WEEK 9 WORKSHOP

MATH2301, SEMESTER 2, 2025

In all the problems, the alphabet is $\{0, 1\}$.

1. Non-regular languages

Show that the following languages are not regular by (a) a pumping-lemma argument and (b) using the Myhill-Nerode theorem. It takes practice to be able to apply either of the two strategies. Do not be discouraged if you find it difficult. It is difficult!

1.1. **Problem.** $L = \{0^m 1^n \mid m \neq n\}$.

Solution.

(1) Using the pumping lemma.
Suppose L is recognised by a DFA M with n states. Consider $w = 0^{n+1}1^{n+1}$. Then M rejects w because $w \notin L$. But M must go into a loop while reading the first $n+1$ zeros. If we repeat a substring of 0^{n+1} that produces a loop, we get $w' = 0^{n+1+l}1^n$ and $w' \in L$. But repeating a loop does not change the end state, so M also rejects w'. This contradicts that M recognises L.

(2) Using the Myhill-Nerode theorem.
Consider the strings $01, 00, 000, \cdots$. We claim that they are pairwise non-equivalent under $\sim_L$. Indeed, consider $x = 0^m$ and $y = 0^n$ where $m \neq n$. Taking $z = 1^m$ gives $xz \notin L$ but $yz \in L$. So there are infinitely many equivalent classes for $\sim_L$, we conclude that L is not regular.

1.2. **Problem.** $L = \{0^n \mid n \text{ is a power of } 2\}$.

Solution.

(1) Pumping lemma.
Suppose there is a DFA M that recognises L and has n states. Consider $w = 0^{2^n}$, which is in L. While reading w, M must go in a loop within the first n bits. Repeating this loop gives $w' = 0^{2^n+l}$, where $l \leqslant n$. Observe that $n < 2^n$ for all $n \geqslant 1$. So $2^n + l < 2^{n+1}$, which is the next power of 2. So $w' \notin L$. But M gives the same decision for w and w'. Contradiction.

(2) Myhill-Nerode.
Consider 0^{2^n} for $n = 1, 2, 3, \cdots$. Show that all of these are pairwise non equivalent.

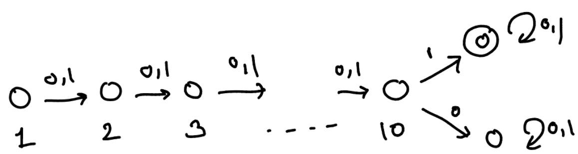
2. Exponential growth

We see how the process of going from an NFA to a DFA can produce exponential blow-up in size.

Let L be the set of strings whose 10-th position is 1 (so the string has to be at least 10 letters long).

2.1. **Problem.** Construct a DFA/NFA for L. It should have about 10 states.

Solution. A DFA for L is



2.2. **Problem.** Using your automaton, construct an NFA for L^{rev}. Describe L^{rev} in words.

Date: September 29, 2025.

Solution. Reverse the arrows, put the start state at the accept state, and make (1) the accept state. This gives an NFA for L^{rev}.

2.3. **Problem.** Think about converting the NFA for L^{rev} to a DFA. Convince yourself that the conversion process would produce a DFA with about 2^{10} states.

Solution. DIY!

2.4. **Problem.** Prove that a DFA for L^{rev} cannot have fewer than 2^{10} states. To do so, show that all strings of length 10 are inequivalent under the Myhill-Nerode equivalence relation $\sim$.

Solution. Suppose x and y are two strings of length 10 that are not equivalent. Suppose x and y differ at the j-th letter, where $1 \leqslant j \leqslant 10$. Take $z = 0^{10-j}$. Then xz and yz differ at the 10th letter from the end. So only one of these two is in L. This shows that x and y are not equivalent.

3. More non-regular languages (if time permits)

Try to decide if the following languages are regular or not.

(1) $\{0^n \mid n \text{ is a perfect square.}\}$
(2) $\{0^n \mid n \text{ is a prime number.}\}$
(3) On the alphabet $\{(,)\}$, the set of well-formed parantheses. For example:
Not well-formed: (or) or)(or ())(
Well-formed: () or ()() or (()()).

4. Counting words in a regular language (if time permits)

This is certainly beyond the scope of the course, and only for your curiosity/amusement.
Let L be a language. Given a number n, let $w(n)$ denote the number of words in L of length n. Define

$$f(z) = \sum_{n=0}^{\infty} w(n) z^n.$$

For example:

(1) If $L = (0|1)^*$, then $w(n) = 2^n$ and $f(z) = \sum 2^n z^n = \frac{1}{1-2z}$.
(2) If $L = 0^*$, then $w(n) = n$ and $f(z) = \sum n z^n = \frac{z}{(1-z)^2}$.

Try to see if you can prove the following.
Theorem. If L is a regular language, then $f(z)$, as defined above, is a rational function (polynomial divided by polynomial).