

# A számítástudomány alapjai

Dr. Ésik Zoltán  
SZTE, Számítástudomány Alapjai Tanszék

|  |            |
|--|------------|
| Bevezetes  |            |
| Bevezetés . . . . .                                      | slide #2   |
| Automaták és formális nyelvek                            |            |
| Szavak és nyelvek. . . . .                               | slide #4   |
| Véges automaták. . . . .                                 | slide #6   |
| Felismerhető nyelvek . . . . .                           | slide #12  |
| Műveletek nyelveken . . . . .                            | slide #14  |
| Reguláris nyelvek. . . . .                               | slide #18  |
| Felismerhető nyelvek zártsági tulajdonságai I . . . . .  | slide #19  |
| Véges nemdeterminisztikus automata . . . . .             | slide #21  |
| Felismerhető nyelvek zártsági tulajdonságai II . . . . . | slide #30  |
| Reguláris kifejezések . . . . .                          | slide #33  |
| Kleene tétele . . . . .                                  | slide #35  |
| Reguláris nyelvek pumpáló lemmája . . . . .              | slide #42  |
| Környezetfüggetlen nyelvtanok . . . . .                  | slide #45  |
| Derivációs fák . . . . .                                 | slide #51  |
| Környezetfüggetlen nyelvek. . . . .                      | slide #56  |
| Chomsky normálforma. . . . .                             | slide #59  |
| Veremautomaták . . . . .                                 | slide #67  |
| Környezetfüggetlen nyelvek pumpáló lemmája . . . . .     | slide #74  |
| Nyelvtanok . . . . .                                     | slide #79  |
| Chomsky-féle hierarchia . . . . .                        | slide #82  |
| Kiszámíthatóságelmélet                                   |            |
| Turing-gépek . . . . .                                   | slide #83  |
| Nemdeterminisztikus Turing-gépek . . . . .               | slide #89  |
| Turing-géppel felsorolható nyelvek. . . . .              | slide #91  |
| A kiszámítás egyéb modelljei . . . . .                   | slide #93  |
| Algoritmusok. . . . .                                    | slide #94  |
| Néhány eldönthető nyelv (probléma) . . . . .             | slide #97  |
| Néhány eldönthetetlen nyelv (probléma) . . . . .         | slide #104 |
| Rice tétele. . . . .                                     | slide #110 |
| A Post megfelelezési probléma. . . . .                   | slide #111 |
| Bonyolultságelmélet                                      |            |
| Bonyolultságelmélet . . . . .                            | slide #115 |
| Turing gépek időigénye. . . . .                          | slide #120 |
| A <b>P</b> nyelvosztály . . . . .                        | slide #122 |
| Az <b>NP</b> osztály . . . . .                           | slide #125 |
| Polinom időben verifikálható nyelvek. . . . .            | slide #127 |
| Polinom idejű visszavezetés. . . . .                     | slide #128 |
| <b>NP</b> -teljes nyelvek . . . . .                      | slide #129 |
| HAMILTON-ÚT <b>NP</b> -teljes . . . . .                  | slide #136 |
| SAT <b>NP</b> -teljes . . . . .                          | slide #140 |
| Az <b>NP</b> szerkezete . . . . .                        | slide #146 |
| A <b>coNP</b> osztály . . . . .                          | slide #147 |
| Tárkonyolultság . . . . .                                | slide #149 |
| Savitch tétele . . . . .                                 | slide #150 |
| A <b>PSPACE</b> és <b>NPSpace</b> osztályok . . . . .    | slide #153 |
| QBF <b>PSPACE</b> -teljes . . . . .                      | slide #154 |
| FÖLDRAJZI JÁTÉK <b>PSPACE</b> -teljes. . . . .           | slide #158 |
| Az <b>L</b> és <b>NL</b> osztályok . . . . .             | slide #160 |

|  |            |
|--|------------|
| Logaritmus tárral való visszavezetés . . . . . | slide #161 |
| Hierarchia . . . . .                           | slide #163 |

## Bevezetés

### Kiszámíthatóság elmélet

1930-as évek közepétől

Mely feladatok oldhatók meg algoritmikusan (=számítógéppel)?

### Bonyolultságelmélet

1970-es évek elejétől

Mely feladatokat lehet hatékonyan megoldani számítógéppel?

Hogyan lehet a számítógéppel megoldható feladatokat osztályozni a megoldásukhoz szükséges erőforrások mennyisége szerint?

### Automaták és formális nyelvek elmélete

1950-es évek közepétől

Alapul szolgál a fenti két területhez de számos egyéb alkalmazási területe is van:

fordítóprogramok, szerkesztők, logika és programozási logikák, szekvenciális áramkörök, idegi folyamatok modellezése, mesterséges intelligencia,...

## Bevezetés

### Automaták és formális nyelvek

Véges automaták és reguláris nyelvek

Veremautomaták és környezetfüggetlen nyelvek

Chomsky-féle hierarchia

### Kiszámíthatóság elmélet

Turing-gépek és a Church–Turing tézis

Eldönthető nyelvek és rekurzívan felsorolható nyelvek

A megállási probléma eldönthetlensége

Egyéb algoritmikusan megoldhatatlan problémák

### Bonyolultságelmélet

Bonyolultsági osztályok

A **P** és **NP** osztályok

Visszavezetés és **NP**-teljes problémák

**PSPACE, L, NL**

## Szavak és nyelvek

**Definíció (Szavak)** Legyen  $\Sigma$  véges nemüres halmaz.  $\Sigma$ -feletti szón a  $\Sigma$  elemeiből (betűiből) képzett véges sorozatot értünk:

$$w = w_1 \dots w_n, \quad w_i \in \Sigma, \quad i = 1, \dots, n$$

Az  $n$  nemnegatív egész számot a  $w$  szó hosszának nevezzük. **Jelölés:**  $|w|$ . A 0 hosszúságú szót az **üres szónak** nevezzük, **jelölése**  $\varepsilon$ .

**Példa**  $\Sigma = \{0, 1\}$   
 $w = 01001 \quad |w| = 5$   
 $w' = 000 = 0^3 \quad |w'| = 3$

**Jelölés**  $\Sigma^*$  jelöli a  $\Sigma$  feletti szavak halmazát.

## Szavak és nyelvek

**Definíció (Nyelv)**  $\Sigma$ -feletti nyelven a  $\Sigma^*$  egy részhalmazát értjük.

**Példa**  $\Sigma = \{0, 1\}$

**Véges nyelvek:**

$$\{0, 01, 001\}, \quad \{\varepsilon\}, \quad \{\varepsilon, 10\}, \quad \emptyset$$

**Végtelen nyelvek:**

$$\Sigma^*, \quad \{0^n 1^m : n, m \geq 0\}, \quad \{0^n 1^n : n \geq 0\}$$

$$\{w \in \Sigma^* : |w|_0 = |w|_1\}, \quad \{0^n 1^m 0^{n+m} : n, m \geq 0\},$$

$$\{0^p : p \text{ prímszám}\}$$

Itt  $|w|_0$  a  $w$ -ben előforduló 0-ák száma.

## Véges automaták

### Definíció (Véges automata)

$$(Q, \Sigma, \delta, q_0, F)$$

$Q$  : állapotok véges, nemüres halmaza

$\Sigma$  : bemenő jelek (betűk) véges, nemüres halmaza

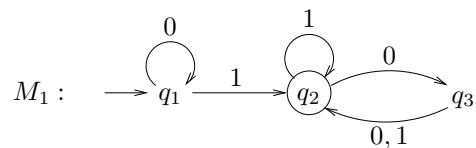
$\delta : Q \times \Sigma \rightarrow Q$  : átmeneti függvény

$q_0 \in Q$  : kezdőállapot

$F \subseteq Q$  : végállapotok halmaza

## Véges automaták

### Véges automaták ábrázolása irányított, címkézett gráffal (átmeneti diagram)



$$Q = \{q_1, q_2, q_3\}$$

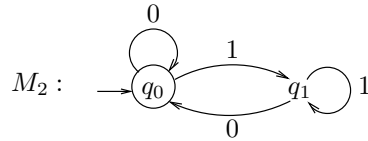
$$\Sigma = \{0, 1\}$$

$$\delta : \begin{array}{c|cc} & 0 & 1 \\ \hline q_1 & q_1 & q_2 \\ q_2 & q_3 & q_2 \\ q_3 & q_2 & q_2 \end{array}$$

Kezdőállapot:  $q_1$

$$F = \{q_2\}$$

## Véges automaták



$$Q = \{q_0, q_1\}$$

$$\Sigma = \{0, 1\}$$

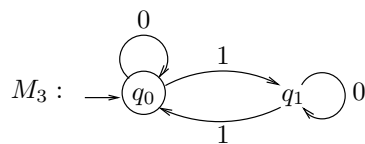
$$\delta :$$

|       | 0     | 1     |
|-------|-------|-------|
| $q_0$ | $q_0$ | $q_1$ |
| $q_1$ | $q_0$ | $q_1$ |

Kezdőállapot:  $q_0$

Végállapothalmaz:  $F = \{q_0\}$

## Véges automaták



$$Q = \{q_0, q_1\}$$

$$\Sigma = \{0, 1\}$$

$$\delta :$$

|       | 0     | 1     |
|-------|-------|-------|
| $q_0$ | $q_0$ | $q_1$ |
| $q_1$ | $q_1$ | $q_0$ |

Kezdőállapot:  $q_0$

$F = \{q_0\}$

## Véges automaták

### Általánosítás

Legyen  $n \geq 1$ .

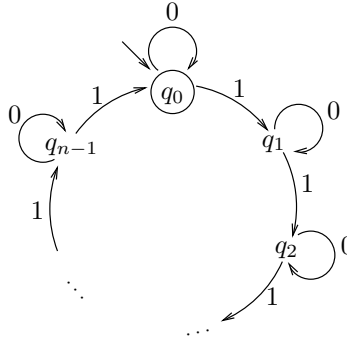
$$Q = \{q_0, \dots, q_{n-1}\}$$

$$\Sigma = \{0, 1\}$$

$$\delta(q_i, j) = q_{i+j \bmod n}$$

Kezdőállapot:  $q_0$

$$F = \{q_0\}$$



## Véges automaták

**Definíció (számítási sorozat)** Legyen  $M = (Q, \Sigma, \delta, q_0, F)$  véges automata,  $q \in Q$ ,  $w = w_1 \dots w_n \in \Sigma^*$  ( $w_i \in \Sigma, i = 1, \dots, n$ ). Azt mondjuk, hogy egy

$$r_0, r_1, \dots, r_n$$

állapotsorozat az  $M$   $q$ -ból induló **számítási sorozata** a  $w$  szón, ha

1.  $r_0 = q$
2.  $r_i = \delta(r_{i-1}, w_i) \quad i = 1, \dots, n$

**Sikeress** az  $r_0, r_1, \dots, r_n$  számítási sorozat, ha  $r_n \in F$ . Azt mondjuk, hogy  $M$  **elfogadja** a  $w$  szót, ha létezik a  $q_0$  kezdőállapotból induló sikeres számítási sorozata a  $w$  szón. Végül az  $M$  által **felismert nyelv**:  $L(M) = \{w \in \Sigma^* : M \text{ elfogadja } w\}$ .



## Felismerhető nyelvek

**Definíció** Egy  $L \subseteq \Sigma^*$  nyelvet **felismerhetőnek** nevezzük, ha létezik olyan  $M$  véges automata, mely  $L$ -et felismeri:  $L = L(M)$ .

**Példa**  $\Sigma = \{0, 1\}$

- $\{w \in \Sigma^* : w \text{ utolsó betűje } 1\} = \{u1 : u \in \Sigma^*\}$
- $\{w \in \Sigma^* : |w|_1 \equiv 0 \pmod{2}\}$
- $\{w \in \Sigma^* : 00 \text{ és } 11 \text{ nem fordulnak elő rész-szóként } w\text{-ben}\}$

felismerhető nyelvek.

**Példa**  $\{0^n 1^n : n \geq 0\}$  nem felismerhető

## $\Sigma^*$ mint monoid

**$\Sigma^*$  mint monoid:**

$u, v \in \Sigma^*$ ,  $u = u_1 \dots u_n, v = v_1 \dots v_m$  konkatenáció  
 $u \cdot v = u_1 \dots u_n v_1 \dots v_m$  művelete

**Tulajdonságok:**

$(u \cdot v) \cdot w = u \cdot (v \cdot w)$  (asszociativitás)  
 $u \cdot \varepsilon = u$  ( $\varepsilon$  egységelem)  
 $\varepsilon \cdot u = u$

Az  $u \cdot v$  helyett általában  $uv$ -t írunk.

## Műveletek nyelveken

Legyenek  $L_1, L_2 \subseteq \Sigma^*$ .

$$\begin{array}{l} L_1 \cup L_2 = \{w \in \Sigma^* : w \in L_1 \text{ vagy } w \in L_2\} \\ L_1 \cap L_2 = \{w \in \Sigma^* : w \in L_1 \text{ és } w \in L_2\} \\ \overline{L_1} = \{w \in \Sigma^* : w \notin L_1\} \end{array} \left. \vphantom{\begin{array}{l} L_1 \cup L_2 \\ L_1 \cap L_2 \\ \overline{L_1} \end{array}} \right\} \begin{array}{l} \text{halmazelméleti} \\ \text{vagy Boole-} \\ \text{féle} \\ \text{műveletek} \end{array}$$
$$\begin{array}{l} L_1 \cdot L_2 = \{uv : u \in L_1, v \in L_2\} \\ L_1^* = \{u_1 \dots u_n : n \geq 0, u_1, \dots, u_n \in L_1\} \end{array} \begin{array}{l} \text{konkatenáció} \\ \text{(Kleene) iteráció} \end{array}$$

$L_1 \cdot L_2$  helyett általában  $L_1L_2$ -t írunk.

**Reguláris műveletek:**

egyesítés, konkatenáció, iteráció

## Műveletek nyelveken

### Példa

- $\{01, 10\} \cdot \{00, 11\} = \{0100, 0111, 1000, 1011\}$
- $\{01\}^* = \{(01)^n : n \geq 0\} = \{w : w\text{-ben } 00 \text{ és } 11 \text{ nem rész-szavak; ha } w \neq \varepsilon, \text{ akkor } w \text{ első betűje } 0, \text{ utolsó betűje } 1\}$
- $\{1, \varepsilon\}\{01\}^*\{0, \varepsilon\} = \{w : 00 \text{ és } 11 \text{ nem rész-szavak}\}$

### Megjegyzés

- $\Sigma$  felfogható a  $\Sigma^*$  részhalmazaként, így alkalmazható rá az iteráció művelete, melynek eredménye a  $\Sigma$ -feletti összes szavak halmaza.
- $\emptyset^* = \{\varepsilon\}$

## Műveletek nyelveken

Néhány azonosság:

$$\begin{array}{ll} L_1 \cup (L_2 \cup L_3) = (L_1 \cup L_2) \cup L_3 & L_1 \cdot (L_2 \cdot L_3) = (L_1 \cdot L_2) \cdot L_3 \\ L_1 \cup L_2 = L_2 \cup L_1 & L \cdot \{\varepsilon\} = L \\ L \cup L = L & \{\varepsilon\} \cdot L = L \\ L \cup \emptyset = L & L \cdot \emptyset = \emptyset \\ & \emptyset \cdot L = \emptyset \end{array}$$

$$\begin{array}{l} L_1 \cdot (L_2 \cup L_3) = (L_1 \cdot L_2) \cup (L_1 \cdot L_3) \\ (L_1 \cup L_2) \cdot L_3 = (L_1 \cdot L_3) \cup (L_2 \cdot L_3) \end{array}$$

## Műveletek nyelveken

$$\begin{array}{l} (L_1 \cup L_2)^* = (L_1^* \cdot L_2)^* L_1^* \\ (L_1 \cdot L_2)^* = \{\varepsilon\} \cup L_1 (L_2 \cdot L_1)^* L_2 \\ L^* = (L^n)^* (\{\varepsilon\} \cup L \cup \dots \cup L^{n-1}) \quad (n \geq 2) \end{array}$$

Rövidítés:

$$\begin{array}{l} L^+ = L \cdot L^* = L^* \cdot L \\ L^n = L \cdot \dots \cdot L \quad (n\text{-szer}), \quad L^0 = \{\varepsilon\} \end{array}$$

Állítás

$$\begin{array}{l} L^* = \cup_{n \geq 0} L^n \\ L^+ = \cup_{n \geq 1} L^n \end{array} \quad \text{tehát } L^* = L^+ \cup \{\varepsilon\}$$

## Reguláris nyelvek

Definíció Egy  $L \subseteq \Sigma^*$  nyelvet **reguláris**nak nevezünk, ha előáll az

$$\emptyset, \{a\} \quad (a \in \Sigma)$$

nyelvekből a három reguláris művelet véges sokszori alkalmazásával.

Tétel (Kleene) Egy nyelv akkor és csak akkor felismerhető, ha reguláris.

## Felismerhető nyelvek zárttsági tulajdonságai I

Állítás A felismerhető nyelvek zártak a hamazelméleti műveletekre:

$$L \text{ felismerhető} \Rightarrow \overline{L} \text{ felismerhető}$$

$$L_1, L_2 \text{ felismerhetőek} \Rightarrow L_1 \cup L_2, L_1 \cap L_2 \text{ felismerhetőek.}$$

### Bizonyítás

**Komplementerképzés:**

$$L = L(M), M = (Q, \Sigma, \delta, q_0, F)$$

$$\text{Legyen } \overline{M} = (Q, \Sigma, \delta, q_0, \overline{F}), \quad \overline{F} = Q - F.$$

Ekkor

$$L(\overline{M}) = \overline{L}.$$

## Felismerhető nyelvek zárttsági tulajdonságai I

Egyesítés és metszetképzés:

$$L_i = L(M_i), \quad M_i = (Q_i, \Sigma, \delta_i, q_i, F_i), \quad i = 1, 2$$

$$\text{Legyen } Q = Q_1 \times Q_2$$

$$\delta : Q \times \Sigma \rightarrow Q, \quad \delta((s_1, s_2), a) = (\delta_1(s_1, a), \delta_2(s_2, a))$$

$$s_1 \in Q_1, \quad s_2 \in Q_2, \quad a \in \Sigma$$

$$q_0 = (q_1, q_2)$$

$$F_{\cup} = F_1 \times Q_2 \cup Q_1 \times F_2$$

$$F_{\cap} = F_1 \times F_2$$

$$M_{\cup} = (Q, \Sigma, \delta, q_0, F_{\cup})$$

$$M_{\cap} = (Q, \Sigma, \delta, q_0, F_{\cap})$$

Ekkor:

$$L(M_{\cup}) = L_1 \cup L_2$$

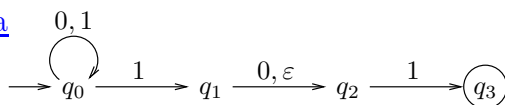
$$L(M_{\cap}) = L_1 \cap L_2$$

## Véges nemdeterminisztikus automata

Véges nemdeterminisztikus automata:

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

Példa



$$Q = \{q_0, q_1, q_2, q_3\}$$

$$\Sigma = \{0, 1\}$$

kezdőállapot:  $q_0$

végállapotok:  $q_3$

| $\delta$ | 0           | 1              | $\varepsilon$ |
|----------|-------------|----------------|---------------|
| $q_0$    | $\{q_0\}$   | $\{q_0, q_1\}$ | $\emptyset$   |
| $q_1$    | $\{q_2\}$   | $\emptyset$    | $\{q_2\}$     |
| $q_2$    | $\emptyset$ | $\{q_3\}$      | $\emptyset$   |
| $q_3$    | $\emptyset$ | $\emptyset$    | $\emptyset$   |

## Véges nemdeterminisztikus automata

Definíció Véges nemdeterminisztikus (üres átmenetekkel ellátott) automata:

$$(Q, \Sigma, \delta, q_0, F)$$

rendszer, ahol  $Q, \Sigma, q_0, F$  ugyanazok, mint véges automatában, továbbá

$$\delta : Q \times \Sigma_\varepsilon \rightarrow P(Q)$$

leképezés, ahol

$$\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$$

$$P(Q) = \{Q' : Q' \subseteq Q\} \quad (Q \text{ hatványhalmaza})$$

## Véges nemdeterminisztikus automata

Definíció (számítási sorozat) Legyen  $M = (Q, \Sigma, \delta, q_0, F)$  véges nemdeterminisztikus automata,  $w \in \Sigma^*, q \in Q$ . Azt mondjuk, hogy egy

$$r_0, r_1, \dots, r_n \quad (n \geq 0)$$

állapotsorozat az  $M$   $q$ -ból induló számítási sorozata a  $w$  szón, ha  $w$  felírható

$$w = w_1 \dots w_n, \quad w_i \in \Sigma_\varepsilon, \quad i = 1, \dots, n$$

alakban úgy, hogy

$$r_0 = q \text{ és } r_i \in \delta(r_{i-1}, w_i) \quad i = 1, \dots, n.$$

**Siker** az  $r_0, r_1, \dots, r_n$  számítási sorozat, ha  $r_n \in F$ . Továbbá  $M$  **elfogadja** a  $w$  szót, ha létezik a  $q_0$  kezdőállapotból induló sikeres számítási sorozata a  $w$  szón. Végül az  $M$  által **felismert nyelv**:  $L(M) = \{w \in \Sigma^* : M \text{ elfogadja } w\text{-t}\}$



## Véges nemdeterminisztikus automata

$$M' = P(M) = (P(Q), \Sigma, \delta', Q_0, \mathcal{F})$$

$$\delta' : P(Q) \times \Sigma \rightarrow P(Q) \quad \delta'(X, a) = \widehat{Y}, Y = \cup_{q \in X} \delta(q, a)$$

$$Q_0 = \{\widehat{q_0}\}$$

$$\mathcal{F} = \{X \subseteq Q : X \cap F \neq \emptyset\}$$

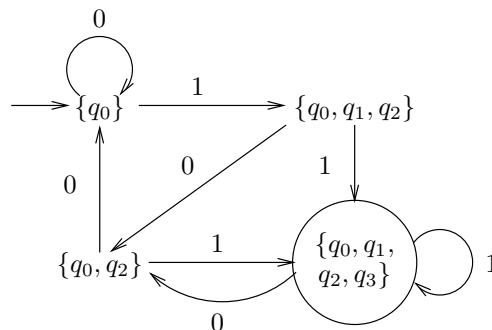
Ekkor

$$L(M') = L(M).$$

$|P(Q)| = 2^{|Q|}$ . Elegendő azonban  $M'$  „összefüggő részét” venni.

## Véges nemdeterminisztikus automata

Példa (A korábbi véges nemdeterminisztikus automata determinizálása)



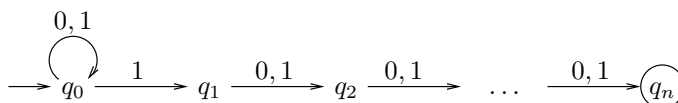


## Véges nemdeterminisztikus automata

Állítás Adott  $n \geq 1$  esetén legyen  $L_n = \{0,1\}^* \cdot \{1\} \cdot \{0,1\}^{n-1} = \{w \in \{0,1\}^* : |w| \geq n \text{ és } w \text{ hátulról } n\text{-ik jele } 1\}$ .

Ekkor  $L_n$  felismerhető  $n + 1$  állapotú nemdeterminisztikus automatával, de minden  $L_n$ -et felismerő (determinisztikus) automatának legalább  $2^n$  állapota van.

### Bizonyítás



felismeri  $L_n$ -et.

## Véges nemdeterminisztikus automata

Tegyük fel, hogy  $M = (Q, \{0,1\}, \delta, q_0, F)$  véges (determinisztikus) automata mely felismeri  $L_n$ -et. Minden  $u \in \{0,1\}^*$  szóra jelölje  $q_0u$  azt az állapotot, melyre a  $q_0$ -ból induló számítási sorozat végződik az  $u$  szón.

Ha  $u \neq v$   $n$ -hosszú szavak  $\{0,1\}^*$ -ban, akkor  $q_0u \neq q_0v$ .

$$\begin{array}{l} u \quad \frac{x}{\quad} \quad \frac{0}{\quad} \quad \frac{y}{\quad} \\ v \quad \frac{x'}{\quad} \quad \frac{1}{\quad} \quad \frac{y'}{\quad} \end{array}$$

$$|y| = |y'| = n - i - 1$$

Ha  $q_0u = q_0v$ , akkor  $q_0u0^i = q_0v0^i$ , így  
 $u0^i \in L_n \Leftrightarrow v0^i \in L_n$ , **ellentmondás**.

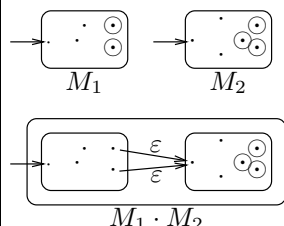
$\Rightarrow |Q| \geq 2^n$ .

## Felismerhető nyelvek zártsági tulajdonságai II

Állítás A felismerhető nyelvek osztálya zárt a konkatenációra:

$$L_1, L_2 \text{ felismerhetők} \Rightarrow L_1 \cdot L_2 \text{ felismerhető}$$

Bizonyítás



$$M_i = (Q_i, \Sigma, \delta_i, q_i, F_i) \quad i = 1, 2$$

$$Q_1 \cap Q_2 = \emptyset \quad L(M_i) = L_i$$

$$M_1 \cdot M_2 = (Q_1 \cup Q_2, \Sigma, \delta, q_1, F_2)$$

$$\delta(q, a) = \begin{cases} \delta_1(q, a) & q \in Q_1 - F_1 \\ \delta_1(q, a) & q \in F_1, a \neq \varepsilon \\ \delta_1(q, a) \cup \{q_2\} & q \in F_1, a = \varepsilon \\ \delta_2(q, a) & q \in Q_2 \end{cases}$$

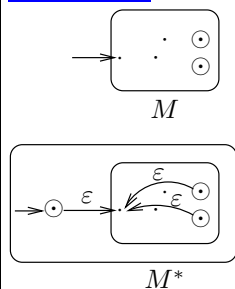
$$L(M_1 \cdot M_2) = L_1 \cdot L_2$$

## Felismerhető nyelvek zártsági tulajdonságai II

Állítás A felismerhető nyelvek osztálya zárt a Kleene-féle iterációra:

$$L \text{ felismerhető} \Rightarrow L^* \text{ felismerhető}$$

Bizonyítás



$$M = (Q, \Sigma, \delta, q_0, F) \quad L(M) = L$$

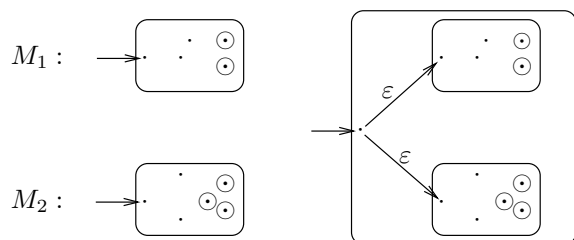
$$M^* = (Q \cup \{s_0\}, \Sigma, \delta_*, s_0, F \cup \{s_0\})$$

$$\delta_*(q, a) = \begin{cases} \delta(q, a) & q \in Q \text{ és } q \notin F \\ \delta(q, a) & q \in F \text{ és } a \neq \varepsilon \\ \delta(q, a) \cup \{q_0\} & q \in F \text{ és } a = \varepsilon \\ \{q_0\} & q = s_0 \text{ és } a = \varepsilon \\ \emptyset & q = s_0 \text{ és } a \neq \varepsilon \end{cases}$$

$$L(M^*) = L^*$$

## Felismerhető nyelvek zártági tulajdonságai II

Megjegyzés Nemdeterminisztikus automatákat felhasználva új, egyszerűbb bizonyítás adható arra, hogy a felismerhető nyelvek zártak az egyesítésre.



$$L(M_1 \cup M_2) = L(M_1) \cup L(M_2)$$

## Reguláris kifejezések

Definíció Legyen  $\Sigma$  véges, nemüres halmaz. Azt mondjuk, hogy  $R$  **reguláris kifejezés** ( $\Sigma$  felett), ha:

1.  $R = a$  valamely  $a \in \Sigma$ -ra, és ekkor  $R$  az  $\{a\}$  nyelvet jelöli, vagy
2.  $R = \emptyset$  és ekkor  $R$  az  $\emptyset$  nyelvet jelöli, vagy
3.  $R = (R_1 + R_2)$  és ekkor  $R$  az  $R_1$  és  $R_2$  által jelölt nyelvek egyesítését jelöli, vagy
4.  $R = (R_1 \cdot R_2)$  és ekkor  $R$  az  $R_1$  és  $R_2$  által jelölt nyelvek konkatenációját jelöli, vagy
5.  $R = (R_1^*)$  és ekkor  $R$  az  $R_1$  által jelölt nyelv iterációját jelöli,

ahol  $R_1, R_2$  reguláris kifejezések.

## Reguláris kifejezések

Megjegyzés A felesleges zárójeleket elhagyjuk és megegyezünk abban, hogy  $*$  erősebben köt, mint  $\cdot$ , ami erősebben köt, mint  $+$ . A  $\cdot$  jelet általában elhagyjuk. **Rövidítés:**  $\emptyset^*$  helyett  $\varepsilon$ -t írunk.

### Példa

- $0(01 + 10)1 + \varepsilon$   $\{\varepsilon, 0011, 0101\}$
- $0(0 + 1)^*1$   $\{w : w \text{ 0-val kezdődik, 1-el végződik}\}$
- $(0 + \varepsilon)(10)^*(1 + \varepsilon)$   $\{w : 00 \text{ és } 11 \text{ nem rész-szavak}\}$
- $(0^2)^*$   $\{w : w \text{ páros hosszú és csak 0-t tartalmaz}\}$
- $(0^*10^*1)^*0^*$   $\{w : w \text{ páros sok 1-est tartalmaz}\}$

Állítás Egy  $L \subseteq \Sigma^*$  nyelv akkor és csakis akkor reguláris, ha jelölhető reguláris kifejezéssel:  $L = |R|$  valamely  $R$  reguláris kifejezésre.

## Kleene tétele

Állítás Minden reguláris nyelv felismerhető

Bizonyítás Legyen  $R$  egy  $\Sigma$  feletti reguláris kifejezés. Az  $R$  felépítése szerinti indukcióval igazoljuk, hogy  $|R| \subseteq \Sigma^*$  felismerhető.

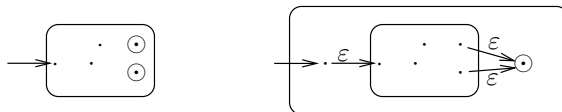
- $R = a, a \in \Sigma$   $\rightarrow \cdot \xrightarrow{a} \odot$
- $R = \emptyset$   $\rightarrow \cdot$
- $R = (R_1 + R_2)$  Ekkor  $|R| = |R_1| \cup |R_2|$  és az állítás következik az indukciós feltevésből és abból, hogy a felismerhető nyelvek zártak az egyesítésre.
- $R = (R_1 \cdot R_2)$  Indukciós feltevés + konkatenációra való zártság.
- $R = (R_1^*)$  Indukciós feltevés + iterációra való zártság.

## Kleene tétele

**Lemma** Minden felismerhető nyelv felismerhető olyan véges nondeterminisztikus automatával, melynek pontosan egy végállapota van, a végállapotból nem indul átmenet, a kezdőállapotba nem vezet átmenet, továbbá a kezdőállapot különbözik a végállapottól:

$$\begin{aligned} M &= (Q, \Sigma, \delta, q_0, \{q_f\}) \\ \delta(q_f, a) &= \emptyset \quad a \in \Sigma_\epsilon \\ q_0 &\notin \delta(q, a) \quad q \in Q, a \in \Sigma_\epsilon \end{aligned}$$

### Bizonyítás



## Kleene tétele

A továbbiakban olyan véges irányított gráfokat fogunk tekinteni, melyeknek:

1. Ki van jelölve egy  $q_0$  „bemenő” csúcsa.
2. Ki van jelölve egy  $q_f$  „kimenő” csúcsa,  $q_0 \neq q_f$ .
3.  $q_0$ -ba nem vezet él és  $q_f$ -ből nem indul él,
4. ettől eltekintve bármely két  $q_1, q_2$  csúcsra pontosan egy  $q_1$ -ből  $q_2$ -be vezető él van.
5. Minden él egy reguláris kifejezéssel van címkézve.

A példákban nem tüntetjük fel az  $\emptyset$ -zal címkézett éleket. A  $q_0$ -tól és  $q_f$ -től különböző csúcsokat **belső csúcsok**nak nevezzük.

## Kleene tétele

Állítás Minden felismerhető nyelv reguláris.

Bizonyítás Legyen  $L = L(M)$ ,  $M = (Q, \Sigma, \delta, q_0, \{q_f\})$  mint az előző lemmában.

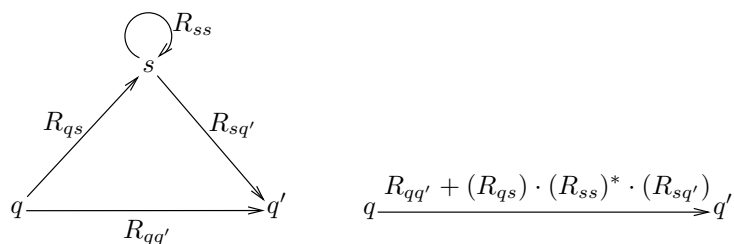
- $M$ -et felfoghatjuk úgy, mint címkézett irányított gráfot. Adott  $q, q'$ -re ( $q \neq q_f, q' \neq q_0$ ) a  $q \rightarrow q'$  él címkéje:

$$\Sigma(a : q' \in \delta(q, a))$$

- Ha a gráfnak nincs belső csúcsa, a  $q_0 \rightarrow q_f$  él címkéje olyan reguláris kifejezés, mely az  $L$ -et jelöli.

## Kleene tétele

- **Redukciós lépés.** Ha a belső csúcsok száma pozitív, akkor 1-gyel csökkentjük ezek számát mindaddig, amíg van belső csúcs. Legyen  $s$  belső csúcs.



**Tehát:** elhagyjuk az  $s$  csúcsot, és minden  $q, q'$ -re ( $q \neq q_f, q' \neq q_0$ ) a  $q \rightarrow q'$  címkéjét

$$R_{qq'}\text{-ről } R_{qq'} + (R_{qs}) \cdot (R_{ss})^* \cdot (R_{sq'})\text{-re}$$

változtatjuk.

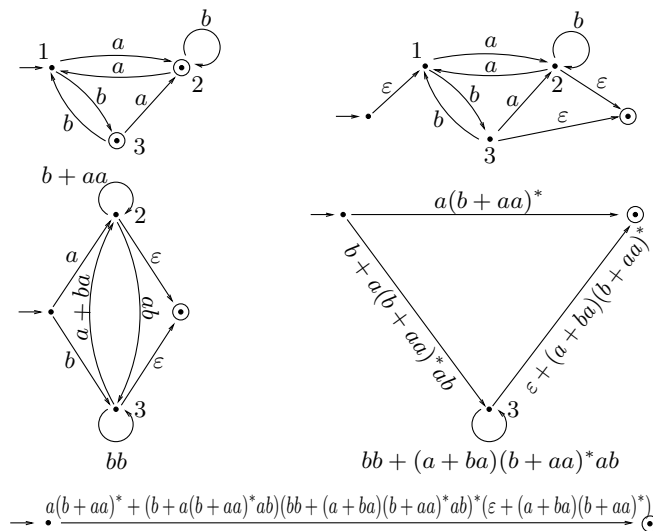
## Kleene tétele

**Indukcióval bizonyítható:** Minden lépésben egy  $q \rightarrow q'$  él címkéje azt a nyelvet jelöli, mely az összes olyan szóból áll, amely mentén  $q$ -ból el lehet  $q'$ -be jutni úgy, hogy minden közbülső állapot már korábban elhagyott.

**Megjegyzés** Az eljárás gyorsítható azzal, hogy kezdetben elhagyjuk az összes olyan csúcsot,

1. amely nem érhető el  $q_0$ -ból, vagy
2. amelyből nem érhető el  $q_f$ .

## Kleene tétele



## Reguláris nyelvek pumpáló lemmája

Minden  $L \subseteq \Sigma^*$  reguláris nyelvhez létezik olyan  $p$  szám, hogy valahányszor az  $u \in L$  szó hossza  $\geq p$ ,  $u$  felírható

$$u = xyz$$

alakban úgy, hogy

1.  $xy^iz \in L$  minden  $i \geq 0$  számra,
2.  $|y| > 0$ ,
3.  $|xy| \leq p$ .

## Reguláris nyelvek pumpáló lemmája

Bizonyítás  $L = L(M)$   $M = (Q, \Sigma, \delta, q_0, F)$  véges automata.

Legyen  $p = |Q|$ . Ha  $u \in \Sigma^*$ ,  $|u| \geq p$  és  $u \in L$ , akkor a  $q_0$ -ból induló  $q_0, q_1, \dots, q_n$  számítási sorozatra az  $u$  szón teljesül, hogy:

1.  $|u| = n \geq p$ ,
2.  $q_n \in F$ ,
3.  $\exists i, j, \quad 0 \leq i < j \leq p, \quad q_i = q_j$ .

Legyen  $x$  az  $u$   $i$  hosszú kezdőszelete,  $y$  az  $x$ -et követő  $j - i$  hosszú rész-szó,  $z$  az  $u$   $n - j$  hosszú zárószelete. Ekkor az  $u = xyz$  felbontásra teljesülnek a Lemma állításai.



## Reguláris nyelvek pumpáló lemmája

**Példa** Az  $L = \{0^n 1^n : n \geq 0\} \subseteq \{0, 1\}^*$  nyelv nem reguláris.

**Bizonyítás** Belátjuk, hogy

$$\forall p \quad \exists u \in L, |u| \geq p \quad \forall x, y, z$$

$$u = xyz, \quad |xy| \leq p, \quad |y| > 0 \quad \Rightarrow \quad \exists i \quad xy^i z \notin L.$$

Legyen  $p$  tetszőleges. Tekintsük az  $u = 0^p 1^p$  szót. Tfh.  $x, y, z$  olyan szavak, melyekre  $u = xyz$ ,  $|xy| \leq p$  és  $|y| > 0$ . Ekkor  $xy$  csupa 0-ból áll, és  $y$  tartalmaz legalább egy 0-át. Ha tehát  $i \neq 1$ , akkor az  $xy^i z$  szóban a 0-ák száma különbözik az 1-ek számától, tehát  $xy^i z \notin L$ .

## Környezetfüggetlen nyelvtanok

**Környezetfüggetlen nyelvtanok:** Chomsky, ~ 1960

Természetes nyelvek struktúrája

Programozási nyelvek

**Nyelvek rekurzív megadása:**

$$L = \{0^n 1^n : n \geq 0\}$$

- $\varepsilon \in L$
- $u \in L \Rightarrow 0u1 \in L$

$$L = \{u \in \{(, )\}^* : u \text{ helyesen zárójelezett}\}$$

- $\varepsilon \in L$
- $u \in L \Rightarrow (u) \in L$
- $u, v \in L \Rightarrow uv \in L$ .

## Környezetfüggetlen nyelvtanok

Definíció Környezetfüggetlen nyelvtan egy  $G = (V, \Sigma, R, S)$  rendszer, ahol

$V$  véges nemüres halmaz: **változók vagy nemterminálisok** halmaza

$\Sigma$  véges nemüres halmaz: **terminálisok** halmaza,  $V \cap \Sigma = \emptyset$

$R$ :  $A \rightarrow w$  alakú **szabályok** véges halmaza, ahol

$$A \in V, w \in (V \cup \Sigma)^*$$

$S \in V$  a **kezdőszimbólum**.

## Környezetfüggetlen nyelvtanok

Példa

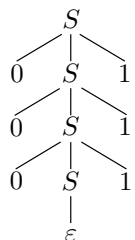
$$\begin{aligned} S &\rightarrow \varepsilon \\ S &\rightarrow 0S1 \end{aligned} \quad (S \rightarrow 0S1 \mid \varepsilon)$$

$$V = \{S\}, \Sigma = \{0, 1\},$$

$$R = \{S \rightarrow \varepsilon, S \rightarrow 0S1\},$$

$$S \Rightarrow 0S1 \Rightarrow 00S11 \Rightarrow 000S111 \Rightarrow 000111$$

kezdőszimbólum:  $S$   
deriváció



derivációs fa

## Környezetfüggetlen nyelvtanok

**Definíció** Legyen  $G = (V, \Sigma, R, S)$  környezetfüggetlen nyelvtan,  $u, v \in (V \cup \Sigma)^*$ .

1. Azt mondjuk, hogy  $u$  **közvetlenül deriválja** a  $v$  szót,  $u \Rightarrow v$ , ha létezik az  $u = u_1 A u_2$  és  $v = u_1 w u_2$  felbontás úgy, hogy  $A \rightarrow w \in R$ .

2. Egy  $u_0, u_1, \dots, u_n$  ( $n \geq 0$ ) sorozatot a  $v$  szó  $u$ -ból való **derivációjának** nevezünk, ha

$$u_0 = u, u_n = v$$

$$u_{i-1} \Rightarrow u_i \quad i = 1, \dots, n$$

3. Azt mondjuk, hogy a  $v$  **deriválható vagy levezethető  $u$ -ból**, ha létezik a  $v$ -nek  $u$ -ból való derivációja:  $u \Rightarrow^* v$ .

4. A  $G$  által generált nyelv:

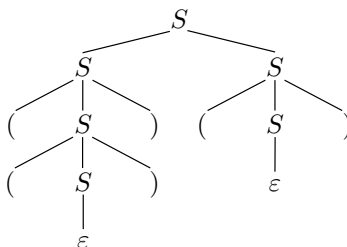
$$L(G) = \{w \in \Sigma^* : S \Rightarrow^* w\}$$

## Környezetfüggetlen nyelvtanok

### Példa

$$S \rightarrow SS \mid (S) \mid \varepsilon$$

$$S \Rightarrow SS \Rightarrow (S)S \Rightarrow ((S))S \Rightarrow (()S \Rightarrow (())(S) \Rightarrow (()())$$





## Derivációs fák


Állítás Ha  $X \Rightarrow^* u \in \Sigma^*$ , akkor létezik olyan  $X$ -ből induló derivációs fa, mely határa  $u$ .

Bizonyítás (vázlat)

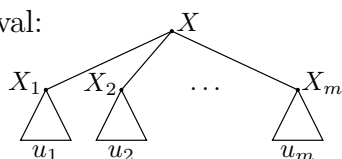
$X = v_0 \Rightarrow v_1 \Rightarrow \dots \Rightarrow v_n = u$ .

$n = 0$ . Ekkor  $X = u \in \Sigma \cup \{\varepsilon\}$ .  $\cdot u$

$n > 0$ . Legyen  $v_1 = X_1 \dots X_m$ . Ekkor  $u$  felbontható  $u_1 \dots u_m$  alakban úgy, hogy minden  $i$ -re  $X_i \Rightarrow^* u_i$  kevesebb, mint  $n$  lépésben.

Így léteznek  derivációs fák.

Ezek felhasználásával:



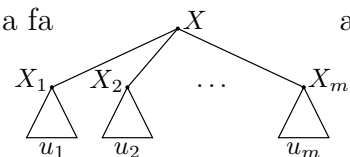
## Derivációs fák

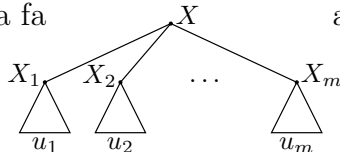
Állítás Ha létezik olyan  $X$ -ből induló derivációs fa, melynek határa az  $u \in \Sigma^*$  szó, akkor  $X \Rightarrow^* u$ .

Bizonyítás (vázlat)

A derivációs fa  $n$  mélysége szerinti indukcióval.

$n = 0$ . Ekkor  $X = u \in \Sigma \cup \{\varepsilon\}$ . Nyilván  $X \Rightarrow^* u$ .

$n > 0$ . Ekkor a fa  alakú.



Az indukciós feltevés szerint  $X_i \Rightarrow^* u_i, i = 1, \dots, m$ . Így  $X \Rightarrow X_1 \dots X_m \Rightarrow^* u_1 X_2 \dots X_m \Rightarrow^* u_1 u_2 X_3 \dots X_m \Rightarrow^* u_1 u_2 \dots u_m = u$ .

## Baloldali deriváció

Megjegyzés Az előző konstrukcióval **baloldali deriváció**hoz jutunk. Azt mondjuk, hogy egy

$$u_0 \Rightarrow u_1 \Rightarrow \dots \Rightarrow u_n$$

deriváció baloldali, ha minden  $i < n$  számra  $u_{i+1}$  úgy áll elő az  $u_i$  szóból, hogy az  $u_i$ -ben előforduló első nemterminálist írjuk át. **Jobboldali derivációk** hasonlóan definiálhatóak.

**Baloldali deriváció jelölése:**

$$u_0 \Rightarrow_l u_1 \Rightarrow_l \dots \Rightarrow_l u_n, \quad \text{vagy} \quad u_0 \Rightarrow_l^* u_n$$

## Derivációs fák és baloldali derivációk

Következmény Legyen  $G = (V, \Sigma, R, S)$  környezetfüggetlen nyelvtan. A következők ekvivalensek egy  $u \in \Sigma^*$  szóra:

1.  $u \in L(G)$
2.  $S \Rightarrow_l^* u$
3. Létezik olyan  $S$ -ből induló derivációs fa, melynek határa  $u$ .

Megjegyzés Egy  $u \in \Sigma^*$  szónak az  $S$ -ből induló derivációs fái és az  $S$ -ből induló baloldali derivációi között kölcsönösen egyértelmű kapcsolat van. A  $G = (V, \Sigma, R, S)$  nyelvtant **egyértelműnek** nevezzük, ha minden  $u \in L(G)$  szónak pontosan egy  $S$ -ből induló baloldali levezetése (derivációs fája) van.

## Környezetfüggetlen nyelvek

Definíció Egy  $L \subseteq \Sigma^*$  nyelvet **környezetfüggetlennek** nevezünk, ha létezik olyan  $G$  környezetfüggetlen nyelvtan, melyre  $L = L(G)$ .

Állítás Minden reguláris nyelv környezetfüggetlen.

### Bizonyítás

$L \subseteq \Sigma^*$ ,  $L = L(M)$ ,  $M = (Q, \Sigma, \delta, q_0, F)$  nondeterminisztikus véges automata.

$G = (Q, \Sigma, R, q_0)$

$R = \{q \rightarrow aq' : q' \in \delta(q, a)\} \cup \{q \rightarrow \varepsilon : q \in F\}$

Ekkor  $L = L(G)$

A megadott nyelvtan jobblineáris.

## Környezetfüggetlen nyelvek

### Másik bizonyítás

$E$  reg. kifejezés  $\longrightarrow G_E = (V_E, \Sigma, R_E, S_E)$  nyelvtan

- $a \in \Sigma$   $G_a = (\{S\}, \Sigma, \{S \rightarrow a\}, S); G_\emptyset = (\{S\}, \Sigma, \emptyset, S)$
- $E = (E_1 + E_2)$   $G_E = (V_{E_1} \cup V_{E_2} \cup \{S\}, \Sigma, R_{E_1} \cup R_{E_2} \cup \{S \rightarrow S_{E_1}, S \rightarrow S_{E_2}\}, S)$   
feltesszük, hogy  $V_{E_1} \cap V_{E_2} = \emptyset, S \notin V_{E_1} \cup V_{E_2}$
- $E = (E_1 \cdot E_2)$   $G_E = (V_{E_1} \cup V_{E_2} \cup \{S\}, \Sigma, R_{E_1} \cup R_{E_2} \cup \{S \rightarrow S_{E_1}S_{E_2}\}, S),$   
 $V_{E_1} \cap V_{E_2} = \emptyset, S \notin V_{E_1} \cup V_{E_2}$
- $E = (E_1)^*$   $G_E = (V_{E_1} \cup \{S\}, \Sigma, R_{E_1} \cup \{S \rightarrow SS_{E_1}, S \rightarrow \varepsilon\}, S), S \notin V_{E_1}$

## Környezetfüggetlen nyelvek

Tétel A környezetfüggetlen nyelvek zártak a reguláris műveletekre.

Bizonyítás Lásd az előző konstrukciókat.

Megjegyzés A környezetfüggetlen nyelvek nem zártak a komplement és metszet képzésre.

## Chomsky normálforma

Definíció Egy  $G = (V, \Sigma, R, S)$  környezetfüggetlen nyelvtan **Chomsky normál formában adott**, ha  $R$  minden szabálya

$$A \rightarrow BC \text{ vagy } A \rightarrow a \text{ alakú,}$$

ahol  $A, B, C \in V, a \in \Sigma$ , esetleg az  $S \rightarrow \varepsilon$  szabály kivételével, de ekkor  $S$  nem fordul elő egyetlen szabály jobboldalán sem.

Tétel Minden környezetfüggetlen nyelv generálható olyan környezetfüggetlen nyelvtannal, mely Chomsky normál formában adott.

1.  $\varepsilon$  jobboldalú szabályok eliminálása.
2. Láncszabályok eliminálása.
3. Jobboldalak átalakítása.



## Chomsky normálforma

Legyen  $G = (V, \Sigma, R, S)$ .

1.1. Meghatározzuk azon  $A \in V$  nemterminálisok halmazát, amelyekre  $A \Rightarrow^* \varepsilon$ .

–  $V_0 := \{A \in V : A \rightarrow \varepsilon \in R\}$

– Mindaddig, amíg van olyan  $A \rightarrow w$  szabály, melyre  $A \in V - V_0$  és  $w \in V_0^+$ , legyen  $V_0 := V_0 \cup \{A\}$ .

Állítás  $A \in V_0 \Leftrightarrow A \Rightarrow^* \varepsilon$ .

1.2. Elhagyjuk az összes  $A \rightarrow \varepsilon$  alakú szabályt, továbbá minden  $A \rightarrow w$ ,  $w \in (V \cup \Sigma)^+$  alakú szabályt helyettesítünk az összes olyan  $A \rightarrow w'$  szabály halmazával, ahol  $w' \neq \varepsilon$  úgy áll elő  $w$ -ből, hogy  $w$ -ben törölünk minden lehetőség szerint néhány  $V_0$ -beli nemterminálist.

## Chomsky normálforma

1.3. Amennyiben  $S \in V_0$ , felvesszünk egy új  $S_0$  nemterminálist és az  $S_0 \rightarrow \varepsilon, S_0 \rightarrow S$  szabályokat.  $S_0$  lesz az új kezdőszimbólum. Ha  $S \notin V_0$ , akkor  $S$  marad a kezdőszimbólum.

Állítás Az így előálló  $G' = (V', \Sigma, R', S')$  nyelvtanra teljesülnek az alábbiak:

(a)  $L(G) = L(G')$

(b)  $R'$  nem tartalmaz  $\varepsilon$  jobboldalú szabályt az esetleges  $S' \rightarrow \varepsilon$  szabály kivételével, amikor is  $S'$  nem fordul elő szabály jobboldalán.

## Chomsky normálforma

Tekintsük a  $G' = (V', \Sigma, R', S')$  nyelvtant.

- 2.1. Minden  $A \in V'$  nemterminálisra határozzuk meg az összes olyan  $B \in V'$  nemterminális halmazát, amelyre  $A \Rightarrow^* B$  teljesül.
- 2.2. Hagyjuk el az összes  $A \rightarrow B$ ,  $A, B \in V'$  láncszabályt, majd vegyük az összes olyan  $A \rightarrow w$ ,  $w \notin V'$  szabályt amelyekre létezik olyan  $B$ , hogy

$$A \Rightarrow^* B \text{ és } B \rightarrow w \in R'.$$

Így kapjuk a  $G'' = (V'', \Sigma, R'', S'')$  nyelvtant.

Állítás  $L(G) = L(G'')$ . Továbbá  $G''$ -re teljesül, hogy láncszabály mentes, és egy esetleges  $S'' \rightarrow \varepsilon$  szabály kivételével minden szabály jobboldala  $\varepsilon$ -tól különböző. Ha  $S'' \rightarrow \varepsilon$  szabály, akkor  $S''$  nem fordul elő szabály jobboldalán.

## Chomsky normálforma

Tekintsük a  $G'' = (V'', \Sigma, R'', S'')$  nyelvtant.

- 3.1. Minden  $a \in \Sigma$ -ra vegyünk fel egy  $X_a$  nemterminálist és az  $X_a \rightarrow a$  szabályt.
- 3.2. Minden  $R''$ -ben lévő nem  $A \rightarrow a$  alakú szabály jobboldalán az  $a$  terminális minden egyes előfordulását helyettesítsük  $X_a$ -val.
- 3.3. Majd minden egyes így átalakított szabályra, mely jobboldalának hossza  $> 2$ , végezzük el az alábbiakat.

- Legyen a szabály  $A \rightarrow A_1 \dots A_n$  ( $n > 2$ )
- Vezessük be az  $A \rightarrow A_1 B_1$ ,  $B_1 \rightarrow A_2 B_2$ ,  $\dots$ ,  $B_{n-2} \rightarrow A_{n-1} A_n$  szabályokat, ahol  $B_1, \dots, B_{n-2}$  új nemterminálisok.
- Az  $A \rightarrow A_1 \dots A_n$  szabályt cseréljük ki az új szabályokkal.

## Chomsky normálforma

Állítás A  $G''$ -ből így kapott  $\bar{G} = (\bar{V}, \Sigma, \bar{R}, \bar{S})$  nyelvtan Chomsky normál formában van és  $L(G) = L(\bar{G})$ .

### Megjegyzés

- Ha  $G = (V, \Sigma, R, S)$  Chomsky normál formában van, akkor  $\varepsilon \in L(G) \Leftrightarrow S \rightarrow \varepsilon \in R$ .
- A bizonyítás algoritmust ad arra, hogyan konvertáljunk egy tetszőleges környezetfüggetlen nyelvtant Chomsky normál formába.

## Chomsky normálforma

### Példa

$$\begin{array}{l} S \rightarrow ASA | aB \\ A \rightarrow B | S \\ B \rightarrow b | \varepsilon \end{array} \quad V_0 = \{A, B\}$$

$$\begin{array}{l} S \rightarrow ASA | AS | SA | S | aB | a \\ A \rightarrow B | S \\ B \rightarrow b \end{array} \quad \begin{array}{l} S \Rightarrow^* S \quad A \Rightarrow^* B \\ A \Rightarrow^* S \end{array}$$

$$\begin{array}{l} S \rightarrow ASA | AS | SA | aB | a \\ A \rightarrow b | ASA | AS | SA | aB | a \\ B \rightarrow b \end{array}$$

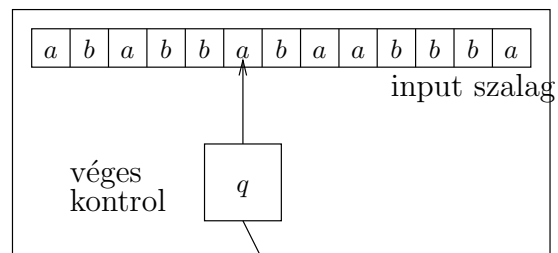
## Chomsky normálforma

$$\begin{aligned}
 S &\rightarrow ASA \mid AS \mid SA \mid X_a B \mid a \\
 A &\rightarrow b \mid ASA \mid AS \mid SA \mid X_a B \mid a \\
 X_a &\rightarrow a \\
 B &\rightarrow b
 \end{aligned}$$

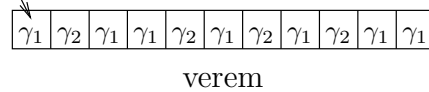
$$\begin{aligned}
 S &\rightarrow AY \mid AS \mid SA \mid X_a B \mid a \\
 Y &\rightarrow SA \\
 A &\rightarrow b \mid AY \mid AS \mid SA \mid X_a B \mid a \\
 X_a &\rightarrow a \\
 B &\rightarrow b
 \end{aligned}$$

## Veremautomaták

Véges automata:



Veremautomata:



## Veremautomaták

**Definíció** Veremautomata egy  $(Q, \Sigma, \Gamma, \delta, q_0, F)$  rendszer, ahol  $Q, \Sigma, q_0, F$  ugyanazok, mint véges automata esetén,

$\Gamma$  véges, nemüres halmaz, a **verem abc**,

$\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \rightarrow P(Q \times \Gamma_\varepsilon)$  az **átmenetfüggvény**.

Az  $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$  veremautomata a következő módon működik. Akkor fogadja el a  $w \in \Sigma^*$  szót, ha

$\exists w_1, \dots, w_m \in \Sigma_\varepsilon, r_0, \dots, r_m \in Q, s_0, \dots, s_m \in \Gamma^*$

1.  $w = w_1 \dots w_m$
2.  $r_0 = q_0, s_0 = \varepsilon$
3.  $\forall i < m (r_{i+1}, b) \in \delta(r_i, w_{i+1}, a)$  ahol  $s_i = at$  és  $s_{i+1} = bt$   
valamely  $a, b \in \Gamma_\varepsilon, t \in \Gamma^*$  esetén.
4.  $r_m \in F$ .

Dr. Ésik Zoltán

A számítástudomány alapjai — slide #68

## Veremautomaták

**Jelölés**  $L(M)$  jelöli az  $M$  által elfogadott  $w \in \Sigma^*$  szavak halmazát.  $L(M)$ -et az  **$M$  által felismert nyelv**nek nevezzük.

**Példa**  $L = \{0^n 1^n : n \geq 0\}$

$Q = \{q_1, q_2, q_3, q_4\}$   $\Sigma = \{0, 1\}$   $\Gamma = \{0, \$\}$  kezdőáll.:  $q_1$

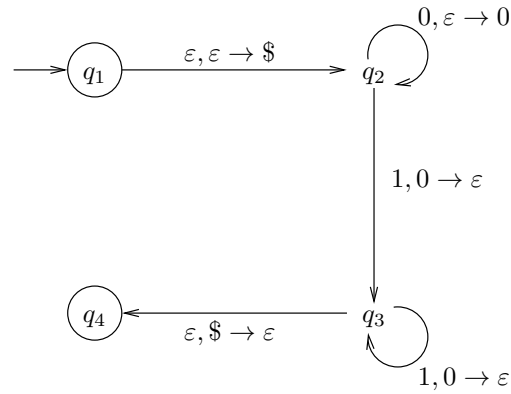
$F = \{q_1, q_4\}$

| Input | 0 |    |                | 1                        |    |               | $\varepsilon$ |    |                          |
|-------|---|----|----------------|--------------------------|----|---------------|---------------|----|--------------------------|
| Verem | 0 | \$ | $\varepsilon$  | 0                        | \$ | $\varepsilon$ | 0             | \$ | $\varepsilon$            |
| $q_1$ |   |    |                |                          |    |               |               |    | $\{(q_2, \$)\}$          |
| $q_2$ |   |    | $\{(q_2, 0)\}$ | $\{(q_3, \varepsilon)\}$ |    |               |               |    |                          |
| $q_3$ |   |    |                | $\{(q_3, \varepsilon)\}$ |    |               |               |    | $\{(q_4, \varepsilon)\}$ |
| $q_4$ |   |    |                |                          |    |               |               |    |                          |

Dr. Ésik Zoltán

A számítástudomány alapjai — slide #69

## Veremautomaták



$(q_1, \varepsilon, 0011) \vdash (q_2, \$, 0011) \vdash (q_2, 0$, 011) \vdash (q_2, 00$, 11) \vdash (q_3, 0$, 1) \vdash (q_3, \$, \varepsilon) \vdash (q_4, \varepsilon, \varepsilon)$

## Veremautomaták

### Megjegyzés

1. A veremautomata fogalmát módosíthatjuk úgy, hogy kezdetben a veremben egy rögzített  $\Gamma$ -beli betű legyen, és úgy is, hogy
2. a verembe az egyes átmenetek esetén 1-nél hosszabb szó is kerülhessen. Ekkor

$$\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \rightarrow P(Q \times \Gamma^*)$$

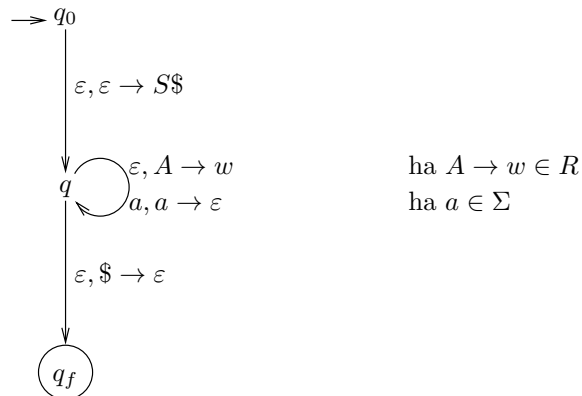
$\delta(q, a, b)$  véges minden  $q \in Q, a \in \Sigma_\varepsilon, b \in \Gamma_\varepsilon$  esetén.

3. A veremautomata **nemdeterminisztikus** modell.

## Veremautomaták

Tétel Minden környezetfüggetlen nyelv felismerhető veremautomatával.

Bizonyítás  $G = (V, \Sigma, R, S)$



## Veremautomaták

Tétel Minden veremautomatával felismerhető nyelv környezetfüggetlen.

Megjegyzés **Determinisztikus** az  $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$  veremautomata, ha

- $|\delta(q, a, b)| \leq 1 \quad q \in Q, a \in \Sigma_\epsilon, b \in \Gamma_\epsilon$
- $\delta(q, \epsilon, b) \neq \emptyset \Rightarrow \delta(q, a, b) = \emptyset \quad q \in Q, a \in \Sigma, b \in \Gamma_\epsilon$
- $\delta(q, a, \epsilon) \neq \emptyset \Rightarrow \delta(q, a, b) = \emptyset \quad q \in Q, a \in \Sigma_\epsilon, b \in \Gamma$

Nem igaz az, hogy minden környezetfüggetlen nyelv felismerhető determinisztikus veremautomatával.

## Környezetfüggetlen nyelvek pumpáló lemmája

**Lemma** Tetszőleges  $L$  környezetfüggetlen nyelvhez van olyan  $p \geq 1$  szám, hogy valahányszor  $w \in L$ ,  $|w| \geq p$ , a  $w$  szó mindig felbontható

$$w = uvxyz$$

alakban úgy, hogy

1.  $\forall i \geq 0 \quad uv^i xy^i z \in L$
2.  $|vy| > 0$
3.  $|vxy| < p$ .

## Környezetfüggetlen nyelvek pumpáló lemmája

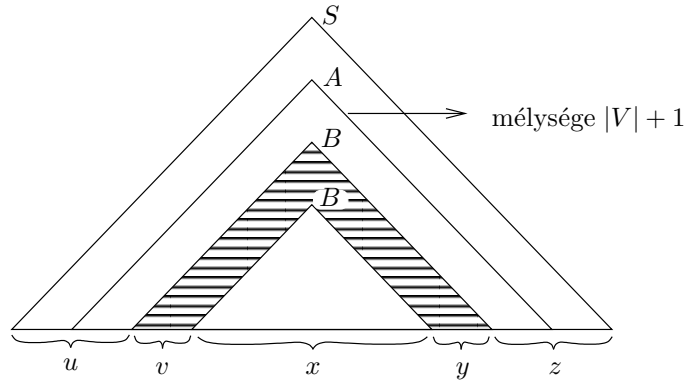
**Bizonyítás** Legyen  $L = L(G)$ , ahol  $G = (V, \Sigma, R, S)$  Chomsky normál formában adott.

- Ha  $t$  az  $X$  nemterminálisból induló derivációs fa mely határa terminális szó és  $t$  mélysége  $\leq n + 1$ , akkor a  $t$  határán lévő szó hossza  $\leq 2^n$ .
- Legyen  $p = 2^{|V|} + 1$ .
- Legyen  $w \in L$ ,  $|w| \geq p$ . Legyen  $t$  a  $w$  szó  $S$ -ből induló derivációs fája. Ekkor  $t$  mélysége  $> |V|$ .

(Fa mélysége: leghosszabb úton lévő élek száma.)



## Környezetfüggetlen nyelvek pumpáló lemmája



Dr. Ésik Zoltán

A számítástudomány alapjai — slide #76

## Környezetfüggetlen nyelvek pumpáló lemmája

Állítás Az  $L = \{a^n b^n c^n : n \geq 0\}$  nyelv nem környezetfüggetlen.

Bizonyítás Belátjuk, hogy  $\forall p > 0 \exists w \in L, |w| \geq p$ , hogy a  $w$  tetszőleges  $w = uvxyz$  felbontására, hogy  $|vy| > 0, |vxy| < p$ , létezik olyan  $i$ , amelyre  $uv^i xy^i z \notin L$ .

Adott  $p$ -hez legyen  $w = a^p b^p c^p$ . Akárhogyan is bontjuk fel  $w$ -ét  $w = uvxyz$  alakban úgy, hogy  $|vy| > 0, |vxy| < p$ , lesz olyan betű, mondjuk  $\xi$ , amely nem fordul elő  $vy$ -ban. Így  $uxz \notin L$ .

Dr. Ésik Zoltán

A számítástudomány alapjai — slide #77

## Környezetfüggetlen nyelvek pumpáló lemmája

Állítás Az  $L = \{w\#w : w \in \{0,1\}^*\}$  nyelv nem környezetfüggetlen.

Bizonyítás Adott  $p$ -hez tekintsük a  $0^p1^p\#0^p1^p$  szót ...

Állítás Az  $L' = \{w\#w' : w, w' \in \{0,1\}^*, w \neq w'\}$  nyelv környezetfüggetlen.

Következmény A környezetfüggetlen nyelvek nem zártak a komplement és metszetképzésre.

Bizonyítás Az  $\{a^n b^n c^m : n, m \geq 0\}$  és  $\{a^n b^m c^m : n, m \geq 0\}$  nyelvek környezetfüggetlenek, de metszetük nem az.

## Nyelvtanok

Definíció (Általános) **nyelvtan** egy  $G = (V, \Sigma, R, S)$  rendszer, ahol  $V, \Sigma, S$  ugyanazok, mint környezetfüggetlen nyelvtan esetén,  $R$  pedig  $u \rightarrow v$  alakú szabályok véges halmaza, ahol  $u, v \in (V \cup \Sigma)^*$  és  $u$  tartalmaz legalább egy nemterminálist.

Definíció Legyen  $G = (V, \Sigma, R, S)$  nyelvtan,  $u, v \in (V \cup \Sigma)^*$ .

- $u \Rightarrow v$  ha létezik olyan  $x, y, y', z \in (V \cup \Sigma)^*$ , amelyre  $u = xyz, v = xy'z, y \rightarrow y' \in R$ .
- $u \Rightarrow^* v$  ha létezik olyan  $n \geq 0$  és  $w_0, w_1, \dots, w_n \in (V \cup \Sigma)^*$ , hogy
$$u = w_0, w_0 \Rightarrow w_1, \dots, w_{n-1} \Rightarrow w_n, w_n = v.$$

A  $G$  által generált nyelv:  $L(G) = \{u \in \Sigma^* : S \Rightarrow^* u\}$ .

## Nyelvtanok

### Példa

$G : S \rightarrow aSBC | \varepsilon \quad L(G) = \{a^n b^n c^n : n \geq 0\}$

$cB \rightarrow Bc$

$aB \rightarrow ab$

$bB \rightarrow bb$

$G' : S_0 \rightarrow S | \varepsilon \quad L(G') = L(G)$

$S \rightarrow aSBC | aBC$

$CB \rightarrow XB$

$XB \rightarrow XC$

$XC \rightarrow BC$

$aB \rightarrow ab$

$bB \rightarrow bb$

$C \rightarrow c$

## Nyelvtanok

**Definíció** **Környezetfüggetlen**nek nevezünk egy  $G = (V, \Sigma, R, S)$  nyelvtant, ha  $R$  minden eleme  $uXv \rightarrow uvw$  alakú, ahol  $u, v, w \in (V \cup \Sigma)^*$ -beli szavak,  $X \in V$ ,  $w \neq \varepsilon$ . Ez alól csak egyetlen kivétel lehet, nevezetesen az  $S \rightarrow \varepsilon$  szabály. Ha ez  $R$ -ben van, akkor kikötjük azt, hogy  $S$  nem fordul elő egyetlen szabály jobboldalán sem.

**Definíció** **Jobblineáris**nek nevezünk egy  $G = (V, \Sigma, R, S)$  nyelvtant, ha  $R$  minden eleme  $A \rightarrow uB$  vagy  $A \rightarrow u$  alakú, ahol  $A, B \in V, u \in \Sigma^*$ .

**Példa**  $G'$  környezetfüggetlen.

## Chomsky-féle hierarchia

Jobblineáris nyelvtan – 3-as típusú nyelvtan

Környezetfüggetlen nyelvtan – 2-es típusú nyelvtan

Környezetfüggő nyelvtan – 1-es típusú nyelvtan

Általános nyelvtan – 0-ás típusú nyelvtan

**Definíció** Legyen  $i \in \{0, 1, 2, 3\}$ . Egy  $L \subseteq \Sigma^*$  nyelvet  $i$  típusúnak nevezünk, ha generálható  $i$  típusú nyelvtannal. **Jelölés:**  $\mathcal{L}_i$

$i = 3$  : jobblineáris nyelvek

$i = 1$  : környezetfüggő nyelvek

**Tétel**  $\mathcal{L}_3 \subseteq \mathcal{L}_2 \subseteq \mathcal{L}_1 \subseteq \mathcal{L}_0$ . Továbbá minden tartalmazás valódi.

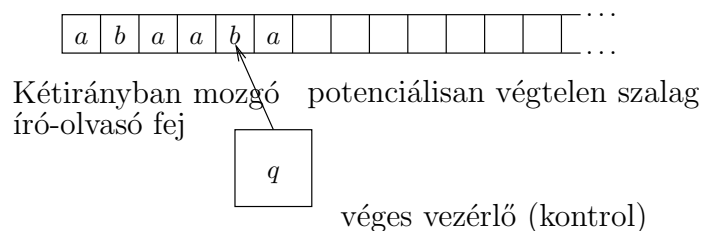
**Tétel**  $\mathcal{L}_3$  a reguláris nyelvek osztálya.

Az  $\mathcal{L}_i, i \in \{0, 1, 2, 3\}$  nyelvek alkotják a Chomsky-féle hierarchiát.

## Turing-gépek

Turing-gép (Alan Turing, 1936)

Véges automata + korlátlan hozzáférésű és nagyságú tár.



## Turing-gépek

**Példa** Az  $L = \{w\#w : w \in \{0,1\}^*\}$  nyelv felismerhető Turing-géppel.

1. A bemenet egyszeri elolvasásával ellenőrizzük, hogy a bemenet egyetlen  $\#$ -et tartalmaz.
2. A szalagot oda-vissza olvasva ellenőrizzük (a vizsgált betűk áthúzásával), hogy a  $\#$  két oldalán ugyanaz a szó helyezkedik-e el.
3. Amennyiben végig egyezést találunk (az összes betű áthúzásra került), akkor a gép elfogadja, különben elutasítja a bementet.

## Turing-gépek

**Definíció** Turing-gép egy  $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$  rendszer, ahol:

1.  $Q$  az **állapotok** véges, nemüres halmaza.
2.  $q_0 \in Q$  a **kezdőállapot**.
3.  $q_{accept} \in Q$  az **elfogadó állapot**.
4.  $q_{reject} \in Q$  az **elutasító állapot**.
5.  $\Sigma$  véges, nemüres halmaz, a **bemenő jelek** (szimbólumok) halmaza.
6.  $\Gamma$  a  $\Sigma$ -t tartalmazó véges halmaz, a **szalag szimbólumainak halmaza**.  $\Gamma - \Sigma$  tartalmaz egy speciális  $\sqcup$  karaktert,  $\Gamma \cap Q = \emptyset$ .
7.  $\delta : (Q - \{q_{accept}, q_{reject}\}) \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$  az **átmenet függvény**.

## Turing-gépek

**Definíció Konfiguráció:**  $uqv$  alakú szó, ahol  $q \in Q$ ,  $u, v \in \Gamma^*$ ,  $v \neq \varepsilon$ . Az  $uqv$  konfiguráció a Turing-gép működésének egy olyan pillanatát reprezentálja, amelyben a szalag tartalma  $uv \sqcup \sqcup \dots$ , a véges kontrol a  $q$  állapotban van, az író-olvasó fej pedig  $v$  első karakterét jelöli ki.

**Definíció Átmenet a konfigurációk között.** Legyen  $uqv$  konfiguráció,  $v$  első betűje  $a$ . Tfh.  $\delta(q, a) = (r, b, R)$ . Ekkor  $uqv \vdash ubrw$ , ahol  $w$  a  $v$ -ből az  $a$  betű elhagyásával kapott szó, ha  $w \neq \varepsilon$ , különben  $w = \sqcup$ . Tfh.  $\delta(q, a) = (r, b, L)$ . Ebben az esetben  $uqv \vdash wrcbv'$ , ahol  $u = wc$ ,  $v = av'$ , illetve  $uqv \vdash urbv'$ , ha  $u = \varepsilon$  és  $v = av'$ . Ha  $q = q_{accept}$  vagy  $q = q_{reject}$ , akkor az  $uqv$  konfiguráció **megállási konfiguráció**.

## Turing-gépek

**Definíció** Legyen  $M$  a fenti Turing-gép. Az  $M$  által felismert nyelv (**Jelölése:**  $L(M)$ ) az összes olyan  $u \in \Sigma^*$  szóból áll, hogy  $q_0 u \sqcup \vdash^* x q_{accept} y$  valamely  $x, y \in \Gamma^*$ ,  $y \neq \varepsilon$  szavakra. (Itt  $\vdash^*$  a  $\vdash$  reláció reflexív-tranzitív lezártja.)

**Definíció** Egy  $L \subseteq \Sigma^*$  nyelv **Turing felismerhető**, vagy **rekurzívan felsorolható**, ha  $L = L(M)$  valamely  $M$  Turing-gépre.

**Megjegyzés** Egy adott Turing-gép nem feltétlenül áll meg egy  $u$  szón, azaz nem biztos, hogy a  $q_0 u \sqcup \vdash^* x q_{accept} y$  vagy  $q_0 u \sqcup \vdash^* x q_{reject} y$  relációk valamelyike teljesül valamely  $x, y$  esetén.

**Definíció** Egy  $L \subseteq \Sigma^*$  nyelv **(Turing-)eldönthető**, vagy **rekurzív**, ha létezik olyan  $M$  Turing-gép, mely minden bemeneten megáll, és amely felismeri az  $L$  nyelvet.

## Turing-gépek

### A Turing-gép néhány variánsa

#### 1. Többszalagos Turing-gép

- $k(> 1)$  szalag, mindegyik külön író-olvasó fejjel
- Kezdetben az első szalag tartalmazza a bemenetet, a többi üres
- $\delta : (Q - \{q_{accept}, q_{reject}\}) \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{L, R\}^k$

Tétel Minden többszalagos Turing-géphez létezik ekvivalens, egyszalagos Turing-gép.

#### 2. Turing-gép két irányban végtelen szalaggal

#### 3. Kétdimenziós Turing-gép

## Nemdeterminisztikus Turing-gépek

#### 4. Nemdeterminisztikus Turing-gép $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$

$$\delta : (Q - \{q_{accept}, q_{reject}\}) \times \Gamma \rightarrow P(Q \times \Gamma \times \{L, R\})$$

$$L(M) = \{u \in \Sigma^* : \exists x, y \quad q_0 u \sqcup \vdash^* x q_{accept} y\}$$

A gép működése az  $u$  szón egy fával reprezentálható. Akkor fogadja el az  $u$  szót, ha a fa valamely levele elfogadási konfiguráció.

Tétel Ha egy  $L$  nyelv felismerhető nemdeterminisztikus Turing-géppel, akkor  $L$  Turing felismerhető.

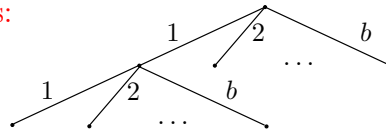
Bizonyítás Legyen adott az  $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$  nemdet. Turing-gép. Elkészítünk egy olyan  $D$  többszalagos (determinisztikus) Turing-gépet, melyre  $L(D) = L(M)$ .  $D$  szélességi kereséssel megvizsgálja adott  $u$  szón az  $M$  kiszámítási fáját. Ha van olyan csúcs, mely elfogadási konfiguráció, akkor elfogadja az  $u$  szót. Különben  $D$  nem áll meg az  $u$  szón.

## Nemdeterminisztikus Turing-gépek

$D$ -nek 3 szalagja van.

- input szalag
- szimulációs szalag
- cím szalag

Címzés:



Egy nemdeterminisztikus Turing-gép **eldönti** az  $L$  nyelvet, ha felismeri, és ha minden kiszámítási út véges és elfogadási vagy elutasítási konfigurációhoz vezet.

Tétel Ha egy nyelv eldönthető nemdeterminisztikus Turing-géppel, akkor eldönthető.

## Turing-géppel felsorolható nyelvek

Definíció Azt mondjuk, hogy az  $M$  2-szalagos Turing-gép **felsorolja az  $L$  nyelvet**, ha üres bemeneti szalaggal indítva másik szalagján, esetleg ismétlésekkel, felsorolja az  $L$  elemeit.

Tétel Egy  $L$  nyelv akkor és csakis akkor sorolható fel Turing-géppel, ha Turing felismerhető.

**$E$  felsoroló Turing-gép  $\Rightarrow$   $M$  felismerő Turing-gép**

$M$  működése: Adott  $w$  szón,

- futtatja  $E$ -t,
- valahányszor  $E$  kiír a kimenő szalagjára egy szót, azt összehasonlítja  $w$ -vel,
- ha egyszer egyezést talál, elfogadja  $w$ -ét.



## Turing-géppel felsorolható nyelvek

$M$  felismerő Turing-gép  $\Rightarrow$   $E$  felsoroló Turing-gép

$E$  működése:

- $i = 1, 2, 3, \dots$  esetén végtelen ciklusban futtatja  $M$ -et legfeljebb  $i$  lépésig az első  $i$  szó mindegyikén ( $\Sigma^*$  elemeit  $w_1, w_2, \dots$  lexikografikusan felsoroljuk).
- Ha valamelyik  $w_j$  szót legfeljebb  $i$  lépésben  $M$  elfogadja, akkor  $E$  kiírja  $w_j$ -ét a kimenő szalagra.

## A kiszámítás egyéb modelljei

A kiszámítás egyéb modelljei:

- $RAM$  vagy regiszter-gép
- minden univerzális programozási nyelv ( $LISP, PASCAL, \dots$ )
- nyelvtanok
- és még számos más modell ( $\lambda$ -kiszámíthatóság, Markov-algoritmus,  $\dots$ )

Ezek mindegyikéről kiderült, hogy **ekvivalens** a Turing-géppel.

Tétel Egy  $L$  nyelv akkor és csakis akkor Turing-felismerhető, ha az  $\mathcal{L}_0$  nyelvosztályba esik.

## Algoritmusok

**Algoritmusok a matematikában:** az ókortól (prímtesztelés, legnagyobb közös osztó, ...)

**Az algoritmus fogalmának formalizálása:** 1930-as évek

**Hilbert 1900:** 10. probléma

Adjunk meg olyan algoritmust, mely tetszőleges egész együtthatós többváltozós polinomról eldönti, van-e olyan zérushelye, mely komponensei egészek.

**Matijasevič 1971:**

Ilyen algoritmus nem létezik.

**Church-Turing tézis 1936:**

Az algoritmus intuitív fogalma = Turing-géppel megoldható algoritmus.

## Algoritmusok

**A tézis bizonyítékai:**

A matematikai formalizmusok ekvivalenciái, a konkrét algoritmusok formalizálásai.

**Matijasevič tétele formálisan:** A

$$\{ p : p \text{ olyan egész együtthatós polinom, melynek létezik egész értékű gyöke } \}$$

nyelv nem dönthető el (nem rekurzív).

A Church-Turing tézis értelmében elegendő algoritmusainkat magas szinten megfogalmazni.

## Algoritmusok

Példa Gráfok összefüggősége

$$L = \{ \langle G \rangle : G \text{ összefüggő gráf} \}$$

*L*-et eldöntő Turing-gép:

Adott  $\langle G \rangle$  bemenet esetén:

1. Válasszuk ki  $G$  első csúcsát és jelöljük meg.
2. Mindaddig, amíg új csúcsok már nem jelölhetők meg, a  $G$  minden egyes még meg nem jelölt csúcsára, ha a csúcs csatlakozik egy már megjelölt csúcshoz, akkor jelöljük meg.
3. Ellenőrizzük, hogy az összes csúcsot megjelöltük-e. Ha igen, akkor elfogadjuk a bemenetet, különben elutasítjuk.

## Néhány eldönthető nyelv (probléma)

$$L_{DFA} = \{ \langle M, w \rangle : M \text{ véges automata, } w \in L(M) \}$$

Állítás  $L_{DFA}$  eldönthető nyelv.

Bizonyítás Egy  $T$  Turing-gép először ellenőrzi, hogy a bemenő szó egy  $\langle M, w \rangle$  alakú szó-e, majd:

1. Átmásolja  $M$  kódját egy munkaszalagjára.
2. Egy másik munkaszalagján szimulálja  $M$  működését a  $w$  szón.

## Néhány eldönthető nyelv (probléma)

$L_{NFA} = \{\langle M, w \rangle : M \text{ nemdet. véges automata, } w \in L(M)\}$

Állítás  $L_{NFA}$  eldönthető nyelv.

Bizonyítás Egy  $T$  Turing-gép először ellenőrzi, hogy a bemenő szó  $\langle M, w \rangle$  alakú-e, ahol  $M$  nemdet. véges automata,  $w$  az  $M$  bemenő szava, majd

- a hatványhalmaz konstrukcióval elkészít egy olyan  $M'$  determinisztikus véges automatát, mely  $M$ -el ekvivalens,
- szimulálja az  $L_{DFA}$  nyelvet eldöntő Turing-gépet az  $\langle M', w \rangle$  bemeneten.

## Néhány eldönthető nyelv (probléma)

$L_{REX} = \{\langle R, w \rangle : R \text{ reguláris kifejezés, } w \in |R|\}$

Állítás Az  $L_{REX}$  nyelv eldönthető.

Bizonyítás A  $T$  Turing-gép először ellenőrzi, hogy a bemenet  $\langle R, w \rangle$  alakú-e, ahol  $R$  reguláris kifejezés és  $w$  szó, majd

- elkészít egy olyan  $M$  véges, nemdeterminisztikus automatát, melyre  $|R| = L(M)$ , és
- az  $L_{NFA}$  nyelvet eldöntő Turing-gépet szimulálva eldönti, hogy  $w \in L(M)$  (azaz  $\langle M, w \rangle \in L_{NFA}$ ) teljesül-e.

## Néhány eldönthető nyelv (probléma)

$L_{CFG} = \{\langle G, w \rangle : G \text{ környezetfüggetlen nyelvtan, } w \in L(G)\}$

$L_{CSG} = \{\langle G, w \rangle : G \text{ környezetfüggő nyelvtan, } w \in L(G)\}$

Állítás  $L_{CSG}$  eldönthető.

Bizonyítás Egy  $T$  Turing-gép először eldönti, hogy bemenete  $\langle G, w \rangle$  alakú-e, ahol  $G$  környezetfüggetlen nyelvtan,  $G = (V, \Sigma, R, S)$ ,  $w \in \Sigma^*$  szó, majd ha ez teljesül:

- Ha  $w = \varepsilon$ , ellenőrzi, hogy  $S \rightarrow \varepsilon$  szabály-e.
- Ha  $w \neq \varepsilon$ , előállítja az összes olyan  $u_0, u_1, \dots, u_k \in (V \cup \Sigma)^*$  ismétlődés nélküli sorozatot, melyre  $|u_i| \leq |w|$ ,  $i = 0, \dots, k$ , majd ellenőrzi, ezek közt van-e olyan, mely a  $w$  szó  $S$ -ből induló derivációja. (A sorozatokat lexikografikus sorrendben állítja elő.)

## Néhány eldönthető nyelv (probléma)

Következmény  $L_{CFG}$  eldönthető.

Bizonyítás Egy Turing-gép először eldönti, hogy a bemenő szó  $\langle G, w \rangle$  alakú-e, ahol  $G$  környezetfüggetlen nyelvtan és  $w$  szó, majd előállít egy olyan  $G'$  Chomsky normál formájú nyelvtant, melyre  $L(G) = L(G')$ . Majd az  $L_{CSG}$  nyelvet eldöntő Turing gépet szimulálva eldönti, hogy  $\langle G', w \rangle \in L_{CSG}$  teljesül-e.

Következmény Minden környezetfüggő nyelv rekurzív.

Bizonyítás Legyen  $L$  adott környezetfüggő nyelv,  $L \subseteq \Sigma^*$ . Létezik olyan  $G_0$  környezetfüggő nyelvtan, melyre  $L = L(G_0)$ . Ha adott  $w \in \Sigma^*$  szó, ahhoz, hogy eldöntsük  $w \in L$  teljesül-e, szimuláljuk az  $L_{CSG}$  nyelvet eldöntő Turing-gépet a  $\langle G_0, w \rangle$  szón.

## Néhány eldönthető nyelv (probléma)

Következmény Minden reguláris és minden környezetfüggetlen nyelv rekurzív.

$$E_{DFA} = \{\langle M \rangle : M \text{ véges, det. automata, melyre } L(M) = \emptyset\}$$

$$E_{NFA} = \{\langle M \rangle : M \text{ véges, nemdet. automata, melyre } L(M) = \emptyset\}$$

Állítás Az  $E_{DFA}$  és  $E_{NFA}$  nyelvek rekurzívak.

Bizonyítás Csak  $E_{NFA}$ -ra. Egy Turing-gép először eldönti, hogy a bemenet  $\langle M \rangle$  alakú-e, ahol  $M$  egy  $(Q, \Sigma, \delta, q_0, F)$  véges nemdeterminisztikus automata, majd eldönti, hogy  $M$  átmeneti grájfjában van-e olyan út, mely a  $q_0$  kezdőállapotból végállapotba vezet.

## Néhány eldönthető nyelv (probléma)

$$EQ_{DFA} = \{\langle M_1, M_2 \rangle : M_1 \text{ és } M_2 \text{ ekvivalens véges det. automaták}\}$$

$$EQ_{NFA} = \{\langle M_1, M_2 \rangle : M_1 \text{ és } M_2 \text{ ekvivalens véges nemdet. automaták}\}$$

Állítás  $EQ_{DFA}$  és  $EQ_{NFA}$  rekurzív nyelvek.

Bizonyítás

$$L(M_1) = L(M_2) \Leftrightarrow (L(M_1) - L(M_2)) \cup (L(M_2) - L(M_1)) = \emptyset$$

$$E_{CFG} = \{\langle G \rangle : G \text{ környezetfüggetlen, } L(G) = \emptyset\}$$

Állítás  $E_{CFG}$  rekurzív.

Bizonyítás

$$G = (V, \Sigma, R, S)$$

$$V_0 = \{A \in V : \exists u \in \Sigma^* \quad A \rightarrow u \in R\}$$

$\vdots$

$$V_{n+1} = V_n \cup \{A : \exists u \in (\Sigma \cup V_n)^* \quad A \rightarrow u \in R\}$$

$$L(G) \neq \emptyset \Leftrightarrow S \in V_k, \text{ ahol } |V| = k.$$

## Néhány eldönthetetlen nyelv (probléma)

$$E_{CSG} = \{\langle G \rangle : G \text{ környezetfüggő, } L(G) = \emptyset\}$$

$$EQ_{CFG} = \{\langle G_1, G_2 \rangle : G_1, G_2 \text{ környezetfüggetlen nyelvtanok,} \\ L(G_1) = L(G_2)\}$$

Állítás A fenti nyelvek nem rekurzívak.

## Néhány eldönthetetlen nyelv (probléma)

$$L_{TM} = \{\langle M, w \rangle : M \text{ Turing-gép, } w \in L(M)\}$$

Tétel Az  $L_{TM}$  nyelv nem rekurzív.

Bizonyítás Indirekt bizonyítás. Tfh.  $L_{TM}$  rekurzív. Ekkor létezik olyan  $H$  Turing-gép, mely eldönti az  $L_{TM}$  nyelvet:

$$H(\langle M, w \rangle) = \begin{cases} \text{accept} & \text{ha } w \in L(M) \\ \text{reject} & \text{ha } w \notin L(M) \end{cases}$$

Ezek után működjön a  $D$  Turing-gép az alábbi módon. Ha adott egy  $M$  Turing-gép  $\langle M \rangle$  kódja, akkor

$$D(\langle M \rangle) = \begin{cases} \text{accept} & \text{ha } \langle M \rangle \notin L(M) \\ \text{reject} & \text{ha } \langle M \rangle \in L(M) \end{cases}$$

$D$  úgy működik, hogy adott  $\langle M \rangle$  bemenetre szimulálja  $H$ -t az  $\langle M, \langle M \rangle \rangle$  szón.

## Néhány eldönthetetlen nyelv (probléma)

Így:

$$D(\langle D \rangle) = \begin{cases} \text{accept} & \text{ha } \langle D \rangle \notin L(D) \\ \text{reject} & \text{ha } \langle D \rangle \in L(D) \end{cases} \quad \text{Ellentmondás}$$

Állítás Az  $L_{TM}$  nyelv rekurzívan felsorolható.

Bizonyítás Az  $U$  univerzális Turing-gép működjön az  $\langle M, w \rangle$  bemenő szón úgy, hogy szimulálja  $M$  működését a  $w$  szón.

Következmény Létezik olyan rekurzívan felsorolható nyelv, mely nem rekurzív.

## Néhány eldönthetetlen nyelv (probléma)

Állítás Ha  $L \subseteq \Sigma^*$  rekurzív, akkor  $\bar{L} = \Sigma^* - L$  is rekurzív.

Bizonyítás Egy  $L$ -et eldöntő Turing-gépben cseréljük fel a  $q_{\text{accept}}$  és  $q_{\text{reject}}$  állapotokat.

Tétel Egy  $L \subseteq \Sigma^*$  nyelv akkor és csak akkor rekurzív, ha  $L$  és  $\bar{L}$  is rekurzívan felsorolható.

Bizonyítás  $\Rightarrow$ : Az előző állítás felhasználásával.

$\Leftarrow$ : Tegyük fel, hogy léteznek az  $L$ -et és  $\bar{L}$ -et felismerő  $M$  és  $\bar{M}$  Turing-gépek. Adott  $w$  szón az  $N$  Turing-gép lépésenként felváltva szimulálja  $M$  és  $\bar{M}$  működését.

Következmény  $\bar{L}_{TM}$  nem rekurzívan felsorolható.



## Néhány eldönthetetlen nyelv (probléma)

$$H_{TM} = \{\langle M, w \rangle : M \text{ Turing-gép, } M \text{ megáll } w\text{-én}\}$$

Állítás A  $H_{TM}$  nyelv nem rekurzív.

Bizonyítás (Visszavezetéssel az  $L_{TM}$  problémából.)

Tegyük fel, hogy  $H_{TM}$  rekurzív. Ekkor működjön a  $T$  Turing-gép az  $\langle M, w \rangle$  bemeneten a következő módon:

1. Szimulálja a  $H_{TM}$ -et eldöntő Turing-gépet az  $\langle M, w \rangle$  szón. Ha  $H_{TM}$  nem fogadja el az  $\langle M, w \rangle$  szót, akkor  $T$  sem fogadja el.
2. Ha  $H_{TM}$  elfogadja az  $\langle M, w \rangle$  szót, akkor  $T$  szimulálja az  $M$  gépet a  $w$  szón.

## Néhány eldönthetetlen nyelv (probléma)

$$E_{TM} = \{\langle M \rangle : L(M) = \emptyset\}$$

Állítás  $E_{TM}$  nem rekurzív.

Bizonyítás (Visszavezetéssel az  $L_{TM}$  problémából.)

Tegyük fel, hogy  $E_{TM}$  rekurzív. Adott  $\langle M, w \rangle$  bemeneten a  $T$  Turing-gép:

1. Elkészít egy olyan  $M_w$  Turing-gépet, mely a  $w$ -én kívül elutasít minden bemenő szót,  $w$ -én pedig úgy működik, mint  $M$ ,
2. Az  $E_{TM}$ -et eldöntő Turing-gépet felhasználva eldönti, hogy  $\langle M_w \rangle \in E_{TM}$  teljesül-e.

## Rice tétele

Tétel (Rice) A rekurzívan felsorolható nyelvek egyetlen nemtriviális tulajdonsága sem dönthető el.

Így nem dönthető el, hogy adott  $M$  Turing-gép:

- üres nyelvet ismer-e fel,
- véges nyelvet ismer-e fel,
- reguláris nyelvet ismer-e fel,
- ...

Következmény Nem dönthető el adott  $M_1, M_2$  Turing-gépekre, hogy  $L(M_1) = L(M_2)$  teljesül-e.

## A Post megfelelezési probléma

A Post megfelelezési probléma:

Dominó készlet:  $\left\{ \begin{bmatrix} b \\ ca \end{bmatrix}, \begin{bmatrix} a \\ ab \end{bmatrix}, \begin{bmatrix} ca \\ a \end{bmatrix}, \begin{bmatrix} abc \\ c \end{bmatrix} \right\}$

Ekkor az alábbi dominósorozat tetején és alján lévő szó megegyezik.

$$\left\{ \begin{bmatrix} a \\ ab \end{bmatrix} \begin{bmatrix} b \\ ca \end{bmatrix} \begin{bmatrix} ca \\ a \end{bmatrix} \begin{bmatrix} a \\ ab \end{bmatrix} \begin{bmatrix} abc \\ c \end{bmatrix} \right\}$$

Az ilyen dominósorozatot a **Dominó készlet megoldásának** nevezzük.

## A Post megfelelezési probléma

Tétel A  $\{\langle D \rangle : D \text{ dominókészlet, melynek van megoldása}\}$  nyelv nem rekurzív, azaz nem létezik olyan algoritmus, mely eldöntené adott dominókészletről, hogy van-e megoldása.

## A Post megfelelezési probléma

Tétel Nem létezik olyan algoritmus, mely tetszőleges környezetfüggetlen nyelvtanról eldöntené, hogy egyértelmű-e.

### Bizonyítás

$P = \{(u_1, v_1), \dots, (u_k, v_k) : u_i, v_i \in \Sigma^*\}$  a *PMP* egy példánya.

$$\begin{array}{l} G_p : S \rightarrow A|B \\ A \rightarrow iAu_i^{-1} \mid iu_i^{-1} \\ B \rightarrow iBv_i^{-1} \mid iv_i^{-1} \end{array} \quad i = 1, \dots, k$$

## A Post megfelekezési probléma

**Tétel** Nem létezik olyan algoritmus, mely adott  $G_1, G_2$  környezetfüggetlen nyelvtanokról eldönti, hogy  $L(G_1) = L(G_2)$  teljesül-e.

### Bizonyítás

$$P = \{(u_1, v_1), \dots, (u_k, v_k) : u_i, v_i \in \Sigma^*\}$$

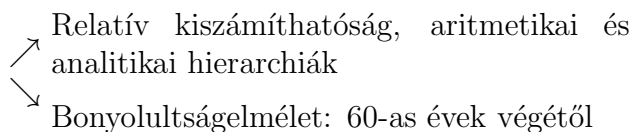
$G_1$  olyan nyelvtan, mely az

$$\{i_1 \dots i_n \# w : w \in \Sigma^*, w \neq (u_{i_1} \dots u_{i_n})^{-1} \\ \text{vagy } w \neq (v_{i_1} \dots v_{i_n})^{-1}, n > 0\} \text{ nyelvet generálja}$$

$G_2$  olyan jobblinéaris nyelvtan, mely az  $\{1, \dots, k\}^+ \# \Sigma^*$  reguláris nyelvet generálja.

## Bonyolultságelmélet

A kiszámíthatóságelmélet kiterjesztései:



**Cél:** A megoldható (eldönthető) problémák osztályozása a megoldáshoz szükséges erőforrások (idő, tár) mennyisége szerint.

legrosszabb eset

átlagosan

## Bonyolultságelmélet

**Definíció** Legyenek  $f, g : N \rightarrow R_+$  függvények, ahol  $N = \{0, 1, \dots\}$ ,  $R_+$  pedig a nemnegatív valós számok halmaza.

$f(n) = \mathcal{O}(g(n))$  ha létezik olyan  $c > 0$  és  $n_0 \in N$ , hogy

$$f(n) \leq c \cdot g(n), \quad n \geq n_0$$

### Példa

- $5n^3 + 6n^2 + 1 = \mathcal{O}(n^3)$
- $n^k = \mathcal{O}(2^n)$
- $\log_2 n = \mathcal{O}(\log_3 n)$
- $\log \log n = \mathcal{O}(\log n)$

## Bonyolultságelmélet

**Példa**  $L = \{0^k 1^k : k \geq 0\}$

$M_1$  : Adott  $w$  szón:

1. Olvassuk végig a szalagot, hogy ellenőrizzük 1-es után nem következik-e 0.
2. Mindaddig, amíg 0 és 1-es is van a szalagon ismételjük az alábbiakat:
  3. Olvassuk végig a szalagot és húzzunk át egy 0-át és egy 1-et.
4. Ha végül nem maradt a szalagon karakter, akkor fogadjuk el a bemenetet. Különben utasítsuk el.

**Időigény:**  $\mathcal{O}(n) + \mathcal{O}(n^2) + \mathcal{O}(n) = \mathcal{O}(n^2)$

## Bonyolultságelmélet

Példa  $L = \{0^k 1^k : k \geq 0\}$

$M_2$  : Adott  $w$  szón:

1. Olvassuk végig a szalagot, és ellenőrizzük, hogy egyetlen 1-est sem követ 0.
2. Mindaddig, amíg legalább egy 0 és egy 1 marad a szalagon, ismételjük az alábbiakat:
  3. Ellenőrizzük, hogy a szalagon maradt karakterek száma páros-e. Ha páratlan elutasítjuk a bemenetet.
  4. Olvassuk végig a szalagot, és húzzuk át minden második 0-át és 1-et (az első 0-át és 1-et áthúzva).
5. Ha végül nem marad áthúzatlan karakter, akkor elfogadjuk a bemenetet. Különben elutasítjuk.

**Időigény:**  $\mathcal{O}(n) + \mathcal{O}(n \log n) + \mathcal{O}(n) = \mathcal{O}(n \log n)$

## Bonyolultságelmélet

Példa  $L = \{0^k 1^k : k \geq 0\}$

$M_3$  : Adott  $w$  szón:

1. Ellenőrizzük, hogy a szalagon nem követ egyetlen 1-et sem 0.
2. Másoljuk a 0-akat egy munkaszalagra.
3. Olvassuk végig az 1-eseket a bemeneten, és minden egyes 1-esre húzzunk át a munkaszalagon egy 0-át.
4. Ha mindegyik 0-át áthúztuk és az 1-eseket is elolvastuk, akkor fogadjuk el a bemenetet. Különben utasítsuk el.

**Időigény:**  $\mathcal{O}(n)$

## Turing gépek időigénye

**Definíció** Legyen  $M$  olyan determinisztikus Turing-gép, amely minden bemeneten megáll. Az  $M$  időigénye az az  $f : N \rightarrow N$  függvény, amelyre  $f(n)$  a legfeljebb  $n$ -hosszú bemeneten az  $M$  által megtett lépések számának maximuma.

**Tétel** Legyen  $t(n) \geq n$ . Minden  $t(n)$  időigényű többszalagos Turing-géphez létezik ekvivalens,  $\mathcal{O}(t^2(n))$  időigényű egyszalagos Turing-gép.

**Bizonyítás** Mint korábban.

## Turing gépek időigénye

**Definíció** Legyen  $T$  olyan nemdeterminisztikus Turing-gép, melynek minden számítási sorozata véges és elfogadáshoz vagy elutasításhoz vezet. A  $T$  időigényét az az  $f : N \rightarrow N$  függvény adja, amelyre tetszőleges  $n$  esetén  $f(n)$  a  $T$  legfeljebb  $n$ -hosszú bemeneten való számítási sorozatai hosszának maximuma (= leghosszabb út hossza a  $T$  legfeljebb  $n$ -hosszú bemeneten való számítási fáiban).

**Tétel** Legyen  $T$  olyan Turing-gép, mint az előző definícióban. Tegyük fel, hogy  $T$  időigénye  $\leq t(n)$ , ahol  $t(n) \geq n$ . Ekkor létezik  $T$ -vel ekvivalens,  $2^{\mathcal{O}(t(n))}$  időigényű determinisztikus Turing-gép.

## A P nyelvosztály

### Definíció

$TIME(t(n)) = \{L : L \text{ eldönthető } \mathcal{O}(t(n)) \text{ időigényű det. Turing-géppel}\}$

Definíció  $\mathbf{P} = \bigcup_{k \geq 1} TIME(n^k)$  Polinomidőben eldönthető nyelvek (problémák)

### Példa ELÉRHETŐSÉG

$\{\langle G, s, t \rangle : G \text{ véges, irányított gráf, } s, t \in G, \text{ létezik } s\text{-ből } t\text{-be vezető út}\}$

## A P nyelvosztály

Adott  $\langle G, s, t \rangle$  bemenet:

1. Jelöljük meg az  $s$  csúcsot.
2. Mindaddig, amíg már nem jelölhető meg új csúcs, ismételjük meg az alábbiakat:
  3. Olvassuk végig az éleket. Ha olyan  $(a, b)$  élet találunk, hogy  $a$  meg van jelölve, de  $b$  nincs, jelöljük meg a  $b$  csúcsot.
4. Ha végül  $t$  is meg van jelölve, akkor elfogadjuk, különben elutasítjuk a bemenetet.

Időigény:  $\mathcal{O}(n^2)$



## A P nyelv osztály

Tétel Minden környezetfüggetlen nyelv a  $P$  osztályba esik.

Bizonyítás Legyen  $L$  környezetfüggetlen nyelv. Ekkor létezik  $L$ -et generáló  $G = (V, \Sigma, R, S)$  Chomsky normál formában adott nyelvtan.

Legyen  $w \in \Sigma^*$ ,  $w = w_1, \dots, w_n$ ,  $n > 0$ .

$$T_{ij} = \{X \in V : X \Rightarrow^* w_i \dots w_j\} \quad 1 \leq i \leq j \leq n$$

A  $T_{ij}$  halmazok mindegyike meghatározható  $\mathcal{O}(n)$  időben.

$$T_{ii} = \{X \in V : X \rightarrow w_i \in R\} \quad i = 1, \dots, n$$

$$T_{ij} = \{X \in V : \exists A, B \quad X \rightarrow AB \in R \\ A \in T_{ik}, B \in T_{k+1j} \text{ valamely } i \leq k < j \text{ esetén } \} \\ \text{ha } i < j$$

**Időigény:  $\mathcal{O}(n^3)$**

## Az NP osztály

**Mai ismereteinkkel csak kimerítő kereséssel megoldható problémák:**

HAMILTON-ÚT

TELJES RÉSZGRÁF

EGÉSZ ÉRTÉKŰ PROGRAMOZÁS

HÁTIZSÁK

KIELÉGÍTHETŐSÉG (SAT)

A fenti problémák mindegyike megoldható polinomidőben nondeterminisztikus Turing-géppel – nem realiztikus számítási modell. Egyikről sem ismert, hogy megoldható-e polinomidőben determinisztikus Turing-géppel, azaz, hogy a  $P$  osztályba esik-e.

## Az NP osztály

### Definíció

$NTIME(t(n)) = \{L : L \text{ eldönthető } \mathcal{O}(t(n)) \text{ időigényű nemdet. Turing-géppel}\}$

**Definíció**  $\mathbf{NP} = \bigcup_{k \geq 1} NTIME(n^k)$

**Állítás** A HAMILTON-ÚT, TELJES RÉSZGRÁF, EGÉSZ ÉRTÉKŰ PROGRAMOZÁS, HÁTIZSÁK, SAT problémák mindegyike az **NP** osztályba esik.

Világos, hogy  $\mathbf{P} \subseteq \mathbf{NP}$ .

**Nyitott kérdés:**  $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ .

$\mathbf{P} = \mathbf{NP}$  akkor és csakis akkor, ha az előző állításban szereplő valamelyik probléma (és akkor mindegyikük) a **P** osztályba esik: **NP** teljes problémák.

## Polinom időben verifikálható nyelvek

**Definíció** Azt mondjuk, hogy egy  $L$  nyelv **polinom időben verifikálható**, ha létezik olyan  $K$  nyelv és  $k$  szám, hogy  $K \in \mathbf{P}$  és

$$L = \{x : \exists y \langle x, y \rangle \in K \wedge |y| = \mathcal{O}(|x|^k)\}$$

**Példa** HAMILTON-ÚT és a többi fent említett probléma (nyelv) mindegyike polinom időben verifikálható.

**Tétel**  $L \in \mathbf{NP} \Leftrightarrow L$  polinom időben verifikálható.

## Polinom idejű visszavezetés

**Definíció** Egy  $f : \Sigma^* \rightarrow \Delta^*$  függvény **polinomidőben kiszámítható**, ha létezik olyan  $M$  polinom időigényű Turing-gép, hogy tetszőleges  $w \in \Sigma^*$  szóra megálláskor a szalag tartalma  $f(w)$ .

**Definíció** Legyenek  $L_1 \subseteq \Sigma_1^*$  és  $L_2 \subseteq \Sigma_2^*$  nyelvek. Az  $L_1$ -nek  $L_2$ -re való **polinomidejű visszavezetése** egy olyan polinomidőben kiszámítható

$$f : \Sigma_1^* \rightarrow \Sigma_2^*$$

függvény, hogy tetszőleges  $w \in \Sigma_1^*$  szóra:

$$w \in L_1 \Leftrightarrow f(w) \in L_2.$$

Azt mondjuk, hogy  $L_1$  **polinomidőben visszavezethető**  $L_2$ -re,  $L_1 \leq_p L_2$ , ha létezik az  $L_1$ -nek polinomidejű visszavezetése az  $L_2$  nyelvre.

## NP-teljes nyelvek

**Állítás** A  $\leq_p$  reláció előrendezés (reflexív és tranzitív).

**Állítás** Az **NP** osztály zárt a polinomidejű visszavezetésre:

$$L_1 \leq_p L_2, L_2 \in \mathbf{NP} \Rightarrow L_1 \in \mathbf{NP}.$$

**Megjegyzés** A **P** is nyilvánvalóan zárt. Továbbá ha  $L_1, L_2 \in \mathbf{P}$ , és  $L_1, L_2$  és komplementeik sem üresek, akkor  $L_1 \leq_p L_2$ .

**Definíció** Egy  $L$  nyelv (probléma) **NP-nehéz**, ha  $L' \leq_p L$  teljesül minden  $L' \in \mathbf{NP}$  nyelvre. Egy  $L$  nyelv (probléma) **NP-teljes**, ha  $L \in \mathbf{NP}$  és  $L$  **NP-nehéz**.

## NP-teljes nyelvek

Állítás Tegyük fel, hogy  $L$  NP-teljes. Ekkor  $\mathbf{P} = \mathbf{NP} \Leftrightarrow L \in \mathbf{P}$ .

Bizonyítás  $\Rightarrow$ : triviális.  
 $\Leftarrow$ : Tegyük fel, hogy  $L \in \mathbf{P}$ . Ekkor tetszőleges  $L' \in \mathbf{NP}$  esetén  $L' \leq_p L$  miatt  $L' \in \mathbf{P}$ . Tehát  $\mathbf{NP} \subseteq \mathbf{P}$ , így  $\mathbf{NP} = \mathbf{P}$ .

Állítás Tegyük fel, hogy  $L \leq_p L'$ . Ha  $L$  NP-nehéz, akkor  $L'$  is az. Továbbá, ha  $L$  NP-nehéz, akkor  $L'$  akkor és csak akkor NP-teljes, ha  $L' \in \mathbf{NP}$ .

Bizonyítás Az első állítás adódik a  $\leq_p$  tranzitivitásából. A második az elsőből következik.

## NP-teljes nyelvek

### Definíció

$\mathbf{SAT} = \{ \langle \varphi \rangle : \varphi \text{ kielégíthető konjunktív normálformában (knf) adott logikai kifejezés (formula)} \}$

**literál:** változó illetve annak negáltja

**tag:** literálok diszjunkciója

**knf:** tagok konjunkciója

Példa  $(x_1 \vee \bar{x}_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3) \in \mathbf{SAT}$ .

Tétel (Cook) SAT NP-teljes.

Bizonyítás Később.

## NP-teljes nyelvek

**Definíció** Legyen  $k \geq 1$ .  $k\text{SAT} = \{ \langle \varphi \rangle : \langle \varphi \rangle \in \text{SAT}, \varphi \text{ minden tagjában } k \text{ literál van} \}$ .

**Kövekezmény**  $3\text{SAT}$  NP-teljes.

**Bizonyítás** Nyilvánvalóan  $3\text{SAT} \in \text{NP}$ . Még belátjuk, hogy  $\text{SAT} \leq_p 3\text{SAT}$ .

**$c$  tag**

$l$

$l_1 \vee l_2$

$l_1 \vee l_2 \vee l_3$

$l_1 \vee l_2 \vee l_3 \vee l_4$

$l_1 \vee l_2 \vee l_3 \vee l_4 \vee l_5$

**$f(c)$  knf**

$l \vee l \vee l$

$l_1 \vee l_2 \vee l_2$

$l_1 \vee l_2 \vee l_3$

$(l_1 \vee l_2 \vee x) \wedge (\bar{x} \vee l_3 \vee l_4)$

$x$  új változó

$(l_1 \vee l_2 \vee x) \wedge (\bar{x} \vee l_3 \vee y) \wedge (\bar{y} \vee l_4 \vee l_5)$

$x$  és  $y$  új változók

## NP-teljes nyelvek

$l_1 \vee \dots \vee l_n, \quad n > 5$

$\varphi$  knf,  $\varphi = c_1 \wedge \dots \wedge c_n$

$(l_1 \vee l_2 \vee x) \wedge (\bar{x} \vee l_3 \vee y) \wedge \dots$

$(\bar{z} \vee l_{n-2} \vee u) \wedge (\bar{u} \vee l_{n-1} \vee l_n)$

$x, y, \dots, z, u$  új változók

$f(\varphi) = f(c_1) \wedge \dots \wedge f(c_n)$  ahol

minden  $f(c_i)$ -ben új változókat

vezetünk be.

**Ekkor  $f$  polinomidőben való visszavezetés:**

$$\langle \varphi \rangle \in \text{SAT} \Leftrightarrow \langle f(\varphi) \rangle \in 3\text{SAT}.$$

## NP-teljes nyelvek

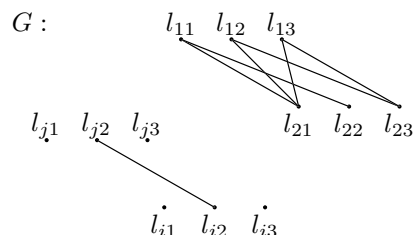
**Def. TELJES RÉSZGRÁF** =  $\{\langle G, k \rangle : G \text{ véges gráf, } k \geq 1, G\text{-nek létezik } k \text{ csúcsú teljes részgráfja}\}$

**Tétel** TELJES RÉSZGRÁF **NP**-teljes.

**Bizonyítás** 3SAT-ból való visszavezetéssel.

$$\varphi = c_1 \wedge \dots \wedge c_k$$

$$c_i = l_{i1} \vee l_{i2} \vee l_{i3}$$



Az összes élet behúzzuk kivéve

- az egy csoporton belül lévő csúcsok köztiket
- az ellentétes literálokkal címkézett csúcsok közt.

## NP-teljes nyelvek

**Állítás**  $\varphi \in 3\text{SAT} \Leftrightarrow \langle G, k \rangle \in \text{TELJES RÉSZGRÁF}$ .

Tehát  $3\text{SAT} \leq_p \text{TELJES RÉSZGRÁF}$ . Mivel 3SAT **NP**-teljes, és TELJES RÉSZGRÁF  $\in$  **NP**, ezért TELJES RÉSZGRÁF is **NP**-teljes.

**Következmény** FÜGGETLEN CSÚCSHALMAZ **NP**-teljes, ahol

**FÜGGETLEN CSÚCSHALMAZ** =  $\{\langle G, k \rangle : \text{a } G \text{ véges gráfnak van } k \text{ elemű független csúcshalmaza}\}$ .

## HAMILTON-ÚT NP-teljes

**HAMILTON-ÚT** =  $\{ \langle G, s, t \rangle : G \text{ irányított gráf, } s, t \text{ csúcsok,} \\ \exists s\text{-ből } t\text{-be Hamilton-út} \}$ .

Tétel HAMILTON-ÚT NP teljes.

Bizonyítás A 3SAT NP-teljeségének felhasználásával.

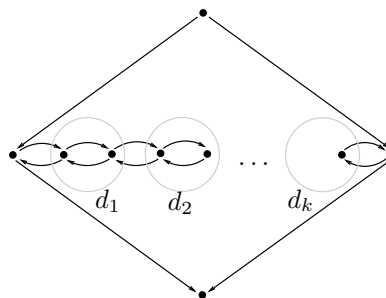
Legyen  $\varphi$  3SAT alakú formula:

$$\varphi = d_1 \wedge \dots \wedge d_k, \quad d_j = a_j \vee b_j \vee c_j$$

Legyenek  $x_1, x_2, \dots, x_l$  a felhasznált változók.

## HAMILTON-ÚT NP-teljes

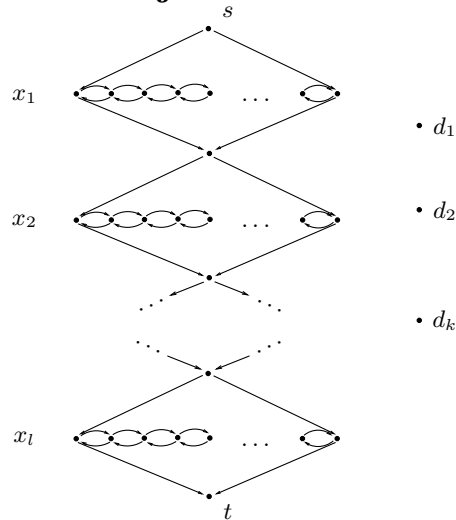
$x_i$ -hez tartozó eszköz:



$d_j$ -hez tartozó eszköz:



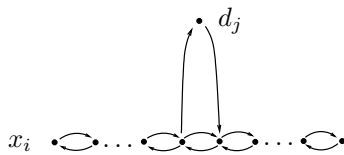
## HAMILTON-ÚT NP-teljes



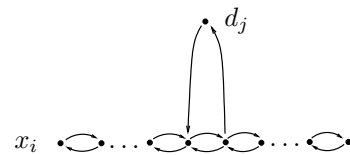
+ a következő fólián  
lévő élek

## HAMILTON-ÚT NP-teljes

$d_j$  tartalmazza  $x_i$ -t



$d_j$  tartalmazza  $\bar{x}_i$ -t



Jelölje  $G_\varphi$  a gráfot. Ekkor:

- (1)  $\langle \varphi \rangle \in 3SAT \Leftrightarrow \langle G_\varphi, s, t \rangle \in \text{HAMILTON-ÚT}$
- (2)  $\langle G_\varphi, s, t \rangle$  polinomidejű Turing-géppel elkészíthető  $\langle \varphi \rangle$ -ből.



## SAT NP-teljes

SAT =  $\{\langle \varphi \rangle : \varphi \text{ kielégíthető knf}\}$

Tétel SAT NP-teljes.

Bizonyítás Csak azt bizonyítjuk, hogy NP-nehéz.

$$L \in \mathbf{NP} \quad L \stackrel{?}{\leq_p} \text{SAT}$$

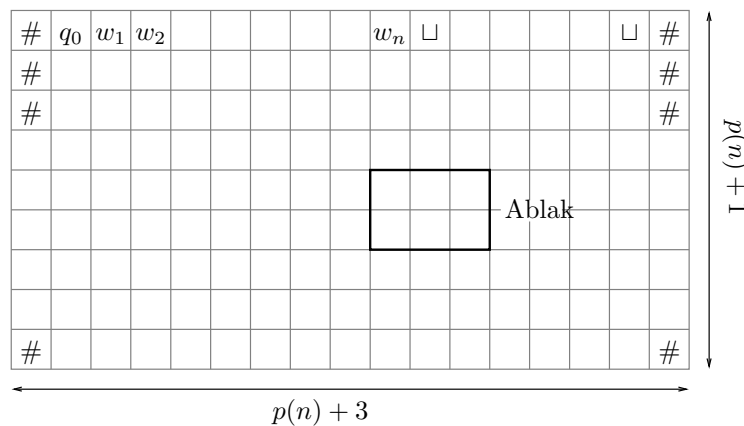
$L = L(M)$   $M$  polinomidőkorlátos nemdet. Turing-gép

Időkorlát:  $p(n) \geq n$

Adott  $w$  bemenő szóhoz el kell készíteni egy  $\varphi$  formulát úgy, hogy:

- (1)  $w \in L \Leftrightarrow \varphi$  kielégíthető,
- (2) a  $w \mapsto \varphi$  hozzárendelés megvalósítható polinom időkorlátos determinisztikus Turing-géppel.

## SAT NP-teljes



$M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$

$1 \leq i \leq p(n) + 1, 1 \leq j \leq p(n) + 3, s \in Q \cup \Gamma \cup \{\#\} \mapsto x_{i,j,s}$  változó

## SAT NP-teljes

Nevezzük az 

|       |       |       |
|-------|-------|-------|
| $a_1$ | $a_2$ | $a_3$ |
| $a_4$ | $a_5$ | $a_6$ |

 elrendezést **legálisnak**, ha megengedett az  $M$  átmeneti relációja által. (Üres átmenet is lehetséges.)

**Állítás** Megadható véges sok olyan  $(a_1, \dots, a_6)$  rendezett legális 6-os úgy, hogy egy táblázat akkor és csak akkor adja meg az  $M$  egy lehetséges működését a  $w$  szón, ha a táblázat első sorában a  $w$ -hez tartozó kezdő konfiguráció van, és minden “ablak” két sora egy legális 6-ost határoz meg.

## SAT NP-teljes

Konvenció:  $x_{i,j,s} = 1$  annak felel meg, hogy a táblázat  $(i, j)$  pozíciójában  $s$  szerepel.

$$\varphi = \varphi_0 \wedge \varphi_{start} \wedge \varphi_{move} \wedge \varphi_{accept}$$

$\varphi_0$ : A táblázat minden egyes mezőjében pontosan egy karakter van, azaz  $\forall i, j \exists! s \ x_{i,j,s}$

$$\varphi_0 = \bigwedge_{i,j} \bigvee_s x_{i,j,s} \wedge \bigwedge_{i,j} \bigwedge_{s \neq t} (\overline{x_{i,j,s}} \vee \overline{x_{i,j,t}})$$

$\varphi_{start}$ : A táblázat első sorában a  $w$ -hez tartozó kezdőkonfiguráció van.

$$\varphi_{start} = x_{1,1,\#} \wedge x_{1,2,q_0} \wedge x_{1,3,w_1} \wedge \dots \wedge x_{1,n+2,w_n} \wedge x_{1,n+3,\square} \wedge \dots \wedge x_{1,p(n)+2,\square} \wedge x_{1,p(n)+3,\#}$$

## SAT NP-teljes

$\varphi_{move}$  : A táblázat minden ablaka legális.

$$\varphi_{move} = \bigwedge_{i,j} \psi_{i,j}$$

$$\psi_{i,j} = \bigvee_{(a_1, \dots, a_6) \text{ legális}} x_{i,j-1,a_1} \wedge x_{i,j,a_2} \wedge x_{i,j+1,a_3} \wedge x_{i+1,j-1,a_4} \wedge x_{i+1,j,a_5} \wedge x_{i+1,j+1,a_6}$$

$$\text{Itt } 1 \leq i \leq p(n), \quad 2 \leq j \leq p(n) + 2$$

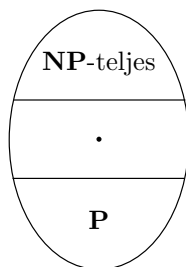
## SAT NP-teljes

$\varphi_{accept}$  : A táblázat utolsó sorában elfogadó konfiguráció van.

$$\varphi_{accept} = \bigvee_{j=2}^{p(n)+2} x_{p(n)+1,j,q_{accept}}$$

## Az NP szerkezete

Tétel Ha  $P \neq NP$ , akkor létezik olyan  $NP$ -beli  $L$  nyelv, amely nincs  $P$ -ben és nem  $NP$ -teljes:



Példa az  $\{(G_1, G_2) : G_1 \text{ és } G_2 \text{ izomorf gráfok}\}$  nyelvről ismert, hogy  $NP$ -ben van, de nem ismert, hogy  $P$ -ben van-e vagy  $NP$ -teljes-e.

## A coNP osztály

Definíció Legyen  $\mathcal{C}$  nyelvek osztálya. Ekkor:  $\text{co}\mathcal{C} = \{\bar{L} : L \in \mathcal{C}\}$ .

Állítás Ha  $\mathcal{C}$  zárt a polinomidejű visszavezetésre, akkor  $\text{co}\mathcal{C}$  is zárt.

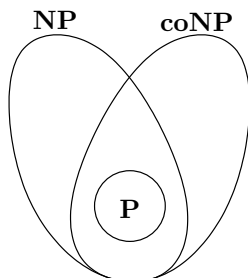
Definíció Legyen  $\mathcal{C}$  nyelvek egy osztálya. Egy  $L$  nyelv  $\mathcal{C}$ -nehéz (a polinomidejű visszavezetésre) ha  $L' \leq_p L$  teljesül minden  $L' \in \mathcal{C}$  nyelvre. Ha  $L$   $\mathcal{C}$ -nehéz és  $L \in \mathcal{C}$ , akkor  $L$   $\mathcal{C}$ -teljes (a polinomidejű visszavezetésre nézve).

Állítás  $L$   $\mathcal{C}$ -nehéz ( $\mathcal{C}$ -teljes)  $\Leftrightarrow \bar{L}$   $\text{co}\mathcal{C}$ -nehéz ( $\text{co}\mathcal{C}$ -teljes).

## A coNP osztály

**Definíció** VALIDITY =  $\{\langle \varphi \rangle : \varphi \text{ azonosan igaz Boole-formula}\}$

**Állítás** VALIDITY coNP-teljes.



Nem ismert, hogy

$$\mathbf{NP} \neq \mathbf{coNP}$$

igaz-e.

(Ha  $\mathbf{P} = \mathbf{NP}$ , akkor  $\mathbf{NP} = \mathbf{coNP}$ , mert  $\mathbf{P} = \mathbf{coP}$  nyilvánvalóan.)

## Tárkonyoltság

**Alapvető különbség:** a tár újra felhasználható.

**További különbség:**

szublineáris tárkonyoltság is érdekes,

a bemenet hossza nem számít a felhasznált tár nagyságába.

**Definíció** Off-line Turing-gép:

- Többszalagos.
- A bemenetet tartalmazó szalagot csak olvashatja. A munkaszalagokra írhat is.

**A tárigénybe csak a munkaszalagokon felhasznált terület számít be.**

## Savitch tétele

**Def.**  $SPACE(f(n)) = \{L : L \text{ eldönthető } \mathcal{O}(f(n)) \text{ tárigényű det. Turing géppel}\}$

$NSPACE(f(n)) = \{L : L \text{ eldönthető } \mathcal{O}(f(n)) \text{ tárigényű nemdet. Turing géppel}\}$

**Tétel (Savitch)** Ha  $f(n) \geq \log n$ , akkor  $NSPACE(f(n)) \subseteq SPACE((f(n))^2)$ .

**Bizonyítás** Tegyük fel, hogy  $L$ -et eldönti az  $M$   $\mathcal{O}(f(n))$  tárigényű nondeterminisztikus Turing-gép. Feltehető, hogy adott  $w$  bemeneten minden elfogadó számítási sorozat ugyanabban az elfogadó konfigurációban végződik.

Ha  $|w| = n$ , a kezdő konfigurációból elérhető konfigurációk száma  $c_1 n 2^{\mathcal{O}(f(n))} = 2^{\mathcal{O}(f(n))}$ , mérete  $\leq c_2 f(n)$ . ( $f(n) \geq \log n$ .)

A továbbiakban csak legfeljebb ilyen méretű konfigurációkat tekintünk.

## Savitch tétele

**TEST** : Adott  $k_1, k_2$  konfigurációkra és  $t, i$  számokra

1. Ha  $t = 1$ , ellenőrizzük, hogy  $k_1 = k_2$  vagy  $k_1$ -ből  $k_2$  legfeljebb egy lépésben megkapható-e.
2. Ha  $t > 1$ , akkor minden  $k$  legfeljebb  $i$  méretű konfigurációra:
  3.  $TEST(k_1, k, \lceil \frac{t}{2} \rceil, i)$
  4.  $TEST(k, k_2, \lceil \frac{t}{2} \rceil, i)$
  5. Ha mindkettő igaz, akkor  $TEST(k_1, k_2, t, i)$  is az.
6. Ha korábban nem ért véget az eljárás **igaz**-zal, akkor  $TEST(k_1, k_2, t, i)$  **hamis**.

## Savitch tétele

Az  $N$  determinisztikus Turing-gép  $i = 1, 2, 3, \dots$  esetén

1. teszteli, hogy a kezdőkonfigurációból elérhető-e az elfogadó konfiguráció úgy, hogy közben csak legfeljebb  $i$  méretű konfiguráción haladunk át,
2. majd ha 1. nem teljesül, teszteli, hogy egyáltalán elérhető-e  $i$  méretű konfiguráció. Ha nem, akkor elutasítja a bemenetet. Különben  $i$ -t 1-gyel növeli.

A legnagyobb  $i$ -érték, melyre  $TEST$  meghívásra kerül  $c_2 f(n)$ -nél nem nagyobb, és a legnagyobb  $t$  érték legfeljebb  $2^{\mathcal{O}(f(n))}$

Rekurzióban a legnagyobb mélység:  $\mathcal{O}(f(n)) = \log 2^{\mathcal{O}(f(n))}$

Egy hívás tárigénye:  $\mathcal{O}(f(n))$

Összes tárigény:  $\mathcal{O}((f(n))^2)$

## A PSPACE és NPSPACE osztályok

### Definíció

$$\mathbf{PSPACE} = \bigcup_{k>0} \mathbf{SPACE}(n^k) \quad \mathbf{NPSPACE} = \bigcup_{k>0} \mathbf{NSPACE}(n^k)$$

### Következmény $\mathbf{PSPACE} = \mathbf{NPSPACE}$

Világos, hogy:  $\mathbf{P} \subseteq \mathbf{NP} \subseteq \mathbf{PSPACE} \subseteq \mathbf{EXP}$ , ahol

$$\mathbf{EXP} = \bigcup_{k>0} \mathbf{TIME}(2^{n^k}).$$

Ismert, hogy  $\mathbf{P} \subset \mathbf{EXP}$ . Az a sejtés, hogy mindegyik tartalmazás valódi.

## QBF PSPACE-teljes

**Def.**  $\text{QBF} = \{ \langle \varphi \rangle : \varphi \text{ igaz prenex alakú zárt Boole-formula} \}$

**Példa**  $\exists x_1 \forall x_2 \exists x_3 [(x_1 \vee x_2) \wedge (x_2 \vee x_3) \wedge (\bar{x}_2 \vee \bar{x}_3)] \in \text{QBF}$   
 $\exists x \forall y [(x \vee y) \wedge (\bar{x} \vee \bar{y})] \notin \text{QBF}$

**Tétel** QBF PSPACE teljes.

**Bizonyítás**  $\text{QBF} \in \text{PSPACE}$

Adott  $\langle \varphi \rangle$  bemeneten

1. Ha  $\varphi$  kvantormentes, akkor csak konstansokat tartalmaz, így értékeljük ki.
2. Ha  $\varphi = \exists x \psi$  alakú, akkor hívjuk az eljárást  $\langle \psi \rangle$ -re először az  $x = 0$  majd az  $x = 1$  értékkel. Ha valamelyik hívás elfogadja bemenetét, akkor  $\langle \varphi \rangle \in \text{QBF}$ , különben  $\langle \varphi \rangle \notin \text{QBF}$ .
3. Ha  $\varphi = \forall x \psi$  alakú, akkor hasonlóan, de abban az esetben fogadjuk el a bemenetet, ha mindkét hívás elfogad.

Tár:  
 $\mathcal{O}(n^2)$

## QBF PSPACE-teljes

**QBF PSPACE-nehéz**

Legyen  $L \in \text{PSPACE}$ , mondjuk  $L$  eldönthető az  $M \mathcal{O}(n^k)$  tárigényű determinisztikus Turing-géppel, ahol  $k \geq 1$ . Most feltehető, hogy a Turing-gép 1-szalagos, mert a tárkorlát szuperlineáris.

Adott  $w$  szóhoz szeretnénk olyan  $|w| = n$ -ben polinom méretű formulát készíteni, mely akkor és csak akkor van QBF-ben, ha  $w \in L(M)$ . Feltehető, hogy  $n$  hosszú szavakon  $M$  elfogadás esetén mindig ugyanabban a konfigurációban áll meg.

A konfigurációk leírásához Boole-változókat használunk, mint Cook tételének bizonyításában. Egy konfiguráció leírásához  $\mathcal{O}(n^k)$  változó szükséges.

Mivel az elérhető konfigurációk hossza  $\mathcal{O}(n^k)$ , ezért az időigény  $2^{\mathcal{O}(n^k)}$ .



## QBF PSPACE-teljes

$\varphi_{c_1, c_2, t}$  jelöljön olyan formulát, mely akkor és csakis akkor igaz, ha a  $c_1$  konfigurációból elérhető  $t$  lépésen belül a  $c_2$  konfiguráció.

$\varphi_{c_1, c_2, 1}$  könnyen felírható, hossza  $\mathcal{O}(n^k)$ .

Legyen  $t > 1$ .

$$\varphi_{c_1, c_2, t} = \exists c [\varphi_{c_1, c, \lceil \frac{t}{2} \rceil} \wedge \varphi_{c, c_2, \lceil \frac{t}{2} \rceil}]$$

Ez a módszer exponenciálisan hosszú formulákat eredményez!

$$\varphi_{c_1, c_2, t} = \exists c \forall d \forall d' \left[ \begin{array}{l} ((d = c_1 \wedge d' = c) \vee \\ (d = c \wedge d' = c_2)) \rightarrow \\ \varphi_{d, d', \lceil \frac{t}{2} \rceil} \end{array} \right]$$

## QBF PSPACE-teljes

### Megjegyzés

- QBF akkor is PSPACE-teljes, ha a formula magja knf, a kvantorok alternálnak, és az első kvantor  $\exists$  (esetleg az utolsó is  $\exists$ ).
- Ekkor QBF felfogható kétszemélyes játékként.

Adott  $\varphi = \exists x_1 \forall x_2 \exists x_3 \dots Q x_n \psi$

- az 1. játékos választja az  $x_1, x_3, \dots$  változók értékét és célja  $\psi$  igazsá tétele,
- a 2. játékos választja felváltva az  $x_2, x_4, \dots$  értékét, célja  $\psi$  hamissá tétele.

$\varphi \in \text{QBF} \Leftrightarrow$  az 1. játékosnak nyerőstratégiája van.

## FÖLDRAJZI JÁTÉK PSPACE-teljes

### FÖLDRAJZI JÁTÉK:

Adott egy  $G$  irányított gráf egy  $p$  kezdőcsúsból. Két játékos  $p$ -ből indulva felváltva egy olyan út csúcsait nevezi meg, mely nem halad át már meglátogatott csúcson. Az a játékos veszít, aki nem tudja folytatni. A probléma annak eldöntése, hogy az első játékosnak van-e nyerő stratégiája.

Tétel A FÖLDRAJZI JÁTÉK PSPACE-teljes.

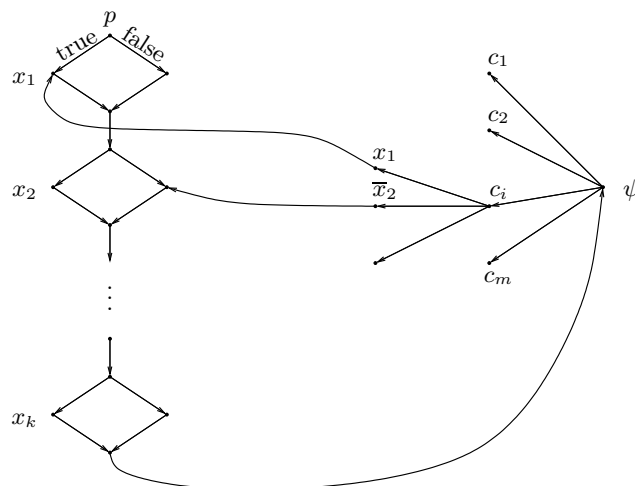
Bizonyítás Az, hogy PSPACE-ben van, a QBF-hez hasonlóan igazolható. Azt, hogy PSPACE-nehéz, a QBF-ből való visszavezetéssel igazoljuk.

Legyen adva a  $\varphi$  formula

$$\varphi = \exists x_1 \forall x_2 \exists x_3 \dots \exists x_k \psi \quad k \text{ páratlan}$$

$$\psi = c_1 \wedge \dots \wedge c_m \quad \text{knf.}$$

## FÖLDRAJZI JÁTÉK PSPACE-teljes



## Az L és NL osztályok

**Definíció**  $L = \text{SPACE}(\log n)$   
 $NL = \text{NSPACE}(\log n)$

**Példa** ELÉRHETŐSÉG  $\in NL$ .

Világos, hogy

$$L \subseteq NL \subseteq P \subseteq PSPACE.$$

Az is ismert, hogy  $NL \subseteq PSPACE$ . Az a sejtés, hogy a fenti tartalmazások mindegyike valódi.

## Logaritmikus tárral való visszavezetés

**Definíció** Legyen  $L_1 \subseteq \Sigma_1^*$ ,  $L_2 \subseteq \Sigma_2^*$ . Azt mondjuk, hogy az  $f : \Sigma_1^* \rightarrow \Sigma_2^*$  függvény az  $L_1$  **logaritmikus tárral való visszavezetése**  $L_2$ -re, ha

$$\forall w \quad (w \in L_1 \Leftrightarrow f(w) \in L_2)$$

és  $f$  kiszámítható  $\mathcal{O}(\log n)$  tárkorlátos determinisztikus Turing-géppel.

**Definíció** Azt mondjuk, hogy  $L_1 \subseteq \Sigma_1^*$  **logaritmikus tárral visszavezethető** az  $L_2 \subseteq \Sigma_2^*$  nyelvre, ha létezik olyan  $f : \Sigma_1^* \rightarrow \Sigma_2^*$  függvény mely az  $L_1$ -nek az  $L_2$ -re való logaritmikus tárral való visszavezetése. **Jelölés:**  $L_1 \leq_l L_2$ .

**Állítás** Ha  $L_1 \leq_l L_2$  akkor  $L_1 \leq_p L_2$ .

**Tétel** A  $\leq_l$  reláció előrendezés.

## Logaritmikus tárral való visszavezetés

Tétel Az ELÉRHETŐSÉG NL-teljes a logaritmikus tárral való visszavezetésre.

Tétel  $NL = coNL$ .

## Hierarchia

