CHAPTER 1
Regular Languages

Contents

• Finite Automata (FA or DFA)
  • definitions, examples, designing, regular operations

• Non-deterministic Finite Automata (NFA)
  • definitions, equivalence of NFAs and DFAs, closure under regular operations

• Regular expressions
  • definitions, equivalence with finite automata

• Non-regular Languages
  • the pumping lemma for regular languages
Non-regular Languages

• To understand the power of finite automata we must also understand their limitation.

• We will show that certain languages cannot be recognized by any finite automaton.

• Try to build an automaton that recognizes the language

$$L = \{0^n1^n : n = m\}.$$  

• The automaton starts by seeing 0 inputs.
• It has to remember the exact number of 0 inputs, since it will later check that number against the number of 1 inputs.
• But the number of 0 inputs can be arbitrary large.
• Intuitively, no finite number of states can remember the exact number of 0 inputs.
• We conclude that this language is not regular.

• The *Pumping Lemma* for regular languages formalize this argument.
Pumping Lemma

**Lemma.** For any regular language $L$,

there exists a number $p \geq 1$ such that

for every word $w \in L$ with at least $p$ letters

there exist $x, y, z$ with $w = xyz$ and $|y| > 0$ and $|xy| \leq p$ such that

for every number $i \geq 0$, $xy^iz \in L$.

• We call $p$ the **pumping number** of $L$, and $xyz$ the **pumping decomposition** of $w$.

**Proof**

• Consider a regular language $L$.
• $L$ is accepted by some finite automaton $M$.
• Let $p$ be the number of states of $M$.
• Now consider a word in $L$ with at least $p$ letters.
• Then $w$ is accepted by $M$ along some path that contains a loop.
• We can construct other paths of $M$ by going through the loop $0,1,2, \ldots$ times.
• These paths also accept words in $L$.
• In other words, any accepting word $w$ of length at least $p$ can be “pumped” to find infinitely many other accepted words.
How to prove that a language is not regular?

• Suppose we want to prove that a language $L$ is not regular.

• We can do this by showing that the pumping lemma does not hold for $L$; that is, we prove the negation of the pumping lemma:

  for any number $p \geq 1$
  there exists a word $w \in L$ with at least $p$ letters such that
  for all $x, y, z$ with $w = xyz$ and $|y| \geq 0$ and $|xy| \leq p$
  there exists a number $i \geq 0$ such that $xy^iz \not\in L$.

• We have to consider all possibilities for the pumping number $p$,
• all possibilities for the pumping decomposition $x,y,z$ (often by case analysis).
• But we are free to choose a single word $w$,
• and a single iteration number $i$.
• Choosing a suitable $w$ is usually the crux of the proof (one needs a bit of creative thinking)
• For $i$, we can typically choose $i=0$ or $i=2$.

• Example: $L = \{0^m1^n : n = m\}$ is not regular.
• Choose any pumping number $p$ (we know only that $p \geq 1$). Choose $w = 0^p1^p$.
• Consider any pumping decomposition $w = xyz$ ($|y| \geq 0$ and $|xy| \leq p$).
• Hence $x = 0^a$ and $y = 0^b$ and $z = 0^{p-a-b}1^p$, for $b \geq 1$.
• Choose $i=2$. Since $b \geq 1$, $xy^2z = 0^{p+b}1^p$ is not in $L$. 
More Examples

Example 2: \( L_2 = \{xx : x \in \{0,1\}^*\} \) is not regular.

- Choose any pumping number \( p \) (we know only that \( p \geq 1 \)).
- Choose \( w = 10^p10^p \).
- Consider any pumping decomposition \( w = xyz \) (\(|y| > 0\) and \(|xy| \leq p\)).
- There are two possibilities;
  a) \( x = 10^a \) and \( y = 0^b \) and \( z = 0^{p-a-b}10^p \), for \( b \geq 1 \).
  b) \( x = \varepsilon \) and \( y = 10^b \) and \( z = 0^{p-b}10^p \).
- Choose \( i=2 \). We need to show that \( xy^2z \) is not in \( L_2 \).
  a) \( xy^2z = 10^{p+b}10^p \), which is not in \( L_2 \), since \( b \geq 1 \)
  b) \( xy^2z = 10^b10^p10^p \), which is not in \( L_2 \), since it contains three 1’s.

Example 3: \( L_3 = \{1^{n^2} : n \geq 0\} \) is not regular.

- Choose any pumping number \( p \) (we know only that \( p \geq 1 \)).
- Choose \( w = 1^{p^2} \).
- Consider any pumping decomposition \( w = xyz \) (\(|y| > 0\) and \(|xy| \leq p\)).
- Hence, \( x = 1^a \) and \( y = 1^b \) and \( z = 1^{p^2-a-b} \), for \( b \geq 1 \) and \( a + b \leq p \).
- Choose \( i=2 \). We need to show that \( xy^2z = 1^{p^2+b} \) is not in \( L_3 \), i.e., \( p^2 + b \) is not a square.
- Indeed, \( b \geq 1 \Rightarrow p^2 + b > p^2 \). \( a + b \leq p \Rightarrow p^2 + b \leq p^2 + p < (p + 1)^2 \).
Proving (non)regularity.

To prove that a language $L$ is regular, there are essentially two options:

1. Find a finite automaton (or regular expression) that defines $L$.
2. Show that $L$ can be built from simpler regular languages using operations that are known to preserve regularity \textit{(i.e., $\cup$, $\cap$, $\circ$, $\ast$)}.

To prove that a language $L$ is not regular, there are again two options:

1. Show that the negation of the pumping lemma holds for $L$.
2. Show that a language that is known to be non-regular can be built from $L$ and languages that are known to be regular using operations that are known to preserve regularity.

Example (of the second proof technique):

$L_4 = \{w \in \{0,1\}^*: w \text{ contains the same number of } 1\text{'s and } 0\text{'s} \}$ is not regular, since $L = L_4 \cap (0^*1^*)$ (if $L_4$ were regular, then $L$ would also be regular, which contradicts the first example).