Introduction to Invariance Principle

If there is repetition, look for what does not change –Arthur Engel.

Freeman Cheng, Idris Tarwala

Erindale Secondary School

Overview

Definition

An invariant is something that doesn't change.

Sometimes we are presented with a problem where some task is repeatedly performed, such as tiling a chessboard with 2x1 tiles. We ask ourselves, in such processes, what doesn't change? Some questions to consider asking yourself:

1. Can a given end state be reached?
2. What are the possible end states?
3. Is there a convergence to an end state?
4. Or is there a period?

(Four questions adapted from PSS by Arthur Engel).

Coloring Arguments

The first three examples can be solved using coloring arguments.

Tiling a Chessboard w/ Two Missing Pieces

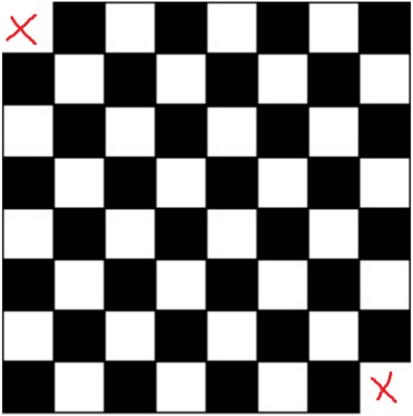
Example 1

Can we completely tile a 8x8 chessboard with two opposite corner pieces removed, with 2x1 tiles? (No overlapping, of course).

At first, we're not sure how to approach this problem, so the best strategy at the moment is to try tiling ourselves.

Example 1

Can we completely tile a 8x8 chessboard with two opposite corner pieces removed, with 2x1 tiles? (No overlapping, of course).



Solution

We list some observations. First, color the chessboard in alternating pattern like in previous slide.

1. Every 2x1 tile covers one black square and one white square.
2. In the modified board there are 32 black squares, and 30 white squares.
3. Observation 1 implies that in any configuration, the number of black squares covered must equal the number of white squares covered. Thus, by observation 2, not all tiles can be covered without overlap.

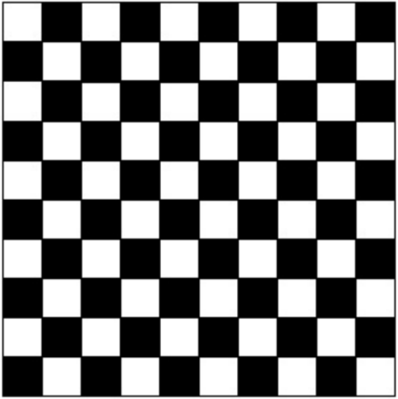
Tiling with T-tetrominos

Example 2

Prove that a 10x10 board cannot be covered by T-shaped tetrominos? (No overlap, of course).

Solution

Color the board like shown below.



Solution

Some observations.

1. The T-tetromino can either cover an area with 3 white, 1 black (type 1) or 1 white, 3 black (type 2).
2. If the board can be completely tiled, it will take $\frac{100}{4} = 25$ tiles.
3. Suppose that we can tile the board completely, then there must be an equal number of type 1s and type 2s since the number of white squares is equal to the number of black squares (they need to balance each other).
4. This is impossible, since 25 is odd.

Colored Points the Plane

Example 3

a) Each point on the plane is colored either red or blue. Prove that there exist two points of the same color of distance 1.
b) Each point on the plane is colored either red, blue, or green. Prove that there exist two points of the same color of distance 1.

Solution

a) Construct an equilateral triangle, then proof by contradiction.
b) Solution orally given. For people reviewing, it is left to the reader as exercise.

Parity (Evens and Odds)

The next three example deal with invariance in parity.

Numbers on a Circle

Example 4

A circle is divided into 6 sectors. The numbers $1, 0, 1, 0, 0,$ and 0 are written in that order. You can increase two adjacent numbers by 1. Is it possible to make all numbers equal?

Again, it is important to try some examples.

Solution

Again, we make some observations.

1. Adding 1 to two adjacent sectors does not affect the difference between the two sectors.
2. Label the values in each sector a, b, c, d, e, f. Any move does not affect the value $a - b + c - d + e - f = 2$.
3. For all values in each sector to equal, $a = b = c = d = e = f \implies a - b + c - d + e - f = 0$, which by observation 2 is impossible.

Numbers on a Blackboard

Example 5

Let n be an odd positive integer. The first $2n$ integers are written on the blackboard. Erase any two integers on the blackboard, and replace them with their difference. Eventually, there will be one number left. Prove that this number is odd.

Solution

Some observations.

1. The initial sum

$$1 + 2 + ... + 2n = \frac{(2n+1)(2n)}{2} = (2n+1)(n)$$

which has to be odd.

2. Let numbers x and y be erased where $x \geq y$. Then, the number $x - y$ is added. The net change is

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

which is even.

3. Every move, the net change is even, so after every move the sum remains odd.

A Codeforces Problem

Example 6

The jury picked an integer x not less than 0 and not greater than $2^{14}-1$. You have to guess this integer. To do so, you may ask no more than 2 queries. Each query should consist of 100 integer numbers $a_1, a_2, ..., a_{100}$ (each integer should be not less than 0 and not greater than $2^{14}-1$). In response to your query, the jury will pick one integer i ($1 \le i \le 100$) and tell you the value of a_i XOR x (the bitwise XOR of a_i and x). There is an additional constraint on the queries: all 200 integers you use in the queries should be distinct. How should we submit the queries?

Solution

Consider 14-digit binary string where leading digits can be 0. We make a construction. Query 1: construct binary numbers with first 7 digits as 1, the last 7 randomized (we are guaranteed over 100 numbers since $2^7 > 100$. Similarly, for query 2, make the last 7 digits 0 and first 7 randomized. Thus, we can uniquely determine our number.

Monovariants

Definition

A monovariant is a property that only changes in one direction (increases or decreases). For example, the total sum of some sequence is strictly decreasing.

The next problem deals with monovariants.

Enemies at Erindale

Example 7

In Erindale, each student has at most three enemies. Prove that the students can be split into two groups, such that each student has at most one enemy in his/her group.

Solution

Consider the algorithm: first split randomly. Then, if a student has two or more enemies in his/her group, move the student to the other group. To prove that this algorithm terminates, consider the sum of all enemy relationships. After each iteration of the algorithm, the sum decreases by at least 1. (Easy to verify this, left to the reader). Since the total sum of enemy relationships in the two groups is a non-negative integer value, algorithm must terminate ie. there will be no student with more than 1 enemy.

References

A significant chunk of problems were Howard Halim's handout on invariance principle (howardhalim.com). Also thanks to Andrew Tang for showing me the CodeForces problem.