COMP2210: Theory of Computation

Lecture 5

Limitations of regular languages

Gennaro Parlato
Learning Outcomes

By the end of this session you should be able to...

• ... *prove* the pumping Lemma
Regular Languages

- we have seen that the class of regular languages is quite robust:
  - it is the class of languages accepted by a family (DFA, NFA, $\epsilon$NFA) of finite automata, the class of languages matched by reg exp

- Q: Are all languages regular?

- A: No. Using a counting argument, there are countably ($\sim N$) many regular expressions but, in general, uncountably ($\sim 2^N$) many languages.
A non-regular language

\{ a^n b^n \mid n \in \mathbb{N} \}  

- **rough idea:** any DFA has finitely many states. To accept \( a^{1000} b^{1000} \) and reject \( a^{1000} b^{999} \) it has to be able to count the number of \( a \)'s and remember how many have been seen. Since the number is unbounded, there is no way to do this with a finite number of states.
Suppose that $M$ is an automaton that accepts $\{a^n b^n \mid n \in \mathbb{N}\}$. Suppose that $M$ has $k$ states. Let $n > k$. Then $\exists q \in Q$ such that:

- Then $M$ accepts also $a^{n-l} b^n$. Contradiction!
Pumping lemma

Suppose that $A$ is regular. There exists $k \in \mathbb{N}$ such that for all strings $x, y, z$ with $xyz \in A$ and $\#y \geq k$:

- there exist strings $u, v, w$ with:
  - $y = uvw$ and $v \neq \epsilon$;
  - $xuv^i wz \in A$ for all $i \geq 0$.

Proof: (Exercise) Hint: It is similar to the one on the previous slide.
Learning Outcomes

You should be able to...

- **prove** the pumping Lemma