Properties of Regular Languages

There are severe limits to the recognition problems that an FA can handle. But this requires proof.

Closure Properties

A set is ***closed*** under an operation if applying that operation to any members of the set always yields a member of the set.

Fact. *The set of regular languages is closed under each Kleene operation.*

That is, if L_1 and L_2 are regular languages, then each of $L_1 \cup L_2$, $L_1 L_2$ and L_1^* is regular.

Proving Closure under Kleene

The easiest approach is to show that the REs for L_1 and L_2 can be combined or adjusted to form the RE for the combination language.

Example: The RE for $L_1 L_2$ is obtained by writing down the RE for L_1 followed by the RE for L_2.

Closure under Complementation

Fact. *The set of regular languages is closed under complementation.*

The complement of language L, written $\overline{L}$, is all strings not in L but with the same alphabet.

The statement says that if L is a regular language, then so is $\overline{L}$.

To see this fact, take ***deterministic*** FA for L and interchange the accept and reject states.

Closure under Intersection

Fact. *The set of regular languages is closed under intersection.*

One approach: Use de Morgan's law:

$$L_1 \cap L_2 = \overline{(\overline{L_1} \cup \overline{L_2})}$$

and that regular languages are closed under union and complementation.

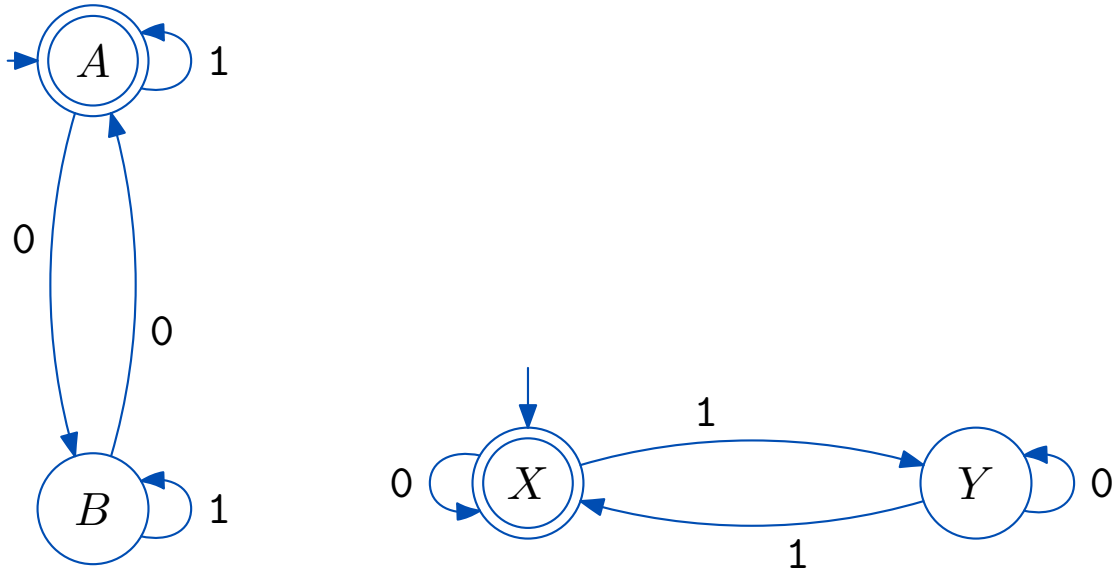
Product Construction for Intersection

Each state in the product is pair of states from the original machines.

Formally, if L_1 is accepted by DFA M_1 with 5-tuple $(Q_1, \Sigma, q_1, T_1, \delta_1)$ and L_2 is accepted by DFA M_2 with 5-tuple $(Q_2, \Sigma, q_2, T_2, \delta_2)$. Then $L_1 \cap L_2$ is accepted by the DFA $(Q_1 \times Q_2, \Sigma, (q_1, q_2), T_1 \times T_2, \delta)$ where $\delta((r, s), \mathsf{x}) = (\delta_1(r, \mathsf{x}), \delta_2(s, \mathsf{x}))$.

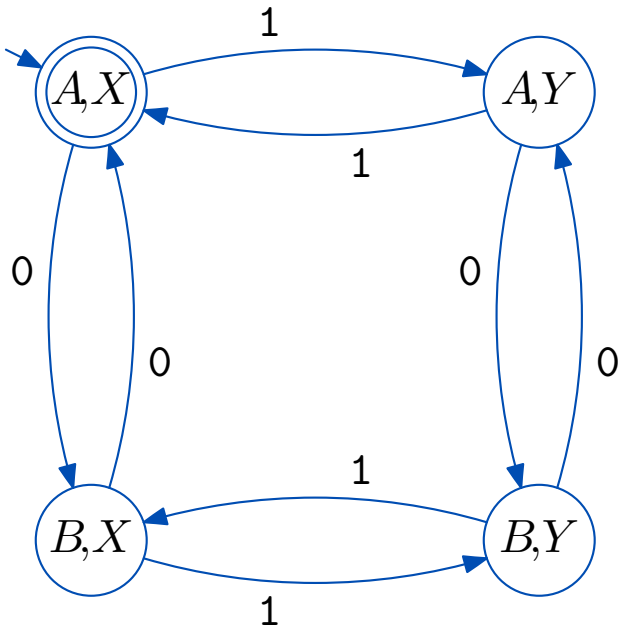
Example: Even 0's and 1's

Suppose L_1 is the binary strings with an even number of 0's, and L_2 the binary strings with an even number of 1's. Then the FAs for these languages both have two states:



And so the FA for $L_1 \cap L_2$ has four states:

Product Construction for Even 0's and 1's



A Nonregular Language: $\mathtt{0}^n\mathtt{1}^n$

There is no FA for

$$B = \{\mathtt{0}^n\mathtt{1}^n : n \geq 0\} = \{\varepsilon, \mathtt{01}, \mathtt{0011}, \mathtt{000111}, \ldots\}$$

Informal argument: Suppose input known to be 0's followed by 1's. The FA has to **count** the 0's: that is, at end of 0's it must be in state unique to number of 0's read. But FA has only fixed finite memory, while there can be arbitrarily many 0's. Impossible.

(In)Distinguishable Strings

Two strings x and y are ***indistinguishable with respect to language*** L if for every string z, it holds that $xz \in L$ if and only if $yz \in L$. Otherwise they are ***distinguishable.***

Example: Say A is the language of binary strings with odd number of `1`'s. Then strings `0` and `10001` are indistinguishable with respect to A.

Distinguishable Strings and States

Theorem. *If M is DFA accepting language L, and x and y are distinguishable strings with respect to L, then M must be in a different state after reading x than after reading y.*

Suppose strings x and y put M in same state. Then for any string z the strings xz and yz put M in the same state. So either xz and yz are both in L or both are out. Hence x and y are indistinguishable.

Sets of Distinguishable Strings

Corollary. *If $\mathcal{D}_L$ is set of **pairwise** distinguishable strings with respect to L, then any DFA for L has at least $|\mathcal{D}_L|$ states. In particular, if $\mathcal{D}_L$ is infinite then L is not regular.*

Example: $\mathtt{0}^n\mathtt{1}^n$

Recall $B = \{\mathtt{0}^n\mathtt{1}^n : n \geq 0\}$.

The set $\mathcal{D}_B = \{\mathtt{0}^j : j \geq 0\}$ is pairwise distinguishable. Well, take any two strings in $\mathcal{D}_B$: say $\mathtt{0}^j$ and $\mathtt{0}^{j'}$ with $j \neq j'$. Appending $\mathtt{1}^j$ to the first produces a string in B, but appending $\mathtt{1}^j$ to the second produces a string not in B. That is, $\mathtt{0}^j$ and $\mathtt{0}^{j'}$ are distinguishable.

Since $\mathcal{D}_B$ is infinite, B is not regular.

Example: Lower Bound on Number of States

Consider the language of all binary strings with an even number of both `0`'s and `1`'s. The set $\{\varepsilon, \texttt{0}, \texttt{1}, \texttt{01}\}$ is pairwise distinguishable. So the product construction DFA we built has the fewest states possible.

(In general, there is a connection between smallest number of states of a DFA and the largest set of pairwise distinguishable strings.)

Practice

Define E as the language of all binary strings with an ***equal*** number of `0`'s and `1`'s.

Show that E is not regular by finding an infinite set of pairwise distinguishable strings.

Solution to Practice

Let D_E be the set of all strings containing only `0`'s. Then for $i \neq j$ the strings $\texttt{0}^i$ and $\texttt{0}^j$ are distinguishable with respect to E, since $\texttt{0}^i\texttt{1}^i \in E$ but $\texttt{0}^i\texttt{1}^j \notin E$.

The set D_E is infinite, and so E is nonregular.

The Pumping Lemma

The infamous Pumping Lemma says that every regular language has a certain repetitiveness about it:

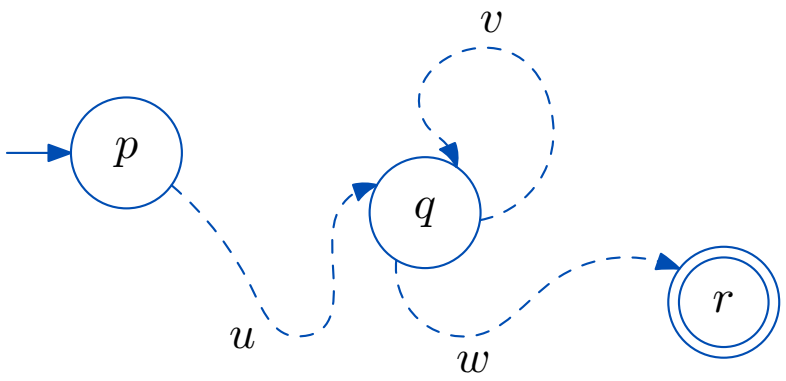
Pumping Lemma. *Let A be a regular language accepted by a DFA with k states. Then, for any string z in A with at least k symbols, one can find an early internal subsegment that can be pumped. That is, z can be split as uvw where:*

- *v is nonempty,*
- *$|uv| \leq k$, and*
- *$uv^i w$ is in A for all $i \geq 0$.*

Proof of Pumping Lemma

Follow the sequence of states around DFA for A on input z. Let q be first of the k states to recur.

Then split the string z as follows:



It follows that the DFA is in same state r no matter whether it has read uw, uvw or uv^2w. Hence, uv^iw is in A for all $i \geq 0$.

Using the Pumping Lemma

The Pumping Lemma is used to show that a language is nonregular, by showing that the lemma is contradicted.

One needs to choose ***one*** string z that does not pump.

Example: $\mathtt{0}^n\mathtt{1}^n$

Consider again language $B = \{\, \mathtt{0}^n\mathtt{1}^n : n \geq 0 \,\}$.

Suppose B were regular.
Then B would be accepted by DFA with k states.
Consider the specific string $z = \mathtt{0}^k\mathtt{1}^k$. This is in B.
Split $z = uvw$ according to Pumping Lemma.
Then since $|uv| \leq k$, it follows that v is composed entirely of $\mathtt{0}$'s. But then uw is not in B (since has fewer $\mathtt{0}$'s than $\mathtt{1}$'s).
This contradicts the Pumping Lemma.

Example: Palindromes

A ***palindrome*** is a word that reads the same backwards as it does forward (such as "level"). Let P be the set of all palindromes for alphabet $\{\mathtt{a}, \mathtt{b}\}$. P is nonregular.

Suppose P were regular. Then it would be accepted by DFA with say k states. Consider the string $z = \mathtt{a}^k\mathtt{b}\mathtt{a}^k$. Split $z = uvw$ according to Pumping Lemma. Then since $|uv| \leq k$, it follows that v is just $\mathtt{a}$'s. Thus uw, is not in P, a contradiction of the Pumping Lemma.

Practice

Define E as the language of all binary strings with an ***equal*** number of `0`'s and `1`'s.

Show that E is not regular by showing that it contradicts the Pumping Lemma.

Practice Solutions

Suppose E were regular. Then it would be accepted by a DFA with say k states.

Consider specific $z = \texttt{0}^k\texttt{1}^k$. Split $z = uvw$ according to the Pumping Lemma. Then because $|uv| \leq k$, v is always just `0`'s. Thus uw is not in E.

This contradicts the Pumping Lemma.

Overview

A regular language is one which has an FA or an RE. Regular languages are closed under union, concatenation, star, and complementation. To show that a language is nonregular, you can show that there is an infinite set of pairwise distinguishable strings, or use the Pumping Lemma and show that there is some string that cannot be pumped.