

# UNIX reguláris kifejezések

- A **gyakorlatban** is használatos regex formalizmus (csak a főbb lehetőségeket tekintjük):
- **Metakarakterek:** `[ , ] , ^ , + , * , |`
- Bármely **a** karakter, ami nem metakarakter, az `{a}` nyelvet jelöli
- `[ab...z]` az `{a, b, ..., z}` nyelvet jelöli; intervallum is megadható a - jellel
  - Pl. `[abq]` az `{a, b, q}` nyelvet, `[abq-s]` pedig az `{a, b, q, r, s}` nyelvet jelöli
- **\*** az **iterációt** jelöli (mint a mi reguláris kifejezéseinkben)
  - Pl. `(00)*` az `{ε, 00, 0000, ...}` nyelvet jelöli
- **+** a **pozitív iterációt** jelöli (mint eddig is)
- **|** **megfelel** a vagy-nak (ami a mi reguláris kifejezéseinkben **a +-nak** felel meg)
  - Pl. `(00 | 11)*` a mi reguláris kifejezésünkkel `(00 + 11)*` alakban adható meg

# UNIX reguláris kifejezések

Tovább lehetőségek:

- A **^** **jel**
  - lehet **tagadás** ha [ után van, pl. [ **^**0–9] azon karakterek halmazát jelöli, melyekben nincsenek számok
  - jelölheti a **sor elejét**
- **\$**: a sor **végére** illeszkedik
- **\n**: **új sor** karakterre illeszkedik
- **\b** egy **szó elejére** vagy **végére** illeszkedik
  - Pl. **\b1 (0 | 1) \*1\b** azokat a legalább kettő hosszú szavakat listázza melyeknek az első és utolsó betűje 1

# UNIX reguláris kifejezések - Példa

Adjunk UNIX reguláris kifejezéseket, melyekkel azon **szavakra** kereshetünk, melyek

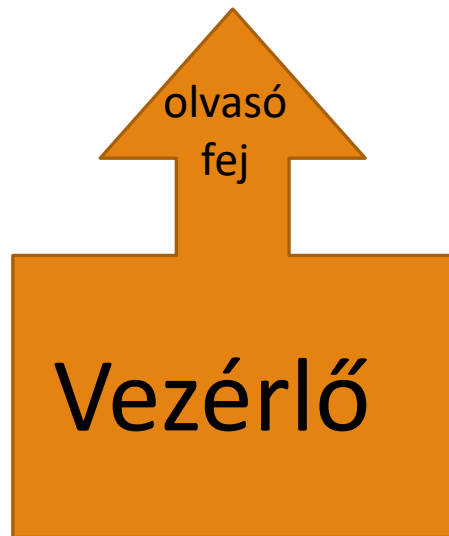
- tartalmazzanak kép vagy tép részszót:
  - `[k|t]ép`
- „lé”-vel kezdődnek:
  - `\blé`
- csak `a, b, c, d, r, s, t, u` betűkből állnak:
  - `\b[a-dr-u]+\b`
- három betűsek és „l”-lel kezdődnek és „p”-re végződnek:
  - `\bl.p\b`
- nem tartalmazzanak „r” betűt:
  - `\b[^r\n]*\b`

Adjunk olyan kifejezést, amivel „m”-mel kezdődő és „s”-sel végződő **sorokat** kereshetünk:

- `^m[^n]*$`

# Nemdeterminisztikus véges automata

l t t j ö n a b e m e n e t



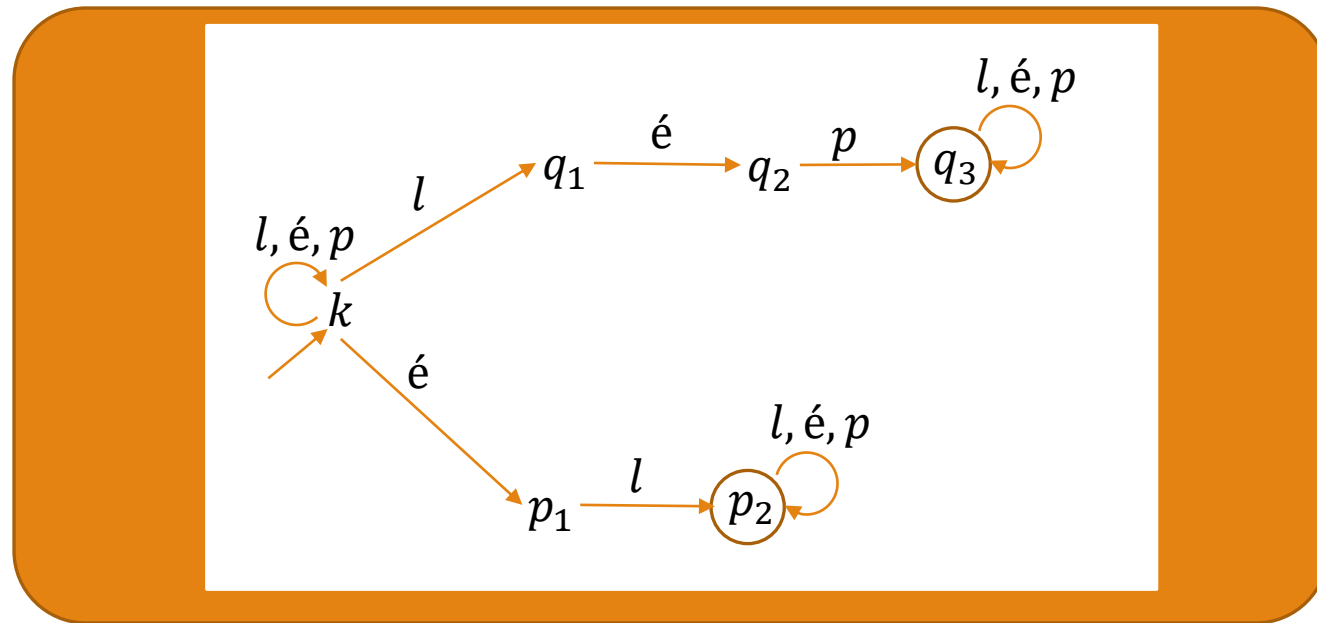
- Adott állapotból adott jel hatására több (esetleg 0) állapotba is átmehet,
- üres szó hatására is állapotot válthat.

# Nemdeterminisztikus véges automata – Motiváció

## Feladat:

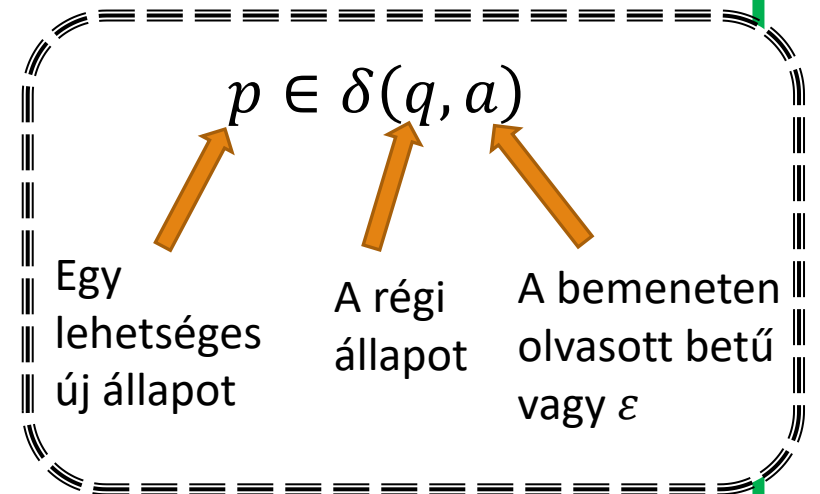
- Döntsük el, hogy a bemenetben szerepel-e részszóként a *lép* vagy az *él* szó
- Az egyszerűség kedvéért feltesszük, hogy a bemenetben csak ez a három betű szerepel

**Nemdeterminisztikus** véges automatát könnyebb felírni, mint determinisztikusot:



# Nemdeterminisztikus véges automata

- (üres átmenetekkel ellátott) **Nemdeterminisztikus véges automata** (röviden NVA):  $(Q, \Sigma, \delta, q_0, F)$ , ahol
  - $Q, \Sigma, q_0, F$  ugyanaz, mint a véges automaták esetében
  - $\delta: Q \times \Sigma_\varepsilon \rightarrow P(Q)$ , az **átmeneti függvény**, ahol
    - $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$
    - $P(Q)$  pedig a  $Q$  részhalmazainak halmaza
- Ha  $p \in \delta(q, a)$ , ahol  $a \in \Sigma_\varepsilon$ , akkor  $q \xrightarrow{a} p$  az  $M$  egy **átmenete**
- A  $\delta$  most is egyértelműen leírható egy **átmeneti diagrammal**, az  $M$  pedig megadható úgy, hogy az átmenetdiagrammban megjelöljük a kezdő- és végállapotokat



# Nemdeterminisztikus véges automaták

Figyelem! Itt minden ugyanúgy van mint a determinisztikus esetben, kivéve, hogy a futás definíciójában  $a_i$  lehet  $\varepsilon$  is!

- Legyen  $M = (Q, \Sigma, \delta, q_0, F)$  egy NVA,  $q \in Q$ , és  $w \in \Sigma^*$
- **Megy  $q$ -ból induló futása egy  $w$  szón:** átmenetek egy  $q_1 \xrightarrow{a_1} q_2 \xrightarrow{a_2} q_3 \dots q_n \xrightarrow{a_n} q_{n+1}$  sorozata, ahol  $n \geq 0$ ,  $q_1 = q$ ,  $a_1, \dots, a_n \in \Sigma_\varepsilon$  és  $w = a_1 a_2 \dots a_n$
- Ez a futás **sikeres**, ha  $q_{n+1} \in F$
- $M$  **elfogadja**  $w$ -t:  $M$ -nek van  $q_0$ -ból induló sikeres futása a  $w$ -n
- Az  $M$  által **felismert nyelv**:

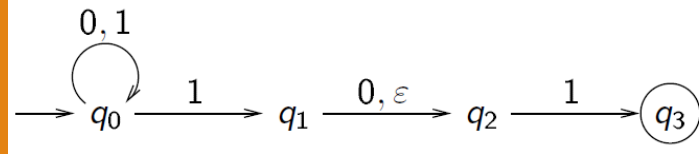
$$L(M) = \{w \in \Sigma^* \mid M \text{ elfogadja } w\text{-t}\}$$

# Nemdeterminisztikus véges automaták

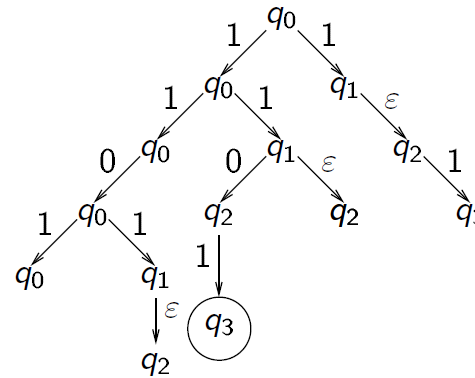
- Egy  $M$  NVA-nak egy adott  $u$  szón **egytől különböző** számú futása is lehet (azaz lehet, hogy egy sincs neki, vagy egynél több van)
- A véges automaták is tekinthetők NVA-knak: egy  $M = (Q, \Sigma, \delta, q_0, F)$  véges automata egy olyan NVA, ahol minden  $a \in \Sigma, q \in Q$  esetén  $|\delta(q, a)| = 1$ 
  - Ekkor  $M$ -nek minden szóra **pontosan egy** futása van
  - A véges automatákat szokás **determinisztikus** véges automatáknak (DVA-knak) is nevezni
- Azt mondjuk, hogy két NVA,  $M_1$  és  $M_2$  **ekvivalensek**, ha  $L(M_1) = L(M_2)$



# Nemdeterminisztikus véges automaták - Példa



- A felismert nyelv:  $\{u \in \{0,1\}^* \mid u \text{ 101-re vagy 11-re végződik}\}$
- Futások az 1101 szó prefixein:



# Nemdeterminisztikus véges automaták

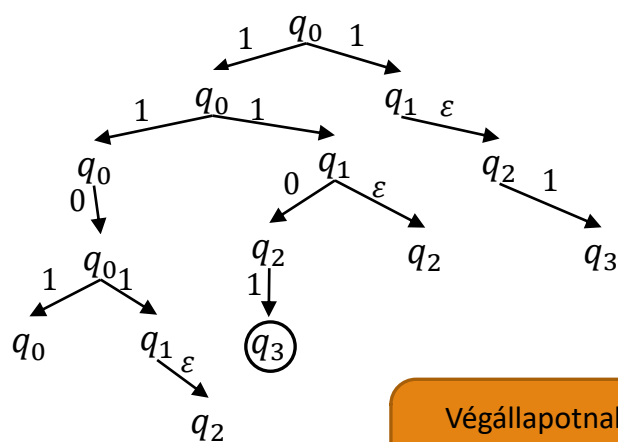
## Bizonyítás

Minden NVA-val felismerhető nyelv felismerhető DVA-val is

Legyen  $M = (Q, \Sigma, \delta, q_0, F)$  egy NVA. Megadunk egy  $M$ -mel **ekvivalens**  $M' = (Q', \Sigma, \delta', q'_0, F')$  DVA-t

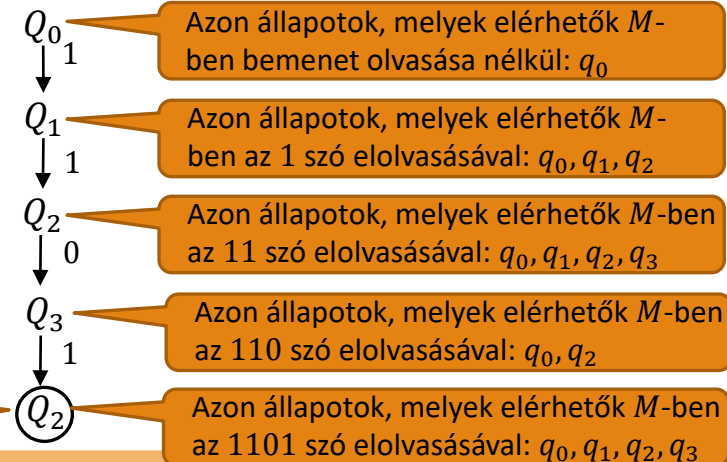
**Ötlet:**  $M'$  egyszerre szimulálja  $M$  összes számítását egy adott szón:

$M$  összes számítása az 1101-en



Végállapotnak jelöljük, mert  $M$  elér végállapotot az 1101 szóval ( $q_3$ -at)

$M'$  egyetlen számítása 1101-en



# Nemdeterminisztikus véges automaták

## Bizonyítás (folyt.)

Az  $\varepsilon$ -átmenetek miatt szükség lesz a következő műveletre,  $X \subseteq Q$ ,

$$\hat{X} = \{s \in Q \mid \exists q \in X, \text{ hogy } M\text{-nek van egy } q\text{-ból induló és } s\text{-be érkező futása az } \varepsilon \text{ szón}\}$$

Legyen  $M' = (Q', \Sigma, \delta', q'_0, F')$ , ahol

- $Q' = P(Q)$  (azaz a  $Q$  részhalmazainak halmaza)
- $q'_0 = \{\widehat{q_0}\}$  (a  $q_0$ -ból  $\varepsilon$ -átmenetekkel elérhető állapotok halmaza)
- $F' = \{X \in Q' \mid X \cap F \neq \emptyset\}$  (azaz a  $Q$  összes olyan részhalma, ami tartalmaz eredeti végállapotot)
- $\delta': P(Q) \times \Sigma \rightarrow P(Q)$ ,

$$\delta'(X, a) = \hat{Y},$$



Az  $Y$ -beli állapotokból  $\varepsilon$ -átmenetekkel elérhető állapotok

$$Y = \bigcup_{q \in X} \delta(q, a)$$



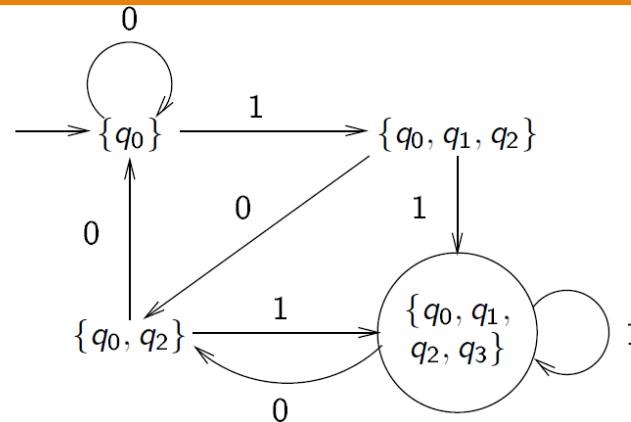
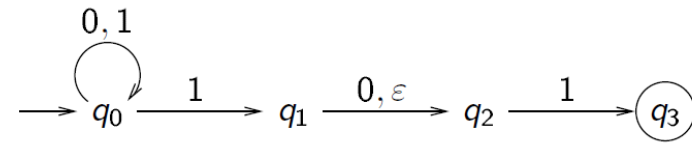
Az  $X$ -beli állapotokból az  $M$  által  $a$ -t olvasva elérhető állapotok

Legyen  $M = (A, B, t, k, V)$  egy tetszőleges NVA és  $q \in A$

- $q$ -t **elérhetőnek** nevezzük, ha  $M$ -nek van  $k$ -ból induló és  $q$ -ba érkező futása valamely alkalmas  $w \in \Sigma^*$  szón

A determinizáló algoritmus módosítható úgy, hogy  $M'$  egy olyan VA legyen, aminek minden állapota elérhető

# Nemdeterminisztikus véges automaták – Példa determinizálásra



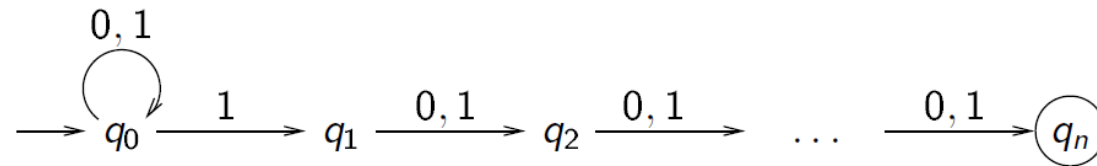
# Nemdeterminisztikus véges automaták

A determinizálás során a kapott automata állapotszáma akár **exponenciális** is lehet az eredeti NVA állapotszámának függvényében

Legyen  $n \geq 1$  és  $L_n = \{0,1\}^* \{1\} \{0,1\}^{n-1} = \{w \in \{0,1\}^* \mid |w| \geq n \text{ és } w \text{ hátulról } n\text{-ik betűje } 1\}$   
Ekkor  $L_n$  felismerhető  $n + 1$  állapotú NVA-val, de minden  $L_n$ -et felismerő (determinisztikus) véges automatának legalább  $2^n$  állapota van

Bizonyítás (csak az első állítás)

$L_n$ -et felismeri a következő NVA:



# NVA-k minimalizálása

Legyen  $N$  egy NVA. Ebben a részben megengedjük, hogy  $N$ -nek akár **több kezdőállapota** is legyen

- Ekkor az  $N$  által felismert nyelv azon  $u$  szavak halmaza, melyekre van  $N$ -nek **valamelyik** kezdőállapotból induló sikeres futása
- Jelöljön  $d(N)$  egy  $N$ -nel ekvivalens DVA-t (a determinizáló algoritmus könnyen kiterjeszthető több kezdőállapotra)
- Az  $N$  egy  $q$  állapototát **csapdaállapotnak** nevezzük, ha nincs olyan  $q$ -ból induló futása  $N$ -nek, ami végállapotba érkezik; jelölje  $n(N)$  azt az NVA-t amit úgy, kapunk, hogy  $N$ -ből elhagyjuk a csapdaállapotokat; világos, hogy  $L(N) = L(n(N))$
- Az  $N$  **fordítottját** (vagy **tükrözését**)  $f(N)$ -nel jelöljük és a következőképpen definiáljuk:
  - $f(N)$  állapotai ugyanazok, mint  $N$ -nek,
  - $N$  kezdőállapotai lesznek  $f(N)$  végállapotai,  $N$  végállapotai lesznek  $f(N)$  kezdőállapotai, és
  - $f(N)$  átmeneteit pedig úgy kapjuk, hogy  $N$  átmeneteit „megfordítjuk”, azaz  $q \xrightarrow{a} p$  akkor és csak akkor átmenete  $N$ -nek, ha  $p \xrightarrow{a} q$  átmenete  $f(N)$ -nek

Tetszőleges  $L \subseteq \Sigma^*$  nyelvre legyen  $L^{-1} = \{ u^{-1} \mid u \in L \}$ , ahol ha  $u = a_1 \dots a_n, a_1, \dots, a_n \in \Sigma$ , akkor  $u^{-1} = a_n \dots a_1$

# NVA-k minimalizálása

Tetszőleges  $N$  NFA-ra  $L(f(N)) = L(N)^{-1}$

Legyen  $M$  egy DVA. Azt mondjuk, hogy  $M$  **minimális**, ha minden  $M$ -mel ekvivalens  $M'$  DVA-ra az  $M'$ -nek legalább annyi állapota van, mint  $M$ -nek

Minden  $M$  NVA-hoz egyértelműen létezik egy **ekvivalens minimális**  $M'$  DVA

- Itt az „egyértelműen” azt jelenti, hogy ha  $M_1$  és  $M_2$  két  $M$ -mel ekvivalens minimális DVA, akkor  $M_1$  és  $M_2$  az állapotaik címkéjétől eltekintve megegyeznek

Bizonyítás (Brzowski NFA minimalizálási algoritmusát közöljük)

Legyen  $M$  egy NVA. Megadunk egy  $M$ -mel ekvivalens minimális  $M'$  DVA-t (az  $M'$  egyértelműségét nem bizonyítjuk)

- Először kiszámítjuk az  $M$  **fordítottját**
  - $M_1 := f(M)$
- Megjegyzés:  $M_1$  még akkor is lehet nemdeterminisztikus ha  $M$  determinisztikus!

# NVA-k minimalizálása

## Bizonyítás (folyt.)

- Ezután elhagyjuk  $M$  **csapdaállapotait**
  - $M_2 := n(M_1)$
- Ezután **determinizáljuk**  $M_2$ -t
  - $M_3 := d(M_2)$
- **Megjegyzés:** Belátható, hogy  $M_3$  az  $L(M)^{-1}$  nyelvet felismerő minimális DVA
- Ezután kiszámítjuk az  $M_3$  **fordítottját** és elhagyjuk a kapott NVA **csapdaállapotait**
  - $M_4 := n(f(M_3))$
- Megint **determinizálunk**:  $M' := d(M_4)$
- A fenti megjegyzés alapján  $M'$  az  $L(M_3)^{-1} = (L(M)^{-1})^{-1} = L(M)$  nyelvet felismerő, tehát az  $M$ -mel ekvivalens minimális DVA



# Példa minimalizálásra

