Chapter 3

Proofs by Induction

3.1 Simple Induction

youtu.be/jOnZTWGpX_I
The Induction Principle

Let us consider a simple problem: what is bigger 2^n or n? In this chapter, we are going to study the simplest way to prove that $2^n > n$ for all positive integers n. First, let us check that it is true for small integers n.

n	1	2	3	4	5	6	7	8
2^n	2	4	8	16	32	64	128	256

We may also note that 2^n is growing faster than n, so we expect that if $2^n > n$ for small integers n, then it is true for all positive integers n.

In order to prove this statement formally, we use the following principle.

Principle 3.1 (The Induction Principle)**.** *Let $P(n)$ be some statement about a positive integer n. Hence, $P(n)$ is true for every positive integer n iff*

base case: *$P(1)$ is true and*

induction step: *$P(k) \implies P(k+1)$ is true for all positive integers k.*

Let us prove now the statement using this principle. We define $P(n)$ be the statement that “$2^n > n$”. $P(1)$ is true since $2^1 > 1$. Let us assume now that $2^n > n$. Note that $2^{n+1} = 2 \cdot 2^n > 2n \geq n + 1$. Hence, we proved the induction step.

Exercise 3.1. *Prove that $(1 + x)^n \geq 1 + nx$ for all positive integers n and real numbers $x \geq -1$.*

3.2 Changing the Base Case

Let us consider functions n^2 and 2^n.

n	1	2	3	4	5	6	7	8
n^2	1	4	9	16	25	36	49	64
2^n	2	4	8	16	32	64	128	256

Note that 2^n is greater than n^2 starting from 5. But without some trick we can not prove this using induction since for $n = 3$ it is not true!

The trick is to use the statement $P(n)$ stating that $(n+4)^2 < 2^{n+4}$. The base case when $n = 1$ is true. Let us now prove the induction step. Assume that $P(k)$ is true i.e. $(k+4)^2 < 2^{k+4}$. Note that $2(k+4)^2 < 2^{k+1+4}$ but $(k+5)^2 = k^2 + 10k + 25 \leq 2k^2 + 16k + 32 = 2(k+4)^2$. Which implies that $2^{k+1+4} > (k+5)^2$. So $P(k+1)$ is also true.

In order to avoid this strange +4 we may change the base case and use the following argument.

Theorem 3.1. *Let $P(n)$ be some statement about an integer n. Hence, $P(n)$ is true for every integer $n > n_0$ iff*

base case: *$P(n_0 + 1)$ is true and*

induction step: *$P(k) \implies P(k+1)$ is true for all integers $k > n_0$.*

Using this generalized induction principle we may prove that $2^n \geq n^2$ for $n \geq 4$. The base case for $n = 4$ is true. The induction step is also true; indeed let $P(k)$ be true i.e. $(k+4)^2 < 2^{k+4}$. Hence, $2(k+4)^2 < 2^{k+1+4}$ but $(k+5)^2 = k^2 + 10k + 25 \leq 2k^2 + 16k + 32 = 2(k+4)^2$.

Let us now prove the theorem. Note that the proof is based on an idea similar to the trick with +4, we just used.

Proof of Theorem 3.1. $\Rightarrow$ If $P(n)$ is true for any $n > n_0$ it is also true for $n = n_0 + 1$ which implies the base case. Additionally, it true for $n = k + 1$ so the induction step is also true.

$\Leftarrow$ In this direction the proof is a bit harder. Let us consider a statement $Q(n)$ saying that $P(n + n_0)$ is true. Note that by the base case for P, $Q(1)$ is true; by the induction step for P we know that $Q(n)$ implies $P(n+1)$. As a result, by the induction principle $Q(n)$ is true for all positive integers n. Which implies that $P(n)$ is true for all integers $n > n_0$.

□

3.3 Inductive Definitions

We may also define objects inductively. Let us consider the sum $1 + 2 + \cdots + n$ a line of dots indicating "and so on" which indicates the definition by induction. In this case, a more precise notation is $\sum_{i=1}^{n} i$.

Definition 3.1. *Let $a(1), \ldots, a(n), \ldots$ be a sequence of integers. Then $\sum_{i=1}^{n} a(i)$ is defined inductively by the following statements:*

- $\sum_{i=1}^{1} a(i) = a(1)$, *and*
- $\sum_{i=1}^{k+1} a(i) = \sum_{i=1}^{k} a(i) + a(k+1)$.

Let us prove that $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$. Note that by definition $\sum_{i=1}^{1} i = 1$ and $\frac{1(1+1)}{2} = 1$; hence, the base case holds. Assume that $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$. Note that $\sum_{i=1}^{n+1} i = \sum_{i=1}^{n} i + (n+1)$ and by the induction hypothesis $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$. Hence, $\sum_{i=1}^{n+1} i = \frac{n(n+1)}{2} + (n+1) = \frac{(n+1)(n+2)}{2}$.

Exercise 3.2. *Prove that $\sum_{i=1}^{n} 2^i = 2^{n+1} - 2$.*

3.4 Analysis of Algorithms with Cycles

Induction is very useful for analysing algorithms using cycles. Let us extend the example we considered in Section 1.3 on page 6.

Let us consider the following algorithm. We prove that it is working correctly.

Algorithm 2 The algorithm that finds the maximum element of $a_1, \ldots, a_n$.

```
function Max($a_1, \ldots, a_n$)
    $r \leftarrow a_1$
    for $i$ from 2 to $n$ do
        if $a_i > r$ then
            $r \leftarrow a_i$
        end if
    end for
    return r
end function
```

First, we need to define $r_1, \ldots, r_n$ the value of r during the execution of the algorithm. It is easy to see that $r_1 = a_1$ and $r_{i+1} = \begin{cases} r_i & \text{if } r_i > a_{i+1} \\ a_{i+1} & \text{otherwise} \end{cases}$. Secondly, we prove by induction that r_i is the maximum of $a_1, \ldots, a_i$. It is clear that the base case for $i = 1$ is true. Let us prove the induction step from k to $k+1$. By the induction hypothesis, r_k is the maximum of $a_1, \ldots, a_k$. We may consider two following cases.

- If $r_k > a_{k+1}$, then $r_{k+1} = r_k$ is the maximum of $a_1, \ldots, a_{k+1}$ since r_k is the maximum of $a_1, \ldots, a_k$.
- Otherwise, a_{k+1} is greater than or equal to $a_1, \ldots, a_k$, hence, $r_{k+1} = a_{k+1}$.

Exercise 3.3. *Show that line 6 in the following sorting algorithm executes $\frac{n(n+1)}{2}$ times.*

Algorithm 3 The algorithm is selection sort, it sorts $a_1, \ldots, a_n$.

```
function SelectionSort($a_1, \ldots, a_n$)
    for $i$ from 1 to $n$ do
        $r \leftarrow a_i$
        $\ell \leftarrow i$
        for $j$ from $i$ to $n$ do
            if $a_j > r$ then
                $r \leftarrow a_j$
                $\ell \leftarrow j$
            end if
        end for
        Swap $a_i$ and $a_\ell$.
    end for
end function
```

3.5 Strong Induction

Sometimes $P(k)$ is not enough to prove $P(k+1)$ and we need all the statements $P(1), \ldots, P(k)$. In this case we may use the following induction principle.

Theorem 3.2 (The Strong Induction Principle)**.** *Let $P(n)$ be some statement about positive integer n. Hence, $P(n)$ is true for every integer $n > n_0$ iff*

base case: *$P(n_0 + 1)$ is true and*

induction step: *If $P(n_0 + 1), \ldots, P(n_0 + k)$ are true, then $P(n_0 + k + 1)$ is also true for all positive integers k.*

Before we prove this theorem let us prove some properties of Fibonacci numbers using this theorem. The Fibonacci numbers are defined as follows: $f_0 = 0$, $f_1 = 1$, and $f_k = f_{k-1} + f_{k-2}$ for $k \geq 2$ (note that they are also defined using strong induction since we use not only f_{k-1} to define f_k).

Theorem 3.3 (The Binet formula)**.** *The Fibonacci numbers are given by the following formula*

$$f_n = \frac{\alpha^n - \beta^n}{\sqrt{5}},$$

where $\alpha = \frac{1+\sqrt{5}}{2}$ and $\beta = \frac{1-\sqrt{5}}{2}$.

Proof. We use the strong induction principle to prove this statement with $n_0 = -1$. Let us first prove the base case, $\frac{(\alpha^0 - \beta^0)}{\sqrt{5}} = 0 = f_0$. We also need to prove the induction step.

- If $k = 1$, then $\frac{(\alpha^1 - \beta^1)}{\sqrt{5}} = 1 = f_1$.

- Otherwise, by the induction hypothesis, $f_k = \frac{\alpha^k - \beta^k}{\sqrt{5}}$ and $f_{k-1} = \frac{\alpha^{k-1} - \beta^{k-1}}{\sqrt{5}}$. By the definition of the Fibonacci numbers $f_{k+1} = f_k + f_{k-1}$. Hence,

  $$f_{k+1} = \frac{\alpha^k - \beta^k}{\sqrt{5}} + \frac{\alpha^{k-1} - \beta^{k-1}}{\sqrt{5}}.$$

  Note that it is enough to show that

  $$\frac{\alpha^{k+1} - \beta^{k+1}}{\sqrt{5}} = \frac{\alpha^k - \beta^k}{\sqrt{5}} + \frac{\alpha^{k-1} - \beta^{k-1}}{\sqrt{5}}. \tag{3.1}$$

  Note that it is the same as

  $$\frac{\alpha^{k+1} - \alpha^k - \alpha^{k-1}}{\sqrt{5}} = \frac{\beta^{k+1} - \beta^k - \beta^{k-1}}{\sqrt{5}}.$$

  Additionally, note that α and β are roots of the equation $x^2 - x - 1 = 0$. Hence, $\alpha^{k+1} - \alpha^k - \alpha^{k-1} = \alpha^{k-1}(\alpha^2 - \alpha - 1) = 0$ and $\beta^{k+1} - \beta^k - \beta^{k-1} = \beta^{k-1}(\beta^2 - \beta - 1) = 0$. Which implies equality (3.1).

□

Now we are ready to prove the strong induction principle.

Proof of Theorem 3.2. It is easy to see that if $P(n)$ is true for all $n > n_0$, then the base case and the induction steps are true. Let us prove that if the base case and the induction step are true, then $P(n)$ is true for all $n > n_0$.

Let $Q(k)$ be the statement that $P(n_0 + 1)$, …, $P(n_0 + k)$ are true. Note that $Q(1)$ is true by the base case for P. Additionally, note that if $Q(k)$ is true, then $Q(k + 1)$ is also true, by the induction step for P. Hence, by the induction principle, $Q(k)$ is true for all positive integers k. Which implies that $P(n_0 + k)$ is true for all positive integers k. □

3.6 Recursive Definitions

Sometimes you wish to define objects using objects of the same form like in the case of inductive definitions but you do not know how to enumerate them using an integer parameter.

One example of such a situation is the definition of an arithmetic formula.

base case: x_i is an arithmetic formula on the variables x_1, …, x_n for all i; if c is a real number, then c is also an arithmetic formula on the variables x_1, …, x_n.

recursion step: If P and Q are arithmetic formulas on the variables x_1, …, x_n, then $(P + Q)$ and $P \cdot Q$ are arithmetic formulas on the variables x_1, …, x_n.

Note that this definition implicitly states that any other expressions are not arithmetic formulas.

We can define recursively the value of such a formula. Let v_1, …, v_n be some integers.

base cases: $x_i\big|_{x_1=v_1,\dots,x_n=v_n} = v_i$; in other words, the value of the arithmetic formula x_i is equal to v_i when $x_1 = v_1$, …, $x_n = v_n$; if c is a real number, then $c|_{x_1=v_1,\dots,x_n=v_n} = c$.

recursion steps: If P and Q are arithmetic formulas on the variables x_1, …, x_n, then

$$(P+Q)\big|_{x_1=v_1,\dots,x_n=v_n} = P\big|_{x_1=v_1,\dots,x_n=v_n} + Q\big|_{x_1=v_1,\dots,x_n=v_n}$$

and

$$(P\cdot Q)\big|_{x_1=v_1,\dots,x_n=v_n} = P\big|_{x_1=v_1,\dots,x_n=v_n} \cdot Q\big|_{x_1=v_1,\dots,x_n=v_n}.$$

For example, $((x_1+x_2)\cdot x_3)$ is clearly an arithmetic formula on the variables x_1, …, x_n. One may expect the value of this formula with $x_1 = 1$, $x_2 = 0$, and $x_3 = -1$ be equal to -1, let us check:

- Note that
$$x_1\big|_{x_1=1,x_2=0,x_3=-1} = 1,$$
$$x_2\big|_{x_1=1,x_2=0,x_3=-1} = 0, \text{ and}$$
$$x_3\big|_{x_1=1,x_2=0,x_3=-1} = -1.$$

- Hence,
$$(x_1+x_2)\big|_{x_1=1,x_2=0,x_3=-1} = 1+0 = 1.$$

- Finally,
$$((x_1+x_2)\cdot x_3)\big|_{x_1=1,x_2=0,x_3=-1} = 1\cdot -1 = -1.$$

A special case of induction which called structural induction is the easiest way to prove properties of recursively defined objects. The idea of this is similar to the idea of strong induction:

- first, we prove the statement for the base case,
- after that we prove the induction step, using the assumption that the statement is true for all the substructures (e.g. subformulas in the previous definition).

To illustrate this method, we prove the following theorem.

Theorem 3.4. *For any arithmetic formula A on x, there is a polynomial p such that $p(v) = A\big|_{x=v}$ for any real value v.*

Proof. **base cases:** If $A = x_i$, then consider the polynomial $p(x) = x$; it is easy to see that $A\big|_{x=v} = v = p(v)$. If $A = c$ where c is a real number, then consider the constant polynomial $p(x) = c$; it is easy to note that $A\big|_{x=v} = c = p(v)$.

induction step: We need to consider two cases. Consider the case when $A = B_1 + B_2$. By the induction hypothesis, there are polynomials q_1 and q_2 such that $B_1\big|_{x=v} = q_1(v)$ and $B_2\big|_{x=v} = q_2(v)$ for all real numbers v. We define $p(x) = q_1(x) + q_2(x)$ (it is a polynomial since sum of two polynomials is a polynomial). It is obvious that $A\big|_{x=v} = B_1\big|_{x=v} + B_2\big|_{x=v} = q_1(v) + q_2(v) = p(v)$.

Another case is $A = B_1 \cdot B_2$. Again, by the induction hypothesis, there are polynomials q_1 and q_2 such that $B_1\big|_{x=v} = q_1(v)$ and $B_2\big|_{x=v} = q_2(v)$ for all real numbers v. We define $p(x) = q_1(x) \cdot q_2(x)$ (it is a polynomial since product of two polynomials is a polynomial). It is obvious that $A\big|_{x=v} = B_1\big|_{x=v} \cdot B_2\big|_{x=v} = q_1(v) \cdot q_2(v) = p(v)$.

□

Exercise 3.4.

- *Define arithmetic formulas with division and define their value (make sure that you handled divisions by* 0*).*
- *Show that for any arithmetic formula with division A on x, there are polynomials p and q such that $\frac{p(v)}{q(v)} = A\big|_{x=v}$ or $A\big|_{x=v}$ is not defined for any real value v.*

3.7 Analysis of Recursive Algorithms

To illustrate the power of recursive definitions and strong induction, let us analyze Algorithm 4. We prove that number of comparisons of this algorithm is bounded by $6 + 2\log_2(n)$. First step of the proof is to denote the worst number of comparisons when we run the algorithm on the list of length n by $C(n)$. It is easy to see that $C(n) = n$ for $n \leq 5$. Additionally, $C(n) \leq 1 + \max(C(\lfloor \frac{n}{2} \rfloor), C(n - \lfloor \frac{n}{2} \rfloor))$ for $n > 5$. As we mentioned we prove that $C(n) \leq 6 + 2\log_2(n)$, we prove it by induction. The base case is clear; let us now prove the induction step. By the induction hypothesis,

$$C(\left\lfloor \frac{n}{2} \right\rfloor) \leq 6 + 2\log_2(\left\lfloor \frac{n}{2} \right\rfloor)$$

and

$$C(n - \left\lfloor \frac{n}{2} \right\rfloor) \leq 6 + 2\log_2(n - \left\lfloor \frac{n}{2} \right\rfloor).$$

Since $\lfloor \frac{n}{2} \rfloor \leq \frac{n}{2}$ and $n - \lfloor \frac{n}{2} \rfloor \leq \frac{n}{2} + 1$, $C(n) \leq 1 + 2\log_2(\frac{n}{2} + 1)$. However,

$$1 + 6 + 2\log_2\left(\frac{n}{2} + 1\right) \leq 6 + 2\log_2\left(\frac{n}{\sqrt{2}} + \sqrt{2}\right) \leq 6 + 2\log_2(n)$$

for $n \geq 5$. As a result, we proved the induction step.

Algorithm 4 The binary search algorithm that finds an element e in the sorted list $a_1, \ldots, a_n$.

```
function BinarySearch($e$, $a_1$, …, $a_n$)
    if $n \leq 5$ then
        for $i$ from 1 to $n$ do
            if $a_i = e$ then
                return i
            end if
        end for
    else
        $\ell \leftarrow \lfloor \frac{n}{2} \rfloor$
        if $a_\ell \leq e$ then
            BinarySearch($e$, $a_1$, …, $a_\ell$)
        else
            BinarySearch($e$, $a_{\ell+1}$, …, $a_n$)
        end if
    end if
end function
```

End of The Chapter Exercises

3.5 Show that there does not exist the largest integer.

3.6 Show that for any positive integer n, $n^2 + n$ is even.

3.7 Show that for any positive integer n, 3 divises $n^3 + 2n$.

3.8 Show that for any integer $n \geq 10$, $n^3 \leq 2^n$.

3.9 Show that for any positive integer n, $\sum_{i=0}^{n} x^i = \frac{1-x^{n+1}}{1-x}$.

3.10 Show that for any matrix $A \in \mathbb{R}^{m \times n}$ $(n > m)$ there is a nonzero vector $x \in \mathbb{R}^n$ such that $Ax = 0$.

3.11 Show that all the elements of $\{0, 1\}^n$ (Binary strings) may be ordered such that every successive strings in this order are different only in one character. (For example, for $n = 2$ the order may be 00, 01, 11, 10.)

3.12 Let $a_0 = 2$, $a_1 = 5$, and $a_n = 5a_{n-1} - 6a_{n-2}$ for all integers $n \geq 2$. Show that $a_n = 3^n + 2^n$ for all integers $n \geq 0$.

3.13 Show that $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$ for all integers $n \geq 1$.

3.14 Show that $\sum\limits_{i=1}^{n} \frac{1}{i(i+1)} = \frac{n}{n+1}$ for all integers $n \geq 1$.

3.15 Show that $\sum\limits_{i=1}^{n} \frac{1}{i^2} \leq 2 - \frac{1}{n}$ for all integers $n \geq 1$.

3.16 Show that $\sum_{i=1}^{n}(2i-1)=n^2$ for any positive integer n.

3.17 Prove that $\sum_{i=1}^{n}\frac{1}{i(i+1)}=\frac{n}{n+1}$ for any positive integer n.

3.18 Prove that $\sum\limits_{i=2}^{n}(i+1)2^i=n2^{n+1}$ for all integers $n>2$.

3.19 Let a_1, …, a_n be a sequence of real numbers. We define inductively $\prod_{i=k}^{n}a_i$ as follows:

- $\prod_{i=1}^{1}a_i=a_1$ and
- $\prod_{i=1}^{k+1}a_i=\left(\prod_{i=1}^{k}a_i\right)\cdot a_{k+1}$.

Prove that $\prod_{i=1}^{n-1}\left(1-\frac{1}{(i+1)^2}\right)=\frac{n+1}{2n}$ for all integers $n>1$.

3.20 Let $f_0=1$, $f_1=1$, and $f_{n+2}=f_{n+1}+f_n$ for all integers $n\geq 0$. Show that $f_n\geq\left(\frac{3}{2}\right)^{n-2}$.

3.21 Show that $f_{n+m}=f_{n-1}f_{m-1}+f_nf_m$.

3.22 Show that two arithmetic formulas $(x_1+x_2)\cdot x_3$ and $x_1\cdot x_3+x_2\cdot x_3$ on the variables x_1, x_2, and x_3 have the same values.

3.23 We say that L is a list of powers of x iff

- either $L=x^k$ for some positive integer k or
- $L=(x^k,L')$ where L' is a list of powers of x and k is a positive integer.

Let L be a list of powers of x. We say that the sum of L with $x=v$ denoted by $\sum L\big|_{x=v}$

- is equal to x^k whether $L=x^k$ and
- is equal to $x^k+\sum L'\big|_{x=v}$ whether $L=(x^k,L')$.

Prove that for any list L of powers of x there is a polynomial such that $\sum L\big|_{x=v}=p(v)$ for all real numbers v.

3.24 Let us define $n!$ as follows: $1!=1$ and $n!=(n-1)!\cdot n$. Show that $n!\geq 2^n$ for any $n\geq 4$.

3.25 Show that $\int\limits_{0}^{+\infty}x^ne^{-x}\,\mathrm{d}x=n!$ for all $n\geq 0$.

3.26 Prove that $\sum\limits_{i=1}^{n}(i+1)2^i=n2^{n+1}$ for all integers $n\geq 1$.

3.27 Show that $\sum_{k=1}^{n}k\cdot k!=(n+1)!-1$.