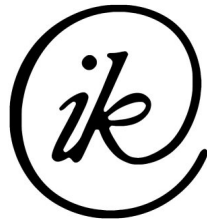


# ZÁRÓVIZSGA TÉTELKIDOLGOZÁS 2015

Szerkesztette: *Balogh Tamás*

2015. június 20.



Ha hibát találsz, kérlek jelezd a [info@baloghtamas.hu](mailto:info@baloghtamas.hu) e-mail címen!



Ez a Mű a Creative Commons Nevezd meg! - Ne add el! - Így add tovább! 3.0 Unported Licenc feltételeinek megfelelően szabadon felhasználható.

## Előszó

A kidolgozásban szereplő tételek teljességéért, hibamentességéért nem vállalok felelősséget. Ha egy új tétel elkészül, azzal automatikusan bővíteni fogom a dokumentumot, de biztosan nem fog elkészülni az összes tétel. A tételek kidolgozása során felhasználtam a tantárgyakhoz tartozó előadásdiákat, az előadók által írt könyveket, a saját jegyzeteimet, valamint az internetet fellelhető anyagokat is. A *Mesterséges Intelligencia* tétel Dr. Gregorics Tibor munkája.

# Tartalomjegyzék

<b>1. Függvények határértéke és folytonossága</b>	<b>6</b>
1.1. Reláció és függvény . . . . .	6
1.2. Függvények határértéke . . . . .	6
1.3. Függvények folytonossága . . . . .	7
1.4. Kompakt halmazon folytonos függvények tulajdonságai . . . . .	8
1.5. Hatványsorok . . . . .	9
1.6. Analitikus függvények . . . . .	10
<b>2. Differenciál- és integrálszámítás</b>	<b>11</b>
2.1. Differenciálszámítás . . . . .	11
2.2. Függvényvizsgálat . . . . .	12
2.2.1. Előzetes vizsgálatok . . . . .	12
2.2.2. Folytonosság . . . . .	12
2.2.3. Monotonitás . . . . .	12
2.2.6. Szélsőérték . . . . .	12
2.2.7. Konvexitás . . . . .	13
2.3. Integrálszámítás . . . . .	13
2.3.4. Riemann-integrál . . . . .	13
2.4. Differenciálegyenlet . . . . .	14
<b>3. Numerikus módszerek</b>	<b>16</b>
<b>4. Számelmélet, gráfok, kódoláselmélet</b>	<b>17</b>
4.1. Relációk, rendezések . . . . .	17
4.2. Függvények és műveletek . . . . .	17
4.3. Számfogalom, komplex számok . . . . .	18
4.4. Leszámlálások véges halmazokon . . . . .	19
4.5. Számelméleti alapfogalmak, lineáris kongruencia-egyenletek . . . . .	19
4.6. Általános és síkgráfok, Euler- és Hamilton gráfok, gráfok adatszerkezetei . . . . .	20
4.7. Polinomok és műveleteik, maradékos osztás, Horner-séma . . . . .	21
4.8. Betűnkénti kódolás, Shannon- és Huffman kód. Hibajavító kódok, kódtávolság, lineáris kódok . . . . .	22
<b>5. Valószínűségszámítási és statisztikai alapok</b>	<b>23</b>
5.1. Diszkrét és folytonos valószínűségi változók . . . . .	23
5.2. Statisztikai becslések . . . . .	26
5.3. Statisztikai próbák . . . . .	26
5.3.1. u-próbák . . . . .	26
5.3.2. t-próbák . . . . .	27
<b>6. Mesterséges intelligencia</b>	<b>28</b>
6.1. Útkeresési probléma . . . . .	28
6.2. Állapottér reprezentáció . . . . .	28
6.3. Keresések . . . . .	28
6.3.1. Lokális keresések . . . . .	29
6.3.2. Visszalépéses keresések . . . . .	29
6.3.3. Gráfkeresések . . . . .	30

6.3.4. Kétszemélyes játékok . . . . .	31
<b>7. Programozás</b>	<b>32</b>
7.1. Egyszerű programozási feladat megoldásának lépései . . . . .	32
7.1.1. Specifikáció . . . . .	32
7.1.2. Tervezés . . . . .	32
7.1.3. Megvalósítás . . . . .	32
7.1.4. Tesztelés . . . . .	33
7.2. Az adattípus fogalma . . . . .	33
7.2.6. Szövegesen . . . . .	34
7.3. Visszavezetés . . . . .	34
7.4. Felsoroló specifikációja . . . . .	34
7.5. Felsorolóra megfogalmazott programozási tételek . . . . .	34
7.5.1. Összegzés . . . . .	34
7.5.2. Számlálás . . . . .	35
7.5.3. Maximum kiválasztás . . . . .	35
7.5.4. Feltételes maximumkeresés . . . . .	36
7.5.5. Kiválasztás . . . . .	36
7.5.6. Lineáris keresés . . . . .	36
7.6. Nevezets gyűjtemények . . . . .	37
7.6.1. Intervallum . . . . .	37
7.6.2. Sorozat . . . . .	37
7.6.3. Tömb . . . . .	37
7.6.4. Szekvenciális fájl . . . . .	37
7.6.5. Halmaz . . . . .	38
<b>8. Programfejlesztési modellek</b>	<b>39</b>
<b>9. Programok fordítása és végrehajtása</b>	<b>39</b>
<b>10. Programnyelvi alapok</b>	<b>39</b>
<b>11. Formális nyelvek</b>	<b>40</b>
11.1. Alapfogalmak . . . . .	40
11.2. Formális nyelvtanok . . . . .	40
11.3. Nyelvtanok osztályozása . . . . .	40
11.3.1. Jelölések . . . . .	40
11.3.2. Chomsky-féle nyelvhierarchia . . . . .	41
11.4. Környezetfüggetlen nyelvek tulajdonságai és elemzésük . . . . .	41
11.4.2. Elemzés . . . . .	41
11.5. Reguláris nyelvek tulajdonságai és alkalmazásai . . . . .	41
11.5.5. Felhasználási területek . . . . .	42
11.6. Automaták . . . . .	42
<b>12. Logika és számításelmélet</b>	<b>44</b>
<b>13. Alapvető algoritmusok és adatszerkezetek</b>	<b>44</b>
<b>14. Haladó algoritmusok</b>	<b>44</b>

<b>15. Operációs rendszerek</b>	<b>44</b>
<b>16. Számítógépes hálózatok és Internet eszközök</b>	<b>44</b>
16.1. Rétegek . . . . .	44
16.2. Fizikai réteg . . . . .	45
16.2.1. Feladata . . . . .	45
16.2.2. NRZ-L . . . . .	45
16.2.3. Manchester kód . . . . .	45
16.2.4. Alapsáv és szélessáv . . . . .	45
16.2.5. Médiumok . . . . .	46
16.3. Adatkapcsolati réteg . . . . .	46
16.3.1. Feladata . . . . .	46
16.3.2. Keretezés . . . . .	46
16.3.3. Hibafelügyelet . . . . .	46
16.3.6. Folyamfelügyelet . . . . .	47
16.3.7. Médiumhozzáférés . . . . .	48
16.4. Hálózati réteg . . . . .	48
16.4.1. Feladata . . . . .	48
16.4.2. Internet Protokoll . . . . .	48
16.4.3. Statikus routing . . . . .	48
16.4.4. Dinamikus routing . . . . .	49
16.4.5. DHCP . . . . .	49
16.5. Szállítói réteg . . . . .	49
16.5.1. Feladata . . . . .	49
16.5.2. TCP . . . . .	49
16.5.3. UDP . . . . .	49
<b>17. Osztott rendszerek</b>	<b>50</b>
<b>18. Adatbázisok tervezése és lekérdezése</b>	<b>50</b>
<b>19. Adatbázisok optimalizálása és konkurencia kezelése</b>	<b>50</b>
<b>20. Felhasznált irodalom</b>	<b>50</b>

# 1. Függvények határértéke és folytonossága

FÜGGVÉNYEK HATÁRÉRTÉKE, FOLYTONOSSÁGA. KOMPAKT HALMAZON FOLYTONOS FÜGGVÉNYEK TULAJDONSÁGAI: HEINE-TÉTEL, WEIERSTRASS-TÉTEL, BOLZANO-TÉTEL. A HATVÁNSOR FOGALMA, CAUCHY-HADAMARD-TÉTEL, ANALITIKUS FÜGGVÉNYEK.

---

## 1.1. Reláció és függvény

**1.1.1 Definíció.** Legyen  $A$  és  $B$  két tetszőleges halmaz. Az  $A \times B$  az  $A$  és a  $B$  halmaz Descartes-szorzata, melynek elemei rendezett párok:

$$(a, b) := \{\{a\}, \{a, b\}\} \quad (a \in A, b \in B).$$

Az  $A \times B$  Descartes-szorzat bármely részhalmazát (bináris) relációnak nevezzük. Ha  $\emptyset \neq r \subset A \times B$  és valamilyen  $a \in A, b \in B$  elemek esetén  $(a, b) \in r$ , akkor az  $a, b$  elemek ( $r$ -)relációban vannak. Az  $r$  értelmezési tartománya:

$$\mathcal{D}_r := \{x \in A : \exists (x, y) \in r\},$$

Az értékkészlete pedig

$$\mathcal{R}_r := \{y \in B : \exists (x, y) \in r\}.$$

**1.1.2 Definíció.** Legyenek  $\emptyset \neq A, B$  halmazok, és  $f \subset A \times B$  reláció. Tetszőleges  $x \in A$  elemre legyen

$$f_x := \{y \in B : (x, y) \in f\}.$$

Az  $f$  reláció függvény, ha tetszőleges  $x \in A$  választással az  $f_x$  halmaz legfeljebb egy elemű. Ekkor tehát

$$\forall x \in \mathcal{D}_f \exists! y \in \mathcal{R}_f : (x, y) \in f.$$

## 1.2. Függvények határértéke

Az  $f$  számsorozat, ha  $\mathcal{D}_f = \mathbb{N}$ . Ekkor  $\exists \lim f$  és  $\lim f = A \in \overline{\mathbb{K}}$ , ha  $\forall K(A)$  környezet esetén  $f_n = f(n) \in K(A)$ . A számsorozatok speciális  $f \in \mathbb{K}_1 \rightarrow \mathbb{K}_2$  függvények, a határérték kiterjeszhető tetszőleges  $f \in \mathbb{K}_1 \rightarrow \mathbb{K}_2$  függvényekre, ahol (egymástól függetlenül)  $\mathbb{K}_1, \mathbb{K}_2 = \mathbb{R}$ , vagy  $\mathbb{C}$ .

Legyen tehát  $f \in \mathbb{K}_1 \rightarrow \mathbb{K}_2$  és  $a \in \mathcal{D}'_f$ . Ekkor az  $f$  függvénynek az  $a$  helyen van határértéke, ha  $\exists A \in \overline{\mathbb{K}_2}$ , amelyre tetszőleges  $K(A) \subset \mathbb{K}_2$  környezetet is véve megadható az  $a$ -nak olyan  $k(a) \subset \mathbb{K}_1$  környezete, amellyel

$$f[k(a) \setminus \{a\} \cap \mathcal{D}_f] \subset K(A)$$

teljesül, vagyis

$$f(x) \in K(A) \quad (a \neq x \in k(a) \cap \mathcal{D}_f).$$

**1.2.1 Tétel.** Tegyük fel, hogy az  $f \in \mathbb{K}_1 \rightarrow \mathbb{K}_2$  függvénynek az  $a \in \mathcal{D}'_f$  helyen van határértéke. Ekkor  $\exists! A \in \overline{\mathbb{K}_2}$ , amely eleget tesz a definíciónak, azaz

$$\forall K(A) \subset \mathbb{K}_2 \exists k(a) \subset \mathbb{K}_1 : f(x) \in K(A) \quad (a \neq x \in k(a) \cap \mathcal{D}_f).$$

**1.2.2 Definíció (Függvényhatárérték fogalma).** Tegyük fel, hogy  $f \in \mathbb{K}_1 \rightarrow \mathbb{K}_2$  függvénynek az  $a \in \mathcal{D}'_f$  helyen létezik határértéke. A definícióban szereplő  $A \in \overline{\mathbb{K}_2}$  szám, vagy "valamelyik végtelen" az  $f$  függvény  $a$  helyen vett határértéke. Jelölése:

$$\lim_a f := \lim_{x \rightarrow a} f(x) := A.$$

**1.2.3 Tétel (Átviteli elv).** Legyen  $f \in \mathbb{K}_1 \rightarrow \mathbb{K}_2$  függvény és  $a \in \mathcal{D}'_f$ .

$$\exists \lim_a f = A \Leftrightarrow \forall (x_n): \mathbb{N} \rightarrow \mathcal{D}_f \setminus \{a\}, \lim(x_n) = a \text{ sorozatra } \exists \lim(f(x_n)) = A.$$

Ha létezik az  $A := \lim_a f$ , akkor az előbb említett  $\forall (x_n)$  sorozatra

$$\lim(f(x_n)) = A.$$

**1.2.4 Tétel.** Tegyük fel, hogy  $f, g \in \mathbb{K}_1 \rightarrow \mathbb{K}_2$ ,  $\mathcal{D}_f \cap \mathcal{D}_g \neq \emptyset$ , valamint az  $a \in (\mathcal{D}_f \cap \mathcal{D}_g)'$  torlódási pontban  $\exists A := \lim_a f$ ,  $B := \lim_a g$  határértékek. Tegyük fel továbbá, hogy

i,  $\exists A + B \in \overline{\mathbb{K}_2}$  összeg, és az  $f + g$  összegfüggvénynek az  $a$  helyen van határértéke, és

$$\lim_a (f + g) = A + B,$$

ii,  $c \in \mathbb{K}_2$  és  $\exists cA \in \overline{\mathbb{K}_2}$  szorzat, és  $cf$  függvénynek az  $a$  helyen van határértéke, és

$$\lim_a (cf) = cA,$$

iii,  $\exists AB \in \overline{\mathbb{K}_2}$ , és az  $fg$  szorzatfüggvénynek az  $a$  helyen van határértéke, és

$$\lim_a (fg) = AB,$$

iv,  $\exists A/B \in \overline{\mathbb{K}_2}$ , és az  $f/g$  hányadosfüggvénynek az  $a$  helyen van határértéke, és

$$\lim_a \frac{f}{g} = \frac{A}{B}.$$

### 1.3. Függvények folytonossága

**1.3.1 Definíció.** Az  $f \in \mathbb{K}_1 \rightarrow \mathbb{K}_2$  függvény az  $a \in \mathcal{D}_f$  pontban folytonos, ha

$$\forall \varepsilon > 0 \exists \delta > 0: |f(x) - f(a)| < \varepsilon \quad (x \in \mathcal{D}_f, |x - a| < \delta).$$

Jelöljük az  $a \in \mathbb{K}_1$  pontban folytonos  $f \in \mathbb{K}_1 \rightarrow \mathbb{K}_2$  függvények halmazát a  $C\{a\}$  jelöléssel.

Speciálisan ha  $\mathbb{K}_1 = \mathbb{K}_2 = \mathbb{R}$ , azaz az  $f \in \mathbb{R} \rightarrow \mathbb{R}$ , akkor

$$f(a) - \varepsilon < f(x) < f(a) + \varepsilon \quad (x \in \mathcal{D}_f, a - \delta < x < a + \delta).$$

**1.3.2 Definíció.** Az  $f \in \mathbb{K}_1 \rightarrow \mathbb{K}_2$  függvény folytonos, ha  $\forall a \in \mathcal{D}_f$  esetén  $f \in C\{a\}$ . A folytonos függvényeket jelöljük a következőképp:  $f \in C$ .

**1.3.3 Megjegyzés.** Legyen  $f \in \mathbb{K}_1 \rightarrow \mathbb{K}_2$ ,  $a \in \mathcal{D}_f$  és  $f \notin C\{a\}$ . Ekkor az  $f$  függvénynek az  $a$  helyen szakadása van.

**1.3.4 Tétel (Folytonosság és műveletek kapcsolata).** Tegyük fel, hogy  $f, g \in C\{a\}$ . Ekkor

$$i, f + g \in C\{a\},$$

$$ii, fg \in C\{a\},$$

$$iii, \text{ ha } g(a) \neq 0, \text{ akkor } f/g \in C\{a\}.$$

**1.3.5 Tétel (Átviteli elv).** Legyen  $f \in \mathbb{K}_1 \rightarrow \mathbb{K}_2, a \in \mathcal{D}_f$ . Ekkor

$$f \in C\{a\} \Leftrightarrow \forall (x_n): \mathbb{N} \rightarrow \mathcal{D}_f, \lim (x_n) = a: \lim (f(x_n)) = f(a).$$

**1.3.6 Tétel.** Legyen  $f, g \in \mathbb{K}_1 \rightarrow \mathbb{K}_2, g \in C\{a\}, f \in C\{g(a)\}$ . Ekkor

$$f \circ g \in C\{a\}.$$

## 1.4. Kompakt halmazon folytonos függvények tulajdonságai

**1.4.1 Definíció (Zárt halmaz).** Az  $A \subset \mathbb{R}$  halmaz zárt, ha  $A = \emptyset$ , vagy  $A \neq \emptyset$ , és tetszőleges konvergens  $(x_n): \mathbb{N} \rightarrow A$  sorozatra a  $\lim (x_n) \in A$ .

**1.4.2 Definíció.** A  $\emptyset \neq A \subset \mathbb{K}$  halmaz felülről (alulról) korlátos, ha

$$\exists k \in \mathbb{R} \forall x \in A: x \leq k \quad (x \geq k).$$

Az  $A$  halmaz korlátos, ha alulról és felülről is korlátos.

**1.4.3 Definíció.** A  $\emptyset \neq A \subset \mathbb{K}$  halmaz kompakt, ha korlátos és zárt.

**1.4.4 Tétel.** A  $\emptyset \neq A \subset \mathbb{K}$  halmaz kompakt, akkor és csak akkor, ha  $\forall (x_n): \mathbb{N} \rightarrow A$  sorozatnak  $\exists (x_{v_n})$  részsorozata, amely konvergens és  $\lim (x_{v_n}) \in A$ .

**1.4.5 Tétel.** Ha az  $f \in \mathbb{K}_1 \rightarrow \mathbb{K}_2 \in C$ , és a  $\mathcal{D}_f$  kompakt, akkor az  $\mathcal{R}_f$  is kompakt.

**1.4.6 Tétel (Weierstrass).** Ha az  $f \in \mathbb{K} \rightarrow \mathbb{R} \in C$  függvény értelmezési tartománya kompakt, akkor az értékkészletének létezik legkisebb és legnagyobb eleme.

**1.4.7 Definíció.** Az  $f \in \mathbb{K}_1 \rightarrow \mathbb{K}_2$  függvény egyenletesen folytonos, ha

$$\forall \varepsilon > 0 \exists \delta > 0: |f(x) - f(t)| < \varepsilon \quad (x, t \in \mathcal{D}_f, |x - t| < \delta).$$

**1.4.8 Tétel (Heine).** Tegyük fel, hogy  $f \in \mathbb{K}_1 \rightarrow \mathbb{K}_2 \in C$ , és  $\mathcal{D}_f$  kompakt. Ekkor az  $f$  egyenletesen folytonos.

**1.4.9 Tétel (Bolzano).** Tegyük fel, hogy valamilyen  $-\infty < a < b < +\infty$  esetén az  $f: [a, b] \rightarrow \mathbb{R} \in C$ , és  $f(a) < 0 < f(b)$ . Ekkor

$$\exists \xi \in (a, b): f(\xi) = 0.$$



## 1.5. Hatványsorok

A végtelen sorok egyik fontos osztálya a függvénysorok. Ezek értelmezéséhez nevezzünk egy  $(f_n)$  sorozatot függvénysorozatnak, ha  $\forall n \in \mathbb{N}$  esetén  $f_n$  függvény. Feltesszük, hogy valamilyen  $\mathcal{D} \neq \emptyset$  halmazzal

$$\mathcal{D}_{f_n} = \mathcal{D} \quad (n \in \mathbb{N}).$$

Ha az  $(f_n)$  függvénysorozatra még az is igaz, hogy  $\mathcal{R}_{f_n} \subset \mathbb{K}$  ( $n \in \mathbb{N}$ ), azaz

$$f_n: \mathcal{D} \rightarrow \mathbb{K} \quad (n \in \mathbb{N}),$$

akkor van értelme az  $(f_n)$  függvénysorozat által meghatározott  $\sum(f_n)$  függvénysornak:

$$\sum(f_n) := \left( \sum_{k=0}^n f_k \right).$$

A  $\sum(f_n)$  függvénysor tehát az

$$F_n := \sum_{k=0}^n f_k \quad (n \in \mathbb{N})$$

részletösszeg-függvények által generált  $(F_n)$  függvénysorozat.

Ha valamilyen  $t \in \mathcal{D}$  esetén az  $(F_n(t))$  sorozat konvergens (azaz a  $\sum(f_n(t))$  végtelen sor konvergens), akkor a  $\sum(f_n)$  függvénysor a  $t$  helyen konvergens. A

$$\mathcal{D}_0 := \{t \in \mathcal{D} : (F_n(t)) \text{ konvergens}\}$$

halmaz a  $\sum(f_n)$  függvénysor konvergencia-tartománya. Ha  $\mathcal{D}_0 \neq \emptyset$ , akkor az

$$F(t) := \lim (F_n(t)) = \sum_{n=0}^{\infty} f_n(t) \quad (t \in \mathcal{D}_0)$$

függvény a  $\sum(f_n)$  függvénysor összegfüggvénye. Az  $a \in \mathbb{K}$  középpont és az  $(a_n): \mathbb{N} \rightarrow \mathbb{K}$  együttható-sorozat segítségével tekintsük a következő függvényeket:

$$f_n(t) := a_n(t - a)^n \quad (t \in \mathcal{D} := \mathbb{K}, n \in \mathbb{N}).$$

Ekkor a  $\sum(f_n)$  függvénysor ( $a$  középpontú,  $(a_n)$  együtthatós) *hatványsor*.

**1.5.1 Tétel (Cauchy-Hadamard).** *Tegyük fel, hogy az  $(a_n): \mathbb{N} \rightarrow \mathbb{K}$  olyan sorozat, amelyre  $\exists \lim (\sqrt[n]{|a_n|})$ , és legyen*

$$r := \begin{cases} +\infty & (\lim (\sqrt[n]{|a_n|}) = 0) \\ \frac{1}{\lim (\sqrt[n]{|a_n|})} & (\lim (\sqrt[n]{|a_n|}) > 0). \end{cases}$$

*Ekkor bármely  $a \in \mathbb{K}$  mellett a  $\sum(a_n(t - a)^n)$  hatványsorra:*

1. *ha  $r > 0$ , akkor  $\forall x \in \mathbb{K}, |x - a| < r$ :  $\sum(a_n(t - a)^n)$  hatványsor az  $x$  helyen abszolút konvergens,*
2. *ha  $r < +\infty$ , akkor  $\forall x \in \mathbb{K}, |x - a| > r$ :  $\sum(a_n(t - a)^n)$  hatványsor az  $x$  helyen divergens.*

Tekintsük a  $\sum(a_n(t - a)^n)$  hatványsort. Ekkor az 1.5.1. tételben szereplő  $r$  szám a hatványsor *konvergencia-sugara*.

## 1.6. Analitikus függvények

Tegyük fel, hogy a  $\sum(a_n(t-a)^n)$  hatványsor  $r$  konvergencia-sugara nem nulla. Ekkor értelmezhetjük az

$$f(x) := \sum_{n=0}^{\infty} a_n(x-a)^n \quad (x \in K_r(a))$$

függvényt, ami nem más mint a  $\sum(a_n(t-a)^n)$  függvény-sor

$$F(x) := \sum_{n=0}^{\infty} a_n(x-a)^n \quad (x \in \mathcal{D}_0)$$

összefüggvényének a leszűkítése a  $K_r(a)$  környezetre:

$$f = F|_{K_r(a)}.$$

Az ilyen jellegű függvényt *analitikus függvénynek* nevezzük. Legyen

$$\exp x := \exp(x) := e^x := \sum_{n=0}^{\infty} \frac{x^n}{n!} \quad (x \in \mathbb{K}),$$

$$\sin x := \sin(x) := \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!} \quad (x \in \mathbb{K}),$$

$$\cos x := \cos(x) := \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!} \quad (x \in \mathbb{K}),$$

$$\operatorname{sh} x := \operatorname{sh}(x) := \sum_{n=0}^{\infty} \frac{x^{2n+1}}{(2n+1)!} \quad (x \in \mathbb{K}),$$

$$\operatorname{ch} x := \operatorname{ch}(x) := \sum_{n=0}^{\infty} \frac{x^{2n}}{(2n)!} \quad (x \in \mathbb{K})$$

az exponenciális-, a szinus-, a koszinusz-, a szinuszhiperbolikus-, és a koszinuszhiperbolikus-függvény.

## 2. Differenciál- és integrálszámítás

JACOBI-MÁTRIX, GRADIENS, PARCIÁLIS DERIVÁLT. SZÉLSŐÉRTÉK, FÜGGVÉNYVIZSGÁLAT, RIEMANN-INTEGRÁL, PARCIÁLIS INTEGRÁLÁS, INTEGRÁLÁS HELYETTESÍTÉSSEL. NEWTON-LEIBNIZ-FORMULA. A KEZDETI ÉRTÉK PROBLÉMA. LINEÁRIS, ILL. MAGASABB RENDŰ LINEÁRIS DIFFERENCIÁLEGYENLETEK.

---

### 2.1. Differenciálszámítás

**2.1.1 Definíció.** Legyen  $f \in \mathbb{R} \rightarrow \mathbb{R}$ ,  $a \in \text{int } \mathcal{D}_f$ . Ekkor  $f$  differenciálható az  $a$  pontban ( $f \in D\{a\}$ ), ha  $\exists A \in \mathbb{R}, \exists \eta \in \mathbb{R} \rightarrow \mathbb{R}$  :

$$f(x) - f(a) = A(x - a) + \eta(x)(x - a) \quad (x \in \mathcal{D}_f)$$

és  $\lim_a \eta = 0$ .

**2.1.2 Definíció (Parciális derivált).** Legyen  $f(x_1, x_2, \dots, x_n): \mathbb{R}^n \rightarrow \mathbb{R}$ .  $f$  parciálisan differenciálható  $x_1$  szerint a  $\mathcal{D}_f$  egy rögzített  $(u_2, \dots, u_n)$  pontjában, ha

$$f(x_1, u_2, \dots, u_n) \in D\{x_1\}.$$

Az előbbi parciális függvény  $x_1$ -beli deriváltját az  $f$   $x_1$  szerinti parciális deriváltjának nevezzük. Jelölése:  $\partial_1 f, \partial_2 f, \dots, \partial_n f$  vagy  $\partial_x f, \dots$

**2.1.3 Példa.**

$$\begin{aligned} f(x, y) &= x^2 - 5xy \\ \partial_x f &= 2x - 5y \\ \partial_y f &= -5x \end{aligned}$$

**2.1.4 Definíció (Gradiens).**  $f(x_1, \dots, x_n)$  gradiense adott  $(u_1, \dots, u_n)$  pontban:

$$\text{grad } f = \nabla f = (\partial_1 f, \dots, \partial_n f).$$

Azaz a gradiens a függvények deriválásának általánosítása többváltozós függvényekre, amelynek eredménye egy vektormező, ami azt mutatja, hogy változik a függvény, és megadja a skalármező legnagyobb megváltozásának irányát is.

**2.1.5 Tétel (Jacobi mátrix).**  $f: \mathbb{R}^n \rightarrow \mathbb{R}^m, a \in \text{int } \mathcal{D}_f, f \in D(a), f = \begin{pmatrix} f_1 \\ \vdots \\ f_m \end{pmatrix}$ , ahol

$f_i \in \mathbb{R}^n \rightarrow \mathbb{R}$  koordináta függvények. Ekkor

$$f'(a) = \begin{pmatrix} \delta_1 f_1(a), \delta_2 f_1(a), \dots, \delta_n f_1(a) \\ \delta_1 f_2(a), \delta_2 f_2(a), \dots, \delta_n f_2(a) \\ \vdots \\ \delta_1 f_m(a), \delta_2 f_m(a), \dots, \delta_n f_m(a) \end{pmatrix} \in \mathbb{R}^{m \times n}$$

**2.1.6 Definíció (Lokális szélsőérték).** Legyen  $f \in \mathbb{R} \rightarrow \mathbb{R}$ ,  $a \in \text{int } \mathcal{D}_f$ .  $f$ -nek az  $a$  helyen lokális maximuma (minimuma) van, ha

$$\exists k(a) : \forall x \in k(a) : f(x) \leq f(a) \quad (f(x) \geq f(a)).$$

Ha  $f$ -nek az  $a$ -ban lokális maximuma, vagy lokális minimuma van, akkor lokális szélsőértéke van  $a$ -ban.

**2.1.7 Tétel (Elsőrendű szükséges feltétel).** Legyen  $f \in \mathbb{R} \rightarrow \mathbb{R}$ ,  $a \in \text{int } \mathcal{D}_f$ ,  $f \in D\{a\}$ . Ha  $f$ -nek lokális szélsőértéke van  $a$ -ban, akkor  $f'(a) = 0$

**2.1.8 Tétel (Elsőrendű elégséges feltétel).** Tegyük fel, hogy  $f \in \mathbb{R} \rightarrow \mathbb{R}$ ,  $a \in \text{int } \mathcal{D}_f$ ,  $f \in D$  és  $f$ -nek  $a$ -ban lokális szélsőértéke van. Ekkor, ha

1. az  $f'$  deriváltfüggvénynek az  $a$ -ban  $(+, -)$  jelváltása van, akkor az  $f$ -nek az  $a$ -ban lokális maximuma van,
2. az  $f'$  deriváltfüggvénynek az  $a$ -ban  $(-, +)$  jelváltása van, akkor az  $f$ -nek az  $a$ -ban lokális minimuma van.

**2.1.9 Definíció (Globális szélsőérték).** Legyen  $f \in \mathbb{R} \rightarrow \mathbb{R}$ ,  $a \in \text{int } \mathcal{D}_f$ .  $f$ -nek  $a$ -ban globális maximuma (minimuma) van, ha

$$\forall x \in \mathcal{D}_f : f(x) \leq f(a) \quad (f(x) \geq f(a)).$$

## 2.2. Függvényvizsgálat

### 2.2.1. Előzetes vizsgálatok

- Páros vagy páratlan? ( $f(x) = f(-x)$  vagy  $-f(x) = f(-x)$ )
- Periodikus-e? ( $\exists p > 0 : \forall x \in \mathcal{D}_f : x \pm p \in \mathcal{D}_f$  és  $f(x \pm p) = f(x)$ ).

### 2.2.2. Folytonosság

- Folytonos-e a függvény? ( $\forall \varepsilon > 0 \exists \delta > 0 : |f(x) - f(a)| < \varepsilon \quad (|x - a| < \delta)$ )
- Van-e szakadása? ( $\exists a \in \mathcal{D}_f : f \notin \mathcal{C}\{a\}$ )

### 2.2.3. Monotonitás

**2.2.4 Tétel.** Tegyük fel, hogy  $f \in \mathbb{R} \rightarrow \mathbb{R}$ ,  $\mathcal{D}_f$  nyílt intervallum,  $f \in D$ . Ekkor

- $f$  monoton növő  $\Leftrightarrow f' \geq 0$ ,
- $f$  monoton fogyó  $\Leftrightarrow f' \leq 0$ .

**2.2.5 Tétel.** Tegyük fel, hogy  $f \in \mathbb{R} \rightarrow \mathbb{R}$ ,  $\mathcal{D}_f$  nyílt intervallum,  $f \in D$ . Ekkor

- $f$  szigorúan monoton növő  $\Leftrightarrow f'(x) \geq 0$  ( $x \in \mathcal{D}_f$ ),  $\forall I \subset \mathcal{D}_f$  nyílt interallumra:

$$\exists t \in I : f'(t) > 0,$$

- $f$  szigorúan monoton fogyó  $\Leftrightarrow f'(x) \leq 0$  ( $x \in \mathcal{D}_f$ ),  $\forall I \subset \mathcal{D}_f$  nyílt interallumra:

$$\exists t \in I : f'(t) < 0.$$

### 2.2.6. Szélsőérték

A tétel korábbi részében szó esik róla.

### 2.2.7. Konvexitás

**2.2.8 Definíció.** Legyen  $I \subset \mathbb{R}$  intervallum,  $f: I \rightarrow \mathbb{R}$ . Ekkor az  $f$  függvény konvex, ha

$$\forall a, b \in I, \forall \lambda \in [0, 1] : f(\lambda a + (1 - \lambda)b) \leq \lambda f(a) + (1 - \lambda)f(b).$$

konkáv, ha

$$\forall a, b \in I \forall \lambda \in [0, 1] : f(\lambda a + (1 - \lambda)b) \geq \lambda f(a) + (1 - \lambda)f(b).$$

**2.2.9 Tétel.** Tegyük fel, hogy az  $f \in \mathbb{R} \rightarrow \mathbb{R}$ ,  $f \in D$ . Ekkor

- $f$  konvex  $\Leftrightarrow f'$  monoton nő,
- $f$  konkáv  $\Leftrightarrow f'$  monoton fogy.

Ha  $f \in D^2$ , akkor

- $f$  konvex  $\Leftrightarrow f'' \geq 0$ ,
- $f$  konkáv  $\Leftrightarrow f'' \leq 0$ .

**2.2.10 Definíció (Inflexiós pont).** Legyen  $f \in \mathbb{R} \rightarrow \mathbb{R}$ ,  $a \in \text{int } \mathcal{D}_f$ ,  $f \in D\{a\}$ . Ha  $f - e_a f$  függvény jelet vált  $a$ -ban, ahol  $e_a f$  az  $f$  függvény  $a$ -beli érintője, akkor az  $f$  függvénynek az  $a$  helyen inflexiója van.

## 2.3. Integrálszámítás

**2.3.1 Definíció (Primitív függvény).** Legyen  $I \subset \mathbb{R}$  nyílt intervallum,  $f: I \rightarrow \mathbb{R}$ . Azt mondjuk, hogy az  $F: I \rightarrow \mathbb{R}$  függvény primitív függvénye  $f$ -nek, ha  $F \in D$ , és  $F' = f$ .

**2.3.2 Definíció (Határozatlan integrál).** Legyen  $I \subset \mathbb{R}$  nyílt intervallum,  $f: I \rightarrow \mathbb{R}$ ,  $F: I \rightarrow \mathbb{R}$ ,  $F \in D$ ,  $F' = f$ . Ekkor legyen

$$\int f := \int f(x)dx := \{F: I \rightarrow \mathbb{R} : F \in D, F' = f\}$$

az  $f$  határozatlan integrálja.

**2.3.3 Tétel (Parciális integrálás).** Tegyük fel, hogy  $I \subset \mathbb{R}$  nyílt intervallum,  $f, g: I \rightarrow \mathbb{R}$ ,  $f, g \in D$ ,  $\int f'g \neq \emptyset$ . Ekkor  $\int fg' \neq \emptyset$ , és

$$\forall F \in \int f'g : \int fg' = \{fg - F + c : c \in \mathbb{R}\}.$$

### 2.3.4. Riemann-integrál

A Riemann-integrál célja az integrálandó ( $f(x)$ ) függvény-görbe alatti területének meghatározása egy adott  $[a, b]$  intervallumon, ahol  $a$  az integrál alsó határa,  $b$  az integrál felső határa. Ehhez az  $[a, b]$ -t részintervallumokra kell osztani:

$$\tau \subset [a, b], a, b \in \tau \text{ és } \tau = \{x_0, x_1, \dots, x_n\} \ (n \in \mathbb{N}), x_0 = a < x_1 < \dots < x_n = b.$$

Legyen

$$m_i = m_i(f) := \inf\{f(x) : x_i \leq x \leq x_{i+1}\} \ (i = 0, \dots, n-1),$$

$$M_i = M_i(f) := \sup\{f(x) : x_i \leq x \leq x_{i+1}\} \ (i = 0, \dots, n-1).$$

Ekkor

$$s(f, \tau) := \sum_{i=0}^{n-1} m_i(x_{i+1} - x_i)$$

az  $f$  függvénynek a  $\tau$  felosztáshoz tartozó alsó összege,

$$S(f, \tau) := \sum_{i=0}^{n-1} M_i(x_{i+1} - x_i)$$

az  $f$  függvények a  $\tau$  felosztáshoz tartozó felső összege.

A Darboux-féle alsó integrál a  $\tau$  felosztáshoz tartozó alsó összeg szuprémuma:

$$I_* = \sup s(f, \tau),$$

A Darboux-féle felső integrál a  $\tau$  felosztáshoz tartozó felső összeg infimuma:

$$I^* = \inf S(f, \tau).$$

**2.3.5 Definíció (Riemann-integrál).** Legyen  $a, b \in \mathbb{R}$ ,  $a < b$ ,  $[a, b]$  korlátos, zárt intervallum,  $f: [a, b] \rightarrow \mathbb{R}$  korlátos függvény. Az  $f$  függvény Riemann-integrálható, ha  $I_*(f) = I^*(f)$ . Ekkor az

$$\int_a^b f := \int_a^b f(x)dx := I_*(f) = I^*(f)$$

szám az  $f$  függvény Riemann-integrálja.

**2.3.6 Definíció (Integrálás helyettesítéssel).** Legyen  $I \subset \mathbb{R}$ ,  $g: [a, b] \rightarrow I$ ,  $f: I \rightarrow \mathbb{R}$ ,  $f \in C$ . Ekkor

$$\int_{g(a)}^{g(b)} f(x)dx = \int_a^b f(g(t))g'(t)dt.$$

**2.3.7 Tétel (Newton-Leibniz-formula).** Tegyük fel, hogy  $[a, b]$  korlátos zárt intervallum,  $f: [a, b] \rightarrow \mathbb{R}$ ,  $f \in R[a, b]$ ,  $\exists F: [a, b] \rightarrow \mathbb{R} : F \in D, F' = f$ . Ekkor

$$\forall F : \int_a^b f = F(b) - F(a).$$

## 2.4. Differenciálegyenlet

A differenciálegyenlet olyan egyenlet, amelyben az ismeretlen kifejezés egy differenciálható függvény ( $f$ ), és az egyenlet az  $f$  és  $f'$  függvény között teremt kapcsolatot.

**2.4.1 Definíció (Differenciálegyenlet).**  $D \subset \mathbb{R}^{n+1}$  tartomány,  $f: D \rightarrow \mathbb{R}^n$  folytonos. Az  $x'(t) = f(t, x(t))$  egyenletet elsőrendű explicit differenciálegyenletnek nevezünk, ahol  $x: I \rightarrow \mathbb{R}^n$  folytonosan differenciálható,  $I \subset \mathbb{R}$  nyílt intervallum, és  $(t, x(t)) \in D \quad \forall t \in I$ .

**2.4.2 Definíció (Kezdeti érték probléma).**  $D \subset \mathbb{R}^{n+1}$  tartomány,  $f: D \rightarrow \mathbb{R}^n$  folytonos,  $(\tau, \xi) \in D, \tau \in \mathbb{R}, \xi \in \mathbb{R}^n$ . Az  $x'(t) = f(t, x(t)) \quad x(\tau) = \xi$  feladatot kezdeti érték problémának nevezünk.

**2.4.3 Definíció (Elsőrendű lineáris differenciálegyenlet).**  $I \subset \mathbb{R}$  nyílt intervallum,  $f, g: I \rightarrow \mathbb{R}$  folytonos. Az  $x' + fx = g$  differenciálegyenletet elsőrendű lineáris differenciálegyenletnek nevezünk.

**2.4.4 Definíció (Magasabb rendű differenciálegyenlet).**  $D \subset \mathbb{R}^{n+1}$  tartomány,  $h: D \rightarrow \mathbb{R}$  folytonos. Az

$$y^{(n)}(t) = h(t, y(t), y'(t), \dots, y^{(n-1)}(t))$$

feladatot  $n$ -ed rendű differenciálegyenletnek nevezünk.

Jel:  $y^{(n)} = h \circ (\text{id}, y, y', \dots, y^{(n-1)})$ .

### 3. Numerikus módszerek

ITERÁCIÓS MÓDSZEREK: LINEÁRIS EGYENLETRENDSZEREKRE ÉS NEMLINEÁRIS EGYENLETEKRE. INTERPOLÁCIÓ: LAGRANGE,-HERMITE-SPLINE INTERPOLÁCIÓ. LEGKISEBB NÉGYZETEK MÓDSZERE



## 4. Számelmélet, gráfok, kódoláselmélet

RELÁCIÓK, RENDEZÉSEK. FÜGGVÉNYEK ÉS MŰVELETEK. SZÁMFOGALOM, KOMPLEX SZÁMOK. LESZÁMLÁLÁSOK VÉGES HALMAZOKON. SZÁMELMÉLETI ALAPFOGALMAK, LINEÁRIS KONGURENCIA-EGYENLETEK. ÁLTALÁNOS ÉS SÍKGRÁFOK, FÁK, EULER- ÉS HAMILTON-GRÁFOK, GRÁFOK ADATSZERKEZETEI. POLINOMOK ÉS MŰVELETEIK, MARADÉKOS OSZTÁS, HORNER-SÉMA. BETŰNKÉNTI KÓDOLÁS, SHANNON- ÉS HUFFMANKÓD. HIBAJAVÍTÓ KÓDOK, KÓDTÁVOLSÁG. LINEÁRIS KÓDOK.

---

### 4.1. Relációk, rendezések

**4.1.1 Definíció.**  $A, B$  tetszőleges halmaz, az  $A \times B = \{(a, b) \mid a \in A \wedge b \in B\}$  az  $A$  és  $B$  halmaz *Descartes-szorzata*. Az  $R \subseteq A \times B$  halmaz *binér reláció*. Ha  $A = B$ , akkor  $R$  egy  $A$ -beli binér reláció (homogén binér reláció).

**4.1.2 Definíció.** Legyen  $R$  egy  $X$ -beli binér reláció. Az  $R$

1. tranzitív, ha  $\forall x, y, z: ((x, y) \in R \wedge (y, z) \in R) \Rightarrow (x, z) \in R$ .
2. szimmetrikus, ha  $\forall x, y: (x, y) \in R \Rightarrow (y, x) \in R$ .
3. antiszimmetrikus, ha  $\forall x, y: ((x, y) \in R \wedge (y, x) \in R) \Rightarrow x = y$ .
4. szigorúan antiszimmetrikus, ha  $\forall x, y: (x, y) \in R \Rightarrow (y, x) \notin R$ .
5. reflexív, ha  $\forall x \in X: (x, x) \in R$ .
6. irreflexív, ha  $\forall x \in X: (x, x) \notin R$ .
7. trichotom, ha  $\forall x, y \in X$  esetén  $x = y, (x, y) \in R, (y, x) \in R$  közül pontosan egy teljesül.
8. dichotom, ha  $\forall x, y \in X$  esetén  $(x, y) \in R$  vagy  $(y, x) \in R$  (esetleg mindkettő).

Az első négy tulajdonság csak  $R$ -től, míg az utolsó négy  $X$ -től is függ.

**4.1.3 Definíció.** Legyen  $X$  egy halmaz. Az  $X$ -beli binér reláció

- Ekvivalenciareláció, ha tranzitív, szimmetrikus, reflexív,
- Részbenrendezés, ha tranzitív, antiszimmetrikus, reflexív,
- Teljes rendezés, ha tranzitív, antiszimmetrikus, reflexív, dichotom.

### 4.2. Függvények és műveletek

**4.2.1 Definíció.** Az  $f$  reláció függvény, ha  $(x, y) \in f \wedge (x, y') \in f \Rightarrow y = y'$ .

**4.2.2 Állítás.** Az  $f: X \rightarrow Y$  függvény injektív, ha

$$(x, y) \in f \wedge (x', y) \in f \Rightarrow x = x'.$$

Az  $f$  függvény szürjektív, ha egész  $Y$ -ra képez, ha pedig injektív és szürjektív is, akkor bijektívnek nevezzük.

**4.2.3 Definíció.** Legyen  $X$  egy halmaz. Egy  $X$ -beli binér művelet egy  $*$ :  $X \times X \rightarrow X$  leképezés,  $X$ -beli unér művelet egy  $*$ :  $X \rightarrow X$  leképezés, nullér művelet pedig  $*$ :  $\{\emptyset\} \rightarrow X$  leképezés, ami  $X$  egy elemének a kijelölését jelenti.

**4.2.4 Definíció.** Legyen  $*$  binér művelet az  $X$ , és  $'$  binér művelet az  $X'$  halmazon. Egy  $\varphi: X \rightarrow X'$  leképezés művelettartó, ha

$$\forall x, y \in X: \varphi(x * y) = \varphi(x) *' \varphi(y).$$

### 4.3. Számfogalom, komplex számok

**4.3.1 Definíció (Semleges elem).** Legyen  $G$  tetszőleges halmaz,  $*$  tetszőleges művelet a  $G$ -n.  $G$  egy  $s$  eleme bal, illetve jobb oldali semleges elem, ha  $s * g = g$  és  $g * s = g$ . Ha egy elem bal és jobb oldali semleges elem is, akkor semleges elem.

**4.3.2 Definíció (Inverz).** Legyen  $G$  tetszőleges halmaz,  $*$  tetszőleges művelet a  $G$ -n.  $s$  semleges eleme  $G$ -nek, és  $g, g^* \in G$ . Ha  $g * g^* = s$  akkor  $g$  a  $g^*$  balinverze,  $g^*$  pedig a  $g$  jobbinverze. Ha  $g^*$  a  $g$  bal és jobbinverze is, akkor a  $g$  inverze.

**4.3.3 Definíció.**  $G$  halmaz,  $*$ :  $G \times G \rightarrow G$ . Ekkor a  $(G, *)$  grupoid. Ha

1.  $\forall x, y, z \in G: (x * y) * z = x * (y * z)$ , akkor  $G$  félcsoport,
2.  $G$  félcsoport, és létezik semleges eleme, akkor  $G$  monoid,
3.  $G$  monoid, és  $\forall g \in G: \exists g^*$  (inverz), akkor  $G$  csoport,
4.  $G$  csoport, és a  $*$  kommutatív, akkor  $G$  Abel-csoport.

**4.3.4 Definíció.**  $(G, +, *)$  gyűrű, ha  $(G, +)$  Abel-csoport,  $(G, *)$  félcsoport, és teljesül mindkét oldali disztributivitás.

**4.3.5 Definíció.** A kommutatív, nullosztómentes (azaz nincs két olyan nullától különböző eleme, amelynek szorzata 0-t adna) gyűrűt *integritási tartomány*-nak nevezzük.

Ha  $R$  rendezett, integritási tartomány, és

$$\begin{aligned} x, y, z \in R \text{ és } x \leq y &\Rightarrow x + z \leq y + z \\ x, y \in R \text{ és } x, y \geq 0 &\Rightarrow x \cdot y \geq 0, \end{aligned}$$

akkor *rendezett integritási tartomány*-nak nevezzük.

**4.3.6 Definíció.** A  $G$  gyűrű *test*, ha  $(G \setminus \{0\}, *)$  Abel-csoport.

**4.3.7 Definíció.** A valós számok halmaza egy felsőhatár tulajdonságú rendezett test.

**4.3.8 Definíció.** A komplex számok halmaza  $\mathbb{C} = \mathbb{R} \times \mathbb{R}$ , azaz a valós számpárok halmaza az

$$\begin{aligned} (x, y) + (x', y') &= (x + x', y + y') \\ (x, y) \cdot (x', y') &= (xx' - y'y, y'x + yy') \end{aligned}$$

műveletekkel.

Az  $(x, y)$  komplex szám algebrai alakja:  $x + iy$ . A  $z = |z|(\cos t + i \cdot \sin t)$  pedig a  $z$  komplex szám trigonometrikus alakja, ahol  $t \in \mathbb{R}$ .

**4.3.9 Tétel (Algebra alaptétele).** Ha  $n \in \mathbb{N}^+$ , valamint  $c_0, c_1, \dots, c_n$  komplex számok,  $c_n \neq 0$ , akkor

$$\exists z \in \mathbb{C}: \sum_{k=0}^n c_k z^k = 0.$$

## 4.4. Leszámlálások véges halmazokon

**4.4.1 Definíció.** Egy  $X$  halmaz önmagára való kölcsönösen egyértelmű leképezései az  $X$  *permutációi*. A permutációk száma  $n = |X|$ -től függ.

$$P_n = \prod_{k=1}^n k.$$

**4.4.2 Definíció.** Az  $A$  halmaz elemeiből készíthető, különböző tagokból álló  $a_1, a_2, \dots, a_k$  sorozatokat, azaz  $\{1, 2, \dots, k\}$   $A$ -ba képező kölcsönösen egyértelmű leképezések az  $A$   $k$ -ad osztályú *variációi*. ha  $A$  véges,  $\#A = n$ , akkor a variációk száma:

$$V_n^k = \frac{n!}{(n-k)!}.$$

Ha az  $a_i$  tagok nem feltétlen különböznek, akkor *ismétléses variáció*-ról beszélünk, melyek száma:

$${}^iV_n^k = n^k.$$

**4.4.3 Definíció.** Az  $A$  halmaz  $k \in \mathbb{N}$  elemű részhalmazai az  $A$   $k$ -ad osztályú *kombinációi*. Ha  $A$  véges,  $\#A = n$ , akkor a számuk:

$$C_n^k = \frac{n!}{k!(n-k)!} = \binom{n}{k}.$$

Az  $f: A \rightarrow \mathbb{N}: \sum_{a \in A} f(a) = k$  függvények az  $A$   $k$ -ad osztályú *ismétléses kombinációi*. Ezek száma  $A$  elemeinek számától függ. Legyen  $\#A = n$ , ekkor:

$${}^iC_n^k = \binom{n+k-1}{k}.$$

**4.4.4 Definíció.** Az  $a_1, a_2, \dots, a_r$  különböző elemek  $i_1, i_2, \dots, i_r$  ismétlődésű *ismétléses permutációi* az olyan  $n = i_1 + i_2 + \dots + i_r$  tagú sorozatok, amelyekben az  $a_j$  elem  $i_j$ -szer fordul elő. Számuk

$$P_n^{i_1, \dots, i_r} = \frac{n!}{i_1! i_2! \dots i_r!}.$$

**4.4.5 Tétel (Binomiális tétel).** Legyen  $R$  kommutatív, egységelemes gyűrű,  $x, y \in R$ ,  $n \in \mathbb{N}$ . Ekkor

$$(x + y)^n = \sum_{k=0}^n \binom{n}{k} x^k y^{n-k}.$$

## 4.5. Számelméleti alapfogalmak, lineáris kongruencia-egyenletek

**4.5.1 Definíció.** Az  $m \in \mathbb{N}$  az  $n \in \mathbb{N}$  *osztó*-ja, az  $n$  pedig az  $m$  többszöröse, ha

$$\exists k \in \mathbb{N}: n = mk.$$

**4.5.2 Definíció.**  $1 < n \in \mathbb{N}$  csak  $1 \cdot n = n \cdot 1$  alakban írható fel, akkor *irreducibilis elem*.

**4.5.3 Definíció.**  $1 < p \in \mathbb{N}$  szám *prím*, ha  $p|ab$  esetén  $p|a \vee p|b$ .

**4.5.4 Definíció.**  $R$  egységelemes integritási tartomány,  $a, b \in R$ ,  $a$  és  $b$  asszociáltak ha

$$a|b \wedge b|a.$$

**4.5.5 Definíció.** Egy elem asszociáltjaitól különböző osztói az elem *valódi osztói*.

**4.5.6 Definíció.**  $R$  egységelemes integritási tartomány,  $a_1, a_2, \dots, a_n \in R$  elemek *legnagyobb közös osztója* a  $b \in R$ , ha  $i = 1, 2, \dots, n$  esetén  $b|a_i$ , és ha  $i = 1, 2, \dots, n$  esetén  $b'|a_i$  akkor  $b'|b$ .

**4.5.7 Definíció.**  $R$  egységelemes integritási tartomány,  $a_1, a_2, \dots, a_n \in R$  elemek *legkisebb közös többszöröse* a  $b \in R$ , ha  $i = 1, 2, \dots, n$  esetén  $a_i|b$ , és ha  $i = 1, 2, \dots, n$  esetén  $a_i|b'$  akkor  $b|b'$ .

**4.5.8 Definíció.**  $a, b, m \in \mathbb{Z}$ ,  $a$  konguens  $b$  modulo  $m$ , ha  $m|a - b$ .

Jelölése:  $a \equiv b \pmod{m}$ .

**4.5.9 Állítás.**  $1 < m, a, b \in \mathbb{Z}$  adott. Keressük az  $ax \equiv b \pmod{m}$  megoldásait. Megoldhatóság: kínai maradéktétel.

## 4.6. Általános és síkgráfok, Euler- és Hamilton gráfok, gráfok adatszerkezetei

**4.6.1 Definíció.** Az *Irányítatlan gráf* egy  $G = (\varphi, E, V)$  hármas, ahol  $V$  a csúcsok halmaza,  $E$  az élek halmaza,  $\varphi$  pedig egy  $E$ -t a  $V$ -beli elemekből álló rendezetlen párok halmazába képező leképezés. Ha valamely  $e \in E, v \in V: v \in \varphi(e)$ , akkor  $e$  illeszkedik  $v$ -re.

**4.6.2 Definíció.** Egy gráf *véges*, ha  $E$  és  $V$  véges halmazok.  $d(v)$  a  $v$  csúcs fokszáma, vagyis a rá illeszkedő élek száma, a csúcsra illeszkedő hurokéleket kétszer számolva. Véges gráf esetén

$$\sum_{v \in V} d(v) = 2|E|.$$

**4.6.3 Definíció.** Ha a gráfban minden csúcs fokszáma  $n$ , akkor  *$n$ -reguláris gráfnak* hívjuk. Például a Petersen-gráf.

**4.6.4 Definíció.** A  $G = (\varphi, E, V)$  és a  $G' = (\varphi', E', V')$  gráfok *izomorfak*, ha

$$\exists f: E \rightarrow E' \wedge g: V \rightarrow V'$$

bijektív leképezés, hogy

$$\forall e \in E, v \in V: v \in \varphi(e) \Leftrightarrow g(v) \in \varphi'(f(e)).$$

**4.6.5 Definíció.** Legyen  $G = (\varphi, E, V)$  gráf, Egy  $G$ -beli  $n$  hosszú séta  $v$ -ből  $v'$ -be egy

$$v_0, e_1, v_1, e_2, \dots, v_{n-1}, e_n, v_n$$

véges sorozat, amelyre  $e_i$  a  $v_{i-1}$  és  $v_i$  csúcsra illeszkedő él, és  $v_0 = v, v_n = v'$ .

Ha  $v = v'$  akkor a séta *zárt*, különben *nyílt*.

Ha a sétában szereplő élek mind különbözőek, akkor a séta *vonat*.

Ha a vonat zárt séta, akkor *zárt vonat*, különben *nyílt vonat*.

Egy séta *út*, ha  $v_0, v_1, \dots, v_n$  csúcsok különbözőek.

A legalább egy hosszú zárt vonalat *körnek* nevezzük, ha kezdő és végpontjai megegyeznek, de egyébként a vonat pontjai különbözőek.

**4.6.6 Definíció.** Egy gráf összefüggő, ha bármely két csúcsa összeköthető sétával.

**4.6.7 Definíció.** Egy gráf *fa*, ha összefüggő és nincs köre.

**4.6.8 Definíció.** Egy gráf *erdő*, ha körmentes.

**4.6.9 Definíció.** Az *Euler-vonal* egy  $v$ -ből  $v'$ -be vezető vonal, amelyben a gráf minden éle szerepel.

**4.6.10 Állítás.** Egy összefüggő véges gráfban pontosan akkor létezik zárt Euler-vonal, ha minden csúcs páros fokú. Ha véges összefüggő gráf  $2s$  páratlan fokú csúcsot tartalmaz, ahol  $s \in \mathbb{N}^+$ , akkor a gráf  $s$  darab páronként éldiszjunkt nyílt vonal egyesítése.

**4.6.11 Definíció.** A *Hamilton-út* egy olyan  $v$ -ből  $v'$ -be vezető út amelyben a gráf minden pontja pontosan egyszer szerepel. A *Hamilton-kör* olyan kör, amelyben a gráf minden csúcsa szerepel.

**4.6.12 Definíció.** A  $G = (\varphi, E, V, c_e, C_e, c_v, C_v)$  hetes címkézett gráf, ahol  $C_e, C_v$  az élcímkék és csúcscímkék halmaza, a  $c_e: E \rightarrow C_e$ , és  $c_v: V \rightarrow C_v$  leképezések az élcímkézés, illetve csúcscímkézés.

**4.6.13 Definíció.** A  $G = (\psi, E, V)$  hármas irányított gráf, ahol  $E$  az élek halmaza,  $V$  a csúcsok halmaza,  $\psi: E \rightarrow V \times V$  leképezés. A csúcs kifoka a belőle induló élek száma, befoka pedig azon élek száma, amelyeknek az adott csúcs a végpontja.

## 4.7. Polinomok és műveleteik, maradékos osztás, Horner-séma

**4.7.1 Definíció.**  $R$  gyűrű. A

$$\sum_{i=0}^n f_i x^i, \quad n \in \mathbb{N}, f_i \in R$$

véges összegeket *egyhatározatlanú polinom*nak nevezzük.

Az  $f = (f_0, f_1, \dots)$  és  $g = (g_0, g_1, \dots)$   $R$ -beli végtelen sorozatok. Ekkor az

$$f + g = (f_0 + g_0, f_1 + g_1, \dots)$$

a két sorozat összege, és az

$$f \cdot g = h = (h_0, h_1, \dots) \text{ ahol } h_k = \sum_{i+j=k} f_i g_j$$

a két sorozat szorzata. Így gyűrűt kapunk.

Tehát  $R$  feletti egyhatározatlanú polinomokon olyan  $R$ -beli  $f = (f_0, f_1, \dots)$  sorozatokat értünk, amelyeknek csak véges sok tagja nem nulla. Az

$$f = f_0 + f_1 x^1 + f_2 x^2 + \dots + f_n x^n \in R[x] \quad (f_n \neq 0)$$

alakú polinom főegyütthatója  $f_n$ , foka:  $n$ , ahol  $R[x]$  az  $R$  feletti egyhatározatlanú polinomok gyűrűje.

**4.7.2 Tétel (Maradékos osztás).**  $R$  egységelemes integritási tartomány,  $f, g \in R[x]$ ,  $g \neq 0$ , és tegyük fel, hogy  $g$  főegyütthatója egység  $R$ -ben. Ekkor

$$\exists! q, r \in R[x]: f = gq + r \quad (\deg(r) < \deg(g)).$$

**4.7.3 Megjegyzés (Horner-elrendezés).** A maradékos osztás tételét alkalmazva az  $f$  és a  $g = x - c$  polinomra azt kapjuk, hogy  $f = (x - c)q + r$ , ahol  $r$  konstans, értéke  $f(c)$ . Így  $n - 1$  szorzással és ugyanannyi összeadással megkaphatjuk  $f(c)$ -t.

## 4.8. Betűnkénti kódolás, Shannon- és Huffman kód. Hibajavító kódok, kódtávolság, lineáris kódok

A kódolás a legáltalánosabb értelemben az üzenetek halmazának egy másik halmazba való leképezését jelenti.

**4.8.1 Definíció (Betűnkénti kódolás).**  $A, B$  ábécé,  $\varphi: A \rightarrow B^*$  leképezés, amelyet természetes módon terjesztünk ki  $\psi: A^* \rightarrow B^*$  leképezéssé. Ha  $a_1 a_2 \dots a_n = \alpha \in A^*$ , akkor  $\alpha$  kódja  $\psi(\alpha) = \varphi(a_1)\varphi(a_2) \dots \varphi(a_n)$ . Ha  $\varphi$  nem injektív, akkor a kód nem felbontható.

**4.8.2 Definíció.** Legyen  $A = \{a_1, \dots, a_n\}$  a kódolandó ábécé,  $p_1, \dots, p_n$  a betűk eloszlása,  $\varphi: A \rightarrow B^+$  egy betűnkénti kódolás,  $l_i$  az  $a_i$  kódjának hossza. Ekkor a kód *átlagos szóhosszúsága*:

$$\bar{l} = \sum_{i=1}^n p_i l_i.$$

Ha adott elemszámú ábécével és eloszlással egy felbontható betűnkénti kód átlagos szóhosszúsága minimális, akkor *optimális kódnak* nevezzük.

**4.8.3 Tétel (Shannon tétele zajmentes csatornára).** Legyen  $B$  elemeinek száma  $r$ . Ha a betűnkénti kódolás felbontható, akkor

$$H_r(p_1, \dots, p_n) \leq \bar{l},$$

ahol  $H_r$  az eloszlás entrópiája.

**4.8.4 Tétel (Shannon-kód létezése).**  $n > 1$  esetén van olyan prefix kód, amelyre

$$\bar{l} < H_r(p_1, \dots, p_n) + 1.$$

**4.8.5 Tétel (Huffman-kód).** A Huffman-kód *optimális kódot ad*. Ehhez rendezzük a kódolandó ábécé elemeit relatív gyakoriságuk sorrendjében, majd osszuk el  $n - 2t - r - 1$ -el, és legyen  $m$  a maradék  $+2$ . Első lépésben az utolsó  $m$  elemet vonjuk össze, az összevont elem gyakorisága legyen az összevont elemek gyakoriságának összege. Rendezzük újra a redukált ábécét, és ismételjük a lépést, de az utolsó  $r$  elemet vonjuk össze. Ismételjük ezt addig amíg már csak  $r$  elem van. Ezeket a kódoló ábécé elemeivel kódoljuk, majd a redukciónak megfelelően visszafelé haladva, az ott összevont betűk kódját az összevonásként kapott betű már meglévő kódjának a kódoló ábécé különböző betűivel való kiegészítésével kapjuk.

**4.8.6 Definíció.**  $u, v$  két egyforma hosszú szó Hamming-távolsága ( $d(u, v)$ ) az azonos pozícióban lévő különböző jegyek száma, és a kód távolsága ( $d(C)$ ) a különböző kódszópárok távolságainak minimuma, ahol  $C$  a kódszavak halmaza.

**4.8.7 Definíció.** Egy kód *t-hibajavító*, ha minden esetben helyesen javít, amikor egy elküldött kódszó legfeljebb  $t$  helyen változik meg. A kód pontosan *t-hibajavító*, ha *t-hibajavító*, de nem *t + 1-hibajavító*.

**4.8.8 Definíció (Lineáris kód).** Ha  $K$  véges test, akkor a  $K$  elemeiből alkotott rendezett  $n$ -esek a komponensenkénti összeadással, valamint az  $n$ -es minden elemének ugyanazzal az elemmel való szorzásával egy  $K$  feletti  $n$ -dimenziós  $K^n$  lineáris teret alkotnak. Ennek bármely altere lineáris kód.

Például: CRC (hibajelzés), Hamming-kód (egyetlen hiba javítása).

## 5. Valószínűségszámítási és statisztikai alapok

DISZKRÉT ÉS FOLYTONOS VALÓSZÍNŰSÉGI VÁLTOZOK, NAGY SZÁMOK TÖRVÉNYE, CENTRÁLIS HATÁRELOSZLÁS TÉTEL. STATISZTIKAI BECSLÉSEK, KLASSZIKUS STATISZTIKAI PRÓBÁK

---

### 5.1. Diszkrét és folytonos valószínűségi változók

**5.1.1 Definíció (Valószínűségi változó).** A valószínűségi változó egy eseménytér elemeihez valós számokat rendelő  $X: Q \rightarrow \mathbb{R}$  függvény.

**5.1.2 Definíció (Diszkrét valószínűségi változó).** Az  $X$  valószínűségi változó diszkrét, ha lehetséges értékeinek száma véges.  $X(Q)$ -val jelöljük.

**5.1.3 Definíció (Diszkrét valószínűségeloszlás).** Az  $X$  változó eloszlása a

$$p: X(Q) \rightarrow \mathbb{R}, p_i = P(X = x_i)$$

függvény.

**5.1.4 Tétel (Tulajdonságai).** Az  $X$  változó diszkrét valószínűségeloszlására igaz, hogy

- $p_i \geq 0 \quad (\forall i \in [1 \dots n])$ ,
- $\sum_{i=1}^n p_i = 1$ .

**5.1.5 Definíció (Diszkrét eloszlásfüggvény).** Az  $X$  változó diszkrét eloszlásfüggvénye az

$$F(x) = \sum_{x_i < x} p_i$$

függvény.

**5.1.6 Definíció (Várható érték).**  $X$  valószínűségi változó várható értéke

$$E(X) = \sum_{i=1}^n x_i p_i.$$

**5.1.7 Definíció (Szórásnégyzet, szórás).** Az  $X$  valószínűségi változó szórásnégyzete az  $(x - m)^2$  változó várható értéke:

$$D^2(X) = \sum_{i=1}^n (x_i - m)^2 p_i \text{ ahol } m = E(X).$$

A szórásnégyzet gyöke a szórás ( $D(X)$ ).

**5.1.8 Tétel (Várható érték és szórásnégyzet tulajdonságai).**  $X, Y$  valószínűségi változók,  $a, b \in \mathbb{R}$ . Ekkor

- $E(aX + b) = aE(X) + b$ ,
- $E(aX + bY) = aE(X) + bE(Y)$ ,
- $D^2(aX + b) = a^2 D^2(X)$ .

**5.1.9 Tétel (Tulajdonságok független valószínűségi változókkal).**  $X, Y$  független valószínűségi változók. Ekkor

- $E(X \cdot Y) = E(X) \cdot E(Y)$ ,
- $D^2(X + Y) = D^2(X) + D^2(Y)$ .

**5.1.10 Definíció (Együttes eloszlás).** Legyen  $X, Y$  valószínűségi változó. Ekkor a

$$p: X(Q) \times Y(Q) \rightarrow \mathbb{R}, p_{ij} = P(X = x_i, Y = y_i)$$

függvény az  $X$  és  $Y$  együttes eloszlása.

**5.1.11 Definíció (Együttes várható érték).** Az  $X, Y$  valószínűségi változók együttes várható értéke

$$E(X, Y) = \sum_{i,j} x_i y_i p_{ij}.$$

**5.1.12 Definíció (Peremeloszlás).** Valaki mondja meg mi a faszom ez! Valaki mondja meg miért kell ez, valaki mondja meg miért nem!

**5.1.13 Definíció (Kovariancia).**  $X, Y$  valószínűségi változók kovarianciája az

$$\text{Cov}(X, Y) = E((X - E(X))(Y - E(Y)))$$

várható érték.

**5.1.14 Tétel.**  $\text{Cov}(X, Y) = E(XY) - E(X)E(Y)$ .

**5.1.15 Állítás.** Legyenek  $X, Y, X_1, \dots, X_n$  valószínűségi változók,  $a, b \in \mathbb{R}$ . Ekkor

- $\text{Cov}(X, X) = D^2(X)$ ,
- $\text{Cov}(X, Y) = \text{Cov}(Y, X)$ ,
- $\text{Cov}(X, a) = 0$ ,
- $\text{Cov}(aX, bY) = ab \text{Cov}(X, Y)$ ,
- $D^2(X + Y) = D^2(X) + D^2(Y) + 2 \text{Cov}(X, Y)$ ,
- $D^2\left(\sum_{i=1}^n X_i\right) = \sum_{i=1}^n D^2(X_i) + 2 \sum_{1 \leq i < j \leq n} \text{Cov}(X_i, X_j)$ ,
- $X, Y$  függetlenek  $\Rightarrow \text{Cov}(X, Y) = 0$ .

**5.1.16 Definíció (Korreláció).** Ha  $X$  és  $Y$  valószínűségi változóknak létezik szórása és kovarianciája, akkor a korrelációjuk

$$R(X, Y) = \frac{\text{Cov}(X, Y)}{D(X)D(Y)}.$$

**5.1.17 Állítás.** A korreláció két valószínűségi változó lineáris kapcsolatát méri:

- $R > 0 \Rightarrow$  pozitív a kapcsolat,



- $R < 0 \Rightarrow$  negatív a kapcsolat,
- $R^2 \approx 1 \Rightarrow$  erős a kapcsolat,
- $R^2 \approx 0.5 \Rightarrow$  közepes a kapcsolat,
- $R^2 \approx 0 \Rightarrow$  gyenge a kapcsolat.

**5.1.18 Definíció (Eloszlásfüggvény).** Az  $x \in \mathbb{R}$ :  $F_X(x) = P(X < x)$  függvény az  $X$  valószínűségi változó eloszlásfüggvénye.

**5.1.19 Tétel (Tulajdonságai).** Az eloszlásfüggvény tulajdonságai:

- $0 \leq F_X(x) \leq 1$ ,
- monoton növő,
- balról folytonos
- $\lim_{x \rightarrow -\infty} F(x) = 0$  és  $\lim_{x \rightarrow \infty} F(x) = 1$ .

**5.1.20 Definíció (Folytonos valószínűségi változó és sűrűségfüggvény).** Az  $X$  valószínűségi változó *abszolút folytonos*, ha

$$\exists f(x): F(x) = \int_{-\infty}^x f(t)dt.$$

Ilyenkor  $f(x)$ -et *sűrűségfüggvénynek* nevezzük.

**5.1.21 Állítás.** Legyen  $X$  abszolút folytonos eloszlású. Ekkor

- $f(x) = F'(x)$ ,
- $f(x) \geq 0$ ,
- $\int_{-\infty}^{\infty} f(x)dx = 1$ ,
- $\forall x: P(X = x) = 0$ ,
- $P(a < X \leq b) = P(a \leq X < b) = F(b) - F(a)$ .

**5.1.22 Definíció (Várható érték).** Abszolút folytonos valószínűségi változó várható értéke

$$E(X) = \int_{-\infty}^{\infty} xf(x)dx.$$

**5.1.23 Tétel (Centrális határeloszlás).** Legyenek  $X_1, X_2, \dots$  független, azonos eloszlású valószínűségi változók.  $E(X_1) = m, D^2(X_1) = \sigma^2 < \infty$ . Ekkor

$$\lim_{n \rightarrow \infty} P\left(\frac{x_1 + \dots + x_{n-nm}}{\sigma\sqrt{n}} < x\right) = \phi(x).$$

**5.1.24 Tétel (Nagy számok törvénye).** Legyenek  $X_1, X_2, \dots$  független, azonos eloszlású valószínűségi változók.  $E(X_1) = m < \infty$ . Ekkor

$$\lim_{n \rightarrow \infty} P\left(\frac{X_1 + \dots + X_n}{n} = m\right) = 1.$$

## 5.2. Statisztikai becslések

5.2.1 Definíció (Torzítatlan becslés).  $T(X)$  statisztika torzítatlan becslés  $\theta$ -nak, ha

$$\forall \theta: E_{\theta}T(X) = \theta.$$

5.2.2 Definíció (Várható érték becslése).

$$E(X) \approx \frac{X_1 + \dots + X_n}{n}.$$

5.2.3 Definíció (Szórás becslése).

$$D^2(X) \approx \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n-1} = S_n^*.$$

5.2.4 Definíció (Likelihood függvény). Legyen  $X = (X_1, \dots, X_n)$  független, azonos eloszlású valószínűségi változókból álló minta. Ekkor a minta közös sűrűségfüggvénye:

$$L := f_a(x) = \prod_{i=1}^n f_a(x_i) \text{ TODO}$$

## 5.3. Statisztikai próbák

Egy mintára vonatkozó feltevések elfogadása vagy elutasítása.

$H_0$  - Nullhipotézis (amit igazolni szeretnénk),

$H_1$  - Ellenhipotézis (amikor  $H_0$  nem teljesül).

	$H_0$ igaz	$H_0$ hamis
$H_0$ elfogadása	Helyes Döntés	Másodfajú hiba
$H_0$ elvetése	Elsőfajú hiba	Helyes döntés

Ha a szórás ismert, akkor  $u$ -próbát alkalmazunk, különben  $t$ -próbát.

### 5.3.1. $u$ -próbák

- Egymintás eset:  $X_1, \dots, X_n \sim N(m, \sigma^2)$  minta, ahol  $m$  paraméter. Ekkor

$$u = \sqrt{n} \frac{\bar{X} - m_0}{\sigma}.$$

$H_0: m = m_0$  nullhipotézist elfogadjuk, ha  $|u| \leq u_{\frac{p}{2}}$ , ahol  $u_{\frac{p}{2}}$  a standard normális eloszlás  $1 - \frac{p}{2}$  kvantilise.

- Kétmintás eset:  $X_1, \dots, X_n \sim (m_1, \sigma_1^2)$  és  $Y_1, \dots, Y_m \sim N(m_2, \sigma_2^2)$  független minták, ahol  $m_1, m_2$  paraméter. Ekkor  $H_0: m_1 = m_2$  nullhipotézis esetén a próbastatisztika:

$$u = \frac{\bar{X} - \bar{Y}}{\sqrt{\frac{\sigma_1^2}{n} + \frac{\sigma_2^2}{m}}}.$$

### 5.3.2. t-próbák

- Egymintás eset:

$$t = \sqrt{n} \cdot \frac{\bar{X} - m_0}{S^{*2}}$$

$H_0$  nullhipotézist elfogadjuk, ha  $|t| \leq t_{\frac{\alpha}{2}, n-1}$ , ahol  $n - 1$  a Student-eloszlás szabadságfoka.

- Kétmintás eset:  $X_1, \dots, X_n \sim (m_1, \sigma_1^2)$  és  $Y_1, \dots, Y_m \sim N(m_2, \sigma_2^2)$  független minták, ahol  $m_1, m_2, \sigma_1 = \sigma_2$  paraméterek. Ekkor a  $H_0: m_1 = m_2$  nullhipotézis esetén a próbastatisztika:

$$t = \sqrt{\frac{nm(n+m-2)}{n+m}} \cdot \frac{\bar{X} - \bar{Y}}{\sqrt{s_x^{*2} + s_y^{*2}}}.$$

## 6. Mesterséges intelligencia

MI PROBLÉMÁK ÉS AZ ÚTKERESÉSI FELADAT KAPCSOLATA. ÁLLAPOTTÉR REPREZENTÁCIÓ. HEURISZTIKUS ÚTKERESŐ ALGORITMUSOK: LOKÁLIS KERESÉSEK (HEGYMÁSZÓ MÓDSZER, TABU-KERESÉS, SZIMULÁLT HŰTÉS), VISSZALÉPÉSES KERESÉS, HEURISZTIKUS GRÁFKERESŐ ELJÁRÁSOK (A, A\*, AC, B ALGORITMUSOK). KÉTSZEMÉLYES JÁTÉKOK.

---

Az MI az intelligens gondolkodás, számítógépes reprodukálása szempontjából hasznos elveket, módszereket, technikákat kutatja, rendszerezi. Megoldandó feladatai: nehezek, mert ezek problématerülete hatalmas, a megoldás megkeresése kellő intuíció hiányában kombinatorikus robbanáshoz vezethet.

### 6.1. Útkeresési probléma

Útkeresési problémaként sok MI feladat fogalmazható meg úgy, hogy a feladat modellje alapján megadunk egy olyan élsúlyozott irányított gráfot, amelyben adott csúcsból adott csúcsba vezető utak jelképezik a feladat egy-egy megoldását. Ezt a feladat *gráfrepresentációjának* nevezzük, amely magába foglal egy úgynevezett  $\delta$ -gráfot (olyan élsúlyozott irányított gráf, ahol egy csúcsból kivezető élek száma véges, és az élek költségére megadható egy  $\delta$  pozitív alsó korlát), az abban kijelölt startcsúcsot és egy vagy több célcsúcot. Ebben a reprezentációs gráfban keresünk egy startcsúcsból kiinduló célcsúcsba futó utat, esetenként egy legolcsóbbt.

### 6.2. Állapottér reprezentáció

Az állapottér reprezentáció egy olyan lehetséges (de nem az egyetlen) módszer a feladatok modellezésére, amelyet aztán természetes módon lehet gráfrepresentációként is megfogalmazni. Négy eleme van:

- *Állapottér*, amely a probléma a homlokterében álló adat(objektum) lehetséges értékeinek (állapotainak) halmaza. Gyakran egy alaphalmaz, amelyet egy alkalmas invariáns leszűkít,
- *Műveletek* (előfeltétel + hatás), amelyek állapotból állapotba vezetnek,
- *Kezdőállapot(ok)* vagy azokat leíró kezdőfeltétel,
- *Célláallapot(ok)* vagy célfeltétel.

Az állapot-gráf (egy speciális reprezentációs gráf) az állapotokat, mint csúcsokat, a műveletek hatásait, mint éleket tartalmazza.

### 6.3. Keresések

Egy általános *kereső rendszer* részei:

- *globális munkaterület*: a keresés memóriája,
- *a keresési szabályok*: a memória tartalmát változtatják meg,
- *vezérlési stratégia*: adott pillanatban alkalmas szabályt választ.

A vezérlési stratégiának van egy általános, elsődleges eleme, lehet egy másodlagos (az alkalmazott reprezentációs modell sajátosságait kihasználó) eleme és a konkrét feladatra építő elem. Ez utóbbi a *heurisztika*, a konkrét feladatból származó extra ismeret, amelyet közvetlenül a vezérlési stratégiába építünk be az eredményesség és hatékonyság javítása céljából.

### 6.3.1. Lokális keresések

A lokális keresések egyetlen aktuális csúcsot és annak szűk környezetét tárolják a globális munkaterületen. Keresési szabályai az aktuális csúcs minden lépésben a szomszédjai közül vett lehetőleg "jobb" gyerekcsúccsal cserélik le. A vezérlési stratégiájuk a "jobbság" eldöntéséhez egy *rátermettségi függvényt* használ, amely annál jobb értéket ad egy csúcsra, minél közelebb esik az a célhoz. Mivel a keresés "elfelejt", hogy honnan jött, a döntések nem vonhatók vissza, ez egy *nem módosítható* vezérlési stratégia.

Lokális kereséssel megoldható feladatok azok, ahol egy lokálisan hozott rossz döntés nem zárja ki a cél megtalálását. Ehhez vagy egy erősen összefüggő reprezentációs-gráf, vagy jó heurisztikára épített célfüggvény kell. Jellemző alkalmazás: adott tulajdonságú elem keresése, függvény optimumának keresése.

- *Hegymászó algoritmus*: Minden lépésben az aktuális csúcs legjobb gyermekére lép, de kizárja a szülőre való visszalépést. Zsákutcába (aktuális csúcsból nem vezet ki él) beragad, körök mentén végtelen ciklusba kerülhet, ha a rátermettségi függvény nem tökéletes.
- *Tabu keresés*: Az aktuális csúcson ( $n$ ) kívül nyilvántartja még az eddig legjobbnak bizonyult csúcsot ( $n^*$ ) és az utolsó néhány érintett csúcsot; ez a (sor tulajdonságú) *tabu halmaz*. Minden lépésben az aktuális csúcs gyermeki közül, kivéve a tabu halmazban lévőket, a legjobbat választja új aktuális csúcsnak, (ezáltal felismeri a tabu halmaz méreténél nem nagyobb köröket), frissíti a tabu halmazt, és ha  $n$  jobb, mint az  $n^*$ , akkor  $n^*$ -ot lecseréli  $n$ -re.
- *Szimulált hűtés algoritmus*: A következő csúcs választása véletlenszerű. Ha a kiválasztott csúcs ( $r$ ) célfüggvény-értéke jobb, mint az aktuális csúcsé ( $n$ ), akkor odalép, ha rosszabb, akkor az új csúcs elfogadásának valószínűsége fordítottan arányos  $f(n) - f(r)$  különbséggel. Ez az arány ráadásul folyamatosan változik a keresés során: ugyanolyan különbség esetén kezdetben nagyobb, később kisebb valószínűséggel fogja a rosszabb értékű  $r$  csúcsot választani.

### 6.3.2. Visszalépéses keresések

A Startcsúcsból az aktuális csúcsba vezető utat (és az arról leágazó még ki nem próbált éleket) tartja nyilván (globális munkaterületen), a nyilvántartott út végéhez egy új (ki nem próbált) élt fűzhet vagy a legutolsó élt törölheti (visszalépés szabálya), a visszalépést a legvégső esetben alkalmazza. A visszalépés teszi lehetővé azt, hogy egy korábbi továbblépésről hozott döntés megváltozhasson. Ez tehát egy *módosítható vezérlési stratégia*. A keresésbe sorrendi és vágó heurisztika építhető. Mindkettő lokálisan, az aktuális csúcsból kivezető, még ki nem próbált élekre vonatkozik.

Visszalépés feltételei: zsákutca, zsákutca torkolat, kör, mélységi korlát.

- VL1 (nincs kör- és mélységi korlát figyelés) véges körmentes irányított gráfokon terminál, és ha van megoldás, akkor talál egyet.

- VL2 (általános)  $\delta$ -gráfokon terminál, és ha van megoldás a mélységi korláton belül, akkor talál egyet.

Könnyen implementálható, kicsi memória igényű, mindig terminál, és ha van (a mélységi korlát alatt), akkor talál megoldást. De nem garantál optimális megoldást, egy kezdetben hozott rossz döntést csak nagyon sok lépés után képes korrigálni és egy zsákutca, szakaszt többször is bejárhat, ha abba többféle úton is el lehet jutni.

### 6.3.3. Gráfkeresések

A globális munkaterületen a startcsúcsból kiinduló már feltárt utat találhatók (ez a *kereső gráf*), külön megjelölve az utak azon csúcsait, amelyeken még nem (vagy nem eléggé jól) ismerjük a rákövetkezőit. Ezek a *nyílt csúcsok*. A keresés szabályai egy nyílt csúcsot terjesztenek ki, azaz előállítják (vagy újra előállítják) a csúcs össze rákövetkezőjét. A vezérlési stratégia a legkedvezőbb nyílt csúcs kiválasztására törekszik, ehhez egy *kiértékelő függvényt* ( $f$ ) használ. Mivel egy nyílt csúcs, amely egy adott pillanatban nem kerül kiválasztásra, később még kiválasztódhat, ezért itt egy módosítható vezérlési stratégia valósul meg.

A keresés minden csúcsához nyilvántart egy odavezető utat ( $\pi$  visszamutató pointerek segítségével), valamint az út költségét ( $g$ ). Ezeket az értékeket működés közben alakítja ki, amikor a csúcsot először felfedezi vagy később egy olcsóbb utat talál hozzá. Mindkét esetben (amikor módosultak a csúcs ezen értékei) a csúcs nyílttá válik. Amikor egy már korábban kiterjesztett csúcs újra nyílt lesz, akkor a már korábban felfedezett leszármazottainál a visszafelé mutató pointerekkel kijelölt út költsége nem feltétlenül egyezik majd meg a nyilvántartott  $g$  értékkel, és az sem biztos, hogy ezek az értékek az eddig talált legolcsóbb útra vonatkoznak.

- Nem-informált gráfkeresések: *mélységi gráfkeresés* ( $f = -g, \forall (n, m)$  élre  $c(n, m) = 1$ ), *szélességi gráfkeresés* ( $f = g, c(n, m) = 1$ ), *egyenletes gráfkeresés* ( $f = g$ ),
- Heurisztikus gráfkeresések  $f$ -je a  $h$  a heurisztikus függvényre épül, amely minden csúcsban hátralevő optimális  $h^*$  költséget becsli. Ilyen az *előre tekintő gráfkeresés* ( $f = h$ ), az A algoritmus ( $f = g + h, h \geq 0$ ), az  $A^*$  algoritmus ( $f = g + h, h^* \geq h \geq 0$   $h$  megengedhető), az  $A^C$  algoritmus ( $f = g + h, h^* \geq h \geq 0, \forall (n, m)$  élre:  $h(n) - h(m) \leq c(n, m)$ ), és B algoritmus ( $f = g + h, h \geq 0$  helyett a  $g$ -t használjuk a kiterjesztendő csúcs kiválasztására azon nyílt csúcsok közül, amelyek  $f$  értéke kisebb, mint az eddig kiterjesztett csúcsok  $f$  értékeinek maximuma).

Véges  $\delta$ -gráfokon minden gráfkeresés terminál, és ha van megoldás, talál egyet. A nevezetes gráfkeresések többsége végtelen nagy gráfokon is találnak megoldást, ha van megoldás. (Kivétel az előre-tekintő keresés és a mélységi korlátot nem használó mélységi gráfkeresés.)

Az  $A^*$ ,  $A^C$  algoritmusok optimális megoldást találnak, ha van megoldás. Az  $A^C$  algoritmus egy csúcsot legfeljebb egyszer terjeszt csak ki.

Egy gráfkeresés memóriaigényét a kiterjesztett csúcsok számával, futási idejét ezek kiterjesztéseinek számával mérjük. (Egy csúcs általában többször is kiterjesztődhet, de  $\delta$ -gráfokban csak véges sokszor.). Az  $A^*$  algoritmusnál a futási idő legrosszabb esetben exponenciálisan függ a kiterjesztett csúcsok számától, de ha olyan heurisztikást választunk, amelyre már  $A^C$  algoritmust kapunk, akkor a futási idő lineáris lesz. Persze ezzel a másik heurisztikával változik a kiterjesztett csúcsok száma is, így nem biztos, hogy egy  $A^C$  algoritmus ugyanazon a gráfon összességében kevesebb kiterjesztést végez, mint egy csúcsot

többször is kiterjesztő  $A^*$  algoritmus. A B algoritmus futási ideje négyzete, és ha olyan heurisztikus függvényt használ, mint az  $A^*$  algoritmus (azaz megengedőt), akkor ugyanúgy optimális megoldást talál (ha van megoldás) és a kiterjesztett csúcsok száma (mellesleg a halmaza is) megegyezik az  $A^*$  algoritmus által kiterjesztett csúcsokéval.

#### 6.3.4. Kétszemélyes játékok

A játékokat állapottér-reprezentációval szokás leírni, és az állapot-gráfot faként ábrázolni.

A *győztes (vagy nem vesztes) stratégia* egy olyan elv, amelyet betartva egy játékos az ellenfél minden lépésére tud olyan választ adni, hogy megnyerje (ne veszítse el) a játékot. Valamelyik játékosnak biztosan van győztes (nem vesztes) stratégiája. Győztes (nem vesztes) stratégia keresése a *játékfaban* kombinatorikus robbanást okozat, ezért e helyett részfa kiértékelést szoktak alkalmazni a soron következő jó lépés meghatározásához.

A *minimax* algoritmus az aktuális állásból felépíti a játékfa egy részét, kiértékeli annak leveleit aszerint, hogy azok által képviselt állások milyen mértékben kedveznek nekünk vagy az ellenfélnek, majd szintenként váltakozva az ellenfél szintjein a gyerekcsúcsok értékeinek minimumát, a saját szintjeinken azok maximumát futtatjuk fel a szülőcsúcsához. Ahonnan a gyökérhez kerül érték, az lesz soron következő lépésünk.

A minimax legismertebb módosítása az *alfa-béta* algoritmus, amely egyfelől kisebb memória igényű (egyszerre csak egy ágat tárolt a vizsgált részfaból) másfelől egy sajátos vágási stratégia miatt jóval kevesebb csúcsot vizsgál meg, mint a minimax.

## 7. Programozás

EGYSZERŰ PROGRAMOZÁSI FELADAT MEGOLDÁSÁNAK LÉPÉSEI (SPECIFIKÁLÁS, TERVEZÉS, MEGVALÓSÍTÁS, TESZTELÉS). AZ ADATTÍPUS FOGALMA (TÍPUSSPECIFIKÁCIÓ, MŰVELETEK, REPREZENTÁCIÓ, INVARIÁNS, IMPLEMENTÁCIÓ). A VISSZAVEZETÉS MÓDSZERE. A FELSOROLÓ TÍPUS SPECIFIKÁCIÓJA. FELSOROLÓRA MEGFOGALMAZOTT PROGRAMOZÁSI TÉTELEK (ÖSSZEGZÉS, SZÁMLÁLÁS, MAXIMUM KIVÁLASZTÁS, FELTÉTELES MAXIMUMKERESÉS, LINEÁRIS KERESÉS, KIVÁLASZTÁS). NEVEZETES GYŰJTEMÉNYEK (INTERVALLUM, TÖMB, SOROZAT, HALMAZ, SZEKVENCIÁLIS INPUTFÁJL) FELSOROLÓI.

---

### 7.1. Egyszerű programozási feladat megoldásának lépései

Egy egyszerű programozási feladatot négy fő lépésben oldhatunk meg, amely leginkább az iparágban gyakran használt vízesés modellre hasonlít. Minden lépés csak az őt megelőző lépés befejezése után indul.

#### 7.1.1. Specifikáció

A specifikáció két részre bontható:

- *Feladatleírás:* általában a megrendelő készíti el, köznapi nyelven írja le, hogy mi a program pontos célja.
- *Követelmények leírása:* A programozó által írt működési modell. A feladat specifikálása során meghatározzuk a bemenő adatok tetszőlegesen rögzített értékeihez tartozó kezdőállapotok halmazát, valamint az ehhez tartozó célállapotok halmazát. A feladat specifikációja tartalmazza:
  - az állapotteret, azaz a feladat lényeges adatainak típusérték-halmazait az egyes adatokhoz tartozó változó nevekkkel együtt;
  - az előfeltételt, amely a kezdőállapotok azon halmazát leíró logikai állítás, amely rögzíti a bemenő változók egy lehetséges, de tetszőleges kezdőértékét (ezeket általában a megfelelő változónév vesszős alakjával jelöljük);
  - az utófeltételt, amely a fenti kezdőállapotokhoz rendelt célállapotok halmazát megadó logikai állítás.

#### 7.1.2. Tervezés

A tervezés célja olyan leírások, adatmodellek, programszerkezetek elkészítése, amely alapján elkészíthető a program. A tervezés nem nyelv-specifikus, így a programozási nyelvek sajátosságait nem kell figyelembe venni. Tervezési hibát a megvalósítás során nehéz javítani, hiszen a módosítás után vizsgálni kell, hogy megfelel-e a specifikációnak és a hiba mélységétől függően akár az egész programot újra kell írni.

#### 7.1.3. Megvalósítás

A megoldandó feladatnak megfelelően ki kell választani a programozási nyelvet, mely segítségével a feladat implementálható a specifikációban meghatározott követelmények szerint. A megvalósítás során a tervezésben meghatározott módszertanokkal, programszerkezetekkel és algoritmusokkal kell dolgozni. A tervezéstől eltérni csak tervezési hiba esetén vagy a kiválasztott programnyelv tulajdonsága miatt érdemes (pl. egyszerűbb megoldást nyújt egy problémára).



Az adott cég/programnyelv kódolási konvencióit fontos betartani, valamint elengedhetetlen a megfelelő kommentezés a későbbi érthetőség és olvashatóság miatt.

#### 7.1.4. Tesztelés

A tesztelés célja, hogy biztosak legyünk abban, hogy a program megfelel a specifikációban leírt követelményeknek. A teszteseteket már a korábbi lépések során elkezdhetjük összegyűjteni, így sokkal átfogóbb tesztelést hajthatunk végre. Többféle szempontból is csoportosíthatjuk a tesztelési módszereket.

- *Fehér doboz tesztelés:* a tesztelő a kód ismeretében hajtja végre a teszteseteket,
- *Fekete doboz tesztelés:* a tesztelő csak a bemeneti és kimeneti adatokat figyeli.

#### Tesztelési szintek

- *Unit teszt:* A kód legkisebb egységeit teszteli, például egy függvény működését,
- *Modul teszt:* Egy-egy modul/komponens működésének tesztelése,
- *Rendszer teszt:* A modulok/komponensek együttes működése. Célja, hogy felfedje a modulok között előforduló hibás kommunikációt.

## 7.2. Az adattípus fogalma

**7.2.1 Definíció (Típus).** A típus a  $\mathcal{T} = \{\rho, I, \mathbb{S}\}$  hármas, ha

1.  $\rho \subseteq E^* \times T$  reprezentációs függvény (ahol  $E$  az elemi típusérték halmaz,  $T$  típusérték halmaz),
2.  $I: E^* \rightarrow \mathbb{L}$  típusinvariáns,
3.  $\mathbb{S} = \{S_1, \dots, S_m\}: \forall i \in [1, m]: S_i \subseteq B_i \times B_i^{**}$  program :  
 $B_i = B_{i_1} \times \dots \times B_{i_m}: \exists j \in [1, m]: B_{i_j} = E^* \wedge \nexists j \in [1, m]: B_{i_j} = T$ .

**7.2.2 Definíció (Típus-specifikáció).** A típus-specifikáció a  $\mathcal{T}_s = (T, I_s, \mathbb{F}_s)$  hármas, ha

1.  $T$  tetszőleges alaphalmaz
2.  $I_s: H \rightarrow \mathbb{L}$  specifikációs invariáns
3.  $T_S = [I_s]$  a típusérték-halmaz
4.  $\mathbb{F} = \{F_1, \dots, F_n\}$  a típusműveletek specifikációja, ahol

$$\forall i \in [1, n]: F_i \subseteq A_i \times A_i, A_i = A_{i_1} \times \dots \times A_{i_n}: \exists j \in [1, n]: A_{i_j} = T.$$

**7.2.3 Definíció (Invariáns).** Olyan tulajdonság, amelyet nem sérthetünk meg, a program futása során mindig fennáll.

**7.2.4 Definíció (Reprezentáció).** A reprezentáció a típus-specifikáció típusérték-halmazának leképzése a konkrét típusban, amit a reprezentációs függvény ad meg. Vagyis reprezentációnak nevezünk azt, hogy egy típust valamilyen típusok, módszerek segítségével valósítunk meg. Például a vermet megvalósíthatjuk egy tömb segítségével.

**7.2.5 Definíció (Implementáció).** A típus implementációja a típusérték-halmaz megadása és a típusműveletek definiálása.

### 7.2.6. Szövegesen

Egy adat típusát több nézőpontból lehet szemlélni. Amikor arra vagyunk kíváncsiak, hogy mire használhatjuk az adatot, azaz melyek a lehetséges értékei és ezekkel milyen műveleteket lehet végezni, akkor a típus felületét, kifelé megmutatkozó arcát nézzük; ezt szokták a típus interfészének, a típus specifikációjának nevezni. A típus-specifikáció tehát a típusérték-halmaz és a típus-műveletek együttese, ahol a típus-műveletek mindegyike egy olyan feladatot ír le, amelyik állapotterében megjelenik a típusérték-halmaz.

Amikor viszont az adat típusának egy adott programozási környezetbe való beágyazása a cél, akkor azt vizsgáljuk, hogy milyen elemekkel helyettesíthetőek, ábrázolhatóak a típusértékek, illetve melyek azok a programok, amelyek hatása a típusműveletekével azonos annak ellenére, hogy nem közvetlenül a típusértékekkel, hanem az azokat helyettesítő elemekkel dolgoznak. (Például egy vermet ábrázolhatunk egy tömbbel és egy indexszel, a `Top()` művelet pedig implementálható a tömb „indexedik” elemének lekérdezésével.) A helyettesítő elemeknek gyakran ki kell elégítenie speciális megkötéseket, ez az ún. típus-invariáns, amelynek a típusértéket helyettesítő elemekre mindig teljesülnie kell. Ha vesszük a típusértékek helyettesítő elemeit, a helyettesítés módját, azaz a típusértékek reprezentációját (ábrázolását) a típus-invariánssal, valamint a típusműveleteket kiváltó programokat, azaz a típusműveletek implementációit, akkor ezeket együtt a típus megvalósításának (realizálásának) nevezzük.

Egy adat típusa magába foglalja a típus-specifikációt és az ennek megfelelő, az ezt megvalósító típus-realizációt.

### 7.3. Visszavezetés

Egy feladat megoldását egy már ismert és megoldott feladat megoldására vezetjük vissza. Két feladat fontosabb részeit megfeleltetjük egymásnak. Ez az analóg programozás.

A programozási tétel egy mintafeladat és annak megoldó programja (mintaprogram). A visszavezetés során a mintafeladat mintaprogramját sablonként használva állítjuk elő a kitűzött feladat megoldó programját úgy, hogy figyelembe vesszük a kitűzött- és a mintafeladat specifikációkban felfedezett eltéréseket.

### 7.4. Felsoroló specifikációja

Jellemezzünk egy olyan objektumot (*felsoroló*), amely segítségével egy felsorolható adatnak (vektor, tömb, szekvenciális input fájl, stb) az elemeit egymás után elő lehet állítani.

A felsoroló objektum véges sok elemi érték felsorolását teszi lehetővé a megfelelő műveletek segítségével:

- `t.First` - a `t:Enor` felsorolót az első elemére állítjuk: `t := First(t)`
- `t.Next` - a soron következő elemre ugrik
- `t.Current` - visszaadja az aktuális elemet
- `t.End` - hamisat ad vissza, ha van aktuális elem, a felsorolás végét igaz értékkel jelzi

### 7.5. Felsorolóra megfogalmazott programozási tételek

#### 7.5.1. Összegzés

**Feladat:**  $t$  egy  $E$ -beli elemeket felsoroló objektum,  $f : E \rightarrow H$ ,  $+$  :  $H \times H \rightarrow H$  asszociatív, bal oldali nulla elemes művelet. Mennyi az  $f$ -nek a  $t$  elemeihez rendelt értékeinek összege?

**Specifikáció:**  $A = (t : \text{enor}(E), s : H)$   
 Ef =  $(t = t')$   
 Uf =  $(s = \sum_{e \in t'} f(e))$

**Algoritmus**

$s := 0$
$t.First()$
$\neg t.End()$
$s := s + f(t.Current())$
$t.Next()$

**7.5.2. Számlálás**

**Feladat:**  $t$  egy  $E$ -beli elemeket felsoroló objektum,  $\beta : E \rightarrow \mathbb{L}$ . A felsoroló hány elemére teljesül a  $\beta$ ?

**Specifikáció:**  $A = (t : \text{enor}(E), c : \mathbb{N})$   
 Ef =  $(t = t')$   
 Uf =  $(c = \sum_{\substack{e \in t' \\ \beta(e)}} 1)$

**Algoritmus**

$c := 0$
$t.First()$
$\neg t.End()$
$\beta(t.Current())$
$c := c + 1$ -
$t.Next()$

**7.5.3. Maximum kiválasztás**

**Feladat:**  $t$  egy  $E$ -beli elemeket felsoroló objektum,  $f : E \rightarrow H$ , és  $t$  nem üres,  $H$  elemei teljesen rendezhetőek. Hol veszi fel  $f$  a  $t$  elemein a maximális értékét?

**Specifikáció:**  $A = (t : \text{enor}(E), \text{elem} : E, \text{max} : H)$   
 Ef =  $(t = t' \wedge |t| > 0)$   
 Uf =  $((\text{max}, \text{elem}) = \text{MAX}_{e \in t'} f(e))$

**Algoritmus**

$t.First()$
$\text{max}, \text{elem} := f(t.Current()), t.Current()$
$t.Next()$
$\neg t.End()$
$f(t.Current()) > \text{max}$
$\text{max}, \text{elem} := f(t.Current()), t.Current()$ -
$t.Next()$

#### 7.5.4. Feltételes maximumkeresés

**Feladat:**  $t$  egy  $E$ -beli elemeket felsoroló objektum,  $f : E \rightarrow H$ ,  $\beta : E \rightarrow \mathbb{L}$ ,  $H$  elemei teljesen rendezhetőek. Határozzuk meg  $t$  azon elemeihez rendelt  $f$  szerinti értékek között a legnagyobbat, amelyek kielégítik a  $\beta$  feltételt.

**Specifikáció:**  $A = (t : \text{enor}(E), l : \mathbb{L}, \text{elem} : E, \text{max} : H)$   
 $Ef = (t = t')$   
 $Uf = ((l, \text{max}, \text{elem}) = \text{MAX}_{\substack{e \in t' \\ \beta(e)}} f(e))$

##### Algoritmus

$l := \downarrow$		
$t.First()$		
$\neg t.End()$		
$\neg \beta(t.Current())$	$l \wedge \beta(t.Current())$	$\neg l \wedge \beta(t.Current())$
SKIP	$f(t.Current()) > \text{max}$ $\text{max} := f(t.Current())$ $\text{elem} := t.Current()$	SKIP
$l, \text{max}, \text{elem} := \uparrow,$ $f(t.Current()), t.Current()$		
$t.Next()$		

#### 7.5.5. Kiválasztás

**Feladat:**  $t$  egy  $E$ -beli elemeket felsoroló objektum,  $\beta : E \rightarrow \mathbb{L}$ , Adjuk meg a  $t$  bejárása során az első olyan elemi értéket, amelyre  $\beta$  igazat ad!

**Specifikáció:**  $A = (t : \text{enor}(E), \text{elem} : E)$   
 $Ef = (t = t' \wedge \exists i \in [1..|t|] : \beta(t_i))$   
 $Uf = ((\text{elem}, t) = \text{SELECT}_{\text{elem} \in t'} \beta(\text{elem}))$

##### Algoritmus

$t.First()$
$\neg \beta(t.Current())$
$t.Next()$
$\text{elem} := t.Current()$

#### 7.5.6. Lineáris keresés

**Feladat:**  $t$  egy  $E$ -beli elemeket felsoroló objektum,  $\beta : E \rightarrow \mathbb{L}$ . Keressük a  $t$  bejárása során az első olyan elemi értéket, amire teljesül a  $\beta$  feltétel.

**Specifikáció:**  $A = (t : \text{enor}(E), \text{elem} : E, l : \mathbb{L})$   
 $Ef = (t = t')$   
 $Uf = ((l, \text{elem}, t) = \text{SEARCH}_{e \in t'} \beta(e))$

##### Algoritmus

$l := \downarrow$
$t.First()$
$\neg l \wedge \neg t.End()$
$l, \text{elem} := \beta(t.Current()), t.Current()$
$t.Next()$

## 7.6. Nevezets gyűjtemények

Több nevezetes gyűjteményt is tudunk a felsoroló típussal reprezentálni.

### 7.6.1. Intervallum

**Gyűjtemény reprezentációja:**  $[m, n]$  intervallum két végpontja  $(m, n)$  + index  $(i)$ .

**Műveletek implementációja**

- $i := m \sim t.First()$
- $i \sim t.Current()$
- $i := i + 1 \sim t.Next()$
- $i > n \sim t.End()$

### 7.6.2. Sorozat

**Gyűjtemény reprezentációja:** Sorozat típusú adat  $(s)$  + indexváltozó  $(i)$ .

**Műveletek implementációja**

- $i := 1 \sim t.First()$
- $i \sim t.Current()$
- $i := i + 1 \sim t.Next()$
- $i > |n| \sim t.End()$

### 7.6.3. Tömb

**Gyűjtemény reprezentációja:** A  $v$  vektor indextartománya  $[m, n]$  + indexváltozó  $(i)$ .

**Műveletek implementációja**

- $i := m \sim t.First()$
- $v_i \sim t.Current()$
- $i := i + 1 \sim t.Next()$
- $i > n \sim t.End()$

### 7.6.4. Szekvenciális fájl

**Gyűjtemény reprezentációja:** Az inputfájlból legutoljára olvasott elemi érték (df) + az utolsó olvasás státusza (sf).

**Műveletek implementációja**

- $df \sim t.Current()$
- $sf, df, f \sim t.First(), t.Next()$
- $sf = abnorm \sim t.End()$

### 7.6.5. Halmaz

**Gyűjtemény reprezentációja:** A felsorolandó elemeket tartalmazó  $h$  halmaz.

**Műveletek implementációja**

- $e \sim t.Current()$
- $e \in h \sim t.First(), t.Next()$
- $h = \emptyset \sim t.End()$

## 8. Programfejlesztési modellek

NAGY RENDSZEREK FEJLESZTÉSI FÁZISAI, KAPCSOLATAIK. AZ OBJEKTUMELVŰ MODELLEZÉS NÉZETRENDSZEREI. STATIKUS MODELL (OSZTÁLYDIAGRAM, OBJEKTUMDIAGRAM). DINAMIKUS MODELL (ÁLLAPOTDIAGRAM, SZEKVENCIADIAGRAM, EGYÜTTMŰKÖDÉSI DIAGRAM, TEVÉKENYSÉGDIAGRAM). HASZNÁLATI ESETEK DIAGRAMJA.

## 9. Programok fordítása és végrehajtása

FORDÍTÁS ÉS INTERPRETÁLÁS ÖSSZEHAONLÍTÁSA. FORDÍTÁSI EGYSÉG ÉS A SZERKEZTÉS FOGALMA. FORDÍTÓPROGRAMOK KOMPONENSEINEK FELADATA ÉS MŰKÖDÉSI ELVEIK VÁZLATOS ISMERTETÉSE. KÓDGENERÁLÁS ASSEMBLYBEN ALAPVETŐ IMPERATÍV VEZÉRLÉSI SZERKEZETEKHEZ. A SZEKVENCIÁLIS ÉS PÁRHUZAMOS/ELOSZTOTT VÉGREHAJTÁS ÖSSZEHAONLÍTÁSA.

## 10. Programnyelvi alapok

KIFEJEZÉSEK KIÉRTÉKELÉSÉNEK SZABÁLYAI. VEZÉRLÉSI SZERKEZETEK: UTASÍTÁSOK, REKURZÍÓ. TÍPUSOK: TÖMB, REKORD, OSZTÁLY, ÖRÖKLŐDÉS, STATIKUS ÉS DINAMIKUS KÖTÉS, POLIMORFIZMUS. GENERIKUSOK. HATÓKÖR/LÁTHATÓSÁG. AUTOMATIKUS, STATIKUS ÉS DINAMIKUS ÉLETTARTAM, SZEMÉTTYŰJTÉS. KONSTRUKTOR, DESTRUKTOR. OBJEKTUMOK MÁSOLÁSA, ÖSSZEHAONLÍTÁSA. ALPROGRAMOK, PARAMÉTERÁTADÁS, TÚLTERHELÉS.

## 11. Formális nyelvek

FORMÁLIS NYELVTANOK ÉS A CHOMSKY-FÉLE NYELVOSZTÁLYOK. AUTOMATÁK: VÉGES AUTOMATA, VEREMAUTOMATA. REGULÁRIS NYELVEK TULAJDONSÁGAI ÉS ALKALMAZÁSAI. KÖRNYEZETFÜGGETLEN NYELVEK TULAJDONSÁGAI ÉS ELEMZÉSÜK

---

### 11.1. Alapfogalmak

**11.1.1 Definíció (Ábécé).** Ábécének nevezünk egy tetszőleges véges szimbólumhalmazt.

**11.1.2 Definíció (Betű).** Ábécé elemi betűk.

**11.1.3 Definíció (Szó).** Az  $X$  ábécé betűinek egy tetszőleges véges sorozata az  $X$  ábécé feletti szó.

**11.1.4 Definíció (Nyelv).**  $X^*$  valamely részhalmaza az  $X$  ábécé feletti nyelv.

**11.1.5 Definíció (Nyelvosztály).** Nyelvek valamely összessége nyelvosztály.

### 11.2. Formális nyelvtanok

**11.2.1 Definíció (Nyelvtan).** A  $G = \langle T, N, \mathcal{P}, S \rangle$  négyes nyelvtan, ahol

- $T$ : terminális jelek ábécéje
- $N$ : a nyelvtani (nemterminális) jelek egymástól diszjunkt ábécéje
- $\mathcal{P}$ : produkciós szabályok egy véges halmaza, ahol minden  $P \in \mathcal{P}$  szabály  $p \rightarrow q$  alakú
- $S$ :  $S \in N$  kezdőszimbólum

**11.2.2 Definíció (Mondatforma).** Terminális és nemterminális szimbólumok véges sorozata  $((T \cup N)^*)$

**11.2.3 Definíció (Generált nyelv).**  $G = \langle T, N, \mathcal{P}, S \rangle$  által generált nyelv  $L(G) := \{u \in T^* \mid S \xrightarrow[G]{*} u\}$ .

**11.2.4 Tétel.** *Nem minden nyelv írható le nyelvtannal.*

### 11.3. Nyelvtanok osztályozása

A szabályok alakja alapján a nyelvtanokat különböző osztályokba sorolhatjuk.

#### 11.3.1. Jelölések

- $N$  nyelvtani jelek ábécéje
- $T$  terminális ábécé
- $S$  a kezdőszimbólum
- $p$  egy szabály bal oldala
- $q$  tetszőleges szabály jobb oldala
- $\alpha_1, \alpha_2 \in (T \cup N)^*$  mondatforma



	Alaptípus		Megszorított típus		Normál forma
0. típus	$p \rightarrow q$	$p, q \in (T \cup N)^*$ és $p = \alpha_1 A \alpha_2$ , $A \in N, \alpha_1, \alpha_2 \in (T \cup N)^*$  (Azaz $p$ -ben van legalább egy nyelvtani jel)	$p \rightarrow q$ $S \rightarrow \varepsilon$	Alap típus $+q \neq \varepsilon$  KES szabály	$A, B, C \in N, t \in T$ $AB \rightarrow A \quad BA \rightarrow A$ $AB \rightarrow AC \quad AB \rightarrow CB$ $A \rightarrow BC$ $A \rightarrow t$  KES
1. típus	$p \rightarrow q$ $S \rightarrow \varepsilon$	0. típus + $\downarrow(p) \leq \downarrow(q)$  KES szabály  hosszúság nemcsökkentő nyelvtan	$\alpha_1 A \alpha_2 \rightarrow \alpha_1 q \alpha_2$  $S \rightarrow \varepsilon$	$A \in N, \alpha_1, \alpha_2 \in (T \cup N)^*$ $q \in (T \cup N)^+$  KES szabály  környezetfüggő nyelvtan	$AB \rightarrow AC \quad AB \rightarrow CB$ $A \rightarrow BC$ $A \rightarrow t$  KES Kuroda normál forma
2. típus	$A \rightarrow q$	$A \in N$ $q \in (T \cup N)^*$  környezetfüggetlen nyelvtan	$A \rightarrow q$  $S \rightarrow \varepsilon$	$A \in N$ $q \in (T \cup N)^+$  KES szabály	$A \rightarrow BC$ $A \rightarrow t$  KES Chomsky normál forma
3. típus	$A \rightarrow uB$ $A \rightarrow u$	$A, B \in N$ $u \in T^*$  reguláris nyelvtan	$A \rightarrow tB$ $A \rightarrow t$  $S \rightarrow \varepsilon$	$A, B \in N$ $t \in T$  KES szabály	$A \rightarrow tB$ $A \rightarrow \varepsilon$  3-as normál forma

### 11.3.2. Chomsky-féle nyelvhierarchia

11.3.3 Tétel.  $\mathcal{L}_3 \subseteq \mathcal{L}_2 \subseteq \mathcal{L}_1 \subseteq \mathcal{L}_0$ .

## 11.4. Környezetfüggetlen nyelvek tulajdonságai és elemzésük

11.4.1 Tétel (Nagy Bar-Hillel-lemma).  $\forall L \in \mathcal{L}_2: \exists p, q > 0$  nyelvfüggő konstansok ( $p = p(L), q = q(L)$ ), hogy ha

$$u \in L, l(u) > p \Rightarrow \exists u = xyzvw \text{ felbontása, ahol} \\ (yv) > 0, l(yzv) \leq q \text{ és } \forall i \geq 0: xy^i z v^i w \in L.$$

Vagyis  $L$  minden elég hosszú szavában van két, egymáshoz közel lévő, nem triviális, párhuzamosan beiterálható részszo.

### 11.4.2. Elemzés

A környezetfüggetlen nyelvtanok segítségével szintaxisfát építhetünk. Sikeres szintaxisfa felépítés esetén az elemzés sikerült, különben szintaktikai hibát találunk. Módszerek: legbal levezetés, legjobb levezetés

## 11.5. Reguláris nyelvek tulajdonságai és alkalmazásai

11.5.1 Definíció (Reguláris nyelv). Reguláris nyelvek az alábbi három tulajdonsággal definiált  $\mathcal{L}_{\text{REG}}$  nyelvosztály elemeit:

- $\mathcal{L}_{\text{REG}}$  tartalmazza az elemi nyelveket:  $\emptyset, \{\varepsilon\}, \{t\}$ , ahol  $t \in U$ ,
- $\mathcal{L}_{\text{REG}}$  zárt az unió, konkatenáció és a lezárás műveletekre,
- $\mathcal{L}_{\text{REG}}$  a legszűkebb olyan nyelvosztály, mely az (i), (ii) feltételeknek megfelel.

11.5.2 Tétel (Kleene tétele).  $\mathcal{L}_{\text{REG}} = \mathcal{L}_3$ .

**11.5.3 Tétel.** Az  $\mathcal{L}_3$  nyelvostály zárt a komplementer, a metszet, a különbség és a szimmetrikus differencia műveletekre, a zártsági tétel miatt pedig zárt az unió, konkatenáció és a lezáras műveletekre is.

**11.5.4 Tétel (Kis Bar-Hillel lemma).**  $\forall L \in \mathcal{L}_3 \exists n = n(L) \in \mathbb{N}$  (nyelvfüggő konstans),  $\forall u \in L, l(u) \geq n$  szó :  $\exists u = xyz$  felbontása ( $x, y, z \in T(L)^*$ ), hogy

- $l(x, y) \leq n$ ,
- $l(y) > 0$ ,
- $\forall i \geq 0 \in \mathbb{Z}: xy^i z \in L$ .

### 11.5.5. Felhasználási területek

A reguláris nyelveket a könnyű kezelhetőségük miatt gyakorlatban rendszeresen alkalmazzzák. Például a fordítóprogramok lexikális elemzője általában reguláris kifejezések segítségével azonosítja a nyelv lexikális elemeit. Mintaillesztéses feladatoknál is reguláris kifejezéssel írható le az illesztendő minta. Ebből Knuth-Morris-Pratt (KMP) automatát generálnak.

## 11.6. Automaták

**11.6.1 Definíció (Véges determinisztikus automata (VDA)).**  $\mathcal{A} = \langle Q, T, \delta, q_0, F \rangle$ , ahol

- $Q$ : állapotok véges halmaza
- $T$ : bemenő ábécé
- $\delta: Q \times T \rightarrow Q$ , állapotátmeneti függvény
- $q_0 \in Q$ : kezdőállapot
- $F \subseteq Q$  elfogadási állapotok halmaza

**11.6.2 Definíció (VDA által felismert nyelv).** Legyen  $\mathcal{A} = \langle Q, T, \delta, q_0, F \rangle$  egy VDA. Az  $\mathcal{A}$  által felismer nyelv:

$$L(\mathcal{A}) = \{u \in T^* \mid \delta(q_0, u) \in F\}.$$

**11.6.3 Definíció (Véges nondeterminisztikus automata (VNDA)).** A VNDA felépítése megegyezik a VDA felépítésével, de az állapotátmenet függvénye részhalmazokba képez:  $\delta: Q \times T \rightarrow 2^Q$ .

**11.6.4 Definíció (Veremautomata (1 verem)).**  $\mathcal{V} = \langle Q, T, \Sigma, \delta, q_0, \sigma_0, F \rangle$ , ahol

- $Q$ : állapotok véges halmaza
- $T$ : bemenő ábécé
- $\Sigma$ : a veremábécé
- $\delta$ : állapotátmenet függvény
- $q_0 \in Q$ : kezdőállapot

- $\sigma_0$ : verem kezdőszimbóluma
- $F$ : végállapotok halmaza

A veremautomata elfogadja a szót, ha üres a verem, és elfogadó állapotban van.

**11.6.5 Definíció (Konfiguráció).** Azon adatok összessége, melyektől a gép működése függ, azaz  $[q, v, \alpha_1, \dots, \alpha_n]$ , ahol

- $q$ : aktuális állapot
- $v$ : az input még olvasatlan része
- $\alpha_i$  az  $i$ -edik verem aktuális tartalma

## 12. Logika és számításelmélet

ÍTÉLETKALKULUS ÉS ELSŐRENDŰ PREDIKÁTUMKALKULUS: SZINTAXIS, SZEMANTIKA, EKVIVALENS ÁTALAKÍTÁSOK, A SZEMANTIKUS KÖVETKEZMÉNY FOGALMA, REZOLÚCIÓ. - A KISZÁMÍTHATÓSÁG FOGALMA ÉS A CHURCH-TURING TÉZIS. A TURING-GÉP. REKURZÍV ÉS REKURZÍVAN FELSOROLHATÓ NYELVEK. ELDÖNTHETETLEN PROBLÉMÁK. NEVEZETES IDŐ- ÉS TÁRBONYOLULTSÁGI OSZTÁLYOK: P, NP, PSPACE. NP-TELJES PROBLÉMÁK.

## 13. Alapvető algoritmusok és adatszerkezetek

...

## 14. Haladó algoritmusok

...

## 15. Operációs rendszerek

FOLYAMATOK MEGVALÓSÍTÁSA, ÜTEMEZŐ ALGORITMUSAIK. PÁRHUZAMOSSÁG, KRITIKUS SZEKCIÓK, KÖLCSÖNÖS KIZÁRÁS MEGVALÓSÍTÁSA. SZEMAFOROK, OSZTOTT MEMÓRIA, ÜZENETKÜLDÉS. BE- ÉS KIMENETI ESZKÖZÖK ÜTEMEZÉSI LEHETŐSÉGEI, HOLTPONTOK. MEMÓRIAKEZELÉS, VIRTUÁLIS MEMÓRIA FOGALMA. LAPOZÁS ÉS SZEGMENTÁLÁS. LAPCSERÉLÉSI ALGORITMUSOK. LEMEZTERÜLET-SZERVEZÉS, REDUNDÁNS TÖMBÖK, FÁJLRENDSZEREK SZOLGÁLTATÁSAI ÉS JELLEMZŐ MEGVALÓSÍTÁSAIK.

## 16. Számítógépes hálózatok és Internet eszközök

FIZIKAI RÉTEG, ADATKAPCSOLATI RÉTEG, HÁLÓZATI RÉTEG, SZÁLLÍTÓI RÉTEG - FELADATOK, MÓDSZEREK, PROTOKOLLOK.

---

### 16.1. Rétegek

A hálózati protokollokat a megvalósított szolgáltatásaik alapján rétegekbe csoportosítjuk. Minden réteg csak az alatta lévő rétegek nyújtotta szolgáltatásait használhatja fel. A legerjedtebb hálózati modellek az **OSI** modell (*Open Systems Interconnection Reference Model*) és a **TCP/IP** modell.

OSI modell	TCP/IP modell
Alkalmazási réteg	Felhasználói réteg
Megjelenítési réteg	
Viszony réteg	
Szállítói réteg	Szállítói réteg
Hálózati réteg	Hálózati réteg
Adakapcsolati réteg	Adatkapcsolati réteg
Fizikai réteg	

## 16.2. Fizikai réteg

### 16.2.1. Feladata

A fizikai réteg mechanikus, elektronikus, funkcionális és procedurális tulajdonságokat definiál egy fizikai kapcsolat felépítéséhez, fenntartásához és befejezéséhez. Az adatokat jelekké alakítja, a jeleket pedig továbbítja.

### 16.2.2. NRZ-L

A legegyszerűbb bitátvitel a feszültség kapcsolása.

- Bit 1 = áram bekapcsolva
- Bit 0 = áram kikapcsolva

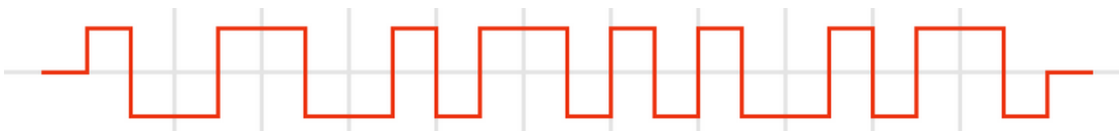


Bit-ek helyett szimbólumokat is küldhetünk, például:  $A = 00, B = 01, \dots$  Ebben az esetben a **Baud** mértékegység adja meg, hogy másodpercenként hány szimbólum kerül átvitelre.

### 16.2.3. Manchester kód

A Manchester kód önütemező, a szinkronizáláshoz szükséges információt tartalmazza a szignál.

- 1 = magasról alacsonyra váltás az intervallum közepén
- 0 = alacsonyról magasra váltás az intervallum közepén



### 16.2.4. Alapsáv és szélessáv

#### Alapsáv

- A jel közvetlenül árammá vagy feszültségváltozássá alakul
- A szignál minden frekvenciával átvitelre kerül
- Probléma: átviteli korlátok

#### Szélessáv

- Az adatok egy széles frekvencia-tartományban kerülnek átvitelre
- Az adatokat vivőhullámra tehetjük (Amplitúdó moduláció (pl. szinusz-görbe))
- Vivőhullámokat megváltoztathatjuk (Frekvencia/fázis moduláció)
- Több vivőhullámot egyszerre használhatunk

### 16.2.5. Médiumok

*Vezetékes átvitel:* UTP, koaxkábel, optikai kábel

*Vezeték nélküli átvitel:* rádióhullám, mikrohullám, infravörös

A médium többszörös használatával növelhetjük az adatátviteli sebességet és az egy területre jutó felhasználók számát. Például: FDMA, TDMA, SDMA, CDMA

## 16.3. Adatkapcsolati réteg

### 16.3.1. Feladata

Az adatkapcsolati réteg hibamentes átvitelt biztosít strukturált adatok továbbításával, így kiszolgálva a hálózati réteget.

- Adatcsomagok keretekre bontása
- hibafelügyelet
- folyamfelügyelet

### 16.3.2. Keretezés

A fizikai réteg bitáramát darabokra, **keretekre** osztjuk.

Lehetséges megoldások a kerethatárok megállapítására:

1. A keret hossza a fejlécben található. Probléma, ha hibásan kerül átvitelre a fejléc. Ekkor a fogadó elveszti az ütemet, és új, értelmetlen kereteket interpretál
2. Fejléc és lezáró: Jelzi a keret kezdetét és végét.
  - *Bytebeszúrás:* speciális Flag Byte-ok jelzik a keret kezdetét és végét. Ha a flag-byte a küldendő adatok között előfordul, akkor egy speciális jellel (Escape) kell jelezni. Ha az is, akkor azt is.
  - *Bitbeszúrás:* Minden keret egy speciális bitmintával (01111110) kezdődik, majd szükség esetén öt db 1-es után beszúrunk egy 0-t, így elérve, hogy a speciális bitminta csak a keret elején fordulhasson elő.
3. Keretezés kódmegsértéssel: Fizikai rétegbeli kódolás megsértése. Pl.: Manchester kódolás csak magas/mély, mély/magas átmenet. Hozzáadjuk a mély/mély-t magas/magas-t

### 16.3.3. Hibafelügyelet

Meg kell állapítani, hogy a keret hibásan vagy helyesen érkezett-e meg. Feladatai a hibák felismerése egy keretben és a hibák javítása egy keretben.

A *redundancia* előfeltétele a hibafelügyeletnek, enélkül egy  $m$  hosszú keret  $2^m$  féle adatot reprezentálhat, egy hibás bit egy új adattartalmat eredményez. Az alapötlet, hogy bizonyos  $m$  bit hosszú adatok nem megengedettek. Ekkor  $2^m$  adat ábrázolásához több mint  $2^m$  lehetséges keret szükséges, vagyis több mint  $m$  bit szükséges egy keretben. Keret hossza tehát  $n > m$ .  $r = m - n$  a redundáns bitek száma.

Egy egyszerű szabály egy redundáns bit hozzáadásához ( $n = m + 1$ ) a paritás. Ez lehet

- *Páratlan bit:* 0-t fűzünk be, ha az 1-es bitek száma páratlan, egyébként 1-est.

- *Páros bit*: 0-t fűzünk be, ha az 1-es bitek száma páros, egyébként 1-est.

**16.3.4 Definíció (Hamming-távolság).** Legyen  $x = x_1, \dots, x_n$ ,  $y = y_1, \dots, y_n$  két üzenet. Az  $x$ ,  $y$  Hamming távolsága  $d(x,y)$  = az 1-es bitek száma ( $x$  XOR  $y$ )-ban.

A Hamming távolság egy metrika:

- nem-negatív,
- idempotens,
- szimmetrikus,
- háromszögegyenlőtlenség.

**16.3.5 Definíció (Üzenethalmaz Hamming távolsága).** Legyen  $S$  (egyenlő hosszú) bit-sztringek halmaza.  $S$  Hamming távolsága

$$d(S) = \min_{x,y \in S, x \neq y} d(x,y).$$

A Hamming távolsággal felismerhető  $d$  bit hiba, ha a megengedett keretek halmazában legalább  $d + 1$  Hamming távolság van.

A Hamming távolsággal  $d$  bit hiba javítható, ha a megengedett keretek halmazában legalább  $2d + 1$  Hamming távolság van.

Hatékony hibafelismerés a **CRC** (Cyclic Redundancy Check), polinom aritmetikán alapul a 2-es maradékosztályok testén ( $\mathbb{Z}_2$ ). A CRC működése:

- $G(x)$  generátorpolinom definiálása, foka  $g$
- $G(x)$  ismert a fogadó és küldő által
- $g$  redundáns bit generálása
- Keret  $M$ , mint  $M(x)$  polinom
- Küldő kiszámolja az osztás maradékát:  $r(x) = x^g M(x) \bmod G(x)$  és elküldi a  $T(x) = x^g M(x) + r(x)$ -t
- Fogadó  $m(x)$ -et fogad, kiszámolja a maradékot:  $m(x) \bmod G(x)$
- Ha nem történt hiba,  $T(x)$  fogadása korrekt,  $m(x) = T(x)$
- Bithiba esetén fogadóhoz  $m(x) = T(x) + E(x)$  érkezik, ahol  $E(x)$  hibapolinom
- Bithibák:  $m(x) \bmod G(x) = (T(x) + E(x)) \bmod G(x) = T(x) \bmod G(x) + E(x) \bmod G(x)$ .

### 16.3.6. Folyamfelügyelet

A küldő és a fogadó adatátviteli sebességének egymáshoz igazítása. Módszerek a szimplex protokoll és a csúszó ablak protokoll.

### 16.3.7. Médiumhozzáférés

A **MAC** (Medium Access control) alréteg csatorna megosztására szolgál a kommunikáció résztvevői között. Ez megoldható statikus multiplexálással és dinamikus csatorna foglalással. Előbbi akkor jó megoldás, ha fix adatráták vannak, ám a számítógépek esetén löketszerűen érkező adatokkal dolgoznak, így a csatornának elegendően nagy kapacitásúnak kell lenni, hogy a csúcsrátát kezelni tudja. Ez pazarlás

Lehetséges MAC protokollok:

- Kollízió alapú protokollok
- Versenymentes protokollok
- Protokollok korlátozott versennyel

Az **ALOHA** módszer kollízió alapú protokoll. Algoritmus: amikor egy csomag kész, azonnal átvitelre kerül, ezekre nyugtákat küldünk, ha nem érkezik meg, újraküldjük.

A **CSMA** is kollízió alapú protokoll. Három változata létezik:

- *1 persistent*: Ha a csatorna nem szabad, várunk amíg az lesz, és ekkor azonnal megkezdődik az átvitel, kollízió esetén véletlenszerű ideig várunk
- *non-persistent*: Ha a csatorna nem szabad, véletlen szerűen választott ideig várunk, majd megvizsgáljuk megint
- *p-persistent*: Ha a csatorna szabad,  $p$  valószínűséggel küldjük a csomagot, kollízió esetén véletlen ideig várunk, majd kezdjük előlről. Ha a csatorna foglalt, folyamatosan várunk, míg szabad nem lesz.

## 16.4. Hálózati réteg

### 16.4.1. Feladata

Az útvonal információk felépítése, a csomagok továbbítása/irányítása.

### 16.4.2. Internet Protokoll

Az **IP** (Internet Protokoll) egy hálózati réteg protokoll.

- Routing-táblákat használ (tartalmazza a cél címekhez a következő számítógép címét a hozzá vezető úton, a cél meghatározhat egy számítógépet vagy egy egész sub-net-et, tartalmaz egy default gateway-t)
- Packet forwarding-al dolgozik (az IP csomag tartalmazza a küldő IP címét és a cél IP címét, ha egy csomag megérkezik egy routerhez, és a cél IP = a saját IP-vel, akkor a csomagot kiszállítja, ha a cél a routing táblában van, akkor továbbítja a megadott gateway-hez, egyébként továbbítja a default-gateway-nek).
- Csomag kezelése: TTL (Time-to-Live) hop-ok számát tartalmazza. A routerben:  $TTL := TTL - 1$ , ha  $TTL \neq 0$ , packet forwarding a routing-tábla alapján, ha  $TTL = 0$ , akkor töröljük, feladónak *expired* üzenet.

### 16.4.3. Statikus routing

A routing táblát manuálisan építjük fel, ez kis és statikus LAN-ok esetén használandó.



#### 16.4.4. Dinamikus routing

A routing tabla felépítése automatizált. Centralizált algoritmus a Link State, ebben az esetben minden állomásnak ismernie kell minden információt. Decentrális algoritmus a Distance Vector: minden router lokálisan dolgozik, csak a szomszédjával kommunikál. A Bellman-Ford algoritmus elosztott változatát használja, azaz minden csomópont csak a direkt szomszédjaival kommunikál

#### 16.4.5. DHCP

A **DHCP** (Dynamic Host Configuration Protocol) dinamikusan oszt IP címet a számítógépeknek konfiguráció nélkül. A számítógép kér egy IP címet a DHCP szervertől, az automatikusan hozzárendeli azt, majd ha a számítógép elhagyja a hálózatot, az IP cím újra kiosztható.

### 16.5. Szállítói réteg

#### 16.5.1. Feladata

- Hibafelügyelet
- Torlódás felügyelet

#### 16.5.2. TCP

A **TCP** (Transport Control Protocol) egy kapcsolatorientált megbízható szolgáltatás byte-folyamokhoz két végpont között. A felhasználói réteg adatáramát csomagokra bontja, amik fogadásáról a másik oldal nyugtákat küld. Routing nincs, végpont-végpont protokoll.

Felépítése: Egy résztvervő socket által azonosított: IP cím és port, egy TCP kapcsolat egy socketpár által azonosított. Az adatcsomagok megérkezését nyugtázza (ACK), a nem nyugtázott csomagokat újraküldi. A kapcsolat indítására SYN flaget használ, amelyben a maximum szegmens-méret is átvitelre kerül. Lezáráskor a küldő FIN flag-et küld, amit a fogadó nyugtáz (ACK). Mindkét irányban le kell zárni a kapcsolatot.

#### 16.5.3. UDP

A **UDP** (User Datagram Protocol) egy nem megbízható, kapcsolat nélküli szállítói réteg-protokoll csomagoknak. A nyugták hiánya miatt akkor használják, ha a gyorsaság fontosabb a megbízhatóságnál, például hangátvitel és egyéb valós idejű multimédia átvitelek.

## 17. Osztott rendszerek

FELADATOK SPECIFIKÁCIÓJA BIZTONSÁGI ÉS HALADÁSI FELTÉTELEKKEL, ABSZTRAKT PÁRHUZAMOS PROGRAM TULAJDONSÁGAI, MEGOLDÁS FOGALMA, NEVEZETES FELADATOK MEGOLDÁSA PÁRHUZAMOS ÉS ELOSZTOTT PROGRAMOKKAL

## 18. Adatbázisok tervezése és lekérdezése

RELÁCIÓS ADATMODELL, EGYED-KAPCSOLAT MODELL ÉS ÁTALAKÍTÁSA RELÁCIÓS ADATMODELLBE. RELÁCIÓS ALGEBRA, SQL. AZ SQL PROCEDURÁLIS KITERJESZTÉSE (PL/SQL VAGY PSM). RELÁCIÓS ADATBÁZIS-SÉMÁK TERVEZÉSE, NORMÁLFORMÁK, DEKOMPOZÍCIÓK.

## 19. Adatbázisok optimalizálása és konkurencia kezelése

AZ ADATBÁZISKEZELŐ RENDSZEREK FELADATA, RÉSZEI. INDEXSTRUKTÚRÁK, LEKÉRDEZÉSEK VÉGREHAJTÁSA, OPTIMALIZÁLÁSI STRATÉGIÁK. TRANZAKCIÓK FELDOLGOZÁSA, NAPLÓZÁS ÉS HELYREÁLLÍTÁS, KONKURENCIA-KEZELÉS.

## 20. Felhasznált irodalom

- Simon P. 2013. Bevezetés az analízisbe I., Budapest
- Gregorics T., Vázlat a Bsc záróvizsga mesterséges intelligenciáról szóló témaköréhez, Budapest