Alex Hubers
CSC:341 Spring 2024

Lab #14A

Due: May 11, 2024

#1 Proper Endpoints

Show that A, the language of properly nested parentheses, is in L.
For example, '(()((())))' $\in A$ but '()((' is not.

#2 Problem Complements

The complement of a language A, written $\overline{A}$, is simply the set of strings not in the language, i.e.,

$$\overline{A} = \{\, w \mid w \notin A \,\}$$

As a concrete example, recall the definition of SAT:

$$\text{SAT} = \{\, \phi \mid \phi \text{ is a satisfiable boolean formula} \,\}.$$

The complement of SAT can therefore be defined as:

$$\overline{\text{SAT}} = \{\, \phi \mid \phi \text{ is not a satisfiable boolean formula} \,\}.$$

For each of the following formulae, determine whether they are in SAT or $\overline{\text{SAT}}$.

1. $(x_1 \vee x_3) \wedge (x_2 \vee x_3) \wedge (x_1 \vee \overline{x_3}) \wedge (\overline{x_2} \vee x_3)$.
2. $(x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2}) \wedge (x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2)$.

#3 Verifying Complements

Analogously, we can consider complexity classes whose languages consist of the complements of languages in the original class. We prefix such classes with "co." For example, the class of languages that are complements of languages in NP is coNP. In the NP case, we can think of coNP as the class of language complements with verifiers that take an input and certificate that report whether the input is not in the original language.

1. An important question we might ask is whether solving a problem's complement is as easy as solving the original problem. In terms of classes, this amounts to asking the question: $\mathsf{NP} = \mathsf{coNP}$? To get at this problem, let's first consider the left-to-right direction of this claim. Suppose that we know that a language A is in NP; thus, we know that there exists a verifier for A. Describe your best attempt at building a verifier for $\overline{A}$ using this assumed verifier for A. Describe where your attempt fails.

2. Now, let's consider the right-to-left direction. Suppose we know that $\overline{A} \in \mathsf{coNP}$; thus, we know there exists a verifier for $\overline{A}$. Describe your best attempt at building a verifier for A using this assumed verifier for $\overline{A}$. Describe where your attempt fails.

3. Finally, summarize your findings from the previous two parts. What is the fundamental difficulty that you encountered when trying to build verifiers in either direction? What does this imply about the relationships between the two classes? Is it clear whether $\mathsf{NP} \subseteq \mathsf{coNP}$ or $\mathsf{coNP} \subseteq \mathsf{NP}$?

#4 Easy Complements

The final section (8.6) of the reading highlights a surprising fact, $\mathsf{NL} = \mathsf{coNL}$! That is, even though we have difficulty navigating between "yes" and "no" answers in NP, we can do so easily in NL! The proof that $\mathsf{NL} = \mathsf{coNL}$ is straightforward—show that $\overline{PATH} \in \mathsf{coNL}$—but the construction is a bit tricky and specific to fit within logspace bounds.

For this exercise, to get a sense of the weirdness involved with reasoning about complements, let's consider a simpler case: $\mathsf{P} = \mathsf{coP}$.

1. Prove that $\mathsf{P} = \mathsf{coP}$. Make sure to prove both directions of this claim ($\mathsf{P} \subseteq \mathsf{coP}$ and $\mathsf{coP} \subseteq \mathsf{P}$).
2. Explain, based on your answer(s) to problem 3, why this proof does not translate to a proof that $\mathsf{NP} = \mathsf{coNP}$.