CSE596 Lecture Wed Sep 6 Fall 2017

Σ : any alphabet.

*: "zero or more"

Σ^* = the set of all strings over Σ.

$\Sigma = \{a, b\}$

Σ^2 $\Sigma^2 = \{aa, ab, ba, bb\}$

$\|\Sigma^3\| = 8$

$\Sigma^1 = \{\text{"a"}, \text{"b"}\}$

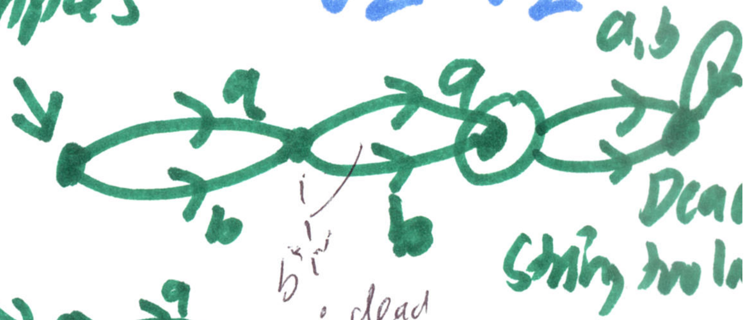
$\Sigma^0 = \{\varepsilon\}$.

$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \dots$

<u>Def^n</u>: A language $A \subseteq \Sigma^*$ is <u>regular</u> if there is a DFA $M = (Q, \Sigma, \delta, s, F)$ such that $\underline{A = L(M)}$.

Examples

Σ^2 is regular: M:



Σ^* is regular: M =

$\Sigma^0 = \{\varepsilon\}$ is regular: M =

complement

<u>Theorem</u>: If A is regular then so is its <u>complement</u>

$\tilde{A} = \Sigma^* \setminus A = \{x \in \Sigma^* : x \notin A\}$.

$A = L(M)$

<u>Proof</u>: Take $M = (Q, \Sigma, \delta, s, F)$ from the defn "A is regul

Now build $M' = (Q, \Sigma, \delta, s, \underline{Q \setminus F})$. This is a DFA, a

$L(M') = \{x : M$ does <u>not</u> accept $x\} = \{x : x \notin A\} = \tilde{A}$. $\boxtimes$

Theorem: If A and B a regular languages, then $A \cap B$ is also a regular language.

Proof: Take DFAs M_A, M_B s.t. $L(M_A) = A$ and $L(M_B) = B$. We have $M_A = (Q_A, \Sigma, \delta_A, s_A, F_A)$

(Σ is the same: A and B are over the same alphabet) $M_B = (Q_B, \Sigma, \delta_B, s_B, F_B)$

with $C = A \cap B$.

Goal: Build $M_C = (Q_C, \Sigma, \delta_C, s_C, F_C)$ s.t. $L(M_C) = ($

ordered pair<State, State> stateC;

Define $Q_C = Q_A \times Q_B =_{def} \{(p, q) : p \in Q_A \text{ and } q \in Q_B\}$

$s_C = (s_A, s_B)$ If we view δ as a function:

$$\delta_C((p,q), a) = \big(\delta_A(p, a), \delta_B(q, a)\big) \qquad \delta_C : Q_C \times \Sigma \to Q_C$$

Another way: If (p, a, r) is an instruction of δ_A viewed as a set of triples in M_A, and (q, a, r') is similar in M_B, then M_C has the tandem instruction $((p,q), a, (r, r'))$. Finally, build

$$F_C = \{(p, q) : p \in F_A \text{ AND } q \in F_B\}.$$

Then $L(M_C) = C$!

Suppose $D = A \cup B$. Then $D = \sim(\tilde{A} \cap \tilde{B})$. But direct: let M_D have $F_D = \{(p, q) : p \in F_A \text{ OR } q \in F_B\}$

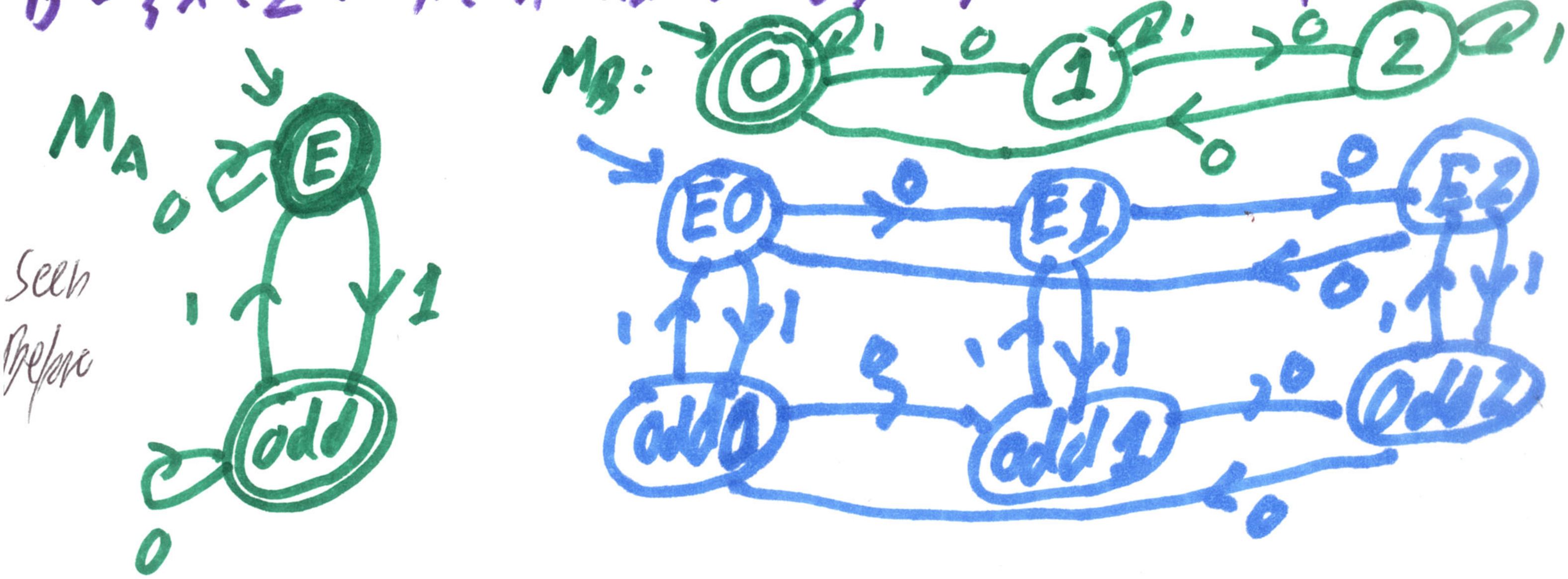
What about $E = A \triangle B = (A \setminus B) \cup (B \setminus A)$? Use $\underline{X}$

(3) (Extra)

Added: Example. $\Sigma = \{0, 1\}$

$A = \{x \in \Sigma^*:$ the number of 1s in x is odd$\}$.

$B = \{x \in \Sigma^*:$ the number of 0s in x is a multiple of $3\}$



For $A \cap B$, **odd 0** is the only accepting state: $F_C = \{odd\ 0\}$

For $A \cup B$, $F_D = \{odd0, odd1, odd2, E0\}$

For $E = A \triangle B$, $F_E = F_C \triangle F_D = \{odd1, odd2, E0\}$.

Usually the tandem machine doesn't come out so "geometrically nice." The "Turing Kit" program was intended to have menu actions for taking any two user-designed DFAs M_A and M_B and computing the machine M_C. The sticking point was: How to *draw* M_C in a nice way? This leads in to the major area of *Graph Drawing*. We've never found Java code for it.

To answer a question from class, suppose we intersect three or more machines. Hey, for any (prime) number p it is easy to build M_p with p states so that $L(M_p) = \{x \in \Sigma^*:$ the number of 2s in x is a multiple of $p\}$. Consider $L(M_2) \cap L(M_3) \cap L(M_5) \cap L(M_7) \cap L(M$ A DFA M_E for the whole must have $2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 = 2{,}310$ states! This gets big quickly...