Formal Language and Automata Theory (CS21004)

Soumyajit Dey
CSE, IIT Kharagpur

Announcements

- The slide is just a short summary
- Follow the discussion and the boardwork
- Solve problems (apart from those we dish out in class)

Table of Contents

Languages that are not regular

$L = \{a^n b^n \mid n \geq 0\} = \{\epsilon, ab, aabb, aaabbb, \cdots\}$

- needs to remember number of a-sand match with b-s. Infinite number of possibilities
- cannot remember with finite number of states
- We further provide a formal arguement

Languages that are not regular

Let a DFA with k states accept L. consider some $n >> k$

- starting from initial state i, $a^n b^n$ leads to the accept state f
- some state must have been visited more than once, let it be p

Let $a^n b^n = uvw$, $j = |v| > 0$ where

- $\hat{\delta}(i, u) = p, \hat{\delta}(p, v) = p, \hat{\delta}(p, w) = f$
- Hence $\hat{\delta}(i, uw) = f$
- $uw = a^{n-j} b^n \notin L$
- Similarly, $\hat{\delta}(i, uv^3 w) = f$ but $uv^3 w = a^{n+2j} b^n \notin L$

♠ Such a DFA does not exist

Pumping Lemma for Regular Languages

Let L be a regular language. Then there exists an integer $p \geq 1$ such that every string w in L of length at least p (p is called the "pumping length") can be written as $w = xyz$ (i.e., w can be divided into three substrings), satisfying the following conditions:

- $|y| \geq 1$
- $|xy| \leq p$
- $\forall i \geq 0,\ xy^i z \in L$

Pumping Lemma for Regular Languages : general version

Let L be a regular language. Then there exists an integer $p \geq 1$ such that every string uwv in L with $|w| \geq p$ can be written as $uwv = uxyzv$ such that

- $|y| \geq 1$
- $|xy| \leq p$
- $\forall i \geq 0,\ uxy^i zv \in L$

standard version is a special case with u, v being empty. Since the general version imposes stricter requirements on the language, it can be used to prove the non-regularity of many more languages, such as $\{a^m b^n c^n \ : \ m \geq 1,\ n \geq 1\}$

Pumping Lemma for Regular Languages

- Necessary but not sufficient condition
- Cannot be used to prove language as regular
- There are non-regular languages which satisfy the lemma
- Violation can be used to prove language as non-regular

Pumping Lemma Ex

$\{a^{2^n} \mid n \geq 0\}$.

- Let this be accepted by a k state DFA. Choose n such that $n >> k$
- Thus $2^n > k$. Hence we may decompose the string a^{2^n} to parts of length i, j, l such that $2^n = i + j + l$ and the intermediate j symbols form a cycle in the DFA
- The DFA will accept a^{2^n+j}
- Note, $i + j \leq k < n \Rightarrow j < n$
- $2^n + j < 2^n + n < 2^n + 2^n = 2^{n+1}$

♠ such a DFA cannot exist

Pumping Lemma Ex

We can show using Pumping Lemma

- $\{a^n b^m \mid n \geq m\}$ is not regular
- $\{a^{n!} \mid n \geq 0\}$ is not regular

More examples of languages not regular

Alternate lines of argument :

- $A = \{w \mid n_a(w) = n_b(w)\} \Rightarrow$ if A is regular than $A \cap L(a^*b^*) = \{a^n b^n \mid n \geq 0\}$ is regular
- $\{a^n b^m \mid n \geq m\}$ is regular
  $\Rightarrow A^R = \{b^m a^n \mid n \geq m\}$ is regular
  $\Rightarrow C = A^R[a \mapsto b, b \mapsto a] = \{a^m b^n \mid n \geq m\}$ is regular
  $\Rightarrow A \cap C = \{a^n b^n \mid n \geq 0\}$ is regular

More examples of languages not regular

- $U \subseteq \mathbb{N}$ is an **ultimately periodic** set, i.e. $\exists n \geq 0$, $\exists p > 0$, $\forall m \geq n$, $m \in U$ iff $m + p \in U$. We call p is the period of U. Every such U is regular
- Ex. $\{0, 3, 7, 9, 19, 20, 23, 26, 29, 32, 35, \cdots\}$: $(n = 20, p = 3), (n = 21, p = 6)$: n, p need not be unique
- Let $A \subseteq \{a\}^*$. A is regular iff $\{m \mid a^m \in A\}$ is **ultimately periodic**

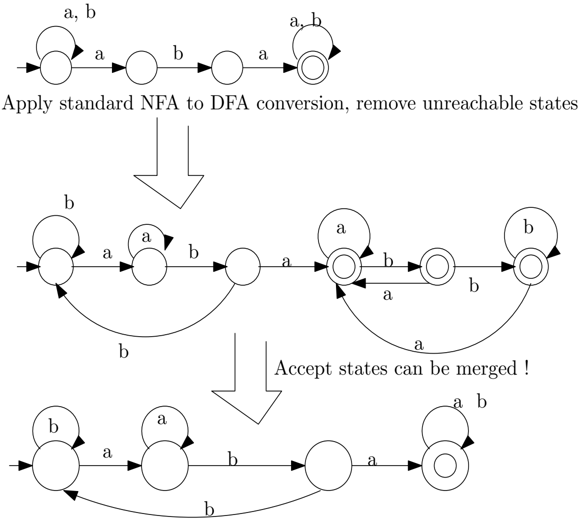
Table of Contents

Equivalence of FAs

When we convert NFA to DFA,

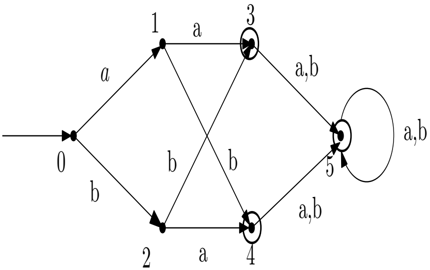
- We ignore unreachable states, keep them you simply have a larger DFA for the same language !!
- Even some reachable states can be merged preserving language equivalence !!

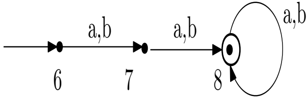
Example



- Not all such cases are as obvious

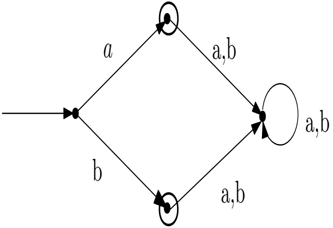
Example

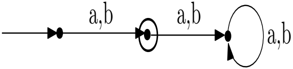
- 



Example

-





How to decide which states to collapse

- Intuitively two states are *mergeable* if they behave similarly (in terms of language acceptance) for the same input string
- Starting from respective states, with the same input string, either both lead to respective final states or none lead to respective final states
- Turns out to be a necessary and sufficient condition
- Such relations among state pairs are *equivalence relations*

Equivalence relation on states

- $\forall p, q \in Q,\ p \approx q$ iff

$$\forall x \in \Sigma^* \, [\hat{\delta}(p, x) \in F \Leftrightarrow \hat{\delta}(q, x) \in F]$$

- reflexive, symmetric, transitive
- for any state p, $[p] = \{q \mid p \approx q\}$
- by definition, equivalence classes are mutually exclusive and exhaustive : every state is in exactly one class/partition,
- $p \approx q \Leftrightarrow [p] = [q]$

Quotient Automaton $M/\approx$ for DFA M

Given $M = (Q, \Sigma, \delta, s, F)$, $M/\approx \stackrel{def}{=} (Q', \Sigma, \delta', s', F')$

- $Q' = \{[p] \mid p \in Q\}$
- $\delta'([p], a) = [\delta(p, a)]$
- $s' = [s]$
- $F' = \{[p] \mid p \in F\}$

Is δ' well defined ??

If $p, q, \in [p]$, is $\delta'([p], a) = [\delta(p, a)] = \delta'([q], a) = [\delta(q, a)]$?
For any $a \in \Sigma$, $y \in \Sigma^*$

$$
\begin{aligned}
\hat{\delta}(\delta(p, a), y) \in F &\Leftrightarrow \hat{\delta}(p, ay) \in F && \text{by definition of } \hat{\delta} \\
&\Leftrightarrow \hat{\delta}(q, ay) \in F && \text{since } p \approx q \\
&\Leftrightarrow \hat{\delta}(\delta(q, a), y) \in F && \text{by definition of } \hat{\delta}
\end{aligned}
$$

Hence, $\delta(p, a) \approx \delta(q, a)$ by definition of $\approx$. So, $[\delta(p, a)] = [\delta(q, a)]$

$p \in F \Leftrightarrow [p] \in F'$

$p \in F \Rightarrow [p] \in F'$ by definition of F'. What about the other direction, i.e. $[p] \in F' \Rightarrow p \in F$??

- What is there to prove ??
- Note that you have $[p]$, that does not specify any p but the overall equivalence class.
- Need to show that all elements of the class are in F rather than one specific member.
- Prove that any such equivalence class is either subset of F or disjoint.

$[p] \in F' \Rightarrow p \in F$

$$
\begin{aligned}
[p] \in F' &\Rightarrow \exists x \in [p], x \in F \\
&\Rightarrow \hat{\delta}(x, \epsilon) = x \in F && \text{by defn. of } \hat{\delta} \\
&\Rightarrow \forall q \approx x, \hat{\delta}(q, \epsilon) \in F && \text{by defn. of } \approx \\
&\Rightarrow \forall q \approx x, \hat{\delta}(q, \epsilon) = q \in F && \text{by defn. of } \hat{\delta} \\
&\Rightarrow \forall q \in [p], q \in F && \forall q \approx x \in [p], q \in [p]
\end{aligned}
$$

Prove the following

- $\forall x \in \Sigma^*, \hat{\delta}'([p], x) = [\hat{\delta}(p, x)]$
- $L(M/\approx) = L(M)$
- $M/\approx$ cannot be collapsed any further

Here is an algorithm for computing the collapsing relation $\approx$ for a given DFA M with no inaccessible states. The algorithm will mark (unordered) pair of states $\{p, q\}$. A pair $\{p, q\}$ will be marked as soon as a reason is discovered why p and q are *not* equivalent.

1. Write down a table of all pairs $\{p, q\}$, initially unmarked.
2. Mark $\{p, q\}$ if $p \in F$ and $q \notin F$ of vice versa.
3. Repeat the following until no more changes occur: If there exists an unmarked pair $\{p, q\}$ such that $\{\delta(p, a), \delta(q, a)\}$ is marked for some $a \in \Sigma$, then mark $\{p, q\}$.
4. When done, $p \approx q$ iff $\{p, q\}$ is not marked.

Table of Contents

DFA : isomorphism

Two deterministic finite automata $M = (Q_M, \Sigma, \delta_M, s_M, F_M)$, $N = (Q_N, \Sigma, \delta_N, s_N, F_N)$, are isomorphic iff $\exists f$, $f : Q_M \rightarrow Q_N$ such that

- $f(s_M) = s_N$
- $\forall p \in Q_M, a \in \Sigma,\ f(\delta_M(p, a)) = \delta_N(f(p), a)$
- $p \in F_M$ iff $f(p) \in F_N$

One is just the renamed version of another. Note, $M/\equiv$, $N/\equiv$ are also isomorphic. $\Rightarrow$ We should be able to define a minimal automata directly from the language itself. All other possible minimal automata will be isomorphic with this.

Myhill-Nerode Relations

Let $R \subseteq \Sigma^*$ be regular with DFA $M = (Q, \Sigma, \delta, s, F)$ for R. M does not have any unreachable states. A relation $\equiv_M$ on Σ^* defined as

- $x \equiv_M y \Leftrightarrow \hat{\delta}(s, x) = \hat{\delta}(s, y)$

$\equiv_M$ is an equivalence relation. Other properties of $\equiv_M$

1. $\forall x, y \in \Sigma^*, a \in \Sigma,\ x \equiv y \Rightarrow xa \equiv ya$: right congruence (show this)
2. $\equiv_M$ refines R : $x \equiv_M y \Rightarrow (x \in R \Leftrightarrow y \in R)$ – every $\equiv_M$-class has either all its elements in R or none of its elements in R, i.e. R is a union of $\equiv_M$-classes
3. The no. of $\equiv_M$ classes is finite ($=$ no. of states in M ?)

Myhill-Nerode Relations

Any equivalence relation on Σ^* which is a right congruence of finite index refining a regular set R is called a Myhill-Nerode Relation

- Just like $M \rightarrow \equiv_M$ we can $\equiv \rightarrow M_\equiv$

Let $\equiv$ be an arbitrary Myhill-Nerode Relation on Σ^* for some $R \subseteq \Sigma^*$, i.e. $\equiv$ is some equivalence Relation on Σ^* which is also right congruence of finite index refining a regular set R

$\equiv \rightarrow M_{\equiv}$

DFA $M_{\equiv} = (Q, \Sigma, \delta, s, F)$

- $Q = \{[x] \mid x \in \Sigma^*\}$ (is finite, why ?)
- $s = [\epsilon]$
- $F = \{[x] \mid x \in R\}$
- $\delta([x], a) = [xa]$ ($y \in [x] \Rightarrow [xa] = [ya]$ by right congruence)

Can show

- $x \in R \Leftrightarrow [x] \in F$: The '$\Rightarrow$' is by defn of F, for $\Leftarrow$, $y \in [x] \in F \Rightarrow x \in F \Rightarrow y \in R$ by refinement
- $\hat{\delta}([x], y) = [xy]$: by induction
- $L(M_{\equiv}) = R$
- $\equiv_{M_{\equiv}}$ is identical to $\equiv$
- $M_{\equiv_M}$ is isomorphic to M

Myhill-Nerode Theorem

Let $R \subseteq \Sigma^*$. The following statements are equivalent:

- R is regular
- there exists a Myhill-Nerode relation for R
- the relation $\equiv_R$ creates a finite partitioning of Σ^*

Example application

Consider $R = \{a^n b^n | n \geq 0\} \subseteq \Sigma^*$. Let R be regular with a Myhill-Nerode relation $\equiv$ on Σ^* for R

- Let $a^m \equiv a^k$ for any $m \neq k$
- By right congruence $a^m b^k \equiv a^k b^k$
- Note $x \equiv y \Rightarrow [x \in R \Leftrightarrow y \in R]$, i.e. an equivalence partition is either inside R or outside R, but cannot span across
- But now we have one equivalence partition containing $a^m b^k, a^k b^k$ where $a^m b^k \notin R$, $a^k b^k \in R$.
- Hence, it is not the case that $a^k \equiv a^m$
- The relation $\equiv$ creates an infinite partitioning of Σ^*

♠ R is not regular