9/14/18

Midterm next Friday

List of topics will be sent today

HW 3 will be assigned Monday

Extra Office hours next Thurs. afternoon (probably 2:30-?)

I will announce preliminary grade cutoffs for the exam in advance.

NFAs and (right)-regular grammars.

Comments: One of the HW problems is to show that the reverse of a ~~Re~~ reg. lang. is a reg. lang.

$\Downarrow$

left-regular grammars give regular language (b/c ~~a te~~ if you take all the prod. rules of a left reg. gram. G and reverse them, you get a right. reg. gram. that produces the reverse language (to the one produced by G.)

Convert a right regular grammar to an NFA

S = start

$V = \{S, A, B, C, D\}$

$\Sigma = \{0, 1\}$

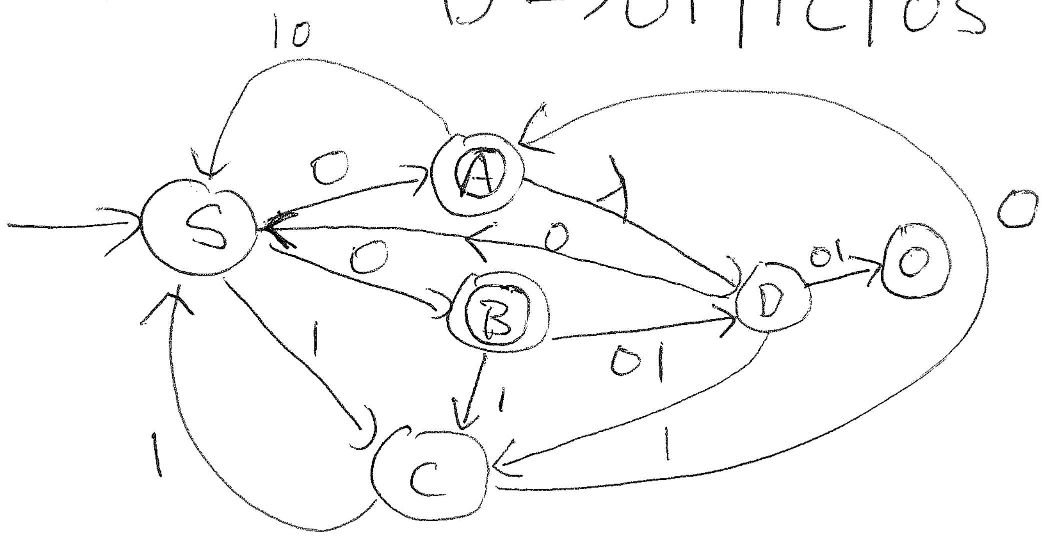
$S \rightarrow 0A \mid 0B \mid 1C$

$A \rightarrow 10S \mid D \mid \lambda$

$B \rightarrow 1C \mid 01D \mid \lambda$

$C \rightarrow 1S \mid 0A$

$D \rightarrow 01 \mid 1C \mid 0S$



Ways to build new regular languages from known to be regular languages.

Suppose L_1 and L_2 are regular languages.

1) The concatenation $L_1 L_2$ is a reg. lang.

2) The union $L_1 \cup L_2$ is regular.

3) The Kleene star $(L_1)^*$ is regular.

4) The complement $\overline{L_1} = \Sigma^* \setminus L_1$ (i.e. every string not in L_1) is regular.

<u>Pf</u>: If I have a <u>DFA</u> for L_1, I can construct a DFA for $\overline{L_1}$ by making all final states non-final states and vice versa.

5) The intersection $L_1 \cap L_2 = \{ w \in \Sigma^*$ in both L_1 and $L_2 \}$ is regular.

Pf: If $M_1 = (Q_1, \Sigma, \delta_1, s_1, F_1)$ is a DFA for L_1, and $M_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$ is a DFA for L_2, construct a DFA $M = (Q, \Sigma, \delta, s, F)$ by:

$$Q = Q_1 \times Q_2 = \{ \text{pairs } (q_1, q_2),\ q_1 \in Q_1,\ q_2 \in Q_2 \}$$

δ is defined by

$$\delta((q_1, q_2), \ell) = (\delta_1(q_1, \ell), \delta_2(q_2, \ell))$$

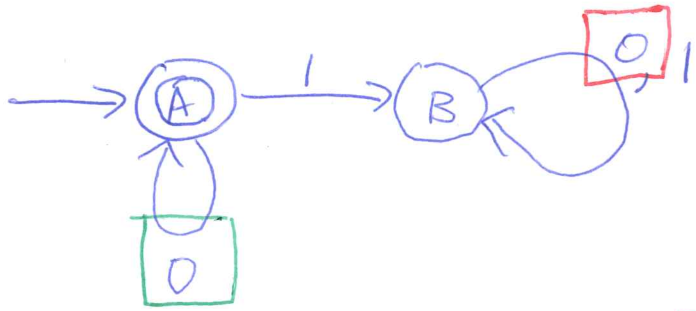
for all $q_1 \in Q_1$, $q_2 \in Q_2$, $\ell \in \Sigma$.

$s = (s_1, s_2)$.

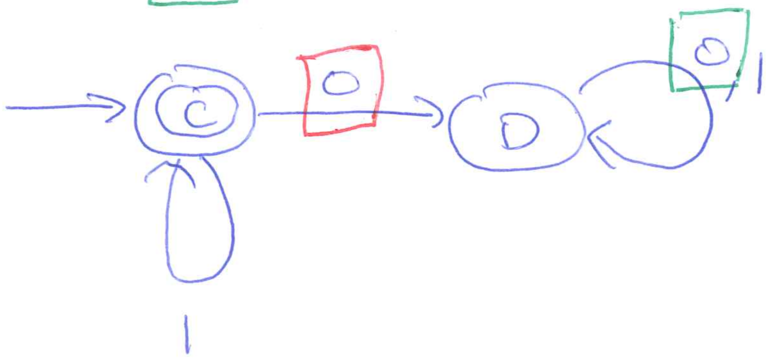
$$F = F_1 \times F_2 = \{ (q_1, q_2) \in Q_1 \times Q_2 \mid q_1 \in F_1,\ q_2 \in F_2 \}$$

Silly example:

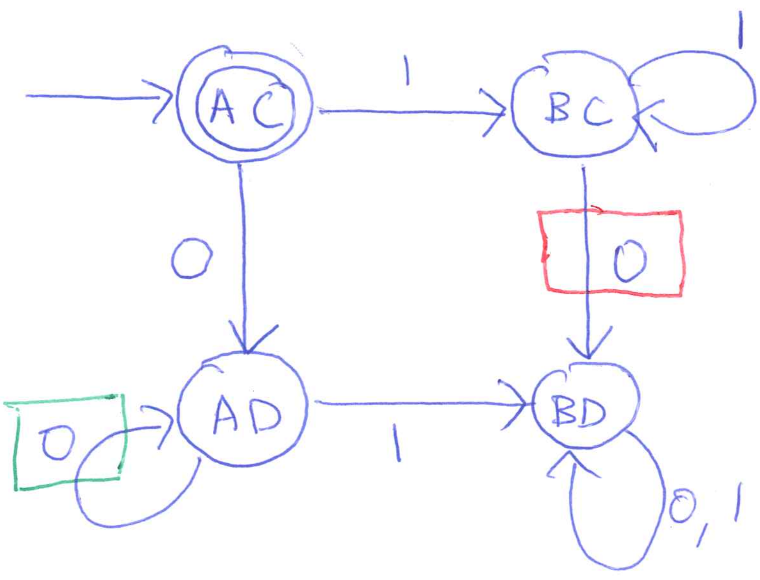
$M_1 =$



$M_2 =$



$M =$



You can combine these rules: e.g., if L_1, L_2, L_3 are regular, then

$$\left[(L_1 \cup L_2) \cap \overline{L_3}\right] \cup \left[L_3 \cap \overline{L_2}\right]$$ is also regular.

(in fact, any Venn diagram region is regular).

<u>Note</u>: There are many ways to create a new regular language out of known to be regular languages. (Some are on HW)

If we have 2 alphabets Σ, Γ, and a function $f: \Sigma \to \Gamma^*$, this automatically gives a function $f^*: \Sigma^* \to \Gamma^*$. Then, if $L \subseteq \Sigma^*$ is regular, $f^*(L)$ is regular.

Pf: Take an NFA for L, and replace ~~all the Σ^* labels by f^*~~ the transition (q, w, q') by $(q, f^*(w), q')$.

E.g. L is a reg. lang. on $\Sigma = \{a, b, c\}$

$L' = \{w \in \{a,b\}^*$ s.t. you obtain w by removing all c's from some string in $L\}$

L' is regular by the function

$$f: \begin{array}{l} a \longmapsto a \\ b \longmapsto b \\ c \longmapsto \lambda \end{array}$$

This removes all the c's