Review II(Slides 72 - 116)

Logics & Induction

Start from the very begining...

HamHam

University of Michigan-Shanghai Jiao Tong University Joint Institute

May 29, 2023

VE203 - Discrete Mathmatics

Summary

- Sets
  - * Definitions & Operations
  - Application: Graph
- Logics
  - Logical Variable & Operations
  - Concepts: Vacuous truth, Tautology, Predicate, Quantifier...
  - * Predicate Logic (First-Order Logic): Truth Tree
  - * Propositional Logic: Natural Deduction Tree
- Induction
  - * Weak/Strong(Complete) Induction (Type I/II Induction)
  - Structual Induction
  - Correctness of Sorting Algorithms
  - Concepts: WOP, Monoid

Updates

1. Textbook: Galliar
2. Some Typo Fixed
3. Format of Truth-Tree Proof

Natural Deduction Tree

General Rule

$$\frac{\text{given that} \ldots}{\text{we conclude that} \ldots}\text{What was done}$$

Strategy

1. Introduction: We want more!
2. Elimination: We want less!

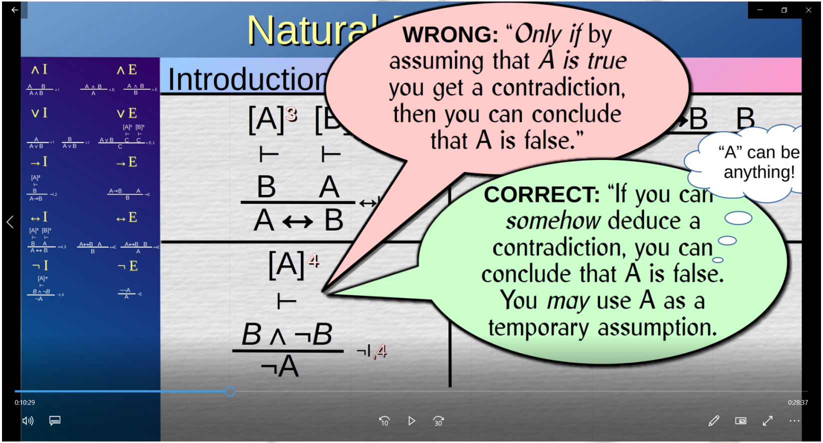
Reference:

https://www.youtube.com/watch?v=v-1ikd_89Mg

Natural Deduction Tree

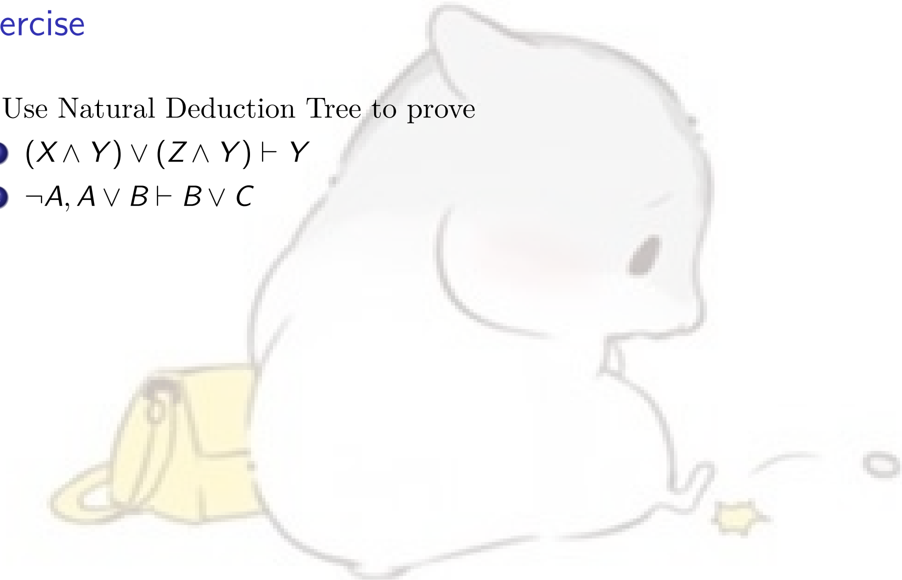
	Introduction	Elimination
$\wedge$		
$\vee$		
$\rightarrow$		
$\neg$		

Introduction on False / Negation



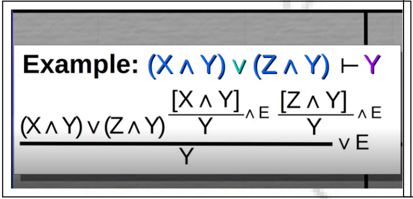
Exercise

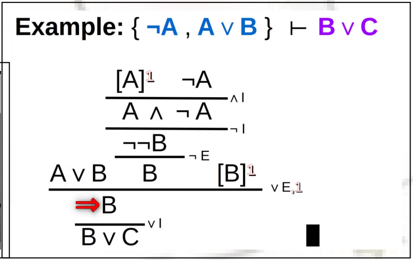
1. Use Natural Deduction Tree to prove
   1. $(X \wedge Y) \vee (Z \wedge Y) \vdash Y$
   2. $\neg A, A \vee B \vdash B \vee C$



Exercise

1. Use Natural Deduction Tree to prove
   1. $(X \wedge Y) \vee (Z \wedge Y) \vdash Y$
   2. $\neg A, A \vee B \vdash B \vee C$





Mathematical Induction

Weak Induction (Type I Induction)

Supppse over $\mathbb{N}$ we have

1. $P(k)$
2. $\forall n \geq k, \quad P(n) \to P(n+1)$

Then $\forall n \geq k, \quad P(n)$. Usually we have $k = 0$ as base case.

Strong(Complete) Induction (Type II Induction)

1. $P(0)$
2. $\forall n \geq 1, \quad [P(0) \wedge P(1) \wedge \cdots \wedge P(n-1) \to P(n)]$

Then $\forall n, \quad P(n)$.

Mathematical Induction

- **Weak induction** is like a line of dominoes where each domino only depends on the one immediately before it. The proof establishes that if the first domino falls, it will topple the next one, which, in turn, knocks down the subsequent one. By this chain reaction, the entire row of dominoes falls.

- **Strong induction** is like a line of dominoes where each domino has the ability to knock down all the dominoes that come after it. By assuming that when any domino falls, it has a cascading effect on all the dominoes ahead, the proof establishes that if the first domino falls, it will eventually cause the entire row of dominoes to collapse.

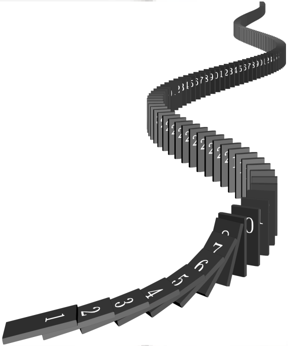


Figure: Falling dominoes

Induction "Paradox" (Slides 93)

For any $n \in \mathbb{N}\backslash\{0\}$, there exist integers a_n and b_n s.t.

$$\left(\frac{1+\sqrt{5}}{2}\right)^n = \frac{a_n + b_n\sqrt{5}}{2}$$

1. Base case: $n = 1$, clear by taking $a_1 = b_1 = 1$
2. Inductive case: assume the IH for $n \geq 1$,

$$\begin{aligned}\left(\frac{1+\sqrt{5}}{2}\right)^{n+1} &= \frac{a_n + b_n\sqrt{5}}{2} \cdot \frac{1+\sqrt{5}}{2} \\ &= \frac{(a_n+5b_n)/2 + ((a_n+b_n)/2)\sqrt{5}}{2}\end{aligned}$$

Induction "Example"¿

Pitfalls in Induction

While mathematical induction is an extremely powerful technique, it must be executed most carefully. In proceeding through an induction proof it can happen quite easily that implicit assumptions are made that are not justified, thereby invalidating the result.

1.4.4. Example. Let us use mathematical induction to argue that every set of $n \geq 2$ lines in the plane, no two of which are parallel, meet in a common point.

The statement is true $n = 2$, since two lines are not parallel if and only if they meet at some point. Since these are the only lines under considerations, this is the common meeting point of the lines.

We next assume that the statement is true for n lines, i.e., any n non-parallel lines meet in a common point. Let us now consider $n + 1$ lines, which we number 1 through $n + 1$. Take the set of lines 1 through n; by the induction hypothesis, they meet in a common point. The same is true of the lines $2, \ldots, n + 1$. We will now show that these points must be identical.

Induction "Example"¿

Pitfalls in Induction

Assume that the points are distinct. Then all lines $2, \ldots, n$ must be the same line, because any two points determine a line completely. Since we can choose our original lines in such a way that we consider distinct lines, we arrive at a contradiction. Therefore, the points must be identical, so all $n+1$ lines meet in a common point. This completes the induction proof.

Where is the mistake in the above "proof" of our (obviously false) supposition?

Exercise

Further Examples of Induction

6. Let M be a set with cardinality card $M = n$, $n \in \mathbb{N}$. Then card $\mathcal{P}(M) = 2^n$.
7. Shootout at the O.K. Corral: an odd number of lawless individuals, standing at mutually distinct distances to each other, fire pistols at each other in exactly the same instant. Every person fires at their nearest neighbor, hitting and killing this person. Then there is at least one survivor.

Solution

The Shootout

Let us discuss the last example in more detail: Let $P(n)$ be the statement that there is at least one survivor whenever $2n+1$ people, fire pistols at each other in the same instant. Each person fires at his nearest neighbor, and all people stand at mutually distinct distances to each other.

For $n=1$, there are three people, A, B and C. Suppose that the distance between A and B is the smallest distance of any two of them. Then A fires at B and vice-versa. C fires at either A or B and will not be fired at, so C survives.

Suppose $P(n)$. Let $2n+3$ people participate in the shootout. Suppose that A and B are the closest pair of people. The A and B fire at each other.

- If at least one other person fires at A or B, then there remain $2n$ shots fired among the remaining $2n+1$ people, so there is at least one survivor.
- If no-one else fires at A or B, then there are $2n+1$ shots fired among the $2n+1$ people and by $P(n)$, there is at least one survivor.

Exercise (DIY)

2. Let a_n be the following expression with n nested radicals:

$$a_n = \sqrt{2 + \sqrt{2 + \cdots + \sqrt{2 + \sqrt{2}}}}$$

Please find the explict formula for a_n.

Solution

Note that a_n can be defined recursively like this: $a_1 = \sqrt{2}$, and $a_{n+1} = \sqrt{a_n + 2}$ for $n \geq 1$. We proceed by induction.

Hypothesis: $a_n = 2\cos\frac{\pi}{2^{n+1}}$

Base case: $a_1 = \sqrt{2}$, and $2 \cdot \cos\frac{\pi}{4} = 2 \cdot \frac{1}{\sqrt{2}} = \sqrt{2}$.

Inductive case: assuming the **IH** is true for n, then

$$
\begin{aligned}
a_{n+1} = \sqrt{2 + a_n} &= \sqrt{2 + 2\cos\frac{\pi}{2^{n+1}}} \\
&= \sqrt{2 + 2\cos\frac{2\pi}{2^{n+2}}} \\
&= \sqrt{2 + 2(2\cos^2\frac{\pi}{2^{n+2}} - 1)} \\
&= \sqrt{4\cos^2\frac{\pi}{2^{n+2}}} = 2\cos\frac{\pi}{2^{n+2}}
\end{aligned}
$$

By induction, we conclude that $a_n = 2\cos\frac{\pi}{2^{n+1}}$.

Sorting Algorithms

We have lots lots of sorting algorithms!

- Demostration: `https://www.sortvisualizer.com/quicksort/`
- Classification
- Costs
  - Time complexity (Worst/Averge/Best)
  - Space complexity
  - Stability

Question

How to choose a sorting algorithms that most satisfy your need?

Quick Sort with Second key word

```
void QuickSort(vector<int>& v, vector<int>& o,
               int left, int right){
    if (left>=right)  return;
    int key = (left+right)/2;// you can also choose right/left
    while (left<right){
        while (left<right && (v[right] > v[key] ||
          v[right]==v[key] && o[right]>o[key]))  right--;
        while (left<right && (v[left] <= v[key] ||
          v[left]==v[key] && o[left]<o[key]))   left++;
        if (left < right) swap(v[left], v[right]);
    }
    swap(v[left], v[key]);   //left== right
    int meet= key;      //divide into two parts
    QuickSort(v, o, left, meet-1);
    QuickSort(v, o, meet+1, right);
}
```

Modified Bubble Sort

```java
public static void bubble_sort(int[] intArr) {
    int max = intArr.length - 1;
    int secondCount = max;  //record where the exchange happen last time
    for (int i = 0; i < max; i++) {
        System.out.println( (i + 1) + "times" );
        boolean flag = true;int lastChangeIndex = 0;
        for (int j = 0; j < secondCount; j++) {
            if (intArr[j] > intArr[j + 1]) {
                swap(Arr[j],Arr[j+1]);
                flag = false;
                lastChangeIndex = j;
            }
            System.out.println("Compare:"+(j+1)+", Result:"+Arrays.toString
        }
        if (flag) break; //already well ordering
        secondCount = lastChangeIndex; //update
    }
}
```

Exercise

3. Try to implement three-way merge sort in C++. If you know Master Theorem, try to calculate the time complexity and compare with the original merge sort.

4. Exercise regarding sorting algorithms:
   - https://leetcode.cn/problems/insertion-sort-list/
   - https://leetcode.cn/problems/kth-largest-element-in-an-array/

Reference

- Lecture Slides from Ve203 FA 2020 by Horst Hohberger.
- Examples from Vv286 Lecture Slides.
- Exercises from 2021-Fall-Ve203 TA Zhao Jiayuan
- Stable Quick Sort, `https://blog.csdn.net/liuchenjane/article/details/72902325`
- Modified Bubble Sort, `https://blog.csdn.net/weixin_43168559/article/details/88873585`
- Picture "Falling Dominoes", `https://crystalclearmaths.com/videos-learning-resources/iterations-repeating-patterns/mathematical-induction/`