CmpE598 Lecture Notes

Fatih Mehmet ATAK

March 20, 2018

Let L_n be a language containing the single string a^n on alphabet $\{a\}$. For any $\epsilon > 0$, there is a PFA with $O(log^2 n)$ states, recognizing L_n with error bound ϵ.

- We will use $O(\frac{\log n}{\log \log n})$ different prime numbers.
- We will use $O(\frac{\log n}{\log \log n})$ states in the machine for every prime number we use.
- The machine starts by randomly choosing one of the primes, say, p.
- Then the remainder $modulo\, p$ of the length of the input is counted, and compared with the desired value.
- We define accept states by marking the $n\, modulo\, p_i$ as the accept state.
- Additionally, once every p steps, a transition to a rejecting state is made with a small probability of the form $c\frac{p}{n}$, where c is a suitable constant.
- The number of used primes is sufficient, to say that, for every input of length less than n, most of the primes give remainders different than $n\, mod\, p$, and the "small" probability is chosen to have the rejection probability high enough for every input length N such that $N \neq n$ and ϵ-fraction of all the primes used have the same remainder as $n\, mod\, p$.

Notes:

- Additional trap state's main purpose is to limit the acceptance ratio of longer strings
- Default layout for DFAs, with single letter alphabets, accepting a single string a^n is called "Spoon Style Machine", and it usually looks like Figure 1.
- Design of the mentioned machine will look like Figure 2.

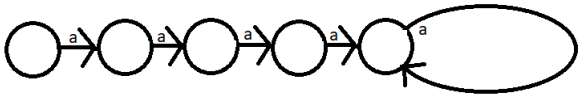

Figure 1: Spoon style DFA

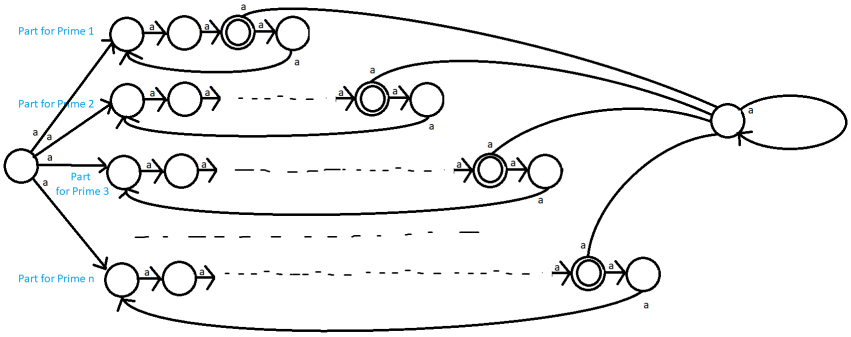

Figure 2: PFA used to recognize L_p with lesser number of states than DFA

Let $L_p = \{a^i | i\, is\, divisible\, by\, p\}$, where p is a prime number. For classical case, we will need $O(p)$ states.

Theorem 1: Any PFA recognizing L_p with error bound $\epsilon > 0$ has at least p states.

Theorem 2: L_p can be recognized by a QFA with $O(\log p)$ states with error bound $\epsilon > 0$.

Since language L_p is on a single letter alphabet, any PFA recognizing it may be described as a Markov Chain.

MARKOV CHAINS

The states of a Markov Chain is divided into ergodic and transient states.

- An **ergodic** set of states is a set which cannot be left when it is entered.
- A **transient** set of states is a set in which every state can be reached from every other state, and which can be left.

States are also divided according to this terminology. *Notice this one refers to single states as opposed to the state sets in above items*

- If a Markov Chain has more than one ergodic state set, then there is no interaction between these sets.
- In that case we have two or more unrelated Markov Chains lumped together.

- If a Markov Chain consists of single ergodic set, it is called an **ergodic chain**.
- Every ergodic chain is either **regular** or **cyclic**.
- If a Markov Chain is regular, then sufficiently high powers of the state transition matrix P of the Markov Chain contain all positive elements. Thus no matter where the process starts, after sufficient time, it can be in any state. More over, there is a limiting vector of probabilities of being in the states of the chain, not dependent on the initial state.
- If a Markov Chain is cyclic, then it has a period d, and its states are subdivided into d cyclic sets ($d > 1$). For a given starting position, it moves through the cyclic sets in a definitive order, returning to the set of the starting state after d steps. Hence dth power of the state transition matrix describes a regular chain.

Theorem 1: Any PFA recognizing L_p with error bound $\epsilon > 0$ has at least p states.

Proof: Assume that the machine has fever than p states. So for every cyclic state of the automaton, the value of d (period) is strictly less than p and since p is prime, d is relatively prime to p.

Let D denote the *Least Common Multiple* of all such values of d. So D is relatively prime to p, and so is any positive power D^n of D.

$a^{D^n} \notin L_p$, but $a^{D^n p} \in L_p$, so the sum of the probabilities of accept states must be greater than or equal to $1 - \epsilon$ for $a^{D^n p}$, but it must be less than or equal to ϵ for a^{D^n}.

Explanation of Proof: Here we expect the Markov Chain to reach a Stationary Distribution. *(A Stationary Distribution is a state, in which probabilities remain constant, no matter how many times you apply transition matrix.)* But in this case, the probabilities alternate between ϵ and $1 - \epsilon$ depending on the number of letters. So, this leads to a contradiction.

Theorem 2: L_p can be recognized by a QFA with $O(\log p)$ states with error bound $\epsilon > 0$.

Proof: We consider the automata U_k, for each $k \in \{1, 2, ..., p-1\}$ Each U_k is a "rotation machine" with two states, where the angle is $\phi = \dfrac{2\pi k}{p}$.

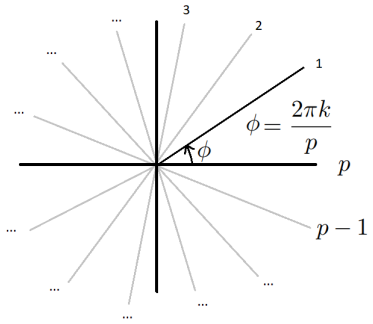

Figure 3: After p rotations, it comes onto $2\pi k = 0$

After reading a^j, U_k is in the superposition of

$$\cos\left(\frac{2\pi jk}{p}\right)|q_0\rangle + \sin\left(\frac{2\pi jk}{p}\right)|q_1\rangle$$

Definition: For any $a^j \notin L$, call U_k "good" if it rejects a^j with probability at least $\frac{1}{2}$.

Claim: For any $a^j \notin L$, at least $\dfrac{(p-1)}{2}$ of all U_k's are "good". The probability of U_k accepting a^j is $\cos^2\left(\dfrac{2\pi jk}{p}\right)$.

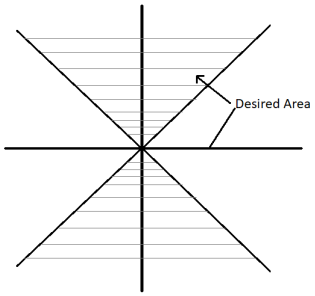

Figure 4: Super positions landing in the "Desired Area" are considered "good".

$$\cos^2\left(\frac{2\pi jk}{p}\right) \leq \frac{1}{2} \text{ iff } \left|\cos\left(\frac{2\pi jk}{p}\right)\right| \leq \frac{1}{\sqrt{2}}$$

This happens iff $\dfrac{2\pi jk}{p}$ is in $\left[2\pi\ell + \dfrac{\pi}{4}, 2\pi\ell + \dfrac{3\pi}{4}\right]$ or in $\left[2\pi\ell + \dfrac{5\pi}{4}, 2\pi\ell + \dfrac{7\pi}{4}\right]$ for some $\ell \in N$

This is so iff $\dfrac{2\pi(jk \mod p)}{p}$ in $\left[\dfrac{\pi}{4}, \dfrac{3\pi}{4}\right]$ or in $\left[\dfrac{5\pi}{4}, \dfrac{7\pi}{4}\right]$.

Since j is relatively prime to p, $\{j \mod p, 2j \mod p, ..., (p-1)j \mod p\}$ are just $\{1, 2, ...(p-1)\}$ in different order.

Given $aj \equiv bj(\mod p)$, you can divide both sides by j, iff j is relatively prime to p.

Let's count: For the case where $p = 8m + 1$, for some m, then

$$\frac{2\pi(jk \mod p)}{p} \; in \; \left[\frac{\pi}{4}, \frac{3\pi}{4}\right] \Rightarrow \frac{2\pi jk}{p} \geq \frac{\pi}{4} \tag{1}$$

and

$$\Rightarrow \frac{2\pi jk}{p} \leq \frac{3\pi}{4} \tag{2}$$

Combine (1) and (2), we get

$$m + 1 \leq k \leq 3m$$

Do the same for the other interval, $\left[\frac{5\pi}{4}, \frac{7\pi}{4}\right]$, we obtain

$$5m + 1 \leq k \leq 7m$$

For other possible cases of p (i.e. $8m + 3$, $8m + 5$ and $8m + 7$), this works out similarly.

** There are $4m$ values of k satisfying this condition.*

Definition: A sequence of U_k's is good for a particular input a^j if at least $\frac{1}{4}$ of al its elements are good for that input string, a^j.

Claim: There is a sequence of length $\lceil 8 \ln p \rceil$ which is good for all $a^j \notin L$

Proof: Consider picking $\lceil 8 \ln p \rceil$ elements randomly from the set of all U_k's.

For a fixed a^j, the probability that we select a good k at each step is at least $\frac{1}{2}$.

What is the probability that less than $\frac{1}{4}$ of the U_k's that I pick will be good?

This is at most $e^{-2\left(\frac{1}{4}\right)^2 8 \ln p} = \frac{1}{p}$, because of the **Chernoff Bound (*)**.

(*) Chernoff Bound is used to calculate a limit for tail distributions of sums of independent random variables. Here we assume picking each of the U_k's are independent random variables and use Chernoff Bound to find a limit for the probability.

So the fraction of sequences which are bad for at least one $j \in \{1, 2, ..., p-1\}$ is at most $\frac{p-1}{p}$.

Explanation of Proof: This proof does not show the actual method to pick a sequence, but it shows the possibility of not being able to pick a suitable sequence is less than 1. That means, there is always a possibility, however small it is, to pick a sequence satisfying our condition.

Build a single QFA using the machines in a "good" sequence. The new QFA branches from its start state with equal probabilities to the starting states of all the machines in the sequence. Note that this machine would accept members of the language with probability 1, and reject nonmembers with probability at least $\frac{1}{8}$.

To decrease the error probability down to a desired value from $\frac{7}{8}$, we can change the procedure described in the previous paragraph as follows: Let us first change each U_k to a much "better" machine "SuperU_k" by taking sufficiently many tensor products of it (constructing a machine that runs sufficiently many copies of it parallelly) such that the resulting machine says yes only if all constituent machines say yes. The probability of an incorrect acceptance is much lower now. We then merge a good sequence containing $O(\log p)$ of these SuperU_k's in the manner described above. The resulting machine's error probability can be tuned by playing with the number of tensor products above. The total number of states is still $O(\log p)$, since the size of the individual SuperU_k's does not depend on p.