

I. FA — DFA, NFA, ... , FA
 II. CFG — RE, RG
 III. TM ... — Pumping lemma

TPDA
 CFG
 non-CF

Regular expression (정규식) over Σ

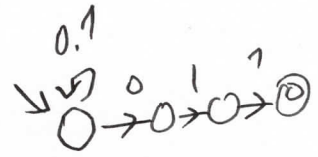
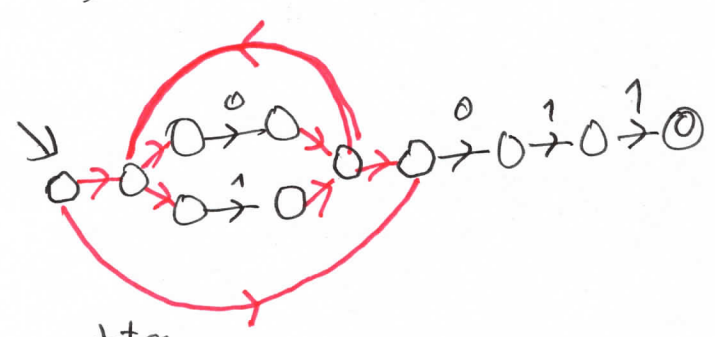
$R \rightarrow R+R \mid R \cdot R \mid R^* \mid (R) \mid \emptyset \mid \epsilon \mid a$
 recursion Basis

→ Context-free grammar
 ... 모호한 (ambiguous) 문법

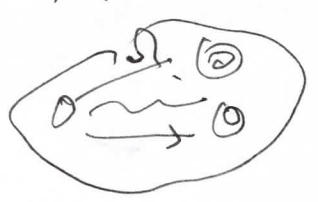
$RE \Rightarrow FA, FA \Rightarrow RE$

$\therefore FA \cong RE$

$FA \Rightarrow RE$
 $(0|1)^*011$



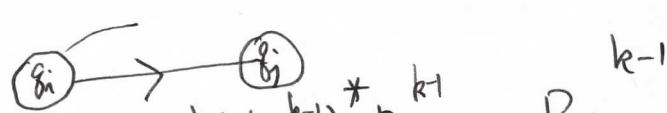
$RE \Rightarrow FA$ — n states



$R_{ij} \in O(n^2)$
 $1 \leq i, j \leq n$
 $R_{ij}^k \in O(n^3)$
 $0 \leq k \leq n$

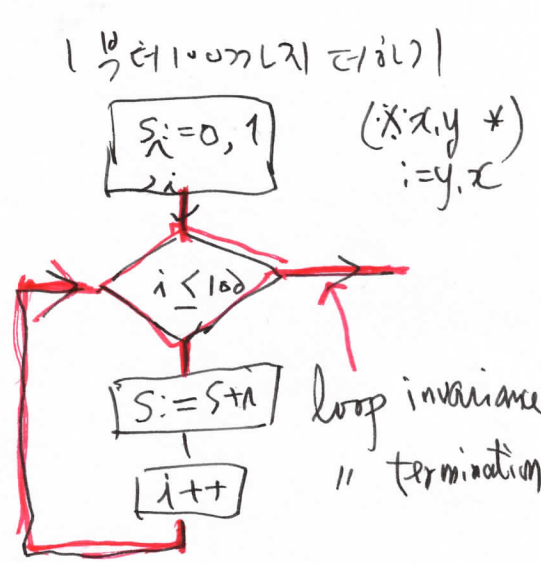
(finite) state iteration

induction on k
 basis k=0



$R_{ij}^k = R_{ik} R_{kj} + R_{ij}^{k-1}$

바보의 두뇌법 — 귀찮
 — 쉽다, 느리다
 똑똑한 두뇌법 — (귀찮)
 — 빠르다



왜 loop 인가?
 같은 일을
 (한번만)
 recursion (induction)

변입방정식은 이항식 R.E 구하기

n-states

n- λ

- Gauss Jordan 재귀법

Upper(Lower) Triangular

* Diagonal

~~(1-k) / (1-k)~~ vs $\frac{1}{1-k}$

$$A = \alpha A + \beta$$

~~\Leftrightarrow~~ $A = \frac{\beta}{1-\alpha}$

\Leftrightarrow $A = (1 + \alpha + \alpha^2 + \dots) \beta$

~~\Leftrightarrow~~ $A = \alpha^* \beta$

