Irregular Languages

CSCI 3130 Formal Languages and Automata Theory

Siu On CHAN

Chinese University of Hong Kong

Fall 2015
Non-regular languages

Are there irregular languages?

Candidate from last lecture:

\[ L = \{0^n10^n1 \mid n \geq 0\} \]

(duplicate of language of \(0^*1 = \{1, 01, 001, 0001, \ldots \}\))
Are there irregular languages?

Candidate from last lecture:

\[ L = \{0^n10^n \mid n \geq 0\} \]

(duplicate of language of \(0^*1 = \{1, 01, 001, 0001, \ldots\}\))

Why do we believe it is irregular?

Seems to require a “DFA” with infinitely many states

After reading the first half, need to remember number of zeros so far

11, 0101, 001001, 00010001, …

Infinitely many possibilities

Let’s formally prove this intuition
Distinct states for 01 and 0001

Claim

If a deterministic automaton accepts $L = \{0^n10^n1 \mid n \geq 0\}$, the state $q$ it reaches upon reading 01 must be different from the state $q'$ it reaches upon reading 0001.
Distinct states for 01 and 0001

Claim

If a deterministic automaton accepts $L = \{0^n10^n1 \mid n \geq 0\}$, the state $q$ it reaches upon reading 01 must be different from the state $q'$ it reaches upon reading 0001.

Why not?

Reason: after going to $q$, if it reads 01 and reaches $r$ ...

If $r$ is not accepting, it rejects 0101
If $r$ is accepting state, it accepts 000101
General case: distinguishable strings

If a deterministic automaton accepts $L$, if there are strings $x$ and $y$ such that $xz \in L$ but $yz \notin L$, then the automaton must be in two different states upon reading $x$ and $y$

Reason:

If $r$ is not accepting, it rejects $xz$  
If $r$ is accepting state, it accepts $yz$
Distinguishable strings

$x$ and $y$ are distinguishable by $L$ if for some string $z$, we have $xz \in L$ and $yz \notin L$ (or the other way round).

If $x$ and $y$ are distinguishable by $L$, any deterministic automaton accepting $L$ must reach different states upon reading $x$ and $y$. 

Diagram:

- $x$ and $y$ lead to different states $q$ and $q'$.
- $x$ and $y$ lead to the same state $q'$.
Requires many states

Strings $x_1, \ldots, x_n$ are called pairwise distinguishable by $L$ if every pair $x_i$ and $x_j$ are distinguishable by $L$, for any $i \neq j$.

If strings $x_1, \ldots, x_n$ are pairwise distinguishable by $L$, any deterministic automaton accepting $L$ must have at least $n$ states.
Pigeonhole principle

If you put 5 balls into 4 bins, then (at least) two balls end up in the same bin

More generally

If you put \( n \) balls into (at most) \( n - 1 \) bins, then (at least) two balls end in the same bin
Pigeonhole principle
Requires many states

If strings $x_1, \ldots, x_n$ are pairwise distinguishable by $L$, any deterministic automaton accepting $L$ must have at least $n$ states.

Otherwise:

If there are (at most) $n - 1$ states, by pigeonhole principle, two different strings $x_i$ and $x_j$ must end up at the same state, but:

If $x_i$ and $x_j$ are distinguishable by $L$, any deterministic automaton accepting $L$ must reach different states upon reading $x_i$ and $x_j$.
$0^n10^n1$ is not regular

Suffices find an infinitely sequence of strings that are pairwise distinguishable by $L = \{0^n10^n1 \mid n \geq 0\}$

After reading the first half, need to remember number of zeros so far
11, 0101, 001001, 00010001, ...

1, 01, 001, 0001, ... are pairwise distinguishable by $L$

Why are $0^i1$ and $0^j1$ distinguishable by $L$? ($i \neq j$)
$0^n10^n1$ is not regular

Suffices find an infinitely sequence of strings that are pairwise distinguishable by $L = \{0^n10^n1 \mid n \geq 0\}$

After reading the first half, need to remember number of zeros so far

$11, 0101, 001001, 00010001, \ldots$

$1, 01, 001, 0001, \ldots$ are pairwise distinguishable by $L$

Why are $0^i1$ and $0^j1$ distinguishable by $L$?  ($i \neq j$)

Take $z = 0^i1$

$0^i10^i1 \in L \quad 0^j10^i1 \notin L$
Which of these are (ir)regular?

\[ L_1 = \{ x \mid x \text{ has the same number of 0s and 1s} \} \]
\[ L_2 = \{ 0^n1^m \mid n > m \geq 0 \} \]
\[ L_3 = \{ x \mid x \text{ has the same number of patterns 01 and 11} \} \]
\[ L_4 = \{ x \mid x \text{ has the same number of patterns 01 and 10} \} \]
\[ L_5 = \{ x \mid x \text{ has a different number of 0s and 1s} \} \]
$L_1 = \text{Same number of 0s and 1s}$

Why does it require infinitely many states to accept?
\[ L_1 = \text{Same number of 0s and 1s} \]

Why does it require infinitely many states to accept?

Need to remember number of 0s (or 1s) read so far

\[ \varepsilon, 0, 00, 000, \ldots \] are pairwise distinguishable by \( L_1 \)

Why are \( 0^i \) and \( 0^j \) distinguishable by \( L_1 \)? \((i \neq j)\)
$L_1 = \text{Same number of 0s and 1s}$

Why does it require infinitely many states to accept?

Need to remember number of 0s (or 1s) read so far

$\varepsilon, 0, 00, 000, \ldots$ are pairwise distinguishable by $L_1$

Why are $0^i$ and $0^j$ distinguishable by $L_1$? ($i \neq j$)

Take $z = 1^i$

\[ 0^i1^i \in L_1 \quad 0^j1^i \notin L_1 \]
\( L_2 = \{0^n1^m \mid n > m \} \)

Like \( L_1 \), need to remember number of 0s read so far

\( \varepsilon, 0, 00, 000, \ldots \) are pairwise distinguishable by \( L_2 \)

Why are \( 0^i \) and \( 0^j \) distinguishable by \( L_2 \)? \( (i > j) \)
\[ L_2 = \{0^n 1^m \mid n > m\} \]

Like \( L_1 \), need to remember number of 0s read so far

\( \varepsilon, 0, 00, 000, \ldots \) are pairwise distinguishable by \( L_2 \)

Why are \( 0^i \) and \( 0^j \) distinguishable by \( L_2 \)? \((i > j)\)

Take \( z = 1^{i-1} \)

\[
0^i 1^{i-1} \in L_2 \quad 0^j 1^{i-1} \notin L_2
\]
\( L_3 = \text{same number of 01s and 11s} \)

Need to remember the number of 01s read so far

\[ \varepsilon, 01, 0101, 010101, \ldots \text{ are pairwise distinguishable by } L_3 \]

Why are \((01)^i\) and \((01)^j\) distinguishable by \(L_3\)? \((i > j)\)
$L_3 = \text{same number of 01s and 11s}$

Need to remember the number of 01s read so far

$\varepsilon, 01, 0101, 010101, \ldots$ are pairwise distinguishable by $L_3$

Why are $(01)^i$ and $(01)^j$ distinguishable by $L_3$? $(i > j)$

Take $z = 1^i$

$(01)^i 1^i \in L_3 \quad (01)^j 1^i \notin L_3$

Example: $010101111 \quad (i = 3)$
$L_4$ = same number of 01s and 10s

$\varepsilon, 01, 0101, 010101, \ldots$ are pairwise distinguishable by $L_4$

Why are $(01)^i$ and $(01)^j$ distinguishable by $L_4$? ($i > j$)

Take $z = (10)^i$

$(01)^i(10)^i \in L_4 \quad (10)^j(10)^i \notin L_4$

Example: 010101101010  ($i = 3$)
$L_4 = \text{same number of 01s and 10s}$

$\varepsilon, 01, 0101, 010101, \ldots$ are pairwise distinguishable by $L_4$

Why are $(01)^i$ and $(01)^j$ distinguishable by $L_4$? ($i > j$)

Take $z = (10)^i$

$(01)^i(10)^i \in L_4$ $(10)^j(10)^i \not\in L_4$

Example: $010101101010$ ($i = 3$)

In fact, $(01)^j(10)^i \in L_4$ because there are as many 01 as 10

In fact, $L_4$ is regular (see Week 2 tutorial)
\( L_5 = \text{different number of 0s and 1s} \)

Is \( L_5 \) irregular?
$L_5 = \text{different number of 0s and 1s}$

Is $L_5$ irregular?

Yes

If $L_5$ were regular, then so is

$$
\overline{L_5} = L_1 = \{x \mid x \text{ has the same number of 0s and 1s}\}
$$

But we saw that $L_1$ is irregular, therefore so is $L_5$
An exercise

\[ L_6 = \text{properly nested strings of parentheses} \quad \Sigma = \{ (, ) \} \]

\( (, ) \), \( (()) \), \( ()() \) are in \( L_6 \)
\( (, ) \), \( )( \) are not

Exercise: show that \( L_6 \) is irregular
What does it mean?
An exercise

\[ L_6 = \text{properly nested strings of parentheses} \quad \Sigma = \{ (, ) \} \]

\( (), (()), ()( ) \) are in \( L_6 \)

\( (, ), )() \) are not

Exercise: show that \( L_6 \) is irregular

What does it mean?

Language = computational problem
DFA = machine with finite memory

\( L_6 \) is irregular \( \Rightarrow \) checking whether (arbitrarily long) strings are properly nested requires unbounded amount of memory