CSC236 winter 2020, week 12: (non-)regularity

Recommended reading: Chapter 7 Vassos course notes, section 7.6.3—

Colin Morris
colin@cs.toronto.edu
http://www.cs.toronto.edu/~colin/236/W20/

March 30, 2020

Reminders

- A3 due Thursday @ 15:00
  - It's short!
  - Extra office hours available by request
- One last tutorial + quiz this Friday
- Also, final Q&A session Wednesday 12:00-14:00
  - These are really worth attending!
- Marking scheme changes
  - Course website will be updated when vote officially closes on Monday
- Exam-like final assessment (worth 20%) to be written April 7-9

Regular languages

A language L is **regular** iff

- L is denoted by a regular expression
- L is accepted by a deterministic FSA
- L is accepted by a non-deterministic FSA

(We now know that all of these criteria are equivalent.)

Proving regularity

A few options to prove that L is regular:

1. Construct an RE, or a DFSA, or an NFSA that matches L.
2. Use closure properties of regular languages. Show that L can be formed by application of union/intersection/complement/Kleene star to some languages that are known to be regular.
3. Use the fact that all finite languages are regular

Example: proving regularity

L_1 = strings over $\{0, 1\}$ of length 236. Prove L_1 is regular.

Example: proving regularity

$L_1 =$ strings over $\{0, 1\}$ of length 236. Prove L_1 is regular.

$L_2 =$ strings over $\{0, 1\}$ where length is a multiple of 236. Prove L_2 is regular.

Are all languages regular?

Big if true

Detour: probing the limits of FSAs

Suppose M is a DFSA such that $\mathcal{L}(M) = \{a^n \mid \exists k \in \mathbb{N}, n = 3k\}$.
What is the minimum number of states M can have?

Proving lower bounds on states

Recall, $\mathcal{L}(M) = \{a^n \mid \exists k \in \mathbb{N}, n = 3k\}$.

Consider

- $\delta^*(s, a) = q_1$
- $\delta^*(s, aa) = q_2$
- $\delta^*(s, aaa) = q_3$

Claim: q_1, q_2, and q_3 are distinct.
Proof: Show that each of the following possibilities leads to a contradiction

- $q_3 = q_1$
- $q_3 = q_2$
- $q_2 = q_1$

Pigeonhole principle

Figure: 10 pigeons $>$ 9 pigeonholes $\implies$ pigeon cohabitation

Recipe: proving lower bound on DFSA states

To prove that any DFSA M that accepts L must have at least n states

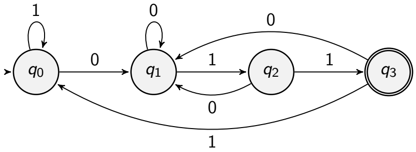
1. Prove that n is *sufficient*, by demonstrating an accepting n-state DFSA
   - (May or may not be necessary, depending on how question is worded)
2. Find n distinct prefixes $x_1, x_2, \dots x_n$, and matching suffixes[1] $y_1, y_2, \dots y_n$, such that
   - $x_j y_k \in L \iff j = k$
   - i.e. for each prefix, exactly one of the suffixes can be concatenated to it to form a string in L
3. Prove minimum of n states by contradiction
   - 3.1 Assume, for sake of contradiction, that $|Q| < n$.
   - 3.2 By the pigeonhole principle, there must be two different prefixes, x_j and x_k that go to the same state, q
   - 3.3 So $\delta^*(q, y_j)$ must be accepting and non-accepting. $\Rightarrow\Leftarrow$

[1] It's actually sufficient to find just $n - 1$ suffixes, i.e. we can get away with having *one* prefix x that doesn't have a matching suffix. See steps 3.2 and 3.3 for the reason why.

Another (worked out) lower bound example

Find the minimum number of states for a DFSA that accepts $L = \{w \in \{0,1\}^* \mid w \text{ ends with '011'}\}$.

Below, we give a 4-state DFSA for L.

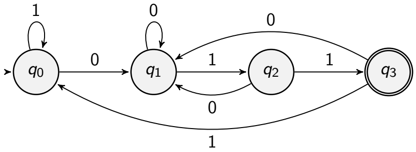


So 4 is sufficient. Is it necessary?

Another (worked out) lower bound example

Find the minimum number of states for a DFSA that accepts $L = \{w \in \{0,1\}^* \mid w \text{ ends with '011'}\}$.

Below, we give a 4-state DFSA for L.



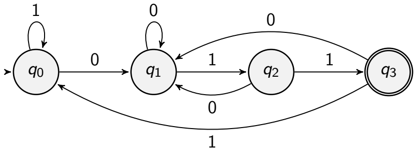
So 4 is sufficient. Is it necessary?

Consider

- $x_0 = \varepsilon$
- $x_1 = 0, y_1 = 11$
- $x_2 = 01, y_2 = 1$
- $x_3 = 011, y_3 = \varepsilon$

Another (worked out) lower bound example

Find the minimum number of states for a DFSA that accepts $L = \{w \in \{0,1\}^* \mid w \text{ ends with '011'}\}$.
Below, we give a 4-state DFSA for L.



So 4 is sufficient. Is it necessary?
Consider

- $x_0 = \varepsilon$
- $x_1 = 0, y_1 = 11$
- $x_2 = 01, y_2 = 1$
- $x_3 = 011, y_3 = \varepsilon$

By inspection, each suffix y_j has exactly one prefix x_j such that $x_j y_j \in L$.
Suppose FSOC a DFSA with < 4 states accepts L. By the pigeonhole principle, there must be a distinct pair, x_j, x_k, such that $\delta^*(s, x_j) = \delta^*(s, x_k) = q$ for some state q.
WLOG, suppose $j \neq 0$. Then $\delta^*(q, y_j)$ must be accepting. But that would mean we also accept, $x_k y_j \notin L$. $\Rightarrow\Leftarrow$

An infinite flock of pigeons

Prove that $L = \{0^n 1^n \mid n \in \mathbb{N}\}$ is non-regular.

Recipe: proving non-regularity via pigeonhole principle

Very similar to recipe for proving lower bound on number of states

To prove that L is non-regular

1. Find a **infinite** family of distinct prefixes $x_1, x_2, \ldots$, and corresponding suffixes $y_1, y_2, \ldots$, such that
   - $x_j y_k \in L \iff j = k$
   - i.e. for each prefix, exactly one of the suffixes can be concatenated to it to form a string in L
2. Prove non-existence of DFSA for L by contradiction
   2.1 Assume, for sake of contradiction, that there is a DFSA M such that $\mathcal{L}(M) = L$. Let n be its number of states.
   2.2 By the pigeonhole principle, there must be two different prefixes, x_j and x_k that go to the same state, q
   2.3 So $\delta^*(q, y_j)$ must be accepting and non-accepting. $\Rightarrow\Leftarrow$

Another approach: the Pumping Lemma

Use whichever approach you prefer. We'll ask you to prove non-regularity, but won't force you to use one approach or the other.

Let L be a regular language. Then there exists $n \in \mathbb{N}$, such that for every $x \in L$ where $|x| \geq n$, x satisfies the following property:

- $\exists y, v, w \in \Sigma^*, x = uvw \wedge v \neq \varepsilon \wedge |uv| \leq n$, and $uv^k w \in L$ for all $k \in \mathbb{N}$

Another approach: the Pumping Lemma

Use whichever approach you prefer. We'll ask you to prove non-regularity, but won't force you to use one approach or the other.

Let L be a regular language. Then there exists $n \in \mathbb{N}$, such that for every $x \in L$ where $|x| \geq n$, x satisfies the following property:

- $\exists y, v, w \in \Sigma^*, x = uvw \wedge v \neq \varepsilon \wedge |uv| \leq n$, and $uv^k w \in L$ for all $k \in \mathbb{N}$

i.e.

If L is regular, then every sufficiently long string in L contains a (non-empty) part that can be repeated ("pumped") any number of times, to keep getting more strings in L.

Pumping Lemma proof sketch

The pigeonhole principle returns

Using Pumping Lemma to prove non-regularity: example

WTS: PAL $= \{x \in \{0,1\}^* \mid x$ is a palindrome$\}$ is non-regular.
Assume, for sake of contradiction, that PAL is regular. Then the Pumping Lemma applies for some value $n \in \mathbb{N}$.