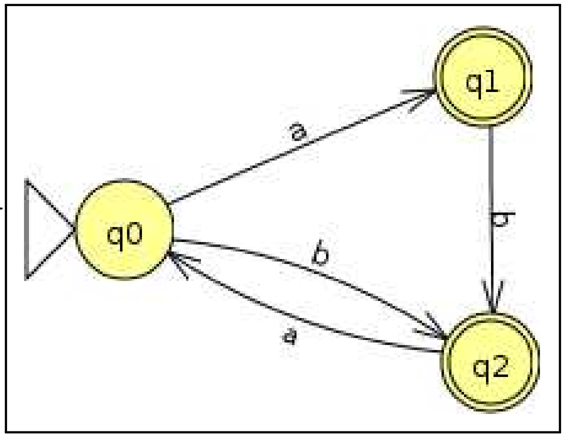
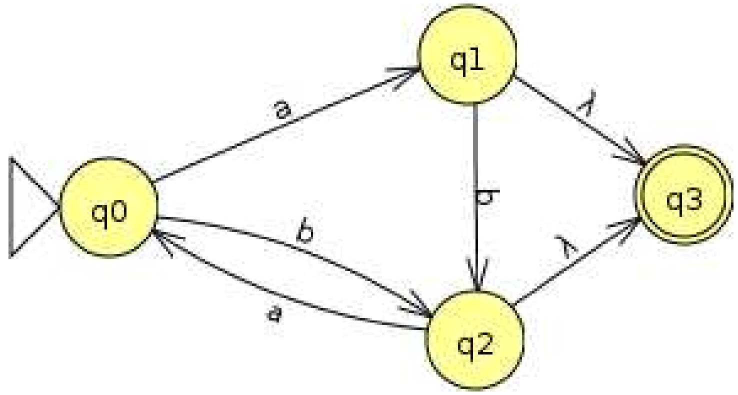
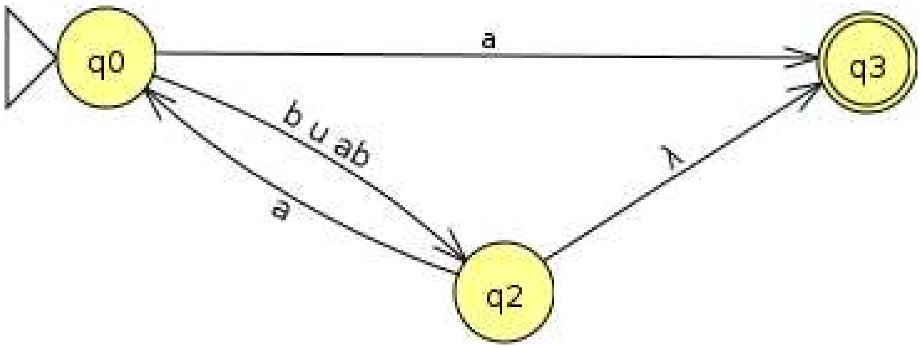
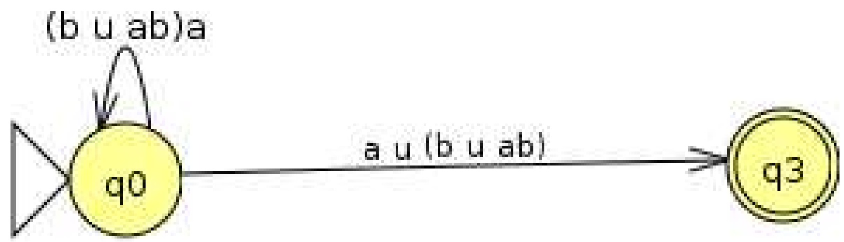
Regular Languages & Finite Automata

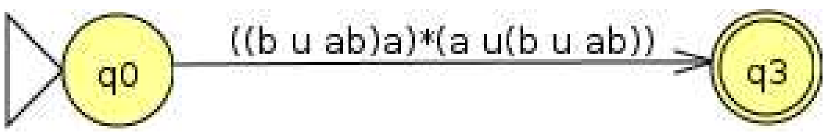
M











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

$$L(M) = L(e)$$

Theorem. Let $L \subseteq \Sigma^*$. Then the following are equivalent

(i) L is m-regular, i.e. $L = L(M)$ for some DFA (PFA, NFA, NFA$_\lambda$)

(ii) L is regular, i.e. $L = L(e)$ for some regular expression

Proof. (i) $\Rightarrow$ (ii) The steps on previous page preserve language

(ii) $\Rightarrow$ (i) 'By induction' (recursion) on a regular expression e

a machine M_e is defined, which is an NFA$_\lambda$ preserving language

that is, $L(e) = L(M_e)$

NFA$_\lambda$ is changed into a DFA in a language preserving way

by considering as new states certain sets of states of the first machine,

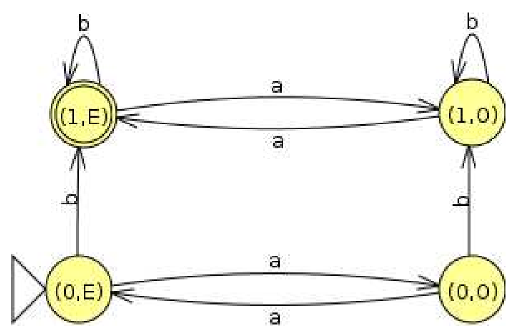
see previous lecture. ■

Let $\Sigma = \{a, b\}$. We will develop a tool to show that

$$\{a^n b^n \in \Sigma^* \mid n \geq 0\}$$

is not regular over Σ.

Consider



What happens if a word of length 4, 5, 6, 7, ... is accepted,

but not necessarily if it is of length 0, 1, 2, 3?

It has made a 'cycle' which can be repeated arbitrarily often!

'baaaa' is accepted, and also all 'baa(aa)^n'. We say 'aa' *can be pomped*.

Theorem. Let $L \subseteq \Sigma^*$ be a regular language

Then there exists a number $p \geq 1$ (pomping number) such that

for every $w \in L$ with $|w| \geq p$ one has the following
($|w|$ is the number of symbols of w)

(i) w can be split in three parts: $w = xyz$

(ii) $|xy| < p$ and $|y| > 0$

(iii) for all $n \geq 0$ one has $xy^n z \in L$. $\exists p \forall w \exists xyz \forall n$

Corollary $L = \{a^n b^n \mid n \geq 0\}$ is not regular

Proof. Suppose L is regular. Let p be as in the pumping lemma

Then $w = a^p b^p \in L$ satisfies $|w| \geq p$

Therefore we could write $a^p b^p = xyz$, with $|xy| < p$ and $xy^n z \in L$

Then $xy = a^q$, with $q < p$. But then $xy^2 z \notin L$. Contradiction. ■

$$\begin{array}{rcl} \neg\exists x.P(x) & \Leftrightarrow & \forall x.\neg P(x) \\ \neg\forall x.P(x) & \Leftrightarrow & \exists x.\neg P(x) \\ \neg\exists x.[Q(x) \;\&\; P(x)] & \Leftrightarrow & \forall x.[Q(x) \;\Rightarrow\; \neg P(x)] \\ \neg\forall x.[Q(x) \;\Rightarrow\; P(x)] & \Leftrightarrow & \exists x.[Q(x) \;\&\; \neg P(x)] \end{array}$$

Pumping lemma. For all regular languages L

$\exists p \forall w \in L.[|w| \geq p \;\Rightarrow$

$\exists xyz.[w = xyz \;\&\; |xy| < p \;\&\; |y| > 0 \;\&\; \forall n \in \mathbb{N}\,.xy^n z \in L]$

Application. In order to show that L is not regular we must show

$\forall p \exists w \in L.[|w| \geq p \;\&$

$\forall xyz.[w = xyz \;\&\; |xy| < p \;\&\; |y| > 0 \;\&\; \exists n \in \mathbb{N}\,.xy^n z \;\notin\; L]$

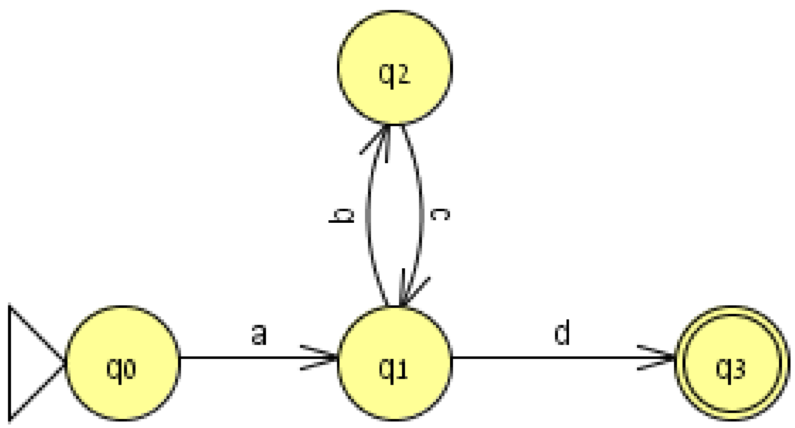
Let L be regular. Then L is m-regular. Let L be accepted by M.

Let M have p states

Then a word w of length $\geq p$ must pass twice a state q

Then $w = xyz$, where we read x to go to q, read y to loop at q, read z to go to a final node. But then $xy^n z$ is accepted for all n. ■

Example $abcd \in L(M)$



Since q_1 is visited twice we can pomp: $a(bc)^n d \in L(M)$