Testing Regularity of Languages

Prof. (Dr.) K.R. Chowdhary

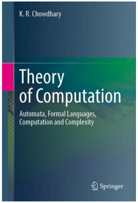
Former Professor & Head, Department of Computer Sc. & Engineering
MBM Engineering College, Jodhpur

SPRINGER NATURE

Springer

K.R. Chowdhary

Theory of Computation



Automata, Formal Languages, Computation and Complexity

Focuses on pedagogy in its writing, that represents a refreshing approach

Ensures comprehensive and enjoyable learning

Undergone a rigorous classroom testing

Get 20% off with this code: SPRAUT

Available on Springer Nature Link

link.springer.com/book/9789819762347

Testing regularity - Intro

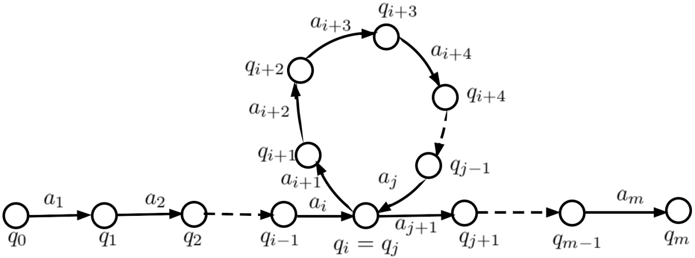
Consider language $L = \{a^n b^n | n \geq 0\}$. While reading from tape the FA has to remember arbitrarily large number of a's to compare later with number of b's. Since, there is no arbitrary size storage in FA, no FA can recognize this language, hence L is not regular.

Other proof: Since a string in L can be arbitrarily large and states are finite, some state will be revisited (say $q_i = q_j, i \neq j$) in the process of recognition. Hence, for some $m \neq n$, there may be $\delta^*(q_0, a^m) = q_i$ and $\delta^*(q_0, a^n) = q_i$.

$$
\begin{aligned}
\delta^*(q_0, a^m a^n) &= \delta^*(\delta^*(q_0, a^m), b^n) \\
&= \delta^*(q_i, b^n) \\
&= q_f.
\end{aligned}
$$

Let $M = (Q, \Sigma, \delta, s, F)$, $|Q| = n, s = q_0, q_m \in F,\ m \geq n$, and $w = a_1 a_2 \dots a_m$. Since $|w| > |Q|$, some states are repeated due to pigeonhole principle. Say, one state revisited is $q_i = q_j$ for $0 \leq i < j \leq m$. Thus, the state sequence visited during the recognition is:

$q_0 \dots q_{i-1} q_i, q_{i+1} \dots q_{j-1} q_j, q_{j+1} \dots q_m$.



The string w is recognized through the path FA as follows:

$$\begin{aligned}
\delta^*(q_0, a_1 a_2 \dots a_m) &= \delta^*(\delta^*(q_0, a_1 a_2 \dots a_i), a_{j+1} a_{j+2} \dots a_m) \\
&= \delta^*(q_i, a_{j+1} a_{j+2} \dots a_m) \\
&= \delta^*(q_j, a_{j+1} a_{j+2} \dots a_m) = q_m \in F.
\end{aligned}$$

Kleene star properties of Regular languages...

Therefore $a_1 a_2 \ldots a_i a_{i+1} \ldots a_j a_{j+1} \ldots a_m \in L(M)$. Also, $a_1 a_2 \ldots a_i a_{j+1} \ldots a_m \in L(M)$. Since, $q_i = q_j$, the substring $a_{i+1} \ldots a_{j-1}$ can be repeated an arbitrary times (pumped), and still the string w will be recognized, i.e.,

$$a_1 a_2 \ldots a_i (a_{i+1} \ldots a_j)^k a_{j+1} \ldots a_m \in L(M), \text{ for } k \geq 0$$

The above is specified in the form of a lemma, given below.

Lemma

(Pumping Lemma.) Given a FA M, $|Q| = n, w \in L(M), |w| \geq n$, there exists a decomposition of w as xyz, such that $|xy| \leq n, |y| \geq 1, k \geq 0$, so that there is always $xy^k z \in L(M)$.

Proof.

The proof has been discussed above using the diagram. If a language string w fails to satisfy the criteria $xy^k z \in L(M)$, then it is not regular. Note that pumping lemma apply to only infinite language, and it is for negative, i.e., used to prove the non-regularity of a language, for that some how we should have strategy to show that $xy^k z \notin L(M)$. □

Testing non-regularity

Example

Show that $L = \{a^n \mid n \text{ is prime}\}$ is non-regular.

Solution

Solution: let $w = xy^k z$, $k \geq 0$, $x = a^p, y = a^q, z = a^r, |q| \geq 1$. *Therefore* $w = a^p(a^q)^k a^r = a^{p+kq+r}$. *Thus, we need to show that* $p + kq + r$ *is not prime. Let us assume that* $k = p + 2q + r + 2$, *we have;*

$$\begin{aligned} p + kq + r &= p + (p + 2q + r + 2)q + r \\ &= p + pq + 2q^2 + rq + 2q + r \\ &= 1(p + 2q + r) + q(p + 2q + r) \\ &= (p + 2q + r)(1 + q) \end{aligned}$$

Since the string $w = a^n$ *can be factorized in pumping lemma, the language is not regular.*

Myhill-Nerode(MN) Theorem

The pumping lemma holds for some non-regular languages only, and does not provide sufficient condition to prove that a language is regular. If pumping lemma fails to prove non-regularity, it does not imply otherwise.

Theorem

(MN.) For $x, y, z \in \Sigma^*$, a "distinguishing extension" z is such that $xz \in F$ but $yz \notin F$. Therefore $x \sim y$ iff there is no distinguishing extension z. The $\sim$ is equivalence relation which divides all $w \in \Sigma^*$ into equivalence classes.

If $x \sim y$, and there is $xz \sim yz$, and $x, y, z \in \Sigma^*$, then equivalence relation is called right invariant. The $x \sim_L y$ is equivalence relation for language L if $xz \in L \Leftrightarrow yz \in L$.

Definition

Index of a equivalence class is total number of equivalence classes in the language. $x \sim_M y$ is equivalence relation for *DFA M* if same state is reachable for inputs $x, y \in \Sigma^*$.

Myhill-Nerode(MN) Theorem

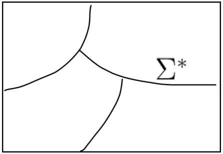
Definition

(ver.2 MN theorem.) If $\exists w \in \Sigma^*$ for states p, q such that $\delta^*(p, w) \in F \wedge \delta^*(q, w) \notin F$), then w is distinguishing string for p, q. If there does not exists any distinguishing string for p, q then they are not equivalent.

Theorem

MN theorem states that L is regular iff $\sim_L$ has finite index, and number of states in the smallest DFA recognizing L is equal to index of the equivalence class in $\sim_L$.

Intuition of above is: if such a minimal automaton is obtained, then any two string x, y driving the automaton into the same state, will be in the same equivalence class. I.e., the equivalence relation $\sim_L$ creates partition set on the strings Σ^*, and size of partition set is number of states in the FA.



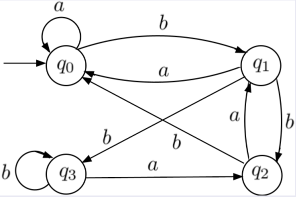
MN Theorem: Example

Example

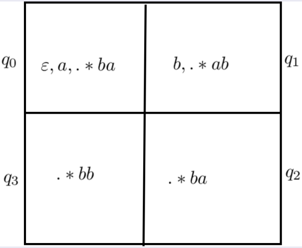
Consider a language on $\Sigma = \{a, b\}$, such that last but one character in w is b.

Solution

The FA and equivalence classes are shown below.



In the diagram below, the substrings in "$\varepsilon, a, .*ba$": before dot sign (ε, a) correspond to equivalent strings x, y in equivalence relation $x \sim y$. The part after dot, i.e. $*ba$ is distinguishing extension z, such that $xz \sim yz$. Patterns in other three equivalence classes are on the same lines.



MN Theorem: Examples

Example

Show that the language on Ł $= \{a^n b^n | n \geq 0\}$ is non-regular.

Solution

Let $S = \{\varepsilon, a, aa, aaa, aaaa, \ldots\}$ is infinite over $\{a, b\}$. Let a^k and a^m are pair-wise distinguishable for $k \neq m$.
Consider distinguishing extension $z = b^m$. Appending z with pair-wise distinguishing strings, we have $a^k b^m \notin L$ and $a^m b^m \in L$. Therefore a^k, a^m are distinguishable w.r.t. L. Since k and m are taken arbitrary numbers, there are arbitrarily large number of pair-wise distinguishing strings. This corresponds to infinite states, hence the language is not regular.

Bibliography

Chowdhary, K.R. (2025). Regular Languages. In: Theory of Computation. Springer, Singapore.
https://doi.org/10.1007/978-981-97-6234-7_5