# Proofs Using the Pumping Lemma

CS 234

## 0 Introduction

This document contains examples of good proofs written up from the lecture material. These are not the only ways to write good proofs.

These proofs may contain footnotes explaining different thought processes that occurred in their construction, to help show you how to think about writing proofs. Commentary may also be provided at the end about alternative approaches.

# 1 Proofs

## Equal Lengths

**Theorem 1.** *For any alphabet $\Sigma$ containing at least 2 letters, the language*

$$\{a^n b^n \mid a, b \in \Sigma \wedge n \in \mathbb{N}\}$$

*is not regular.*

*Proof.* Let $\Sigma$ be an arbitrary alphabet containing at least 2 letters, and let $L = \{a^n b^n \mid a, b \in \Sigma \wedge n \in \mathbb{N}\}$.

Now suppose for the sake of contradiction that $L$ is regular. If $L$ is regular, then the pumping lemma applies, so there is some nonzero bound $m$ such that all words in $L$ of length at least $m$ can be pumped.

Since $\Sigma$ contains at least 2 distinct letters, let 2 of those distinct letters be $c$ and $d$.[1] Consider the word $c^m d^m$. This word is in $L$, and because $|c^m d^m| = 2m > m$, this word can be pumped.

In particular, $c^m d^m = xyz$ for some $x, y, z \in \Sigma^*$ where $|xy| \leq m$ and $|y| \geq 1$. Because $|xy| \leq m$ and the first $m$ characters of $c^m d^m$ are all $c$, we know $y = c^{|y|}$. Pumping the word once therefore yields the word $xy^2z = c^{m+|y|}d^m$. By the pumping lemma, this word must also be in the language $L$. However, since $|y| \geq 1$, then $m + |y| \neq m$, so the word cannot be in the language $L$. This is a contradiction.

Since assuming $L$ is regular leads to a contradiction, we can conclude that $L$ is not regular, completing the proof.

□

[1] This is a small wrinkle I added to this proof. Just because $a, b \in \Sigma$ does not mean that we know they are distinct. If they happened to be equal, then the rest of this proof would end up with a string of the form $a^{2m}$, and any pumping would yield $a^{2m+k|y|}$ for some $k$. In the event that $|y|$ is even (and it might be because we can't control it) or $k$ is even, this pumped string actually is of even length and thus writable as $a^\ell a^\ell$ for some $\ell$. Thus, it would be in $L$, ruining the contradiction we are going for.

## Equal Lengths

**Theorem 2.** *$\{a^n \mid n$ is prime$\}$ is not regular.*

*Proof.* Let $L = \{a^n \mid n \text{ is prime}\}$.

Now suppose for the sake of contradiction that $L$ is regular. If $L$ is regular, then the pumping lemma applies, so there is some nonzero bound $m$ such that all words in $L$ of length at least $m$ can be pumped.

Since, as proven previously, there are infinitely many primes, there must be some prime $p$ larger than $m$. Consider now the word $a^p$. This word is in $L$, and because $p > m$, this word can be pumped.

In particular, $a^p = xyz$ for some $x, y, z$ where $|y| \geq 1$. Because all the characters of $a^p$ are all $a$, we know all characters in $x$, $y$, and $z$ are also $a$. Pumping the word $p$ times therefore yields the word $xy^{p+p|y|}z = a^{p(1+|y|)}$. By the pumping lemma, this word must also be in the language $L$. However, since $p(1 + |y|)$ is divisible by both $p$ and $1 + |y|$, both of which are at least 2, the number $p(1 + |y|)$ cannot be prime.[2] Thus the word $a^{p(1+|y|)}$ cannot be in the language $L$. This is a contradiction.

Since assuming $L$ is regular leads to a contradiction, we can conclude that $L$ is not regular, completing the proof.

□

[2] I leave this argument somewhat implicit, but I do point out the key intermediate step you must reach to make the argument work. That is, a prime number is only divisible by 1 and itself, so if both numbers are at least 2, then it cannot be prime. Also $p$ must be at least 2 because it is strictly greater than $m$ which is at least 1, and $|y| + 1 \geq 2$ because $|y| \geq 1$. These points are somewhat trivial to derive and ancillary to the main argument, so it is ok to leave them at this level of implicitness, in particular because I trust my audience (you) to understand, especially when the key intermediate step *is* made explicit.