CSE491/596 Lecture Friday 9/18/20: Myhill-Nerode Theorem

Given a DFA $M = (Q, \Sigma, \delta, s, F)$ and two strings $x, y \in \Sigma^*$, suppose $\delta^*(s, x)$ and $\delta^*(s, y)$ both give the same state q. Then for any further string $z \in \Sigma^*$, the computations on the strings xz and yz go through the same states after q. In particular, they end at the same state r.

- If $r \in F$, then $xz \in L$ and $yz \in L$, where $L = L(M)$.
- If $r \notin F$, then $xz \notin L$ and $yz \notin L$.
- Either way, $L(xz) = L(yz)$, for all z.

Suppose, on the other hand, we have strings x, y for which there exists a string z such that

$$L(xz) \neq L(yz).$$

Then M cannot process x and y to the same state. Moreover, this goes for *any* DFA M such that $L(M) = L$. In particular, every such DFA must at least *have* two states.

Now let us build some definitions around these ideas. Given any language L (not necessarily regular) and strings x, y "over" the alphabet Σ that L is "over", define:

- x and y are L-*equivalent*, written $x \sim_L y$, if for all $z \in \Sigma^*$, $L(xz) = L(yz)$.
- x and y are *distinctive for* L, written $x \nsim_L y$, if there exists $z \in \Sigma^*$ s.t. $L(xz) \neq L(yz)$.

Lemma 1. The relation $\sim_L$ is an equivalence relation.

Proof: We need to show that it is
- Reflexive: $x \sim_L x$ is obvious.
- Symmetric: indeed, $y \sim_L x$ immediately means the same as $x \sim_L y$.
- Transitive: Suppose $w \sim_L x$ and $x \sim_L y$. This means:
  - for all $v \in \Sigma^*$, $L(wv) = L(xv)$ and
  - for all $z \in \Sigma^*$, $L(xz) = L(yz)$.
    Because v and z range over the same span of strings, it *follows* that
  - for all $z \in \Sigma^*$, $L(wz) = L(xz)$ and $L(xz) = L(yz)$.
    Hence we get:
  - for all $z \in \Sigma^*$, $L(wz) = L(yz)$.
    So $w \sim_L y$.

This ends the proof. ☒

Any equivalence relation on a set such as Σ^* partitions that set into disjoint *equivalence classes*. So $x \nsim_L y$ is the same as saying x and y belong to different equivalence classes. [I intended to give an example but skipped it after the initial loss of time: Start with the language E of strings having an even number of 1s. Then the relation $\sim_E$ has exactly two equivalence classes: one for an even number of 1s, the other for odd. Now if you make E_3 be the language where the number of 1s is a multiple of 3, you get 3 equivalence classes. And so on...]

Now say that a set S of strings is ***Pairwise Distinctive for L*** if all of its strings belong to separate equivalence classes under the relation $\sim_L$. Other names we will use are "distinctive set" and "**PD set**" for L. This is the same as saying:

- for all $x, y \in S$, $x \neq y$, there exists $z \in \Sigma^*$ such that $L(xz) \neq L(yz)$.

Thus we can re-state something we said above as:

Lemma 2. If L has a PD set S of size 2, then any DFA M such that $L(M) = L$ must process the two strings in S to different states, so M must have at least 2 states.

Note: "L has" does not mean S must be a subset of L, it just means "has by association." Now we can take this logic further:

Lemma $\boldsymbol{k}$. If L has a PD set S of size k, then any DFA M such that $L(M) = L$ must process the k strings in S to different states, so M must have at least k states.

I've worded this to try to make it as "obvious" as possible, but actually it needs proof: Suppose we have a DFA M with $k-1$ or fewer states such that $L(M) = L$. Then there must be (at least) two strings in S that M processes to the same state. This follows by the **Pigeonhole Principle**.

[tell story]
[finish proof]
Then explain why we get the infinite case:

Lemma ∞. If L has a PD set S of size ∞, then any DFA M such that $L(M) = L$ must process the strings in S to different states, so M must have at least ∞ states...but then M is not a *finite* automaton. So L is not accepted by any finite automaton...which means **L is not a regular language.** ☒

Myhill-Nerode Theorem, first half: If L has an infinite PD set, then L is not regular.

Example: $L = \left\{a^n b^n : n \geq 0\right\}$. $\Sigma = \{a, b\}$. $S = \left\{a^n : n \geq 0\right\} = a^*$. Let any $x, y \in S$, $x \neq y$, be given. Then there are different numbers i and j such that $x = a^i$ and $y = a^j$. Take $z = b^i$. Then $xz = a^i b^i \in L$, but $yz = a^j b^i \notin L$, because $i \neq j$. Thus $L(xz) \neq L(yz)$. Thus for all $x, y \in S$ with $x \neq y$, there exists z such that $L(xz) \neq L(yz)$. Thus S is PD for L. Since S is infinite, L is not regular, by MNT. ☒

[Then I drew a connection from this to the idea of playing the spears-and-dragons game when you can save any number of spears. In the basic case where you can save at most 1 spear the DFA has 3 states, and these are mandated because $S = \{\epsilon, \$, D\}$ is a PD set of size 3. In particular, even though both $x = \epsilon$ and $y = \$$ are strings **in** the language L_1 of the 1-spear game, they are distinctive **for** L_1 because $z = D$ kills you in the former case (i.e., $xz = \epsilon D = D \notin L_1$) but you stay alive in the latter case (i.e., $yz = \$D \in L_1$). If you can save up to 2 spears, then $\epsilon, \$, \$\$$ are three distinctive strings (plus D to make a fourth). Well, if you can save unlimited spears, then $S_\infty = \{\epsilon, \$, \$\$, \$\$\$, \ldots\}$ becomes an infinite PD set by similar logic to the $\left\{a^n\, b^n\right\}$ example. So the most liberal form of the game gives no longer a regular language.]