a *strict partial order* iff it is transitive and asymmetric.

So the prerequisite relation, $\to$, on subjects in the MIT catalogue is a strict partial order. More familiar examples of strict partial orders are the relation, $<$, on real numbers, and the proper subset relation, $\subset$, on sets.

The subset relation, $\subseteq$, on sets and $\leq$ relation on numbers are examples of *reflexive* relations in which each element is related to itself. Reflexive partial orders are called *weak* partial orders:

Definition 2.2. A binary relation, R, on a set A, is

- *reflexive* iff $a\, R\, a$ for all $a \in A$,
- *antisymmetric* if
$$a\, R\, b \quad \text{implies} \quad \neg(b\, R\, a)$$
for all $a \neq b \in A$,
- a *weak partial order* iff it is transitive, reflexive and antisymmetric.

[1] MIT's Committee on Curricula has the responsibility of watching out for such bugs that might creep into departmental requirements.

Some authors define partial orders to be what we call weak partial orders, but we'll use the phrase "partial order" to mean either a weak or strict one.

For weak partial orders in general, we often write an ordering-style symbol like $\preceq$ or $\sqsubseteq$ instead of a letter symbol like R. (General relations are usually denoted by a letter like R instead of a cryptic squiggly symbol, so $\preceq$ is kind of like Prince.) Likewise, we generally use $\prec$ or $\sqsubset$ to indicate a strict partial order. We also write $b \succeq a$ to mean $a \preceq b$ and $b \succ a$ to mean $a \prec b$.

Two more examples of partial orders are worth mentioning:

Example 2.3. Let A be some family of sets and define aRb iff $a \supset b$. Then R is a strict partial order.

For integers, m, n we write $m \mid n$ to mean that m *divides* n, namely, there is an integer, k, such that $n = km$.

Example 2.4. The divides relation is a weak partial order on the nonnegative integers.

2.2 Representing Partial Orders by Set Containment

When a class of objects are defined by axioms, like the axioms for partial orders, it can help to have a way to "represent" them explicitly by known objects. Partial orders can be represented by the subset relation on a collection of sets. Namely, if R is a weak partial order on a set, A, we can let each element $a \in A$ correspond to the set $R\{a\}$. Since

$$a\,R\,b \quad \text{iff} \quad R\{a\} \subseteq R\{b\} \tag{1}$$

holds for all $a, b \in A$, we have completely captured the weak partial order R by the subset relation on the corresponding sets. A similar correspondence shows that strict partial orders can be represented by the proper subset relation, $\subset$.

Problem 1. Prove the iff assertion (1).

Problem 2. Verify that the relations in Examples 2.3 and 2.4 are partial orders.

Definition 2.5. A relation, R, on a set, A, is *irreflexive* iff for all $a \in A$, it is *not* true that $a\,R\,a$.

Problem 3. Prove that a binary relation is a strict partial order iff it is transitive and irreflexive.

2.3 Total Orders

The familiar order relations on numbers have an important additional property: given any two numbers, one will be bigger than the other. Partial orders with this property are said to be *total*[2] *orders*:

Definition 2.6. Let R be a binary relation on a set, A, and let a, b be elements of A. Then a and b are *comparable* with respect to R iff ($a\,R\,b$ or $b\,R\,a$). A partial order under which every two distinct elements are comparable is called a *total order*.

So $<$ and $\leq$ are total orders on $\mathbb{R}$. On the other hand, the subset relation is *not* total, since, for example, any two distinct finite sets of the same size will be incomparable under $\subseteq$. The prerequisite relation on Course 6 required subjects is also not total because, for example, neither 8.01 nor 6.001 is a prerequisite of the other.

[2] "Total" is an overloaded term when talking about partial orders: being a total order is a much stronger condition than being a partial order that is a total relation. For example, any weak partial order such as $\subseteq$ is a total relation.

2.4 Products of Relations

Taking the product of two relations is a useful way to construct new relations from old ones.

The product, $R_1 \times R_2$, of relations R_1 and R_2 is defined to be the relation with

$$\begin{aligned} \text{domain}\,(R_1 \times R_2) &::= \text{domain}\,(R_1) \times \text{domain}\,(R_2)\,, \\ \text{codomain}\,(R_1 \times R_2) &::= \text{codomain}\,(R_1) \times \text{codomain}\,(R_2)\,, \\ (a_1, a_2)\,(R_1 \times R_2)\,(b_1, b_2) &\quad \text{iff} \quad [a_1\, R_1\, b_1 \text{ and } a_2\, R_2\, b_2]. \end{aligned}$$

Example 2.7. Define a relation, Y, on age-height pairs of being younger *and* shorter. This is the relation on the set of pairs (y, h) where y is a natural number ≤ 2400 which we interpret as an age in months, and h is a natural number ≤ 120 describing height in inches. We define Y by the rule

$$(y_1, h_1)\, Y\, (y_2, h_2) \quad \text{iff} \quad y_1 \leq y_2 \wedge h_1 \leq h_2.$$

That is, Y is the product of the $\leq$-relation on ages and the $\leq$-relation on heights.

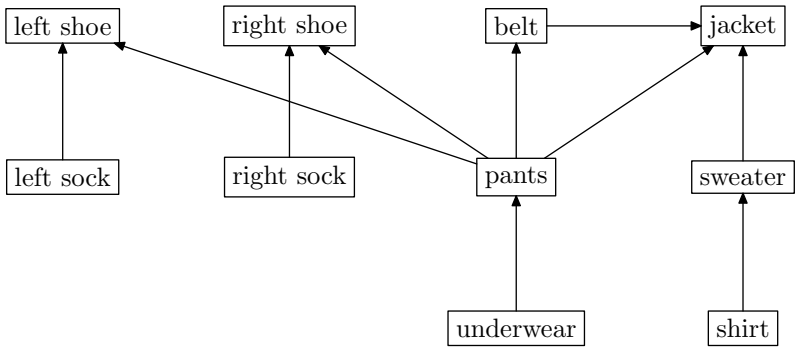
Products preserve several of the relational properties we have considered. Namely, it's not hard to verify that if R_1 and R_2 are both transitive, then so is $R_1 \times R_2$. The same holds for reflexivity, irreflexivity, and antisymmetry. This implies that if R_1 and R_2 are both partial orders, then so is $R_1 \times R_2$.

On the other hand, the property of being a total order is not preserved. For example, the age-height relation Y is the product of two total orders, but it is not total: the age 240 months, height 68 inches pair, (240,68), and the pair (228,72) are incomparable under Y.

2.5 Topological Sorting

Scheduling problems are a common source of partial orders: there is a set, A, of tasks and a set of constraints specifying that starting a certain task depends on other tasks being completed beforehand. We can picture the constraints by drawing labelled boxes corresponding to different tasks, with an arrow from one box to another if the first box corresponds to a task that must be completed before starting the second one.

Example 2.8. Here is a drawing describing the order in which you could put on clothes. The tasks are the clothes to be put on, and the arrows indicate what should be put on directly before what.



When we have a partial order of tasks to be performed, it can be useful to have an order in which to perform all the tasks, one at a time, while respecting the dependency constraints. This amounts

to finding a total order that is consistent with the partial order. This task of finding a total ordering that is consistent with a partial order is known as *topological sorting*.

Definition 2.9. A *topological sort* of a partial order, $\prec$, on a set, A, is a total ordering, $\sqsubset$, on A such that

$$a \prec b \quad \text{implies} \quad a \sqsubset b.$$

For example,

$$\text{shirt} \sqsubset \text{sweater} \sqsubset \text{underwear} \sqsubset \text{leftsock} \sqsubset \text{rightsock} \sqsubset \text{pants} \sqsubset \text{leftshoe} \sqsubset \text{rightshoe} \sqsubset \text{belt} \sqsubset \text{jacket},$$

is one topological sort of the partial order of dressing tasks given by Example 2.8; there are several other possible sorts as well.

Topological sorts for partial orders on finite sets are easy to construct by starting from *minimal* elements:

Definition 2.10. Let $\preceq$ be a partial order on a set, A. An element $a \in A$ is *minimum* iff it is $\preceq$ every other element of A. The element a is minim*al* iff no other element is $\preceq a$.

In a total order, minimum and minimal elements are the same thing. But a partial order may no minimum element but lots of minimal elements. There are four minimal elements in the clothes example: leftsock, rightsock, underwear, and shirt.

To construct a total ordering for getting dressed, we pick one of these minimal elements, say shirt. Next we pick a minimal element among the remaining ones. For example, once we have removed shirt, sweater becomes minimal. We continue in this way removing successive minimal elements until all elements have been picked. The sequence of elements in the order they were picked will be a topological sort. This is how the topological sort above for getting dressed was constructed.

For this method of topological sorting to work, we need to be sure there is always a minimal element. This is sort of obvious, but noting that an infinite partially ordered set might have no minimal element —consider $<$ on the $\mathbb{Z}$ —it would be good to prove that minimal elements exist.

Lemma 2.11. *Every partial order on a nonempty finite set has a minimal element.*

Proof. Let R be a strict partial order on a set, A. Define the *weight* of an element $a \in A$ to be $|R\{a\}|$ —the number of elements in the set $R\{a\}$. Since A is finite, the weights of all elements in A are nonnegative integers, so there must be an $a_0 \in A$ with the smallest weight.

Now suppose $|R\{a_0\}| \neq 0$. Then there is an element $a_1 \in R\{a_0\}$, which implies (by transitivity of R) that $R\{a_1\} \subseteq R\{a_0\}$, and hence $|R\{a_1\}| \leq |R\{a_0\}|$. But since R is strict, $a_1 \in R\{a_0\} - R\{a_1\}$, so in fact $|R\{a_1\}| < |R\{a_0\}|$, contradicting the fact the a_0 has the smallest weight.

This contradiction implies that $|R\{a_0\}| = 0$, which means that no element is related by R to a_0, that is, a_0 is minimal.

A similar argument works in the case that R is a weak partial order.

□

So our construction shows:

Theorem 2.12. *Every partial order on a finite set has a topological sort.*

In fact, the domain of the partial order need not be finite: we won't prove it, but *all* partial orders, even infinite ones, have topological sorts.

There are many other ways of constructing topological sorts. For example, instead of starting "from the bottom" with minimal elements, we could start "from the top" by picking *maximal* elements:

Definition 2.13. Let $\preceq$ be a partial order on a set, A. An element $a \in A$ is *maximum* iff it is $\succeq$ every other element of A. The element a is maxim*al* iff no other element is $\succeq a$.

Problem 4. **(a)** Prove that there is at most one minim*um* element in any partial order.

(b) Give an example of a partial order with exactly one minimal element, but no minimum element. *Hint:* It will have be infinite.

2.6 Parallel Task Scheduling

For a partial order of task dependencies, topological sorting provides a way to execute tasks sequentially without violating the dependencies. But what if we have the ability to execute more than one task at the same time? For example, say tasks are programs, the partial order indicates data dependence, and we have a parallel machine with lots of processors instead of a sequential machine with only one. How should we schedule the tasks? Our goal should be to minimize the total *time* to complete all the tasks. For simplicity, let's say all the tasks take the same amount of time and all the processors are identical.

So, given a finite partially ordered set of tasks, how long does it take to do them all, in an optimal parallel schedule? We can also use partial order concepts to analyze this problem.

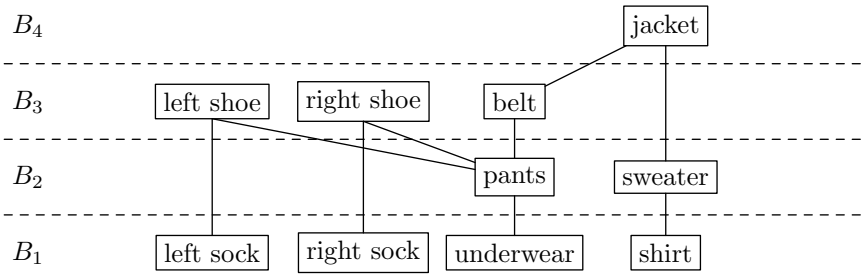
In the clothes example, we could do all the minimal elements first (leftsock, rightsock, underwear, shirt), remove them and repeat. We'd need lots of hands, or maybe dressing servants. We can do pants and sweater next, and then leftshoe, rightshoe, and belt, and finally jacket.

We can't do any better, because the sequence underwear, pants, belt, jacket must be done in that order. A set of tasks that must be done in sequence like this is called a *chain*.

Definition 2.14. A *chain* in a partial order is a set of elements such that any two elements in the set are comparable.

In other words, a chain is a totally ordered subset of the elements in a partial order. Clearly, the parallel time must be at least the size of any chain. For if we used less time, then two tasks in the chain would have to be done at the same time, violating the dependency constraints.

A largest chain is also known as a *critical path*. So we need at least t steps, where t is the size of a largest chain. Fortunately, it is always possible to use only t parallel steps. The idea is to let B_1 be all the minimal elements and schedule them first. Then remove all the elements in B_1, let B_2 be the elements that now become minimal, and schedule them next, and so on. For getting dressed, here is a picture of the schedule obtained in this way:



Theorem 2.15. *Let R be strict partial order on a set, A. If the longest chain in A is of size t, then there is a partition[3] of A into t blocks, $B_1, B_2, \ldots, B_t$, such that for each block, B_i, all tasks that have to precede tasks in B_i are in smaller-numbered groups. That is,*

$$RB_1 = \emptyset, \text{ and} \tag{2}$$
$$RB_i \subseteq B_1 \cup B_2 \cup \cdots \cup B_{i-1}, \tag{3}$$

for $1 < i \leq t$.

Corollary 2.16. *For R and t as above, it is possible to schedule all tasks in t steps.*

Proof. Schedule all the elements of B_i at time i. This satisfies the dependency requirements, because all the tasks that any task depends on are scheduled at preceding times. □

Corollary 2.17. *Parallel time = Size of largest chain.*

So it remains to prove Theorem 2.15:

Proof. A chain is said to *begin* with its smallest element and *end* with its largest element, if any.

Construct the sets B_i as follows:

$$B_i ::= \{a \in A \mid \text{the largest chain ending in } a \text{ is of size } i\}.$$

This gives just t sets, because the largest chain is of size t. Also, each $a \in A$ belongs to exactly one B_i. To complete the proof, notice that if $a \in B_1$, then a must be minimal, and since R is strict we have $RB_1 = \emptyset$ proving (2).

Now suppose $1 < i \leq t$, and assume for the sake of contradiction that (3) does not hold. That is, there is an $a \in B_i$ and $b \in A$ such that $b\,R\,a$, and $b \notin B_1 \cup B_2 \cup \cdots \cup B_{i-1}$. Then by definition of the B_j's, there is a chain of size $> i-1$ ending at b. Also, since R is strict, a is not in the chain ending at b. So we can add a to the end of the chain to obtain a chain of size $> i$ ending in a, contradicting the fact that that $a \in B_i$. □

[3] Partitioning a set, A, means "cutting it up" into non-overlapping, nonempty pieces. The pieces are called the blocks of the partition. More precisely, a *partition* of A is a set $\mathcal{B}$ whose elements are nonempty subsets of A such that

- if $B, B' \in \mathcal{B}$ are distinct sets, then $B \cap B' = \emptyset$, and
- $\bigcup_{B \in \mathcal{B}} B = A$.

So with an unlimited number of processors, the time to complete all the tasks is the size of the largest chain. It turns out that this theorem is good for more than parallel scheduling. It is usually stated as follows.

Definition 2.18. An *antichain* in a partial order is a set of elements such that any two elements in the set are incomparable.

Corollary 2.19. *If the largest chain in a partial order is of size, t, then the domain can be partitioned into t antichains.*

Proof. Let the antichains be the sets B_i defined as in the proof of Theorem 2.15.

We should verify that each B_i is an antichain, namely, if a, b are distinct elements of B_i, then they are incomparable. But suppose to the contrary that there exist two elements $a, b \in B_i$ such that a and b are comparable, say $a \, R \, b$. Then, as in the proof of Theorem 2.15, by adding b at the end of the chain of size i ending at a, we obtain a chain of size $i + 1$ ending at b, contradicting the assumption that $b \in B_i$. □

2.7 Dilworth's Lemma

We can use the Corollary 2.19 to prove a famous result[4] about partially ordered sets:

Lemma 2.20 (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. Assume there is no chain of size greater than t, that is, the largest chain is of size $\leq t$. Then by Corollary 2.19, the n elements can be partitioned into at most t antichains. Let ℓ be the size of the largest antichain. Since every element belongs to exactly one antichain, and there are at most t antichains, there can't be more than ℓt elements, namely, $\ell t \geq n$. So there is an antichain with at least $\ell \geq n/t$ elements. □

Corollary 2.21. *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 2.20. □

Example 2.22. In the dressing partially ordered set, $n = 10$.

Try $t = 3$. There is a chain of size 4.

Try $t = 4$. There is no chain of size 5, but there is an antichain of size $4 \geq 10/4$.

Example 2.23. Suppose we have a class of 101 students. Then using the product partial order, Y, from Example 2.7, we can apply Dilworth's Lemma to conclude that there is a chain of 11 students who get taller as they get older, or an antichain of 11 students who get taller as they get younger, which makes for an amusing in-class demo.

Quick Exercise: What is the size of the longest chain that is guaranteed to exist in any partially ordered set of n elements? What about the largest antichain?

[4] Lemma 2.20 also follows from a more general result known as Dilworth's Theorem which we will not discuss.

3 Induction

Induction is by far the most powerful and commonly-used proof technique in Discrete Mathematics and Computer Science. In fact, the use of induction is a defining characteristic of *discrete* —as opposed to *continuous* —Mathematics.

To understand how induction works, suppose there is a professor who brings to class a bottomless bag of assorted miniature candy bars. She offers to share in accordance with two rules. First, she numbers the students 0, 1, 2, 3, and so forth for convenient reference. Now here are the two rules:

1. Student 0 gets candy.
2. If student n gets candy, then student $n + 1$ also gets candy, for every $n \in \mathbb{N}$.

You can think of the second rule as a compact way of writing a whole sequence of statements, one for each natural value of n:

- If student 0 gets candy, then student 1 also gets candy.
- If student 1 gets candy, then student 2 also gets candy.
- If student 2 gets candy, then student 3 also gets candy. ⋮

Now suppose you are student 17. By these rules, are you entitled to a miniature candy bar? Well, student 0 gets candy by the first rule. Therefore, by the second rule, student 1 also gets candy, which means student 2 gets candy as well, which means student 3 get candy, and so on. So the professor's two rules actually guarantee candy for *every* student, no matter how large the class. You win!

This kind of reasoning is an instance of

The Principle of Induction. Let $P(n)$ be a predicate. If

- $P(0)$ is true, and
- for all $n \in \mathbb{N}$, $P(n)$ implies $P(n + 1)$,

then $P(n)$ is true for all $n \in \mathbb{N}$.

Here's the correspondence between the induction principle and sharing candy bars. Suppose that $P(n)$ is the predicate, "student n gets candy". Then the professor's first rule asserts that $P(0)$ is true, and her second rule is that for all $n \in \mathbb{N}$, $P(n)$ implies $P(n + 1)$. Given these facts, the induction principle says that $P(n)$ is true for all $n \in \mathbb{N}$. In other words, everyone gets candy.

The intuitive justification for the general induction principle is the same as for everyone getting a candy bar under the professor's two rules. So the Principle of Induction is universally accepted as an obvious, sound proof method. What's not so obvious is how much mileage we get by using it.

4 Using Induction

Induction often works directly in proving that some statement about natural numbers holds for all of them. For example, here is a classic formula:

Theorem 4.1. *For all* $n \in \mathbb{N}$,

$$1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2} \tag{4}$$

The left side of equation (4) represents the sum of all the numbers from 1 to n. Here the dots ($\cdots$) indicate a pattern you're supposed to be able to guess so you can mentally fill in the remaining terms.

The meaning of this sum is not so obvious in a couple of special cases:

- If $n = 1$, then there is only one term in the summation, and so $1 + 2 + 3 + \cdots + n = 1$. Don't be misled by the appearance of 2 and 3 and the suggestion that 1 and n are distinct terms!
- If $n \leq 0$, then there are no terms at all in the summation. By convention, the sum in this case is 0.

So while the dots notation is convenient, you have to watch out for these special cases where the notation is misleading! (In fact, whenever you see the dots, you should be on the lookout to be sure you understand the pattern.)

We could eliminate the need for guessing by rewriting the left side of (4) with ***summation notation***:

$$\sum_{i=1}^{n} i \qquad \text{or} \qquad \sum_{1 \leq i \leq n} i.$$

Both of these expressions denote the sum of all values taken on by the expression to the right of the sigma as the variable, i, ranges from 1 to n. Both these summation expressions make it clear what (4) means when $n = 1$. The second expression makes it clear that when $n = 0$, there are no terms in the sum, though you still have to know the convention that a sum of no numbers equals 0 (the *product* of no numbers is 1, by the way).

Now let's use the induction principle to prove Theorem 4.1. Suppose that we define predicate $P(n)$ to be "$1 + 2 + 3 + \cdots + n = n(n+1)/2$". Recast in terms of this predicate, the theorem claims that $P(n)$ is true for all $n \in \mathbb{N}$. This is great, because the induction principle lets us reach precisely that conclusion, provided we establish two simpler facts:

- $P(0)$ is true.
- For all $n \in \mathbb{N}$, $P(n)$ implies $P(n+1)$.

So now our job is reduced to proving these two statements. The first is true because $P(0)$ asserts that a sum of zero terms is equal to $0(0+1)/2 = 0$.

The second statement is more complicated. But remember the basic plan for proving the validity of any implication: *assume* the statement on the left and then *prove* the statement on the right. In this case, we assume $P(n)$:

$$1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2} \tag{5}$$

in order to prove $P(n+1)$:

$$1 + 2 + 3 + \cdots + n + (n+1) = \frac{(n+1)(n+2)}{2} \tag{6}$$

These two equations are quite similar; in fact, adding $(n+1)$ to both sides of equation (5) and simplifying the right side gives the equation (6):

$$\begin{aligned} 1 + 2 + 3 + \cdots + n + (n+1) &= \frac{n(n+1)}{2} + (n+1) \\ &= \frac{(n+2)(n+1)}{2} \end{aligned}$$

Thus, if $P(n)$ is true, then so is $P(n+1)$. This argument is valid for every natural number n, so this establishes the second fact required by the induction principle. In effect, we've just proved that $P(0)$ implies $P(1)$, $P(1)$ implies $P(2)$, $P(2)$ implies $P(3)$, etc., all in one fell swoop.

With these two facts in hand, the induction principle says that the predicate $P(n)$ is true for all natural n, so the theorem is proved.

4.1 A Template for Induction Proofs

The proof of Theorem 4.1 was relatively simple, but even the most complicated induction proof follows exactly the same template. There are five components:

1. **State that the proof uses induction.** This immediately conveys the overall structure of the proof, which helps the reader understand your argument.

2. **Define an appropriate predicate $P(n)$.** The eventual conclusion of the induction argument will be that $P(n)$ is true for all natural n. Thus, you should define the predicate $P(n)$ so that your theorem is equivalent to (or follows from) this conclusion. Often the predicate can be lifted straight from the claim, as in the example above. The predicate $P(n)$ is called the "induction hypothesis". Sometimes the induction hypothesis will involve several variables, in which case you should indicate which variable serves as n.

3. **Prove that $P(0)$ is true.** This is usually easy, as in the example above. This part of the proof is called the "base case" or "basis step". (Sometimes the base case will be $n = 1$ or even some larger number, in which case the starting value of n also should be stated.)

4. **Prove that $P(n)$ implies $P(n+1)$ for every natural number n.** This is called the "inductive step" or "induction step". The basic plan is always the same: assume that $P(n)$ is true and then use this assumption to prove that $P(n+1)$ is true. These two statements should be fairly similar, but bridging the gap may require some ingenuity. Whatever argument you give must be valid for every natural number n, since the goal is to prove the implications $P(0) \to P(1)$, $P(1) \to P(2)$, $P(2) \to P(3)$, etc. all at once.

5. **Invoke induction.** Given these facts, the induction principle allows you to conclude that $P(n)$ is true for all natural n. This is the logical capstone to the whole argument, but many writers leave this step implicit.

Explicitly labeling the *base case* and *inductive step* may make your proofs clearer.

4.2 A Clean Writeup

The proof of Theorem 4.1 given above is perfectly valid; however, it contains a lot of extraneous explanation that you won't usually see in induction proofs. The writeup below is closer to what you might see in print and should be prepared to produce yourself.

Proof. We use induction. The induction hypothesis, $P(n)$, will be equation (4).

Base case: $P(0)$ is true, because both sides of equation (4) equal zero when $n = 0$.

Inductive step: Assume that $P(n)$ is true, where n is any natural number. Then

$$\begin{aligned} 1+2+3+\cdots+n+(n+1) &= \frac{n(n+1)}{2}+(n+1) && \text{by induction hypothesis} \\ &= \frac{(n+1)(n+2)}{2} && \text{by simple algebra} \end{aligned}$$

which proves $P(n+1)$.

So it follows by induction that $P(n)$ is true for all natural n. □

Induction was helpful for *proving the correctness* of this summation formula, but not helpful for *discovering* it in the first place. We'll show you some tricks for finding such formulas in a few weeks.

4.3 Powers of Odd Numbers

A proof in class that $\sqrt[n]{2}$ is irrational used the "obvious":

Fact. The nth power of an odd number is odd, for all nonnegative integers, n.

Instead of taking this fact for granted, we can prove it by induction. The proof will require a simple Lemma.

Lemma. *The product of two odd numbers is odd.*

To prove the Lemma, note that the odd numbers are, by definition, the numbers of the form $2k+1$ where k is an integer. But

$$(2k+1)(2k'+1) = 2(2kk'+k+k')+1,$$

so the product of two odd numbers also has the form of an odd number, which proves the Lemma.

Now we will prove the Fact using the induction hypothesis

$$P(n) ::= \text{if } a \text{ is an odd integer, then so is } a^n.$$

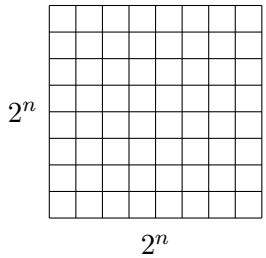
The base case $P(0)$ holds because $a^0 = 1$, and 1 is odd.

For the inductive step, suppose $n \geq 0$, a is an odd number and $P(n)$ holds. So a^n is an odd number. Therefore, $a^{n+1} = a^n a$ is a product of odd numbers, and by the Lemma a^{n+1} is also odd. This proves $P(n+1)$, and we conclude by induction that $P(n)$ holds for nonnegative integers n.

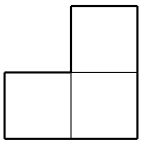
4.4 Courtyard Tiling

Induction served purely as a proof technique in the preceding examples. But induction sometimes can serve as a more general reasoning tool.

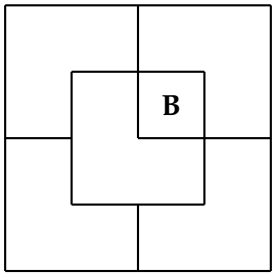
MIT recently constructed the Stata Center which houses the Computer Science and AI Laboratory. During development, the project went further and further over budget, and there were some radical fundraising ideas. One rumored plan was to install a big courtyard with dimensions $2^n \times 2^n$:



One of the central squares would be occupied by a statue of a wealthy potential donor. Let's call him "Bill". (In the special case $n = 0$, the whole courtyard consists of a single central square; otherwise, there are four central squares.) A complication was that the building's unconventional architect, Frank Gehry, supposedly insisted that only special L-shaped tiles be used:



A courtyard meeting these constraints exists, at least for $n = 2$:



For larger values of n, is there a way to tile a $2^n \times 2^n$ courtyard with L-shaped tiles and a statue in the center? Let's try to prove that this is so.

Theorem 4.2. *For all $n \geq 0$ there exists a tiling of a $2^n \times 2^n$ courtyard with Bill in a central square.*

Proof. (doomed attempt) The proof is by induction. Let $P(n)$ be the proposition that there exists a tiling of a $2^n \times 2^n$ courtyard with Bill in the center.

Base case: $P(0)$ is true because Bill fills the whole courtyard.

Inductive step: Assume that there is a tiling of a $2^n \times 2^n$ courtyard with Bill in the center for some $n \geq 0$. We must prove that there is a way to tile a $2^{n+1} \times 2^{n+1}$ courtyard with Bill in the center ....

□

Now we're in trouble! The ability to tile a smaller courtyard with Bill in the center isn't much help in tiling a larger courtyard with Bill in the center. We haven't figured out how to bridge the gap between $P(n)$ and $P(n+1)$.

So if we're going to prove Theorem 4.2 by induction, we're going to need some *other* induction hypothesis than simply the statement about n that we're trying to prove.

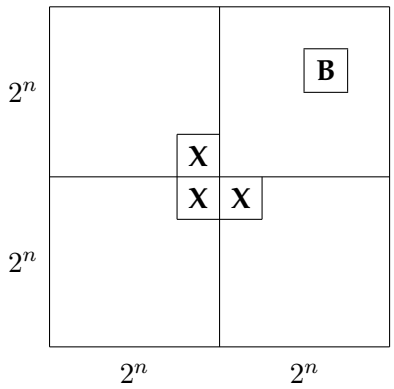
When this happens, your first fallback should be to look for a *stronger* induction hypothesis; that is, one which implies your previous hypothesis. For example, we could make $P(n)$ the proposition that for *every* location of Bill in a $2^n \times 2^n$ courtyard, there exists a tiling of the remainder.

This advice may sound bizarre: "If you can't prove something, try to prove something grander!" But for induction arguments, this makes sense. In the inductive step, where you have to prove $P(n) \longrightarrow P(n+1)$, you're in better shape because you can *assume* $P(n)$, which is now a more powerful statement. Let's see how this plays out in the case of courtyard tiling.

Proof. (successful attempt) The proof is by induction. Let $P(n)$ be the proposition that for every location of Bill in a $2^n \times 2^n$ courtyard, there exists a tiling of the remainder.

Base case: $P(0)$ is true because Bill fills the whole courtyard.

Inductive step: Assume that $P(n)$ is true for some $n \geq 0$; that is, for every location of Bill in a $2^n \times 2^n$ courtyard, there exists a tiling of the remainder. Divide the $2^{n+1} \times 2^{n+1}$ courtyard into four quadrants, each $2^n \times 2^n$. One quadrant contains Bill (**B** in the diagram below). Place a temporary Bill (**X** in the diagram) in each of the three central squares lying outside this quadrant:



Now we can tile each of the four quadrants by the induction assumption. Replacing the three temporary Bills with a single L-shaped tile completes the job. This proves that $P(n)$ implies $P(n+1)$ for all $n \geq 0$. The theorem follows as a special case. □

This proof has two nice properties. First, not only does the argument guarantee that a tiling exists, but also it gives an algorithm for finding such a tiling. Second, we have a stronger result: if Bill wanted a statue on the edge of the courtyard, away from the pigeons, we could accommodate him!

Strengthening the induction hypothesis is often a good move when an induction proof won't go through. But keep in mind that the stronger assertion must actually be *true*; otherwise, there isn't much hope of constructing a valid proof! Sometimes finding just the right induction hypothesis requires trial, error, and insight. For example, mathematicians spent almost twenty years trying to prove or disprove the conjecture that "Every planar graph is 5-choosable"[5]. Then, in 1994, Carsten Thomassen gave an induction proof simple enough to explain on a napkin. The key turned out to be finding an extremely clever induction hypothesis; with that in hand, completing the argument is easy!

4.5 A Faulty Induction Proof

False Theorem. *All horses are the same color.*

Notice that no n is mentioned in this assertion, so we're going to have to reformulate it in a way that makes an n explicit. In particular, we'll (falsely) prove that

False Theorem 4.3. *In every set of $n \geq 1$ horses, all are the same color.*

This a statement about all integers $n \geq 1$ rather ≥ 0, so it's natural to use a slight variation on induction: prove $P(1)$ in the base case and then prove that $P(n)$ implies $P(n+1)$ for all $n \geq 1$ in the inductive step. This is a perfectly valid variant of induction and is *not* the problem with the proof below.

Proof. The proof is by induction on n. The induction hypothesis, $P(n)$, will be

$$\text{In every set of } n \text{ horses, all are the same color.} \tag{7}$$

Base case: ($n = 1$). $P(1)$ is true, because in a set of horses of size 1, there's only one horse, and this horse is definitely the same color as itself.

Inductive step: Assume that $P(n)$ is true for some $n \geq 1$. that is, assume that in every set of n horses, all are the same color. Now consider a set of $n+1$ horses:

$$h_1,\ h_2,\ \ldots,\ h_n,\ h_{n+1}$$

[5] 5-choosability is a slight generalization of 5-colorability. Although every planar graph is 4-colorable and therefore 5-colorable, not every planar graph is 4-choosable. If this all sounds like nonsense, don't panic. We'll discuss graphs, planarity, and coloring in two weeks.

By our assumption, the first n horses are the same color:

$$\underbrace{h_1,\ h_2,\ \ldots,\ h_n,}_{\text{same color}}\ h_{n+1}$$

Also by our assumption, the last n horses are the same color:

$$h_1,\ \underbrace{h_2,\ \ldots,\ h_n,\ h_{n+1}}_{\text{same color}}$$

So h_1 is the same color as the remaining horses besides h_{n+1}, and likewise h_{n+1} is the same color as the remaining horses besides h_1. So h_1 and h_{n+1} are the same color. That is, horses $h_1, h_2, \ldots, h_{n+1}$ must all be the same color, and so $P(n+1)$ is true. Thus, $P(n)$ implies $P(n+1)$.

By the principle of induction, $P(n)$ is true for all $n \geq 1$. □

We've proved something false! Is Math broken? Should we all become poets?

The error in this argument is in the sentence that begins, "So h_1 and h_{n+1} are the same color." The "..." notation creates the impression that there are some remaining horses besides h_1 and h_{n+1}. However, this is not true when $n = 1$. In that case, the first set is just h_1 and the second is h_2, and there are no remaining horses besides them. So h_1 and h_2 need not be the same color!

This mistake knocks a critical link out of our induction argument. We proved $P(1)$ and we *correctly* proved $P(2) \longrightarrow P(3)$, $P(3) \longrightarrow P(4)$, etc. But we failed to prove $P(1) \longrightarrow P(2)$, and so everything falls apart: we can not conclude that $P(2)$, $P(3)$, etc., are true. And, of course, these propositions are all false; there are horses of a different color.

Students sometimes claim that the mistake in the proof is because $P(n)$ is false for $n \geq 2$, and the proof assumes something false, namely, $P(n)$, in order to prove $P(n+1)$. You should think about how to explain to such a student why this claim would get no credit on a 6.042 exam.

5 Strong Induction

5.1 The Strong Induction Principle

A useful variant of induction is called *strong induction*. Strong induction and ordinary induction are used for exactly the same thing: proving that a predicate $P(n)$ is true for all $n \in \mathbb{N}$.

Principle of Strong Induction. Let $P(n)$ be a predicate. If

- $P(0)$ is true, and
- for all $n \in \mathbb{N}$, $P(0)$, $P(1)$, ..., $P(n)$ *together* imply $P(n+1)$,

then $P(n)$ is true for all $n \in \mathbb{N}$.

The only change from the ordinary induction principle is that strong induction allows you to assume more stuff in the inductive step of your proof! In an ordinary induction argument, you assume that $P(n)$ is true and try to prove that $P(n+1)$ is also true. In a strong induction argument, you may assume that $P(0)$, $P(1)$, ..., and $P(n)$ are *all* true when you go to prove $P(n+1)$. These extra assumptions can only make your job easier.

5.2 Products of Primes

As a first example, we'll use strong induction to prove one of those familiar facts that is almost, but maybe not entirely, obvious:

Lemma 5.1. *Every integer greater than 1 is a product of primes.*

Note that, by convention, any number is considered to be a product consisting of one term, namely itself. In particular, every prime is considered to be a product whose terms are all primes.

Proof. We will prove Lemma 5.1 by strong induction, letting the induction hypothesis, $P(n)$, be

$$n+2 \text{ is a product of primes.}$$

So Lemma 5.1 will follow if we prove that $P(n)$ holds for all $n \geq 0$.

Base Case: $P(0)$ is true because $0+2$ is prime, and so is a product of primes by convention.

Inductive step: Suppose that $n \geq 0$ and that $i+2$ is a product of primes for every natural number $i < n+1$. We must show that $P(n+1)$ holds, namely, that $n+3$ is also a product of primes. We argue by cases:

If $n+3$ is itself prime, then it is a product of primes by convention, so $P(n+1)$ holds in this case.

Otherwise, $n+3$ is not prime, which by definition means $n+3 = km$ for some natural numbers k, m such that $2 \leq k, m < n+3$. So $k-2$ is a natural number less than $n+1$, which means that $(k-2)+2$ is a product of primes by induction hypothesis. That is, k is a product of primes. Likewise, m is a product of primes. So $km = n+3$ is also a product of primes. Therefore, $P(n+1)$ holds in this case as well.

So $P(n+1)$ holds in any case, which completes the proof by strong induction that $P(n)$ holds for all natural numbers, n.

□

Despite the name, strong induction is actually no more powerful than ordinary induction. In other words, any theorem that can be proved with strong induction could also be proved with ordinary induction (using a slightly more complicated induction hypothesis). But strong induction can make some proofs a bit easier. On the other hand, if $P(n)$ is easily sufficient to prove $P(n+1)$, then it's better to use ordinary induction for simplicity.

5.3 Making Change

The country Inductia, whose unit of currency is the Strong, has coins worth 6S (6 Strongs), 10S and 15S. Although the Inductians have some trouble making small change like 11S or 29S, it turns out that they can collect coins to make change for any number of Strongs greater than 29S.

Strong induction makes this easy to prove for $n + 1 > 35$, because then $(n + 1) - 6 > 29$, so by strong induction the Inductians can make change for exactly $((n+1)-6)$S, and then they can add a 6S coin to get $(n + 1)$S. So the only thing to do is check that they can make change for all the amounts from 30 to 35, which is not too hard to do.

Here's a detailed writeup using the official format:

Proof. We prove the Inductians can make change for any amount greater than 29S by strong induction. The induction hypothesis, $P(n)$ will be:

> If $n > 29$, then there is a collection of coins whose value is n Strongs.

Notice that $P(n)$ is an implication. When the hypothesis of an implication is false, we know the whole implication is true. In this situation, the implication is said to be *vacuously* true. So $P(n)$ will be vacuously true whenever $n \leq 29$.[6]

We now proceed with the induction proof:

Base case: $P(0)$ is vacuously true.

Inductive step: We assume $P(i)$ holds for all $i \leq n$, and prove that $P(n + 1)$ holds. We argue by cases:

Case ($n + 1 \leq 29$): $P(n + 1)$ is vacuously true in this case.

Case ($n + 1$ = 30): $P(30)$ holds because the Inductians can use five 6S coins.

Case ($n + 1$ = 31): Use a 6S coin, a 10S coin and a 15S coin.

Case ($n + 1$ = 32): Use two 6S coins, and two 10S coins.

Case ($n + 1$ = 33): Use three 6S coins, and a 15S coin.

Case ($n + 1$ = 34): Use a four 6S coins, and a 10S coin.

Case ($n + 1$ = 35): Use a two 10S coins and a 15S coin.

Case ($n+1 > 35$): Then $n \geq (n+1)-6 > 29$, so by the strong induction hypothesis, the Inductians can make change for $((n + 1) - 6)$S. Now by adding a 6S coin, they can make change for $(n + 1)$S.

So in any case, $P(n + 1)$ is true, and we conclude by strong induction that for all $n > 29$, the Inductians can make change for nS.

□

[6] A more elegant approach that avoids these vacuous cases is to define

$$P'(n) ::= \text{there is a collection of coins whose value is } n + 30 \text{ Strongs}$$

and prove that $P'(n)$ holds for all $n \geq 0$.

5.4 Unstacking

Here is another exciting 6.042 game that's surely about to sweep the nation!

You begin with a stack of n boxes. Then you make a sequence of moves. In each move, you divide one stack of boxes into two nonempty stacks. The game ends when you have n stacks, each containing a single box. You earn points for each move; in particular, if you divide one stack of height $a + b$ into two stacks with heights a and b, then you score ab points for that move. Your overall score is the sum of the points that you earn for each move. What strategy should you use to maximize your total score?

As an example, suppose that we begin with a stack of $n = 10$ boxes. Then the game might proceed as follows:

Stack Heights										**Score**
$\underline{10}$										
5	$\underline{5}$									25 points
$\underline{5}$	3	2								6
$\underline{4}$	3	2	1							4
2	$\underline{3}$	2	1	2						4
$\underline{2}$	2	2	1	2	1					2
1	$\underline{2}$	2	1	2	1	1				1
1	1	$\underline{2}$	1	2	1	1	1			1
1	1	1	1	$\underline{2}$	1	1	1	1		1
1	1	1	1	1	1	1	1	1	1	1
								Total Score	$=$	45 points

On each line, the underlined stack is divided in the next step. Can you find a better strategy?

5.4.1 Analyzing the Game

Let's use strong induction to analyze the unstacking game. We'll prove that your score is determined entirely by the number of boxes —your strategy is irrelevant!

Theorem 5.2. *Every way of unstacking n blocks gives a score of $n(n-1)/2$ points.*

There are a couple technical points to notice in the proof:

- The template for a strong induction proof is exactly the same as for ordinary induction.
- As with ordinary induction, we have some freedom to adjust indices. In this case, we prove $P(1)$ in the base case and prove that $P(1), \ldots, P(n)$ imply $P(n+1)$ for all $n \geq 1$ in the inductive step.

Proof. The proof is by strong induction. Let $P(n)$ be the proposition that every way of unstacking n blocks gives a score of $n(n-1)/2$.

Base case: If $n = 1$, then there is only one block. No moves are possible, and so the total score for the game is $1(1-1)/2 = 0$. Therefore, $P(1)$ is true.

Inductive step: Now we must show that $P(1), \ldots, P(n)$ imply $P(n+1)$ for all $n \geq 1$. So assume that $P(1), \ldots, P(n)$ are all true and that we have a stack of $n+1$ blocks. The first move must split this stack into substacks with positive sizes a and b where $a+b = n+1$ and $0 < a, b \leq n$. Now the total score for the game is the sum of points for this first move plus points obtained by unstacking the two resulting substacks:

$$\begin{aligned}
\text{total score} &= (\text{score for 1st move}) \\
&\quad + (\text{score for unstacking } a \text{ blocks}) \\
&\quad + (\text{score for unstacking } b \text{ blocks}) \\
&= ab + \frac{a(a-1)}{2} + \frac{b(b-1)}{2} && \text{by } P(a) \text{ and } P(b) \\
&= \frac{(a+b)^2 - (a+b)}{2} = \frac{(a+b)((a+b)-1)}{2} \\
&= \frac{(n+1)n}{2}
\end{aligned}$$

This shows that $P(1), P(2), \ldots, P(n)$ imply $P(n+1)$.

Therefore, the claim is true by strong induction. □

Problem 5. Define the ***potential***, $p(S)$, of a stack, S, of blocks to be $k(k+1)/2$ where k is the number of blocks in S. Define the potential, $p(A)$, of a set, A, of stacks to be the sum of the potentials of the stacks in A.

Generalize Theorem 5.2 to show that for any set, A, of stacks, if a sequence of moves starting with A leads to another set, B, of stacks, then the score for this sequence of moves is $p(A) - p(B)$.

6 The Well Ordering Principle

Another proof method closely related to induction depends on the

Well Ordering Principle. Every *nonempty* set of *nonnegative integers* has a *smallest* element.

Do you believe this statement? Seems sort of obvious, right? But notice how tight it is: it requires a *nonempty* set —it's false for the empty set which has *no* smallest element because it has no elements at all! And it requires a set of *nonnegative* integers —it's false for the set of *negative* integers and also false for some sets of nonnegative *rationals* —for example, the set of positive rationals. So, the Well Ordering Principle captures something special about the natural numbers.

While the Well Ordering Principle may seem obvious, it looks nothing like the induction axiom, and it's harder to see offhand why it is useful. But in fact, it's as powerful as strong induction. We'll explain this after we introduce a template for well ordering principle proofs resembling the template in Section 4.1 for a proof by strong induction.

In fact, looking back, we took the Well Ordering Principle for granted in proving Lemma 2.11, that every finite partial order has a minimal element. We even implicitly relied on the Well Ordering Principle in the proof in Week 2 Notes that $\sqrt{2}$ is irrational. That proof assumed that any nonzero

fraction can be written in *lowest terms*, that is, in the form m/n where m and n are integers with no common factors. How do we know this is always possible?

Suppose to the contrary that there is a nonzero fraction that cannot be written in lowest terms. Now let C be the set of positive integers that are numerators of such fractions. Then C is nonempty. To prove this, suppose m/n is a nonzero fraction that cannot be written in lowest terms. Then neither can $-m/(-n)$, so one of m or $-m$ must be in C.

Therefore, by Well Ordering, there must be a smallest integer, $m_0 \in C$. So by definition of C, there must be some integer, n_0, such that the fraction m_0/n_0 cannot be in written in lowest terms. This means that m_0 and n_0 must have a common factor, $p > 1$. But then $(m_0/p)/(n_0/p)$ cannot be in lowest terms either, since it equals m_0/n_0. So $m_0/p \in C$ by definition of C. But $m_0/p < m_0$, which contradicts the fact that m_0 is the smallest element of C.

Since the assumption that C is nonempty leads to a contradiction, it follows that C must be empty. That is, that there are no numerators of fractions that can't be written in lowest terms, and hence there are no such fractions at all.

We've been using the Well Ordering Principle on the sly from early on!

Here is a standard way to organize a well ordering proof.

To prove that "$P(n)$ is true for all $n \in \mathbb{N}$" using the Well Ordering Principle:

- Define the set, C, of *counterexamples* to P being true. Namely, define
$$C ::= \{n \in \mathbb{N} \mid \neg P(n)\}.$$
- Assume for proof by contradiction that C is nonempty.
- By the Well Ordering Principle, there will be a smallest element, s, in C.
- Reach a contradiction (somehow) —often by showing how to use s to find another member of C that is smaller than s. (This is the open-ended part of the proof task.)
- Conclude that C must be empty, that is, no counterexamples exist. QED

Now we can explain why the Well Ordering Principle is as powerful a proof method as Strong Induction. In fact, we will explain hwo to take any proof by Strong Induction and reformat it into a Well Ordering proof.

Here's how: suppose that we have a proof Strong Induction with induction hypothesis $P(n)$. Then we start a Well Ordering proof by defining the set of counterexamples to P, and then assuming there is a smallest counterexample, s. This means that $P(s)$ is false, but also $P(0), P(1), \ldots, P(s-1)$ are all true. At this point we reuse the proof of the inductive step in the Strong Induction proof, which shows that since $P(0), P(1), \ldots, P(s-1)$ are all true, then $P(s)$ is also true. This contradicts the assumption that $P(s)$ is false, so we have the contradiction needed to complete the Well Ordering Proof that $\forall n.\, P(n)$.

Problem 6. Use strong induction to prove the Well Ordering Principle. *Hint:* Prove that if a set of nonnegative integers contains an integer, n, then it has a smallest element.

Mathematicians commonly use the Well Ordering Principle because it can lead to shorter proofs than induction. On the other hand, well ordering proofs typically involve proof by contradiction, so using it is not always the best approach. The choice of method is really a matter of style—but style does matter.