CMPSCI 250: Introduction to Computation

Lecture #13: More Induction
David Mix Barrington
12 March 2013

More Induction

- Basic Mathematical Induction
- Varying the Start Point; Odds or Evens
- Proving a Greedy Algorithm Optimal
- Strong Induction: The Idea
- Strong Induction Examples: Factoring, Nim, Stamps
- Triangulating Polygons
- The Well-Ordering Principle

Basic Mathematical Induction

- Last time we introduced the proof method of **mathematical induction**, which allows us to prove a statement of the form ∀x:P(x), where the type of the variable x is “natural number”.

- We first prove the **base case**, the statement P(0). Then we assume the **inductive hypothesis**, the statement P(n) with n an arbitrary natural number. From that we carry out the **inductive step** by proving the statement P(n+1).

- Mathematical induction works when the inductive hypothesis provides facts that are useful in carrying out the inductive step -- the same objects need to occur in both the statements P(n) and P(n+1), or at least the objects of P(n) and P(n+1) need to be related in a known way. For example, if P(n) is of the form “S(n) = f(n)”, where S(n) is a sum of terms T(i) for i from i to n, and f(n) is some function, we would use the fact that S(n+1) = S(n) + T(n+1).

Varying the Start Point

- Last time I justified the validity of mathematical induction by referring to the **Peano axioms** for the natural numbers, and in particular the axiom that says that any set S of natural numbers, that contains 0 and is closed under the successor operation, must be the set of *all* natural numbers. If we let S be the set of numbers n for which P(n) is true, an induction proof shows that S has these properties, so that P(n) must be true for all natural numbers n.

- But what if (as in most of Rosen's examples) we have a statement P(n) that is true only for all **positive integers**? It's pretty clear that an induction with base case P(1) should work. In fact we can show $\forall n:(n \geq c) \rightarrow P(n)$, for any value of c, by showing the base case P(c) and then proving $\forall n: (P(n) \wedge (n \geq c)) \rightarrow P(n+1)$. We may or may not need to assume $n \geq c$ in order to prove P(n+1).

- For example, let's prove $2^n \leq n!$ for all n with $n \geq 4$. Our base case is P(4) which says that $2^4 \leq n!$ or $16 \leq 24$, which is true. If we assume $2^n \leq n!$, we can derive $2^{n+1} = 2 \cdot 2^n \leq 2 \cdot n!$ (by the IH) $\leq (n+1)n! = (n+1)!$ and P(n+1) is true.

Induction on the Odds or the Evens

- How would we prove that P(n) is true for all **odd positive integers**, if it is not true for even integers? We couldn't then use P(n) to prove P(n+1), because if n were odd this implication could be false.

- But if we define Q(k) to be P(2k+1), then proving P(n) for all odd positive integers is the same thing as proving Q(k) for all natural numbers k. We could do this by ordinary induction, proving Q(0) and ∀k: Q(k) → Q(k+1). But note that this is the same thing as proving a base case of P(1) and then proving that P(n) → P(n+2) is true for any *odd* n.

- Similarly we can prove P(n) for all even naturals by proving P(0) and ∀n:P(n) → P(n+2), or prove it for all positive even naturals by using P(2) as our base case.

- Similarly, letting Q(k) = P(n+c), proving ∀k:Q(k) proves ∀n:(n ≥ c) → P(n).

iClicker Question #1: Induction by Threes

- Suppose my statement P(n) is “Any 2 by n rectangle can be tiled with L-shaped triominoes”, and I want to prove that P(n) is true for every natural number n that is divisible by 3. **Which two steps** should I prove?

- (a) Base case P(3), inductive step P(n) → P(n+1)

- (b) Base case P(3), inductive step P(n) → P(n+3)

- (c) Base case P(0), inductive step P(n) → P(n+3)

- (d) None of these will work because P(n) is not true for all n divisible by 3.

Proving a Greedy Algorithm Optimal

- Rosen gives an example (#12, pp. 324-325) of using induction to prove that a particular **greedy algorithm** is optimal.

- We have a lecture hall and want to schedule as many events as possible from a set $\{E_1,...,E_n\}$. Each event E_i has a start time s_i and a finish time f_i, and events may not overlap. The greedy algorithm is to begin with the event that has the earliest finish time. Then we look at the events that start after that time, and choose the one of those that finishes first. At each point we look at the remaining events that could possibly fit, and choose the one that finishes first. This continues until there are no events remaining that will fit.

- We let P(k) be the statement "this algorithm will take any set of events where the optimal schedule has k events, and will schedule k events". P(1) is the base case, and we will always schedule one event. So we assume that all sets of events that allow k to be scheduled will be scheduled correctly.

Finishing the Optimality Proof

- Now assume that we have a set of events, and that there exists a schedule of k+1 events that do not overlap. Let E be the event *in that schedule* that has the earliest start time. The other k events in the given schedule must all start after E has finished.

- Our greedy algorithm will begin by picking the event that finishes first. This is either E, or some event E' that finishes before E (or at the same time). Consider the set of events that begin after E' finishes. There *is* a possible schedule of k events out of this set, because the schedule we are given has k such events. Therefore, *by the inductive hypothesis*, our greedy algorithm will find a schedule of k events out of this set.

- This means that our greedy algorithm will find a schedule of k+1 events for the original set, because it will choose its first event and then operate on exactly the set of events that start after its first event finishes. Thus we have proved P(k+1) using P(k) as a hypothesis, and we have completed the proof.

iClicker Question #2: Greedy Algorithms

- The preceding algorithm assumed that all events were of equal value. Suppose that along with its start and finish time, each event also had a **score**, and our goal was to find a schedule with the largest possible total score. **Which of the following** answers the question of whether the same greedy algorithm is always optimal?

- (a) Yes, because greedy algorithms are always optimal.

- (b) No, because there could be an event of very high score that starts before the earliest event finishes, and the greedy algorithm would not choose it.

- (c) Yes, because the same induction proof we just saw is still valid.

- (d) No, because the induction proof we just saw is no longer valid.

Strong Induction: The Idea

- Ordinary mathematical induction depends on some relationship between the statements P(n) and P(n+1). But there are situations where there is no relationship between those two statements, and yet the truth of P(n+1) depends on other statements P(k).

- For example, let P(n) be "n is a product of primes". P(59) and P(60) are both true, but the factorization of 59 has nothing to do with the factorization of 60. To factor 60, we might find its smallest prime factor, which is 2, then factor what is left when we divide by 2. So P(60) depends on P(30), not P(59).

- In **strong induction**, we still prove a base case of P(0) (or perhaps P(c) for some larger c). We assume the **strong inductive hypothesis**, which says that P(i) is true for all i as long as $c \leq i \leq n$. Then we use this hypothesis to prove P(n+1). The conclusion is again that P(n) is true for all n with $n \geq c$.

Why is Strong Induction Valid?

- Strong induction may at first appear to be cheating, since we have more inductive hypotheses available, not just P(n) but P(c), P(c+1), P(c+2),..., all the way up to P(n), but we need to prove only the same conclusion P(n+1).

- An intuition for why we get away with this is that when we use P(n) → P(n+1) in an ordinary induction proof, we have already used the inductive step proof to prove P(c) → P(c+1), P(c+1) → P(c+2),..., and so on up to P(n-1) → P(n). So all those earlier values of P(i) are known to be true when we need them.

- Here is a formal proof (using ordinary induction) that strong induction is valid. Define Q(n) to be the statement ∀i:(c ≤ i ≤ n) → P(i). So Q(n) says that P(i) is true for every value of i fron c through n. What do we need to prove ∀n:Q(n) by ordinary induction? Our base case is Q(c) and our inductive step is Q(n) → Q(n+1). But suppose we prove Q(n) → P(n+1) instead? We then have Q(n), which we assumed, and P(n+1), which we proved. But what is Q(n+1)? It is just Q(n) ∧ P(n+1), by the definition. So the steps of strong induction suffice to prove ∀n:Q(n) by ordinary induction, and ∀n:P(n) follows from this.

Strong Induction Example: Factoring

- Let's use strong induction to formally prove the "easy half" of the **Fundamental Theorem of Arithmetic**, that every positive integer has *at least one* factorization into primes. (The "hard half" is that any two prime factorizations of the same number are reorderings of each other.)

- Our statement P(n) will be "n is a product of primes". P(1) is true because the empty product gives 1. Assume that P(i) is true for all i with $1 \leq i \leq n$, and we will prove P(n+1). Look for the smallest number d, with $d \geq 2$, such that d divides n+1. There must be such a number because n+1 divides itself, and we know that $n+1 \geq 2$. Furthermore, this d must be prime, because if it had a proper divisor e then e would also divide n+1 and d would not be smallest.

- Once we have this d, we know by the strong inductive hypothesis that P((n+1)/d) is true, because $(n+1)/d \leq n$. So (n+1)/d is the product of primes, and n+1 is the product of those same primes and the prime number d.

Strong Induction Example: Nim

- **Nim** is a game somewhat similar to Chomp, where two players alternate in removing objects from a collection. In Nim, we have a positive number of **piles**, each with a positive number of **matches**. A legal move is to choose a pile and take some or all of the matches, taking at least one match. In Rosen's example the goal is to take the last match, but Nim is more often played like Chomp where the goal is to *avoid* taking the last match.

- Rosen gives a proof by strong induction that if there are two piles of equal size, the *second* player has a winning strategy. Let P(n) be the statement "the second player wins the game with two piles, each of size n". We prove P(1) by a direct analysis. The first player has to take the match in one pile, and then the second player can take the last match by taking the other pile.

- Now assume that P(i) is true whenever $1 \leq i \leq n$ and consider the game with two piles of size n+1. The first player takes from one pile, leaving k matches. If k=0 the second player takes all the matches in the other pile and wins. If k is positive she takes *all but k* matches in the other pile, and wins as P(k) is true.

iClicker Question #3: Nim

- Suppose we have three piles of size 2, and the goal is again to take the last match. **What should the first player do** to win the game?

- (a) It doesn't matter, because the first player always wins this game.

- (b) She should take one match from one pile, because something similar worked in Chomp.

- (c) She should take both matches from one pile, because she then becomes the second player in a game with two equal piles, and we proved she can win.

- (d) It doesn't matter, because the second player has a winning strategy no matter what she does.

Strong Induction Example: Stamps

- Suppose that stamps are worth either 5 or 6 cents, and we want to pay postage of exactly k cents. (These problems were devised when postage was much cheaper than it is today.) Can we do it? By trying out possibilities, it looks like we can make 6, 10, 11, 12, 15, 16, 17, 18, 20, and all larger amounts. How can we prove that we can make all k with $k \geq 20$?

- The easy inductive steps to prove would be $P(k) \rightarrow P(k+5)$ and $P(k) \rightarrow P(k+6)$. If our strong inductive hypothesis gives us $P(k-4)$, we can use the inductive step to get $P(k+1)$. But this would not work to prove, for example, $P(19) \rightarrow P(24)$, because we don't have $P(19)$ and $P(19)$ isn't even true.

- But if we prove five base cases, $P(20)$ through $P(24)$, the inductive step $P(k) \rightarrow P(k+5)$ is enough to get all larger numbers. And we have $20 = 5+5+5+5$, $21 = 6+5+5+5$, $22 = 6+6+5+5$, $23 = 6+6+6+5$, and $24 = 6+6+6+6$. If $k \geq 24$, the strong inductive hypothesis for k gives us $P(k-4)$, and we can prove $P(k+1)$.

Triangulating Polygons

- A **polygon** is a geometrical figure made up of line segments in a cycle -- we call it an **n-gon** if it has n sides, so a 3-gon is a triangle, a 4-gon is a quadrilateral, and so forth. An **interior diagonal** is a line from one **vertex** to another that stays within the polygon. A **triangulation** of a polygon is a set of interior diagonals that divides it into triangles.

- Rosen uses strong induction to prove that every n-gon (where $n \geq 3$) has a triangulation into n-2 triangles. Let P(n) be the statement that every n-gon has a triangulation with n-2 triangles. The base case P(3) is obvious, because the 3-gon is itself a triangle and $3-2 = 1$. If we can find any interior diagonal, it divides the n-gon into an i-gon and a j-gon, and the strong IH tells us that these have triangulations into i-2 and j-2 triangles respectively. And we have that $i + j = n+2$, because the interior diagonal becomes two sides.

- If the n-gon is **convex**, every diagonal is an interior diagonal, but Rosen goes to some trouble to prove carefully that every n-gon with $n \geq 4$ has one.

The Well-Ordering Principle

- Suppose we have a set with an **total order relation** on it, so we can say that exactly one of the statements $x < y$, $x = y$, and $x > y$ is true for any x and y in S. Then S is **well-ordered** if every non-empty subset has a smallest element.

- The natural numbers are well-ordered, but the integers and reals are not. The statement that the naturals are is called the **Well-Ordering Principle**.

- We can prove the Well-Ordering Principle by strong induction. Let P(n) be "every set S containing n has a least element". P(0) is true because 0 is the least element in S if it is there. Assume that P(i) is true for all $i \leq n$. If S contains n+1, either n+1 is the least element or S has an element i with $i \leq n$, and must have a least element because P(i) is true.

- We can also use the Well-Ordering Principle to prove ordinary induction valid. Suppose we have proved P(0) and $\forall n:P(n) \rightarrow P(n+1)$. Suppose P(k) were false for some k. There would then be a least k such that P(k) is false. This k can't be 0, and can't be positive because then P(k-1) would imply P(k).

iClicker Question #4: Well-Ordering

- **Which statement** about the natural numbers is **not true**?
- (a) Every nonempty set of natural numbers has a largest element.
- (b) Every empty set of natural numbers does not have a largest element.
- (c) Every finite set of natural numbers has a largest element.
- (d) Every infinite set of natural numbers has a smallest element.