GB13604 - Maths for Computer Science

Lecture 2 – Proofs, Part 2

Claus Aranha
caranha@cs.tsukuba.ac.jp

College of Information Science

2018-10-10

Last updated October 9, 2018

Last Class Review

- Proofs
  - Proof by Contradiction
  - Proof by Cases
- The Well Ordered Principle
- Predicate Satisfiability (SAT)
- Predicate Validity

Exercise Discussion

For This Lecture...

Textbook Chapters 4,5,6 (and a bit of 7)

- Sets
- Induction
- State Machines

Mathematical Data Structures

The Set

Definition of Set

- The most fundamental of mathematical data types;
- A collection of mathematical objects
  - ... circular definition: what is a collection?

Examples:

- Real Numbers $\mathbb{R}$,
- Complex numbers $\mathbb{C}$,
- Empty Set $\varnothing$

More examples of Sets

- {7, “Aranha”, $\pi/2$, TRUE}
- {TRUE, 7, $\pi/2$, “Aranha”}
- {7, π} = {7, π, 7}

- Mathematical sets can mix different “types”
- Mathematical sets do not care about “order”
- Mathematical sets do not have duplicates.

Set Membership

The most fundamental property of a set is membership.

- A = {7,TRUE,π}
- $7 \in A$
- 7 is an element of *A*,
- $3 \notin A$

- $7 \in \mathbb{Z}$
- $\mathbb{Z} \in \{3, \mathbb{Z}, 7\}$
- A set can be a member of another set.

Definition of Subset

Subset

- $A \subset B$ means that every element of A is also an element of B
- $A \subset B$ equiv $\forall x, x \in A \rightarrow x \in B$
- $\mathbb{Z} \subset \mathbb{R}, \mathbb{R} \subset \mathbb{C}, \{3\} \subset \{5, 3, 7\}$

Important!

- $A \subset A$
- $\forall X$ is a set, $\varnothing \subset X$

Difference between Membership and Subset

- $3 \in \{3, 5, 6\}$
- $3 \not\subset \{3, 5, 6\}$

- $\{3\} \subset \{3, 5, 6\}$
- $\{3\} \notin \{3, 5, 6\}$

Prove that the empty set subsets everything

1. $A \subset B$ means that $\forall x, x \in A \rightarrow x \in B$
2. If $A = \varnothing$ then $x \in A$ is FALSE for $\forall x$
3. Replace "$\forall x \in A$" with FALSE
4. FALSE $\rightarrow x \in B$ is always TRUE.
   (FALSE $\rightarrow X$ is always TRUE)
5. Therefore, $\varnothing \subset B$ is TRUE $\forall B$

Predicate Definition of Set Membership

In many cases, we use a predicate to determine membership in a set. Let $P(X)$ be a predicate that defines set A. If $P(X)$ is true for a certain X, then $X \in A$.

Example 1

- $A = x \in \mathbb{N}, \{x < 12 \text{ AND } x \text{ is prime}\}$
- $A = \{2, 3, 5, 7, 11\}$

Example 2

- $B = x \in \mathbb{N}, \{x \text{ is prime AND } x + 2 \text{ is prime}\}$
- $B = \{3(5), 5(7), 11(13), 17(19), 29(31), \ldots\}$

The Power Set

The Power set of A is a special set composed of ALL subsets of A.

$$POW(A) = \forall x \subset A, x \in POW(A)$$

For example:

$$POW(\{T, F\}) = \{\{T\}, \{F\}, \{T, F\}, \varnothing\}$$

Also:

$$\mathbb{N} \in POW(\mathbb{R}), \mathbb{N} \subset \mathbb{R}, \mathbb{N} \notin \mathbb{R}$$

Operations on Sets

We can use operations on sets to create new sets:

- Union: $A \cup B \to x \in A \vee x \in B$
- Intersection: $A \cap B \to x \in A \wedge x \in B$

Union and intersection are distributive:

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$

Let's prove this.

Proof: $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

We prove this by a sequence of IFF.

1. $x \in A \cup (B \cap C)$ **iff**
2. $x \in A \vee x \in (B \cap C)$ **iff** (definition of union)
3. $x \in A \vee (x \in B \wedge x \in C)$ **iff** (definition of intersection)
4. $(x \in A \vee x \in B) \wedge (x \in A \vee x \in C)$ **iff** (distributive prop.)
5. $(x \in A \cup B) \wedge (x \in A \cup C)$ **iff** (definition of union)
6. $x \in (A \cup B) \cap (A \cup C)$ **done.** (fefintion of intersection)

Set Subtraction and Complement

- Subtraction: $A - B \rightarrow x \in A \wedge x \notin B$
- Complement: $\overline{A} = D - A$, where D is the domain.

Binary Relations

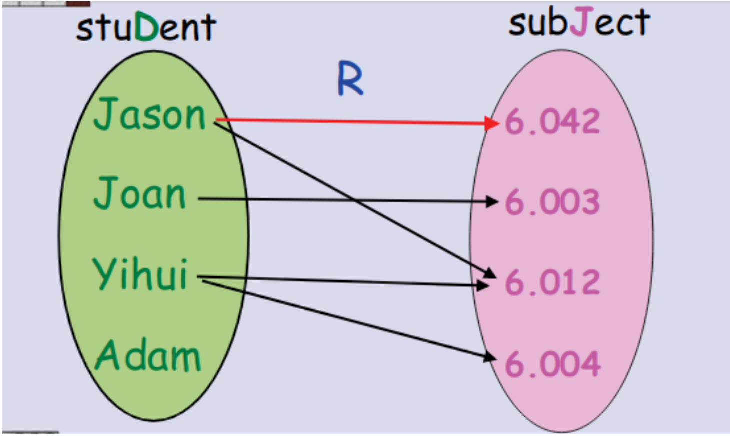
Relations and Functions

- Functions are a special case of binary relations
- Binary relations associate the elements of one set (the domain), with the elements of another set (the co-domain)

- We discussed this when talking about membership in sets (from $\mathbb{N}$ to the set of Even numbers).
- We also see relations in: Relational Databases (SQL, mySQL), counting the size of sets, and theory of computing.

Initial Example

Consider the relation Student Registered for Course – **R**



Why is this different than a function?

R – Student Registered for Course

Components of the relation:

- Domain: List of Students;
- Co-domain: List of Classes;
- Relation Graph: List of “arrows” linking students and courses.

- R(Jason) = $\{6.042, 6.012\}$
- Jason R 6.042
- $R(\{Jason, Yihui\}) = \{6.042, 6.012, 6.004\}$

Relations and Inverse Relations

Relation:

$$R(X) ::= j \in J | \exists d \in X.dRj$$

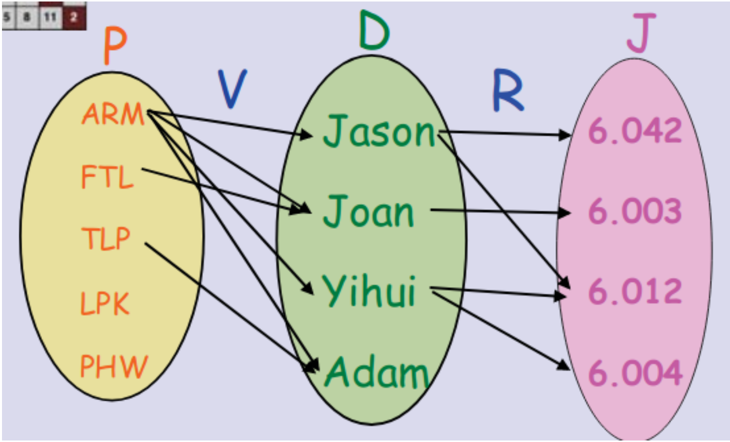
Reverse Relation:

$$R^{-1}(Y) ::= d \in S | \exists j \in Y.dRj$$

- $R(Jason) = \{6.042, 6.012\}$
- $R^{-1}(6.012) = \{Jason, Yihui\}$

Composite Relations

Let's imagine a second relation between professors and students.



Composite Relations

We can define the relation **V** in the same way that we defined **R**.

But we can also compose the two relations, **V and R**, to get the set of courses that a professor's students are enrolled:

- $R(V(X))$ or $(R \circ V)(X)$
- $R(V(FTL)) = \{6.003\}$

Binary Relations

We can classify relations depending on the number of “arrows” coming out of the domain, or coming in to the co-domain.

Classification based on the Domain

- **Total Relation**: Every element has ≥ 1 out arrows.
- **Function**: Every element has ≤ 1 out arrows.

Classification based on the Co-Domain

- **Surjection**: Every element has ≥ 1 in arrows
- **Injection**: Every element has ≤ 1 in arrows

Finally:

- **Bijection**: A relation is a **surjection function**

Binary Relations: Example

$g : \mathbb{R} \times \mathbb{R} \to \mathbb{R}.g(x, y) = 1/(x - y)$

- This is a **function** (each x,y has only one output)
- This is not a **total function** (g(x=y) is not defined)

$g_o : \mathbb{R}^2 - \{x, y | x = y\} \to \mathbb{R}.g_o(x, y) = 1/x - y)$

- g_o has the same graph (arrows) as g, but different domain.
- g_o is a total function.

Size of Finite Sets

We can use the characteristics of relations to estimate the size of sets (domains and co-domains).

- A bijection B $\rightarrow |A| = |B|$
- A function surjection B $\rightarrow |A| \geq |B|$
- A total injection B $\rightarrow |A| \leq |B|$

Set Size Example: Finite power sets and binary strings

What is the size of the Power Set of a finite set?

- Make a bijection between the power set and the binary string
- Calculate the size of a binary string
- Establish equality

Induction

An initial induction

Suppose I want to color $\mathbb{N} \geq 0$ using the following rule:

- Number 0 is red
- Any integer next to a red number is also red

Using these rules, how do the numbers look like?

Red integers using logical statements:

0,1,2,3,4,...

- $R(0)$ is True
- $R(0) \rightarrow R(1); R(1) \rightarrow R(2); R(2) \rightarrow R(3); \ldots$
- $R(n) \rightarrow R(n+1);$

We can summarize that as:

$$\frac{R(0), \forall n. R(n) \rightarrow R(n+1)}{\forall m. R(m)}$$

Example Induction Proof

$$1 + r + r^2 + r^3 + \ldots + r^n = \frac{r^{n+1} - 1}{r - 1}$$

(for $r \neq 1$)

- First Step: Prove $P(0)$
- Second Step: Prove that $P(n) \rightarrow P(n+1)$

Proof by induction on n

First Step: Prove $P(0)$

- $P(0) = r^0 = 1$
- $P(0) = \frac{r^{0+1}-1}{r-1} = \frac{r-1}{r-1} = 1$

Second Step: Prove $P(n) \rightarrow P(n+1)$

- $P(n+1) = 1 + r + r^2 + \ldots + r^n + r^{n+1} = P(n) + r^{n+1}$
- $P(n+1) = \frac{r^{n+1}-1}{r-1} + r^{n+1} = \frac{(r^{n+1}-1)+(r^{n+1}(r-1))}{r-1}$
- $P(n+1) = \frac{r^{n+1}-1+r^{n+2}-r^{n+1}}{r-1} = \frac{r^{n+2}-1+r^{n+1}-r^{n+1}}{r-1}$
- $P(n+1) = \frac{r^{n+2}-1}{r-1} \quad \square$

Review: Proof Template for Induction

Proof by induction on n
Proof hypothesis: $P(n) = \ldots$ for all $n \in \mathbb{N}.n \geq 0$

First we prove $P(0)$.
$\ldots$ *(calculate that P(0) is True)*
$\ldots$

Second we prove that $\forall n \geq 0, P(n) \rightarrow P(n+1)$
$\ldots$ *(calculate P(n+1) using P(n))*
$\ldots$

This completes the proof that $P(n)$ for all $n \in \mathbb{N}$

Example 2: The Bill Square Induction Proof

Note: Better do this on the blackboard

- Situation: 2^n square park with a statue in the middle
- Park must be formed by L-shaped tiles. Prove that the park is possible for any *n*.
- Proof Try One: n=0, park has 1 tile. Ok. n = n, I have 4 parks with $2^{n/2}$ with the statue in the middle... what do I do? I am stuck.
- OK, let's prove something STRONGER! Let's prove that we can put the statue ANYWHERE.
- Proof Try Two: n=0, park has 1 tile. Same thing. n = n, I have 4 n-1 parks that I can put the statue anywhere. I choose one location arbitrarily for the statue, and the other three statues I put in the center of the park, and replace with an L-shaped tile. Success!

Lessons from the Bill Square Proof

- This proof gives me a recursive procedure to find the locations of all tiles. (A program!)
- It is interesting that we need a STRONGER hypothesis to make the proof EASIER.

A bogus induction proofs

Understanding proofs includes the ability to find mistakes in proofs. Let's see an example.

Proof: All horses are of the same color

Proof (By induction on n)

Proof: All horses are of the same color

Proof (By induction on n)

- $P(n) ::=$ for any set with exactly n horses, all horses have the same color.

Proof: All horses are of the same color

Proof (By induction on n)

- $P(n) ::=$ for any set with exactly n horses, all horses have the same color.
- **Base Case:** $(n = 1)$. Any set with one horse has one color.

Proof: All horses are of the same color

Proof (By induction on n)

- $P(n) ::=$ for any set with exactly n horses, all horses have the same color.
- **Base Case:** $(n = 1)$. Any set with one horse has one color.
- **Inductive case:** Assume any set with n horses, all have the same color.

Proof: All horses are of the same color

Proof (By induction on n)

- $P(n) ::=$ for any set with exactly n horses, all horses have the same color.
- **Base Case:** $(n = 1)$. Any set with one horse has one color.
- **Inductive case:** Assume any set with n horses, all have the same color.
- **From P(n), try to prove P(n+1):**
  - Consider the set of n+1 horses: $H = h_1, h_2, \ldots, h_n, h_{n+1}$

Proof: All horses are of the same color

Proof (By induction on n)

- $P(n) ::=$ for any set with exactly n horses, all horses have the same color.
- **Base Case:** $(n = 1)$. Any set with one horse has one color.
- **Inductive case:** Assume any set with n horses, all have the same color.
- **From P(n), try to prove P(n+1):**
  - Consider the set of n+1 horses: $H = h_1, h_2, \ldots, h_n, h_{n+1}$
  - subset A: $h_1, h_2, \ldots, h_n$ all have the same color (because we assume $P(n)$)

Proof: All horses are of the same color

Proof (By induction on n)

- $P(n) ::=$ for any set with exactly n horses, all horses have the same color.
- **Base Case:** $(n = 1)$. Any set with one horse has one color.
- **Inductive case:** Assume any set with n horses, all have the same color.
- **From P(n), try to prove P(n+1):**
  - Consider the set of n+1 horses: $H = h_1, h_2, \ldots, h_n, h_{n+1}$
  - subset A: $h_1, h_2, \ldots, h_n$ all have the same color (because we assume $P(n)$)
  - subset B: $h_2, \ldots, h_n, h_{n+1}$ also all have the same color! (also because of $P(n)$)

Proof: All horses are of the same color

Proof (By induction on n)

- $P(n) ::=$ for any set with exactly n horses, all horses have the same color.
- **Base Case:** $(n = 1)$. Any set with one horse has one color.
- **Inductive case:** Assume any set with n horses, all have the same color.
- **From P(n), try to prove P(n+1):**
  - Consider the set of n+1 horses: $H = h_1, h_2, \dots, h_n, h_{n+1}$
  - subset A: $h_1, h_2, \dots, h_n$ all have the same color (because we assume $P(n)$)
  - subset B: $h_2, \dots, h_n, h_{n+1}$ also all have the same color! (also because of $P(n)$)
  - Therefore, all horses in H have the same color!

Proof: All horses are of the same color

Proof (By induction on n)

- $P(n) ::=$ for any set with exactly n horses, all horses have the same color.
- **Base Case:** $(n = 1)$. Any set with one horse has one color.
- **Inductive case:** Assume any set with n horses, all have the same color.
- **From P(n), try to prove P(n+1):**
  - Consider the set of n+1 horses: $H = h_1, h_2, \ldots, h_n, h_{n+1}$
  - subset A: $h_1, h_2, \ldots, h_n$ all have the same color (because we assume $P(n)$)
  - subset B: $h_2, \ldots, h_n, h_{n+1}$ also all have the same color! (also because of $P(n)$)
  - Therefore, all horses in H have the same color!
- Proof complete???? What is wrong?

What is wrong?

The proof that $P(n) \rightarrow P(n+1)$ is wrong.

What is wrong?

The proof that $P(n) \to P(n+1)$ is wrong.

- The proof has to be valid for all $n \geq 1$
- If $n = 1$

What is wrong?

The proof that $P(n) \to P(n+1)$ is wrong.

- The proof has to be valid for all $n \geq 1$
- If $n = 1$
- Set $H = h_1, h_2$

What is wrong?

The proof that $P(n) \to P(n+1)$ is wrong.

- The proof has to be valid for all $n \geq 1$
- If $n = 1$
- Set $H = h_1, h_2$
- subset $A = h_1$, subset $B = h_2$, and $A \cap B = \varnothing$

What is wrong?

The proof that $P(n) \rightarrow P(n+1)$ is wrong.

- The proof has to be valid for all $n \geq 1$
- If $n = 1$
- Set $H = h_1, h_2$
- subset $A = h_1$, subset $B = h_2$, and $A \cap B = \varnothing$

- Note that $n = 1$ the only problem with the proof!

Strong Induction

- In regular induction, you assume P(n) to show P(n+1)
- In strong induction, you assume P(0), P(1), P(2) . . . P(n), and use all of them to show P(n+1)

Strong Induction Example: Stacking Game

- Begin with a stack of 10 blocks
- Divide it in two (a,b): for example, 2 and 8 blocks.
- You get $a \times b$ points: 10 points
- Repeat with the new stacks until all stacks have 1 block.

What is the best strategy?

- Simple strategy: 1+9, 1+8, 1+7, 1+6...
- CS strategy: 5+5, 2+3 and 2+3, ...

Strong Induction Example: Stacking Game

- Begin with a stack of 10 blocks
- Divide it in two (a,b): for example, 2 and 8 blocks.
- You get $a \times b$ points: 10 points
- Repeat with the new stacks until all stacks have 1 block.

What is the best strategy?

- Simple strategy: 1+9, 1+8, 1+7, 1+6... 45 points!
- CS strategy: 5+5, 2+3 and 2+3, ...

Strong Induction Example: Stacking Game

- Begin with a stack of 10 blocks
- Divide it in two (a,b): for example, 2 and 8 blocks.
- You get $a \times b$ points: 10 points
- Repeat with the new stacks until all stacks have 1 block.

What is the best strategy?

- Simple strategy: 1+9, 1+8, 1+7, 1+6... 45 points!
- CS strategy: 5+5, 2+3 and 2+3, ... 45 points!

Proof: All strategies have the same score (Part I)

Let us prove by strong inductions that all strategies for the stack game with “n” blocks have the score:

$$C(n) = \frac{n(n-1)}{2}$$

Base Cases: 0, 1

- When the stack has 0 blocks, I have no moves, so 0 points.
- When the stack has 1 block, I have no moves, so 0 points.

$$C(0) = \frac{0(0-1)}{2}, C(1) = \frac{1(1-1)}{2} = 0$$

Proof: All strategies have the same score (Part II)

Inductive Case $C(n+1)$
By strong induction, we assume that $C(0) \dots C(n)$ are true.

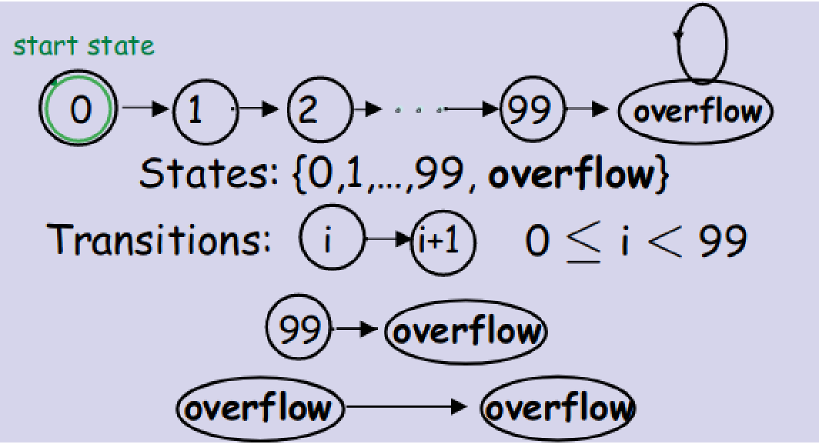
- I can split a $n+1$ stack into: k and $n+1-k$ ($k \geq 1$)
- The score is:
  $C(n+1) = k \times (n+1-k) + C(k) + C(n+1-k)$
- Using the inductive assumption: $C(m) = \frac{m(m-1)}{2}$:
- $C(n+1) = \frac{2k(n+1-k)}{2} + \frac{k(k-1)}{2} + \frac{(n+1-k)(n-k)}{2}$
- ... You continue from here ;-)

State Machines

Definition

- Model step-by-step processes
- Computations, Algorithms, Logic Circuits

Simple Example



Example: Linear Robot 1.0

Imagine a robot that moves back and forth in a straight line. The robot has two speeds:

- Forward, where it moves exactly five squares foward.
- Back, where it moves exactly three squares back.

Starting from position **0**, is it possible for the robot to reach position 4?

Example: Linear Robot 1.1

Imagine a robot that moves back and forth in a straight line. The robot has two speeds:

- Forward, where it moves exactly **nine squares foward.**
- Back, where it moves exactly three squares back.

Starting from position **0**, is it possible for the robot to reach position 4?

Why is it impossible for robot 1.1 reach square 4?

Preserved Invariant States

Preserved Invariants are variables in a state machine that are not modified by the actions of the computation steps.

Example: The position of robot 1.1 is always $n + 3k$ (n is the initial state, $k \in \mathbb{Z}$

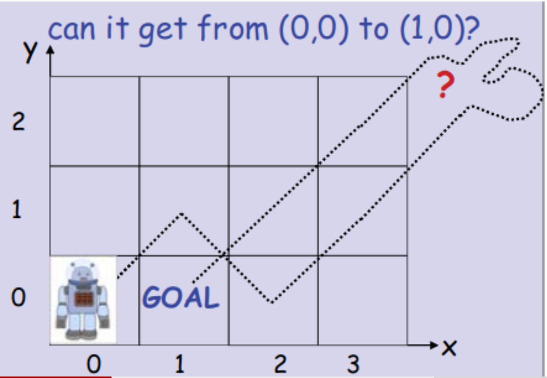
Preserved Invariants can be used to perform induction on state machines:

- Prove that the preserved invariant, $P(s)$, holds for initial state s_0
- Prove that all transitions $P(s)$ to $P(s')$ do not change the invariant.
- Conclude that $P(s)$ holds for the entire computation.

Example 2: Diagonal Robot

Let's use invariants to prove or disprove the following:

Given a robot in $\mathbb{Z}^2$, that moves on the diagonals: (+1, +1), (-1,-1), (+1,-1), (-1,+1). Is it possible for the robot to reach position (1,0) from the initial position (0,0)?



Example 2: Diagonal Robot

We can notice that one preserved invariant of the robot is that the sum of its coordinates is always even (or always odd):

- P(0,0) is true (0+0 is even).
- The steps of the robot are:
  - +1+1 = +2
  - -1-1 = -2
  - +1-1 = 0
  - -1+1 = 0

From the steps/transitions. we see that if the sum of (x,y) is an oven number, any of the sucessor states will keep the same preserved invariant.

Example 3: Fast Exponentiation

- Please watch lecture video 1.9.1
- To prove that an algorithm is correct, we need to prove two thngs:
  - Prove that if the machine stops, the program is always correct (Correct output is a preserved invariant).
  - Prove that the program halts at some point. (follow an integer variable, and make sure that it decreases at every step)

Extra Topics

- Recusive Data Type and Structural Induction (1.10)
- Infinite Sets (1.11)