

Main topics of the week:

- **Pairwise distinguishability**
- **State minimization and Smallest DFA**
- **Midterm review**

The following material is based on problems 1.51 and 1.52 in Sipser.

After seeing the DFA and NFA equivalence and the constructive proof to find the equivalent DFA for an NFA with its exponential state explosion, it is natural to ask whether we can determine the “smallest” DFA (in terms of number of states) that recognizes a particular language. First of all, we might ask whether such a smallest DFA is guaranteed to exist. Well, we would need to be able to determine whether two DFAs are equivalent. Since regular languages are closed under complement (one of the exercises) and intersection, we know that the set difference of two regular languages is regular and this means there is a DFA to recognize it. We can also determine if a DFA recognizes the empty language – this would simply involve a path exploration to see if there are any paths that lead to an accept state. If so, the language is non empty, if not, the language is empty. So we could determine if the difference of two languages is empty (two differences, really: $A \cap \sim B$ and $B \cap \sim A$) and thus if the languages are the same, and thus if the DFAs recognizing them are equivalent. Now, we go back to a DFA and consider all equivalent DFAs, and partially order them according to the number of states. This assures us there is a DFA recognizing the language with a minimum number of states.

Definition: Let x and y be any strings and L be any language. We say that x and y are **distinguishable** by L if some string z exists such that exactly one of the strings xz and yz is in L . Otherwise for every string z , $yz \in L \Rightarrow xz \in L$ and we say that x and y are **indistinguishable** by L . In this case, we write $x \equiv_L y$.

Lemma: Indistinguishability is an equivalence relation.

Proof: This is pretty obvious. Certainly it is reflexive and symmetric. And it’s pretty clearly transitive since if $x \equiv_L y$ and $y \equiv_L w$, and z is any string, and $wz \in L$, then $yz \in L$ by the indistinguishability of y and w , and thus $xz \in L$ by the indistinguishability of x and y . So $x \equiv_L w$.

Definition: Let L be a language and X a set of strings. We say that X is **pairwise distinguishable by L** if every two distinct strings in X are distinguishable by L . We define the **index of L** to be the maximum number of elements in any set that is pairwise distinguishable by L , and this could be finite or infinite.

Lemma: If L is recognized by a DFA with k states, then L has index at most k .

Proof: Let M be a k -state DFA that recognizes L . Suppose that L has index greater than k , i.e., there is a set X with more than k elements and X is pairwise distinguishable by L . Then there must be two distinct strings x and y in X which transition to the same state in

M (by the pigeonhole principle, since there are only k states, but $k+1$ or more strings, they can't all transition to unique states). This would also imply that for any string z , xz and yz transition to the same state. In particular, if yz transitions to an accept state, so does xz , thus $x \equiv_L y$. But this contradicts the assumption of X being pairwise distinguishable by L . So our lemma is proved.

Lemma: If the index of L is a finite number k , then it is recognized by a DFA with k states.

Proof: We will construct a DFA M to recognize L . Let $X = \{s_1, s_2, \dots, s_k\}$ be pairwise distinguishable by L (as guaranteed by the index of L being k). Define M as follows:

- 1) The states of M are $\{q_1, q_2, \dots, q_k\}$ (i.e., one state for each string in X)
- 2) $\delta(q_i, a) = q_j$ where $s_j \equiv_L s_i a$. We know that s_j exists since otherwise $s_i a$ would be distinguishable from every element of X , meaning that $X \cup \{s_i a\}$ is a larger set pairwise distinguishable by L .
- 3) The accept states of M are $\{q_i \mid s_i \in L\}$
- 4) The start state of M is the q_i such that $s_i \equiv_L \epsilon$.

We claim that M recognizes L . The start state corresponds to a string indistinguishable from ϵ . Each transition moves to the state corresponding to a string indistinguishable by appending the input character. And the accept states correspond to strings indistinguishable from the strings in L . Moreover, since $x \equiv_L s_i$ implies $xa \equiv_L s_i a$ for any a ($s_i a z = s_i(a z) \in L \Rightarrow x(a z) = x a z \in L$), we can see that the computation sequence for any string $a_1 a_2 \dots a_n \in L$ is just the set of states associated with the equivalence classes of $\epsilon, a_1, a_1 a_2, a_1 a_2 a_3, \dots, a_1 a_2 \dots a_n$ and the last corresponds to an accept state of M . For the other direction, any string accepted is likewise indistinguishable from the string of its acceptance state, namely a string in L , so must itself be in L (using ϵ in the indistinguishability definition).

Theorem: L is regular if and only if it has finite index. Moreover, its index is the size of the smallest DFA recognizing it.

Proof: By the first lemma, if L is regular, it has index at most k , where k is the number of states in a DFA recognizing it. By the second lemma, if L has finite index k , it is recognized by a DFA with k states, so is regular. As for the smallest part, suppose L has index k . Then we have a DFA with k states recognizing L by the construction in the lemma. If there was a DFA with a smaller number of states, then the first lemma would mean that the index of L was no more than that number of states, i.e., smaller than k . But we chose k to be the index of L , so this would be a contradiction.

So what do these results say? They suggest a way to find the minimal state DFA for a language by basing it on the equivalence classes of the indistinguishability relation. In practice, an algorithm to minimize the states of a DFA would proceed toward indistinguishability by degrees, i.e., indistinguishability by strings of length n , and using this to lump states together. Eventually, this algorithm would achieve a steady state, at which point the DFA would have minimal states.