Pumping lemma for regular languages

Regular language is language accepted by an FA.

Theorem

Suppose L is a language over the alphabet Σ. If L is accepted by a finite automaton M, and if n is the number of states of M, then

$\forall$ *for every $x \in L$*
satisfying $|x| \geq n$
$\exists$ *there are three strings u, v, and w,*
such that $x = uvw$ and the following three conditions are true:
(1) $|uv| \leq n$,
(2) $|v| \geq 1$
$\forall$ *and (3) for all $m \geq 0$, $uv^m w$ belongs to L*

[M] Thm. 2.29

Pumping lemma for regular languages

Theorem

If

$\forall$ *for every* $n \geq 1$

$\exists$ *there exists* $x \in L$

with $|x| \geq n$

such that

$\forall$ *for every decomposition* $x = uvw$

with (1) $|uv| \leq n$,

and (2) $|v| \geq 1$

$\exists$ *(3) there exists* $m \geq 0$,

such that

$uv^m w \notin L$

then L *is not a regular language.*

[M] Thm. 2.29

Example

$L = \{ \, x \in \{a, b\}^* \mid n_a(x) > n_b(x) \, \}$ is not accepted by FA

[M] E 2.31

Pumping lemma as a game

Given a language L, to prove L is not a regular language:

1. Opponent picks n.
2. We choose a string $x \in L$ with $|x| \geq n$.
3. Opponent picks u, v, w with $x = uvw$, $|uv| \leq n$, $|v| \geq 1$.
4. If we can find $m \geq 0$ such that $uv^m w \notin L$, then we win.

If we can always win, then L does not fulfil the pumping lemma.

≈[VU Automata & Complexity] L3

Example

$L = \{ a^{i^2} \mid i \geq 0 \}$ is not accepted by FA

Proof...

[M] E 2.32

Example

$L = \{ a^{i^2} \mid i \geq 0 \}$ is not accepted by FA

Proof. Suppose L is accepted by an FA of n states. Consider $x = a^{n^2}$. Let u, v, w be such that $x = uvw$, $|uv| \leq n$ and $|v| \geq 1$. Hence, $u = a^i$, $v = a^j$ and $w = a^k$ such that $i + j + k = n^2$, $i + j \leq n$ and $j \geq 1$. Consider uv^2w. Note that

$$|uv^2w| = i + 2j + k = n^2 + j > n^2.$$

Moreover,

$$n^2 + j \leq n^2 + n < n^2 + 2n + 1 = (n+1)^2.$$

Hence, the length of uv^2w lies in between two consecutive squares, so it cannot be a square.

[M] E 2.32

$L = \{ a^{i^2} \mid i \geq 0 \}$

Fun fact

$L^4 = \{a\}^*$

Lagrange's four-square theorem

Let L be the set of legal C programs.
$x = \mathtt{main}()\{\{\{...\}\}\}$

[M] E 2.33

Not a characterization

$L = \{a^i b^j c^j \mid i \geq 1 \text{ and } j \geq 0\} \cup \{b^j c^k \mid j \geq 0 \text{ and } k \geq 0\}$

This example does not have to be known for the exam.

[M] E 2.39

Decision problem: problem for which the answer is 'yes' or 'no':

Given . . . , is it true that . . . ?

Given an undirected graph $G = (V, E)$*,*
does G *contain a Hamiltonian path?*

Given a list of integers $x_1, x_2, \ldots, x_n$*,*
is the list sorted?

decidable $\iff$ $\exists$ algorithm that decides

$M = (Q, \Sigma, \delta, q_0, A)$
membership problem $\quad x \in L(M)$?

Specific to M: Given $x \in \Sigma^*$, is $x \in L(M)$?

Arbitrary M: Given FA M with alphabet Σ, and $x \in \Sigma^*$, is $x \in L(M)$?

Decidable (easy)

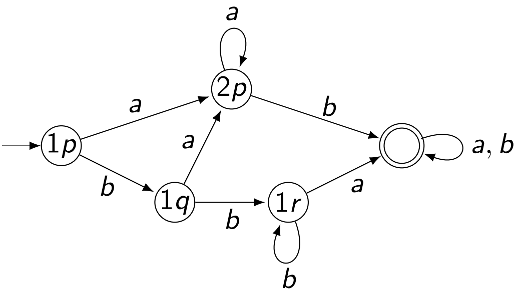
[M] E 2.34

The rest of Example 2.34 does not have to be known for the exam

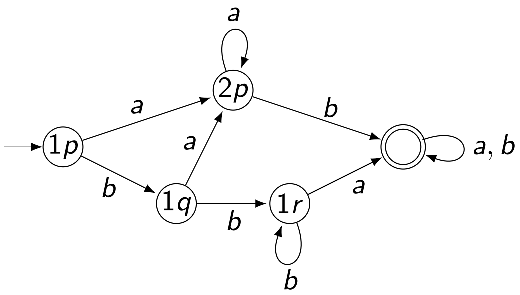
Section 3

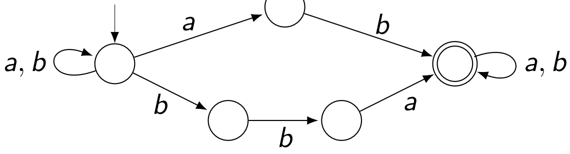
Non-Determinism, Regular Expressions, and Kleene's Theorem

2. Non-Determinism, Regular Expressions, and Kleene's Theorem
   - Non-determinism
   - Allowing Λ-transitions
   - Definitions
   - Making the automaton deterministic



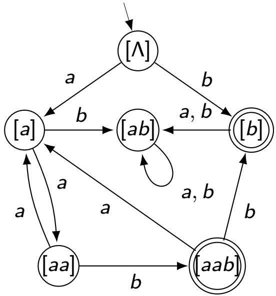
[M] see ↪E.2.18 (product construction)





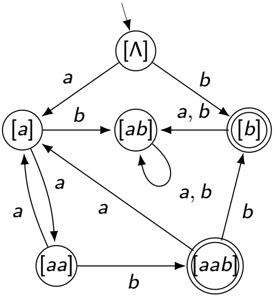
[M] see ↪E.2.18 (product construction)

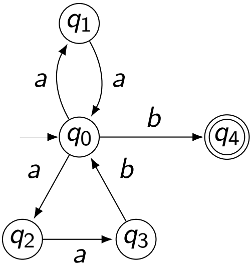
$L = \{aa, aab\}^*\{b\}$



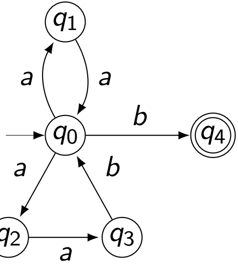
[M] E 3.6

$L = \{aa, aab\}^*\{b\}$

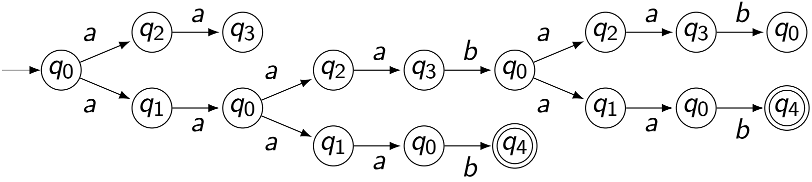




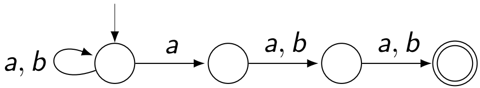
[M] E 3.6



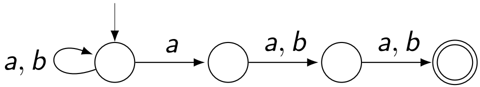
$x = aaaabaab$



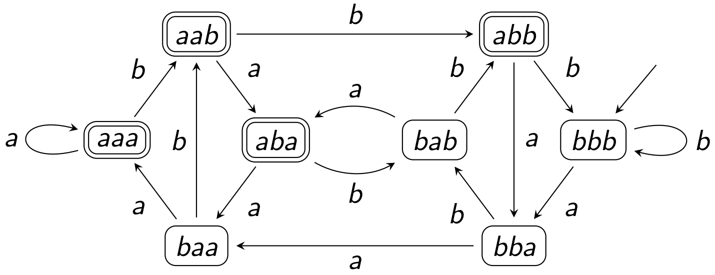
[M] E 3.6. also ↪E 2.22



[M] E. 2.24



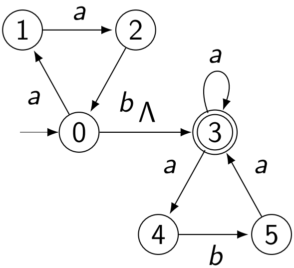
Also $\hookrightarrow$deterministic



$n + 1$ versus 2^n states.

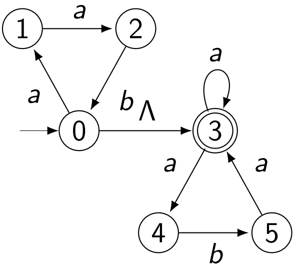
[M] E. 2.24

$L = \{aab\}^*\{a, aba\}^*$

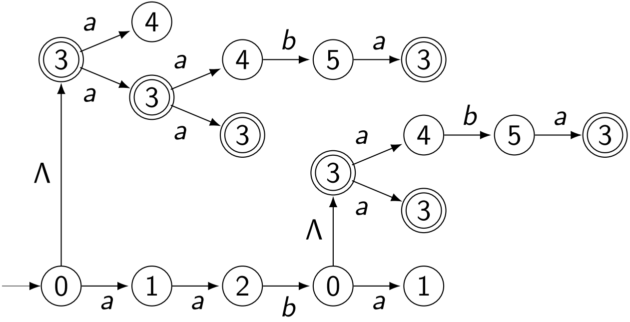


NFA

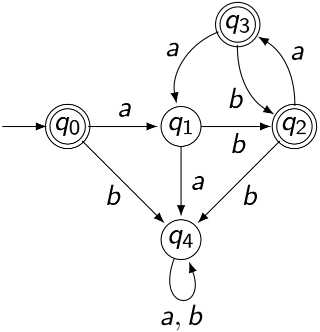
[M] E 3.9



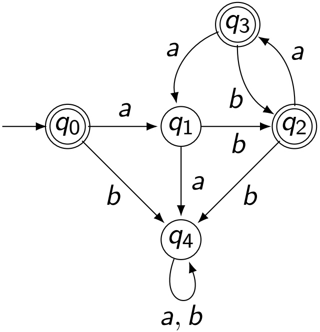
$x = aababa$

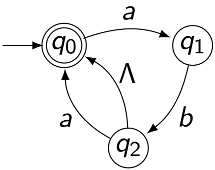


Example: $\{ab, aba\}^*$



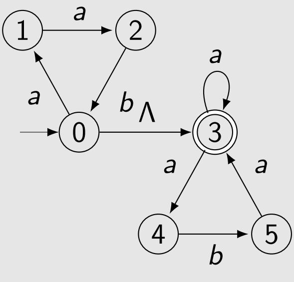
Example: $\{ab, aba\}^*$





Example

$\{aab\}^*\{a, aba\}^*$



NFA

δ	Λ	a	b
0	$\{3\}$	$\{1\}$	$\varnothing$
3	$\varnothing$	$\{3, 4\}$	$\varnothing$

Definition (FA)

[deterministic] finite automaton 5-tuple $M = (Q, \Sigma, q_0, A, \delta)$,
- Q finite set *states*;
- Σ finite *input alphabet*;
- $q_0 \in Q$ *initial* state;
- $A \subseteq Q$ *accepting* states;
- $\delta : Q \times \Sigma \to Q$ *transition function*.

[M] D 2.11 Finite automaton

[L] D 2.1 Deterministic finite accepter, has ‘final’ states

5-tuple $M = (Q, \Sigma, q_0, A, \delta)$

Definition ($\hookrightarrow$FA)

[deterministic] finite automaton
– $\delta : Q \times \Sigma \to Q$ *transition function*;

Definition (NFA)

nondeterministic finite automaton (with Λ-transitions)
– $\delta : \ldots$

5-tuple $M = (Q, \Sigma, q_0, A, \delta)$

Definition ($\hookrightarrow$FA)

[deterministic] finite automaton
– $\delta : Q \times \Sigma \to Q$ *transition function*;

Definition (NFA)

nondeterministic finite automaton (with Λ-transitions)
– $\delta : Q \times (\Sigma \cup \{\Lambda\}) \to 2^Q$

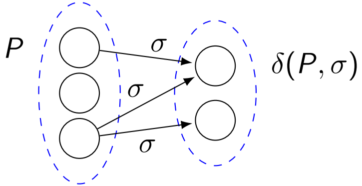
Where 2^Q the set of all subsets of Q:

$$2^Q = \{S \mid S \subseteq Q\}$$

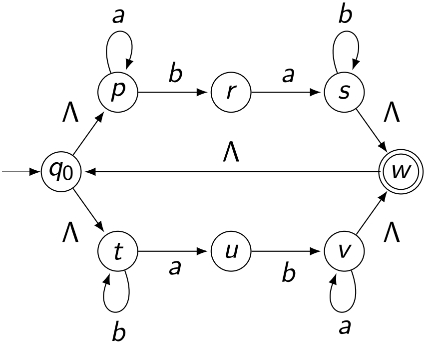
Extended transition function without Λ-transitions

Extend δ to subsets P:

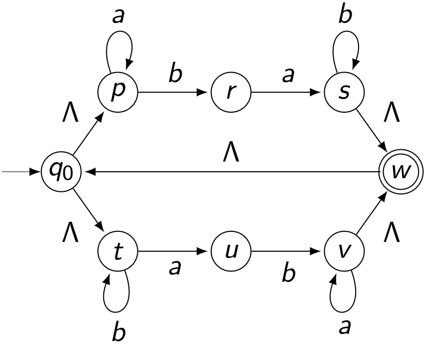
$\delta(P, \sigma) = \bigcup_{p \in P} \delta(p, \sigma) = \{q \in Q \mid q \in \delta(p, \sigma) \text{ for some } p \in P\}$.



$\delta^*(q, \Lambda) = \{q\}$
$\delta^*(q, y\sigma) = \delta(\delta^*(q, y), \sigma)$

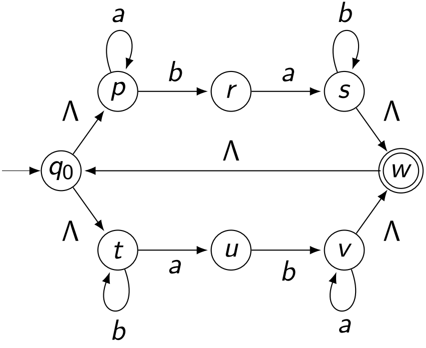


$$\Lambda(\{q_0\}) = \ldots$$
$$\Lambda(\{r\}) = \ldots$$
$$\Lambda(\{v\}) = \ldots$$



$\Lambda(\{q_0\}) = \{q_0, p, t\}$
$\Lambda(\{r\}) = \{r\}$
$\Lambda(\{v\}) = \{v, w, q_0, p, t\}$

$\{r, v\} \subseteq \Lambda(\{r, v\})$
then also $\{w\} \subseteq \Lambda(\{r, v\})$
then also $\{q_0\} \subseteq \Lambda(\{r, v\})$
then also $\{p, t\} \subseteq \Lambda(\{r, v\})$
$\Lambda(\{r, v\}) = \{r, v, w, q_0, p, t\}$



NFA $M = (Q, \Sigma, q_0, A, \delta)$ $\quad S \subseteq Q$

$\Lambda(\{q_0\}) = \{q_0, p, t\}$
$\Lambda(\{r\}) = \{r\}$
$\Lambda(\{v\}) = \{v, w, q_0, p, t\}$

$\{r, v\} \subseteq \Lambda(\{r, v\})$
then also $\{w\} \subseteq \Lambda(\{r, v\})$
then also $\{q_0\} \subseteq \Lambda(\{r, v\})$
then also $\{p, t\} \subseteq \Lambda(\{r, v\})$
$\Lambda(\{r, v\}) = \{r, v, w, q_0, p, t\}$

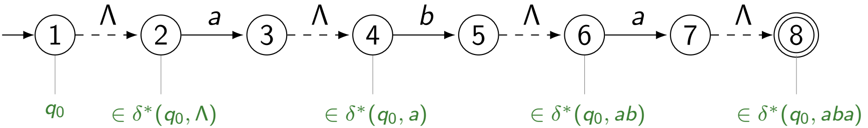
Definition

– $S \subseteq \Lambda(S)$
– $q \in \Lambda(S)$, then $\delta(q, \Lambda) \subseteq \Lambda(S)$

[M] D 3.13

Extended transition function with Λ-transitions

NFA $M = (Q, \Sigma, q_0, A, \delta)$

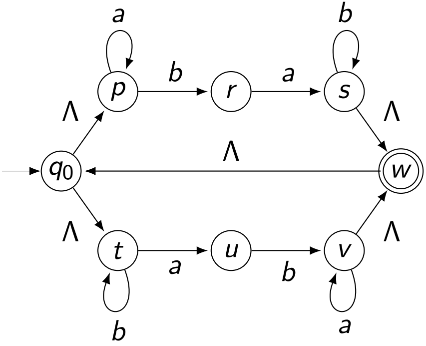


Definition

- $\delta^*(q, \Lambda) = \Lambda(\,\{q\}\,) \qquad q \in Q$
- $\delta^*(q, y\sigma) = \Lambda(\,\delta(\,\delta^*(q, y), \sigma\,)\,) \qquad q \in Q,\ y \in \Sigma^*,\ \sigma \in \Sigma$

$\delta^*(1, a) = \ldots$

[M] D 3.14



Λ
$\Lambda(\{q_0\}) = \{q_0, p, t\}$
$\delta^*(q_0, \Lambda) = \{q_0, p, t\}$

$\Lambda \cdot a$
$\delta(\{q_0, p, t\}, a) = \{p, u\}$
$\delta^*(q_0, a) = \Lambda(\{p, u\}) = \{p, u\}$

$a \cdot b$
$\delta(\{p, u\}, b) = \{r, v\}$
$\delta^*(q_0, ab) = \Lambda(\{r, v\}) =$
$\{r, v, w, q_0, p, t\}$

$ab \cdot a$
$\delta(\{r, v, w, q_0, p, t\}, a) = \{s, v, p, u\}$
$\delta^*(q_0, aba) = \Lambda(\{s, v, p, u\}) =$
$\{s, w, q_0, p, t, v, u\}$

[M] E 3.15

NFA $M = (Q, \Sigma, q_0, A, \delta)$

Theorem

$q \in \delta^*(p, x)$ iff there is a path in [the transition graph of] M from p to q with label x (possibly including Λ-transitions).

$\delta^*(q_0, x) = \varnothing$ no path for x from initial state

Definition

A string $x \in \Sigma^*$ is accepted by $M = (Q, \Sigma, q_0, A, \delta)$ if $\delta^*(q_0, x) \cap A \neq \varnothing$.
The *language $L(M)$ accepted* by M is the set of all strings accepted by M.

[M] D 3.14 [L] D 2.2