COMP 1805 Discrete Structures I Fall 2021
Problem Set 3 (Lectures 8-14)
Due Monday Nov 8 2:30pm

Purpose & Goals
The questions in this problem set provide challenging yet doable independent practice relating to

- proofs by mathematical induction,
- proofs by strong induction,
- proofs by structural induction,
- predicate logic, and
- propositional logic.

Instructions
Make sure your submission follows the following guidelines.

- Answer each problem to the best of your ability. Partial credit is your friend!
- You may work with your peers, but you must write your solutions in your own words on your own. Do not search for solutions or hints, post the problem set, or otherwise violate academic integrity. Violations (regardless of intent) must be reported to the Dean of Academic Integrity.
- **Type** your solutions into a **pdf format** so that they are legible and professional. You may hand-draw diagrams. Submit your pdf to gradescope. Submit early and often so that last-minute technical problems don't derail you. **Hand-written, non-pdf, or emailed solutions will not be marked, no exceptions.** See piazza for typesetting help.
- Each problem should start on a new page of the document.

Tips
Look at which lectures are deemed most relevant to this problem set in the header of this document, and strongly consider watching the lectures and completing the associated drills and tutorials. Refer to the problem solving tips guide, which also includes homework-specific suggestions.

How to Get Help
When you are stuck and need a little or big push, **please ask for help!**

- piazza. Read the problem set FAQs or search using the search bar. If that fails, post a public question phrasing your query as generally as possible so that other students can benefit. If this is not possible because it reveals your approach, post privately to Instructors. This gives your question the best chance of being answered promptly. Do not direct message or email course staff unless explicitly told this is okay.
- discord. Try piazza first to avoid asking a duplicate questions. Discord is best for informal opinions and comradery; for questions that might be useful to others, **please use piazza.** Don't be offended if questions on discord are answered with "look/ask on piazza."
- office hours. Use for questions that require more back and forth.

Self-Evaluation (10%) When you have completed this problem set, the last thing you should do is complete your self-evaluation on brightspace. However, you might want to keep the evaluation criteria in mind as you do your work.

You will be asked some leading questions to help you asses some "intangible" skills you might have practiced on this assignment, such as:

- do you feel more comfortable with any form of induction (weak, strong, structural)?
- do you generally understand the principle of induction (that it isn't magic, or cheating)?
- do you feel more comfortable with propositional logic and truth tables?
- do you feel more comfortable with predicate logic and equivalences?
- do you feel more comfortable with the definition and notation on the problem set?
- did you use examples to help understand a problem?
- did you draw a diagram to help understand a problem?
- did you try to break your problem into cases and attack each case separately?
- could you explain the problems in your own words to someone else in this class?
- did you find similar or related statements that we've talked about in lecture or tutorial and did you try to understand how the problems were similar and how they were different?
- did you ask a question on piazza or discord that required you to communicate what you understood and what was still confusing?
- did you answer a peer's question on piazza or discord?
- did you read over your solutions and make sure each statement followed from the previous or from an already derived or known fact?
- did you imagine trying to read your proofs as if you'd never seen them before to make sure you have communicated your thoughts clearly, concisely, and precisely?

You do not have to do all of these, but I hope you do some! You will be asked to give yourself a letter grade on these intangibles (A+ - F), and justify your selection with 2-3 sentences.

For more information on the self-evaluation (including why we have self-evaluations) please see the Self-Evaluation section of the syllabus.

---

[1] (*14 pts.*) **Binary**

Tutorial 5's proof of the correctness of `toBinary(m)` establishes that any nonnegative integer m can be represented in binary. Now you'll show that this binary representation is unique—or, at least, unique up to leading zeros. (For example, we can represent 7 in binary as 111 or 0111 or 00111, but only 111 has no leading zeros.)

Prove that every nonnegative integer m that can be represented as a n-bit string is *uniquely* represented as a n-bit string. In other words, prove the following claim:

Claim. For any integer $n \geq 0$, let $a := \langle a_n, a_{n-1}, \ldots, a_0 \rangle$ and $b := \langle b_n, b_{n-1}, \ldots, b_0 \rangle$ be two $(n+1)$-bit sequences. If $\sum_{i=0}^{n} a_i 2^i = \sum_{i=0}^{n} b_i 2^i$ then $a_i = b_i$ for all $i \in \{0, 1, \ldots, n\}$.

Your proof should be by mathematical induction on n. Use the 7-step process from Lectures 7-10.

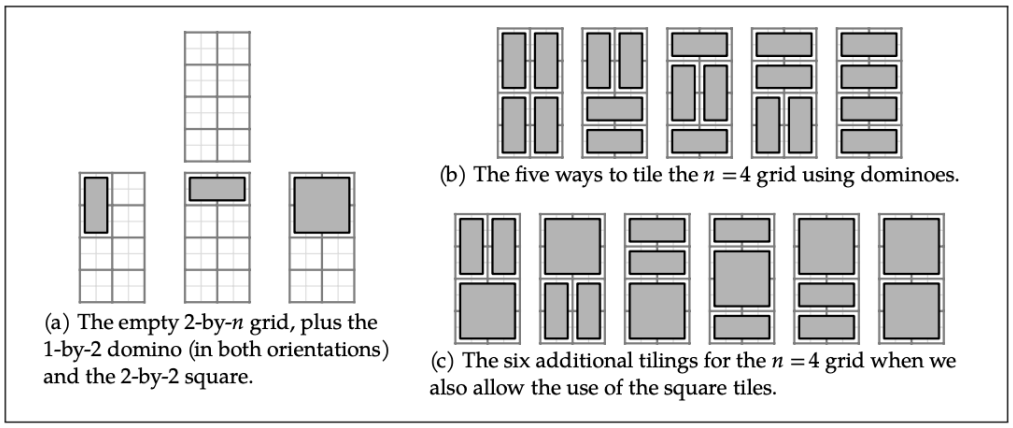
Hint: Note that this is a proof of an "if LHS then RHS" statement, rather than a LHS=RHS so you'll want to adjust everything accordingly.

Hint: Use the fact about sums of powers of 2 from Lecture 7.

[2] (*16 pts.*) **Tiling**

Recall the Fibonacci numbers defined in Lecture 8. If we instead define $g_1 = 1, g_2 = 1$ and $g_n = g_{n-1} + 2g_{n-2}$ for $n \geq 3$ we will get a different sequence of numbers (due to the different recurrence.) What's neat about this new sequence of numbers is that it can be used to "count" different tilings of a 2-by-n grid. Let me explain further.

Suppose you are given a 2-by-n grid that you must tile, using either 1-by-2 dominoes or 2-by-2 squares. The dominoes can be arranged either vertically or horizontally, but they may not overlap, they may not "hang off" the board, and they must cover every square of the grid. (See Figure below.)



(a) The empty 2-by-n grid, plus the 1-by-2 domino (in both orientations) and the 2-by-2 square.

(b) The five ways to tile the $n = 4$ grid using dominoes.

(c) The six additional tilings for the $n = 4$ grid when we also allow the use of the square tiles.

Prove by strong induction on n that the number of different ways of tiling the 2-by-n grid for any $n \geq 1$ is precisely g_{n+1}. (Be careful: it's easy to accidentally count some configurations twice—for example, make sure that you count only once the tiling of a 2-by-3 grid that uses three horizontal dominoes.)

Hint: Try out some examples before trying to prove the result. Look at the figure above.

Hint: Use the 7-step process used in Lectures 8, 10, and Tutorial 5 (for strong induction). There are also some Puzzlers relating to tiling that may or may not be helpful.

[3] (*14 pts.*) **Palindromes!**

Consider the set $\mathsf{PAL} := \{x \in \{a,b\}^* : x = \mathsf{rev}(x)\}$, where $\mathsf{rev}(x)$ denotes the reverse of string x. For example, $abba, aba, bb, a \in \mathsf{PAL}$, but $ab, bba, baba \notin \mathsf{PAL}$.

We can define a set S recursively as follows. A string $s \in S$ if

0. $s = \varepsilon, a, b$, or
1. $s = as_1a$ or $s = bs_1b$ where $s_1 \in S$.

The goal of this problem and the next will be to prove that this recursive set S is a subset of PAL. We *could* show the other direction of containment using induction as well, but we'll leave that for another time (or not).

Use structural induction to prove that $S \subseteq \mathsf{PAL}$. In particular, prove that for any $s \in S$, $s \in \mathsf{PAL}$.

Hint: Note that you're trying to prove $s \in \mathsf{PAL}$ which on face value isn't of our usual LHS=RHS form. But if you look at the definition of PAL you can find an equivalent statement using LHS=RHS.

Hint: Use the 7 steps process of structural induction used in Lectures 9, 11, 12 and Tutorial 5.

Hint: See the examples in Drill 10 for practice with the definitions of this problem.

[4] (*12 pts.*) **Predicate & Propositional Logic**

a. *(4 pts.)* Consider the proposition: *if* $(p \vee q) \Rightarrow r$ *then* $p \Rightarrow r$.

- Explain what it means in words (as far as that is possible);
- Construct its truth table. (Don't forget that the words *if* and *then* correspond to propositional operators and should not be ignored.)
- Is this a tautology?
- Is it satisfiable?

b. *(4 pts.)* Give a proof by construction that there is an unsatisfiable formula in CNF over n variables that has 2^n clauses. Your proof should

- for any $n \geq 1$ describe the unsatisfiable formula φ in CNF with 2^n clauses, and
- argue that the formula is unsatisfiable.

You might want to look at Lecture 12 Drills for some practice with CNF. But as a reminder, a formula φ is in CNF if $\varphi = C_1 \wedge C_2 \wedge \ldots \wedge C_m$ for some $m \geq 1$, where each clause $C_i = (\ell_1 \vee \ell_2 \vee \cdots \vee \ell_{m_i})$ for some m_i, and each ℓ_i is either a variable or the negation of a variable.

Hint: Do try this on small examples first. Try $n = 1$, $n = 2$ and $n = 3$ before you try to prove anything. Once you describe your construction, consider demonstrating it on an example in your solution to help the grader understand and follow along.

c. *(4 pts.)* While the quantifiers $\forall$ and $\exists$ are by far the most common, there are some other quantifiers that are sometimes used. For example, some people use $\exists!\ x \in \mathbb{Z} : P(x)$ to denote "there exists a unique $x \in \mathbb{Z}$ such that $P(x)$" which is true when there is one and only one value of x in the set $\mathbb{Z}$ such that $P(x)$ is true.

Prove that $\exists!$ is "syntactic sugar," in that for any expression φ that uses $\exists! x \in \mathbb{Z} : P(x)$, there is a logically equivalent φ' that uses only the quantifiers $\forall$ and $\exists$; standard propositional logic notation $(\wedge, \neg, \vee, \Rightarrow)$; standard equality/inequality notation $(=, \geq, \leq <, >)$; and the predicate P in the question.

Don't forget to follow the general syntax of predicate logic. You cannot use shorthand such as $\forall y < x$, for example.