The Pumping Lemma for Regular Languages

Prof. Shrisha Rao
IIIT Bangalore

srao@iiitb.ac.in

2025-08-21

A Small Exercise

Give a DFA/NFA for the language $\{0^n 1^n \mid n \geq 0\}$.
This language consists of strings ϵ, 01, 0011, 000111, etc.

Lessons Learned

- Not every language is regular. There exist well-defined languages that are not regular.
- Not every infinite subset of a regular language is regular ($\{0^n 1^n \mid n \geq 0\}$ is a subset of $(0+1)^*$ which is regular). (Regularity is NOT closed under the subset/superset relations.)

Proving that a language is not regular

- This is done using the Pumping Lemma for regular languages.
- It relies on the fact that any regular language is accepted by a deterministic automaton with a finite number of states, so that for all strings in the language beyond a certain length, a certain state in the automaton must repeat. (Think *pigeonhole principle*.)

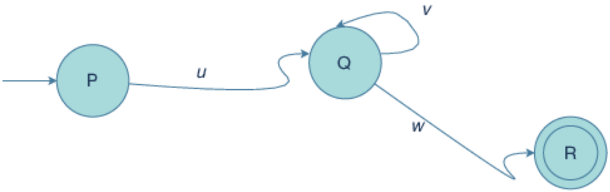
Pumping Lemma for Regular Languages

Let L be a regular language accepted by a DFA with k states. Then for any string $z \in L$ with $|z| \geq k$ (z is of length at least k), we can find an early internal substring that can be "pumped." That is, z can be split into three parts uvw, where

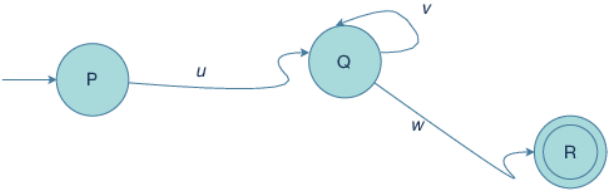
- $|v| \geq 1$ (v is not the empty string);
- $|uv| \leq k$ (uv is of length less than k); and
- $uv^iw \in L$, $\forall i \geq 0$.

Proof

Let z be a string of length at least k in a regular language L. If a DFA with k states accepts L, then by the pigeonhole principle, there must be a state Q that occurs twice (or more) while processing z.



Proof—cont'd



u is the portion of z from start state P until the first occurrence of Q.

v is the portion of z between visits to Q.

w is the remainder of the string, from Q to R (the accept state).

Considering that the DFA is in the same state whether it gets to B the first, second, or ith time, if it accepts uvw, it must also accept

$$uv^2w, uv^3w, \dots uv^iw \text{ for all } i \geq 0.$$

Using the Pumping Lemma

- The pumping lemma cannot be used to prove that a language L is regular; rather, it is used to prove that L is *not* regular, because the lemma is contradicted by it.
- The trick is to find a sufficiently long string z in L so that no matter how some non-empty substring v in z is pumped, we always come up with a string which is not in L.
- In many cases, the closure property of regular languages under complementation (if L is regular, then so is $\overline{L}$) is used.

Example: Proving that $\{0^n 1^n\}$ is not regular

Assume that $L = \{0^n 1^n \mid n \geq 0\}$ is accepted by some DFA with a finite k number of states.
Consider $z = 0^k 1^k$, a string in L. Split z as uvw, as per the Pumping Lemma. Now because $|uv| \leq k$, it follows that v consists solely of 0s. But then on pumping multiple copies of v, we end up with strings of the form $0^a 1^b$, with $a > b$, which are not in L.

☞ Therefore, L is not a regular language.

Exercises

Use the Pumping Lemma for the following.

1. Prove that the language of repeated strings over {0,1} $\{w\#w \mid w \in (0+1)^*\}$ is not regular. (The symbol $\#$ is a special marker that is only used once.)
2. Prove that the language of all *palindromes* over {0,1} $\{w\#w^R \mid w \in (0+1)^*\}$ is not regular.
3. Prove that the language of all *unbalanced* brackets using the symbols (and) is not regular. Strings (()), ((()())), etc., are balanced, but (())),)(()), etc., are not.
4. Prove that the unary language $\{1^c \mid c \text{ is a composite number}\}$ is not regular. (Numbers other than primes, e.g., 4, 6, 8, 9, 10, etc., are composites.)

Hints

- Some problems are straight-forward (but need to be worked out properly).
- In some other problems, a simple application of the Pumping Lemma is not possible. One has to use the complementarity property of regular languages.
- "If a language is regular, then its complement is regular" gives us, "If the complement of a language is not regular, then the language itself is not regular either."
- Thus, the trick there is to identify the complement languages for these cases, and use the Pumping Lemma on them.

The Language of Unbalanced Brackets

- It is not easy to prove that L, the language of all unbalanced brackets is not regular, using the Pumping Lemma.
- However, the language $\overline{L}$, the language of balanced brackets, can be easily shown to be not regular in that way.
- Proof hint: by contradiction, assume that $\overline{L}$ is regular, and is accepted by a DFA with k states. Consider the string $(^k)^k$.
- Therefore, $\overline{L}$ is not regular, and by the complementarity property, neither is L.

The Language 1^c

- Here again, it is not easy to directly prove that this language L is not regular, using the Pumping Lemma.
- Consider $\overline{L} = \{1^p \mid p \text{ is a prime}\}$ instead.
- Proof hint: If $\overline{L}$ is accepted by a DFA with k states, then there exist integers $a, b > 0$ such that $a + bi$ is a prime for all $i > 0$. However, this is clearly false, as $a + bi$ is not a prime when $i = a, 2a$, etc.
- Therefore, $\overline{L}$ is not regular, and by the complementarity property, neither is L.