

GB13604 - Maths for Computer Science

Lecture 1 – Introduction to Proofs

Claus Aranha
caranha@cs.tsukuba.ac.jp

College of Information Science

2020-10-07

Last updated October 1, 2020

Lecture 1 – Outline

In this lecture, we introduce the concept of **mathematical proofs**:

- **Section 1:** What are proofs, and why we need them;
- **Section 2:** Basic proof methods;
- **Section 3:** The Well Ordering Principle (WOP);
(an important proof method)
- **Section 4:** Logical formulas and satisfiability;

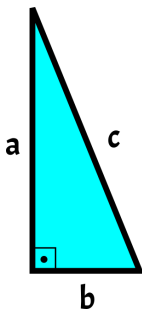
This lecture covers the textbook's chapters 1, 2 and 3.

Part 1: Introduction to Proofs.

- 1 Introduction to Proofs
- 2 Proof Methods
- 3 Well Ordering Principle
- 4 Logical Formulas

What is a proof?

Some concepts are easy to understand, but not easy to show that they are true.



- Pythagoras Theorem:

$$a^2 + b^2 = c^2$$

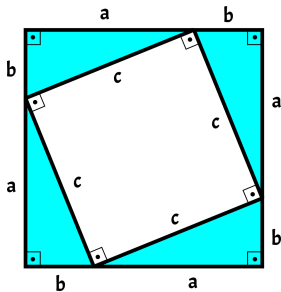
- It is easy to show this is true for **any one triangle**.
- But how do you show it is true for **all** triangles?

The proof of the Pythagoras theorem is not obvious: there are more than 100 different proofs!

What is a proof?

One Pythagoras Proof

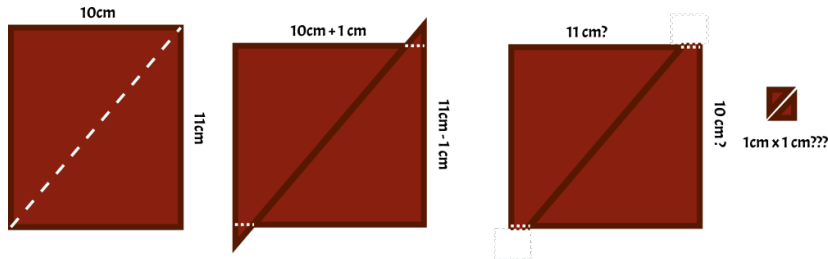
- **Proof:** by geometric construction
- Arrange four identical triangles;
- Show that internal angles are right;
- Internal square area: c^2
- External square area: $(a + b)^2$
- $(a + b)^2 = c^2 + 4(\text{area triangle})$
- $(a + b)^2 = c^2 + 4(\frac{ab}{2})$
- $a^2 + 2ab + b^2 = c^2 + 2ab$
- $a^2 + b^2 = c^2$



Remember: There are many other possible proofs. (Beautiful proofs, short proofs, wrong proofs, etc.)

False Proofs

Infinite Chocolate!



- What is wrong with the proof above?
- **Be careful!** A false proof can have many correct steps, and **only one** impossible step.

We can use proofs to show that something is **incorrect** as well!

Proofs and Computer Science

Why are proofs important for Computer Science?

- We don't use proofs only for mathematical equations.
- We can use proofs to **show that a program is correct**. (or incorrect)
- Example cases:
 - Use proofs to show that the result of a program is correct for any input;
 - Use proofs to show that one type of input will cause a bug in the program;
 - Use proofs to show that a program finishes in N steps;

Proofs and Computer Science

Example:

- Is the program below correct or incorrect?
- Can you show by using a proof?

```
int triangle_type(int a, int b, int c)
// a, b, c are the length of the sides of a triangle
if (a == b)
    if (b == c)
        return "all sides are equal";
    else
        return "two sides are equal";
else if (b == c)
    return "two sides are equal";
else
    return "all sides are different";
```


Next Part:

Proof Techniques: How do we prove something?

- 1 Introduction to Proofs
- 2 Proof Methods**
- 3 Well Ordering Principle
- 4 Logical Formulas

What is a Proof?

- A **proposition** is a statement that can be **True** or **False**.
 - This room has 40 chairs.
 - Every intelligent being feels pain.
 - Please say your name.
 - $513 \times 435 = 223165$
 - Every even integer greater than 2 is the sum of two primes.
 - It is raining now.
- A **proof** is a method of proving the truth or falsehood of a proposition.
 - **mathematical proofs** normally use logical steps to show the truth of a mathematical proposition.

Proof Examples

- Pitagoras by pictures
- Getting rich with triangles
- $1 == -1$

Morals of Proofs

- Make sure that you are applying the rules properly.
- Mindless calculation does not replace understanding.

Common Terms used in Proofs

- Proposition:
- Predicate:
- Axiom:
- Proof:
- Theorem:
- Lemma:
- Corollary:

Common Terms used in Proofs

- **Proposition:**
A statement that is either true or false
- **Predicate:**
A preposition that depends on variables
- **Axiom:**
A preposition that is accepted to be true
- **Proof:**
A sequence of axioms and proved statements that conclude with the proposition of interest
- **Theorem:**
An important true proposition
- **Lemma:**
A simpler proposition that is useful to prove later propositions
- **Corollary:**
A proposition that follows from a theorem in a few logical steps

Our first proof method: Modus Ponens

$$\frac{P, P \text{ implies } Q}{Q} \text{ or } \frac{P, P \rightarrow Q}{Q}$$

What does “Modus Ponens” mean?

- If P is true.
- and if P being true **requires** that Q is true too.
- then Q is true.

How can we use Modus Ponens to prove something?

- We want to prove Q.
- Prove that when P is true, Q **must** be true
- Prove that P is true
- **therefore**, Q must be true.

Proof By Contradiction

A trivial proof:

$$\sqrt[3]{1332} \leq 11$$

Proof By Contradiction

If an assertion **implies something false**

Then the **assertion must be false!**

Better Example: $\sqrt{2}$ is irrational

Think a little bit by yourselves first.

Better Example: $\sqrt{2}$ is irrational

Let's prove by contradiction:

- 1 Assume that $\sqrt{2}$ is rational
- 2 Therefore $\sqrt{2} = \frac{m}{n}$, and m and n are integers with **no common prime factors** ($n \neq 0$).
- 3 Therefore $n\sqrt{2} = m$, $2n^2 = m^2$, and m^2 is even.
- 4 If m^2 is even, then m is even too. $m = 2k$ for some integer k .
- 5 Therefore $2n^2 = (2k)^2$, $2n^2 = 4k^2$, $n^2 = 2k^2$, and n^2 is even.
- 6 If n^2 is even, then n is even too. **n and m are both even (contradiction).**

Proof By Cases

Prove that these two code samples are the same:

Code 1

```
If (X > 0 OR (X <= 0 AND Y > 100))  
    print("Hello!")
```

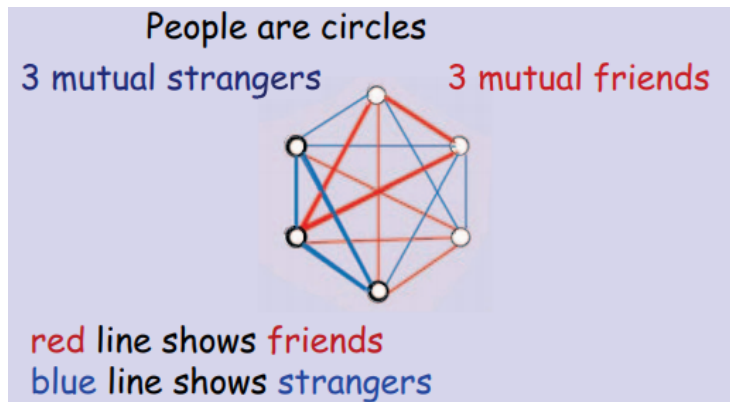
Code 2

```
If (X > 0 OR Y > 100)  
    print("Hello!")
```

Proof By Cases

- “Proof by Cases” breaks a complicated problem into easier, smaller sub-problems.
- It is important to make sure that **the cases cover all possibilities**, or the proof is not complete.

Proof By Cases: Friends and Strangers



- Six people, every two are either friends or strangers.
- **Claim:** There is always a set of 3 mutual friends or 3 mutual strangers.

Friends and Strangers, and Ramsey's Theorem

For any k , every **large enough group** of people will contain k mutual friends OR k mutual strangers.

- $R(3) = 6$
- $R(4) = 18$
- $R(5) = \text{unknown!}$

A bogus proof by cases: Prove $2a^2 > a$

- 1 This proof is by case analysis.
- 2 There are two cases:
 - Case 1: a is positive
 - Case 2: a is negative
- 3 One of these cases must always hold, because an integer is either positive or negative.
- 4 Case 1: Suppose a is positive.
- 5 Since a is an integer, we must have that $a \geq 1$.
- 6 Hence, $2a^2 = 2a \times a \geq 2a \times 1 > a$.
- 7 This implies the claim holds in Case 1.
- 8 Case 2: Suppose a is negative.
- 9 Since a is an integer, we must have that $a \leq -1$.
- 10 Hence, $2a^2 \geq 2 \times (-1 \times -1) = 2 > -1 \geq a$.
- 11 This implies the claim holds in Case 2.
- 12 The claim therefore holds in both cases.

- 1 Introduction to Proofs
- 2 Proof Methods
- 3 Well Ordering Principle**
- 4 Logical Formulas

The Well Ordering Principle

- It is a very obvious (but very useful) principle in Mathematics;
- It is so obvious that you have already used it without knowing;

The Well Ordering Principle

Every non-empty set of
Non-negative Integer Numbers
has one **smallest element**

The Well Ordering Principle

Obvious? **yes**

Trivial? **no**

- Every non-empty set of non-negative **rational** numbers has one smallest element?
- Every non-empty set of **integers** numbers has one smallest element?

Well Ordering Examples

- What is the smallest age of the U.Tsukuba students?
- What is the smallest number of cells in any animal?
- What is the smallest number of coins = 876 yens?

Proof $\sqrt{2}$ is irrational using well ordering

- if $\sqrt{2}$ is rational, then exist m, n so that $\sqrt{2} = \frac{m}{n}$
- We can always find $m, n > 0$ such as they have no common factors.
- **Why** always?

Proof $\sqrt{2}$ is irrational using well ordering

- Suppose that we choose the **smallest** m, n .
- Using the same idea as the previous proof, we show that both numbers must be divisible by two. ($m' = m/2, n' = n/2$)
- Now we found a number **smaller than the smallest!** (contradiction!)

More Proofs Using the Well Ordering Principle

- (Easy) Every integer $i > 1$ is a product of primes.
- (Medium) Every number is **Postal**.
 - A number n is postal if $n + 8$ can be composed of a sum of “threes” and “fives”
- (Difficult) $1 + r + r^2 + \dots + r^n = \frac{r^{n+1}-1}{r-1}$

General form for a Well Ordering proof

You want to prove that $\forall n \in \mathbb{N}, P(n)$ using WOP.

- ① Define a set of counter examples C , $C ::= \{n \in \mathbb{N} \mid \text{not } P(n)\}$
- ② Assume the minimum element of C exists, m , by WOP
- ③ Find a contradiction, for example:
 - Find a contradiction $c \in C, c < m$;
 - Show that $P(m)$ is actually True;

- 1 Introduction to Proofs
- 2 Proof Methods
- 3 Well Ordering Principle
- 4 Logical Formulas**

Propositions and Logic

Why Mathematical Language?

- Greeks carry swords or javelins.
- Greeks carry bronze or copper swords.

Mathematical Language

- Mathematical Language helps create non-ambiguous statements.
- We will not through all Logic operators here.
- However, it is important to understand that they are based on **binary** or **boolean** logic.

Mathematical Language / Binary Logic

Example: X XOR Y

X	Y	X XOR Y
TRUE	TRUE	FALSE
TRUE	FALSE	TRUE
FALSE	TRUE	TRUE
FALSE	FALSE	FALSE

- A Truth Table is a way to understand a logic operator.
- We can use logic operators to transform ambiguous natural language sentences into clear logical propositions.
 - Greeks carry bronze or copper swords.
 - Greek carry bronze sword XOR greek carry copper sword.

Binary Logic and Truth Tables

The **truth table** allows us to analyze a logical formula:

- Is it always true? Is it always false?
- Is it equivalent to another logical formula?

To analyze a formula using the truth table, I need to analyse the value of each variable.

Evaluation of a Formula

Given the following variables:

P = True, Q = True, R = False

How do we evaluate the following formula?

$\text{NOT}(\text{NOT}(P) \text{ OR } Q) \text{ AND } (R \text{ OR } (P \text{ XOR } Q))$

Comparison of Two Formulas

We can decide whether two logical formulas are **equivalent** if the final column of their truth table is identical.

For example, let's prove DeMorgan's Law:

$\text{NOT}(\text{P OR Q})$ **equiv to** $\text{NOT}(\text{P}) \text{ AND } \text{NOT}(\text{Q})$

Satisfiability and Validity

- A logic formula is **satisfiable** if it is true for **at least one** assignment.
 - A logic formula is **valid** if it is true for **all** assignments.
-
- **Satisfiable**: $\text{NOT}(B)$
 - **Not Satisfiable**: $B \text{ AND } \text{NOT}(B)$
 - **Valid**: $B \text{ OR } \text{NOT}(B)$

Checking for Validity and Satisfiability

Checking if a logic formula is satisfiable or not is a **very important problem** in CS.

But how to do it?

Alert! If you try to use a truth table, the size of the table grows with the number of variables:

- 1 variable - 2 lines
- 2 variables - 4 lines
- 10 variables - 1024 lines
- n variables - 2^n lines...

Checking for Validity and Satisfiability

- Is there an efficient way to test for satisfiability? (SAT)
- The Efficient SAT problem is equivalent to the $P=NP$ problem
- The validity problem is also related to the SAT problem.

Logic Quantifiers

- For all: \forall
- Exists: \exists

What is a Predicate?

A predicate is a proposition with variables in it:

$$P(X, Y) ::= [X + 2 = Y]$$

The truth value of a predicate depends on the values of the variables:

- $X = 1, Y = 3$, $P(X, Y)$ is True
- $X = 2, Y = 2$, $P(X, Y)$ is False

Quantifiers

- $\forall x$ – For ALL X
- $\exists y$ – There exists SOME Y

$\forall x$ works like **AND**. For example:

$$\forall x, x \in \{1, 2, 3\} | P(x) \text{ equiv } P(1) \text{ AND } P(2) \text{ AND } P(3)$$

$\exists y$ works like **OR**. For example:

$$\forall x, x \in \{1, 2, 3\} | P(x) \text{ equiv } P(1) \text{ OR } P(2) \text{ OR } P(3)$$

Quantifiers Example

For $x, y \in \mathbb{N}$ (x and y range over the integers).

$$Q(Y) ::= \exists x. x < y.$$

- $Q(3)$ is **True**. ($[x < 3]$ is T for $x = 1$)
- $Q(1)$ is **True**. ($[x < 1]$ is T for $x = 0$)
- $Q(0)$ is **False**. ($[x < 0]$ is not T for any $x \in \mathbb{N}$)

What about a simple example for \forall ?

Ordering Quantifiers

What is the difference when we order \exists and \forall ?

Example 1: Medicines

$\forall d \in \text{diseases}. \exists m \in \text{medicine}.$
 $m \text{ cures } d$

Example 2: Panacea

$\exists m \in \text{medicine}. \forall d \in \text{diseases}.$
 $m \text{ cures } d$

We need to be careful when writing mathematical notation!

Validity and Predicates

- Propositional Validity: A **proposition** is true for all truth assignments of variables.
 - Example: $(P \text{ implies } Q) \text{ OR } (Q \text{ implies } P)$
- Predicate Calculus Validity: A **predicate** is valid when it is true for all domains.
 - Example: $\forall z.[P(z) \wedge Q(z)] \rightarrow [\forall x.P(x) \wedge \forall y.Q(y)]$

Conclusion

Important Ideas from this lecture

- Proofs are sequences propositions that establish the truth or falsehood of an statement.
- Proof Techniques are organized ways to construct a proof;
 - Proof By Cases;
 - Contradiction;
 - Well Ordering Principle, etc;
- Predicate Logic use logical operators to show the truth or falsehood of a predicate;
 - Concepts of Validity and Satisfiability;
- There is a close relationship between proving an statement, and proving the correctness of a computer program

Reminder: Exercise sheet at manaba

- The homework for this lecture is on manaba;
- You have to submit your homework before the next lecture;
- The lecturer will be available for questions at the lecture time, so start the exercise during the lecture time;
- You can discuss the exercise with other students, but your homework is **individual**

Slide Credits

These slides were made by Claus Aranha, 2020. You are welcome to copy, re-use and modify this material, following the CC-SA-NC license.

These slides are based on "Mathematics For Computer Science (Spring 2015)", by Albert Meyer and Adam Chlipala, MIT OpenCourseWare. <https://ocw.mit.edu>.

Individual images in some slides might have been made by other authors. Please see the following slides for information about these cases.

Image Credits I