Mathematical Induction II

Discrete Mathematics

Courtesy of Andrei Bulatov

Principle of Mathematical Induction

- *Proof by mathematical induction*:

  To prove that a statement that assert that some property $P(n)$ is true for all positive integers n, we complete two steps

  Basis step: *We verify that $P(1)$ is true.*

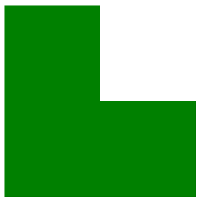
  Inductive step: *We show that the conditional statement*

  $$P(k) \to P(k+1) \text{ is true for all positive integers } k$$

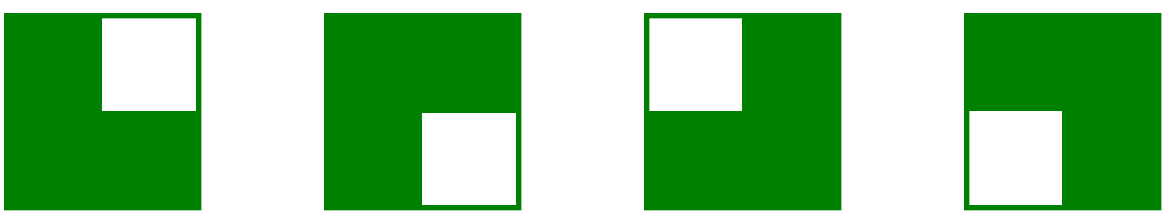
- *To prove the conditional statement, we assume that $P(k)$ is true (it is called inductive hypothesis) and show that under this assumption $P(k+1)$ is also true*

Example (Triomino tiling)

- *Let n be a positive integer. Show that every $2^n \times 2^n$ checkerboard with one square removed can be tiled using triominos*

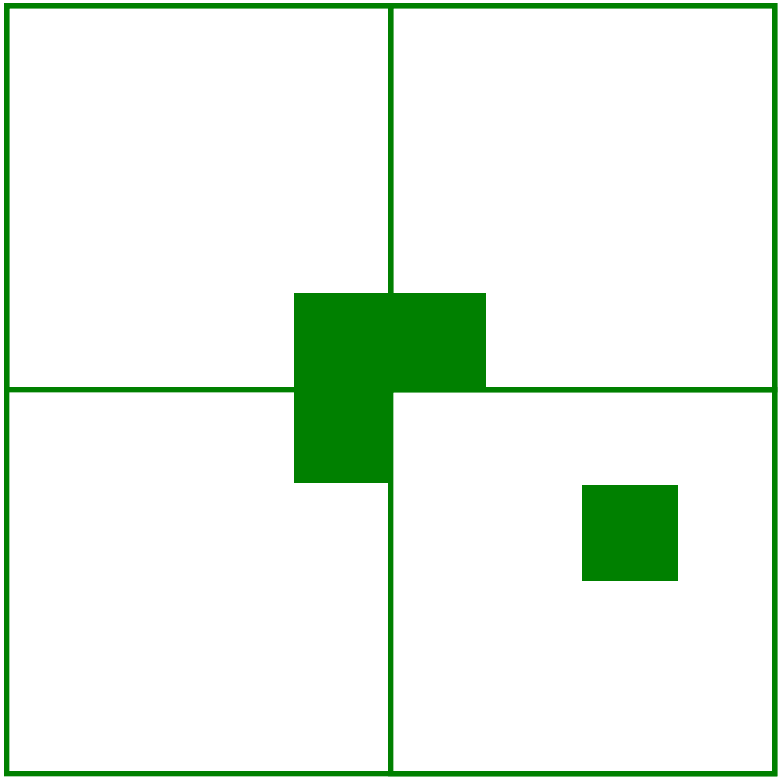


- *$P(n)$ denotes the statement above*
- *Basis step: $P(1)$ is true, as 2×2 checkerboards with one square removed have one of the following shapes*



Triomino (cntd)

- *Inductive step: Suppose that P(k) is true that is every* $2^k \times 2^k$ *checkerboard with one square removed can be tiled with triominos.*
  We have to prove P(k + 1), that is, every $2^{k+1} \times 2^{k+1}$ *checkerboard without one square can be tiled.*



Split the big checkerboard into 4 half-size checkerboards

WLOG assume the bottom right is missing a square. Put one triomino as shown.

We have 4 $2^k \times 2^k$ *checkerboards, each without one square. By the induction hypothesis, they can be tiled.*

Match game

- *The game starts with two piles of matches*

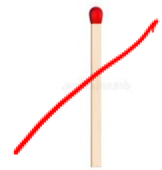

- *The two players take turns removing any number of matches from a pile of their choice*
- *The player who takes the last match wins*
- *Theorem:*

  if the two piles are the same size initially, then the second player always has a winning strategy

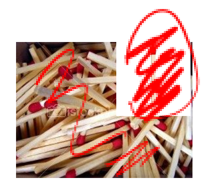
Match game


- *Let P(n) denote "player 2 wins if both piles have size n"*

- *Base case: P(1)*
  Player two wins by default




- *Inductive step: Assume P(k)*
  Consider a game with piles of size k+1 initially.




  Suppose player 1 removes 1 match from the left pile

  Then player 2 removes 1 from the right to give two piles of size k

What if player 1 removes more than 1 match?

Principle of Strong Induction

- *In the last proof mathematical induction is not enough. We need to know that the game is winnable for piles of any (smaller) size*
- *In these instances we can use the principle of strong induction.*

- *Proof by strong induction: To prove that $P(n)$ is true for all positive integers n, we complete two steps:*
- *Basis step: Verify that $P(1)$ is true.*
- *Inductive step: Show that the statement*

$$\underline{[\, P(1) \wedge P(2) \wedge \ldots \wedge P(k)} \,] \rightarrow P(k + 1)$$

is true for all positive integers k.

Match game (redux)

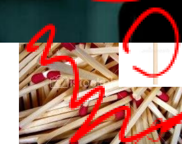
- *Let P(n) denote "both piles have size n"*

- *Base case: P(1)*

  Player two wins by default

- *Inductive step: Assume P(i) for all i up to k*

  Consider a game with piles of size k+1 initially.



  Suppose player 1 removes x matches from the left

  Then player 2 removes x from the right to give two piles of size k+1-x

  Since x > 0, k+1-x is at most k, so by the inductive hypothesis player 2 has a winning strategy

Why Induction Works? Well Ordering

- *The positive integers are well-ordered:*

  Every non-empty subset of $\mathbb{N}$ contains a least element.

- *Are the integers well ordered? Real numbers?*
- *Theorem: well ordering implies the principle of induction*

  Suppose that mathematical induction is not valid.

  Then there is a predicate $P(n)$ such that $P(1)$ is true,

  $\forall k\, (P(k) \to P(k+1))$ is true, but there is n such that $P(n)$ is false

  Let $T \subseteq \mathbb{N}$ be the set of all n such that $P(n)$ is false.

  By the principle of well-ordering T contains a least element k

  As $P(1)$ is true, $k \neq 1$.

  Since k is the least element of T, $P(k - 1)$ is true

  But this implies $P(k)$ is true, hence a contradiction!

Applications of induction: Recursively Defined Functions

- *Induction can also be used to define things.*
- *In this case it's called a recursive definition*

- *To define a function* $f: \mathbb{N} \to \mathbb{R}$ *recursively we complete two steps:*

  Basis step: define $f(1)$

  Inductive step: For all k *define* $f(k + 1)$ *as a function of* $f(k)$,

  or, more generally, as a function of $f(1), f(2), \dots, f(k)$.

- *Give a recursive definition of* $f(n) = 2^n$

  Basis step: $f(0) = 1$

  Inductive step: $f(k + 1) = 2 \cdot f(k)$.

Factorial

- *Another useful recursively defined function is factorial*
- $f(n) = n!$

  Basis step: $0! = 1$

  Inductive step: $(k + 1)! = k! \cdot (k + 1)$

n	$n!$
0	1
1	1
2	2
3	6
4	24

n	$n!$
5	120
6	720
7	5040
8	40320
9	362880

Fibonacci Numbers

- *Usually, Fibonacci numbers are thought of as a sequence of natural numbers, but as we know such a sequence can also be viewed as a function from* $\mathbb{N}$.
- *F(n)*
- *Basis step: F(1) = F(2) = 1*
- *Inductive step: F(k + 1) = F(k) + F(k – 1)*

n	1	2	3	4	5	6	7	8	9	10	11	12	13
F(n)	1	1	2	3	5	8	13	21	34	55	89	144	233

Binet's formula

$$F(n) = \frac{\varphi^n - (1-\varphi)^n}{\sqrt{5}}$$

where φ *is the golden ratio*

$$\varphi = \frac{1+\sqrt{5}}{2} \approx 1.618033988749$$

Applications of induction: Recursively Defined Sets and Structures

- *Induction can also be used to define structures*

- *We need to complete the same two steps:*

  Basis step: Define the simplest structure possible

  Inductive step: A rule, how to build a bigger structure from smaller ones.

Well Formed Propositional Statements

- *What is a well formed statement?*

  $(p \to q) \land \neg r$ *is well formed*

  $(p \to q) \neg \land r$ *is not*

- *Recursive definition of well formed formulas*

  Basis step: A primitive proposition is a well formed formula

  Inductive step: If Φ *and* Ψ *are well formed statements, then*

  $\neg\Phi$, $(\Phi \land \Psi)$, $(\Phi \lor \Psi)$, $(\Phi \to \Psi)$, $(\Phi \leftrightarrow \Psi)$, $(\Phi \oplus \Psi)$

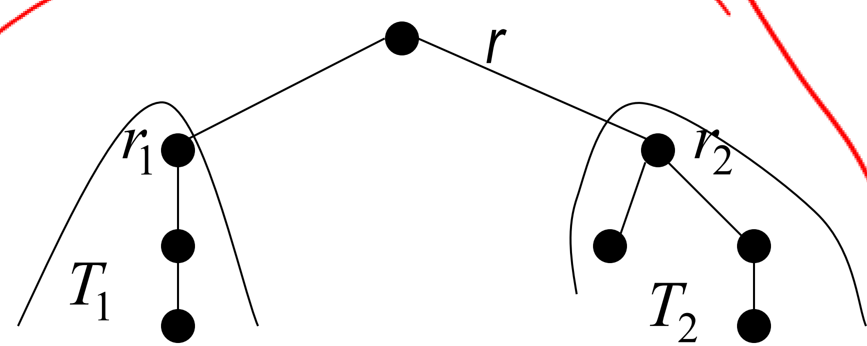
  are well formed statements

- *Computer programs use recursive definitions like these to define most data structures*

Rooted Trees

- *A binary tree is a graph formed by the following recursive definition*

- *Basis case: A single vertex or nothing is a binary tree* ● r
- *Inductive step: Suppose that* T_1, T_2 *are disjoint binary trees with roots* $r_1, r_2,$ *respectively. Then the graph formed by starting with a root* r*, and adding an edge from* r *to each of the vertices* $r_1, r_2,$ *is also a binary tree.*



Structural Induction

- *To prove properties or design algorithms working with recursively defined structures we need structural induction*
- *To prove a statement using structural induction we complete two steps*

  Basis step: Prove that the property is true for the simplest structure

  Inductive step: Assuming that the property is true for all simpler structures, prove it for a more complex structure

Practice

Exercises from the Book:

7th edition: 8, 12 (page 342), 1b, 3a, 7b, 26 (page 357 – 358)

8th edition: 8, 12 (page 363), 1b, 3a, 7b, 28 (page 378 – 379)