Automata 2/2

Thinh Tien Nguyen, Ph.D.

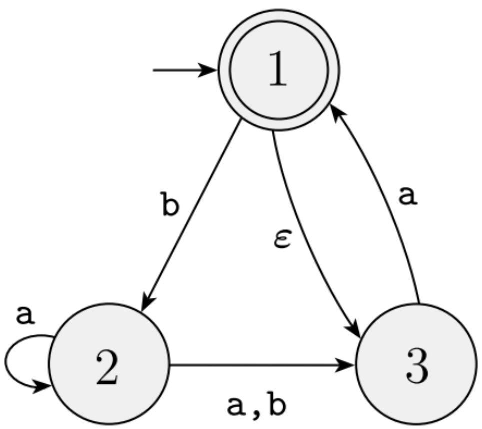
Online lecture

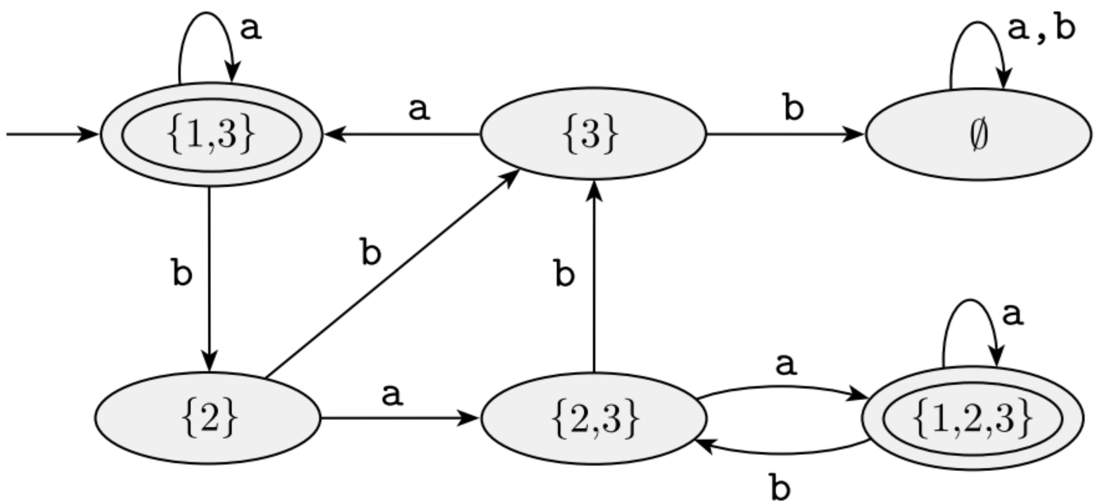
Equivalence

Property:

Every non-deterministic automaton has an equivalent deterministic automaton and vice versa.

Ex:





Regular languages

Definition:

An automaton M **recognizes** a language L if M accepts all strings in L. A language L is a **regular language** if there is a finite automaton M recognizes it.

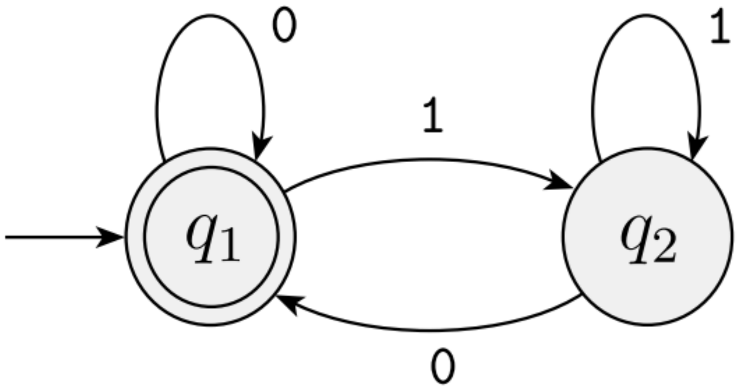
Property:

If A and B are two regular languages, so do

- **Union**: $A \cup B = \{x|\, x \in A \text{ or } x \in B\}$.
- **Concatenation**: $A \circ B = \{xy|\, x \in A \text{ and } y \in B\}$.
- **Star**: $A^* = \{x_1x_2 \ldots x_k|\, k \geq 0 \text{ and each } x_i \in A\}$.

Regular languages

Ex:



L(M):={w|w is the empty string or ends with 0}

Regular expressions

Definition:

Say that R is a ***regular expression*** if R is

1. a for some a in the alphabet Σ,
2. ε,
3. $\emptyset$,
4. $(R_1 \cup R_2)$, where R_1 and R_2 are regular expressions,
5. $(R_1 \circ R_2)$, where R_1 and R_2 are regular expressions, or
6. (R_1^*), where R_1 is a regular expression.

Regular expressions

Ex:

In the following instances, we assume that the alphabet Σ is $\{0,1\}$.

1. $0^*10^* = \{w \mid w \text{ contains a single } 1\}$.
2. $\Sigma^*1\Sigma^* = \{w \mid w \text{ has at least one } 1\}$.
3. $\Sigma^*001\Sigma^* = \{w \mid w \text{ contains the string } 001 \text{ as a substring}\}$.

Regular expressions

Property:

Any regular language can be described by a regular expression and vice versa. Thus a regular expression is equivalent to a finite automaton.

Ex:

$(a \cup b)^*$