

## Induction, II.

### 1 Good Proofs and Bad Proofs

In a purely technical sense, a mathematical proof is verification of a proposition by a chain of logical deductions from a base set of axioms. But the *purpose* of a proof is to provide readers with compelling evidence for the truth of an assertion. To serve this purpose effectively, more is required of a proof than just logical correctness: a good proof must also be clear. These goals are complimentary; a well-written proof is more likely to be a correct proof, since mistakes are harder to hide. Here are some tips on writing good proofs:

**State your game plan.** A good proof begins by explaining the general line of reasoning, e.g. “We use induction” or “We argue by contradiction”. This creates a rough mental picture into which the reader can fit the subsequent details.

**Keep a linear flow.** We sometimes see proofs that are like mathematical mosaics, with juicy tidbits of reasoning sprinkled judiciously across the page. This is not good. The steps of your argument should follow one another in a clear, sequential order.

**Explain your reasoning.** Many students initially write proofs the way they compute integrals. The result is a long sequence of expressions without explanation. This is bad. A good proof usually looks like an essay with some equations thrown in. Use complete sentences.

**Avoid excessive symbolism.** Your reader is probably good at understanding words, but much less skilled at reading arcane mathematical symbols. So use words where you reasonably can.

**Simplify.** Long, complicated proofs take the reader more time and effort to understand and can more easily conceal errors. So a proof with fewer logical steps is a better proof.

**Introduce notation thoughtfully.** Sometimes an argument can be greatly simplified by introducing a variable, devising a special notation, or defining a new term. But do this sparingly, since you’re requiring the reader to remember all this new stuff. And remember to actually *define* the meanings of new variables, terms, or notations; don’t just start using them.

**Structure long proofs.** Long programs are usually broken into a hierarchy of smaller procedures. Long proofs are much the same. Facts needed in your proof that are easily stated, but not readily proved are best pulled out and proved in a preliminary lemma. Also, if you are repeating essentially the same argument over and over, try to capture that argument in a general lemma and then repeatedly cite that instead.

**Don't bully.** Words such as “clearly” and “obviously” serve no logical function. Rather, they almost always signal an attempt to bully the reader into accepting something which the author is having trouble justifying rigorously. Don't use these words in your own proofs and go on the alert whenever you read one.

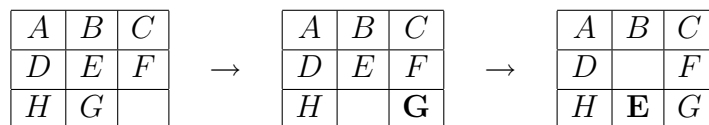
**Finish.** At some point in a proof, you'll have established all the essential facts you need. Resist the temptation to quit and leave the reader to draw the right conclusions. Instead, tie everything together yourself and explain why the original claim follows.

The analogy between good proofs and good programs extends beyond structure. The same rigorous thinking needed for proofs is essential in the design of critical computer systems. When algorithms and protocols only “mostly work” due to reliance on hand-waving arguments, the results can range from problematic to catastrophic. An early example was the Therac 25, a machine that provided radiation therapy to cancer victims, but occasionally killed them with massive overdoses due to a software race condition. More recently, a buggy electronic voting system credited presidential candidate Al Gore with *negative* 16,022 votes in one county. In August 2004, a single faulty command to a computer system used by United and American Airlines grounded the entire fleet of both companies— and all their passengers.

It is a certainty that we'll all one day be at the mercy of critical computer systems designed by you and your classmates. So we really hope that you'll develop the ability to formulate rock-solid logical arguments that a system actually does what you think it does.

## 2 The 8 Puzzle

Here is a puzzle. There are 8 lettered tiles and a blank square arranged in a  $3 \times 3$  grid. Any lettered tile adjacent to the blank square can be slid into the blank. For example, a sequence of two moves is illustrated below:



We'll only be considering legal moves in this lecture!

In the leftmost configuration shown above, the G and H tiles are out of order. We can find a way of swapping G and H so that they are in the right order, but then other letters

may be out of order. Can you find a sequence of moves that puts these two letters in the correct order, but returns every other tile to its original position? Some experimentation suggests that the answer is probably “no”, so let’s try to prove that.

We’re going to take an approach that is frequently used in the analysis of software and systems. We’ll look for an *invariant*, a property of the puzzle that is always maintained, no matter how you move the tiles around. If we can then show that putting the G and H tiles in the correct order would violate the invariant, then we can conclude that this is impossible.

Let’s see how this game plan plays out. Here is the theorem we’re trying to prove:

**Theorem 1.** *No sequence of legal moves transforms the board below on the left into the board below on the right.*

A	B	C
D	E	F
H	G	

A	B	C
D	E	F
G	H	

We’ll build up a sequence of observations, stated as lemmas. Once we achieve a critical mass, we’ll assemble these observations into a complete proof.

Define a *row move* as a move in which a tile slides horizontally and a *column move* as one in which a tile slides vertically. Assume that tiles are read top-to-bottom and left-to-right like English text, that is, the *natural order*, defined as follows:

1	2	3
4	5	6
7	8	9

So when we say two tiles are “out of order”, we mean that the larger letter precedes the smaller letter in this natural order.

Our difficulty is that one pair of tiles (the G and H) is out of order initially. An immediate observation is that row moves alone are of little value in addressing this problem:

**Lemma 2.** *A row move does not change the order of the tiles.*

*Proof.* A row move moves a tile from cell  $i$  to cell  $i + 1$  or vice versa. This tile does not change its order with respect to any other tile. Since no other tile moves, there is no change in the order of any of the other pairs of tiles.  $\square$

Let’s turn to column moves. This is the more interesting case, since here the order can change. For example, the column move shown below changes the relative order of the pairs  $(G, H)$  and  $(G, E)$ .

A	B	C
D	F	
H	E	G

→

A	B	C
D	F	G
H	E	

**Lemma 3.** *A column move changes the relative order of exactly two pairs of tiles.*

*Proof.* Sliding a tile down moves it after the next two tiles in the order. Sliding a tile up moves it before the previous two tiles in the order. Either way, the relative order changes between the moved tile and each of the two it crosses. The relative order between any other pair of tiles does not change.  $\square$

These observations suggest that there are limitations on how tiles can be swapped. Some such limitation may lead to the invariant we need. In order to reason about swaps more precisely, let's define a term referring to a pair of items that are out of order:

**Definition 1.** *A pair of letters  $L_1$  and  $L_2$  is an **inversion** if  $L_1$  precedes  $L_2$  in the alphabet, but  $L_1$  appears after  $L_2$  in the puzzle order.*

For example, in the puzzle below, there are three inversions: (D, F), (E, F) and (G, E).

A	B	C
F	D	G
E	H	

There is exactly one inversion (G,H) in the start state:

A	B	C
D	E	F
H	G	

There are no inversions in the end state:

A	B	C
D	E	F
G	H	

Let's work out the effects of row and column moves in terms of inversions.

**Lemma 4.** *During a move, the number of inversions can only increase by 2, decrease by 2 or remain the same.*

*Proof.* By Lemma 2, a row move does not change the order of the tiles; thus, in particular, a row move does not change the number of inversions.

By Lemma 3, a column move changes the relative order of exactly 2 pairs of tiles. There are three cases: If both pairs were originally in order, then the number of inversions after the move goes up by 2. If both pairs were originally inverted, then the number of inversions after the move goes down by 2. If one pair was originally inverted, and the other was originally in order, then the number of inversions stays the same (since the changing the former pair makes the number of inversions smaller by 1, and changing the latter pair makes the number of inversions larger by 1).  $\square$

We are almost there. If the number of inversions only change by 2, then what about the parity? That is, the “parity” of a number refers to whether the number is even or odd. For example, 7 and -5 have odd parity, and 18 and 0 have even parity.

Since adding or subtracting 2 from a number does not change the parity, we have the following corollary:

**Corollary 5.** *Neither a row nor a column move ever changes the parity of the number of inversions.*

Now we can bundle up all these observations and state an *invariant*, that is, a property of the puzzle that never changes, no matter how you slide the tiles around.

**Lemma 6.** *In every configuration reachable from the position shown below, the parity of the number of inversions is odd.*

row 1	A	B	C
row 2	D	E	F
row 3	H	G	

*Proof.* We use induction. Let  $P(n)$  be the proposition that after  $n$  moves from the above configuration, the parity of the number of inversions is odd.

*Base case:* After zero moves, exactly one pair of tiles is inverted (H and G), which is an odd number. Therefore,  $P(0)$  is true.

*Inductive step:* Now we must prove that  $P(n)$  implies  $P(n + 1)$  for all  $n \geq 0$ . So assume that  $P(n)$  is true; that is, after  $n$  moves the parity of the number of inversions is odd. Consider any sequence of  $n + 1$  moves  $m_1, \dots, m_{n+1}$ . By the induction hypothesis  $P(n)$ , we know that the parity after moves  $m_1, \dots, m_n$  is odd. By Corollary 5, we know that the parity does not change during  $m_{n+1}$ . Therefore, the parity of the number of inversions after moves  $m_1, \dots, m_{n+1}$  is odd, so we have that  $P(n + 1)$  is true.

Thus,  $P(n)$  implies  $P(n + 1)$  for all  $n \geq 0$ .

By the principle of induction,  $P(n)$  is true for all  $n \geq 0$ . □

The theorem we originally set out to prove is restated below. With this invariant in hand, the proof is simple.

**Theorem.** *No sequence of moves transforms the board below on the left into the board below on the right.*

A	B	C		A	B	C
D	E	G		D	E	F
<b>H</b>	<b>G</b>			<b>G</b>	<b>H</b>	

*Proof.* In the target configuration on the right, the total number of inversions is zero, which is even. Therefore, by Lemma 6, the target configuration is unreachable. □

Could we get the letters in order if the initial location of the blank were in a different place? The answer is no, but why not? Because the parity of the number of inversions would still be even.

This kind of puzzle was originally invented by Sam Lloyd in 1874. The original version was  $4 \times 4$ , and called the *15 puzzle*. It was very popular in its day. Somewhat like the Rubik's cube, but here there is no solution!

Turns out the proof that you can't solve the  $4 \times 4$  version is trickier than the  $3 \times 3$  version. In this case, you want to get from the first configuration to the second:

1	2	3	4
5	6	7	8
9	10	11	12
13	15	14	

→

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

Where does the proof from the  $3 \times 3$  version break down? In the  $4 \times 4$  case, a column move changes the order of 3 pairs! So the parity can change. The proof can still be found by finding a more complicated invariant.

If you ever played with Rubik's cube, you know that there is no way to rotate a single corner, swap two corners, or flip a single edge. All these facts are provable with invariant arguments like the one above. In the wider world, invariant arguments are used in the analysis of complex protocols and systems, often to show that you don't get into a really bad state. For example, in analyzing the software and physical dynamics a nuclear power plant, one might want to prove an invariant to the effect that the core temperature never rises high enough to cause a meltdown.

### 3 Unstacking

Here is another wildly fun 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
10		
5	5	25 points
5	3 2	6
4	3 2 1	4
2	3 2 1 2	4
2	2 2 1 2 1	2
1	2 2 1 2 1 1	1
1	1 2 1 2 1 1 1	1
1	1 1 1 1 2 1 1 1 1	1
1	1 1 1 1 1 1 1 1 1 1	1
total score		= 45 points

Can you find a better strategy?

### 3.1 Strong Induction

We'll analyze the unstacking game using a variant of induction 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  $n \in \mathbb{N}$ ,  $P(0), P(1), \dots, P(n)$  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  $P(0), P(1), \dots, P(n - 1)$ , and  $P(n)$  are *all* true when you go to prove  $P(n + 1)$ . These extra assumptions can only make your job easier.

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 can also be proved with ordinary induction. However, an appeal to the strong induction principle can make some proofs a bit simpler. On the other hand, if  $P(n)$  is easily sufficient to prove  $P(n + 1)$ , then use ordinary induction for simplicity.

### 3.2 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 7.** *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  $P(1), \dots, P(n - 1)$  imply  $P(n)$  for all  $n \geq 2$  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  $P(1), \dots, P(n - 1)$  imply  $P(n)$  for all  $n \geq 2$ . So assume that  $P(1), \dots, P(n - 1)$  are all true and that we have a stack of  $n$  blocks. The first move must split this stack into substacks with sizes  $k$  and  $n - k$  for some  $k$  strictly between 0 and  $n$ . 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:

$$\begin{aligned}
 \text{total score} &= (\text{score for 1st move}) \\
 &\quad + (\text{score for unstacking } k \text{ blocks}) \\
 &\quad + (\text{score for unstacking } n - k \text{ blocks}) \\
 &= k(n - k) + \frac{k(k - 1)}{2} + \frac{(n - k)(n - k - 1)}{2} \\
 &= \frac{2nk - 2k^2 + k^2 - k + n^2 - nk - n - nk + k^2 + k}{2} \\
 &= \frac{n(n - 1)}{2}
 \end{aligned}$$

The second equation uses the assumptions  $P(k)$  and  $P(n - k)$  and the rest is simplification. This shows that  $P(1), P(2), \dots, P(n)$  imply  $P(n + 1)$ .

Therefore, the claim is true by strong induction. □

### 3.3 Postage Stamps

Let's see an example of how strong induction can be used to solve a problem:

*Given an unlimited supply of 3 cent and 5 cent stamps, what postages are possible?*

Let's first try to guess the answer before we try to prove it. Let's begin filling in a table that shows the values of all possible combinations of 3 and 5 cent stamps. The column heading is the number of 5 cent stamps and the row heading is the number of 3 cent stamps.

	0	1	2	3	4	5	...
0	0	5	10	15	20	25	...
1	3	8	13	18	23	...	
2	6	11	16	21	...		
3	9	14	19	24	...		
4	12	17	22	...			
5	15	20	...				
...	...	...					

So far, the values that are missing are 1,2,4,7. Why don't we try to prove the following wild claim:

**Claim 8.** *For all  $n \geq 8$ , it is possible to produce  $n$  cents of postage from 3¢ and 5¢ stamps.*

First, let's discuss how we would go about proving such a crazy claim. Let's try a proof by strong induction. The induction hypothesis will be

$$P(n) \equiv \text{if } n \geq 8, \text{ then } n\text{¢ postage can be produced using 3¢ and 5¢ stamps} \quad (1)$$

For the base case, should we start with  $P(0)$ ? No,  $P(0)$ , won't be interesting because  $P(n)$  is *vacuously* true for all  $n < 8$ . Let's start with the base case of  $P(8)$ : Since  $8 = 3 + 5$ , we are all set.

In the inductive step we have to show how to produce  $n + 1$  cents of postage, assuming the strong induction hypothesis that we know how to produce  $k$ ¢ of postage for all values of  $k$  such that  $8 \leq k \leq n$ . A simple way to do this is to let  $k = n - 2$  and produce  $k$ ¢ of postage; then add a 3¢ stamp to get  $n + 1$  cents.

Are we done? No – there is a pitfall in this method. The inductive step is still fine as long as  $n \geq 10$ . However, if  $n + 1$  is 9 or 10, then we can not use the trick of creating  $n + 1$  cents of postage from  $n - 2$  cents and a 3 cent stamp. In these cases,  $n - 2$  is less than 8. None of the strong induction assumptions help us make less than 8¢ postage. Fortunately, making  $n + 1$  cents of postage when  $n + 1 = 9, 10$  can be easily done directly.

Here is the full proof:

*Proof.* The proof is by strong induction. The induction hypothesis,  $P(n)$ , is given by (1).

**Base case:**  $n = 8$ .  $P(8)$  is true since  $8 = 3 + 5$ .

**Inductive step:** In the inductive step, we assume that it is possible to produce postage worth  $8, 9, \dots, n$  cents in order to prove that it is possible to produce postage worth  $n + 1$  cents.

There are three cases:

1.  $n + 1 = 9$ :  $P(n + 1)$  holds by using three 3¢ stamps.
2.  $n + 1 = 10$ :  $P(n + 1)$  holds by using two 5¢ stamps.
3.  $n + 1 > 10$ : We have  $n \geq 10$ , so  $n - 2 \geq 8$  and by strong induction we may assume we can produce exactly  $n - 2$  cents of postage. With an additional 3¢ stamp we can therefore produce  $n + 1$  cents of postage.

So in every case,  $P(0), P(1), \dots, P(n - 1)$  and  $P(n)$  all true implies that  $P(n + 1)$  is true. By strong induction, we have concluded that  $P(n)$  is true for all natural numbers  $n \geq 8$ .  $\square$

### 3.4 A Wrong Proof

Now let's see an example of how *not* to use strong induction.

**Theorem 9 (Not!).** *All natural numbers are even.*

*Proof:(???)* The proof is by strong induction. The induction hypothesis  $P(n)$  is

$$P(n) \equiv n \text{ is even} \tag{2}$$

**Base case:**  $n = 0$  is even, so  $P(0)$  is true.

**Inductive step:**  $\forall n \geq 0$ , assume  $P(0), P(1), \dots, P(n)$  to prove  $P(n + 1)$ , i.e., assume that  $0, 1, \dots, n$  are even. Consider  $n + 1$ .  $P(n)$  tells us that  $n$  is even, and  $P(1)$  tells us that 1 is even. Thus,  $n + 1$  is even, since a sum of two even numbers is even. Thus  $P(n + 1)$  is true. By strong induction, we have concluded that  $P(n)$  is true for all natural numbers.  $\square$

*Where is the bug?*  $P(0)$  is true, so the base case is true. For the inductive step, this proof shows that  $P(1)$  implies  $P(2)$ ,  $P(1)$  and  $P(2)$  imply  $P(3)$ , and that for all  $n \geq 1$ ,  $P(1), \dots, P(n)$  imply  $P(n + 1)$ . But, we are missing the key "domino" – that  $P(0)$  implies  $P(1)$ : that is, when  $n = 0$ , we are making a circular argument by assuming  $P(1)$  is true in order to prove that  $P(1)$  is true! In this case,  $P(1)$  is false, so the induction fails.

The moral of the story is that you should always check the edge cases!