

Theory of Computation

CS 378, Fall 2022

Small Assignment 1

Due **BY THE BEGINNING OF CLASS** Thursday, September 15

- For this Smaller Assignment, our standard file naming conventions apply: Please submit your typewritten answers in a PDF file named

CS378_SA1_<userid>.pdf

where <userid> is replaced by your Colby userid—for example, my file would be called CS378_SA1_eaaron.pdf. Please use your full userid, including your class year.

To submit this Small Assignment, please use our standard HW submission process: Place the properly named file in your **Private** directory in your CS378 filespace on **filer**. (That is, go to filer.colby.edu/Courses/CS378 to find your folder in the course's filespace; you should find a sub-folder called **Private** in your filespace.) **Please make sure all Problem Set and Smaller Assignment submissions are to your Private folders!**

- As usual for small assignments in CS378 this semester, this is due by the beginning of the next class meeting, and it will be marked on a $\checkmark + / \checkmark / \checkmark - / 0$ scale.
- *A general note for CS378:* As always, please present answers cleanly and **explain them clearly and thoroughly**, giving all details needed to make your answers easy to understand; typed-up (rather than handwritten) answers are required, and using L^AT_EX to typeset your answers is appreciated. (Feel free to talk with your Prof. about using L^AT_EX!) Graders may not award full credit to incomplete or illegible solutions. Clear communication *is* the point, on every assignment.

In general in CS378, unless explicitly specified otherwise, answers should be accompanied by explanations. Answers without explanations may not receive full credit. Please feel free to ask me any questions about explanations that might come up!

Exercises

1. Recall our recursive definition of propositional logic expressions:

Base Given an initial set A of propositional letters (e.g., $p, q, r, \dots$), all elements of A are *PLEs*.

Induction If P, Q are *PLEs*, then the following are also *PLEs* (note that the parentheses are part of the expressions):

- $(\neg P)$
- $(P \wedge Q)$
- $(P \vee Q)$
- $(P \rightarrow Q)$
- $(P \leftrightarrow Q)$

For terminology, we will say the symbols $\neg$, $\wedge$, $\vee$, $\rightarrow$, and $\leftrightarrow$ are the operator symbols of our PLEs.

Consider the following Claim:

Claim: A fully parenthesized propositional logic expression has exactly twice as many parentheses as operator symbols.

Give a fully detailed proof of the Claim by structural induction. As usual in detailed proofs for CS378, simply referring to high-level intuitions is not sufficient—please use formal notation and refer to formal definitions to provide details needed for a full proof.

Hint: It's possible that the proofs for individual inductive cases will have many similarities, but all cases should be spelled out in a full proof for this exercise. Copy-and-paste in L^AT_EX can be a timesaver!