Automata Theory and Formal Languages

Sommer Term 2026

8^{th} Lecture

Pumping Lemma

Prof. Dr. Matthias Mnich *Institute for Algorithms and Complexity*

Pumping Lemma

Prove that a Language is Regular

We know how to prove that a language is regular:

Prove that a Language is Regular

We know how to prove that a language is regular:

$\Rightarrow$ Construct a regular grammar, an NFA/DFA/ε-FA, or a regular expression

Prove that a Language is Regular

We know how to prove that a language is regular:

$\Rightarrow$ Construct a regular grammar, an NFA/DFA/ε-FA, or a regular expression

But how to prove that a language is *not* regular?

Prove that a Language is Regular

We know how to prove that a language is regular:

$\Rightarrow$ Construct a regular grammar, an NFA/DFA/ε-FA, or a regular expression

But how to prove that a language is *not* regular?

$\Rightarrow$ Difficult

Prove that a Language is Regular

We know how to prove that a language is regular:

$\Rightarrow$ Construct a regular grammar, an NFA/DFA/ε-FA, or a regular expression

But how to prove that a language is *not* regular?

$\Rightarrow$ Difficult

Requires to prove that no regular grammar (NFA/DFA/ε-FA/regular expression) can exist.

Pumping Lemma

The so called *Pumping Lemma* allows to prove for certain languages (not all) that they are not regular.

Pumping Lemma

The so called *Pumping Lemma* allows to prove for certain languages (not all) that they are not regular.

Pumping Lemma

Let L be a regular language. Then there exists a number $n \in \mathbb{N}$ such that all strings $x \in L$ with $|x| \geq n$ can be divided into

$$x = uvw$$

such that

1. $|uv| \leq n$
2. $|v| \geq 1$
3. $\forall m \in \mathbb{N}_0$ it holds that $uv^m w \in L$

Pumping Lemma

Pumping Lemma (as a logic expression)

$$
\begin{aligned}
L \in \mathcal{L}_3 \Rightarrow \exists n \in \mathbb{N} : \forall x \in L : |x| \geq n \Rightarrow \exists u, v, w : \;\; & x = uvw \land \\
& |uv| \leq n \land \\
& |v| > 0 \land \\
& \forall m \in \mathbb{N}_0 : uv^m w \in L
\end{aligned}
$$

where $\mathcal{L}_3$ are the regular languages.

Pumping Lemma

Idea of Proof

The pumping lemma is based on the fact that there exists a DFA M accepting L. If L is an infinite language it contains strings that are longer than the number of states that M. Thus, the automaton has to run through a cycle in order to accept the string x.

Pumping Lemma

Idea of Proof

The pumping lemma is based on the fact that there exists a DFA M accepting L. If L is an infinite language it contains strings that are longer than the number of states that M. Thus, the automaton has to run through a cycle in order to accept the string x.

Proof

If L is finite there is a string x_{max} within L that has maximum length. Let $k = |x_{\text{max}}|$. If one now chooses $n = k + 1$ then the assumption $|x| \geq n$ is wrong for all strings $x \in L$ such that the logical statement of the pumping lemma is fulfilled.

Pumping Lemma

If L is infinite then there exists a DFA $M = (Z, \Sigma, \delta, z_1, E)$ with $L = L(M)$ and $n = |Z|$.

Pumping Lemma

If L is infinite then there exists a DFA $M = (Z, \Sigma, \delta, z_1, E)$ with $L = L(M)$ and $n = |Z|$.

Now let x be an arbitrary string in L with $|x| \geq n$.

Pumping Lemma

If L is infinite then there exists a DFA $M = (Z, \Sigma, \delta, z_1, E)$ with $L = L(M)$ and $n = |Z|$.

Now let x be an arbitrary string in L with $|x| \geq n$.

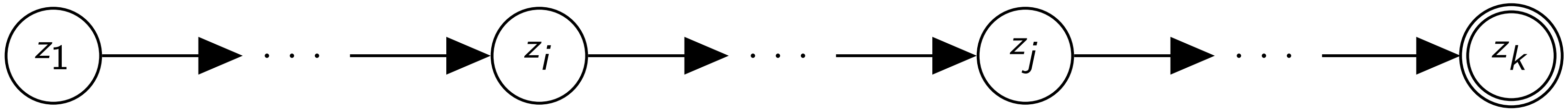
Let $z_1 \to \cdots \to z_k$ be the state sequence traversed when accepting x. In particular we have $k = |x| + 1$. Since M has only n states, there has to be a cycle when n characters have been read. Thus there must be $i < j$ with $i, j \in \{1, \ldots, n+1\}$ and $z_i = z_j$.

Pumping Lemma

If L is infinite then there exists a DFA $M = (Z, \Sigma, \delta, z_1, E)$ with $L = L(M)$ and $n = |Z|$.

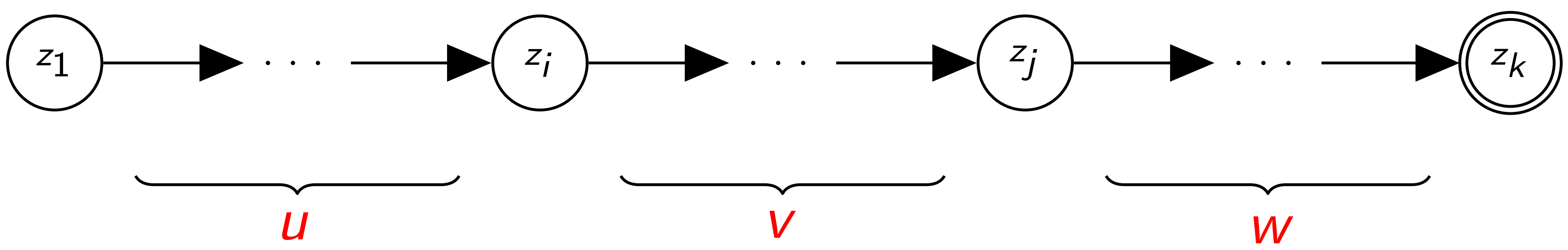
Now let x be an arbitrary string in L with $|x| \geq n$.

Let $z_1 \to \cdots \to z_k$ be the state sequence traversed when accepting x. In particular we have $k = |x| + 1$. Since M has only n states, there has to be a cycle when n characters have been read. Thus there must be $i < j$ with $i, j \in \{1, \ldots, n+1\}$ and $z_i = z_j$.



Pumping Lemma

Now let u be the substring of x that has been read when traversing $z_1 \to \cdots \to z_i$, let v be the substring of x that has been read when traversing $z_i \to \cdots \to z_j$, and let w be the substring of x that has been read when traversing $z_j \to \cdots \to z_k$. Hence we have $x = uvw$.



Pumping Lemma

With this choice of u, v, w we can now check the conditions of the pumping lemma:

Pumping Lemma

With this choice of u, v, w we can now check the conditions of the pumping lemma:

1. The length of uv is $j - 1$ and thus not larger than n

Pumping Lemma

With this choice of u, v, w we can now check the conditions of the pumping lemma:

1. The length of uv is $j - 1$ and thus not larger than n
2. The string v is not empty since $i < j$

Pumping Lemma

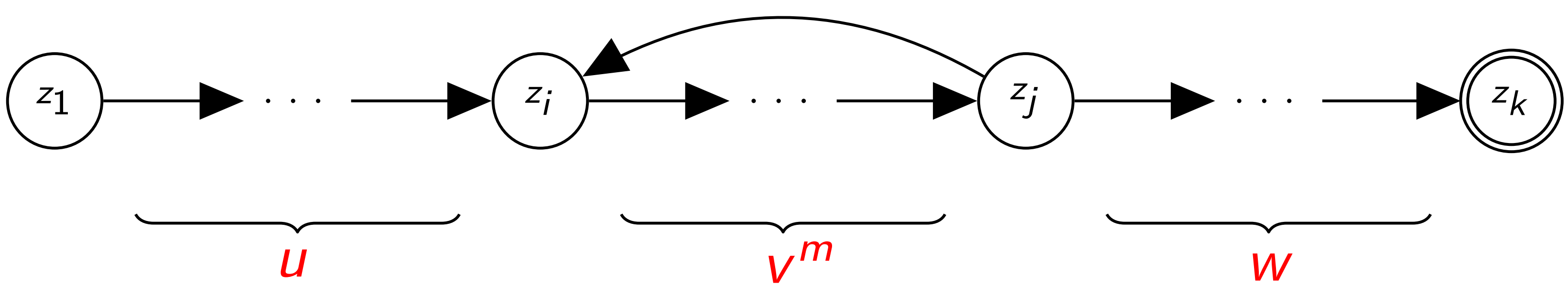
With this choice of u, v, w we can now check the conditions of the pumping lemma:

1. The length of uv is $j - 1$ and thus not larger than n
2. The string v is not empty since $i < j$
3. For an arbitrary $m \in \mathbb{N}_0$ the string $uv^m w$ can also be accepted by the automaton. To this end the automaton first reads the first $j - 1$ characters. Then it traverses m times the cycle $z_i \to \cdots \to z_j$. Finally it reads w and finishes in the accepting state z_k. Note that the cycle may also be traversed 0 times.

Pumping Lemma

With this choice of u, v, w we can now check the conditions of the pumping lemma:

1. The length of uv is $j - 1$ and thus not larger than n
2. The string v is not empty since $i < j$
3. For an arbitrary $m \in \mathbb{N}_0$ the string $uv^m w$ can also be accepted by the automaton. To this end the automaton first reads the first $j - 1$ characters. Then it traverses m times the cycle $z_i \to \cdots \to z_j$. Finally it reads w and finishes in the accepting state z_k. Note that the cycle may also be traversed 0 times.



Pumping Lemma

Remark
The pumping lemma is usually used to show that a language is *not* regular.

Negation of the Pumping Lemma

Lets have a look at the logic expression

$$
\begin{aligned}
L \in \mathcal{L}_3 \Rightarrow \exists n \in \mathbb{N} : \forall x \in L : (|x| \geq n \Rightarrow \exists u, v, w : {} & x = uvw \wedge \\
& |uv| \leq n \wedge \\
& |v| > 0 \wedge \\
& \forall m \in \mathbb{N}_0 : uv^m w \in L)
\end{aligned}
$$

Negation of the Pumping Lemma

Lets have a look at the logic expression

$$
\begin{aligned}
L \in \mathcal{L}_3 \Rightarrow \exists n \in \mathbb{N} : \forall x \in L : (|x| \geq n \Rightarrow \exists u, v, w : x = uvw \wedge \\
|uv| \leq n \wedge \\
|v| > 0 \wedge \\
\forall m \in \mathbb{N}_0 : uv^m w \in L)
\end{aligned}
$$

Recall that

$$A \Rightarrow B$$

is equivalent to

$$\neg B \Rightarrow \neg A$$

Negation of the Pumping Lemma

Thus

$$\begin{aligned}\forall n \in \mathbb{N} : \exists x \in L : \neg(|x| \geq n \Rightarrow \exists u, v, w : {} & x = uvw \wedge \\ & |uv| \leq n \wedge \\ & |v| > 0 \wedge \\ & \forall m \in \mathbb{N}_0 : uv^m w \in L) \\ & \Rightarrow L \notin \mathcal{L}_3\end{aligned}$$

Negation of the Pumping Lemma

Now recall $\neg(A \Rightarrow B) \hat{=} A \wedge \neg B$. Thus

Negation of the Pumping Lemma

Now recall $\neg(A \Rightarrow B) \hat{=} A \wedge \neg B$. Thus

$$\begin{aligned} \forall n \in \mathbb{N} : \exists x \in L : |x| \geq n \wedge \forall u, v, w : (&\neg x = uvw \vee \\ &\neg |uv| \leq n \vee \\ &\neg |v| > 0 \vee \\ &\neg \forall m \in \mathbb{N}_0 : \ uv^m w \in L) \Rightarrow L \notin \mathcal{L}_3 \end{aligned}$$

Negation of the Pumping Lemma

Now recall $\neg(A \Rightarrow B) \hat{=} A \wedge \neg B$. Thus

$$\begin{aligned}\forall n \in \mathbb{N} : \exists x \in L : |x| \geq n \wedge \forall u, v, w : (&\neg x = uvw \vee \\ &\neg |uv| \leq n \vee \\ &\neg |v| > 0 \vee \\ &\neg \forall m \in \mathbb{N}_0 : \; uv^m w \in L) \Rightarrow L \notin \mathcal{L}_3\end{aligned}$$

Then we use $\neg A \vee \neg B \vee \neg C \hat{=} \neg(A \wedge B \wedge C)$ and pull the negation into the last expression:

$$\begin{aligned}\forall n \in \mathbb{N} : \; \exists x \in L. |x| \geq n \wedge \forall u, v, w : \; &\neg(x = uvw \wedge \\ &|uv| \leq n \wedge \\ &|v| > 0 \;) \vee \\ &\exists m \in \mathbb{N}_0 : \; uv^m w \notin L \Rightarrow L \notin \mathcal{L}_3\end{aligned}$$

Negation of the Pumping Lemma

Then we use $\neg A \vee B \hat{=} A \Rightarrow B$:

Negation of the Pumping Lemma

Then we use $\neg A \vee B \hat{=} A \Rightarrow B$:

$$\begin{aligned}\forall n \in \mathbb{N} : \exists x \in L. |x| \geq n \wedge \forall u, v, w : \ & (x = uvw \wedge \\ & |uv| \leq n \wedge \\ & |v| > 0 \ ) \Rightarrow \\ & \exists m \in \mathbb{N}_0 : uv^m w \notin L \Rightarrow L \notin \mathcal{L}_3\end{aligned}$$

Negation of the Pumping Lemma

Then we use $\neg A \vee B \hat{=} A \Rightarrow B$:

$$
\begin{aligned}
\forall n \in \mathbb{N} : \exists x \in L. |x| \geq n \wedge \forall u, v, w : \quad & (x = uvw \wedge \\
& |uv| \leq n \wedge \\
& |v| > 0 \ ) \Rightarrow \\
& \exists m \in \mathbb{N}_0 : \ uv^m w \notin L \Rightarrow L \notin \mathcal{L}_3
\end{aligned}
$$

This means in order to show that a language is not regular we have to pick a *clever* string x that is larger than n and show that we can generate a string that is not part of the language by applying $uv^m w$. This has to be done for any(!) decomposition $x = uvw$, we thus have to keep it generic.

Example Pumping Lemma

Theorem

The language

$$L = \{a^k b^k \mid k \geq 1\}$$

is not regular.

Example Pumping Lemma

Proof

For any $n \in \mathbb{N}_0$ we pick the string $x = a^n b^n$. The string has length $2n$ and thus $|x| \geq n$.

Example Pumping Lemma

Proof

For any $n \in \mathbb{N}_0$ we pick the string $x = a^n b^n$. The string has length $2n$ and thus $|x| \geq n$.

We now take an arbitrary decomposition $x = uvw$ but apply the restrictions $|uv| \leq n$ and $|v| > 0$. Since $|uv| \leq n$ we know that uv has to be in the first half of the string.

Example Pumping Lemma

Proof

For any $n \in \mathbb{N}_0$ we pick the string $x = a^n b^n$. The string has length $2n$ and thus $|x| \geq n$.

We now take an arbitrary decomposition $x = uvw$ but apply the restrictions $|uv| \leq n$ and $|v| > 0$. Since $|uv| \leq n$ we know that uv has to be in the first half of the string.

The first half of the string just contains a-s and thus v contains only a-s.

Example Pumping Lemma

Proof

For any $n \in \mathbb{N}_0$ we pick the string $x = a^n b^n$. The string has length $2n$ and thus $|x| \geq n$.

We now take an arbitrary decomposition $x = uvw$ but apply the restrictions $|uv| \leq n$ and $|v| > 0$. Since $|uv| \leq n$ we know that uv has to be in the first half of the string.

The first half of the string just contains a-s and thus v contains only a-s.

Now we consider $x' = uv^0w = uw = a^{n-|v|}b^n$. Since $|v| > 0$ this string contains less a-s than b-s. By definition of the language L, the string x' thus cannot be in the language.

Example Pumping Lemma

Proof

For any $n \in \mathbb{N}_0$ we pick the string $x = a^n b^n$. The string has length $2n$ and thus $|x| \geq n$.

We now take an arbitrary decomposition $x = uvw$ but apply the restrictions $|uv| \leq n$ and $|v| > 0$. Since $|uv| \leq n$ we know that uv has to be in the first half of the string.

The first half of the string just contains a-s and thus v contains only a-s.

Now we consider $x' = uv^0w = uw = a^{n-|v|}b^n$. Since $|v| > 0$ this string contains less a-s than b-s. By definition of the language L, the string x' thus cannot be in the language.

Consequently all assumptions of the negated PL are fulfilled and in turn L is not regular.

Example Pumping Lemma

Remark

The proof can also be written in the form of an indirect proof. Assume that L is regular and there will be a contradiction (i.e. that $uv^m w$ is not part of the language)

Example of non-regular language fulfilling PL

As discussed before, fulfilling the PL is not sufficient for a language to be regular. Lets consider the language

$$\begin{aligned} L &= L_1 \cup L_2 \\ &= \{a^m b^n c^n \mid m, n \geq 1\} \cup \{b^m c^n \mid m, n \geq 0\} \end{aligned}$$

Example of non-regular language fulfilling PL

As discussed before, fulfilling the PL is not sufficient for a language to be regular. Lets consider the language

$$\begin{aligned} L &= L_1 \cup L_2 \\ &= \{a^m b^n c^n \mid m, n \geq 1\} \cup \{b^m c^n \mid m, n \geq 0\} \end{aligned}$$

This language is not regular since L_1 is not regular (because of $b^n c^n$) and since the regular languages are closed under intersection and $L \cap \overline{L_2} = L_1$. Since L_2 is regular, the regularity of L would imply that L_1 also needs to be regular, which is a contradiction. Therefore, L is not regular.

Example of non-regular language fulfilling PL

As discussed before, fulfilling the PL is not sufficient for a language to be regular. Lets consider the language

$$\begin{aligned} L &= L_1 \cup L_2 \\ &= \{a^m b^n c^n \mid m, n \geq 1\} \cup \{b^m c^n \mid m, n \geq 0\} \end{aligned}$$

This language is not regular since L_1 is not regular (because of $b^n c^n$) and since the regular languages are closed under intersection and $L \cap \overline{L_2} = L_1$. Since L_2 is regular, the regularity of L would imply that L_1 also needs to be regular, which is a contradiction. Therefore, L is not regular.

However, L does fulfill the PL. Let $z \in L$ be an arbitrary string of L. We will consider a division $z = uvw$ where $u = \varepsilon$ and v is the first character of z. The first character is either an a or a b.

Example of non-regular language fulfilling PL

In case that it is an a we check the string uv^iw and see that this new string has a different number of a-s but the suffix b^nc^n is preserved. Thus $uv^iw \in L$.

Example of non-regular language fulfilling PL

In case that it is an a we check the string uv^iw and see that this new string has a different number of a-s but the suffix b^nc^n is preserved. Thus $uv^iw \in L$.

In case the first character of z is a b we check the string uv^iw and see that this new string has a different number of b-s but the suffix c^n is preserved. Thus $uv^iw \in L$.