Practice Problem 1

- Def: $\texttt{cycle}(s) = \{s_i ... s_{|s|} s_1 ... s_{i-1} \mid 1 \leq i \leq |s|\}$
- Ex: $\texttt{cycle}(123) = \{123, 231, 312\}$.
- Def: $\texttt{cycle}(L) = \cup_{s \in L} \texttt{cycle}(s)$
- Prove or disprove: if L is regular, then $\texttt{cycle}(L)$ is also regular.

Practice Problem 2

- Def: $\texttt{merge}(x, []) = \{x\}$
- Def: $\texttt{merge}([], y) = \{y\}$
- Def: $\texttt{merge}(x : xs, y : ys) =$
  $\{x \circ s | s \in \texttt{merge}(xs, y : ys)\} \cup$
  $\{y \circ s | s \in \texttt{merge}(x : xs, ys)\}$
- Ex: $\texttt{merge}(ab, 12) = \{ab12, a1b2, a12b, 1ab2, 1a2b, 12ab\}$
- Def: $\texttt{merge}(L_1, L_2) = \cup_{s_1 \in L_1, s_2 \in L_2} \texttt{merge}(s_1, s_2)$
- Prove or disprove: if L_1 and L_2 are regular, then $\texttt{merge}(L_1, L_2)$ is also regular.

Practice Problem 3

- Def: `permute` is the permutation function.
- Prove or disprove: if L is regular, then $\texttt{permute}(L)$ is also regular.

Practice Problem 4

- Let $\Sigma = \{0, 1\}$
- Let $n(s)$ = number of 1's in s
- Prove or disprove: if for every predicate p, $L_1 = \{1^n | p(n) = \texttt{true}\}$ is regular iff $L_2 = \{s | p(n(s)) = \texttt{true}\}$ is regular

Practice Problem 5

- Let $\Sigma = \{0, 1\}$
- Let $bin(s)$ = value of s read as binary number
- Prove or disprove: if for every predicate p,
  $L_1 = \{1^n | p(n) = \texttt{true}\}$ is regular iff
  $L_2 = \{s | p(bin(s)) = \texttt{true}\}$ is regular

Practice Problem 6

- A read-twice DFA, on deciding input w, is fed $w \circ w$. (So it gets to read the input twice).
- Ex: to decide the string "Hello", it would be fed "HelloHello"
- Prove or disprove: a read-twice DFA can only recognize regular languages.

Practice Problem 7

- A read-twice-reverse DFA, on deciding input w, is fed $w \circ w^r$, where w^r is w reversed.
- Ex: to decide the string "Hello", it would be fed "HelloolleH"
- Prove or disprove: a read-twice-reverse DFA can only recognize regular languages.