Introduction to Computability Theory

Lecture4: Non Regular Languages

Lecture Slide Reference: UC San Diego

Lecture Outline

1. Motivate the Pumping Lemma.
2. Present and demonstrate the **pumping** concept.
3. Present and prove the **Pumping Lemma**.
4. Use the pumping lemma to prove some languages are not regular.

Introduction and Motivation

In this lecture we ask: Are all languages regular?

The answer is negative.

The simplest example is the language

$$B = \{a^n b^n \mid n \geq 0\}$$

Try to think about this language.

Introduction and Motivation

If we try to find a DFA that recognizes the language $B = \{a^n b^n \mid n \geq 0\}$, it seems that we need an infinite number of states, to "remember" how many a-s we saw so far.

Note: This is **not a proof!**

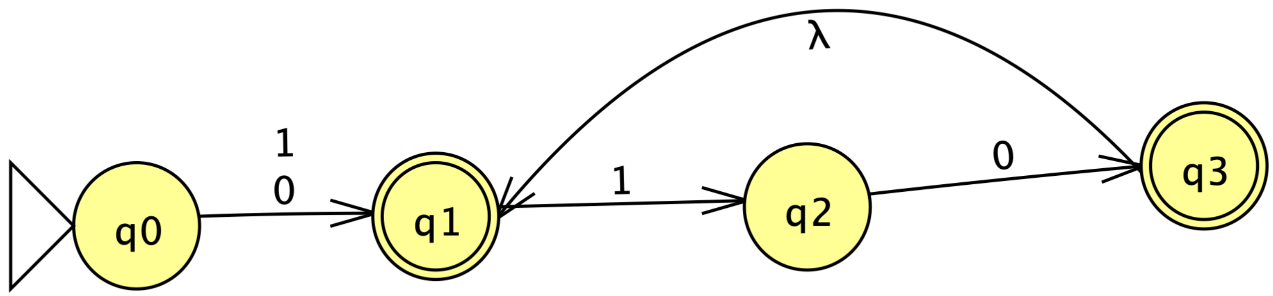
Perhaps a DFA recognizing B exists, but we are not clever enough to find it?

Introduction and Motivation

The **Pumping Lemma** is the formal tool we use to prove that the language B (as well as many other languages) is not regular.

What is Pumping?

Consider the following NFA, denoted by N:



It accepts all words of the form (0 U 1)(01)*

.

What is Pumping?

Consider now the word $110 \in L(N)$.

Pumping means that the word 110 can be divided into two parts: 1 and 10, such that for any $i \geq 0$, the word $1(10)^i \in L(n)$.

We say that the word 110 can be **pumped**.

For $i = 0$ this is called ***down pumping***.

For $i > 1$ this is called ***up pumping***.

What is Pumping?

A more general description would be:

A word $w \in L$, **can be pumped** if $w = xy$ and for each $i \geq 0$, it holds that $xy^i \in L$.

Note: the formal definition is a little more complex than this one.

The Pumping Lemma

Let A be a regular language. There exists a number p such that for every $w \in A$, if $|w| \geq p$ then w may be divided into three parts $w = xyz$, satisfying:

1. for each $i \geq 0$, it holds that $xy^i z \in A$.
2. $|y| > 0$.
3. $|xy| \leq p$.

Note: Without req. 2 the Theorem is **trivial**.

Proof of the Pumping Lemma

Let D be a DFA recognizing A and let p be the number of states of D. If A has no words whose length is at least p, the theorem holds **vacuously**. Let $w \in A$ be an arbitrary word such that $|w| \geq p$. Denote the symbols of w by $w = w_0, w_2, \ldots, w_m$ where $m = |w| \geq p$.

Proof of the Pumping Lemma

Assume that $q_0, q_1, \ldots, q_p, \ldots q_m$ is the sequence of states that D goes through while computing with input w. For each k, $0 \le k < m$, $\delta(q_k, w_k) = q_{k+1}$. Since $w \in A$, $q_m \in F_D$.

Since the sequence $q_0, q_1, \ldots, q_p$ contains $p+1$ states and since the number of states of D is p, that there exist two indices $0 \le i < j \le p$, such that $q_j = q_i$.

Proof of the Pumping Lemma

Denote $x = w_1 w_2 \dots w_{i-1}$, $y = w_i w_{i+1} \dots w_{j-1}$ and $z = w_j w_{j+1} \dots w_m$.

Note: Under this definition $|y| > 0$ and $|xy| \le p$.

By this definition, the computation of D on $x = w_1 w_2 \dots w_{i-1}$ starting from q_0, ends at q_i .

By this definition, the computation of D on $z = w_j w_{j+1} \dots w_m$, starting from q_j , ends at q_m which is an accepting state.

Proof of the Pumping Lemma

The computation of D on $y = w_i w_{i+1} \ldots w_{j-1}$ starting from q_i, ends at q_j. Since $q_i = q_j$, this computation starts and ends at the same state.

Since it is a circular computation, it can repeat itself k times for any $k \geq 0$.

In other words: for each $i \geq 0$, $xy^i z \in A$.

Q.E.D.

Illustration of Pumping

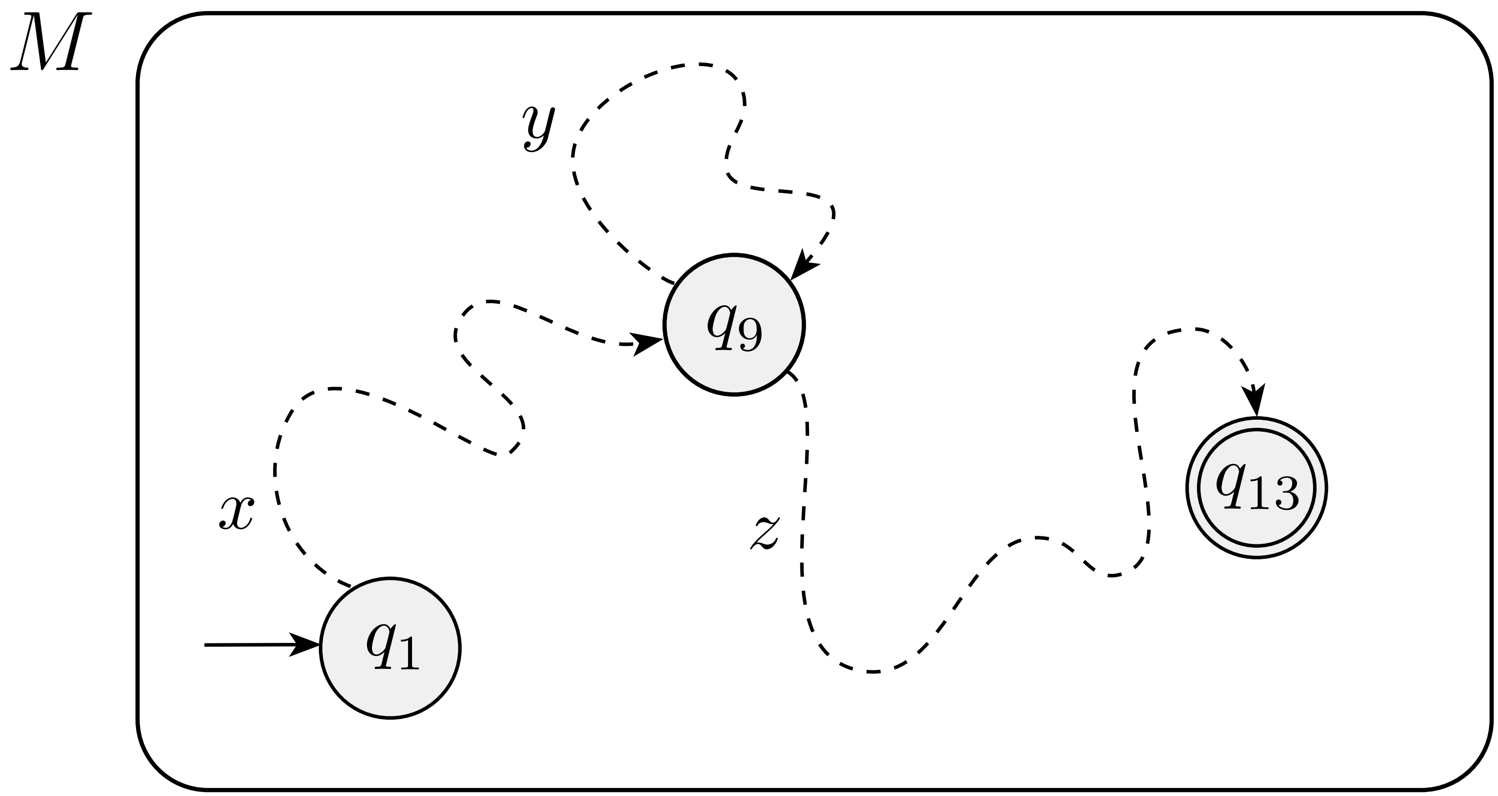


FIGURE 1.72
Example showing how the strings x, y, and z affect M