CSE 105
THEORY OF COMPUTATION

Fall 2021

http://cseweb.ucsd.edu/classes/fa21/cse105-a/
Today's learning goals

• Apply the Pumping Lemma in proofs of nonregularity
• Identify some nonregular sets
Pumping Lemma

If $A$ is a regular language, then there is a $p$ (number of states in DFA recognizing $A$) where, if $s$ is any string in $A$ of length at least $p$, then $s$ may be divided into three pieces, $s = x y z$ such that

- and
- for each $i \geq 0$, $x y^i z \in A$,
- $|x y| \leq p$. 

Sipser p. 78 Theorem 1.70
Which of the following sentences best describes the pumping lemma:

A) It is a property of every regular language
B) It is a property of every non-regular language
C) It is a closure property of regular languages
D) It is a closure property of non-regular languages
E) I don’t know
Answer

- The pumping lemma asserts a property satisfied by every regular language
- But we use it to prove nonregular languages
  - Assume L is regular (for contradiction)
  - Apply pumping property
  - Get a contradiction
  - Therefore L is not regular
Using the Pumping Lemma

Claim: The set $L = \{0^n1^n \mid n \geq 0\}$ is not regular.
Using the Pumping Lemma

Claim: The set $L = \{0^n1^n \mid n \geq 0\}$ is not regular.

Proof: Assume, towards a contradiction, that $L$ is regular.

Pumping Lemma gives property of all regular sets. Can we get a contradiction by assuming that the Pumping Lemma applies to this set?
Claim: The set $L = \{0^n1^n \mid n \geq 0\}$ is not regular.

Proof:

- Assume, **towards a contradiction**, that $L$ is regular.
- Therefore, the Pumping Lemma applies to $L$ and gives us some number $p$, the pumping length of $L$.
- In particular, this means that every string in $L$ that is of length $p$ or more can be "pumped".

...Idea: can we find some long string in $L$ that can't be?
Using the Pumping Lemma

Claim: The set \( L = \{0^n1^n \mid n \geq 0\} \) is not regular.

Proof: … In particular, this means that every string in \( L \) that is of length \( p \) or more can be "pumped".

Goal: pick a string \( s \) in \( L \) of length at least \( p \) that cannot be pumped, i.e., such that

- for any division of \( s \) as \( s = xyz \) with \( |y| > 0 \) and \( |xy| \leq p \)
- there is some value \( i \geq 0 \) with \( xyz \) not in \( L \)

So we have a contradiction, and \( L \) is not regular.
Claim: The set $L = \{ 0^n1^n \mid n \geq 0 \}$ is not regular.

Proof: …

Goal: pick a string $s$ in $L$ of length at least $p$ such that any division of $s$ as $s = xyz$ with $|y| > 0$ and $|xy| \leq p$ gives some value $i \geq 0$ with $xy^iz$ not in $L$.

Choose $s = 0^p1^p$. Consider any $s = xyz$ with $|y| > 0$, $|xy| \leq p$. 

Using the Pumping Lemma

Claim: The set \( L = \{0^n1^n \mid n \geq 0\} \) is not regular.

Proof: …

Goal: pick a string \( s \) in \( L \) of length at least \( p \) such that any division of \( s \) as \( s = xyz \) with \( |y| > 0 \) and \( |xy| \leq p \) gives some value \( i \geq 0 \) with \( xy^iz \) not in \( L \).

Choose \( s = 0^p1^p \). Consider any \( s = xyz \) with \( |y| > 0 \), \( |xy| \leq p \).

Since \( |xy| \leq p \), \( x = 0^m \), \( y = 0^n \), \( z = 0^r1^p \) with \( m + n + r = p \), \( n > 0 \).
Using the Pumping Lemma

Claim: The set \( L = \{0^n1^n \mid n \geq 0\} \) is not regular.

Proof: …

Goal: pick a string \( s \) in \( L \) of length at least \( p \) such that \( \text{any} \) division of \( s \) as \( s = xyz \) with \( |y| > 0 \) and \( |xy| \leq p \) gives some value \( i \geq 0 \) with \( xy^iz \) not in \( L \)

Choose \( s = 0^p1^p \). Consider any \( s = xyz \) with \( |y| > 0, |xy| \leq p \).

Since \( |xy| \leq p \), \( x = 0^m, y = 0^n, z = 0^r1^p \) with \( m+n+r = p, n > 0 \).

Picking \( i = 0 \): \( xy^iz = xz = 0^m0^r1^p = 0^{m+r}1^p \), not in \( L \)!
Using the Pumping Lemma

Claim: The set \( L = \{0^n1^n \mid n \geq 0\} \) is not regular.

Proof: …

Goal: pick a string \( s \) in \( L \) of length at least \( p \) such that any division of \( s \) as \( s = xyz \) with \( |y|>0 \) and \( |xy|\leq p \) gives some value \( i\geq0 \) with \( xy^iz \) not in \( L \)

Choose \( s = 0^p1^p \). Consider any \( s = xyz \) with \( |y|>0, |xy|\leq p \).

Since \( |xy|\leq p \), \( x=0^m, y=0^n, z=0^r1^p \) with \( m+n+r=p, j>0 \).

Picking \( i=0 \): \( xy^iz = xz = 0^m0^r1^p = 0^{m+r}1^p \), not in \( L \)!

Contradicts the Pumping Lemma! So \( L \) must not be regular.
Using the Pumping Lemma

Claim: The set \( L = \{0^n1^n \mid n \geq 0\} \) is not regular.

Proof:
Assume towards a contradiction \( L \) is regular.

So by Pumping Lemma, \( L \) has a pumping length, call it \( p \).

FACT: \( p \) is a pumping length for \( L \) (by definition).

CLAIM: \( p \) is not a pumping length for \( L \).

Conclude: contradiction!
Key ingredients in proof

**Claim:** Language L is not regular.

**Proof:** Assume, towards a contradiction, that L is regular. By the Pumping Lemma, there is a pumping length p for L.

*Consider the string s = ......*

You must pick s carefully: we want |s|≥p and s in L.

*Confirm these facts as part of your proof*

Now we will prove a contradiction with the statement "s can be pumped"

Consider an **arbitrary** choice of x,y,z such that s = xyz, |y|>0, |xy|≤p.

**This means that...** What properties are guaranteed about x,y,z?

**Consider i=...** In this case, xy^i*z = ...., which is not in L, a contradiction with the Pumping Lemma applied to L and so L is not regular.
Another example

Claim: The set \( \{a^n b^m a^n \mid m, n \geq 0\} \) is not regular.

Proof: \( \ldots \text{Consider the string } s = \ldots \).

You must pick \( s \) carefully: we want \( |s| \geq p \) and \( s \) in \( L \).

Now we will prove a contradiction with the statement "\( s \) can be pumped".

Which choices of \( s \) cannot be used to complete the proof?

A. \( s = a^p b^p \)  B. \( s = ab^p a \)  C. \( s = a^p b^p a^p \)  D. \( s = a^p b a^p \)

E. None of the above (all of these choices work).
Another example

Claim: The set \( \{a^m b^m a^n | m, n \geq 0\} \) is not regular.

Proof: … Consider the string \( s = \ldots \).

You must pick \( s \) carefully: we want \( |s| \geq p \) and \( s \) in \( L \).

Now we will prove a contradiction with the statement "\( s \) can be pumped".

Consider an arbitrary choice of \( x, y, z \) such that \( s = xyz \), \(|y| > 0\), \(|xy| \leq p\). This means that… What properties are guaranteed about \( x, y, z \)?

Consider \( i = \ldots \). In this case, \( xyz^i = \ldots \), which is not in \( L \), a contradiction with the Pumping Lemma applying to \( L \) and so \( L \) is not regular.
Claim: The set \{w \ w^R \mid w \text{ is a string over } \{0,1\} \} is not regular.

Proof: Consider the string \(s = \ldots\)

You must pick \(s\) carefully: we want \(|s| \geq p\) and \(s\) in \(L\).

Now we will prove a contradiction with the statement "\(s\) can be pumped"

Consider \(i = \ldots\)

Which \(s\) and \(i\) let us complete the proof?

A. \(s = 0^p0^p, i=2\)  
B. \(s = 0110, i=0\)  
C. \(s = 0^p110^p, i=1\)  
D. \(s = 1^p001^p, i=3\)  
E. None of them
How do we choose $i$?

Claim: The set $\{0^j1^k \mid j, k \geq 0 \text{ and } j \geq k\}$ is not regular.

Proof: …Consider the string $s = \ldots$

You must pick $s$ carefully: we want $|s| \geq p$ and $s \in L$.

Now we will prove a contradiction with the statement "$s$ can be pumped"

Consider $i = \ldots$

Which $s$ and $i$ let us complete the proof?

A. $s = 0^p1^p$, $i=2$  
B. $s = 0^p1^p$, $i=p$  
C. $s = 0^p1^p$, $i=1$
D. $s = 0^p1^p$, $i=0$  
E. I don't know
Regular sets: not the end of the story

• Many **nice / simple / important** sets are not regular

• Limitation of the finite-state automaton model
  • Can't "count"
  • Can only remember finitely far into the past
  • Can't backtrack
  • Must make decisions in "real-time"

• We know computers are more powerful than this model…

*Which conditions should we relax?*
The next model of computation

- **Idea**: allow *some* memory of unbounded size
- **How?**
  - Generalization of regular expressions: *Context-free grammars*
  - Generalization for DFA: *Pushdown Automata*
Diagonalization

- Let R be the set of syntactically valid regular expressions over binary alphabet \{0,1\}
  - R is a language over the alphabet $\Sigma = \{0,1,\varepsilon,\{,\},\cup,(,),\ast\}$
  - Encode $\Sigma$ as 3-bit “bytes”: 000, 001, 010, ..., 111
- For each regular expression $E$ in R we have
  - A string $\text{encode}(E)$ in $\{0,1\}^*$
  - A set of strings $L(E) \subset \{0,1\}^*$
- Let $D = \{ \text{encode}(E) : \text{encode}(E) \text{ is not in } L(E) \} \subset \{0,1\}^*$
- Challenge Question: is $D$ regular?
Claim: D is not regular

- \( D = \{ \text{encode}(E) : \text{encode}(E) \text{ is not in } L(E) \} \subset \{0,1\}^* \)
- Assume D is regular
- Then, \( D = L(E) \) for some Regular Expression E
- Let \( w = \text{encode}(E) \). Question: is \( w \) in \( D \)?

A) Yes
B) No
C) I don’t know
D) I entered an infinite loop and my brain exploded
D is not regular

- **D = \{ \text{encode}(E) : \text{encode}(E) \text{ is not in } L(E) \} \subset \{0,1\}^\ast**
- Assume D is regular
- Then, **D = L(E)** for some Regular Expression E
- Let **w = encode(E)**. **Question**: is **w** in D?
  - Yes? Then “encode(E) is not in L(E)” is true, ie. w is not in D
  - No? Then “encode(E) is not in L(E)” is false, ie. w is in D
- So, w is in D if and only if w is not in D
More fun with regular expressions

- Let $R$ be the set of syntactically valid regular expressions over binary alphabet $\{0,1\}$
  - $R$ is a language over the alphabet $\Sigma = \{0,1,\varepsilon,\{,\},\cup,(),^*\}$
  - Encode $\Sigma$ as 3-bit “bytes”: 000, 001, 010, …, 111

- Questions:
  - Is $R$ a regular language (over $\Sigma$)?
  - Is $\text{encode}(R)$ a regular language (over $\{0,1\}$)?

- If not, how can you formally describe the set of syntactically valid regular expressions?