4 Showing that a language is not regular

Regular languages are languages which can be recognized by a computer with finite (i.e. fixed) memory. Such a computer corresponds to a DFA. However, there are many languages which cannot be recognized using only finite memory, a simple example is the language

$$L = \{0^n 1^n \mid n \in \mathbb{N}\}$$

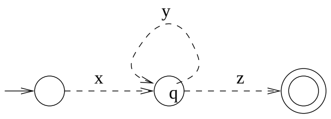
i.e. the language of words which start with a number of 0s followed by the **same** number of 1s. Note that this is different to $L(0^*1^*)$ which is the language of words of sequences of 0s followed by a sequence of 1s but the umber has not to be identical (and which we know to be regular because it is given by a regular expression).
Why can L not be recognized by a computer with fixed finite memory? Assume we have 32 Megabytes of memory, that is we have $32*1024*1024*8 = 268435456$ bits. Such a computer corresponds to an enormous DFA with $2^{268435456}$ states (imagine you have to draw the transition diagram). However, the computer can only count until $2^{268435456}$ if we feed it any more 0s in the beginning it will get confused! Hence, you need an unbounded amount of memory to recognize n.
We shall now show a general theorem called *the pumping lemma* which allows us to prove that a certain language is not regular.

4.1 The pumping lemma

Theorem 4.1 *Given a regular language L, then there is a number $n \in \mathbb{N}$ such that all words $w \in L$ which are longer than n ($|w| \geq n$) can be split into three words $w = xyz$ s.t.*

1. *$y \neq \epsilon$*
2. *$|xy| \leq n$*
3. *for all $k \in \mathbb{N}$ we have $xy^kz \in L$.*

Proof: For a regular language L there exists a DFA A s.t. $L = L(A)$. Let us assume that A has got n states. Now if A accepts a word w with $|w| \geq n$ it must have visited a state q twice:



We choose q s.t. it is the first cycle, hence $|xy| \leq n$. We also know that y is non empty (otherwise there is no cycle).
Now, consider what happens if we feed a word of the form xy^iz to the automaton, i.e. s instead of y it contains an arbitrary number of repetitions of y, including the case $i = 0$, i.e. y is just left out. The automaton has to accept all such words and hence $xy^iz \in L$

□

4.2 Applying the pumping lemma

Theorem 4.2 *The language $L = \{0^n 1^n \mid n \in \mathbb{N}\}$ is not regular.*

Proof: Assume L would be regular. We will show that this leads to contradiction using the pumping lemma.
Now by the pumping lemma there is an n such that we can split each word which is longer than n such that the properties given by the pumping lemma hold. Consider $0^n1^n \in L$, this is certainly longer than n. We have that $xyz = 0^n1^n$ and we know that $|xy| \leq n$, hence y can only contain 0s, and since $y \neq \epsilon$ it must contain at least one 0. Now according to the pumping lemma $xy^0z \in L$ but this cannot be the case because it contains at least one 0 less but the same number of 1s as 0^n1^n.
Hence, our assumption that L is regular must have been wrong.

□

It is easy to see that the language

$$\{1^n \mid n \text{ is even}\}$$

is regular (just construct the appropriate DFA or use a regular expression). However what about

$$\{1^n \mid n \text{ is a square}\}$$

where by saying n is a square we mean that is there is an $k \in \mathbb{N}$ s.t. $n = k^2$. We may try as we like there is no way to find out whether we have a got a square number of 1s by only using finite memory. And indeed:

Theorem 4.3 *The language $L = \{1^n \mid n$ is a square$\}$ is not regular.*

Proof: We apply the same strategy as above. Assume L is regular then there is a number n such we can split all longer words according to the pumping lemma. Let's take $w = 1^{n^2}$ this is certainly long enough. By the pumping lemma we know that we can split $w = xyz$ s.t. the conditions of the pumping lemma hold. In particular we know that

$$1 \leq |y| \leq |xy| \leq n$$

Using the 3rd condition we know that

$$xyyz \in L$$

that is $|xyyz|$ is a square. However we know that

$$\begin{aligned} n^2 &= |w| \\ &= |xyz| \\ &< |xyyz| && \text{since } 1 \leq |y| = |xyz| + |y| \\ &\leq n^2 + n && \text{since } |y| \leq n \\ &< n^2 + 2n + 1 \\ &= (n+1)^2 \end{aligned}$$

To summarize we have

$$n^2 < |xyyz| < (n+1)^2$$

That is $|xyyz|$ lies between two subsequent squares. But then it cannot be a square itself, and hence we have a contradiction to $xyyz \in L$.
We conclude L is not regular. □
Given a word $w \in \Sigma^*$ we write w^R for the word read backwards. I.e. $\mathtt{abc}^R = \mathtt{bca}$. Formally this can be defined as

$$\begin{aligned} \epsilon^R &= \epsilon \\ (xw)^R &= w^R x \end{aligned}$$

We use this to define the language of even length palindromes

$$L_{\text{pali}} = \{ww^R \mid w \in \Sigma^*$$

I.e. for $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ we have $\mathtt{abba} \in L_{\text{pali}}$. Using the intuition that finite automata can only use finite memory it should be clear that this language is not regular, because one has to remember the first half of the word to check whether the 2nd half is the same word read backwards. Indeed, we can show:

Theorem 4.4 *Given $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ we have that L_{pali} is not regular.*

Proof: We use the pumping lemma: We assume that $L_{\mathtt{pali}}$ is regular. Now given a pumping number n we construct $w = \mathtt{a}^n\mathtt{bba}^n \in L_{\text{pali}}$, this word is certainly longer than n. From the pumping lemma we know that there is a splitting of the word $w = xyz$ s.t. $|xy| \leq n$ and hence y may only contain 0s and since $y \neq \epsilon$ at least one. We conclude that $xz \in \mathrm{L}_{\text{pali}}$ where $xz = \mathtt{a}^m\mathtt{bba}^n$ where $m < n$. However, this word cannot be a palindrome since only the first half contains any a s.
Hence our assumption L_{pali} is regular must be wrong. □
The proof works for any alphabet with at least 2 different symbols. However, if Σ contains only one symbol as in $\Sigma = \{1\}$ then L_{pali} is the language of an even number of 1s and this is regular $L_{\text{pali}} = (11)^*$.