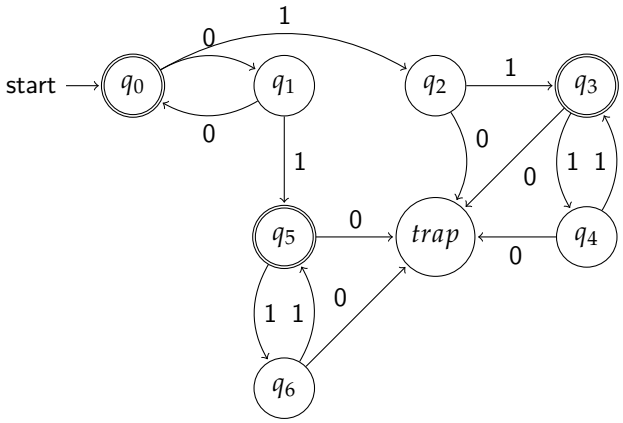
CS208 Tutorial 5: More on automata theory

---

1. Consider the DFA shown below that accepts the language $\{0^n 1^m \mid n+m \text{ is even}\}$. Assume that the trap state loops back to itself on all letters of Σ.



   (a) Using the method discussed in class, find all distinguishable and indistinguishable pairs of states in the above DFA. You can record this by constructing an upper-triangular (or lower-triangular) matrix with 8 rows and 8 columns (corresponding to 8 states of the DFA), as discussed in class.

   (b) Find all equivalence classes of the indistinguishability relation obtained above.

   (c) Using one state from each equivalence class to represent all states of the class, construct a minimal DFA for the language represented by the above DFA.

2. Consider a language $L \subseteq \Sigma^*$ for some finite alphabet Σ. As discussed in class, the *Nerode equivalence* $\sim_L$ is an equivalence relation over Σ^* such that for any $x, y \in \Sigma^*$, $x \sim_L y$ if and only if for every $z \in \Sigma^*$, $xz \in L \iff yz \in L$. The relation $\sim_L$ partitions Σ^* into equivalence classes of words. Hence, each equivalence class of $\sim_L$, viewed as a set of words, is a language by itself.

   Recall further from our discussion in class:

   - The Nerode equivalence is well-defined for every (regular or non-regular) language L over Σ.
   - The *Myhill-Nerode Theorem* states that L is regular if and only if the number of equivalence classes of $\sim_L$ is finite.
   - If the number of equivalence classes of $\sim_L$ equals $k \in \mathbb{N}$, then the unique (upto isomorphism) minimal DFA recognizing L has k states.

   In this problem we will explore the Nerode equivalence and some of its variants.

   (a) For each of the following languages L, describe (in any suitable form) the equivalence classes of $\sim_L$ as languages over $\{0, 1\}$.

      (a) L is the language corresponding to $(\mathbf{00} + \mathbf{11})^*$

      (b) L is the language $\{0^i 1^j \mid i \leq j\}$

   (b) Define an equivalence relation $\sim_R$ such that for any $x, y \in \Sigma^*$, $x \sim_R y$ if and only if for every $z \in \Sigma^*$, $zx \in L \iff zy \in L$. *Note the difference of $\sim_R$ from the Nerode equivalence $\sim_L$.*

      (i) Show that the number of equivalence classes of $\sim_R$ is finite if and only if L is regular.

      (ii) Let L_{rev} denote the language formed by reversing each string in L. Show that if the number of equivalence classes of $\sim_R$ is k, then the size of the unique minimal DFA recognizing L_{rev} is also k.

3. dA hacker must figure out what a language L is in order to break into a top-secret system. The hacker knows that the language L is regular and that it is over the alphabet $\{0,1\}$. However, no other information about L is directly available. Instead, an oracle is available that only answers "Yes" or "No" in response to specific types of queries, labeled Q1 and Q2 below.

   Q1 **Does there exist any DFA with n states that recognizes L?**
   For every $n > 0$, the oracle truthfully responds "Yes" or "No" to this query.

   Q2 **Does word w belong to L?**
   For every $w \in \{0,1\}^*$, the oracle truthfully responds "Yes" or "No" to this query.

   We are required to help the hacker re-construct a minimal DFA for L. Towards this end, we will proceed systematically as follows.

   (a) Show that if the minimal state DFA for L has N states, then N can be determined using a sequence of $\mathcal{O}(\log_2 N)$ Q1 queries.
   Hint: Use galloping (or exponential) search.

   (b) Show that it is possible to find a word $w \in L$ or determine that $L = \emptyset$ using at most 2^N Q2 queries.
   Hint: Consider any word in L and repeatedly apply the Pumping Lemma to remove loops in the path from the initial state to an accepting state.

   (c) Once we know the minimal count of states, say N, for a DFA for L, we will construct the Nerode equivalence classes $\sim_L$ for L. Recall from our discussion in class that there are exactly N of these, and each equivalence class can be uniquely identified with a state of the minimal DFA recognizing L.
   For any two distinct equivalence classes of $\sim_L$, show the following:
      (i) There exist words $w_1, w_2 \in \Sigma^*$, where $|w_1| \leq N-1$ and $|w_2| \leq N-1$ such that w_1 belongs to the first equivalence class and w_2 to the second. We will use $[w_1]$ to denote the first equivalence class and $[w_2]$ to denote the second, in the discussion below.
      (ii) For $[w_1] \neq [w_2]$, there is a word $x \in \Sigma^*$ of length $\leq N \times (N-1) - 1$ such that $w_1 \cdot x \in L$ and $w_2 \cdot x \notin L$ or vice versa.
      (iii) For $[w_1] \neq [w_2]$, there exists an edge labeled 0 (resp. 1) from the state corresponding to $[w_1]$ to the state corresponding to $[w_2]$ iff for all $x \in \Sigma^*$, where $|x| \leq N \times (N-1) - 1$, $w_1 \cdot 0 \cdot x$ (resp. $w_1 \cdot 1 \cdot x$) and $w_2 \cdot x$ are either both in L or both not in L.

   Using all the above results, design an algorithm that helps the hacker reconstruct the minimal DFA for L. Give an upper bound on the count of Q2 queries needed for this re-construction, in terms of the count N of the states of the minimal DFA for L.

4. **Takeaway**: You can view this question as a continuation of Question 1 on Nerode equivalences and their variants. Define an equivalence relation $\sim_S$ such that for any $x, y \in \Sigma^*$, $x \sim_R y$ if and only if for every $u, v \in \Sigma^*$, $uxv \in L \iff uyv \in L$.

   (i) Show that the number of equivalence classes of $\sim_S$ is finite if and only if L is regular.

   (ii) Assuming that L is regular, if the minimal DFA recognizing L has k states, show that the number of equivalence classes of $\sim_S$ is at most k^k

5. **Takeaway**: Let $\Sigma = \{a\}$.

   (i) Show that for every language L (regular or not) over Σ, the language $L^* = \bigcup_{i=0}^{\infty} L^i$ is regular.

   (ii) Show that for every regular language L over Σ, there exist two finite sets of words S_1 and S_2 and an integer $n > 0$ such that $L = S_1 \cup S_2 \cdot (a^n)^*$

6. **Takeaway**: The *star-height* of a regular expression $\mathbf{r}$, denoted $\mathsf{SH}(\mathbf{r})$, is a function from regular expressions to natural numbers. It is defined inductively as follows:

   - $\mathsf{SH}(\mathbf{0}) = \mathsf{SH}(\mathbf{0}) = \mathsf{SH}(\boldsymbol{\varepsilon}) = \mathsf{SH}(\boldsymbol{\Phi}) = 0$.
   - $\mathsf{SH}(\mathbf{r_1} + \mathbf{r_2}) = \mathsf{SH}(\mathbf{r_1} \cdot \mathbf{r_2}) = \max(\mathsf{SH}(\mathbf{r_1}), \mathsf{SH}(\mathbf{r_2}))$
   - $\mathsf{SH}(\mathbf{r}^*) = \mathsf{SH}(\mathbf{r}) + 1$

   Give a regular expression $\mathbf{r}$ over $\Sigma = \{0, 1\}$ such that the following hold:

   - $\mathsf{SH}(\mathbf{r}) > 0$, and
   - Every regular expression with star-height $< \mathsf{SH}(\mathbf{r})$ represents a language different from that represented by $\mathbf{r}$.

   You must give brief justification why no regular expression with lesser star-height can represent the same language.