SEPTEMBER 14, 20

Class Notes

HW # 4

$A \cdot B = \{x \mid \underbrace{x \notin A}_{C}$ and $\cap$ $\underbrace{\text{not } x \text{ not contain string in } B \text{ as substring}}_{D}$.

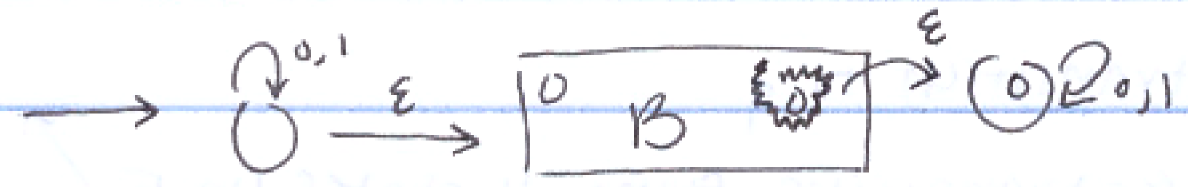
$A \cdot B = C \cap D$

$= \overline{(\bar{C} \cup \bar{D})}$

$\Rightarrow$ $\bar{C}$ is regular

$\bar{D}$ is regular $\Rightarrow$ x do contain substrings of B

x = ..εbε...?



$\boxed{\text{LAST CLASS}}$ REGULAR LANGUAGE

Def: R is a RE

① $R = \varepsilon$ ④ $R = R_1 \cup R_2$

② $R = \emptyset$ ⑤ $R = R_1 \circ R_2$ $(R_1 R_2)$

③ $R = a \in \Sigma$ ⑥ $R = R_1^*$

example: $0\Sigma^*0 \cup 1\Sigma^*1 \cup 0 \cup 1 = \{x \mid x$ starts and ends with the same symbol$\}$

① $(0 \cup \varepsilon) \cdot (1 \cup \varepsilon) = \{\varepsilon, 1, 0, 01\}$

② $1^* \circ \emptyset = \emptyset$

$A \cdot B = \{xy \mid x \in A \text{ and } y \in B\}$

③ $\emptyset^* = \{\varepsilon\}$

$A^* = \{x_1 \ldots x_k \mid k \geq 0 \ \& \ x_i \in A\}$

NOTE: $R \cup \emptyset = R$

$R \cup \{\varepsilon\} \neq R$ in general

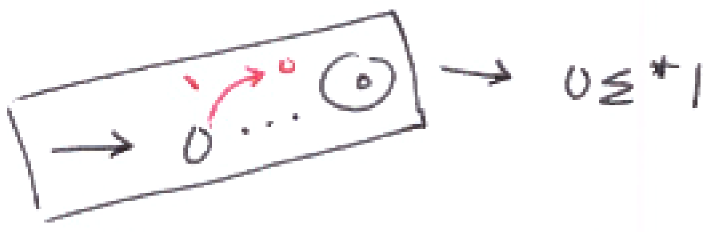
$R \cdot \emptyset = \emptyset \neq R$ in general

$R \circ \{\varepsilon\} = R$

regular

DFA/NFA/RE

Theorem 1.54 A language is regular iff some RE describes it



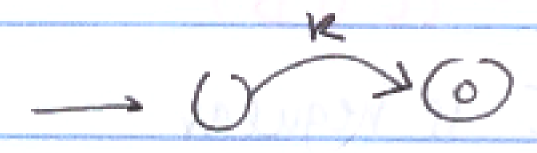
Lemma 1.60 If a language is regular, it has a RE

Pf/ Assume L is a regular language with DFA $D = (Q, \Sigma, \delta, q, F)$

We construct a R ∈ R for L, i.e. $L(R) = L(D) = L$

Idea: Step 1: convert D into "generalized NFA" (GNFA), G, whose transitions are labeled by regular expressions (REs)

Step 2: Recursively reduce size of G until only



is left

Then R is the desired RE.

STEP 1: FORMALLY

(1) Add a start state, q_{start}, with ε-transition to q

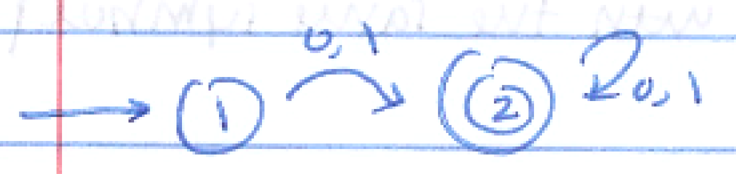
(2) Add a new accepting end state, q_{end}, with ε-transitions from all states in F

Make F non-accepting.

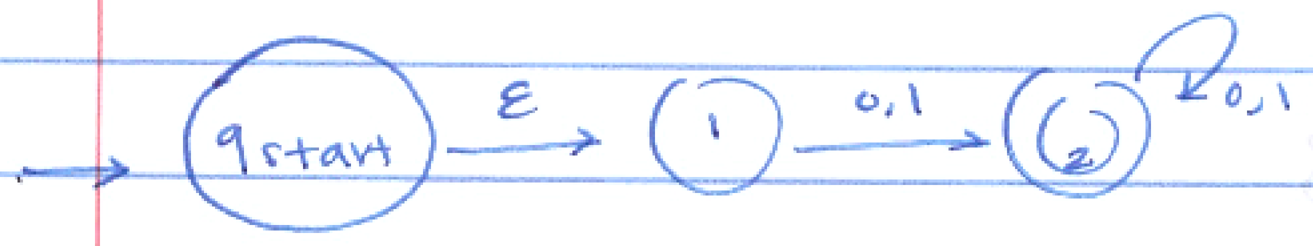
(3) For each transition in the machine D with multiple labels, replace the labels with their union

(4) Between states with no transition, add ∅-transition (except not into q_{start} & out of q_{end})

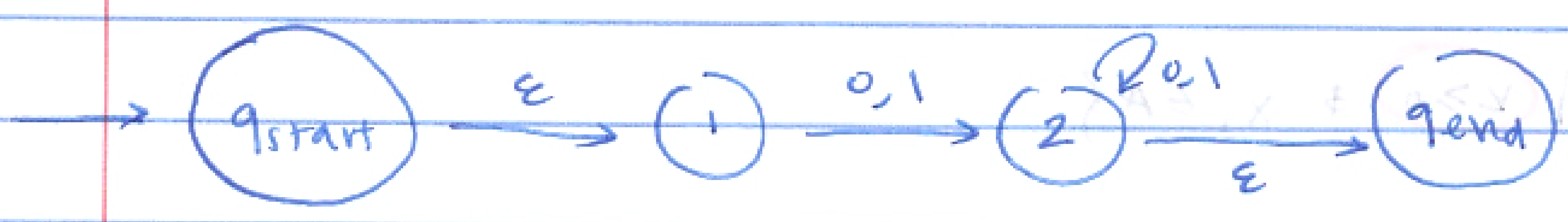
Example DFA D.



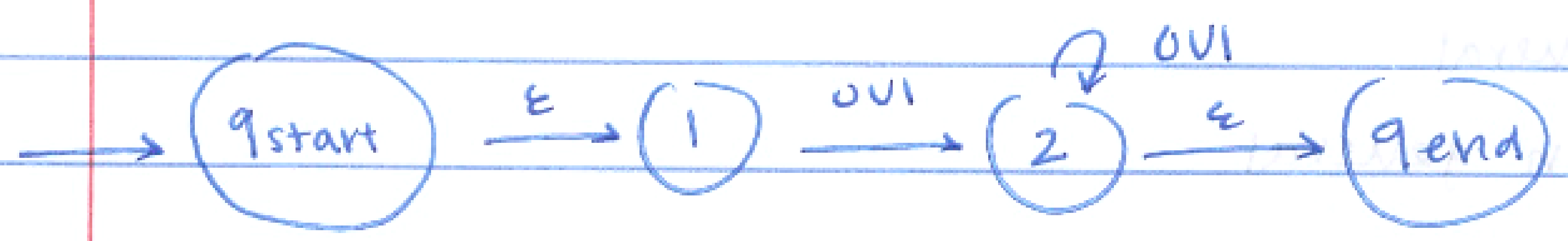
(1)



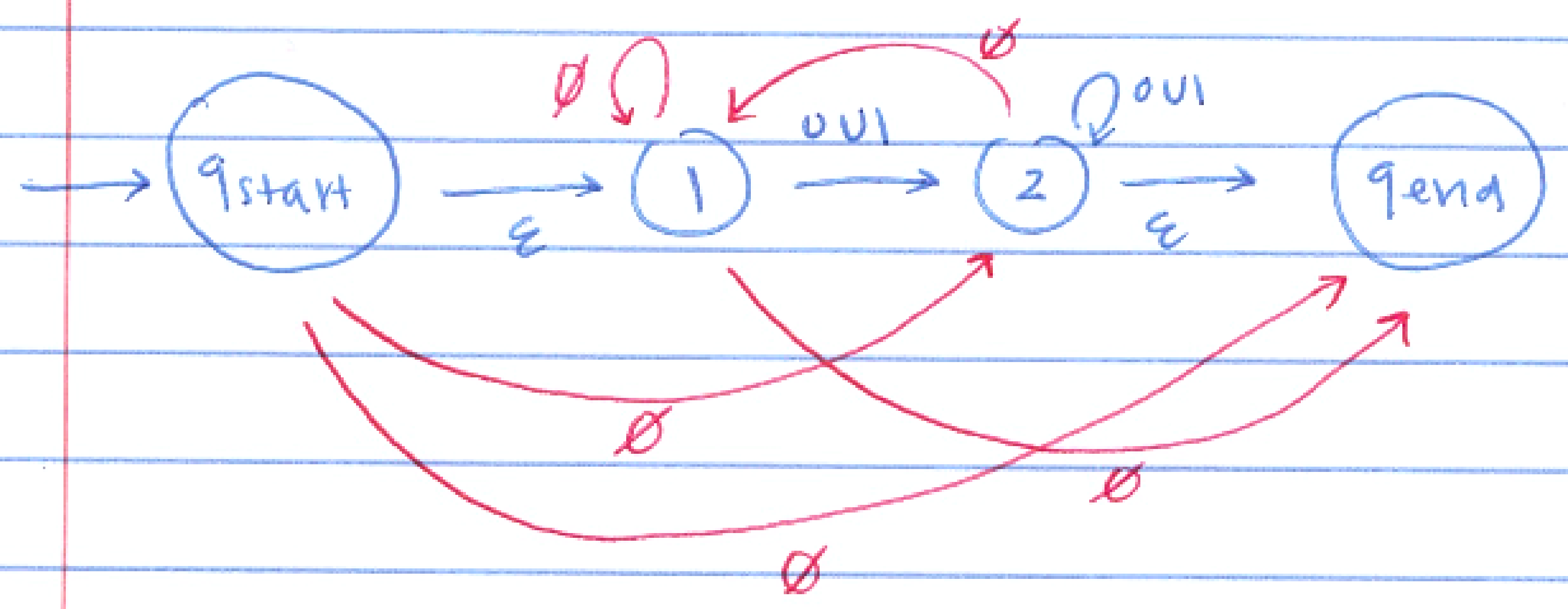
(2)



(3)



(4)



GOAL: How to reduce # states to 2? → 

Step 2: Formally

convert (G): a) let k be # of states in G

GNFA

b) if k=2, return RE on transition from q_{start} to q_{end}.

c) Else: pick any $q_{rip} \in Q$ s.t. $q_{rip} \notin \{q_{start}, q_{end}\}$

Remove q_{rip} from Q

For all pairs of states $q_i \in Q \setminus \{q_{rip}, q_{end}\}$ and $q_j \in Q \setminus \{q_{rip}, q_{start}\}$, set

self loop

$\delta(q_i, q_j) = (R_1 \circ R_2^* \circ R_3) \cup R_4$

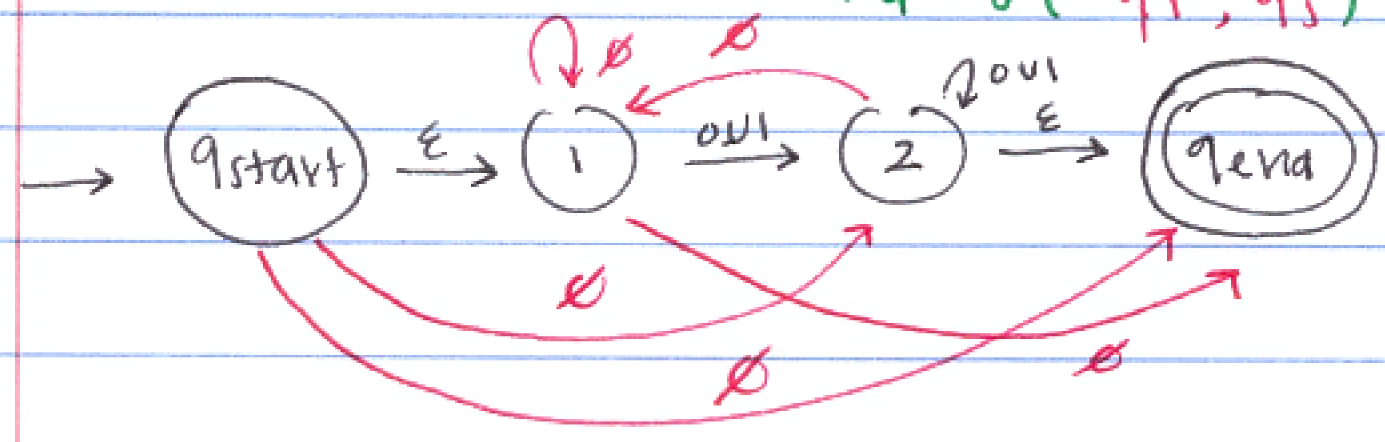
$R_1 = \delta(q_i, q_{rip})$

$R_2 = \delta(q_{rip}, q_{rip})$

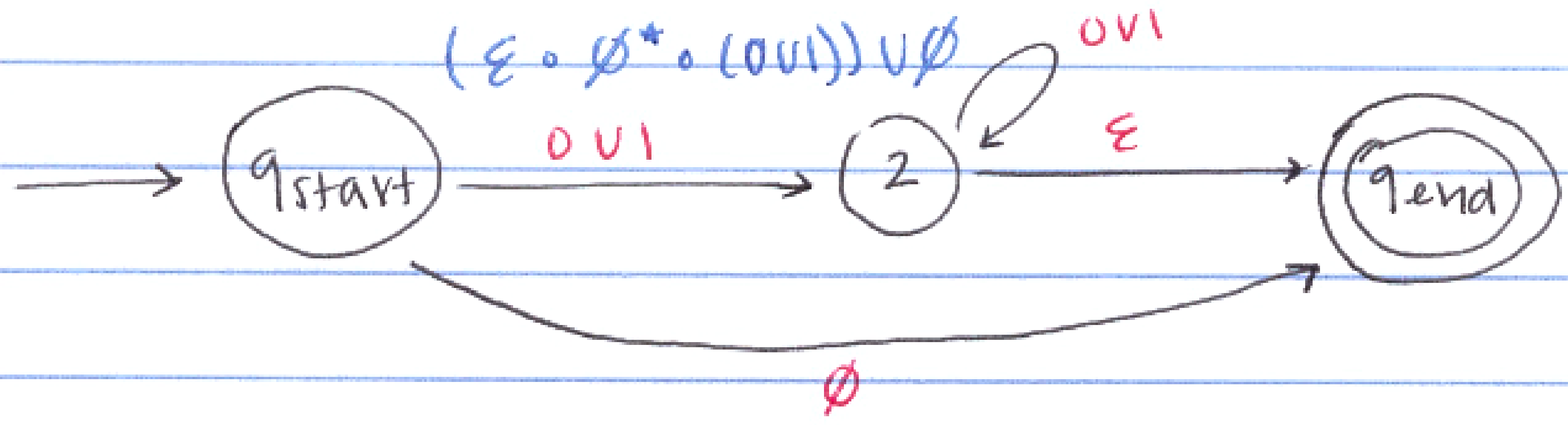
$R_3 = \delta(q_{rip}, q_j)$

$R_4 = \delta(q_i, q_j)$

example



let q_{rip} = 1



let q_{rip} = 2