

HW4 - Pumping Lemma + Myhill Nerode

COMP361 — Suhas Arehalli

Spring 2025

Instructions

As with previous homeworks,

- Deadlines are posted on the course website.
- Solutions should be turned in as a typeset pdf, preferably using LaTeX. I recommend a tool like Overleaf.
- Assignments are individual, and you should not share solutions with other students. However, you are free to discuss problems within the bounds of the academic integrity policy in the syllabus, and should indicate in your assignment the people you got assistance from (including instructors, peers, and preceptors!).
- Proofread your work before submission, and consult the proof style guide to make sure your submissions are both correct and clear.
- Graded problems will be indicated by an asterisk (*) next to the problem number. You are still expected to solve all of the problems.

Questions

1. (Based on Sipser 1.48). **Myhill-Nerode Theorem.** Before completing this problem, it may be helpful to review some of the discussion of the terminology that leads up to this theorem that we saw in class. Since we didn't follow Sipser's presentation exactly, I've summarized our discussion in the Review section at the end of the document. Also note that once this problem is complete, you may use the Myhill-Nerode theorem freely on homeworks/exams to prove regularity or non-regularity.
 - (a) *Prove that if L is recognized by a DFA with k states, the index of L must be $\leq k$.
***HINT**:* This proof will have similar structure to that of the Pumping Lemma. Proceed by contradiction, and maybe consider problem 2 from the prior homework!
 - (b) *Prove that if L has finite index k , then L is recognized by a DFA with exactly k states.
***HINT**:* Construct the DFA. You don't need to formally show that it recognizes L via an inductive argument, but make sure it is clear why the DFA you constructed should recognize L

- (c) Conclude that a language L is regular iff L has a finite index.
 - (d) Conclude that the index of a language L is the number of states in the minimal DFA that recognizes it.
2. Prove, via the following steps, that $L = \{0^n 1^n \mid n \geq 0\}$ is non-regular using the Myhill-Nerode Theorem.
- (a) Consider $\equiv_L$ for the given L . Determine if the following strings are indistinguishable or distinguishable by L . Provide a distinguishing extension z if they are distinguishable.
 - (a) 0^5 and 0^6
 - (b) 0^5 and $0^6 1$
 - (c) $0^6 1$ and $0^7 1^2$
 - (d) Provide an infinite set that is pairwise distinguishable by L . As a bonus, see if you can characterize the full set of equivalence classes of $\equiv_L$, though that is not necessary here (be sure you're confident why)!
 - (e) Conclude via Myhill-Nerode and your previous answer that L is non-regular.
3. Consider the language $L = \{a^n > a^m \mid n > m\}$ with alphabet $\Sigma = \{a, >\}$. For example, $aaaa > a \in L$, but $a > aa \notin L$
- (a) *Prove that L is non-regular via the Pumping Lemma.
 - (b) *Prove that L is non-regular via the Myhill-Nerode Theorem
4. Consider the Language $L = \{w \in \{0, 1\}^* \mid w \text{ is divisible by } 3\}$
- (a) Familiarize yourself (or refamiliarize yourself) with binary representations of non-negative integers.
 - (b) Work from small strings to construct equivalence classes under $\equiv_L$. How many equivalence classes are there? Prove via induction that every string $w \in \Sigma^*$ is Nerode-equivalent to one of the classes you construct.
 - (c) Construct the smallest DFA that recognizes L using Myhill-Nerode.

Review

Before beginning the following problems, it may be helpful to review the following facts from Discrete Math and prior lectures. Feel free to skim or skip this section if you're confident, but you do so at your own peril.

An **Equivalence Relation** is a relation that is symmetric, reflexive, and transitive (review Chapter 0 if this terminology is unfamiliar!). In class, we constructed an equivalence relation in the following steps

Def. For a language L and strings $x, y \in \Sigma^*$, x and y are **distinguishable by L** iff there exists some string $z \in \Sigma^*$ such that $xz \in L$ but $yz \notin L$ or $yz \in L$ but $xz \notin L$. We call that z a **distinguishing extension**.

Thus, x and y are **indistinguishable by L** iff $\forall z \in \Sigma^*, xz \in L \text{ iff } yz \in L$.

Def. x is **Nerode-congruent** to y (notated $x \equiv_L y$) iff x and y are indistinguishable by L .

In class, we argued that $\equiv_L$ is an equivalence relation!

In Discrete, you likely saw that we can use equivalence relations to partition a set into **Equivalence Classes** — sets of elements that are equivalent to each other. Formally, we can look at the equivalence class that any string $w \in \Sigma^*$ belongs to under $\equiv_L$ by constructing

$$C_w = \{x \in \Sigma^* \mid x \equiv_L w\}$$

That is, the equivalence class containing w contains every string indistinguishable from w by L .

A common question we ask about equivalence classes is how many there are under a particular equivalence relation. We also try to describe what properties elements in these classes share.

For instance, we can consider the parity relation: for $x, y \in \mathbb{Z}$, $x \equiv_{\text{parity}} y$ iff $x = y \pmod 2$ (or, x and y share the same parity). We can see that every pair of even numbers is equivalent to each other, and every pair of odd numbers is equivalent to each other, and no pair of even and odd numbers is equivalent. Thus, we have 2 equivalence classes: $C_{\text{odd}} = \{x \in \mathbb{Z} \mid x \text{ is odd}\}$ and $C_{\text{even}} = \{x \in \mathbb{Z} \mid x \text{ is even}\}$.

Moving to Nerode-equivalence, we saw in class that we can construct equivalence classes over Σ^* once we have a language L . Let's consider

$$L = \{w \in \Sigma^* \mid |w| \text{ is even}\}.$$

We then argued that every even length string is indistinguishable by L from every other even length string, and that the same went for odd strings (in short - for fixed z , can x, y with the same length-parity have xz, yz with different length parity? No). However, even and odd length strings were, in fact, distinguishable (xz and yz will have different parity for every z !)! Thus we also had 2 equivalence classes, $C_{\text{even}} = \{w \in \Sigma^* \mid |w| \text{ is even}\}$ and $C_{\text{odd}} = \{w \in \Sigma^* \mid |w| \text{ is odd}\}$.

We then continued by defining a few more helpful terms, drawing from Sipser 1.47:

Def. Given a set $X \subseteq \Sigma^*$ and a language L , X is **Pairwise Distinguishable by L** iff for any $x, y \in X$ such that $x \neq y$, x and y are distinguishable by L .

Def. The **index** of a language L is the size of the largest set $X \subseteq \Sigma^*$ that is still pairwise distinguishable by L . Note that such a finite index might not exist if X is not finite.

Note that, once we've developed a notion of equivalence classes, we can think of a pairwise distinguishable X as drawing elements from different equivalence classes under $\equiv_L$, and the index as the number of equivalence classes under $\equiv_L$.

These set us up to prove the Myhill-Nerode Theorem in Problem 1.