F.S.M. Σ = input chars

S = finite set of states

$s_0 \in S$

$F \subseteq S$

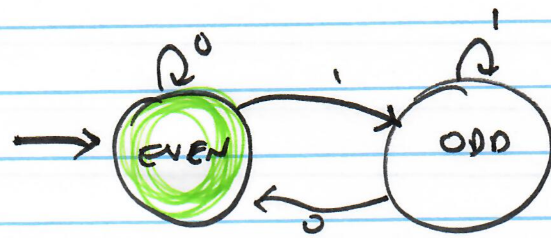
$\delta : \Sigma \times S \to S$

set of all subsets of S

$P(S)$ 2^S

$S = \{1,2,3\}$

$P(S) = \{\phi, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\}$



F.S.M $\Rightarrow$ language (set of strings)

"languages" that can be recognized by F.S.M.

$\rightarrow$ REGULAR LANGUAGES

1) Any finite set of strings is regular

2) Set $\{\epsilon\}$ is regular (ϵ = "empty string")

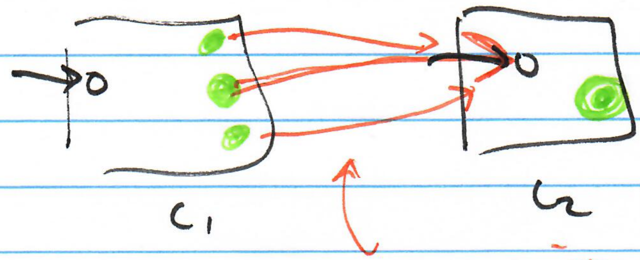
CLOSURE THM

3) L_1, L_2 regular $\Rightarrow$ $L_1 \cup L_2$ is regular

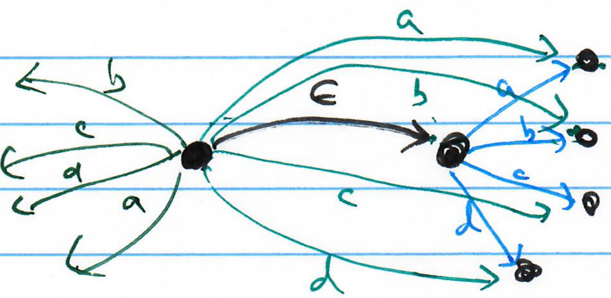
$$\left.\begin{array}{l} M_1 = (\Sigma, S_1, s_0 \in S_1, F_1 \subseteq S_1, \delta_1) \\ M_2 = (\Sigma, S_2, t_0 \in S_2, F_2 \subseteq S_2, \delta_2) \end{array}\right\} (\Sigma, S_1 \times S_2, (s_0, t_0), F_1 \times S_2 \cup S_1 \times F_2, \delta$$

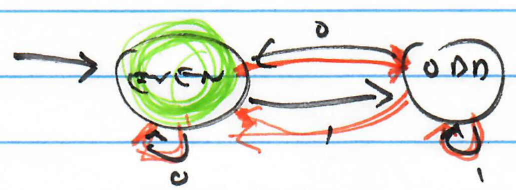
4) $\Rightarrow$ $L_1 \cap L_2$ is regular $F_1 \times F_2$

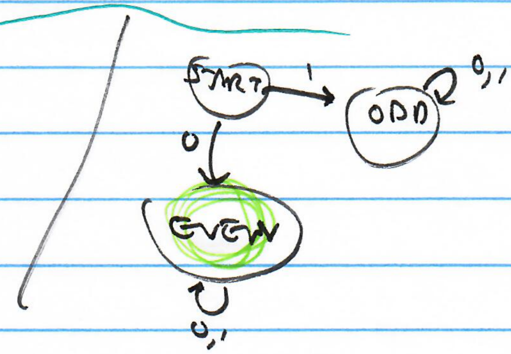
4) Concatenation $\left.\begin{matrix} L_1 \\ L_2 \end{matrix}\right\}$ regular $\Rightarrow L_1 L_2$



ε- transitions / no additional power

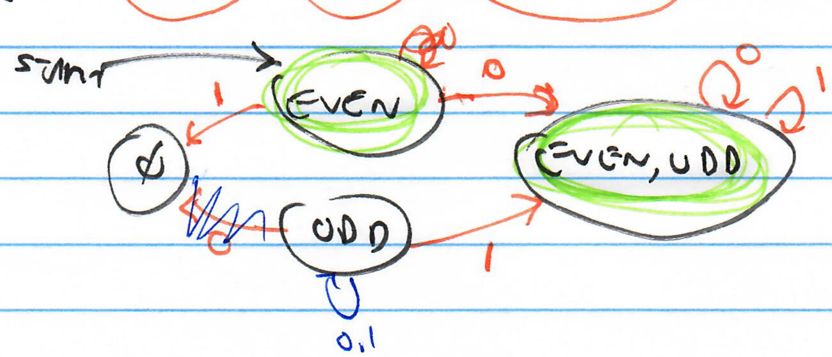






$S = \{EVEN, ODD\}$

$P(S) = \{\emptyset, \{EVEN\}, \{ODD\}, \{EVEN, ODD\}\}$



5) If L is regular, then $L^{REVERSE}$ is regular

6) Kleene closure L

L^*

L L L L L $\quad L^5$

$L^0 = \{\epsilon\}$

L

$LL \quad L^2$

$LLL \quad L^3$

$LLLL \quad L^4$

$\vdots$

$L^0 \cup L \cup L^2 \cup L^3 \cup L^4 \cup \dots = L^*$

$\{0,1\}^* \quad \{0\}^*$

$(0+1)^* \quad 0^*$ — REGULAR LANGUAGE

REGULAR EXPRESSIONS

(,)

+

*

$0(0+1)^*1$

$(00+01+10+11)^*$ = all even length strings

$(0+1)(00+01+10+11)^*$ = all odd length strings

$(0+1)^*0$

$\cup \; \cdot \; \cup \; {}^{*}$

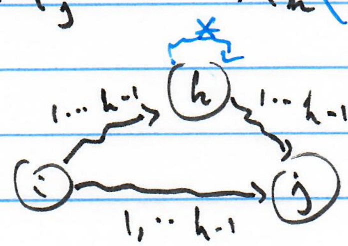
REGULAR EXPRESSIONS $\longleftrightarrow$ LANGUAGES RECOGNIZED BY F.S.M.

$$\boxed{\text{F.S.M} \longrightarrow \text{R.E.}}$$

R_{ij}^{k} = regular expression for the set of strings that take F.S.M. from state i to state j using only intermediate states $\{1, 2, \dots k\}$

$R_{ij}^{0} \longrightarrow$ regular

$$R_{ij}^{k} = R_{ij}^{k-1} + R_{ik}^{k-1}\left(R_{kk}^{k-1}\right)^{*} R_{kj}^{k-1}$$



n states

R_{ij}^{n} $\qquad R^{n}_{0,f_1} + R^{n}_{0,f_2} + \dots$

$\{\varepsilon, 01, 0011, 000111, 00001111, \dots\} = \{0^n 1^n \mid n \geq 0\}$

we add a new starting state to the new machine, and add epsilon transitions from this new state to each of the final states of M.

Note that reversing the directions of transitions might mean that we have two transitions on the same symbol coming out of a state, so that the new machine needs to be nondeterministic.

Now, as an exercise, show that L^* is regular when L is by connecting final states to the start state with an epsilon move.

3.1 An example of non-regular language

Consider the language L=$\{0^n1^n|n = 0, 1, 2, \cdots\}$. This is not a regular language. If it was, we should have an finite state automaton which accepts L. Suppose this automaton has N states, then if we input a string 0^i1^i of length $2i, i > N$, by the pigeon hole principle, when we move around and reach the first "1", we move i times and must hit a certain state at least twice. So there will be a loop starting and ending at this state. Cutting the substring which labels the loop or repeating this substring for any times will give us another string which is also accepted by the automaton. But the string can't satify that we have the same number of "0"s and "1"s!

You can find a more detailed proof in Rosen. The same analysis gives us

pumping lemma: Let L be a regular set. Then there is a constant n such that if z is any word in L, and $|z| \geq n$, we may write z = uvw in such a way that $|uv| \leq n$, $|v| \geq 1$, and for all $i \geq 0$, uv^iw is in L.

Pumping lemma is a very useful tool to prove that a language is *not* regular.