2110355 FORMAL LANGUAGES AND AUTOMATA THEORY

THEORY AND METHODS OF PROOF

Athasit Surarerks, *Dr. en Inf.*

How sure am I of my answers ?

THEORY AND METHODS OF PROOF

Take a few minutes to try to answer these questions for yourself.

$\lfloor x - 1 \rfloor = \lfloor x \rfloor - 1$?

$\lfloor x - y \rfloor = \lfloor x \rfloor - \lfloor y \rfloor$?

THEORY AND METHODS OF PROOF

The underlying theme of this topic is the question of how to determine the truth or falsity of a mathematical statement.

Understanding mathematical induction.

TECHNICAL WORDS

THEOREM
PROOF
RULES OF INFERENCES
LEMMA
COROLLARY
CONJECTURE

TECHNICAL WORDS

A theorem is a statement that can be shown to be true.
A sequence of statements used to demonstrate a theorem is called a proof.
The rules of inferences, which are the means used to draw conclusions from other assertions, tie together the steps of a proof.
A lemma is a simple theorem used in the proof of other theorems.
A corollary is a proposition that can be established directly from a theorem that has been proved.
A conjecture is a statement whose truth value is unknown.

ContenTs

- Methods of Proof
  - Direct proof
  - Disproof by counterexample
  - Indirect Argument
    - Contradiction
    - Contraposition
- Mathematical Induction
  - Well-ordering Principle
  - Mathematical Induction
  - Recursive definition
- Recursive Algorithms
- Program Correctness
- Asymptotic notations

INTRODUCTION

Show that we will be home by sunset.

It is not sunny this afternoon and it is colder than yesterday. We will go swimming only if it is sunny. If we do not go swimming, then we will take a canoe trip. If we take a canoe trip, then we will be home by sunset".

Let p be "It is sunny this afternoon",
q be "It is colder than yesterday",
r be "We will go swimming",
s be "We will take a canoe trip",
t be "We will be home by sunset".

1	$\neg p \wedge q$	Hypothesis
2	$\neg p$	Simplification using step1
3	$r \rightarrow p$	Hypothesis
4	$\neg r$	Modus tollens using step 2 and 3.
5	$\neg r \rightarrow s$	Hypothesis
6	s	Modus ponens using step 4 and 5.
7	$s \rightarrow t$	Hypothesis
8	t	Modus ponens using step 6 and 7.

DIRECT PROOF

RULES OF INFERENCE

Universal instantiation
$\forall x P(x)\ \therefore P(c)$ if $c \in U$.
Universal generalization
$P(c)$ for an arbitrary $c \in U \therefore \forall x P(x)$
Existential instantiation
$\exists x P(x) \therefore P(c)$ for some element $c \in U$
Existential generalization
$P(c)$ for some element $c \in U \therefore \exists x P(x)$

DIRECT PROOF

Proving existential statements

∃ x in $\mathcal{D}$ such that Q(x) is true

If and only if

Q(x) is true at least one x in $\mathcal{D}$.

One way to prove this is to find an x in $\mathcal{D}$ that makes Q(x) true.

DIRECT PROOF

Proving existential statements

∃ x in $\mathcal{D}$ such that P(x)→Q(x) is true

If and only if

P(x)→Q(x) is true at least one x in $\mathcal{D}$.

One way to prove this is to find an x in $\mathcal{D}$
that makes P(x)→Q(x) true.

DIRECT PROOF

Proving existential statements

Example

Prove the following: there exists an integer x such that it can be written into two ways as a sum of two prime numbers.

Proof: Let x = 10.
Since 10 = 5+5 and 10 = 3+7 and 3, 5 and 7 are prime numbers, 10 can be written into 2 ways as a sum of two prime numbers. QED

CONSTRUCTIVE PROOFS OF EXISTENCE

DIRECT PROOF

Example

Proving existential statements

Show that there are n consecutive composite positive integers for every positive integers n.

Prove that : $\forall n\ \exists x$ (x+i is composite for i = 1 2 3 … n)

Proof: Let x = (n+1)! +1. Consider the integers

x+1, x+2, x+3, …, x+n.

Note that i+1 divides x+i = (n+1)! + (i+1) for i = 1 2 3 … n.

Hence, n consecutive composite integers have been given.

QED

CONSTRUCTIVE PROOFS OF EXISTENCE

DIRECT PROOF

Proving existential statements

Example

Show that there is a prime greater than n for every positive integer n.

Prove that : $\forall n \, \exists x$ (x is prime and $x > n$).

Proof: Consider the integer $n!+1$.

There is at least one prime divides $n!+1$.

Note that $n!+1 \equiv 1 \pmod{k}$ for k = 1 2 3 … n.

Hence, any prime factor of $n!+1$ must be greater than n.

QED

NONCONSTRUCTIVE PROOFS OF EXISTENCE

DIRECT PROOF

Proving existential statements

Example

Prove the following: there exists an integer x such that

If x is divisible by 3 then $5x^2$ is divisible by 6.

Proof: Let x = 6.

6 is divisible by 3 and

$5 \times 6^2 = 180$.

180 is divisible by 6. QED

DIRECT PROOF

Proving universal statements

$\forall$x in $\mathcal{D}$, if P(x) then Q(x).

DIRECT PROOF

Proving universal statements

Example

If n is an even integer between 6 and 20,

then n can be written as a sum of two prime numbers.

Proof:

6 = 3+3	**8 = 3+5**	**10 = 3+7**	**12 = 5+7**
14 = 7+7	**16 = 11+5**	**18 = 11+7**	**20 = 13+7**

QED

Since $\mathcal{D}$ is finite (or finite number of elements in $\mathcal{D}$ satisfied P),

this statement can be proved by the

method of exhaustion.

DIRECT PROOF

Proving universal statements

Proof by cases

To prove that $(p_1 \vee p_2 \vee p_3 \ldots \vee p_n) \rightarrow q$.
This can be shown by prove that
$(p_1 \rightarrow q) \wedge (p_2 \rightarrow q) \wedge (p_3 \rightarrow q) \wedge \ldots \wedge (p_n \rightarrow q)$

DIRECT PROOF

Proving universal statements

EXAMPLE

Prove that
"If n is not divisible by 3, then $n^2 \equiv 1 \pmod 3$".
Proof:
Suppose that n is not divisible by 3.
Case 1 : $n \equiv 1 \pmod 3$.
So $n = 3k+1$. Since $n^2 = 3(3k^2+2k)+1$.
Hence $n^2 \equiv 1 \pmod 3$.
Case 2 : $n \equiv 2 \pmod 3$.
So $n = 3k+2$. Since $n^2 = 3(3k^2+4k+1)+1$
Hence $n^2 \equiv 1 \pmod 3$. QED.

DIRECT PROOF

Proving universal statements

method of generalizing from the generic particular

TO SHOW THAT

"Every element of a domain satisfies a certain property:
Suppose x is a particular but arbitrarily chosen
element of the domain
and show that x satisfies the property".

DIRECT PROOF

Proving universal statements

METHOD OF DIRECT PROOF

- **Express the statement to be proved in the form**
  $\forall$x in $\mathcal{D}$, if P(x) then Q(x).
- **Start the proof by supposing x is a particular**
  but arbitrarily chosen element of $\mathcal{D}$
  for which the hypothesis P(x) is true
- **Show that Q(x) is true**
  by using definitions, previously established results, and the rules for logical inference.

DIRECT PROOF

Proving universal statements

Example

Prove that if the sum of any two integers is even, then so is their difference.

Proof:

- $\forall$x,y in Z, if x+y is even then x-y is even.
- Suppose that x+y = 2k for some integer k.
- Show that x-y is even. QED

DIRECT PROOF

Proving universal statements

Example

Prove that if the sum of any two integers is even, then so is their difference.

Proof:

- $\forall$x,y in Z, if x+y is even then x-y is even.
- Suppose that x+y = 2k for some integer k.

$x+y = 2k$; $x = 2k-y$; $x-y = 2k-2y = 2(k-y)$

- Then x-y is even. QED

DIRECT PROOF

Proving universal statements

Trivial proof

Suppose that the conclusion of $p \rightarrow q$ is true. Then this statement is true since it has the form $T \rightarrow T$ or $F \rightarrow T$. Hence, it we can show that q is true, then a proof can be given.

EXAMPLE

P(n) is the proposition "For two positive integers a,b, if a<b then $a^n \leq b^n$".

Show that P(0) is true.

Proof: $a^0 = b^0 = 1$.Then P(0) is true. QED

DIRECT PROOF

Proving universal statements

Vacuous proof

Suppose that the hypothesis of $p \rightarrow q$ is false. Then this statement is true since it has the form $F \rightarrow T$ or $F \rightarrow F$. Hence, if we can show that p is false, then a proof can be given.

EXAMPLE

P(n) is the proposition "if n>1 then $n^2>n$".

Show that P(0) is true.

Proof: Since 0<1, P(0) is true. QED

DIRECT PROOF

Proving universal statements

Fallacies

- Fallacy of affirming the conclusion
- Fallacy of denying the hypoythesis.

EXAMPLEs. $P(n) \rightarrow Q(n)$

We have that Q(n) is true. Conclusion : P(n)

We have that $\neg$P(n) is true. Conclusion : $\neg$Q(n).

DISPROOF BY COUNTEREXAMPLE

Consider the question of disproving a statement of the form

$\forall$x in $\mathcal{D}$, if P(x) then Q(x).

Showing that this statement is false is equivalent to showing that its negation is true.

Show that
$\exists$x in $\mathcal{D}$, P(x) and $\neg$Q(x) are true.

DISPROOF BY COUNTEREXAMPLE

EXAMPLE

For any real numbers a and b,

If $a^2 = b^2$ then $a = b$.

Since 5,-5 are real numbers

$5^2 = 25$ and $(-5)^2 = 25$, but $5 \neq -5$.

INDIRECT ARGUMENT

- Method of contradiction
- Method of contraposition

INDIRECT ARGUMENT

Method of contradiction

1. Suppose the statement to be proved is false.
2. Show that this supposition leads logically to a contradiction.
3. Conclude that the statement to be proved is true.

INDIRECT ARGUMENT

Method of contradiction

EXAMPLE

The sum of any rational number and irrational number is irrational.

Proof: Let r= a/b be a rational number (a,b are integers), and let s be an irrational number.
Suppose that r+s is a rational number.
Then, r+s = c/d which c and d are integers.
We have that
r+s = a/b+s = c/d implies s = a/b-c/d = (bc-ad)/bd.
This implies that s is a rational number.
This contradicts the supposition that s is irrational. QED

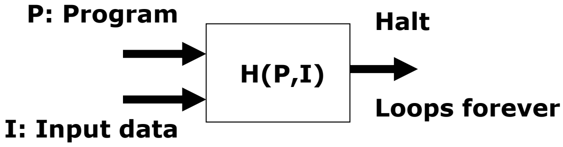
INDIRECT ARGUMENT

Method of contradiction

EXAMPLE

The Halting problem is unsolvable problem.

H should be able to determine if P will halt when it is given a copy of itself as input.



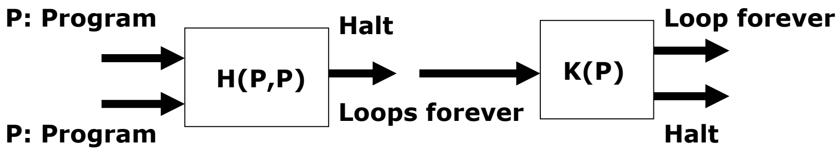
INDIRECT ARGUMENT

Method of contradiction

EXAMPLE

The Halting problem is unsolvable problem.

H should be able to determine if P will halt when it is given a copy of itself as input.



Consider if it is given K as input.

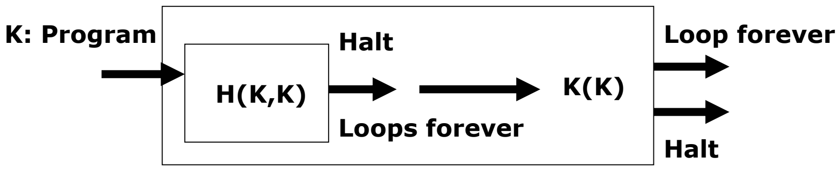
INDIRECT ARGUMENT

Method of contradiction

EXAMPLE

The Halting problem is unsolvable problem.

Contradiction:
If the output of H(K,K) is "halt", then K(K) loops forever.
If the output of H(K,K) is "loop forever", then K(K) halts.



Consider if it is given K as input.

INDIRECT ARGUMENT

Method of contradiction

EXAMPLE

The Halting problem is unsolvable problem.

Proof: Assume that H(P,I) be a machine that can determine if program P willl halt or loop forever with the inout I.
K(P,I) writes « halt » if P halts with the input I and writes « loop forever » if P loops forever with the input I.
We construct a machine K(P) that the output is « halt » if the output of H(P,P) is « loop forever », otherwise K(P) writes « loop forever ».
When K(P) writes « halt », it contradicts with H(K,K) loops.
When K(P) writes « loop forever », it contradicts with H(K,K).
This completes the proof.

QED

INDIRECT ARGUMENT

Method of contraposition

- Express the statement to be proved in the form

  $\forall x$ in $\mathcal{D}$, if P(x) then Q(x)
- Rewrite this statement in the contrapositive form

  $\forall x$ in $\mathcal{D}$, if Q(x) is false then P(x) is false.
- Prove the contrapositive by a direct proof
  1. Suppose x is an element of $\mathcal{D}$ such that Q(x) is false.
  2. Show that P(x) is false.

INDIRECT ARGUMENT

Method of contraposition

EXAMPLE

If the square of an integer is even, the integer is even.

Proof: For all integers n, if n^2 is even, then n is even.
CONTAPOSITIVE: For all integers n, if n is odd then n^2 is odd.
Since n is odd, then n = 2a + 1 where a is an integer,

$$n^2 = 4a^2 + 4a + 1 \text{ is odd.}$$

This completes the proof of the contrapositive. QED

MATHEMATICAL INDUCTION

Francesco Maurolico
1494-1575

First use of mathematical induction to prove that The sum of the first n odd positive integers equals n^2.
Gave a table of secants.

MATHEMATICAL INDUCTION

- Well-ordering principle
- Mathematical induction
- Recursive definition

MATHEMATICAL INDUCTION

A proof by mathematical induction that P(n) is true for every positive integer n consists of two steps:

- BASIC step: P(1) is shown to be true.
- INDUCTIVE step: P(n)$\rightarrow$P(n+1) is shown to be true for every positive integer n.

WHY ?

MATHEMATICAL INDUCTION

WELL-ORDERING PRINCIPLE

Every nonempty set of nonnegative integers has a least element.

MATHEMATICAL INDUCTION

Why mathematical induction is valid ?

Mathematical induction:

$$\left(P(1) \wedge \forall n\ P(n) \rightarrow P(n+1)\right) \rightarrow \forall n\ P(n).$$

We have to show that this statement is a tautology statement.

MATHEMATICAL INDUCTION

Why mathematical induction is valid ?

Proof: Suppose we know that P(1) is true and that P(n)→P(n+1) is true for all positive integers n.

To show that P(n) must be true for all positive integer n, assume that there is at least one positive integer for which P(n) is false.

Let S be the set of positive integers for which P(n) is false.

S is nonempty set.

By well-ordering principle, let k be the least element of S.

So P(k-1) must be true, and P(k) is false.

Since P(n)→P(n+1) and P(k-1) are true, P(k) is true.

This contradicts the proof that P(k) is false. QED

MATHEMATICAL INDUCTION

Example: Show that the sum of the first n odd positive integers is n^2.

Proof: We have to prove that

$$P(n) : 1+3+5+...+(2n-1)=n^2.$$

It is clear that P(1) is true. Suppose that P(n) is true.

But $P(n+1) = 1+3+5+...+(2n+1)$

$= n^2+2n+1$

$= (n+1)^2$.

This completes the proof. QED

MATHEMATICAL INDUCTION

Example: Show that a set with n element has 2^n subsets.

Proof: Let P(n) be the proposition that a set with n elements has 2^n subsets.

It is clear that P(1) is true. Suppose that P(n) is true.

Let T be the set with n+1 elements.

Write $T=S\cup\{a\}$ where a is an element of T and

$T-S=\{a\}$. S has 2^n subsets. For each subset X of S,

The subset of T can be obtained by X and $X\cup\{a\}$.

Then T has $2\times2^n=2^{n+1}$.

This implies that P(n+1) is true.

This completes the proof. QED

MATHEMATICAL INDUCTION

Example: Show that Harmonic numbers
$H_k \geq 1 + (n/2)$ where $k = 2^n$.

$H_x = 1 + (1/2) + (1/3) + \ldots + (1/x)$.

Proof: Let P(n) be $H_k \geq 1 + (n/2)$ for every integers n.

P(0) is true, since $H_1 = 1 \geq 1 + (0/2) = 1$.

Assume that for all n, $P(n) \rightarrow P(n+1)$ is true.

Suppose P(n) is true. But

$$
\begin{aligned}
H_{k+1} &= H_k + (1/2^n+1) + (1/2^n+2)+\ldots+(1/2^{n+1}) \\
&\geq (1+(n/2)) + (1/2^n+1)+\ldots+(1/2^{n+1}) \\
&\geq (1+(n/2)) + 2^n(1/2^{n+1}) \\
&= (1+(n/2))+(1/2) \\
&= 1+((n+1)/2)
\end{aligned}
$$

This completes the proof. QED

MATHEMATICAL INDUCTION

The second principle of mathematical induction

Basic step:

P(1) is shown to be true.

Induction step:

$(P(1) \wedge P(2) \wedge P(3) \wedge \ldots \wedge P(n)) \rightarrow P(n+1)$ is shown to be true for every positive integer n.

MATHEMATICAL INDUCTION

The second principle of mathematical induction

Example: Show that if n is an integer greater than 1, then n can be written as the product of primes.

Proof: n=2 is true. Assume that for n=2,3,…,k, we have that n can be written as the product of primes.

Case: n+1 is prime.

Case: n+1 is composite. Then n=ab, $2 \leq a \leq b < n+1$.
By the induction hypothesis, both a and b can be written as the product of primes.
This completes the proof. QED

MATHEMATICAL INDUCTION

RECURSIVE DEFINITION

Model that Outcome processes

- occur repeatedly
- according to definite patterns

EXAMPLE
At the end of each month, John can earn n^2 US$ plus if he has n US$ at the beginning of the month.

MATHEMATICAL INDUCTION

RECURSIVE DEFINITION

The first month	1	US$
The 2^{nd} month	2	US$
The 3^{rd} month	6	US$
The 4^{th} month	42	US$
...		

EXAMPLE

At the end of each month, John can earn n^2 US$ plus if he has n US$ at the beginning of the month.

$$P(n) = P(n-1)^2 + P(n-1)$$

P(n) = US$ he has at the end of the n^{th} month, with P(1) = 1.

MATHEMATICAL INDUCTION

RECURSIVE DEFINITION

RECURSIVELY DEFINED FUNCTIONS

To define a function with the set of nonnegative integers as its domain,

- Specify the value of the function at zero.
- Give a rule for finding its value as an integer from its values at smaller integers.

MATHEMATICAL INDUCTION

RECURSIVE DEFINITION

EXAMPLE

$f(0) = 3$
$f(n+1) = 2f(n) + 3.$

$f(1) = 2f(0)+3 = 6+3 = 9$
$f(2) = 2f(1)+3 = 18+3 = 21$
$f(3) = 2f(2)+3 = 42+3 = 45$
$f(4) = 2f(3)+3 = 90+3 = 93$
…

MATHEMATICAL INDUCTION

RECURSIVE DEFINITION

EXAMPLE

Given an inductive definition of the factorial function.

$F(n) = n!$

Define the initial value $F(0) = 1$, since $0!=1$.

Find $F(n+1)$ from $F(n)$.

$$\begin{aligned} F(n+1) &= (n+1)! \\ &= (n+1)\, n! \\ &= (n+1)\, F(n). \end{aligned}$$

MATHEMATICAL INDUCTION

RECURSIVE DEFINITION

EXAMPLE

Let $f(0)=0$, $f(1)=1$ and $f(n)=f(n-1)+f(n-2)$, for $n = 2, 3, 4, \dots$

$f(2) = f(1)+f(0) = 1+0 = 1$
$f(3) = f(2)+f(1) = 1+1 = 2$
$f(4) = f(3)+f(2) = 2+1 = 3$
$f(5) = f(4)+f(3) = 3+2 = 5$
$f(6) = f(5)+f(4) = 5+3 = 8$
$f(7) = f(6)+f(5) = 8+5 = 13$
…

MATHEMATICAL INDUCTION

RECURSIVE DEFINITION

EXAMPLE

Show that $f(n) > \alpha^{n-2}$ where $\alpha=(1+5^{1/2})/2$, for every $n \geq 3$.

Proof: $f(3) = 2 > \alpha$ and $f(4) = 3 > (3+5^{1/2})/2 = \alpha^2$.
Since α is a solution of $x^2-x-1=0$, $\alpha^2 = \alpha+1$.
Therefore, $\alpha^{n-1}= (\alpha+1)\,\alpha^{n-3} = \alpha^{n-2}+ \alpha^{n-3}$.
For $n \geq 5$, $f(n-1)> \alpha^{n-3}$ and $f(n)> \alpha^{n-2}$.
We have that $f(n+1)=f(n)+f(n-1) > \alpha^{n-2}+ \alpha^{n-3}= \alpha^{n-1}$.
This completes the proof. QED

MATHEMATICAL INDUCTION

RECURSIVE DEFINITION

Lamé Theorem

gcd(a,b) by Euclidean algorithm, the number of divisions ≤ 5k where $b < 10^k$ and $a \geq b$.

Find gcd(a,b) where $a \geq b$.

Let $a = r_0$ and $b = r_1$.

Rewrite
$r_0 = r_1 q_1 + r_2$ where $0 \leq r_2 < r_1$
$r_1 = r_2 q_2 + r_3$ where $0 \leq r_3 < r_2$
$r_2 = r_3 q_3 + r_4$ where $0 \leq r_4 < r_3$
…
$r_{n-2} = r_{n-1} q_{n-1} + r_n$ where $0 \leq r_n < r_{n-1}$
$r_{n-1} = r_n q_1$
gcd(a,b) = r_n and
The number of divisions = n.

MATHEMATICAL INDUCTION

RECURSIVE DEFINITION

Lamé Theorem

gcd(a,b) by Euclidean algorithm, the number of divisions ≤ 5k where $b < 10^k$ and $a \geq b$.

Then $r_1 > r_2 > r_3 > \ldots > r_{n-1} > r_n \geq 1 = f(2)$.
$r_{n-1} \geq 2r_n \geq 2f(2) = f(3)$
$r_{n-2} \geq r_n + r_{n-1} \geq f(2)+f(3) = f(4)$
$r_{n-3} \geq r_{n-1} + r_{n-2} \geq f(3)+f(4) = f(5)$
…
$r_2 \geq r_3 + r_4 \geq f(n-1)+f(n-2) = f(n)$.
$b = r_1 \geq r_2 + r_3 \geq f(n)+f(n-1) = f(n+1)$
$> \alpha^{n-1}$.

MATHEMATICAL INDUCTION

RECURSIVE DEFINITION

Lamé Theorem

gcd(a,b) by Euclidean algorithm, the number of divisions $\leq 5k$ where $b < 10^k$ and $a \geq b$.

$b > \alpha^{n-1}$.

Since $\log_{10}\alpha \sim 0.203 > 1/5$.

$\log_{10}b > \log_{10}\alpha^{n-1} > (n-1)/5$.

Let $b < 10^k$.

$(n-1)/5 < k$.

Since n is an integer, then $n-1 \leq 5k$.

This completes the proof. QED.

MATHEMATICAL INDUCTION

RECURSIVE DEFINITION

Example: Let S be defined recursively by

- $3 \in S$
- if $x,y \in S$, then $x+y \in S$.

Show that S is the set of positive number divisible by 3.

Proof: Let A be a set of positive integer divisible by 3. We show that S=A or $A \subset S$ and $S \subset A$.

We have to show $A \subset S$. Let P(n) be "3n belongs to S".

Since 3 is in S, it is clear that P(1) is true.

Assume that $\forall n\ P(n) \rightarrow P(n+1)$. Suppose P(n) is true.

But P(n+1) = 3(n+1) = P(n)+3 which is in S.

then P(n+1) is true.

MATHEMATICAL INDUCTION

RECURSIVE DEFINITION

Example: Let S be defined recursively by

- $3 \in S$
- if $x,y \in S$, then $x+y \in S$.

Show that S is the set of positive number divisible by 3.

Proof: Now we show that $S \subset A$.
It is clear that $3|3$, then 3 is in A.
We show that if $x \notin A$, then $x \notin S$.
Let $x \notin A$, then $3 \nmid x$. So for all integers k, $x \neq 3k = \underbrace{3+3+\ldots+3}_{\textbf{k times}}$.
Since all elements in S excluded 3 are generated by the second rules, they should be written in the form of the sum of 3. This shows that $x \notin S$. This completes the proof. QED

MATHEMATICAL INDUCTION

PROBLEM

GOLDBACH's conjecture
Christian Goldbach
1690-1764

Every even positive integer greater than 4 is the sum of two primes.

No counterexample has been found, although it has been verified for all even positive integers up to 10^{14}.

Recursive AlgorithmS

We reduce the solution to a problem with a particular set of input to the solution of the same problem with smaller input values.

Recursive AlgorithmS

Definition

An algorithm is called recursive if it solves a problem by reducing it to an instance of the same problem with smaller input.

Example: Given a positive integer n, find a^n.
By recursively definition of an, $a^n = a^{n-1} \times a$.
Then an algorithm Power(a,n) can be solved by Power(a,n-1) and so on.

```
Power(a,n)
If (n=1)     then Power(a,1) := 1
             else Power(a,n) := Power(a,n-1)×a
end.
```

RecursivE AlgorithmS

Example : Binary search algorithm

IS 14 IN THIS TABLE ?

2	5	7	8	11	14	15	18

RecursivE AlgorithmS

Example : Binary search algorithm

IS 14 IN THIS TABLE ?

14 ≥11 YES

2	5	7	8	11	14	15	18

RecursiveAlgorithmS

Example : Binary search algorithm

IS 14 IN THIS TABLE ?

14 ≥11 YES
14 ≥15 NO

2	5	7	8	11	14	15	18

RecursiveAlgorithmS

Example : Binary search algorithm

IS 14 IN THIS TABLE ?

14 ≥11 YES
14 ≥15 NO
14 ≥14 YES

2	5	7	8	11	14	15	18

RecursiveAlgorithmS

Example : Binary search algorithm

```
BINARYSEARCH(x,s,e)
begin
        m := ⌈ (s+e)/2 ⌉
        if x = m then  return m.
        if s = e then return 0.
        if x > m then  s := m
                else     e := m.
        return BINARYSEARCH(x,s,e)
end
```

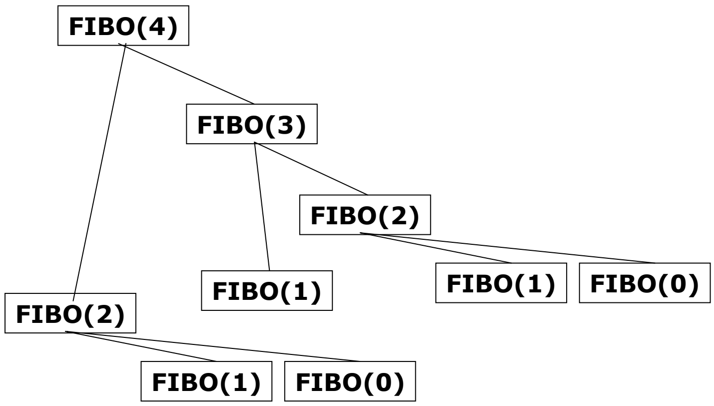
RecursiveAlgorithmS

Example : Find n^{th} Fibonacci number

```
FIBO(n)
begin
        if n = 0 then return 0.
        if n = 1 then return 1.
        return FIBO(n-1)+FIB(n-2).
end
```

Recursive AlgorithmS

Example : Find nth Fibonacci number



Recursive AlgorithmS

Example : Find nth Fibonacci number

```
class Fibo{
public static void main(String[] args){
        int x = 0; int y = 1; System.in.read(int n);
        if (n=0) System.out.println(0);
        if (n=1) System.out.println(1);
        if (n>1)
        for (int j = 1; j < n) {
                z = x + y; x = y; y = z}
        System.out.println(z);
        }
}
```

ProgramVerificatioN

PROGRAM IS CORRECT

ANSWER IS OBTAINED.

INITIAL ASSERTION

FINAL ASSERTION

PROGRAM ALWAYS TERMINATES.

PARTIAL CORRECTNESS

ProgramVerificatioN

DEFINITION

A program or program segment S is said to be partially correct with respect to the initial assertion p and the final assertion q if whenever p is true for the input values of S and S terminates, the q is true for the output values of S, denoted by p{S}q, called *Hoare triple*.

EXAMPLE

Let p:x=1, and q:z=3 with S={y:=2; z:=x+y;}.

Thus p{S}q is true.

Program Verification

DEFINITION

A program or program segment S is said to be partially correct with respect to the initial assertion p and the final assertion q if wherever p is true for the input values of S and S terminates, the q is true for the output values of S, denoted by p{S}q, called *Hoare triple*.

RULE OF INFERENCE (composition rule)

$p\{S\}q \ ; \ q\{R\}r \ \therefore \ p\{S;R\}r$

Program Verification

CONDITION STATEMENT

STATEMENT: if (condition) S.

RULE OF INFERENCE

$(p\wedge \text{condition})\{S\}q$;

$(p\wedge\neg \text{condition})\rightarrow q$;

$\therefore$ p{if(condition)S}q.

EXAMPLE Show that p{if(x<0) x:=0}q where $q: x\geq 0$.

In the case that $p: x<0$, we have $p\wedge(x<0)$ is true.

$(p\wedge(x<0))\{S\}q$ is true.

In the case that $p: x\geq 0$, we have that $p\wedge\neg(x<0)$ is true.

$(p\wedge\neg(x>0))\rightarrow q$ is true.

Thus p{if(condition) S}q is true.

Program VerificatioN

CONDITION STATEMENT

STATEMENT: if (condition) S_1 else S_2.

RULE OF INFERENCE

$(p \wedge condition)\{S_1\}q$;
$(p \wedge \neg condition)\{S_2\}q$;
$\therefore$ p{if(condition)S_1 else S_2}q.

EXAMPLE Show that p{if(x<0) y:=-x else y:=x}q where q:y≥0.

In the case that p:x<0, we have $p \wedge (x<0)$ is true.

$(p \wedge (x<0))\{S_1\}q$ is true.

In the case that p:x≥0, we have that $p \wedge \neg(x<0)$ is true.

$(p \wedge \neg(x>0))\{S_2\}q$ is true.

Thus p{if(condition) S_1 else S_2}q is true.

Program VerificatioN

LOOP INVARIANTS

STATEMENT: while (condition) S.

RULE OF INFERENCE

$(p \wedge condition)\{S\}p$;
$\therefore$ p{while(condition) S}$(\neg condition \wedge p)$.

EXAMPLE

i :=1; factorial := 1;
while i < n { ++i ; factorial:=factorial*1}

Let p:factorial=i! and i≤n.

We have that $(p \wedge (i \leq n))\{S\}p$ is true.

We can conclude that p{while(condition) S}$(\neg condition \wedge p)$ is true.

Asymptotic notationS

It is the asymptotic complexity of an algorithm which ultimately determines the size of problems that can be solved by the algorithm.

Asymptotic notationS

Little-o notation	o
Little-omega notation	ω
Theta notation	Θ
Big-O notation	O
Big-omega notation	Ω

Little-o notation

Informally,

saying some equation $f(n) = o(g(n))$ means $f(n)$ becomes insignificant relative to $g(n)$ as n approaches infinity.

More formally

it means for all $c > 0$, there exists some $k > 0$ such that

$$0 \le f(n) < cg(n) \text{ for all } n \ge k.$$

The value of k must not depend on n, but may depend on c.

Note:

As an example, $f(n) = 3n + 4$ is $o(n^2)$ since for any c we can choose $k > (3+ (9+16c))/2c$.

$3n + 4$ is not $o(n)$. $o(f(n))$ is an upper bound.

That is

$$\mathbf{o(g(n)) = \{f(n) \mid lim_{n\to\infty}(f(n)/g(n))=0\}.}$$

ω notation

Informally,

saying some equation $f(n) = (g(n))$ means $g(n)$ becomes insignificant relative to $f(n)$ as n goes to infinity.

More formally,

it means that for any positive constant c, there exists a constant k, such that

$$0 \le cg(n) < f(n) \text{ for all } n \ge k.$$

The value of k must not depend on n,
but may depend on c.

That is

$$\boldsymbol{\omega(g(n)) = \{f(n) \mid lim_{n\to\infty}(f(n)/g(n))=\infty \}.}$$

Θ notation

Informally,

saying some equation $f(n) = \Theta(g(n))$ means it is within a constant multiple of $g(n)$.

More formally,

it means there are positive constants c1, c2, and k, such that

$$0 \le c_1 g(n) \le f(n) \le c_2 g(n) \text{ for all } n \ge k.$$

The values of c1, c2, and k must be fixed for the function f and must not depend on n.

That is

$$\Theta(g(n)) = \{f(n) \mid \lim_{n\to\infty}(f(n)/g(n))=c, c\neq 0, c\neq\infty\}.$$

Big-O notation

Informally,

saying some equation $f(n) = O(g(n))$ means it is less than some constant multiple of $g(n)$.

More formally

it means there are positive constants c and k, such that

$$0 \le f(n) \le cg(n) \text{ for all } n \ge k.$$

The values of c and k must be fixed for the function f and must not depend on n.

That is

$$O(g(n)) = o(g(n)) \cup \Theta(g(n)).$$

Ω notation

Informally,

saying some equation $f(n) = \Omega(g(n))$ means it is more than some constant multiple of $g(n)$.

More formally,

it means there are positive constants c and k, such that

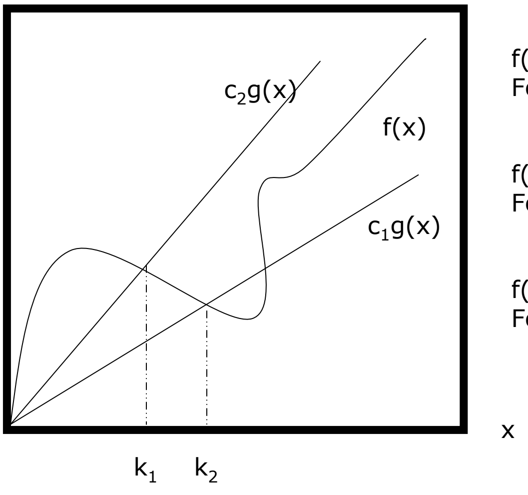
$$0 \le cg(n) \le f(n) \text{ for all } n \ge k.$$

The values of c and k must be fixed for the function f and must not depend on n.

That is

$$\mathbf{\Omega(g(n))} = \omega(g(n)) \cup \mathbf{\Theta(g(n))}.$$

Asymptotic Notations



$f(x) = \Omega(g(x))$
For all $x \ge k_2$,
$f(x) \ge c_1g(x)$.

$f(x) = O(g(x))$
For all $x \ge k_1$,
$f(x) \le c_2g(x)$.

$f(x) = \Theta(g(x))$
For all $x \ge k_1, k_2$,
$c_1g(x) \le f(x) \le c_2g(x)$.

Asymptotic Notations

SPECIAL ORDERS OF GROWTH

constant	: $\Theta(1)$
logarithmic	: $\Theta(\log n)$
polylogarithmic	: $\Theta(\log^c n)$, $c \geq 1$
sublinear	: $\Theta(n^a)$, $0 < a < 1$
linear	: $\Theta(n)$
quadratic	: $\Theta(n^2)$
polynomial	: $\Theta(n^c)$, $c \geq 1$
exponential	: $\Theta(c^n)$, $c > 1$