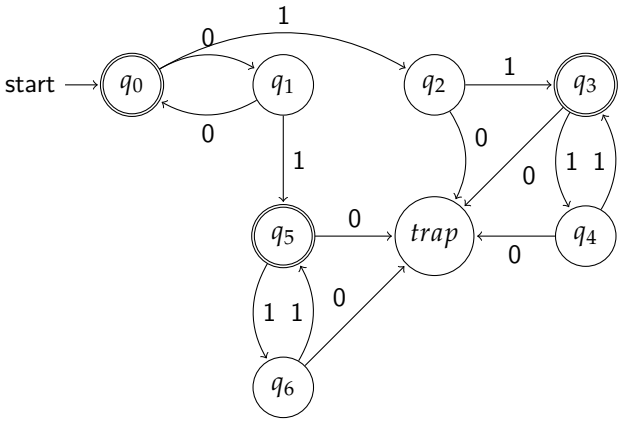
CS208 Tutorial 5: More on automata theory

---

1. Consider the DFA shown below that accepts the language $\{0^n1^m \mid n+m \text{ is even}\}$. 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.

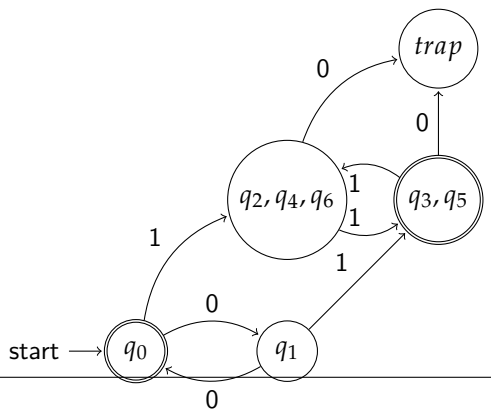
Solution:

(a) Table as shown (State 7 is trap). Each entry is a distinguishing string, if it exists

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	X	ϵ	ϵ	01	ϵ	01	ϵ	ϵ
1		X	0	ϵ	0	ϵ	0	0
2			X	ϵ		ϵ		1
3				X	ϵ		ϵ	ϵ
4					X	ϵ		1
5						X	ϵ	ϵ
6							X	1
7								X

(b) The equivalence classes are $\{q_0, q_1, (q_2, q_4, q_6), (q_3, q_5), trap\}$

(c) The minimized DFA is as shown below:



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.

Solution:

(a) (a) For a regular language like the one in this question, one way to obtain the Nerode equivalence classes is to first construct the *minimal* DFA for the language, and then for each state q in the minimal DFA, list down the set of strings that bring you from the start state of the DFA to state q. Clearly, all such strings w must be in the same Nerode class, since for every string $x \in \Sigma^*$, whether $wx \in L$ or not simply depends on whether you can reach the accepting state of the minimal DFA from q on reading x. Similarly, every string w' that doesn't bring you from the start state to q can't be Nerode equivalent to w since w' must be brining you to a different state q', starting from the start state. However, since q and q' are two different states in the minimal DFA for L, there is a distinguishing string x' such that x' is accepted starting from q and not accepted starting from q' or vice versa. It follows that one of $w.x'$ and $w'.x'$ is in L and the other isn't. Hence, w and w' can't be in the same Nerode class.

We leave it as an exercise for you to construct the minimal DFA for the given language L. Once you do that, it is easy to follow the steps outlined above to find the following Nerode equivalence classes:

$S_1 = L$ (set of strings that bring you to the accepting state of the DFA), $S_2 = (00+11)^*0, S_3 = (00+11)^*1, S_4 = \Sigma^* - \{S_1 \cup S_2 \cup S_3\}\}$

(b) Since this language is not regular (why? Try using the Pumping Lemma for regular languages), we can't use the above method for finding the Nerode equivalence classes. Hence, we have to look into the specifics of the language and try to construct infinitely many Nerode equivalence classes (Myhill-Nerode theorem guarantees that there are infinitely many Nerode equivalence classes for a non-regular language).

It is easy to see that all strings w not belonging to 0^*1^* are equivalent, since no matter what string x you concatenate to w, the string $w.x$ is not in 0^*1^*, and hence not in L. So all strings not in 0^*1^* form one Nerode equivalence class, say C_1

Let us now focus on strings in 0^*1^*. Consider one such string $w = 0^i1^j$, where $j \geq i$ and $w \neq \varepsilon$ (i.e. it is not the case that $j = i = 0$). Notice that for every string $x \in 1^*$, $w.x \in L$. Similarly, for every string $x \notin 1^*$, $w.x \notin L$. Hence, all strings $w = 0^i1^j$, where $j \geq i$ and $w \neq \varepsilon$ are in the same Nerode equivalence class, say C_2 (they cannot be distinguished by any string $x \in \Sigma^*$). Moreover, C_1 is different from C_2, since ε distinguishes any string in C_1 from any string in C_2. What if $w = \varepsilon$?. The string $x = 01$ distinguishes w from every string in $C_1 \cup C_2$. Hence ε is in a class, say C_3, by itself.

What about strings of the form 0^i1^j, where $i > j$. Consider any such string $w = 0^i1^j$. The string $x = 1^{i-j}$ distinguishes w from every string in C_1. The string ε distinguishes w from every string in C_2. The string $x = 01$ distinguishes w from ε. The string $x = 1^{\min(i-j,i'-j')}$ distinguishes w from every string $w' = 0^{i'}1^{j'}$, where $i' > j'$ and $i - j \neq i' - j'$. Finally, for every string $w' = 0^{i'}1^{j'}$, where $i' > j'$ and $i - j = i' - j'$, no string x can distinguish w from w'. Therefore, all strings $0^{i+k}1^i$ belong to the same Nerode equivalence class for each $k > 0$, and the classes corresponding to k_1, k_2, where $k_1 \neq k_2$ are distinct.

Summarizing all the above cases, the Nerode equivalence classes are: $C_1 = L(0^*1^*)^c, C_2 = L \setminus \{\varepsilon\}, C_3 = \{\varepsilon\}, C_{3+k} = \{0^{i+k}1^i | i \geq 0\}$, for every $k \geq 1$.

(b) Let $\sim_N$ denote the Nerode equivalence relation for the language L_{rev} formed by reversing each string in L. Now, $x \sim_R y$ if and only if for every $z \in \Sigma^*$, $zx \in L \iff zy \in L$, ie $x^Rz^R \in L_{rev} \iff y^Rz^R \in L_{rev}$ for every $z \in \Sigma^*$, ie $x^R \sim_N y^R$.

Consider the function f mapping equivalence classes of $\sim_R$ to equivalence classes of $\sim_N$ such that for any $x \in \Sigma^*$, $f([x]_R) = [x^R]_N$. Since for any $x, y \in \Sigma^*$, $x \sim_R y$ if and only if $x^R \sim_N y^R$, f is well defined and also an injection, and since $(x^R)^R = x$ for any $x \in \Sigma^*$, $[x]_N = f([x^R]_R)$, ie f is a surjection as well. Hence there exists a bijection between the equivalence classes of $\sim_R$ and the equivalence classes of $\sim_N$, ie they have the same cardinality.

Now, by the *Myhill-Nerode Theorem*, the number of equivalence classes of $\sim_N$ is finite if and only if L_{rev} is regular. Now, for any language L, L_{rev} is regular if and only if L is regular (this can be seen by reversing the transitions in the DFA recognizing L to get an NFA recognizing L_{rev}, and also noting that $(L_{rev})_{rev} = L$).

Considering this, along with the fact the set of equivalence classes of $\sim_R$ has the same cardinality as the set of equivalence classes of $\sim_N$, we get that $\sim_R$ has a finite number of equivalence classes if and only if L is regular. Moroever, if $\sim_R$ has k equivalence classes, then so does $\sim_N$, which, by the *Myhill-Nerode Theorem* means that the unique minimal DFA recognizing L_{rev} has k states.

3. A 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.

Solution:

(a) We make Q1 queries using powers of 2 $(1, 2, 4, \ldots)$ until it returns "yes". Say, it returns yes for 2^{k+1}. Then, we know the smallest DFA representing the language has size between $2^k + 1$ and 2^{k+1}. We perform binary search over this space. Total number of queries are $k + 2 + \mathcal{O}(\log(2^{k+1} - 2^k)) = 2k + 2 \in \mathcal{O}(\log_2 N)$

(b) **Claim**: A DFA whose language is non-empty having N states accepts a word of length at most $N-1$. Proof is left as an exercise to the reader (Use ideas similar to Pumping Lemma). Hence, we can make Q2 queries over all possible words having length less than or equal to $N-1$. Either we conclude that the language is empty or find a word belonging to the language in at most $2^0 + 2^1 + \ldots 2^{N-1} = 2^N - 1$ Q2 queries

(c) (i) If q_1 and q_2 are the states corresponding to these equivalence classes in the minimal DFA (and q_0 is the initial state), then a word w is in the equivalence class of q_1 iff $\hat{\delta}(q_0, w) = q_1$ (and similarly for q_2). Since equivalence classes are by definition non-empty, such a word w_1 necessarily exists. We can remove cycles in the path this word takes from q_0 to q_1 word to ensure that it's length is at most $N-1$ (Similarly for w_2).

(ii) Let q_1 be the state corresponding to $[w_1]$ and q_2 the state corresponding to $[w_2]$. Since $[w_1]$ and $[w_2]$ are distinct equivalence classes, there must exist a string x such that exactly one of w_1x and w_2x are in L. We will show that there exists such an x with length at most $N-2$.

Consider an equivalence relation $\sim_k$ over the state set Q of the minimal DFA where $q_1 \sim_k q_2$ iff for every string x of length at most k, $\hat{\delta}(q_1, x) \in F \iff \hat{\delta}(q_2, x) \in F$. Some observations:

i. $q_1 \sim_0 q_2$ iff $q_1 \in F \iff q_2 \in F$

ii. $q_1 \sim_{k+1} q_2 \implies$ q_1 $\sim_k q_2$, ie the equivalence classes of $\sim_{k+1}$ are subsets of those of $\sim_k$

By the second observation, $\sim_{k+1}$ has at least as many equivalence classes as $\sim_k$, and if the number of equivalence classes is the same, then $\sim_{k+1}$ and $\sim_k$ are identical. Note that $\sim_0$ has 2 equivalence classes. This means that in the number of equivalence classes of the sequence $\sim_0, \sim_1, \sim_2, \ldots$ keeps increasing from 2, until some k where $\sim_k = \sim_{k+1}$, after which it remains constant. Since the number of equivalence classes of $\sim_0$ is 2, and the number of equivalence classes of $\sim_k$ is at most N, k can be at most $N-2$.

Now, for distinct q_1, q_2, since the DFA is minimal, there exists some string x such that exactly one of $\hat{\delta}(q_1, x)$ and $\hat{\delta}(q_2, x)$ lies in F. Say $|x| = p$. Then $q_1 \nsim_p q_2$. If $p > N-2$, then $\sim_p = \sim k$, ie $q_1 \nsim_k q_2$, ie there is some x' of length at most k such that exactly one of $\hat{\delta}(q_1, x')$ and $\hat{\delta}(q_2, x')$ is in F. Since k is at most $N-2$, this means for any distinct q_1 and q_2 there will exist a string x of length at most $N-2$ such that exactly one of $\hat{\delta}(q_1, x)$ and $\hat{\delta}(q_2, x)$ lies in F. This means that for distinct $[w_1]$ and $[w_2]$ there will exist an x of length at most $N-2$ such that exactly one of w_1x and w_2x is in L.

Algorithm:
Find the value of N (Part a). Consider all words having length less than or equal to $N-1$. Find equivalence classes over these words by taking pairs at a time and iterating over all words having length less than or equal to $N-2$. If you find a distinguisher, they're in different equivalence classes, else they are in the same equivalence class. Each equivalence class now represents a state. Accepting state is simply found by using Q2 on one word in each equivalence class. To find the transition function, we can simply consider the shortest words in each equivalence class and use their prefixes to construct the path from the starting state. The starting state is the class which contains epsilon.

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

Solution: Say L is regular and is recognized by minimal DFA $(Q, \Sigma, \delta, q_0, F)$.

If $x \sim_S y$, then for every state $q \in Q$ we must have $\hat{\delta}(q_0, x) = \hat{\delta}(q_0, y)$. To see this, note that since $(Q, \Sigma, \delta, q_0, F)$ is the minimal DFA recognizing L, for every $q \in Q$ there exists $u \in \Sigma^*$ such that $\hat{\delta}(q_0, u) = q$. Furthermore, if $q_1 \neq q_2$ are distinct states in Q, then there exists $v \in \Sigma^*$ such that exactly one of the following are true:

- $\hat{\delta}(q_1, v) \in F$
- $\hat{\delta}(q_2, v) \in F$

(otherwise the states q_1 and q_2 could be merged). Now, if $x \sim_S y$, but if there is some q such that $\hat{\delta}(q, x) \neq \hat{\delta}(q, y)$ (call these q_1 and q_2), then there exists $u \in \Sigma^*$ such that $\hat{\delta}(q_0, u) = q$ and there exists $v \in \Sigma^*$ such that (WLOG) $\hat{\delta}(q_1, v) \in F$ and $\hat{\delta}(q_2, v) \notin F$. This means that there exist $u, v \in \Sigma^*$ such that $\hat{\delta}(q_0, uxv) \in F$ but $\hat{\delta}(q_0, uyv) \notin F$, which means $uxv \in L$ but $uyv \notin L$, contradicting the definition of $\sim_S$.

Moreover, if $\hat{\delta}(q, x) = \hat{\delta}(q, y)$ for every $q \in Q$, then for every $u, v \in \Sigma^*$, $\hat{\delta}(q_0, uxv) = \hat{\delta}(q_0, uyv)$, ie $uxv \in L \iff uyv \in L$, ie $x \sim_S y$. Therefore, for any $x, y \in \Sigma^*$, $x \sim_S y$ if and only if for every $q \in Q$, $\hat{\delta}(q, x) = \hat{\delta}(q, y)$.

Consider the set of functions from Q to itself, denoted by Q^Q. Consider the function f mapping equivalence classes of $\sim_S$ to elements of Q^Q such that for every $q \in Q$, $f([x]_S)(q) = \hat{\delta}(q, x)$. By the previous result, f is well defined and an injection. Therefore, there exists an injection from the equivalence classes of $\sim_L$ to Q^Q, a finite set.

Therefore, if L is regular, then the number of equivalence classes of $\sim_L$ must be finite, and is at most $\left|Q^Q\right|$, where Q is the set of states of the minimal DFA recognizing L. If $|Q| = k$, then we get that the number of equivalence classes is at most k^k.

It is easier to show the other direction of the implication, ie if the number of equivalence classes of L is finite, then L must be regular. This can be done by constructing a DFA recognizing L. Consider the DFA whose set of states $Q = \{[x]_S : x \in \Sigma^*\}$ (ie the set of states is the set of equivalence classes of $\sim_S$), and transitions are of the form $[x]_S \xrightarrow{a} [xa]_S$, for any $a \in \Sigma$ (ie $\delta([x], a) = [xa]$). This transition function is well defined, since if $x \sim_S y$, then $xa \sim_S ya$ for any $a \in \Sigma$. The initial state is taken to be $q_0 = [\varepsilon]_S$ and the set of final states is $F = \{[x]_S : x \in L\}$. It can be shown that the language recognized by this automaton is precisely L (note that $\hat{\delta}(q_0, x) = [x]$ and if $x \sim_S y$ then $x \in L \iff y \in L$).

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 $L = S_1 \cup S_2 \cdot (a^n)^*$

> **Solution:** (a) To those interested, please read here
>
> (b) Refer to the solution of Tut 4, Question 5

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:

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

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

   Solution: The answer to this specific question is really simple if you think about the definition of star height. However, the study of star heights of regular expressions and about the hierarchy of languages corresponding to increasing star heights is very interesting. For those interested in knowing more about star heights, a good starting point is here

   For this specific question, you can simply take the regular expression $\mathbf{0}^*$, which has star height 1. What are the regular expressions with star height < 1. These are $\mathbf{1}, \mathbf{0}, \boldsymbol{\varepsilon}, \boldsymbol{\Phi}$ and combinations of these regular expressions using $+$ and $\cdot$. All of these represent languages with finitely many words, while $\mathbf{0}^*$ represents a language with infinitely many words.

   However, the study of the star height hierarchy is not just limited to star heights of 0 and 1. It extends to all star heights (see here for more details).