T065001: Introduction to Formal Languages

Lecture 5: Finite-state automata, regular languages, nondeterminism (4)

Chapter 1.4 in Sipser's textbook

2025-05-19

(Lecture slides by Yih-Kuen Tsay)

Some questions

Chapters 1.1 – 1.3 showed that three different ways of characterizing languages (i.e., using *deterministic finite automata*, *nondeterministic finite automata*, and *regular expressions*) have the same expressive power. Any language that can be defined in one of these three ways is “regular”.

(From Chapter 1.1)

Definition (1.16)

A language is called a *regular language* if some finite automaton recognizes it.

(From Chapter 1.2)

Corollary (1.40)

A language is regular if and only if some nondeterministic finite automaton recognizes it.

(From Chapter 1.3)

Theorem (1.54)

A language is regular if and only if some regular expression describes it.

Some questions

Chapters 1.1 – 1.3 showed that three different ways of characterizing languages (i.e., using *deterministic finite automata*, *nondeterministic finite automata*, and *regular expressions*) have the same expressive power. Any language that can be defined in one of these three ways is "regular".

Moreover, the class of regular languages is closed under various operations such as union, concatenation, star, intersection, complement, and reverse.

Some questions

Chapters 1.1 – 1.3 showed that three different ways of characterizing languages (i.e., using *deterministic finite automata*, *nondeterministic finite automata*, and *regular expressions*) have the same expressive power. Any language that can be defined in one of these three ways is “regular”.

Moreover, the class of regular languages is closed under various operations such as union, concatenation, star, intersection, complement, and reverse.

This leads to the following questions:

(1) Are all languages regular?

Some questions

Chapters 1.1 – 1.3 showed that three different ways of characterizing languages (i.e., using *deterministic finite automata*, *nondeterministic finite automata*, and *regular expressions*) have the same expressive power. Any language that can be defined in one of these three ways is "regular".

Moreover, the class of regular languages is closed under various operations such as union, concatenation, star, intersection, complement, and reverse.

This leads to the following questions:

(1) Are all languages regular?
Answer: No!

Some questions

Chapters 1.1 – 1.3 showed that three different ways of characterizing languages (i.e., using *deterministic finite automata*, *nondeterministic finite automata*, and *regular expressions*) have the same expressive power. Any language that can be defined in one of these three ways is "regular".

Moreover, the class of regular languages is closed under various operations such as union, concatenation, star, intersection, complement, and reverse.

This leads to the following questions:

(1) Are all languages regular?
Answer: No!

(2) What are some examples of nonregular languages?

Some questions

Chapters 1.1 – 1.3 showed that three different ways of characterizing languages (i.e., using *deterministic finite automata*, *nondeterministic finite automata*, and *regular expressions*) have the same expressive power. Any language that can be defined in one of these three ways is "regular".

Moreover, the class of regular languages is closed under various operations such as union, concatenation, star, intersection, complement, and reverse.

This leads to the following questions:

(1) Are all languages regular?
Answer: No!

(2) What are some examples of nonregular languages?
Answer: Coming soon...

Some questions

Chapters 1.1 – 1.3 showed that three different ways of characterizing languages (i.e., using *deterministic finite automata*, *nondeterministic finite automata*, and *regular expressions*) have the same expressive power. Any language that can be defined in one of these three ways is “regular”.

Moreover, the class of regular languages is closed under various operations such as union, concatenation, star, intersection, complement, and reverse.

This leads to the following questions:

(1) Are all languages regular?
Answer: No!

(2) What are some examples of nonregular languages?
Answer: Coming soon...

(3) How can we prove that a language is not regular?

Some questions

Chapters 1.1 – 1.3 showed that three different ways of characterizing languages (i.e., using *deterministic finite automata*, *nondeterministic finite automata*, and *regular expressions*) have the same expressive power. Any language that can be defined in one of these three ways is "regular".

Moreover, the class of regular languages is closed under various operations such as union, concatenation, star, intersection, complement, and reverse.

This leads to the following questions:

(1) Are all languages regular?
Answer: No!

(2) What are some examples of nonregular languages?
Answer: Coming soon...

(3) How can we prove that a language is not regular?
Answer: This is the main topic of today's lecture.

Nonregular Languages

- To understand the power of finite automata we must also understand their limitations.
- Consider the language $B = \{0^n 1^n \mid n \geq 0\}$.
- To recognize B, a machine will have to remember how many 0s have been read so far. This cannot be done with any finite number of states, since the number of 0s is not limited.
- $C = \{w \mid w \text{ has an equal number of 0s and 1s}\}$ is not regular, either.

Nonregular Languages

- To understand the power of finite automata we must also understand their limitations.
- Consider the language $B = \{0^n 1^n \mid n \geq 0\}$.
- To recognize B, a machine will have to remember how many 0s have been read so far. This cannot be done with any finite number of states, since the number of 0s is not limited.
- $C = \{w \mid w$ has an equal number of 0s and 1s$\}$ is not regular, either.

  Surprisingly, $D = \{w \mid w$ has an equal number of occurrences of 01 and 10 as substrings$\}$ is regular. See this week's exercises.

Nonregular Languages

- To understand the power of finite automata we must also understand their limitations.
- Consider the language $B = \{0^n1^n \mid n \geq 0\}$.
- To recognize B, a machine will have to remember how many 0s have been read so far. This cannot be done with any finite number of states, since the number of 0s is not limited.
- $C = \{w \mid w \text{ has an equal number of 0s and 1s}\}$ is not regular, either.

  Surprisingly, $D = \{w \mid w \text{ has an equal number of occurrences of 01 and 10 as substrings}\}$ is regular. See this week's exercises.
- To prove that a language is not regular, we will use a techique based on the *pumping lemma*.

The Pumping Lemma

Theorem (1.70)

If A is a regular language, then there is a number p (the pumping length) such that, if s is any string in A and $|s| \geq p$, then s may be divided as $s = xyz$ satisfying:

1. *for each $i \geq 0$, $xy^iz \in A$ (string s can be "pumped"),*
2. *$|y| > 0$, and*
3. *$|xy| \leq p$.*

The Pumping Lemma

Theorem (1.70)

If A is a regular language, then there is a number p (the pumping length) such that, if s is any string in A and $|s| \geq p$, then s may be divided as $s = xyz$ satisfying:

1. *for each $i \geq 0$, $xy^iz \in A$ (string s can be "pumped"),*
2. *$|y| > 0$, and*
3. *$|xy| \leq p$.*

Proof:

- Let $M = (Q, \Sigma, \delta, q_1, F)$ be a DFA that recognizes A.
- We assign the pumping length p to be the number of states of M.
- We show that any string s in A of length at least p may be broken into xyz satisfying the three conditions.

The Pumping Lemma (cont.)

Consider any string $s \in A$ whose length n satisfies $n \geq p$.

The Pumping Lemma (cont.)

Consider any string $s \in A$ whose length n satisfies $n \geq p$.

The sequence of states that M goes through when its input is s has length $n + 1$.

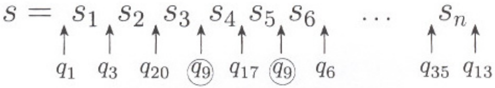

FIGURE 1.71

Example showing state q_9 repeating when M reads s

The Pumping Lemma (cont.)

Consider any string $s \in A$ whose length n satisfies $n \geq p$.

The sequence of states that M goes through when its input is s has length $n + 1$.

Since $n + 1 > p$, the sequence must contain a repeated state according to the pigeonhole principle.

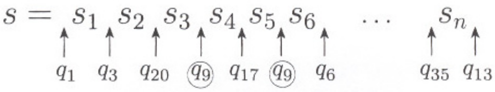

FIGURE 1.71
Example showing state q_9 repeating when M reads s

The Pumping Lemma (cont.)

Now, let y be the substring of s that is read between the first two repeated states, and let x and z be the parts before and after y, so that $\boxed{s = xyz}$.

The Pumping Lemma (cont.)

Now, let y be the substring of s that is read between the first two repeated states, and let x and z be the parts before and after y, so that $\boxed{s = xyz}$.

$\Rightarrow$ Condition 2 ($|y| > 0$) and condition 3 ($|xy| \leq p$) are satisfied.

What about condition 1 (for each $i \geq 0$, the string xy^iz is also in A)?

The Pumping Lemma (cont.)

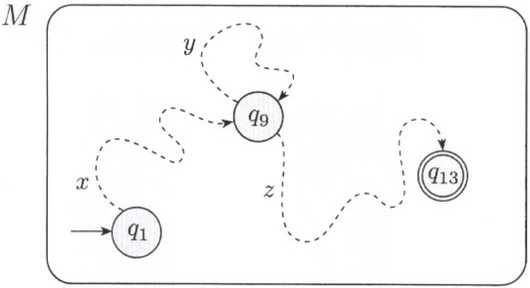

FIGURE 1.72

If we run M on input xy^iz for any integer $i \geq 0$ (e.g., $xyyz$, or $xyyyyyz$, or xz, etc.), we end up in the same state as if we run M on input xyz.

$\Rightarrow$ M will also accept xy^iz for every integer $i \geq 0$.

The Pumping Lemma (cont.)

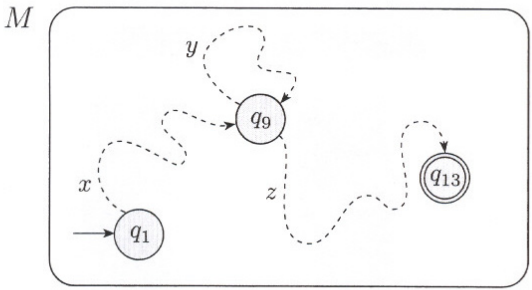

FIGURE 1.72

If we run M on input xy^iz for any integer $i \geq 0$ (e.g., $xyyz$, or $xyyyyyz$, or xz, etc.), we end up in the same state as if we run M on input xyz.

$\Rightarrow$ M will also accept xy^iz for every integer $i \geq 0$.

$\Rightarrow$ Condition 1 is satisfied. This concludes the pumping lemma's proof.

The Pumping Lemma (cont.)

Theorem (1.70)

If A is a regular language, then there is a number p (the pumping length) such that, if s is any string in A and $|s| \geq p$, then s may be divided as $s = xyz$ satisfying:

1. *for each $i \geq 0$, $xy^iz \in A$ (string s can be "pumped"),*
2. *$|y| > 0$, and*
3. *$|xy| \leq p$.*

The Pumping Lemma (cont.)

Theorem (1.70)

If A is a regular language, then there is a number p (the pumping length) such that, if s is any string in A and $|s| \geq p$, then s may be divided as $s = xyz$ satisfying:

1. *for each $i \geq 0$, $xy^iz \in A$ (string s can be "pumped"),*
2. *$|y| > 0$, and*
3. *$|xy| \leq p$.*

Example: To illustrate the pumping lemma, let $A = \{0^m 1^n \mid m, n \geq 0\}$. This A is regular because it is represented by the regular expression 0^*1^*.

The Pumping Lemma (cont.)

Theorem (1.70)

If A is a regular language, then there is a number p (the pumping length) such that, if s is any string in A and $|s| \geq p$, then s may be divided as $s = xyz$ satisfying:

1. *for each $i \geq 0$, $xy^iz \in A$ (string s can be "pumped"),*
2. *$|y| > 0$, and*
3. *$|xy| \leq p$.*

Example: To illustrate the pumping lemma, let $A = \{0^m 1^n \mid m, n \geq 0\}$. This A is regular because it is represented by the regular expression 0^*1^*.

If we define $p = 1$ and take any $s \in A$ with $|s| \geq 1$ then we can write $s = xyz$, where $x = \varepsilon$, y is the first symbol of s, and z is everything else.

For each $i \geq 0$, xy^iz still has the form 0^m1^n. $\Rightarrow$ Conditions 1–3 are OK.

The Pumping Lemma (cont.)

Theorem (1.70)

If A is a regular language, then there is a number p (the pumping length) such that, if s is any string in A and $|s| \geq p$, then s may be divided as $s = xyz$ satisfying:

1. *for each $i \geq 0$, $xy^iz \in A$ (string s can be "pumped"),*
2. *$|y| > 0$, and*
3. *$|xy| \leq p$.*

The pumping lemma is an especially useful tool for proving that certain languages are not regular.

(Technique: Proof by contradiction.)

Proving Nonregularity

Below are the steps in applying the pumping lemma to prove that a language B is not regular:

- Assume toward contradiction that B is regular.
- Then, from the pumping lemma, any string in B that is long enough (at least of the pumping length p) can be pumped.
- Find a particular long string s.
- Consider every possible division of s as xyz.
- The divisions may be grouped into a few patterns/cases; we may always require $|xy| \leq p$, according to the pumping lemma.
- Show that, in each of the division patterns, $xy^iz \notin B$ for some $i \geq 0$, a contradiction. Then, conclude that B can't be regular.

Proving Nonregularity

Below are the steps in applying the pumping lemma to prove that a language B is not regular:

- Assume toward contradiction that B is regular.
- Then, from the pumping lemma, any string in B that is long enough (at least of the pumping length p) can be pumped.
- Find a particular long string s.
- Consider every possible division of s as xyz.
- The divisions may be grouped into a few patterns/cases; we may always require $|xy| \leq p$, according to the pumping lemma.
- Show that, in each of the division patterns, $xy^iz \notin B$ for some $i \geq 0$, a contradiction. Then, conclude that B can't be regular.

Next, five examples to show how this technique works.

Example Nonregular Languages

$B = \{0^n 1^n \mid n \geq 0\}$ is not regular.

Example Nonregular Languages

$B = \{0^n 1^n \mid n \geq 0\}$ is not regular.

Proof: Suppose that B is regular. Apply the pumping lemma:

Example Nonregular Languages

$B = \{0^n 1^n \mid n \geq 0\}$ is not regular.

Proof: Suppose that B is regular. Apply the pumping lemma:

- Let s be $0^p 1^p$, where p is the pumping length (for B).

Example Nonregular Languages

$B = \{0^n 1^n \mid n \geq 0\}$ is not regular.

Proof: Suppose that B is regular. Apply the pumping lemma:

- Let s be $0^p 1^p$, where p is the pumping length (for B).
- Three cases of dividing s as xyz (where $|y| > 0$):

Example Nonregular Languages

$B = \{0^n 1^n \mid n \geq 0\}$ is not regular.

Proof: Suppose that B is regular. Apply the pumping lemma:

- Let s be $0^p 1^p$, where p is the pumping length (for B).
- Three cases of dividing s as xyz (where $|y| > 0$):

1. $\overbrace{0 \cdots \underbrace{0 \cdots 0}_{y} \cdots 0}^{p}\overbrace{1 \cdots 1}^{p}$: xy^2z will have more 0s than 1s and so is not in B.

Example Nonregular Languages

$B = \{0^n 1^n \mid n \geq 0\}$ is not regular.

Proof: Suppose that B is regular. Apply the pumping lemma:

- Let s be $0^p 1^p$, where p is the pumping length (for B).
- Three cases of dividing s as xyz (where $|y| > 0$):
    1. $\overbrace{0 \cdots \underbrace{0 \cdots 0}_{y} \cdots 0}^{p} \overbrace{1 \cdots 1}^{p}$: xy^2z will have more 0s than 1s and so is not in B.
    2. Similarly, for $0 \cdots 0 1 \cdots \underbrace{1 \cdots 1}_{y} \cdots 1$.

Example Nonregular Languages

$B = \{0^n 1^n \mid n \geq 0\}$ is not regular.

Proof: Suppose that B is regular. Apply the pumping lemma:

- Let s be $0^p 1^p$, where p is the pumping length (for B).
- Three cases of dividing s as xyz (where $|y| > 0$):

1. $\overbrace{0 \cdots \underbrace{0 \cdots 0}_{y} \cdots 0}^{p}\overbrace{1 \cdots 1}^{p}$: xy^2z will have more 0s than 1s and so is not in B.
2. Similarly, for $0 \cdots 0 1 \cdots \underbrace{1 \cdots 1}_{y} \cdots 1$.
3. $0 \cdots 0 \underbrace{0 \cdots 0 1 \cdots 1}_{y} 1 \cdots 1$: xy^2z will have some 0s after 1s and so is not in B.

Example Nonregular Languages

$B = \{0^n1^n \mid n \geq 0\}$ is not regular.

Proof: Suppose that B is regular. Apply the pumping lemma:

- Let s be 0^p1^p, where p is the pumping length (for B).
- Three cases of dividing s as xyz (where $|y| > 0$):
  1. $\overbrace{0\cdots\underbrace{0\cdots0}_{y}\cdots0}^{p}\overbrace{1\cdots1}^{p}$: xy^2z will have more 0s than 1s and so is not in B.
  2. Similarly, for $0\cdots01\cdots\underbrace{1\cdots1}_{y}\cdots1$.
  3. $0\cdots0\underbrace{0\cdots01\cdots1}_{y}1\cdots1$: xy^2z will have some 0s after 1s and so is not in B.

All possible cases result in a contradiction with condition 1.
Thus, B cannot be regular.

Example Nonregular Languages (cont.)

$C = \{w \mid w \text{ has an equal number of 0s and 1s}\}$ is not regular.

Example Nonregular Languages (cont.)

$C = \{w \mid w \text{ has an equal number of 0s and 1s}\}$ is not regular.

Proof: Suppose that C is regular. Apply the pumping lemma:

- Let s be 0^p1^p, like in the proof for B.

Example Nonregular Languages (cont.)

$C = \{w \mid w \text{ has an equal number of 0s and 1s}\}$ is not regular.

Proof: Suppose that C is regular. Apply the pumping lemma:

- Let s be 0^p1^p, like in the proof for B.
- But, how do we deal with $0\cdots0\underbrace{0\cdots01\cdots1}_{y}1\cdots1$?

Example Nonregular Languages (cont.)

$C = \{w \mid w \text{ has an equal number of 0s and 1s}\}$ is not regular.

Proof: Suppose that C is regular. Apply the pumping lemma:

- Let s be 0^p1^p, like in the proof for B.
- But, how do we deal with $0\cdots0\underbrace{0\cdots01\cdots1}_{y}1\cdots1$?
- We have $|y| > 0$ and $|xy| \leq p$ by conditions 2 and 3.

  $\overbrace{\underbrace{0\cdots\cdots0}_{xy}0\cdots0}^{p}\overbrace{1\cdots1}^{p}$: xy^2z will have more 0s than 1s and so is not in C.

In other words, by applying conditions 2 and 3. of the pumping lemma, we see that the y-part of s must consist of 0s only. Then pumping up s gives a contradiction with condition 1. Thus, C cannot be regular.

Example Nonregular Languages (cont.)

$F = \{ww \mid w \in \{0, 1\}^*\}$ is not regular.

Example Nonregular Languages (cont.)

$F = \{ww \mid w \in \{0,1\}^*\}$ is not regular.

Proof: Suppose that F is regular. Apply the pumping lemma:

- Let s be $0^p 1 0^p 1$.

Example Nonregular Languages (cont.)

$F = \{ww \mid w \in \{0,1\}^*\}$ is not regular.

Proof: Suppose that F is regular. Apply the pumping lemma:

- Let s be 0^p10^p1.
- Since $s \in F$ and $|s| \geq p$, the pumping lemma guarantees that s can be split into x, y, z while conditions 1–3 are satisfied.

  For any such x, y, z, conditions 2 and 3 give $|y| > 0$ and $|xy| \leq p$. But then:

Example Nonregular Languages (cont.)

$F = \{ww \mid w \in \{0,1\}^*\}$ is not regular.

Proof: Suppose that F is regular. Apply the pumping lemma:

- Let s be 0^p10^p1.
- Since $s \in F$ and $|s| \geq p$, the pumping lemma guarantees that s can be split into x, y, z while conditions 1–3 are satisfied.

  For any such x, y, z, conditions 2 and 3 give $|y| > 0$ and $|xy| \leq p$. But then:

  $\overbrace{\underbrace{0\cdots\cdots 0}_{xy}\,0\cdots 0}^{p}\,1\,\overbrace{0\cdots 0}^{p}\,1$: xy^2z will have more than p 0s before the first 1 and so is not in F.

  This contradicts condition 1. Thus, F cannot be regular.

Example Nonregular Languages (cont.)

$D = \{1^{n^2} \mid n \geq 0\}$ is not regular.

Example Nonregular Languages (cont.)

$D = \{1^{n^2} \mid n \geq 0\}$ is not regular.

Proof: Suppose that D is regular. Apply the pumping lemma:

- Let s be 1^{p^2}.

Example Nonregular Languages (cont.)

$D = \{1^{n^2} \mid n \geq 0\}$ is not regular.

Proof: Suppose that D is regular. Apply the pumping lemma:

- Let s be 1^{p^2}.
- $\overbrace{\underbrace{1 \cdots\cdots 1}_{xy} 1 \cdots 1}^{p^2}$: xy^2z will have $(p^2 + |y|)$ 1s.

Example Nonregular Languages (cont.)

$D = \{1^{n^2} \mid n \geq 0\}$ is not regular.

Proof: Suppose that D is regular. Apply the pumping lemma:

- Let s be 1^{p^2}.
- $\overbrace{\underbrace{1 \cdots\cdots 1}_{xy} 1 \cdots 1}^{p^2}$: xy^2z will have $(p^2 + |y|)$ 1s.
- Again, we assume $|xy| \leq p$. Together with $|y| > 0$, we have $0 < |y| \leq p$.

Example Nonregular Languages (cont.)

$D = \{1^{n^2} \mid n \geq 0\}$ is not regular.

Proof: Suppose that D is regular. Apply the pumping lemma:

- Let s be 1^{p^2}.
- $\overbrace{\underbrace{1 \cdots\cdots 1}_{xy} 1 \cdots 1}^{p^2}$: xy^2z will have $(p^2 + |y|)$ 1s.
- Again, we assume $|xy| \leq p$. Together with $|y| > 0$, we have $0 < |y| \leq p$.
- It follows that $p^2 < p^2 + |y| < p^2 + 2p + 1 = (p+1)^2$ and so xy^2z is not in D.

This contradicts condition 1. Thus, D cannot be regular.

Example Nonregular Languages (cont.)

$E = \{0^m 1^n \mid m > n\}$ is not regular.

Example Nonregular Languages (cont.)

$E = \{0^m 1^n \mid m > n\}$ is not regular.

Proof: Suppose that E is regular. Apply the pumping lemma:

- Let s be $0^{p+1}1^p$.

Example Nonregular Languages (cont.)

$E = \{0^m 1^n \mid m > n\}$ is not regular.

Proof: Suppose that E is regular. Apply the pumping lemma:

- Let s be $0^{p+1}1^p$.
- We assume $|y| > 0$ and $|xy| \leq p$: $\overbrace{\underbrace{0\cdots\cdots0}_{xy}0\cdots0}^{p+1}\overbrace{1\cdots1}^{p}$.

Example Nonregular Languages (cont.)

$E = \{0^m 1^n \mid m > n\}$ is not regular.

Proof: Suppose that E is regular. Apply the pumping lemma:

- Let s be $0^{p+1}1^p$.
- We assume $|y| > 0$ and $|xy| \le p$: $\overbrace{\underbrace{0 \cdots\cdots 0}_{xy} 0 \cdots 0}^{p+1} \overbrace{1 \cdots 1}^{p}$.
- The strings xy^2z, xy^3z, etc. all have more 0s than 1s and are actually in E!

Example Nonregular Languages (cont.)

$E = \{0^m 1^n \mid m > n\}$ is not regular.

Proof: Suppose that E is regular. Apply the pumping lemma:

- Let s be $0^{p+1}1^p$.
- We assume $|y| > 0$ and $|xy| \leq p$: $\overbrace{\underbrace{0 \cdots\cdots 0}_{xy} 0 \cdots 0}^{p+1} \overbrace{1 \cdots 1}^{p}$.
- The strings xy^2z, xy^3z, etc. all have more 0s than 1s and are actually in E!
- But, by "pumping down," we get $xy^0z = xz$ which cannot have more 0s than 1s and so is not in E.

I.e., by selecting $i = 0$ in condition 1, we obtain a contradiction. Thus, E cannot be regular.