CSCE 2100: Computing Foundations 1

Proofs by Induction (Weak Induction)

Tamara Schneider
Jorge Reyes-Silveyra

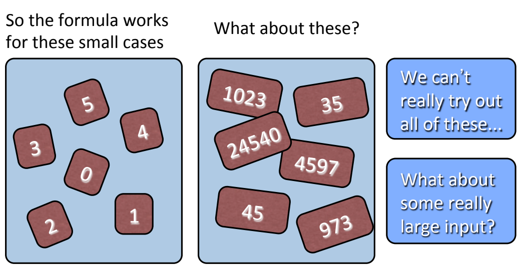
Fall 2012

Proofs by Induction [1]

- You have determined a formula that lets you solve a hard problem (maybe time-consuming) a lot faster and easier.
- You have played around with small values and your formula seems to work…
- But how can you be sure that it will work for all values (and that you will not fail the next test, because you are planning on using your formula)?

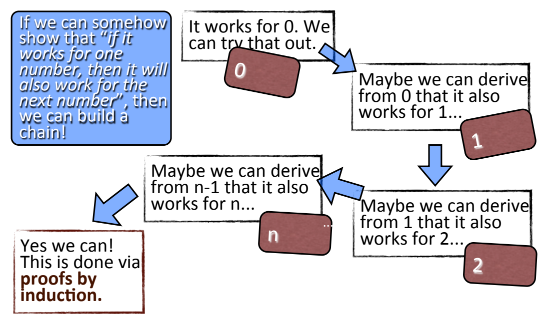
2

Proofs by Induction [2]



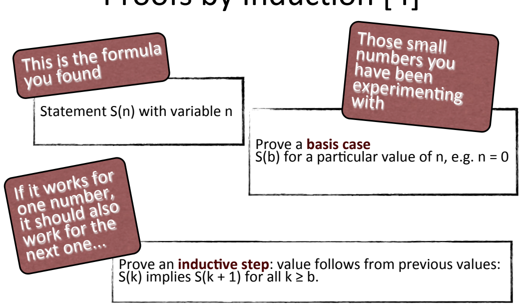
3

Proofs by Induction [3]



4

Proofs by Induction [4]



5

Components

- We will see that each proof by induction consists of the following components:
  1. The **statement** S(n) that we need to proof
  2. A **basis case**
  3. The **induction hypothesis**
  4. A "**target**", i.e. what is it that we need to proof?
  5. The **inductive step**, i.e. we need to show that if it is true for k, then it automatically is true for k+1.

6

Example 1: Summation Formula [1]

- Assignment to keep students busy: add all integers from 1 to 100
- Gauss' observation:

  1 + 2 + 3 + ... + 99 + 100
  100 + 99 + 98 + ... + 2 + 1
  101 + 101 + 101 + ... + 101 + 101

- 100 * 101 counts all numbers twice -> (100*101)/2

Example 1: Summation Formula [2]

1. Statement

S(n) = $\sum_{i=1}^{n} i = \frac{n \times (n+1)}{2}$

2. Basis Case

- Basis case: n = 1. The sum of all numbers from 1 to 1 is 1

$$\sum_{i=1}^{1} i = 1$$

- The basis case proves to be correct. We now need to show that it is true for all k > 1.

$$\frac{1 \times (1+1)}{2} = \frac{2}{2} = 1 \quad \checkmark$$

Example 1: Summation Formula [3]

$$\sum_{i=1}^{n} i = \frac{n \times (n+1)}{2}$$

3. Induction Hypothesis

- Assume that the statement is true for k ≥ 1.

$$\sum_{i=1}^{k} i = \frac{k \times (k+1)}{2}$$

4. "Target"

- Show that it also holds for k+1

$$\sum_{i=1}^{k+1} i \overset{?}{=} \frac{(k+1) \times ((k+1)+1)}{2}$$

Example 1: Summation Formula [4]

$$\sum_{i=1}^{n} i = \frac{n \times (n+1)}{2}$$

5. Inductive Step

$$\sum_{i=1}^{k+1} i = \sum_{i=1}^{k} i + (k+1)$$

Ind. Hyp

$$= \frac{k \times (k+1)}{2} + (k+1) = \frac{k \times (k+1) + 2 \times (k+1)}{2}$$

$$= \frac{(k+1) \times (k+2)}{2} \quad \blacksquare$$

Proofs by Induction – Recap

So it works for some small k: S(b) is true...

→ It also works for the next one: S(b+1) is true

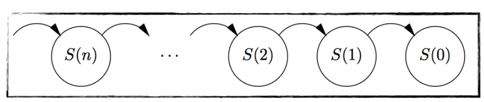
→ And the next one: S(b+2)

→ And the next one: S(b+3)

...

Proof that S(n) is true for all nonnegative integers above some lower limit b.

All Values for n Lead Back to Basis



The proof for each statement depends on the previous one. For k ≥ b:

- The proof for S(n) uses S(n-1)
- The proof for S(n-1) uses S(n-2)

  ...

- The proof for S(b+1) uses S(b) (basis case reached)

Components of Proofs by Induction

1. Statement — The statement is given. You need to show that it is true for any n.

2. Basis Case — Plug in a small value b (as small as allowed by the statement) and show that it works:
The right-hand side (RHS) and the left-hand side (LHS) of the equation should lead to the same result or conclusion.

3. Induction Hypothesis — You need to make this assumption for any proof by induction. (So make sure to write this: Assume that the statement is true for k ≥ b). Write down the statement for the specific proof.

13

Components of Proofs by Induction

4. "Target" — What is it that you need to prove in the next step? Re-write the statement for (k+1) instead of n.
Important: As opposed to the induction hypothesis, you can NOT use this for your proof as something that holds, since it is what you need to prove.

5. Inductive Step — You will start with the left-hand side (LHS) of your "target" and show how you can transform it into the right-hand side (RHS) of your target.
You will have to use the induction hypothesis during this process!

14

Why does this work?

Induction is axiomatic

- Axiom: A statement or proposition that is regarded as being established, accepted or self-evidently true.

- So we can only argue that it seems plausible:
We show that the basis case S(b) is true. Then that S(k) implies S(k+1). Therefore, S(b) implies S(b+1). S(b+1) implies S(b+2) etc. Eventually we will reach S(n) for any value n ≥ b.

15

Why do we do this?

- We cannot just use a formula because it "seems" to be correct.
- Sometimes a certain concept seems to have a relationship that "looks" correct. A proof by induction can help us determine if it really is correct.
- We need to formally proof mathematical statements so that we can use them in the future.

16

Inductive Proof Gone Wrong… [1]

- It may seem at first glance that the procedure will always work, no matter if the statement is actually true.
- Let us try to proof a statement that we know is false:

$$S(n) = \sum_{i=1}^{n} i = n^2$$

We can easily see that S(n) is false, e.g. for n=3

$$LHS: \sum_{i=1}^{3} i = 1+2+3 = 6 \qquad 6 \neq 9$$

$$RHS: 3^2 = 9$$

17

Inductive Proof Gone Wrong… [2]

- But let us try to proof the statement anyway:

1. Statement $$S(n) = \sum_{i=1}^{n} i = n^2$$

2. Basis Case

- The smallest value that seems to be "allowed" by the statement is one. So let n=1:

$$LHS: \sum_{i=1}^{1} i = 1 \qquad 1 = 1$$

$$RHS: 1^2 = 1$$

18

Inductive Proof Gone Wrong… [3]

$$S(n) = \sum_{i=1}^{n} i = n^2$$

3. Induction Hypothesis

- Since we have shown the existence of a basis case n=1, we now assume that S(k) holds for some k ≥ 1. So for k ≥ 1 we assume:

$$\sum_{i=1}^{k} i = k^2$$

19

Inductive Proof Gone Wrong… [4]

$$S(n) = \sum_{i=1}^{n} i = n^2$$

4. "Target"

- What is it that we need to proof? We need to proof that if it works for k, then it also works for k+1:

$$\sum_{i=1}^{k+1} i \overset{?}{=} (k+1)^2$$

20

Inductive Proof Gone Wrong… [5]

$$S(n) = \sum_{i=1}^{n} i = n^2$$

5. Inductive Step

- We start with the LHS of the target, use the inductive hypothesis and try to the obtain the RHS of the target:

$$\sum_{i=1}^{k+1} i = 1+2+3+...+k+(k+1)$$

$$= \sum_{i=1}^{k} i + (k+1)$$

Ind. Hyp

$$= k^2 + k + 1 \neq (k+1)^2$$

We can not show that these are equal.

21

Example 2: Sum of the powers of 2 [1]

1. Statement

$$\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$$

2. Basis Case

- Base case: n = 0
  Show that the formula holds for n = 0 S(0)

$$\sum_{i=0}^{0} 2^i = 2^0 = 1$$

$$2^{0+1} - 1 = 2 - 1 = 1 \quad \checkmark$$

- The base case proves to be correct. We now need to show that it is true for all n > 0.

22

Example 2: Sum of the powers of 2 [2]

$$\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$$

3. Induction Hypothesis

Assume that the statement is true for k ≥ 0.

$$\sum_{i=0}^{k} 2^i = 2^{k+1} - 1$$

4. "Target"

Show that S(k) implies S(k+1), i.e. it also holds for k+1

$$\sum_{i=0}^{k+1} 2^i \overset{?}{=} 2^{(k+1)+1} - 1$$

23

Example 2: Sum of the powers of 2 [2]

$$\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$$

5. Inductive Step

Proof the statement by transforming the LHS of the target into the RHS. Use the inductive hypothesis in the process!!!

$$\sum_{i=0}^{k+1} 2^i = \sum_{i=0}^{k} 2^i + 2^{n+1}$$

Ind. Hyp

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

■

24

Example 3: Error-Detection [1]

- American Standard Code for Information Interchange (ASCII)
- 7-bit code for each character
  - A: 1000001
  - C: 1000011
- Parity: count number of 1's
  - If even: prepend 0
  - If odd: prepend 1

25

Example 3: Error-Detection [2]

- 8-bit code with parity bit
  - A: 01000001
  - C: 11000011
- Number of 1's is now always even
- Can any two code words differ in exactly one position?
- How many code words can be represented?
- Can errors be detected?
- Can errors be fixed?

26

Example 3: Error-Detection [3]

1. Statement

S(n): If C is any set of bit strings of length n that is error detecting, — LHS

then C contains at most 2^{n-1} strings — RHS

RECALL: error detecting = all strings differ in more than 1 position

Base case
S(0) or S(1) or S(2) ... ?

27

Example 3: Error-Detection [4]

S(n): If C is any set of bit strings of length n that is error detecting, — LHS

then C contains at most 2^{n-1} strings — RHS

- S(0)
  - The set contains strings of length 0
  - It contains only the empty string, i.e. one string
- S(n) claims that the set contains 2^{n-1} strings, i.e. 1/2 string.
- S(0) cannot be the basis case

NOT A BASIS CASE

28

Example 3: Error-Detection [5]

S(n): If C is any set of bit strings of length n that is error detecting, — LHS

then C contains at most 2^{n-1} strings — RHS

2. Basis Case

- S(1)
  - The set contains strings of length 1
  - How many distinct strings of length 1 can be formed?
  - How many of these strings are allowed in an error detecting code?
- S(n) claims that the set contains 2^{n-1} strings, i.e. 1 string.
- Is this a valid basis case?

29

Example 3: Error-Detection [6]

For S(1) the statement is true, since it allows only a single code word and $2^{1-1}=1$

S(n): If C is any set of bit strings of length n that is error detecting, — LHS

then C contains at most 2^{n-1} strings — RHS

3. Induction Hypothesis

- Let k≥1. Assume that

LHS An error-detecting set of strings of length k

RHS has at most 2^{k-1} strings

30

Example 3: Error-Detection [7]

S(n): If C is any set of bit strings of length n that is error detecting, — LHS

then C contains at most 2^{n-1} strings — RHS

4. "Target"

- We want to show that

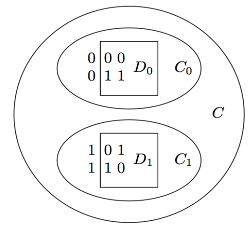
? LHS An error-detecting set of strings of length (k+1)
RHS has at most $2^{(k+1)-1}$ strings

31

Example 3: Error-Detection [8]

5. Inductive Step

- Split set C of strings with length k+1 into 2 sets:
  - Code words starting with 0 and
  - Code words starting with 1
- Examine D_0 and D_1.
  - Since they are both in C, all code words must differ by at least 2 positions

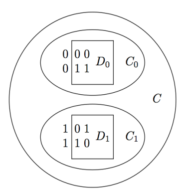


32

Example 3: Error-Detection [9]

Ind. Hyp

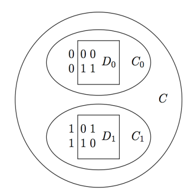
- D_0 and D_1 each contain error-detecting codes of length k and
  - Why?
  - So they each can have at most 2^{k-1} strings.
- Therefore C_0 and C_1 have at most 2^{k-1} strings each.



33

Example 3: Error-Detection [10]

- W have determined that C_0 and C_1 have at most 2^{k-1} strings each.
  - It follows that C has at most $2^{k-1} + 2^{k-1} = 2^k$ strings.
- This proves that S(k) implies S(k+1) and that S(n) is true for all $n \geq 1$. ∎



34

Example 3: Error-Detection [11]

- What practical implications does this have for us?
  - Assume you want to generate an error-detecting code that must contain x different code words....
  - Now we can derive how long the code words must be while using only the minimum number of bits.
- Why do we care about the length of the code words?

35

Example 4: Divisibility [1]

1. Statement

- S(n): 9^n-1 (LHS) is divisible by 8 (RHS)

2. Basis Case

- Basis: n=0

  $9^0-1 = 0$ 0 is divisible by 8 ✔

36

Example 4: Divisibility [2]

3. Induction Hypothesis

- The basis proves to be correct. We now make the assumption that S(n) is true for some $k \geq 1$.

So we assume that 9^k-1 is divisible by 8.

LHS RHS

– > $9^k-1 = 8t$ (t indicates how often 8 fits into 9^k-1)

LHS RHS

37

Example 4: Divisibility [3]

4. "Target"

- The basis proves to be correct. We now need to show that

9^{k+1}-1 is? divisible by 8.

LHS RHS

– For the RHS we expect to see an expression indicating that the LHS is divisible by 8.

38

Example 4: Divisibility [4]

5. Inductive Step

(Show that S(k) -> S(k+1))

- We assume that 9^k-1 is divisible by 8. (**$9^k-1 = 8t$**)
- We need to show that 9^{k+1}-1 is divisible by 8.

$9^{k+1}-1 = 9 \times 9^k - 1$

$= 9 \times 9^k - 1 - 8 + 8$

$= 9 \times 9^k - 9 + 8 = (9 \times 9^k - 9) + 8$ Ind. Hyp

$= 9 \times (\mathbf{9^k - 1}) + 8 = 9 \times \mathbf{8t} + 8$

$= 8 \times 9t + 8 = 8 \times (9t + 1)$ ■

Since $9^{k+1}-1 = 8 \times (9t + 1)$, 9^{k+1}-1 is divisible by 8 and S(k+1) holds.

39

Summary Weak Induction

1. Determine S(n) to be proved for all $n \geq b$.
2. Determine and prove basis case S(b).
3. Assume that the statement is true for k.
4. Find out what you actually need to prove (target).
5. Show that S(k) implies S(k+1). (make use of S(k) during the proof)
6. Conclude that S(n) is true for all $n \geq b$.

40