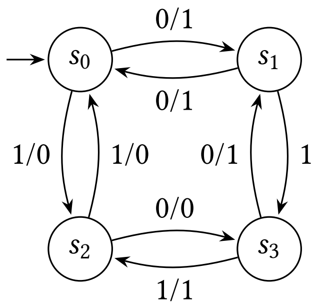
$\mathcal{H}$omework #6 **$\mathcal{D}$iscrete $\mathcal{M}\forall$th**
Automata Theory **Spring 2024**

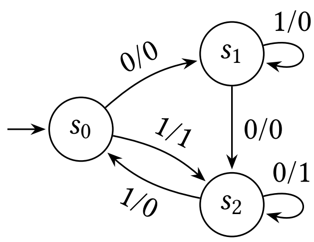
1. For each given regular expression P, construct a DFA (Deterministic Finite Automaton), and find the number of accepted word of length at most 5, i.e. the size of the set $\mathcal{L}' = \{w \in \mathcal{L}(P) \mid |w| \le 5\}$. For "any" (`.`) and "negative" (`[^.]`) matches, assume that the alphabet is $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$.
   (a) $P_1 =$ `ab*`
   (b) $P_2 =$ `a+b?c`
   (c) $P_3 =$ `[^cd]+c{3}`
   (d) $P_4 =$ `[^a](.|ddd)?`
   (e) $P_5 =$ `d(a|bc)*`
   (f) $P_6 =$ `((a|ab)[cd]){2}`

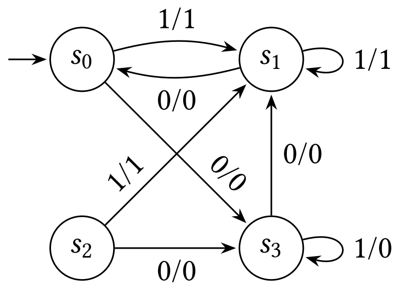
2. Describe the set of strings defined by each of these sets of productions in EBNF (extended Backus-Naur form).
   (a) $\langle \mathit{string} \rangle ::= \langle L \rangle\texttt{+}\ \langle D \rangle\texttt{?}\ \langle L \rangle\texttt{+}$
   $\langle L \rangle ::= \mathtt{a} \mid \mathtt{b} \mid \mathtt{c}$
   $\langle D \rangle ::= \mathtt{0} \mid \mathtt{1}$
   (b) $\langle \mathit{string} \rangle ::= \langle \mathit{sign} \rangle\texttt{?}\ \langle N \rangle$
   $\langle \mathit{sign} \rangle ::= \text{`+'} \mid \text{`-'}$
   $\langle N \rangle ::= \langle D \rangle\ (\langle D \rangle \mid \mathtt{0})^*$
   $\langle D \rangle ::= \mathtt{1} \mid \ldots \mid \mathtt{9}$
   (c) $\langle \mathit{string} \rangle ::= \langle L \rangle^*\ (\langle D \rangle\texttt{+})\texttt{?}\ \langle L \rangle^*$
   $\langle L \rangle ::= \mathtt{x} \mid \mathtt{y}$
   $\langle D \rangle ::= \mathtt{0} \mid \mathtt{1}$
   (d) $\langle \mathit{string} \rangle ::= \langle C \rangle\ \langle R \rangle^*$
   $\langle C \rangle ::= \mathtt{a} \mid \ldots \mid \mathtt{z} \mid \mathtt{A} \mid \ldots \mid \mathtt{Z}$
   $\langle D \rangle ::= \mathtt{0} \mid \ldots \mid \mathtt{9}$
   $\langle R \rangle ::= \langle C \rangle \mid \langle D \rangle \mid \text{`\_'}$

3. Let $\mathcal{G} = \langle V, T, S, P \rangle$ be the phrase-structure grammar with vocabulary $V = \{\mathtt{A}, \mathtt{S}\}$, terminal symbols $T = \{\mathtt{0}, \mathtt{1}\}$, start symbol $S = \mathtt{S}$, and set of productions P: $\mathtt{S} \to \mathtt{1S}$, $\mathtt{S} \to \mathtt{00A}$, $\mathtt{A} \to \mathtt{0A}$, $\mathtt{A} \to \mathtt{0}$.
   (a) Show that `111000` belongs to the language generated by $\mathcal{G}$.
   (b) Show that `11001` does not belong to the language generated by $\mathcal{G}$.
   (c) What is the language generated by $\mathcal{G}$?

4. Find the output generated from the input string `01110` for each of the following Mealy machines.







5. Construct a Moore machine for each of the following descriptions.
   (a) Determine the residue modulo 3 of the input treated as a binary number. For example, for input ε (which corresponds to "value" 0) the residue is 0; `101` (5 in decimal) has residue 2; and `1010` (value 10) has residue 1.
   (b) Output the residue modulo 5 of the input from $\{0, 1, 2\}^*$ treated as a ternary (base 3) number.
   (c) Output A if the binary input ends with `101`; output B if it ends with `110`; otherwise output C.

6. Show that regular languages are *closed* under the following operations.
   (a) Union, that is, if L_1 and L_2 are regular languages, then $L_1 \cup L_2$ is also regular.
   (b) Concatenation, that is, if L_1 and L_2 are regular languages, then $L_1 \cdot L_2$ is also regular.
   (c) Kleene star, that is, if L is a regular language, then L^* is also regular.
   (d) Complement, that is, if L is a regular language, then $\overline{L} = \Sigma^* - L$ is also regular.
   (e) Intersection, that is, if L_1 and L_2 are regular languages, then $L_1 \cap L_2$ is also regular.

7. Determine whether the following languages are regular or not. For non-regular languages, use Pumping lemma to prove that they are not regular. For each regular language, provide a regular expression and construct an ε-NFA.

   (a) $L_1 = \{w \in \{0,1\}^* \mid \text{length of } w \text{ is odd}\}$

   (b) $L_2 = \{0^n 1^n \mid n \in \mathbb{N}\}$

   (c) $L_3 = \{w \in \{0,1\}^* \mid w \text{ contains an even number of 1s}\}$

   (d) $L_4 = \{1^{n^2} \mid n \in \mathbb{N}\}$

8. Consider a finite-state automaton $M = (\Sigma, Q, q_0, F, \delta)$ and a non-negative integer k. Let R_k be the relation on the set of states of M such that $s\ R_k\ t$ if and only if for every input string $w \in \Sigma^*$ with $|w| \le k$, $\delta(s,w)$ and $\delta(t,w)$ are both final states or both not final states. Furthermore, let R^* be the relation on the set of states of M such that $s\ R^*\ t$ if and only if for every input string $w \in \Sigma^*$, regardless of length, $\delta(s,w)$ and $\delta(t,w)$ are both final states or both not final states.

   (a) Show that for every nonnegative integer k, R_k is an equivalence relation on S.
   Two states s and t are called k-equivalent if $s\ R_k\ t$.

   (b) Show that R^* is an equivalence relation on S.
   Two states s and t are called $*$-equivalent if $s\ R^*\ t$.

   (c) Show that if two states s and t are k-equivalent ($k > 0$), then they are also $(k-1)$-equivalent.

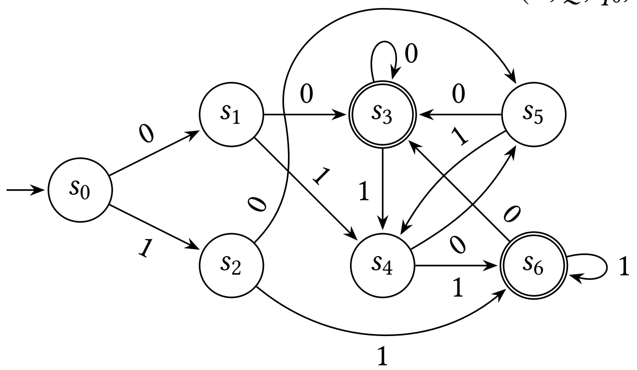
   (d) Show that the equivalence classes of R_k are a *refinement* of the equivalence classes of R_{k-1}.

   (e) Show that if two states s and t are k-equivalent for every non-negative integer k, then they are $*$-equivalent.

   (f) Show that all states in a given R^*-equivalence class are final or all are not final.

   (g) Show that if two states s and t are $*$-equivalent, then $\delta(s,a)$ and $\delta(t,a)$ are also $*$-equivalent for all $a \in \Sigma$.

9. Consider the finite-state automaton $M = (\Sigma, Q, q_0, F, \delta)$ depicted below.



   (a) Find the k-equivalence classes of M for $k = 0, 1, 2, 3$.

   (b) Find the $*$-equivalence classes of M.

   (c) Construct the quotient automaton $\overline{M}$ of M.
   ▹ The quotient automaton $\overline{M}$ of the deterministic finite-state automaton $M = (\Sigma, S, s_0, F, \delta)$ is the finite state automaton $\overline{M} = (\Sigma, \overline{S}, [s_0]_{R^*}, \overline{F}, \overline{\delta})$, where the set of states $\overline{S}$ is the set of R^*-equivalence classes of S; the transition function $\overline{\delta}$ is defined by $\overline{\delta}([s]_{R^*}, a) = [\delta(s,a)]_{R^*}$ for all states $[s]_{R^*}$ of $\overline{M}$ and input symbols $a \in \Sigma$; and $\overline{F}$ is the set consiting of R^*-equivalence classes of final states of M.

10. Solve the following regex crosswords. Fill each cell with a single ASCII character (an uppercase letter, a digit, a punctuation mark, or a space). Each row/column, when read left to right or top to bottom must match the regular expression(s) given for that row/column.

<table>
<tr><th></th><th>[^SPEAK]+</th><th>EP|IP|EF</th></tr>
<tr><th>HE|LL|O+</th><td></td><td></td></tr>
<tr><th>[PLEASE]+</th><td></td><td></td></tr>
</table>

<table>
<tr><th></th><th>(FI|A)+</th><th>(YE|OT)K</th><th>(.)[IF]+</th><th>[NODE]+</th><th>(FY|F|RG)+</th></tr>
<tr><th>(Y|F)(.)\2[DAF]\1</th><td></td><td></td><td></td><td></td><td></td></tr>
<tr><th>(U|O|I)*T[FRO]+</th><td></td><td></td><td></td><td></td><td></td></tr>
<tr><th>[KANE]*[GIN]*</th><td></td><td></td><td></td><td></td><td></td></tr>
</table>

<table>
<tr><th></th><th>[^NRU](NO|ON)</th><th>(D|FU|UF)+</th><th>(FO|A|R)*</th><th>(N|A)*</th></tr>
<tr><th>[RUNT]*</th><td></td><td></td><td></td><td></td></tr>
<tr><th>O.*[HAT]</th><td></td><td></td><td></td><td></td></tr>
<tr><th>(.)*DO\1</th><td></td><td></td><td></td><td></td></tr>
</table>

<table>
<tr><th></th><th>[^MCI]+</th><th>.A</th><th>(TM|BF)</th><th></th></tr>
<tr><th>(CAT|A-T)+</th><td></td><td></td><td></td><th>[^KI\sP]+</th></tr>
<tr><th>[MA\-\sE]+</th><td></td><td></td><td></td><th>(M|APS|EA)*</th></tr>
<tr><th></th><th>[AI][E\s]</th><th>[A\-Z]+</th><th>[\sT\-M]+</th><th></th></tr>
</table>

<table>
<tr><th></th><th>(.)\1(.)\2</th><th>[C\sOU]+</th><th>[^PU\sH]+</th><th></th></tr>
<tr><th>.[LUH]+</th><td></td><td></td><td></td><th>.*L+</th></tr>
<tr><th>(P|K)[^U]+</th><td></td><td></td><td></td><th>[PUF\s]*</th></tr>
<tr><th>.*C+[TIF]</th><td></td><td></td><td></td><th>[TIC]*</th></tr>
<tr><th>(NO|ONE|ION)*</th><td></td><td></td><td></td><th>[NOI\sE]+</th></tr>
<tr><th></th><th>[PIF]+</th><th>.*[OWE]*</th><th>(TN|LF|TF)*</th><th></th></tr>
</table>

<table>
<tr><th></th><th>.*E.*</th><th>[^P]I(IT|ME)</th><th>(EM|FE)(IT|IP)</th><th>(TS|PE|KE)*</th></tr>
<tr><th>(EP|ST)*</th><td></td><td></td><td></td><td></td></tr>
<tr><th>T[A-Z]*</th><td></td><td></td><td></td><td></td></tr>
<tr><th>.M.T</th><td></td><td></td><td></td><td></td></tr>
<tr><th>.*P.[S-X]+</th><td></td><td></td><td></td><td></td></tr>
</table>