Modeling Computation

Introduction to Formal Languages and Automata

Regular Expressions

Regular Expressions

- Formal specifications of finite automata.
- Any language that is accepted by a finite automaton can be described by a regular expression
- Such languages are called Regular Languages
- NFAs and DFAs accept regular languages.

Operations on Languages (languages are sets)

- Union: $L_1 \cup L_2$
- Intersection: $L_1 \cap L_2$
- Complement $\overline{L}$
- Difference $L_1 - L_2$
- Concatenation: $L_1 L_2$
- Kleene Star : L^*

Concatenation L_1L_2 of languages L_1, L_2 is $\{uv \mid u \in L_1, v \in L_2\}$

Ex: $L_1 = \{a^n \mid n > 0\}$

$L_2 = \{b^m \mid m > 0\}$

$L_1L_2 = \{a^n b^m \mid n > 0, m > 0\}$

L^n denotes L concatenated with itself n times. $L^3 = L(LL)$

Kleene Star L^* of language L is the infinite union

$$\{\varepsilon\} \cup L \cup L^2 \cup L^3 \cup \dots = \bigcup_{i=0}^{\infty} L^i$$

Ex: $L = \{a\}$

$$L^* = \{\varepsilon, a, aa, aaa, \dots\} = \{a^n \mid n \geq 0\}$$

Ex: $L = \{00, 1a\}$

$$L^* = \{\varepsilon, \overbrace{00, 1a}^{L}, \overbrace{0000, 001a, 1a00, 1a1a}^{L^2}, 000000, 00001a, 001a00, 001a1a, \dots$$

Theorem: If languages L and M are accepted by finite automata, then so are:

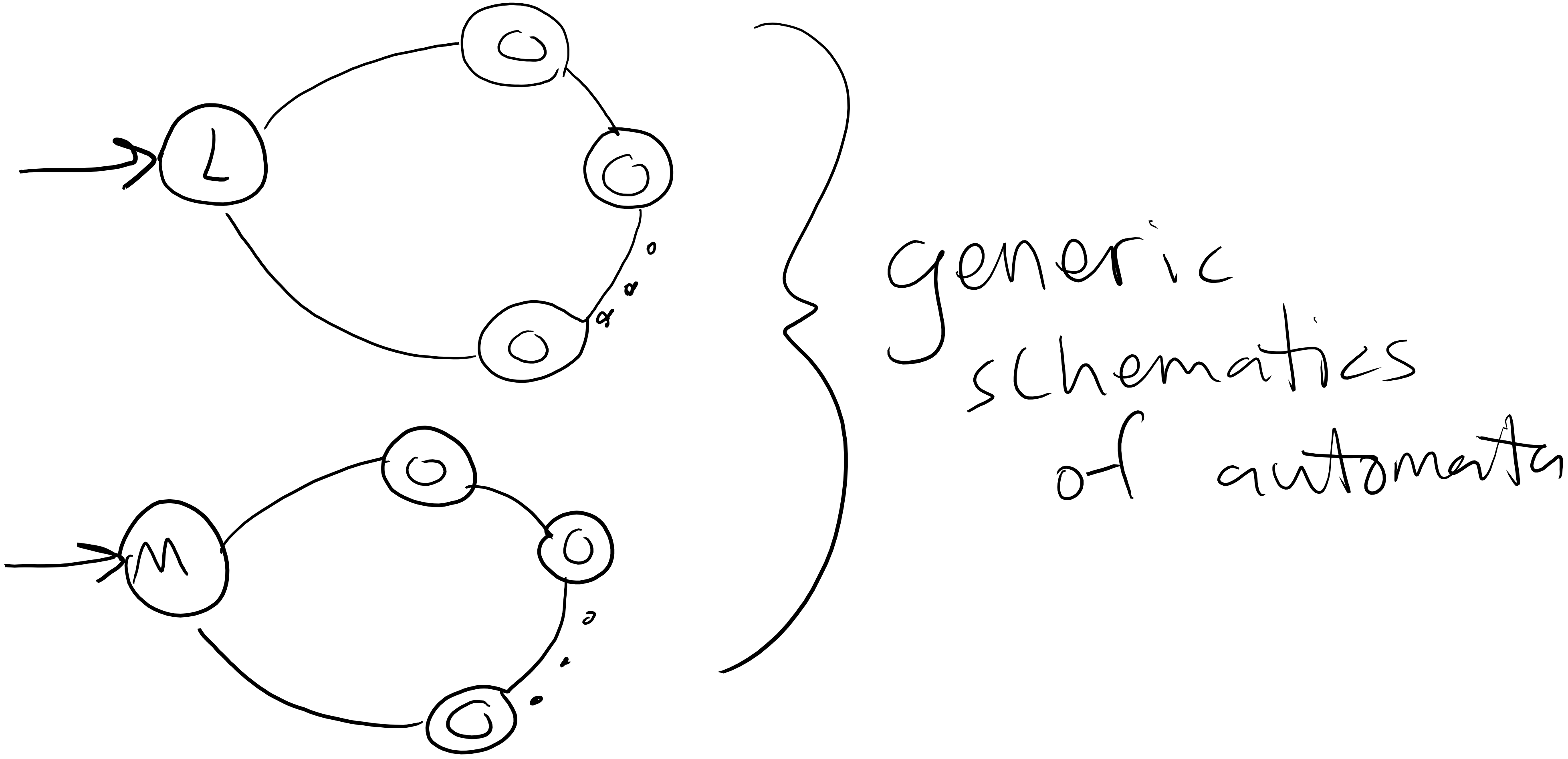
- $L \cup M$ union
- $L \cap M$ intersection
- $\overline{L}$ complement $(= \Sigma^* - L)$
- $L - M$ difference
- LM concatenation
- L^* Kleene Star

Proof (by construction)

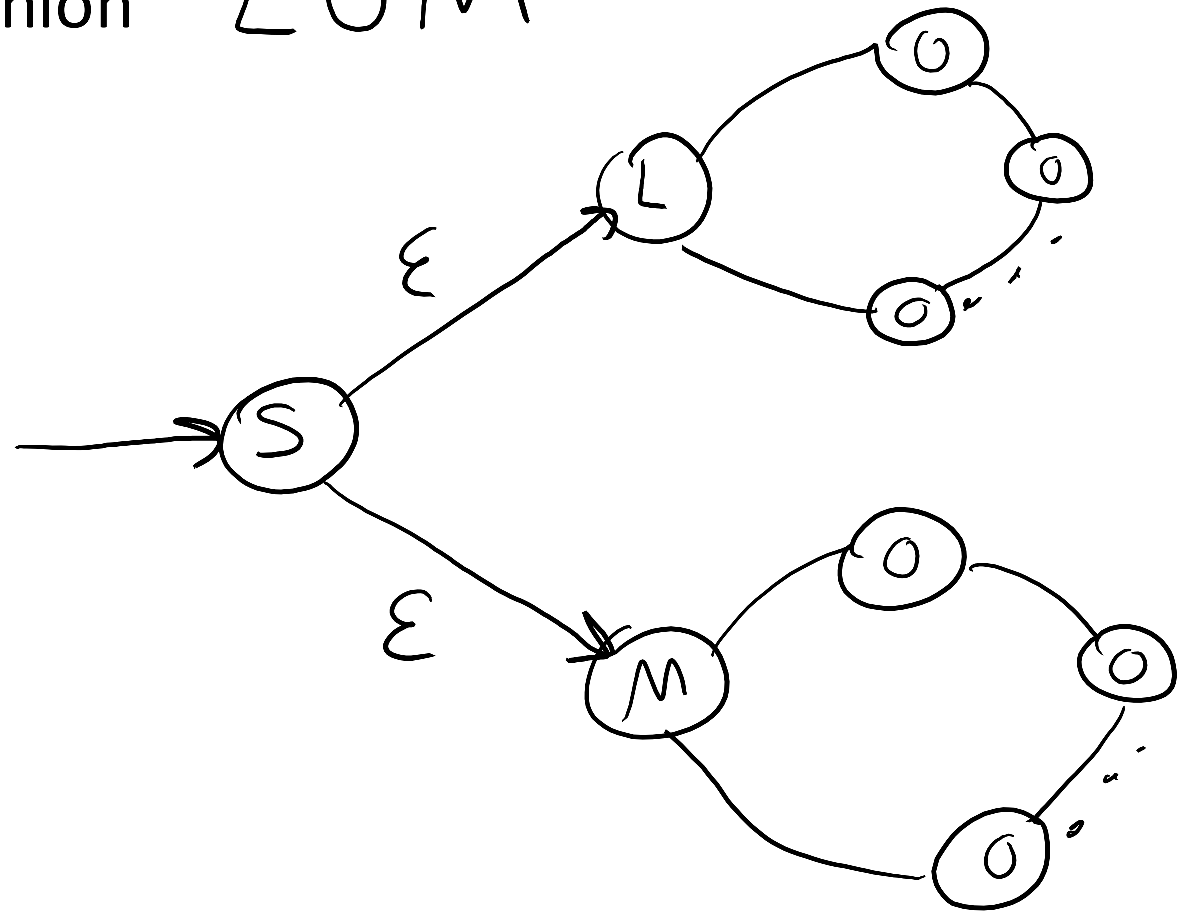
Suppose L, M are accepted by automata A,B.

A:

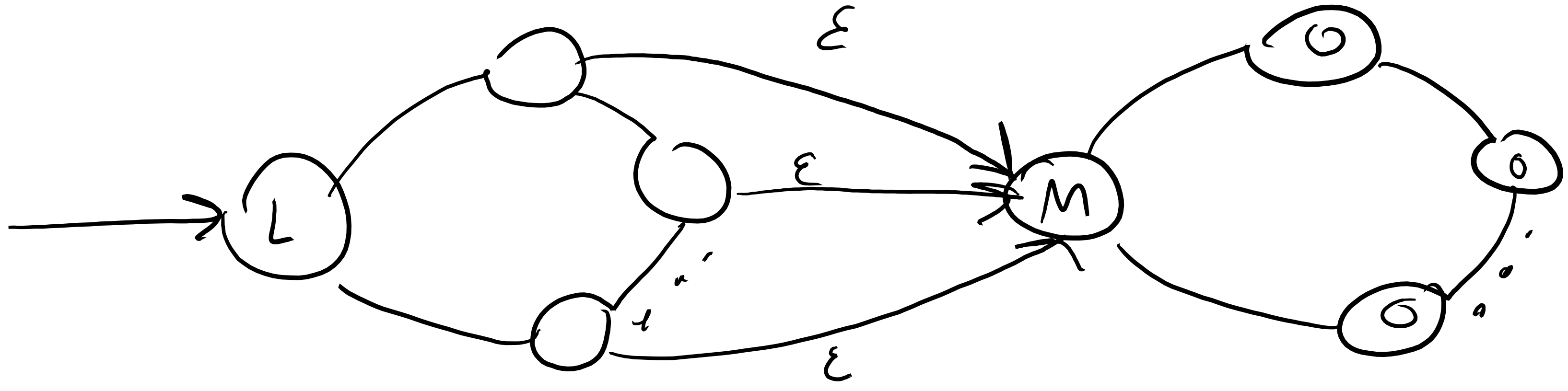
B:



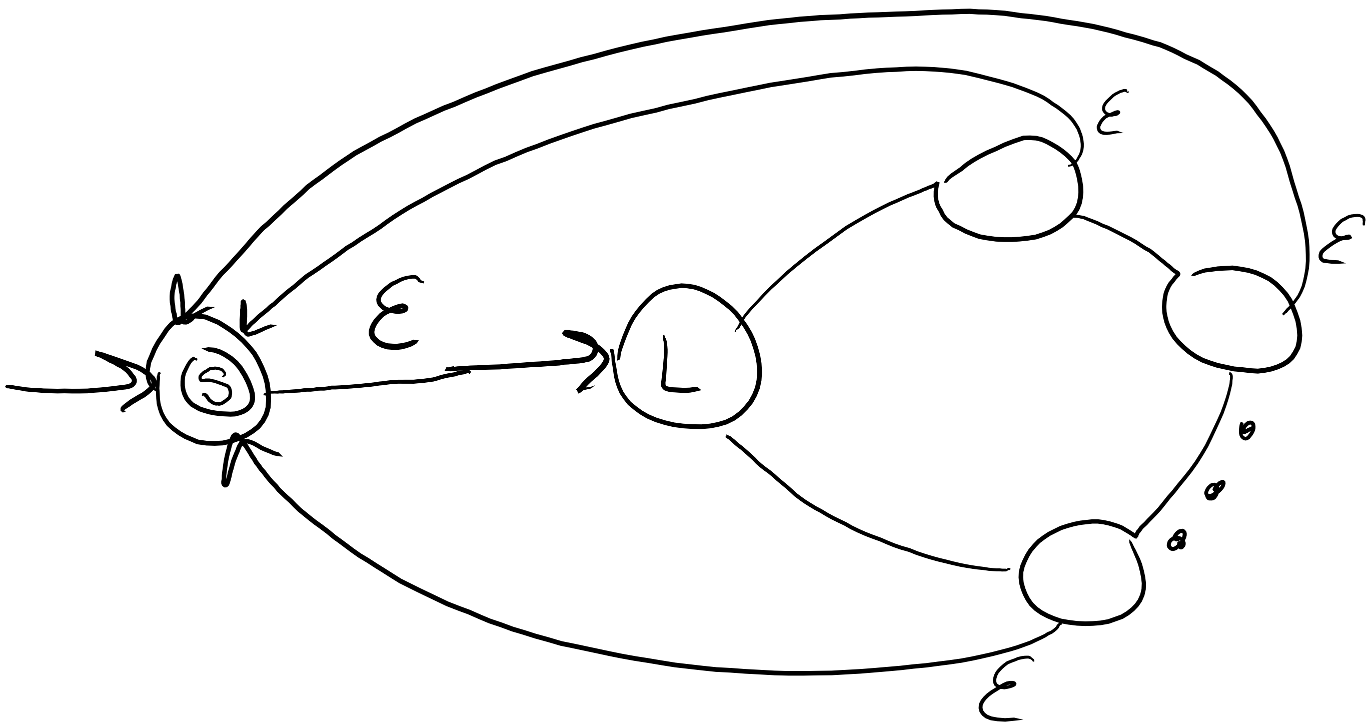
Union $L \cup M$



Concatenation LM



Kleene Star L^*



Complement $\overline{L}$: every favorable state becomes unfavorable. every state becomes favorable.

Must convert to DFA first

Intersection $L \cap M$

$$L \cap M = \overline{\overline{L} \cup \overline{M}} \qquad \neg(\neg A \vee \neg B) = A \wedge B$$

Difference $L - M$

$$L - M = L \cap \overline{M} = \overline{\overline{L} \cup M}$$

A regular expression is a string over the alphabet $\Sigma \cup \{(,),\varepsilon,\emptyset,\cup,*\}$

Recursive definition:

Basis Step: $\emptyset$, ε, $a \in \Sigma$ are regular expressions.
ground elements

Recursive Step: If α, β are regular expressions,
then so are

$(\alpha \cup \beta)$

$(\alpha \beta)$

α^*

ex: $(a^* b^*)$ ← easy

$(a^* \cup ((ab)^* \cup b^*)^*)^*$

← not so much

$L(\alpha)$ is the language represented by regular expression α.

$L(\emptyset) = \emptyset$

$L(\epsilon) = \{\varepsilon\}$

$L(a) = \{a\}$

If α, β are regular expressions, then

$L(\alpha \cup \beta) = L(\alpha) \cup L(\beta)$

$L(\alpha\beta) = L(\alpha)L(\beta)$

$L(\alpha^*) = (L(\alpha))^*$

Exercise: What language is represented by $(ab^*) \cup (a^*b)$

$$L((ab^*) \cup (a^*b))$$

$$L((ab^*)) \cup L((a^*b))$$

$$(L(a)L(b^*)) \cup (L(a^*)L(b))$$

$$(L(a)(L(b))^*) \cup ((L(a))^* L(b))$$

$$(\{a\}\{b\}^*) \cup (\{a\}^*\{b\}) = \{w \mid w \text{ is of the form } ab^n \text{ or } a^n b,\ n \geq 0\}$$