Week 4: Eval

Authors: TODO: Cohort Name (names of all who contributed)

Collaborators and Resources: TODO: Replace with any additional collaborators and non-course resources you used

This is a template to help with your write-up for Week 3. The actual problem you will write up will be selected by your Cohort Leader at the Assessed Cohort Meeting.

Clone the Problem Set 4 Template Repository

See the Week 1 template for directions on Getting Started with LaTeX. Similarly to Week 1, one member of your cohort should create a copy of the Problem Set 4 repository, by following these steps (we recommend doing this together, with the one creating the repository sharing her screen for everyone to follow along):

1. Download the Problem Set 4 template from: https://uvatoc.github.io/docs/ps4template.zip
2. In Overleaf, click on Create First Project or New Project in Overleaf and select Upload Project from the menu.
3. Click Select a .zip file and then select the ps4template.zip file you downloaded in step 1.
4. Share the repository with your cohortmates by clicking the "Share" button at the top right of the overleaf window, and entering your cohortmates email addresses in the sharing form.

Click on ps4.tex to see the LaTeX source for this file, which is the file you will modify to prepare your solution. The first thing you should do in ps4.tex is set up your cohort name as the author of the submission by replacing the line, `\submitter{TODO: your name}`, with your the name of your cohort (e.g., `\submitter{Cohort Hopper (Ada Lovelace, Don Knuth)}`). For the list of cohort members, this should usually be everyone in your cohort, but if someone did not contribute during the week, they should not be included in your submission list (and should have informed us about their absence separately).

Before submitting your week4.pdf file, also remember to:

– List your collaborators and resources, replacing the TODO in `\collaborators{TODO: replace ...}` with your collaborators and resources. You do not need to include

– Replace the second line in ps4.tex, `\usepackage{uvatoc}` with `\usepackage[response]{uvatoc}` so the directions do not appear in your final PDF. You can do this by using the LaTeX comment token, %. The rest of the line after a % is treated as a comment. You'll notice after you to this, when you Recompile the document, most of it will disappear (everything in \directions is left out, so only your solution will appear in the submitted document).

Problem 1 *Finite vs. Infinite functions*

To fully appreciate this week's content, it is critical that you understand the difference between finite and infinite functions. The *Textbook section 1.7* defines a finite function to have a fixed-sized input and infinite functions to have unbounded input. Check your understanding of the difference between these two by answering (with proof) the following:

What is the cardinality of the set of all finite functions of the form $\{0,1\}^n \rightarrow \{0,1\}$?

What is the cardinality of the set of all finite functions with binary inputs?

What is the cardinaltiy of the set of all infinite functions with binary inputs?

Problem 2 *Equal to Constant Function (Textbook exercise 5.3)*

For every $k \in \mathbb{N}$ show that there exists a NAND-CIRC straightline program of no more than $c \cdot k$ lines (where c is a constant) which computes $\texttt{EQUALS}_{x'} : \{0,1\}^k \rightarrow \{0,1\}$ where $\texttt{EQUALS}_{x'}(x) = 1$ if and only if $x = x'$.

Problem 3 *Domino Computers*

The *10,000 Domino Computer* adds together two 4-digit numbers. Last week you were tasked with building a circuit of your own for addition as a programming problem. Exercise 4.5 of the textbook (which we're not asking you to complete at this time) concludes that you can make a NAND circuit to add together two n-bit numbers using no more than $9n$ gates. If we assume the Domino Computer used only NAND gates, how many dominoes are required to implement each gate on average? How many dominoes would be required for `LOOKUP_8`?

Use *Theorem 3.12 from the textbook* to give a lower bound on the number of dominoes that would be required if this domino computer for `LOOKUP_8` was re-implemented with AON gates.

Problem 7 *Equal to Constant Function (Textbook exercise 5.3)*

For every $k \in \mathbb{N}$ show that there exists a NAND-CIRC straightline program of no more than $c \cdot k$ lines (where c is a constant) which computes $\texttt{EQUALS}_{x'} : \{0,1\}^k \rightarrow \{0,1\}$ where $\texttt{EQUALS}_{x'}(x) = 1$ if and only if $x = x'$.

Problem 8 *Domino Computers*

The *10,000 Domino Computer* adds together two 4-digit numbers. Last week you were tasked with building a circuit of your own for addition as a programming problem. Exercise 4.5 of the textbook (which we're not asking you to complete at this time) concludes that you can make a NAND circuit to add together two n-bit numbers using no more than $9n$ gates. If we assume the Domino Computer used only NAND gates, how many dominoes are required to implement each gate on average? How many dominoes would be required for `LOOKUP_8`?

Use *Theorem 3.12 from the textbook* to give a lower bound on the number of dominoes that would be required if this domino computer for `LOOKUP_8` was re-implemented with AON gates.

