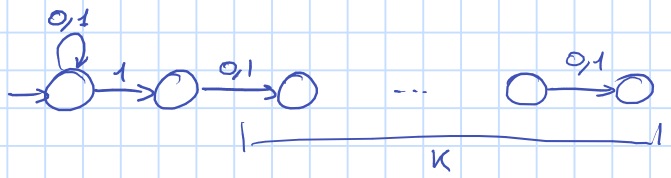
L4. 2022-09-14

Recall DFAs as powerful as NFAs but less efficient.

th There is a language recognizable by a NFA with $K+2$ states that requires $\geq 2^K$ DFA states.

proof. consider the language of strings st $s_{n-k} = 1$.
(i.e. $K+1$-th character from end is a 1).

NFA:



DFA: assume for the sake of contradiction a DFA with $< 2^K$ states exists.

Consider set of 2^K strings $\{1w \mid w \in \{0,1\}^K\}$.

By pigeonhole there exist $s \neq s'$ st $q_s = q_{s'}$. — last state of comp(M, s) / " " (M, s')

Let i be the last position st $s_i \neq s_i'$; assume wlog $s_i = 1$, $s_i' = 0$.

Let $t = s0^{i-1}$, $t' = s'0^{i-1}$.

$t = s_1 \dots s_{i-1} 1 \underbrace{s_{i+1} \dots s_{k+1} 0 \dots 0}_{k}$ $\Rightarrow$ in language.

$t' = s_1' \dots s_{i-1}' 0 \underbrace{s_{i+1}' \dots s_{k+1}' 0 \dots 0}_{k}$ $\Rightarrow$ not in language.

By construction, t and t' end in same state, ie $q_t = q_{t'}$.

But t is in the language and should be accepted, while t' is not and should be rejected. this means state q_t is both accepting and not accepting!!

Hence our assumption is false and the opposite, i.e. every DFA recognizing the language has $\geq 2^K$ states, must be true. ✗

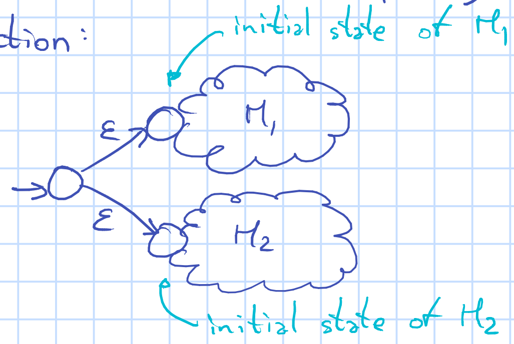
Regular Operations and Languages.

We saw how to build automaton for the complement of a language. Which other operations can we do?

Union: $L_1 \cup L_2 = L_1 | L_2 = \{ s : s \in L_1 \text{ or } s \in L_2 \}$

M_1, M_2 corresponding automata.

Construction:



* Note: $L_1 \cap L_2 = \overline{(\overline{L_1} \cup \overline{L_2})}$ (de Morgan).

Concatenation: $L_1 L_2 = \{ s_1 s_2 : s_1 \in L_1, s_2 \in L_2 \}$

Kleene Star: $L^* = \{ s^k : s \in L, k \in \mathbb{Z}, k \geq 0 \}$
$= \{$any number of repetitions of a string in the language$\}$.