Massachusetts Institute of Technology | Handout 10
6.042J/18.062J: Mathematics for Computer Science | February 13, 1998
Professor Albert R. Meyer

Notes for Recitation 2 (2/13/98)

We'll look at several more examples of proof by induction. Aside from the induction method, these examples demonstrate some Math that comes up a lot in Computer Science: sets, Boolean variables, and modular arithmetic. After that, we'll look at some variations on induction and wrap up with some team problems.

1 Examples of Proof by Induction

Theorem 1.1 *(Distributive Law) Let $A, B_1, B_2, \ldots, B_n$ be sets where $n \geq 1$. Then*

$$A \cap (B_1 \cup B_2 \cup \ldots \cup B_n) = (A \cap B_1) \cup \ldots \cup (A \cap B_n).$$

We saw a special case of this theorem last week in tutorial. In particular, we looked at the case of $n = 2$, i.e. $A \cap (B_1 \cup B_2) = (A \cap B_1) \cup (A \cap B_2)$. This was proven by showing that each set contains the other. We will actually use this result in the proof below of the more general theorem.

Here is the game plan for the proof. We'll take the induction hypothesis directly from the theorem statement. This means that in the inductive step, we will have to show that $A \cap (B_1 \cup B_2 \cup \ldots \cup B_n \cup B_{n+1})$ is equal to $(A \cap B_1) \cup \ldots \cup (A \cap B_n) \cup (A \cap B_{n+1})$. We can almost do this; by induction, we can transform the almost-as-big expression $A \cap (B_1 \cup B_2 \cup \ldots \cup B_n)$ into something good. All that remains is to handle that extra term, B_{n+1}. We'll do this by using the $n = 2$ result from last week.

This basic strategy, proving something for $n = 2$ and then generalizing with induction, is very common. In fact, we'll use exactly the same approach in the next example as well.

Proof. The proof is by induction on n.

Induction hypothesis: Let $P(n)$ be the predicate

$$A \cap (B_1 \cup B_2 \cup \ldots \cup B_n) = (A \cap B_1) \cup \ldots \cup (A \cap B_n).$$

Base case: $P(0)$ is true because both sides of the equation are equal to A.

Inductive step: For all $n \geq 0$, assume

$$A \cap (B_1 \cup B_2 \cup \ldots \cup B_n) = (A \cap B_1) \cup \ldots \cup (A \cap B_n)$$

in order to prove

$$A \cap (B_1 \cup B_2 \cup \ldots \cup B_n \cup B_{n+1}) = (A \cap B_1) \cup \ldots \cup (A \cap B_n) \cup (A \cap B_{n+1}).$$

Let $C = B_1 \cup B_2 \cup \ldots \cup B_n$. Then we can rewrite the left side of $P(n+1)$ as follows:

$$\begin{aligned}
A \cap (B_1 \cup B_2 \cup \ldots \cup B_n \cup B_{n+1}) &= A \cap (C \cup B_{n+1}) \\
&= (A \cap C) \cup (A \cap B_{n+1}) \\
&= (A \cap (B_1 \cup B_2 \cup \ldots \cup B_n)) \cup (A \cap B_{n+1}) \\
&= (A \cap B_1) \cup \ldots \cup (A \cap B_n) \cup (A \cap B_{n+1})
\end{aligned}$$

In the first step, we replace a bunch of terms with the variable C. This simplifies the original expression enough so that we can use the result from last week, which gives the second equality. In the third step, we substitute that bunch of terms back for C so that in the last step, we can use the induction hypothesis. This proves that for all $n \geq 0$, $P(n) \Rightarrow P(n+1)$. ■

DeMorgan's Law for boolean variables is

$$\overline{x \vee y} = \overline{x} \wedge \overline{y}$$

where x and y are Boolean variables. We can prove this with the truth table method:

x	y	$\overline{x \vee y}$	$\overline{x} \wedge \overline{y}$
T	T	F	F
T	F	F	F
F	T	F	F
F	F	T	T

By inspection of the truth table, $\overline{x \vee y} = \overline{x} \wedge \overline{y}$ in all cases.

The next theorem generalizes DeMorgan's Law to an arbitrary number of Boolean variables in much the same way that the previous theorem generalized the distributive law to an arbitrary number of sets.

Theorem 1.2 *(DeMorgan's Law) For Boolean variables $x_1, x_2, \ldots, x_n$ and $n \geq 1$,*

$$\overline{(x_1 \vee x_2 \vee \ldots \vee x_n)} = \overline{x_1} \wedge \overline{x_2} \wedge \ldots \wedge \overline{x_n}.$$

We will follow the same strategy as before. $P(n)$ and $P(n+1)$ are very similar except that $P(n+1)$ contains one more variable. We will bridge this gap between $P(n)$ and $P(n+1)$ with the two-variable special case of the theorem proven above.

Proof. The proof is by induction on n.

Induction hypothesis: Let $P(n)$ be the predicate:

$$\overline{(x_1 \vee x_2 \vee \ldots \vee x_n)} = \overline{x_1} \wedge \overline{x_2} \wedge \ldots \wedge \overline{x_n}$$

Base case: The predicate $P(1)$ true because both sides are equal to $\overline{x_1}$.

Inductive step: For all $n \geq 1$ assume

$$\overline{(x_1 \vee x_2 \vee \ldots \vee x_n)} = \overline{x_1} \wedge \overline{x_2} \wedge \ldots \wedge \overline{x_n}$$

in order to prove

$$\overline{(x_1 \vee x_2 \vee \ldots \vee x_n \vee x_{n+1})} = \overline{x_1} \wedge \overline{x_2} \wedge \ldots \wedge \overline{x_n} \wedge \overline{x_{n+1}}.$$

Let $y = (x_1 \vee x_2 \vee \ldots \vee x_n)$. Then we can rewrite the left side of $P(n+1)$ as follows:

$$\begin{aligned}
\overline{(x_1 \vee x_2 \vee \ldots \vee x_n \vee x_{n+1})} &= \overline{(y \vee x_{n+1})} \\
&= \overline{y} \wedge \overline{x_{n+1}} \\
&= \overline{x_1 \vee x_2 \vee \ldots \vee x_n} \wedge \overline{x_{n+1}} \\
&= \overline{x_1} \wedge \overline{x_2} \wedge \ldots \wedge \overline{x_n} \wedge \overline{x_{n+1}}
\end{aligned}$$

The rationale for each step exactly parallels the previous proof. In the first step, we replace a bunch of terms with the variable y. This simplifies the original expression so that we can use the two variable result. This gives the second equality. In the third step, we substitute the bunch of terms back for y. Finally, we use the induction hypothesis. This proves that for all $n \geq 0$, $P(n) \Rightarrow P(n+1)$. ∎

A Review of Modular Arithmetic

Recall the $\equiv$ mod notation from Rosen 2.3: $x \equiv y \pmod{q}$ iff $q|(x-y)$ for integers x, y, q. In words, this says x is "congruent" to y "mod" q if and only if q evenly divides $x - y$; another way to formulate this is to say that x and y have the same remainder after division by q. For example, $15 \equiv 3 \pmod{6}$ since 6 evenly divides $15 - 3 = 12$; both 15 and 3 have remainder 3 after division by 6.

While $x \equiv y \pmod{q}$ is a statement, like an equation, the similar-looking notation $a \bmod b$ stands for an integer quantity, namely, $a \bmod b$ is the remainder when a is divided by b. More precisely, this is the number r such that $a = bq + r$ where q is an integer and $0 \leq r < b$. The number b is required to be at least 1. For example, $15 \bmod 6$ is equal to 3 because $15 = 6 \cdot 2 + 3$. Now we have:

$$x \equiv y \pmod{q} \quad \text{iff} \quad [x \bmod q = y \bmod q].$$

Theorem 1.3 $\forall n \in \mathbb{N} : n^3 \equiv n \pmod{3}$

While the plan for writing an induction proof is clear, there is no rigid plan for finding such a proof. There are, however, a few good tricks to know. There are usually two difficulties in completing a proof by induction: choosing the right induction hypothesis $P(n)$ and completing the inductive step. (Sometimes proving the base case is also hard.) The most obvious candidate for an induction hypothesis is the theorem statement itself.

Method 1 *When searching for a proof by induction, first try to lift an induction hypothesis directly from the theorem statement.*

To complete the inductive step, we must prove that for all $n \geq 0$, $P(n) \Rightarrow P(n+1)$. Typically $P(n)$ and $P(n+1)$ are very similar statements, but with some small difference or gap. The challenge is to clearly identify this gap and then to bridge it with some extra facts. Working "backward" is often helpful.

Method 2 *To prove $P(n) \Rightarrow P(n+1)$, write down $P(n+1)$ and then use $P(n)$ to reduce this statement to a simpler assertion; this assertion is the extra fact needed to bridge the gap from $P(n)$ to $P(n+1)$ and complete the proof.*

We can apply these principles to obtain a proof by induction of the preceding theorem. To begin, we can lift the induction hypothesis directly from the theorem statement. Let $P(n)$ be the predicate $n^3 \equiv n \pmod{3}$. Second, we have to prove that for all $n \geq 0$, $P(n) \Rightarrow P(n+1)$. Using the definition of $\equiv \pmod{3}$, the predicate $P(n+1)$ is equivalent to $3 \mid (n+1)^3 - (n+1)$. Expanding terms in $P(n+1)$ gives $3 \mid n^3 + 3n^2 + 2n$. We can now use $P(n)$ to simplify $P(n+1)$ by subtracting off the quantity that $P(n)$ asserts is divisible by 3. This gives $3 \mid 3n^2 + 3n$. This obvious statement is the extra fact needed to bridge the gap from $P(n)$ to $P(n+1)$. In short, we can show that $P(n)$, together with the extra fact $3 \mid 3n^2 + 3n$, implies $P(n+1)$. Here is the concise proof:

Proof. The proof is by induction. Let $P(n)$ be the proposition $n^3 \equiv n \pmod{3}$. In the base case, the proposition is true because $n^3 = n = 0$ and $3 \mid (0 - 0)$. For $n \geq 0$, assume $3 \mid n^3 - n$ to prove $3 \mid (n+1)^3 - (n+1)$.

$$\begin{aligned}
& 3 \mid (n^3 - n) \\
\Rightarrow\ & 3 \mid (n^3 - n) + 3(n^2 + n) \\
\Rightarrow\ & 3 \mid n^3 + 3n^2 + 3n + 1 - n - 1 \\
\Rightarrow\ & 3 \mid (n+1)^3 - (n+1)
\end{aligned}$$

The first implication follows because $3(n^2+n)$ is divisible by 3. The remaining implications involve only rewriting the expression on the right. This proves that for all $n \in \mathbb{N}$, $P(n)$ implies $P(n+1)$, completing the proof. ■

The final concise proof might seem mysterious to anyone not privy to the scratchwork we did beforehand. For example, one might ask how we had the foresight to introduce the magic term $3(n^2+n)$. Of course, this was not foresight at all; we worked backward.

By the way, there is also a simple direct proof of this theorem w/o induction:

$$n^3 - n = n(n^2 - 1) = n(n+1)(n-1) = (n-1)n(n+1).$$

But $n-1, n$, and $n+1$ are three consecutive integers, so one of them must be divisible by 3. Hence, $3 \mid n^3 - n$. ■

2 Proving Statements with Two Variables

Sometimes the statement of a theorem involves two natural-valued variables. In such situations, panicking is not strictly necessary, but sometimes unavoidable. Here is an example:

Theorem 2.1 *For all n and k such that $n \geq k \geq 0$, $k!$ divides $n(n-1)(n-2)\ldots(n-k+1)$.*

First a notation clarification. If $k = 0$, then the expression $n(n-1)(n-2)\ldots(n-k+1)$ is a product with no terms, which we define to be 1. If $k = 1$, then the expression is n.

There are several ways to approach this problem. One way is to define a sort of two-dimensional array of predicates $Q(n,k)$. Then we could prove sufficient base cases and implications so that every predicate $Q(n,k)$ followed from some proven bases cases through a chain of proven implications. The theorem would then follow from a fancy variant of the induction axiom.

A second approach is to eliminate a variable and use ordinary induction. Eliminating a variable may not be easy. The next two theorems will demonstrate different ways that sometime work.

The first way is to introduce a quantifier into the induction hypothesis. In this example, we can use the hypothesis, "For all k such that $n \geq k \geq 0$, $k! \mid n(n-1)\ldots(n-k+1)$. If we prove this statement for all $n \geq 0$, then we have proven the theorem. Though the induction hypothesis contains two variables, we only need to induct on n; you never have to induct on a quantified variable. In effect, we will prove a whole row of $Q(n, k)$ predicates in each step of the proof.

Proof. The proof is by induction on n.

Induction hypothesis: Let $P(n)$ be the predicate that for all k such that $n \geq k \geq 0$, $k! \mid n(n-1)\ldots(n-k+1)$.

Base case: $P(0)$ is true because k is then restricted to be 0, and $0! = 1$ divides every integer.

Inductive step: Assume that for all k such that $n \geq k \geq 0$, $k! \mid n(n-1)\ldots(n-k+1)$ in order to prove that for all k such that $(n+1) \geq k \geq 0$, $k! \mid (n+1)n(n-1)\ldots(n-k+2)$.

There are three cases to check. If $k = 0$, then the statement is true because $0! \mid 1$. If $k = n+1$, then the statement is true because $(n+1)! \mid (n+1)n(n-1)\ldots(1)$.

All that remains is to prove that $k! \mid (n+1)n(n-1)\ldots(n-k+2)$ for k in the range $n \geq k > 0$. By induction we know:

$$\begin{aligned} &k! \mid n(n-1)\ldots(n-k+1) \\ &(k-1)! \mid n(n-1)\ldots(n-k+2) \end{aligned}$$

(We need to use these two facts, and this is the reason why $k = 0$ and $k = n+1$ are treated as special cases. If $k = n+1$, then the first fact does not follow by induction. If $k = 0$, then the second fact does not follow.)

If we multiply both sides of the second fact by k, and then add both facts together, we obtain:

$$\begin{aligned} & k! \mid n(n-1)\ldots(n-k+1) + kn(n-1)\ldots(n-k+2) \\ \Rightarrow\ & k! \mid ((n-k+1)+k)n(n-1)\ldots(n-k+2) \\ \Rightarrow\ & k! \mid (n+1)n(n-1)\ldots(n-k+2) \end{aligned}$$

This shows that for all $n \geq 0$, $P(n) \Rightarrow P(n+1)$ and completes the proof. ∎

The algebra in this proof is sneaky and one probably woudn't think of this proof right away. The key idea here is working with a quantifier in the induction hypothesis.

Sharing Chocolate

We are given a chocolate bar with $m \times n$ squares of chocolate and our task is to divide it into mn individual squares. We are only allowed to split one piece of chocolate at a time using a vertical or a horizontal break.

For example, suppose that the chocolate bar is 2×2. The first split makes two pieces, both 2×1. Each of these pieces requires one more split to form single squares. This gives a total of three splits. We would like to prove the following general result:

Theorem 2.2 *To divide up a chocolate bar with $m \times n$ squares, we need exactly $mn - 1$ splits.*

Once again, we have two variables. If one variable can be eliminated, then ordinary induction will suffice. We could try the previous technique of formulating an induction hypothesis with a quantifier.

But let's try a second approach. Intuitively, to break up a big chocolate bar, we need one split to make two pieces, and then we can break up the two pieces recursively. This suggests a proof using strong induction on the *size* of the chocolate bar. The size can be defined to be mn, the area measured in chocolate squares.

This gives a second trick for using ordinary induction to prove a theorem with two variables: apply induction on a *function* of the variables.

Proof. The proof is by strong induction on mn.

Inductive hypothesis: Let $P(k)$ be the predicate that a chocolate bar of size k requires $k-1$ splits.

Base case: In the base case, $P(1)$ is true because there is only a single square of chocolate and so $1 - 1 = 0$ splits are required.

Inductive step: For all $k \geq 1$, assume that a chocolate bar of size s requires $s - 1$ splits where $k \geq s \geq 1$ in order to prove that a chocolate bar of size $k + 1$ requires k splits.

First, we will have to break the chocolate bar of size $k + 1$ into two smaller pieces of size p and q where $p + q = k + 1$. Then, by strong induction, breaking up the two pieces will require $p - 1$ and $q - 1$ splits. The total number of splits is therefore $1 + (p - 1) + (q - 1) = p + q - 1 = (k + 1) - 1 = k$. ■

3 Team Problems

1. An n-player tournament, consists of some set of $n \geq 2$ elements called *players*, and, for every two distinct players, a specification that one of the players beats the other. That is, player p_1 beats player p_2 iff p_2 does not beat p_1, for all players $p_1 \neq p_2$, and no player beats him/herself.

   A sequence of players $p_1, p_2, \ldots, p_k$ (for some $k \leq n$), such that player p_i beats player p_{i+1} for $1 \leq i < k$ is called a *ranking* of these players. If also player p_k beats player p_1, the ranking is called a *k-cycle*.

(a) Prove that in every tournament, either there is a "champion" player that beats every other player, or there is a 3-cycle.

Solution: (Proof by induction.) The induction hypothesis $P(n)$, is "In an n-player tournament, either there is a player that beats every other player, or there is a 3-cycle."

Base case $n = 2$: By definition of a 2-player tournament, one of the players beats the only other player and therefore is the champion.

Inductive step: Assume that $P(n)$ is true in order to show that $P(n+1)$ is true.

Consider an $n+1$-player tournament. If there is a champion, we are done. So we may assume that every player is beaten by some other player. Now remove any player, say player p, and treat the remaining n players as an n-player tournament. There are two cases:

1) Every player in the n-player tournament is beaten by some other player in the tournament.

2) There is some champion in the n-player tournament who beats every other player in the n-player tournament.

In case (1), we have by induction hypothesis a 3-cycle of players in the n-player tournament. Since a 3-cycle in the n-player tournament will also be a 3-cycle in the full $n+1$-player tournament, we are OK in this case.

In case (2), there is some "champion", q, in the n-player tournament. Since everybody in the $(n+1)$-player tournament is beaten by somebody, q must be beaten by a player not in the n-player tournament, that is, p beats q. For the same reason, there is some player r that beats p. Since r is in the n-player tournament, r is beaten by the champion q. So we have found a 3-cycle p, q, r.

(b) A *consistent ranking* is a sequence $p_1, p_2, \ldots, p_n$ of all n players in the tournament such that p_i beats p_j iff $i < j$, for $1 \leq i, j \leq n$. Conclude that a tournament has no consistent ranking iff some subset of three of its players has no consistent ranking.

2. The well-matched strings of left and right parentheses, *WMP*'s for short, are defined inductively as follows:

   - The string of length two consisting of a left parenthesis followed by a right parenthesis, namely, `()`, is a WMP.
   - If $W_1 W_2 \ldots W_k$ is a sequence of $k \geq 1$ WMP's, then $(W_1 W_2 \ldots W_k)$ is a WMP.

   The *count* of a string of one or more parentheses is defined as follows:

   - count(`)`) $= -1$,
   - count(`(`) $= 1$,
   - of s is a string of parentheses and p is a single parenthesis, then $\text{count}(sp) = \text{count}(s) + \text{count}(p)$.

A string, Z, of one or more parentheses is a *well-counting* string of parentheses, a *WCP* for short, iff $\text{count}(Z) = 0$ and $\text{count}(X) > 0$ for any proper prefix, X, of Z. (X is a *prefix* of Z means that $Z = XY$ for some string Y. X is a *proper* prefix if it is a prefix of Z and its length is positive and strictly less than the length of Z.)

(a) Prove that every WMP is a WCP.

Solution: First prove that every WMP is a WCP, by strong induction on the length of the WMP:

$$P(n) ::\equiv \text{If } W \text{ is a length } n \text{ WMP, then } W \text{ is a WCP.}$$

Base Cases: $n \leq 2$. $W = \texttt{()}$ is the only possible WMP of length ≤ 2. Now W has count $1 = -1 = 0$, and its only proper prefix is the initial left parenthesis, which has count 1. So W is a WCP.

Inductive Step: Assume $P(k)$ is true for all $k \leq n$. We need to show that $P(n+1)$ holds.

Let W be a length $n+1$ WMP for $n \geq 2$. By definition of WMP, $W = \texttt{(}W_1 W_2 ... W_k\texttt{)}$, where W_i is a WMP for $1 \leq i \leq k$. By inductive hypothesis, W_i is a WCP for each i.

We note the "obvious" fact (which itself could be proved by induction if need be) that for any strings of parentheses U, V,

$$\text{count}(UV) = \text{count}(U) + \text{count}(V).$$

From this, we have

$$\begin{aligned}
\text{count}(W) &= \text{count}(\texttt{(}W_1 W_2 \ldots W_k\texttt{)}) \\
&= \text{count}(\texttt{(}) + \text{count}(W_1) + \cdots + \text{count}(W_k) + \text{count}(\texttt{)}) \\
&= 1 + 0 + \cdots + 0 - 1 \\
&= 0.
\end{aligned}$$

Now let X be a proper prefix of W. We must show that $\text{count}(X) > 0$. But X must be of the form:

$$X = \texttt{(}W_1 W_2 \ldots W_j Y$$

for some $j < k$ and Y a prefix of W_{j+1}. Since W_{j+1} is a WCP, $\text{count}(Y) \geq 0$. So

$$\begin{aligned}
\text{count}(X) &= \text{count}(\texttt{(}) + \text{count}(W_1) + \cdots + \text{count}(W_j) + \text{count}(Y) \\
&= 1 + 0 + \ldots + 0 + \text{count}(Y) \\
&\geq 1 + 0 > 0.
\end{aligned}$$

■

(b) Conversely, prove that every WCP is a WMP.

Solution: we show that every WCP is a WMP by induction on the length of the WCP:

$$Q(n) ::\equiv \text{If } W \text{ is a length } n \text{ WCP, then } W \text{ is a WMP.}$$

Base Cases: $n \leq 2$. It follows from the definition of the count of a string of parentheses that the only WCP with length ≤ 2 is `()`. This string is a WMP by definition.

Inductive Step: Assume that $Q(k)$ is true for all $k \leq n$, we need to prove that $Q(n+1)$ is also true. Let X be a length $n+1$ WCP for any $n \geq 2$.

Any WCP it must be of length ≥ 2, so it must begin with a parenthesis, and this parenthesis is a proper prefix of the WCP. Since the count of that parenthesis must be positive, it has to be a left parenthesis. Likewise, any WCP must end with a right parenthesis.

So $X = (Y)$ for some string Y. Since the counts of the initial and final parentheses, namely 1 and -1, sum to zero, and $\text{count}(X) = 0$, it must also be the case that $\text{count}(Y) = 0$. Also, since any proper prefix of X is a left parenthesis followed by a prefix of Y, and the count of any proper prefix of X is positive, it follows that the count of any prefix of Y is nonegative.

Let k be the number of positive length prefixes of Y with count zero. Let $Y_1, Y_1Y_2, \ldots, Y_1Y_2\ldots Y_k$ be these prefixes. Now for $i < k$ we have,

$$0 = \text{count}(Y_1 \ldots Y_{i+1}) = \text{count}(Y_1 \ldots Y_i) + \text{count}(Y_{i+1}) = 0 + \text{count}(Y_{i+1}) = \text{count}(Y_{i+1}).$$

Note that $Y_1Y_2\ldots Y_k = Y$ since Y is a prefix of itself and $\text{count}(Y) = 0$. Let V be a proper prefix of Y_{i+1}. Then $Y_1 \ldots Y_i V$ is a prefix of Y and so has nonegative count. In fact, it must have positive count, since the shortest prefix longer than $Y_1 \ldots Y_i$ with count zero is $Y_1 \ldots Y_{i+1}$ by definition. So

$$0 < \text{count}(Y_1 \ldots Y_i V) = \text{count}(Y_1 \ldots Y_i) + \text{count}(V) = 0 + \text{count}(V) = \text{count}(V).$$

Thus, we have shown that the string Y_{i+1} is a WCP for $0 \leq i < k$. By inductive hypothesis, every Y_{i+1} is also a WMP. Hence $Y = Y_1 \ldots Y_k$ is a sequence of WMP's, and therefore $X = (Y)$ is WMP by definition. ■