PROOF METHODS

Required reading:

- "Understanding Induction",
  in the course handbook.

Additional reference:

- "How to read and do proofs", by Solow.
- "How to Prove It", by Velleman.

Motivation

As computer scientists, we need to know things like:

- This chip will behave according to the specifications no matter what sequence of events occurs.
- This technique for finding the shortest route from location A to location B works no matter what the map is.
- A graph with n nodes can't have more than $\frac{n^2-n}{2}$ edges, no matter what n is.

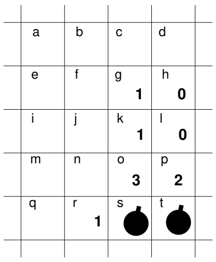
These are very strong statements! How many different sequences of events / maps / integers n are there?

The only way to be sure of such a statement is to prove it.

Let's review some basics about doing proofs.

Minesweeper

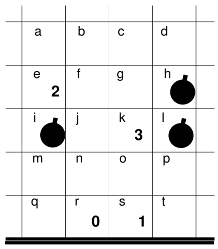
Problem: Figure out where the bombs are on a given grid.



Legend:

- Number: No bomb here, but tells how many adjacent squares have bombs. Corners count.
- Blank: We have no information.
- The grid continues in all directions, unless a wall is shown (a double thick line).

Game example I



Question: If j has a bomb, what do we know about t?

Prove: if j has a bomb then t has a bomb.
Proof: by conditional proof.

> Assume j has a bomb.
>
> > k has 3 bombs adjacent,
> > according to the diagram.
> > j, l, and h are adjacent to k.
> > They all have bombs; that makes 3.
> > n, o, and p are also adjacent to k.
> > So n, o, and p have no bombs.
> > s has 1 bomb adjacent,
> > according to the diagram.
> > r, n, o, p, and t are adjacent to s.
> > r, n, o, and p have no bombs.
> > So t has a bomb.
>
> So if j has a bomb then t has a bomb.

Conditional Proof

Prove: $A \Rightarrow B$
Proof: Assume that A is true.

> *Using the assumption as if it were true, prove B.*
>
> Therefore $A \Rightarrow B$.

You can make assumptions within assumptions. (Then indenting really helps.)

What does $A \Rightarrow B$ mean?

It's like a promise of guarantee: As long as A is true, B must also be true.

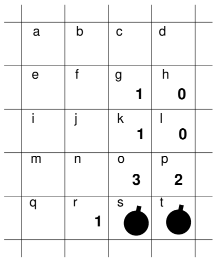
So if A is false, the promise hasn't been broken (no matter what B is), and so $A \Rightarrow B$ is true in that case.

This does not correspond to what "implies" means in English! Don't let that confuse you. Just remember that $A \Rightarrow B$ is defined to mean the same thing as $\neg A \vee B$.

Here is a complete specification:

A	B	$A \Rightarrow B$	$\neg A \vee B$
false	false	true	true
false	true	true	true
true	false	false	false
true	true	true	true

Game example II



Question: Does square f have a bomb?

Prove: f doesn't have a bomb.
Proof: by contradiction.

Assume f *does* have a bomb.

k has 1 bomb adjacent,
according to the diagram.
f is adjacent to k and has a bomb.
So j and n do not have bombs.
So o has 2 bombs adjacent.
But o has 3 bombs adjacent,
according to the diagram.
Contradiction!

So f does not have a bomb.

We say, among other things, that "j and n do not have bombs". We are not claiming that this is true; just that it is true *under the assumption that f does have a bomb*.

To make this clear, it is helpful to indent for the duration of an assumption.

In the proof we also say "Assume f does have a bomb". We are not claiming that f actually does have a bomb; we are just assuming it temporarily, *for the sake of argument*.

Proof by Contradiction

Prove: Q
Proof: Assume that Q is false.

Prove that some contradiction follows,
e.g., that $x > 0$ and $x \not> 0$.

Therefore Q must be true.

Instead of proving that a thing is true, we prove that it can't be false. Sometimes this is easier.

The flip side: Instead of proving that a thing is false, we prove that it can't be true:

Prove: Q is false.
Proof: Assume that Q is true.

Prove that some contradiction follows,
e.g., that $x > 0$ and $x \not> 0$.

Therefore Q must be false.

Nesting Proofs

You can nest one proof technique inside another.

Question: What conclusion can be drawn from this proof?

Proof: Assume that $S1$ is true.

Assume that $S2$ is true.

⋮

Derive a contradiction.

Game example III

a	b 0	c 0	d 0
e	f 1	g 1	h 1
i 1	j	k	l 2
m	n	o 1	p
q	r	s	t

Question: Does n have a bomb?

Prove: n does not have a bomb.
Proof: by cases.
g has 1 bomb adjacent,
according to the diagram.
All adj. squares except j and k lack bombs.
So either (j has a bomb) or (k has a bomb).
Assume (j has a bomb).
o has 1 bomb adjacent.
j and n are both adjacent to o.
So n does not have a bomb.
Assume (k has a bomb).
o has 1 bomb adjacent.
k and n are both adjacent to o.
So n does not have a bomb.
So n does not have a bomb.

Proof by Cases

Prove: Q
Proof: Either C_1, or C_2, ..., or C_n is true.
Prove that $C_1 \Rightarrow Q$
Prove that $C_2 \Rightarrow Q$
⋮
Prove that $C_n \Rightarrow Q$
Therefore Q is true in *any* case.

If we choose the cases well, it will be easier to prove several sub-proofs than to prove the original statement.

The cases must be exhaustive, or the "either" statement will be false, and the proof invalid.

Exercise. Prove: for all integers n, $|n| \geq n$, using the two cases (1) $n \geq 0$, and (2) $n < 0$.

Options

There are many ways to express the same proof. For a more sophisticated reader, we may bunch many steps together.

There often are also many different ways to prove a statement.

Prove: n does not have a bomb.
Proof: by contradiction. [Alternative proof]
Assume n has a bomb.
o has 1 bomb adjacent,
according to the diagram.
n is adjacent to o and has a bomb.
So none of j, k, l, p, t, s, or r has one.
So g has no bomb's adjacent.
But g has 1 bomb adjacent,
according to the diagram.
Contradiction!
So n does not have a bomb.

Tackling a Proof

We start with a set (possibly empty) of assumptions, which we treat as facts, and a statement to be proven — our goal.

We enlarge the facts until we reach our goal.

We can also reason backwards, like this: "If I could prove X, then I would be done (because X implies that my goal must be true). So now I have a new goal, X."

Proof critic

A proof is like a court case:

- Each step must be justified (or trivial to justify).
- Otherwise, your argument is invalid and you'll lose your case — even if what you're arguing is true!

Imagine a proof critic sitting on your shoulder.

Proving Statements of the Form "for all $x, \ldots$"

You have probably seen many proofs like this:

$$\begin{aligned}
&(x-1)(x+3)(x+6) \\
&\quad = (x^2+2x-3)(x+6) \\
&\quad = x^3+2x^2-3x+6x^2+12x-18 \\
&\quad = x^3+8x^2+9x-18
\end{aligned}$$

This is actually short for the following:

Prove: for all real numbers x,
$(x-1)(x+3)(x+6) = x^3+8x^2+9x-18$.

Proof: Let x be $\boxed{\text{any}}$ real number.

$$\begin{aligned}
&(x-1)(x+3)(x+6) \\
&\quad = (x^2+2x-3)(x+6) \\
&\quad = x^3+6x^2+2x^2+12x-3x-18 \\
&\quad = x^3+8x^2+9x-18
\end{aligned}$$

Therefore, for all real numbers x,
$(x-1)(x+3)(x+6) = x^3+8x^2+9x-18$.

Universal Generalization

Prove: For all x of some sort, $P(x)$ is true.
Proof:

Let x be anything of that sort.
Making no *other* assumptions about x, prove $P(x)$ is true.
So for all x of that sort, $P(x)$ is true.

Since we haven't assumed anything about the value of x in doing the proof, the steps must be valid for *all* possible values of x.

This technique is called universal generalization because it allows us to make a generalization that applies to *all* x of some sort.

("For all", often written $\forall$, is known as a "universal quantifier").

Scope of Variables

Just like variables in a program, variables in a proof have scope: the part of the proof within which they can be referred to.

Universal Generalization "introduces" a new variable and begins its scope.
(This is analogous to a variable declaration.)

Programming languages have syntax rules that define where a scope ends (e.g., at the matching "}"). In a proof, we can indicate where a scope ends using indentation, or careful wording.

Note that the same name may be used for two *different* variables at different points within the same proof, just like the many integers `i` in a program.

Example Proof

Prove: For all numbers x, $x^2 = 3x$ implies $x = 3$.
Proof:

$x^2 = 3x$.
Therefore $x = 3$. (Divide both sides by x.)
Q.E.D.

Be a proof critic and ask yourself these questions:

- What is the scope of every variable in this proof? Does that show you any weaknesses in the proof?
- What proof technique(s) are being used? Rewrite the proof to show any techniques explicitly.
- Is the proof valid? If not, fix it.

Proof Exercises

These require universal Generalization, plus some of the other techniques.

1. **Prove:** for all integers $n > 2$, $(n^2 - 3n + 2 > 0)$.
2. **Prove:** for all integers n, $\left\lfloor \frac{n+1}{2} \right\rfloor = \left\lceil \frac{n}{2} \right\rceil$.
3. **Prove:** for all integers n, n^2 is even $\Rightarrow n$ is even.

Note: $\lfloor x \rfloor$ is the largest integer $\leq x$, and $\lceil x \rceil$ is the smallest integer $\geq x$.

Sometimes U.G. isn't enough

Universal generalization gives you very little to "lean on". Sometimes you get stuck.

Induction is another technique for proving statements that are universally quantified, and it gives you a lot to lean on.

Dominoes Analogy

Suppose I have an infinite number of dominoes lined up. Let's say I can convince you that every domino is 6cm tall and is 4cm away from the domino before it. Say also that I convince you I'm going to knock over the 1st domino.

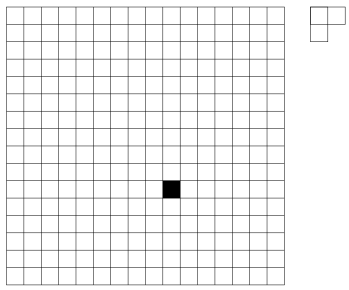
Do you believe that every domino will eventually fall?

If so, you have made an infinite number of conclusions. Did you have to think through each one?

Tiling a Kitchen

My kitchen is 16 by 16 units square. The floor is empty except for one unit square that is taken up by the fridge.

Can you tile it using the shape of tile shown (and with no overlapping tiles)?



What if I move the fridge somewhere else?
What if the kitchen is 2^n by 2^n (and I'm not telling you what n is, other than that it's at least 0)?!

Proof by Induction

Prove: For all $n \geq 1$, $S(n)$ true.
Proof: by induction on n.

Base Case: Prove $S(1)$ is true
Let $k \geq 1$ be an arbitrary integer.
Induction Hypothesis:
Assume $S(k)$ is true.
Induction Step:
Prove that $S(k+1)$ must also be true.

Conclusion: For all $n \geq 1$, $S(n)$ is true.

We now have two things to prove (base case and induction step) but each is easier to prove than the original statement.

- The base case is usually trivial
- In the induction step, we can assume $S(k)$.

Understanding The Induction Step

To understand why the conclusion is justified, one must understand what the induction part (everything but the base case) proves.

Let $k \geq 1$ be an arbitrary integer.

Assume $S(k)$.

$\vdots$

$S(k+1)$.

Thus, $\boxed{}$

Thus, $\boxed{}$

So that's what the induction part proves. What good is that? It's like a crank ...

The "Process" of Induction

Already proven facts:

(I) S(1)

(II) For any k >= 1, S(k) implies S(k+1)

Facts that follow from what was proven:

from (I): S(1)

substitute k=1 into (II): S(1) implies S(2)

S(2)

substitute k=2 into (II): S(2) implies S(3)

S(3)

substitute k=3 into (II): S(3) implies S(4)

S(4)

substitute k=4 into (II): S(4) implies S(5)

S(5)

∘
∘
∘

Example of Proof by Induction

In a **complete** graph, every node is connected to every other node.

Let $C(n)$ be the number of edges in a complete graph with n nodes.

Prove: For all $n \geq 0$, $C(n) = \frac{n^2-n}{2}$.

Base Case: Prove $C(0) = \frac{(0^2-0)}{2}$.
A complete graph with no nodes has no edges,
so $C(0) = 0$. Also, $\frac{(0^2-0)}{2} = \frac{0}{2} = 0$.
Thus, $C(0) = \frac{(0^2-0)}{2}$.

Let $k \geq 0$ be an arbitrary integer.
Induction Hypothesis: Assume $C(k) = \frac{(k^2-k)}{2}$.

Induction Step: Prove $C(k+1) = \frac{((k+1)^2-(k+1))}{2}$.
Consider a complete graph with $k+1$ nodes.
If we take out one node (and the k edges that connect it to each of the other k nodes), we have a complete graph with k nodes.
That graph has $\frac{(k^2-k)}{2}$ edges, by our induction hyp.
So our original complete graph with $k+1$ nodes has all those edges, plus the k we took out.
So the total number of edges

$$= \frac{(k^2-k)}{2} + k = \frac{(k^2-k)+2k}{2} = \frac{k^2+k}{2} = \frac{(k^2+2k+1)-(k+1)}{2} = \frac{(k+1)^2-(k+1)}{2}.$$

Thus $C(k+1) = \frac{((k+1)^2-(k+1))}{2}$.

Conclusion: For all $n \geq 0$, $C(n) = \frac{(n^2-n)}{2}$.

Scope of variables in this proof

Exercise: Circle every variable in the proof we just saw. Identify the scope of each variable.

Notice that we used variable name n in the statement to be proven and the conclusion, but we switched to k for the induction part.

The proof would be equally valid if we'd used n there too, but it would be a *different* n. We changed variable names to emphasize this.

Aside: Recurrence Relations

Here is one way to define the value of $C(n)$:
$C(0) = 0$
$C(n) = C(n-1) + (n-1)$ for all $n \geq 1$.

This makes sense because we can create a complete graph with n nodes by taking a complete graph with $n-1$ nodes, adding a new node to it, and hooking it in to the other $n-1$ nodes using $n-1$ edges
(as we saw in the proof).

It is not a circular definition because it is grounded by the definition of $C(0)$.

This is an example of a **recurrence relation**: a set of equations or inequalities that describe a mathematical function in terms of its value on smaller inputs.

Another recurrence relation

Suppose I have a chocolate cake, and each day I plan to eat half of it what I have left. Let $L(n)$ denote how much of the cake I have left on the morning of day n.
$L(1) = 1$
$L(n) = \frac{L(n-1)}{2}$, for all $n \geq 2$.

Recurrence relations are convenient for expressing many things in computer science.

They are also unsatisfying: To get a feel for $L(n)$, you'd have to plug in some values.

For this reason, we try to "solve" a recurrence relation: find an equivalent definition that is not self-referential.

What is the solution for $L(n)$?

In later courses, you'll learn about uses for recurrence relations, and general techniques for solving them.

Other Sorts of Induction

There are many variations on induction, including:

- using a base value other than 1,
- using more than one base case,
- using increments other than 1, and
- "strong induction", wherein we use a stronger, but equally valid, induction hypothesis. Strong induction is not more "powerful" than regular induction; i.e., it is not capable of proving statements that cannot be proven using regular induction.

See "Understanding Induction" for discussion and examples of these sorts of induction.

Also, be sure to read the final section, on common flaws in inductive proofs.

Writing Induction Proofs

Which type to use?

Here's a template for the structure of an induction proof:

> **Base Case**: Prove $S(\boxed{})$.
>
> Let $k \geq \boxed{}$ be an arbitrary integer.
>
> **Induction Hypothesis**: Assume $\boxed{}$ is true.
>
> **Induction Step**: Prove $S(\boxed{})$ is true.
>
> **Conclusion**: $S(n)$ is true for all $n \geq \boxed{}$.

The boxes show the decisions you have to make about a proof's structure.

How do you make those decisions?

How to pick a structure

1. Fill in the Conclusion, as desired.
2. Try to convince yourself (informally) that S is true for some arbitrary value. Would it help you to know that:
   - it's true for a value 1 smaller?
     Then make IH 1 smaller than IS.
   - it's true for a value 2 smaller?
     Then make IH 2 smaller than IS.
   - it's true for *all* smaller values?
     Then use strong induction.
   - (If no smaller versions help, you are not doing induction!)

   Fill in the Induction Hypothesis and Induction Step accordingly.
3. (Try using just one Base Case.)
   Fill in the Base Case with the smallest value claimed in the Conclusion.
4. Fill in the "$k \geq$" part with that same value, to connect with the IH.

Now check the validity of your structure (see below). If invalid, go back and redo steps 3 and 4 to correct it.

Eventually you can abandon this technique; you will often just know what structure will be valid and will work for your S.

Checking that your structure is valid

It is easy to write a "proof" whose structure is invalid, *i.e.*, that doesn't prove the Conclusion.

To check the validity of your induction proof structure:

- Fill in the "crank" that your induction part proves:
  For any $k \geq \boxed{}$ $\boxed{}$ implies $\boxed{}$.
- List your base cases.
- Confirm that you can "turn the crank" to get every statement claimed by your Conclusion.

Exercises: Which of these proof structures is valid for proving that

$S(n)$ is true for all $n \geq 0$?

Base Case: Prove $S(0)$ is true.
Let $k > 0$ be an arbitrary integer.
Induction Hypothesis: Assume $S(k)$ is true.
Induction Step: Prove $S(k+1)$ is true.

Base Case: Prove $S(0)$ is true.
Let $k \geq 1$ be an arbitrary integer.
Induction Hypothesis: Assume $S(k-1)$ is true.
Induction Step: Prove $S(k)$ is true.

Base Case: Prove $S(0)$ is true and $S(1)$ is true.
Let $k \geq 0$ be an arbitrary integer.
Induction Hypothesis: Assume $S(k)$ is true.
Induction Step: Prove $S(k+2)$ is true.

Base Case: Prove $S(0)$ is true.
Induction Hyp: Assume $S(k)$ is true
for all integers $k \geq 0$.
Induction Step: Prove $S(k+1)$ is true.

Base Case: Prove $S(0)$ is true.
Let $k \geq 0$ be an arbitrary integer.
Induction Hyp: Assume $S(j)$ is true
for all integers j where $0 \leq j \leq k$.
Induction Step: Prove $S(k+1)$ is true.

Checking that your content is valid

Of course the proof structure alone is not a proof. The meat is in:

- the sub-proof that the base case(s) are true, and
- the sub-proof of the Induction Step.

Remember your proof critic when checking these.

What's wrong with this?

Statement: Let $S(n)$ represent the statement
"Any set of n computers are all the same colour."
Prove: $S(n)$ is true for all $n \geq 1$.

Base Case: Prove $S(1)$.
Any single computer is the same colour as itself.
Thus, $S(1)$ is true.

Let $k \geq 1$ be an arbitrary integer.
Induction Hypothesis: Assume $S(k)$ is true,
that is, any set of k computers are all the same colour.

Induction Step: Prove $S(k+1)$ is true.
Consider any set of $k+1$ computers.
Remove any one computer from the set, and there are k left.
By the induction hypothesis, these must all be the same colour.
This is true no matter *which* one we remove.
So all $k+1$ computers in this set are the same colour.
Thus for any set of $k+1$ computers, they all are the same colour.
Thus $S(k+1)$ is true.

Conclusion: $S(n)$ is true for all $n \geq 1$.
In other words, all computers are the same colour.

Induction isn't always the answer

We've spent a lot of time on induction. That doesn't mean it is the ultimate tool for all proofs. Sometimes a proof can be done best using another technique.

Exercise: Prove the following fact two ways: first using induction, and then using universal generalization.

$$\forall n > 2, \quad n^2 > 4.$$

The induction proof can be done correctly, but since it is more complex it introduces more ways to make a mistake.