Matematikai és számítástudományi ismeretek

6. tétel

1. ELSŐ RÉSZ

1.1. Gráf fogalma és megadásának módjai.

1.1.1. A gráf fogalma

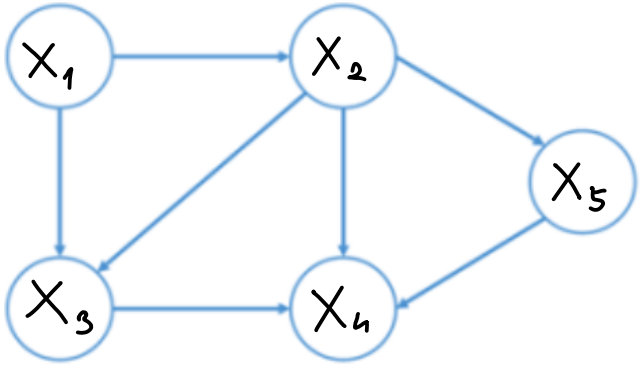
A gráf hálós adatszerkezet.

Hálós adatszerkezet: minden adatelemnek tetszőleges számú megelőzője és tetszőleges számú rákövetkezője lehet. Egy elem lehet egy másiknak (beleértve saját magát is) a megelőzője, rákövetkezője, mindkettő vagy egyik sem.

A gráf csúcsok és élek halmaza. Egy él két csúcs közötti kapcsolat. Egy gráfot az határoz meg, hogy mely csúcsai vannak élekkel összekötve.

1.1.2. A gráf megadási módjai

1. Ábrával



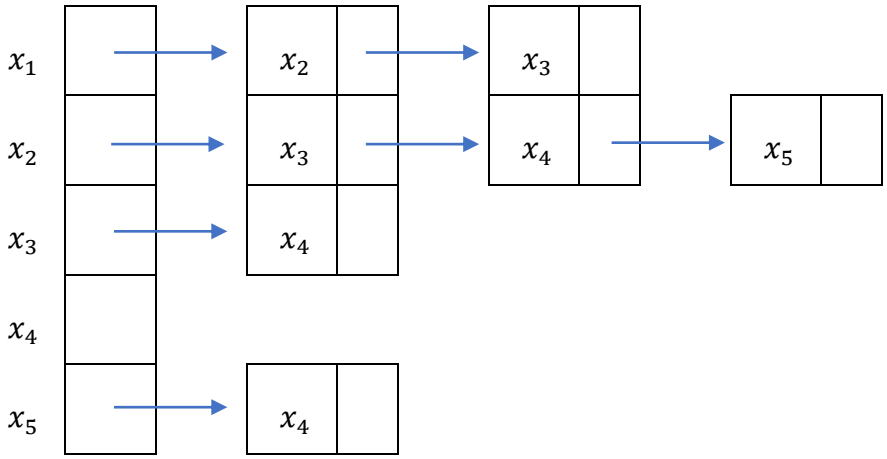
2. Az N ponthalmaz (csúcshalmaz) és az A élhalmaz tételes felsorolásával

$$N = \{ x_1, x_2, x_3, x_4, x_5 \}$$

$$A = \{ (x_1, x_2), (x_1, x_3), (x_2, x_3), (x_2, x_4), (x_2, x_5), (x_3, x_4), (x_5, x_4) \}$$

3. Szétszórt reprezentáció – Szomszédsági listával (multilistával)

A kezdő csúcspontból listaszerűen felsoroljuk az onnan elérhető csúcspontokat. A listaelem az adatelem értékén kívül egy mutatót tartalmaz, amely a következő listaelem címét tartalmazza.



4. Folytonos reprezentáció – Szomszédsági mátrixszal (csúcsmátrixszal)

Ahol i a sor, j az oszlop, n a csúcsok száma, ekkor a gráfot egy $n \times n$ -es mátrixszal ábrázoljuk, a sorokat és az oszlopokat a gráf csúcsaival címkézzük.

Egy címkézetlen M mátrix formájú ábrázolás esetén a következő képlettel írhatjuk le a mátrix kitöltésének a módját (E az élek halmaza)

$$C[i,j] = \begin{cases} 1 & \text{ha } (i,j) \in E \\ 0 & \text{ha } (i,j) \notin E \end{cases}$$

	x_1	x_2	x_3	x_4	x_5
x_1	0	1	1	0	0
x_2	0	0	1	1	1
x_3	0	0	0	0	1
x_4	0	0	0	0	0
x_5	0	0	0	1	0

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

1.2. Egyszerű, irányított és irányítatlan gráfok.

1.2.1. Egyszerű gráf:

Ebben a gráfban bármely két csúcs között legfeljebb egy él lehet (kizárjuk a többszörös éleket) és nem engedünk meg olyan éleket, amelyeknek kezdő- és végpontja azonos (a hurkokat).

1.2.2. Irányított gráf:

Ezesetben az éleknek irányuk van.

1.2.3. Irányítatlan gráf:

Ezesetben az élekhez nincs irány rendelve, vagyis nem teszünk különbséget az „A-ból B-be”, illetve a „B-ből A-ba” menő élek között.

1.3. Séta, út, összefüggőség.

1.3.1. Séta:

Sétának nevezzük egy gráf csúcsainak és éleinek olyan sorozatát, melyben minden él végpontja megegyezik a következő él kezdőpontjával – feltéve, hogy létezik következő él. A sétában a csúcsok és az élek tetszés szerint ismétlődhetnek.

1.3.2. Út:

Útnak nevezzük a csúcsok és élek olyan sorozatát, amelyben nem ismétlünk sem éleket, sem csúcsokat.

1.3.3. Összefüggőség:

Egy gráf összefüggő, ha (élei esetleges irányításáról megfeledkezve) bármely két csúcs között van út.

1.4. Nevezetes gráfok: páros gráf, teljes gráf, fa, kör, súlyozott gráf.

1.4.1. Páros gráf:

Egy gráf páros, ha nincs benne páratlan hosszúságú kör.

1.4.2. Teljes gráf:

Olyan gráf, melynek bármely két csúcsa között van él.

1.4.3. Fa:

Olyan gráf, mely összefüggő és körnélküli. A fa csúcsainak száma=élek száma+1

1.4.4. Kör:

Körnek nevezzük azt az utat, amelynek kezdő- és végpontja azonos.

1.4.5. Súlyozott gráf:

Súlyozott gráf esetében egy w súlyfüggvény is rendelhető az élekhez.

2. MÁSODIK RÉSZ

2.1. Generatív nyelvtanok, nyelvosztályok, a Chomsky-hierarchia.

2.1.1. Generatív nyelvtanok:

$G = (N, \Sigma, S, P)$, ahol N a nemterminális ábécé, Σ terminális ábécé, S kezdőszimbólum, P helyettesítési szabályok. $L(G)$ a G által generált nyelv (szavak halmaza).

Tehát a generatív grammatika alkotóelemei:

- Nemterminális ábécé – segédszimbólumok a generálás során
- Terminális ábécé – a generálandó nyelv ábécéje
- Kezdő nemterminális – kezdőszimbólum
- Helyettesítési szabályok – a generálás során mely szavak helyettesíthetők mely más szavakkal

Példa: $N = \{S\},\ \Sigma = \{a, b\},\ S \in N,\ P = \{S \rightarrow \lambda \mid S \rightarrow aSb\}$ (λ az üres szó)

2.1.2. Nyelvosztályok:

A különböző nyelvtanokat bizonyos formai tulajdonságok alapján osztályokba soroljuk. Az osztályozás alapját a helyettesítési szabályok alakjára vonatkozó megszorítások képezik abban a hierarchiában, amelyet az elmélet egyik megalapozója, Noam Chomsky vezetett be. Ő alkotta meg

azt a 4 nyelvosztályt, amelyeket a mai napig a grammatikák és nyelvek kategorizálására használunk.

0. típusú nyelvtanok – rekurzívan felsorolható nyelvtanok:

$\alpha \to \beta$, ahol α és β nemterminálisokból és terminálisokból álló szavak, és α tartalmaz legalább egy nemterminális szimbólumot.

1. típusú nyelvtanok – környezetfüggő nyelvtanok:

$\alpha\, A\, \beta \to \alpha\, \gamma\, \beta$ vagy $S \to \lambda$, ahol A nemterminális, γ egy nemterminálisokból és terminálisokból álló akár üres szó, α és β nemterminálisokból és terminálisokból álló szavak

2. típusú nyelvtanok – környezetfüggetlen nyelvtanok:

$A \to \alpha$, ahol A nemterminális, α egy nemterminálisokból és terminálisokból álló, akár üres, szó.

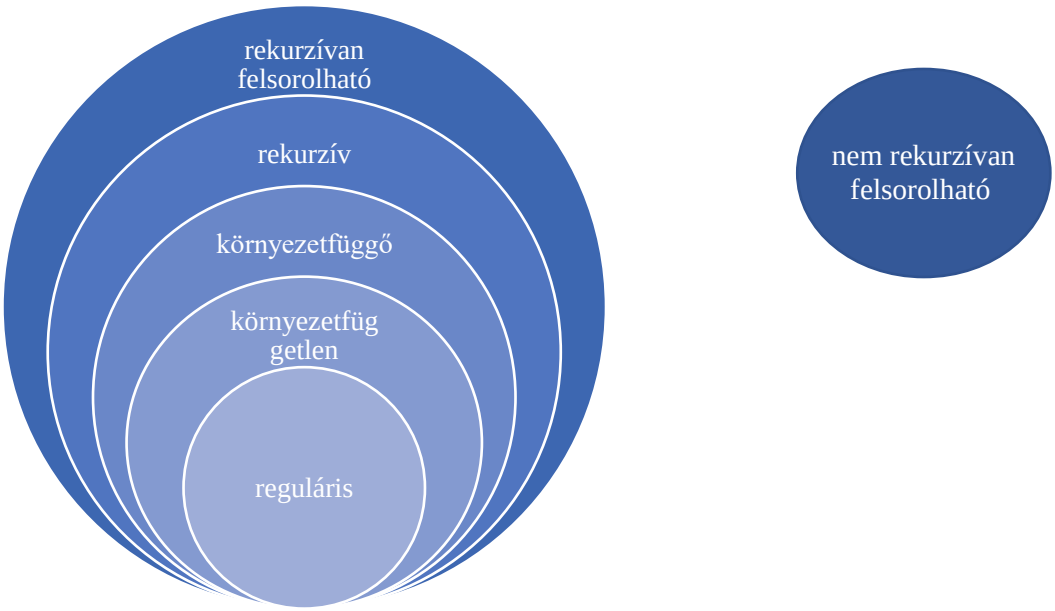
3. típusú nyelvtanok – reguláris/szabályos nyelvtanok:

$A \to aB \mid a \mid \lambda$, ahol A, B nemterminálisok, a terminális

2.1.3. Chomsky-hierarchia:

$$L(REG) \subset L(CF) \subset \mathrm{L(CS)} \subset \mathrm{L(REC)} \subset L(RE)$$

ahol REG = reguláris, CF = környezetfüggetlen (context-free), CS = környezetfüggő (context-sensitive), RE = rekurzívan felsorolható (recursively enumerable).



Reguláris nyelvtan:

$$A \to aB \mid a \mid \lambda \ , \qquad A, B \in N, \ \ a \in \Sigma$$

Környezetfüggetlen nyelvtan:

$$A \to \alpha \ (\mathrm{A} \to \text{bármi}) \ , \qquad A \in N, \ \ \alpha \in (N \cup \Sigma)^*$$

Környezetfüggő nyelvtan:

$$\alpha \, A \, \beta \ \to \alpha \, \gamma \, \beta \ , \qquad |\alpha \, A \, \beta \mid \leq |\alpha \, \gamma \, \beta|$$

Rekurzív nyelv:

> Egy L nyelv rekurzív, ha van olyan Turing gép, ami minden bemeneten megáll, a $w \in L$ szavakon elfogadó állapotban áll meg, a $w \notin L$ szavakon pedig nem elfogadó állapotban áll meg. (T Turing gép eldönti L-et)

Rekurzívan felsorolható nyelv:

> Egy L nyelv rekurzívan felsorolható, ha van olyan Turing gép, ami minden $w \in L$ szü bemeneten elfogadó állapotban áll meg. A $w \notin L$ szavakon vagy nem elfogadó állapotban áll meg, vagy egyáltalán nem áll meg. (T Turing gép elfogadja L-et)

2.2. Véges automaták, lineáris idejű felismerés, veremautomaták.

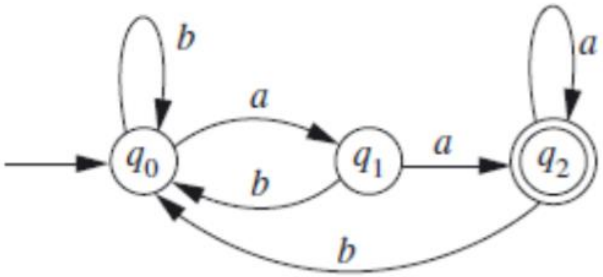
2.2.1. Véges automaták:

Determinisztikus véges automata

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

ahol Q = véges állapothalmaz, Σ = véges bemeneti ábécé, $q_0 \in Q$ = kezdőállapot, $A \subseteq$ Q = vég/elfogadási állapotok, $\delta: Q \times \Sigma$ = állapotátmenet függvény.

Példa:



$M = (\{q_0, q_1, q_2\}, \{a, b\}, q_0, \{q_2\}, \delta)$, ahol δ a következő:

$\delta(q_0, a) = q_1$ $\quad$ $\delta(q_1, a) = q_2$ $\quad$ $\delta(q_2, a) = q_2$

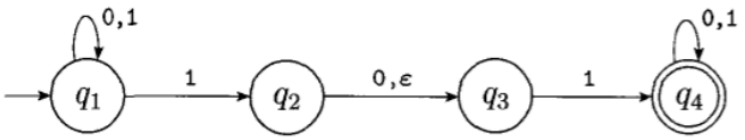
$\delta(q_0, b) = q_0$ $\quad$ $\delta(q_1, b) = q_0$ $\quad$ $\delta(q_2, b) = q_0$

Nemdeterminisztikus véges automata

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

ahol Q = véges állapothalmaz, Σ = véges bemeneti ábécé, $q_0 \in Q$ = kezdőállapot, $A \subseteq$ Q = vég/elfogadási állapotok, $\delta: Q \times (\Sigma \cup \{\lambda\}) \rightarrow 2^Q$ = állapotátmenet függvény.

Példa:

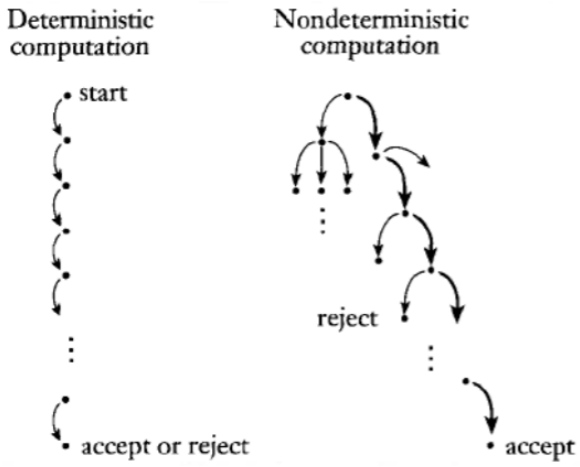


Több lehetőség ugyanarra a bemenetre és megjelenik az üresszó átmenet

$M = (\{q_1, q_2, q_3, q_4\},\ \{0,1\},\ q_1,\ \{q_4\},\ \delta)$, ahol δ a következő:

	0	1	2
q_1	q_1	q_1, q_2	
q_2	q_3		q_3
q_3		q_4	
q_4	q_4	q_4	

Determinisztikus véges automata vs Nemdeterminisztikus véges automata



2.2.2. Lineáris idejű felismerés:

A reguláris nyelvek esetén a szóprobléma nagyon hatékonyan megoldható. Ha megszerkesztünk egy az adott nyelvet elfogadó determinisztikus véges automatát, akkor annak segítségével a szót betűnkként elolvasva végig követve az automata futását (legkésőbb) a szó végére érve megkapjuk a választ a kérdésre: ha végállapotba jutottunk a szó végén, akkor a szó benne van az adott reguláris nyelvben; ha nem végállapotba jutottunk, vagy (parciális automata esetén) időközben elakadtunk a feldolgozással, akkor a keresett szó nincs a nyelvben.

Tehát a probléma valós időben megoldható, ahány betűből áll az input szó, annyi lépés után tudjuk a választ.

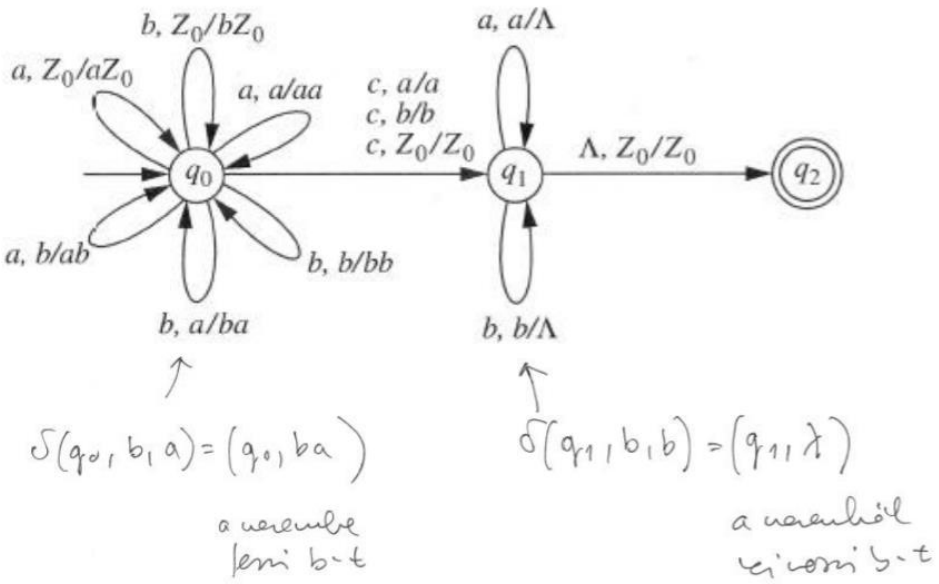
2.2.3. Veremautomaták:

Definíció:

$$M = (Q, \mathrm{T}, \Gamma, q_0, Z_0, \delta, F)$$

ahol Q = állapothalmaz, T = bemeneti ábécé, Γ = veremábécé, $q_0 \in Q$ = kezdőállapot, $\mathrm{Z}_0 \in \Gamma$ = kezdeti veremtartalom, δ = állapotátmenet reláció, $F \subseteq \mathrm{Q}$ = végállapotok halmaza.

Példa:



Move number	State	Input	Stack symbol	Move(s)
1	q_0	a	Z_0	(q_0, aZ_0)
2	q_0	b	Z_0	(q_0, bZ_0)
3	q_0	a	a	(q_0, aa)
4	q_0	b	a	(q_0, ba)
5	q_0	a	b	(q_0, ab)
6	q_0	b	b	(q_0, bb)
7	q_0	c	Z_0	(q_1, Z_0)
8	q_0	c	a	(q_1, a)
9	q_0	c	b	(q_1, b)
10	q_1	a	a	(q_1, Λ)
11	q_1	b	b	(q_1, Λ)
12	q_1	Λ	Z_0	(q_2, Z_0)
	(all other combinations)			none