CSC236 Lecture 01: Theory of Computation

Hisbaan Noorani

September 10, 2021

Contents

1 Why reason about computing

- You're not just hackers anymore

  Sometimes you need to analyze code before it runs. Sometimes it should never be run!

- Can you test everything?

  Infinitely many inputs: integers, strings, lists.

- Careful, you might get to like it. . . (?!*)

2 How to reason about computing

- It's messy. . .

  interesting problems fight back.

You need to draft, re-draft, and re-re-draft.

You need to follow blind alleys until you find a solution.

You can also find a solution that isn't wrong, but could be better.

- It's art...

  Strive for correctness, clarity, surprise, humor, pathos, and others.

3 How to do well in this course

- read the syllabus as a two-way promise
- question, answer, record, synthesize

  try annotating blank slides.
- collaborate with respect

  You need computer science friends who are respectful and constructively critical.

4 Assume that you already know

- Chapter 0 material from *Introduction to Theory of Computation.*
- CSC110/111 material, especially proofs and big-$\mathcal{O}$.

5 By December you'll know

- understand and use several flavours of induction.

  some of these flavours will taste new
- Formal languages, regular languages,regular expressions

  Sets of strings
- complexity and correctness of programs — both recursive and iterative

6 domino fates foretold

$$[P(0) \wedge (\forall n \in \mathbb{N}, P(n) \implies P(n+1))] \implies \forall n \in \mathbb{N}, P(n)$$

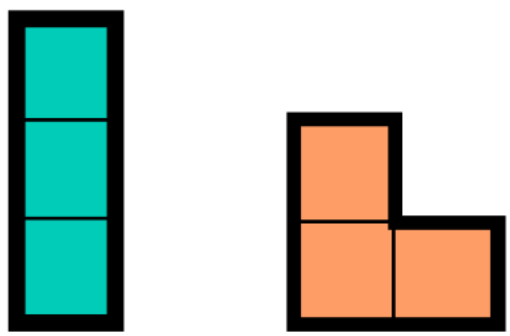
If the initial case works, and each case that works implies its successor works, then all cases work

7 Simple induction outline

- inductive step: introduce n and inductive hypothesis $H(n)$
  - derive conclusion $C(n)$: show that $C(n)$ follows from $H(n)$, indicating **where** you use $H(n)$ and why that is valid.
- Verify base case(s): verify that the claim is true for any cases not covered in the inductie step
- In simple induction $C(n)$ is just $H(n+1)$

8 Trominoes

See `https://en.wikipedia.org/wiki/Tromino`



Can an $n \times n$ square grid, with one subsquare removed, be tiled (covered withotu overlapping) by "chair" trominoes?

- 1×1: Yes.
- 2×2: Yes.
- 3×3: No. The remaining number of squares is not divisible by 3.
- 4×4: Yes.

$P(n)$: a $2^n \times 2^n$ square grid, with one subsquare removed, can be tiled (covered without overlapping) by "chair" trominoes.

Pf:

i. Induction on n

Let n be an arbitrary, fixed, natural number. (Let $n \in \mathbb{N}$).

Assume $P(n)$, that is a $2^n \times 2^n$ grid, with one square removed can be tiled with "chairs."

I will prove $P(n+1)$, that is a $2^{n+1} \times 2^{n+1}$ grid, with one square removed can be tiled by chairs.

Let G be a $2^{n+1} \times 2^{n+1}$ grid with one square removed. Notice that G can be decomposed into four $2^n \times 2^n$ disjoint quadrant grids. We may assyme, WLOG (wihout loss of generality) that the missing square is in the upper-right quadrant, since otherwise just rotate it there, and rotate back when done. By $P(n)$ I can tile the upper-right quadrant, minus the missing square. By $P(n)$ 3 more times, I can tile the remaining 3 quadrants, omitting for a moment the 3 tiles nearest the centre of G, with chairs. The briefly omitted squares form a chair! So I complete the tiling by adding one more chair. Thus $P(n+1)$.

ii. Base Case

A $2^0 \times 2^0$ grid, with one square removed, is just empty space! This can be tiled with 0 charis. So $P(0)$ is true.