

Problem Set 1: (Un)Natural Numbers

Due: **4:19pm, Friday, 13 September**

This assignment is designed to develop your skills with formal definitions and provide some practice with inductive reasoning.

Collaboration Policy: You should do this assignment by yourself and submit your own answers. You may discuss the problems with anyone you want and it is also fine to get help from anyone on problems with LaTeX or Jupyter/Python. You are permitted to use any resources you find for this assignment. You should note in the *Collaborators and Resources* box below the people you collaborated with and any external resources you used (you may exclude resources you used exclusively for help with LaTeX or Jupyter/Python).

Collaborators and Resources: TODO: replace this with your collaborators and resources (if you did not have any, replace this with *None*)

This problem set has two parts, one of which should be completed by writing your answers in the ‘ps1.tex’ LaTeX template, and the other in the PS1 Jupyter notebook. As with Problem Set 0, you will submit (in collab) both a PDF file with your answers to the questions in this template, and your Jupyter notebook.

To get started with this assignment, follow directions similar to those from ps0:

1. Download the Problem Set 1 template from: <https://uvatoc.github.io/ps/ps1.zip>
2. In Overleaf, click on New Project in Overleaf and select Upload Project from the menu.
3. Click Select a .zip file and then select the ps1.zip file you downloaded in step 1.

In the left side of the browser, you should see a file directory containing ps1.ipynb, a Jupyter notebook that you’ll use in the second part (but shouldn’t edit or view in Overleaf), ps1.tex, the template you will modify for this problem set, and uvatoc.sty, a style file that defines useful macros for cs3102 (you are welcome to look at this file, but should not need to modify it). You can click on ps1.tex to see the LaTeX source for this file.

LaTeX Template. As with ps0, the first thing you should do in ps1.tex is set up your name as the author of the submission by replacing the line, `\submitter{TODO: your name}`, with your name and UVA id, e.g., `\submitter{Grace Hopper (gmh1a)}`.

Before submitting your ps1.pdf file, also remember to:

- List your collaborators and resources, replacing the TODO in `\collaborators{TODO: replace ...}` with your collaborators and resources. (Remember to update this before submitting if you work with more people.)
- Replace the second line in ps1.tex, `\usepackage{uvatoc}` with `\usepackage[response]{uvatoc}` so the directions do not appear in your final PDF.

Problem 1 *Induction Practice*

Prove that for any natural number $n \geq 2$, $n! < n^n$.

Note: For this problem (and any other problems where we don't explicitly state that you should use a particular definition), you can use the intuitive informal definition of natural numbers, and assume all of the familiar operations are defined and behave as expected.

Problem 2 *Higher Induction Practice*

Prove that any binary tree of height h has at most 2^{h-1} leaves.

Note: We haven't defined a *binary tree* (and the book doesn't). An adequate answer to this question will use the informal understanding of a binary tree which we expect you have entering this class, but an excellent answer will include a definition of a binary tree and connect your proof to that definition.

Problem 3 *Addition is Commutative*

For this problem, we will use the successor definition of Natural Numbers (from Class 2 and 3):

Definition 1 (Natural Numbers) We define the *Natural Numbers* as:

1. **0** is a Natural Number.
2. If n is a Natural Number, **S**(n) is a Natural Number.

We will use this definition of addition (from class 3):

Definition 2 (Sum) The *sum* of two Natural Numbers a and b (denoted as $a + b$) is defined as:

1. If a is **0**, then $a + b$ is b .
2. Otherwise, a is **S**(p) for some Natural Number p , and $a + b$ is **S**($p + b$).

Prove that addition (as defined above) is *commutative* (that is, for all Natural Numbers a and b , $a + b$ is $b + a$). Note that what “is” means here is they are exactly the same representation (we are not using $=$, since we haven't defined it for our number representation). You can think of all the operations we have defined as just manipulating strings of symbols, and “ x is y ” meaning that x and y are exactly the same sequences of symbols.

Problem 4 *Countable Programs*

Prove that the set of all Python programs that you can execute on your laptop is *countable*.

Problem 5 *Countable Graphs*

The book defines an *undirected graph* (Definition 1.3). We modify this by adding one word to define a *undirected finite graph*:

Definition 3 (Undirected finite graph) An undirected finite graph $G = (V, E)$ consists of a *finite* set V of vertices and a set E of edges. Every edge is a size two subset of V .

Prove that the set of all undirected finite graphs is *countably infinite*.

Problem 6 *Uncountable Sets*

(★)¹ Prove that the set of all undirected graphs (using the book's Definition 1.3, without the constraint that V is finite that was added for the previous problem) is not countable. (Note: be careful in any argument that you make that the graphs you are counting as actually *different*. The nodes on the graph have no labels, so the only way for two graphs to be different is if they have different structure.)

Jupyter Problems

The ps1.zip file includes ps1.ipynb, a Jupyter notebook.

To run this notebook, execute:

```
jupyter notebook ps1.ipynb
```

The notebook contains additional problems and starting code for this assignment. Follow the directions there. Like for Problem Set 0, you will edit the provided ps1.ipynb notebook, and submit your completed jupyter notebook as your assignment.

¹When a problem is marked with a ★, it means we think this problem is challenging enough that students who are not able to solve it well can still get “full credit” (i.e., *gold star*) for the assignment without submitting a good answer to the problem. We still hope everyone will attempt these problems and learn from trying to solve them, but you shouldn't get overly frustrated if you are not able to solve a ★ problem.