Lineáris egyenletrendszer

Definíció

Az

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ \vdots \qquad\qquad & \quad \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m \end{aligned}$$

alakú egyenletrendszert, ahol az a_{ij} $(i \in \{1, \ldots, m\}, j \in \{1, \ldots, n\})$ és a b_k $(k \in \{1, \ldots, m\})$ valós számok ismertek, $x_1, \ldots, x_n$ ismeretlenek, lineáris egyenletrendszernek nevezzük.

- a_{ij}: az egyenletrendszer együtthatói
- b_k: szabad tagok, vagy konstansok
- az egyenletrendszer alapmátrixa, ill. kibővített mátrixa:

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ a_{21} & \cdots & a_{2n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} \quad \text{és} \quad A|b = \left(\begin{array}{ccc|c} a_{11} & \cdots & a_{1n} & b_1 \\ a_{21} & \cdots & a_{2n} & b_2 \\ \vdots & & \vdots & \vdots \\ a_{m1} & \cdots & a_{mn} & b_m \end{array} \right)$$

A lineáris egyenletrendszer mátrixos alakja: $Ax = b$.

Definíció

A lineáris egyenletrendszer

- megoldható, ha van megoldása, azaz létezik olyan $(x_1, \ldots, x_n)$ vektor, hogy $Ax = b$ fennáll;
  - határozott, ha pontosan 1 megoldása van;
  - határozatlan, ha több megoldása van;
- ellentmondásos, ha nincs megoldása.

Definíció

Egy mátrix rangja alatt a mátrix sorai (vagy oszlopai), mint vektorok által alkotott vektorrendszer rangját értjük. Jelölés: rang(A).

Tétel – rangkritérium

- Egy lineáris egyenletrendszer pontosan akkor oldható meg, ha rang(A) = rang$(A|b)$.
- Ha megoldható és rang$(A) = n$ (ahol n az ismeretlenek száma), akkor határozott, ha rang$(A) < n$, akkor határozatlan.

A lineáris egyenlet mátrixos alakja , egy mátrix rangja, megoldható, határozott, határozatlan, ellentmondásos, Tétel - rangkritérium

Lineáris egyenletrendszerek megoldáshalmaza

Definíció

A lineáris egyenletrendszer homogén, ha $b = 0$, azaz ekkor mátrixos alakja $Ax = 0$. Egyébként a lineáris egyenletrendszer inhomogén.

Megjegyzés: egy homogén lineáris egyenletrendszernek a nullvektor mindig megoldása.

Állítás – homogén lineáris egyenletrendszer megoldásai

Egy homogén lineáris egyenletrendszer összes megoldása alteret alkot $\mathbb{R}^n$-ben, melynek dimenziója $n - \text{rang}(A)$. Ezt az alteret megoldástérnek nevezzük.

Állítás – inhomogén lineáris egyenletrendszer megoldásai

Ha $Ax = b$ megoldható, akkor megoldáshalmaza $x_0 + H$ alakú, ahol

- x_0 a lineáris egyenletrendszer egy rögzített megoldása;
- H a megfelelő homogén lineáris egyenletrendszer (azaz $Ax = 0$) megoldástere.

Lineáris egyenletrendszerek megoldáshalmaza, homogén, inhomogén, homogén lineáris egyenletrendszerek megoldásai, inhomogén egyenletrendszerek megoldásai

Gauss elimináció

Lineáris egyenletrendszerek megoldása

Két módszer:

1. Gauss-elimináció
2. Cramer-szabály: csak akkor használható, ha $n = m$ és A reguláris.

1. Gauss-elimináció: Az alábbi ekvivalens átalakítások nem változtatják meg a lineáris egyenletrendszer megoldáshalmazát:
   - Egyenlet szorzása $\lambda \neq 0$-val. (Egyenlet $\rightsquigarrow$ kibővített mátrix sorai.)
   - Egy egyenlethez hozzáadni egy másik egyenlet λ-szorosát.
   - Egyenletek sorrendjének megváltoztatása.
   - Elhagyni olyan egyenletet, amely egy másik egyenlet λ-szorosa.
   - Ismeretlenek felcserélése együtthatóikkal együtt, minden egyenletben. (Alapmátrixban: oszlopcsere.)

   Ezen átalakítások segítségével az egyenletrendszer kibővített mátrixát (ahol a mátrix sorai felelnek meg 1-1 egyenletnek) trapéz alakúra hozzuk (főátló alatt csupa 0), ahonnan visszahelyettesítéssel adódnak a megoldások.
   - Ha az eljárás közben $(0 \dots 0| \neq 0)$ sor adódik, akkor az egyenletrendszer ellentmondásos.
   - Ha az eljárás végén n sor marad, akkor az egyenletrendszer határozott, ha kevesebb, akkor határozatlan.

Lineáris egyenletrendszerek megoldása, Gauss-eliminánció

A **lineáris egyenletrendszer** olyan többismeretlenes egyenletrendszer, ahol minden ismeretlen elsőfokon (azaz első hatványon) szerepel.

Definíció. Az

$$(1)\qquad \begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ \vdots \qquad\qquad & \quad \vdots \\ a_{k1}x_1 + a_{k2}x_2 + \cdots + a_{kn}x_n &= b_k \end{aligned}$$

egyenletek halmazát, ahol x_j valós számok (j = 1,...,n) az ismeretleneket, a_{ij} adott valós számok (i = 1,2,...,k, j = 1,2,...,n) az ismeretlenek együtthatóit jelentik és a b_i -k adott valós számok, lineáris egyenletrendszernek nevezzük.

Megjegyzés.

(1) Amennyiben $b_1 = b_2 =...= bk_k= 0$, az egyenletrendszert homogénnak, ellenkező esetben inhomogénnak mondjuk.

(2) A $c_1,c_2,...,c_n$ számok az egyenletrendszer megoldását adják, ha az x_i ismeretlenek helyére helyettesítve minden egyenletet kielégítenek.

Az általános lineáris egyenletrendszerek megoldására az egyik legtermészetesebben adódó, egyszerű és gyakorlati szempontból is jól alkalmazható eljárás a Gauss-féle kiküszöbölés, amelynek számos fontos elméleti következménye is van. Speciális egyenletrendszerekre vonatkozik a Cramer-szabály, amely a determinánsok segítségével ad képletet a megoldásra. A jelen fejezetben vezetjük be a lineáris függetlenséget és a mátrix rangját is, amelyek a későbbiekben is alapvető szerepet játszanak. Mindezek egyik alkalmazásaként visszatérünk a négyzetes mátrixok körében az invertálhatóság és a nullosztók kérdésére.

Gauss - kiküszöbölés

Egy *k* egyenletből álló *n* ismeretlenes *lineáris egyenletrendszer* általános alakja

$$\begin{aligned} &\alpha_{11}x_1 + \alpha_{12}x_2 + \ldots + \alpha_{1n}x_n = \beta_1 \\ &\alpha_{21}x_1 + \alpha_{22}x_2 + \ldots + \alpha_{2n}x_n = \beta_2 \\ &\quad\vdots \\ &\alpha_{k1}x_1 + \alpha_{k2}x_2 + \ldots + \alpha_{kn}x_n = \beta_k \end{aligned}$$

ahol az α_{ij} együtthatók és a β_i konstansok egy *T* kommutatív test elemei. Az egyenletek száma (*k*) és az ismeretlenek száma (*n*) egymástól függetlenül is tetszőleges lehet (tehát pl. semmiképpen sem szorítkozunk csak a *k*=*n* esetre).
Az egyenletrendszer egy *megoldás*án *T*-beli elemek egy olyan $\gamma_1,\ldots,\gamma_n$ sorozatát értjük, amelyeket a megfelelő x_i-k helyére beírva, valamennyi egyenletben egyenlőség teljesül.
Van olyan egyenletrendszer, amelynek nincs megoldása, van, amelyik egyértelműen oldható meg (azaz pontosan egy megoldása van) és van olyan, amelyet (egynél) több megoldás is kielégít. (Ez utóbbi esetben elég óvatosan fogalmaztunk, annak ellenére, hogy például valós számokra vonatkozó egyenletrendszereknél megszoktuk, hogy egynél több megoldás esetén a megoldásszám végtelen. Látni fogjuk, hogy végtelen test esetén ez valóban mindig így van. Azonban véges, mondjuk *t* elemű test esetén az összes szóba jövő $x_1,\ldots,x_n$-re is csak t^n lehetőségünk van, tehát eleve nem lehet végtelen sok megoldás.)
Az alábbi kérdésekre keressük a választ: (a) mi a feltétele annak, hogy egy egyenletrendszer megoldható legyen; (b) (megoldhatóság esetén) hány megoldás van; (c) hogyan lehet az összes megoldást áttekinteni; (d) milyen módszerrel juthatunk el (egy vagy az összes) megoldáshoz

Ebben a pontban a fenti kérdésekre a *Gauss-féle kiküszöbölés* (röviden *Gauss-kiküszöbölés* vagy latinosan *Gauss-elimináció*) segítségével adjuk meg a választ.
Az eljárás során az alábbi lépéseket fogjuk végezni, amelyek valamennyien az eredetivel ekvivalens egyenletrendszerekhez vezetnek (azaz olyanokhoz,

amelyeknek pontosan ugyanazok a megoldásai, mint az eredetinek):

E1. Valamelyik egyenletet egy nullától különböző *T*-beli elemmel (a továbiakban: *skalár*ral) végigszorozzuk.
E2. Valamelyik egyenlethez egy másik egyenlet skalárszorosát hozzáadjuk.

E3. Két egyenletet felcserélünk.

E4. Az olyan egyenleteket, ahol valamennyi együttható és minden jobb oldali konstans is 0, elhagyjuk.

Ezeket a lépéseket *elemi ekvivalens átalakítások*nak nevezzük.
Az elemi ekvivalens átalakítások segítségével az egyenletrendszerből az alább részletezett módon egymás után ki fogjuk *küszöbölni* az ismeretleneket.
Tegyük fel, hogy $\alpha_{11}\neq 0$. Az első egyenletet osszuk végig α_{11}-gyel (azaz alkalmazzuk E1-et az α_{11} reciprokával), majd minden $i>1$-re az i-edik egyenletből vonjuk ki az első egyenlet α_{i1}-szeresét. Ezzel a többi egyenletből *kiküszöböl*tük x_1-et.
Tegyük fel, hogy az így kapott egyenletrendszerben az új $\alpha_{22}\neq 0$. Ekkor az előző eljárást megismételhetjük: a második egyenletet végigosztjuk α_{22}-vel, majd minden $i>2$-re az i-edik egyenletből kivonjuk a második egyenlet α_{i2}-szeresét stb.
Ha valamikor megakadtunk, pl. az előbb $\alpha_{22}=0$ volt, de mondjuk $\alpha_{52}\neq 0$, akkor a második és az ötödik egyenletet felcseréljük, és így haladunk tovább.

Ha ez sem megy, azaz minden $i\geq 2$ esetén $\alpha_{i2}=0$, akkor a harmadik ismeretlenre térünk át, vagyis α_{23}-at vizsgáljuk stb.
Nemsokára néhány konkrét példán keresztül illusztráljuk, hogyan fest mindez a gyakorlatban és hogyan juthatunk el így az egyenletrendszer megoldásához. Előtte azonban érdemes némi technikai egyszerűsítést bevezetni.

Vegyük észre, hogy a fenti lépések nyomon követéséhez elég csak az együtthatók és a jobb oldali konstansok változását figyelni, az x_i,+ és = „jeleket” fölösleges mindig újra leírni. Ezért az egyenletrendszert egyszerűbben jellemezhetjük mátrixok segítségével: az α_{ij} együtthatókból képezett $k\times n$-es A mátrixot az egyenletrendszer *együtthatómátrix*ának nevezzük, a jobb oldali konstansokkal kibővített $k\times(n+1)$-es mátrixot pedig az egyenletrendszer *kibővített mátrix*ának nevezzük

és $A|\underline{b}$-vel jelöljük, azaz

$$A=\begin{pmatrix}\alpha_{11} & \alpha_{12} & \dots & \alpha_{1n}\\ \alpha_{21} & \alpha_{22} & \dots & \alpha_{2n}\\ \vdots & & & \\ \alpha_{k1} & \alpha_{k2} & \dots & \alpha_{kn}\end{pmatrix},\quad A|\underline{b}=\left(\begin{array}{cccc|c}\alpha_{11} & \alpha_{12} & \dots & \alpha_{1n} & \beta_1\\ \alpha_{21} & \alpha_{22} & \dots & \alpha_{2n} & \beta_2\\ \vdots & & & & \\ \alpha_{k1} & \alpha_{k2} & \dots & \alpha_{kn} & \beta_k\end{array}\right)$$

A kibővített mátrixban az együtthatók alkotta rész és a jobb oldali konstansok közé iktatott függőleges vonallal jelezzük, hogy a kétféle típusú elemek eltérő szerepet játszanak az egyenletrendszerben.
(Az $A|\underline{b}$ felírásnál $\underline{b}$ a jobb oldalon álló β_i konstansokból képezett „vektort” jelöli, erről bővebben ennek a pontnak a végén lesz szó.)
Azonnal adódik, hogy az egyenletekkel végzett E1–E4 elemi ekvivalens átalakításoknak a kibővített mátrixnál a sorokkal végzett hasonló változtatások felelnek meg:

M1. Valamelyik sort egy nullától különböző skalárral végigszorozzuk.

M2. Valamelyik sorhoz egy másik sor skalárszorosát hozzáadjuk.

M3. Két sort felcserélünk.

M4. A csupa 0-ból álló sorokat elhagyjuk.

A kibővített mátrixon végzett fenti lépéseket ***elemi sorekvivalens átalakítások***nak nevezzük.
(Az ekvivalens lépések „visszacsinálhatósága" érdekében formailag teljesebb, ha E4-nél, illetve M4-nél az ilyen egyenletek, illetve sorok hozzávételét is megengedjük, de ennek gyakorlati alkalmazására nyilván sosincs szükség.)

Most három, valós számokra vonatkozó egyenletrendszeren mutatjuk be a kiküszöbölési eljárást.

P1 példa:

$$x_1 + 2x_2 = 3$$

$$4x_1 + 5x_2 = 6$$

$$7x_1 + 8x_2 = 9$$

Ennek kibővített mátrixa $\left(\begin{array}{cc|c} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{array}\right)$ A jelzett kiküszöbölési eljárásnak megfelelően vonjuk ki a második sorból az első sor 4-szeresét, a harmadik sorból pedig az első sor 7-szeresét. Így az $\left(\begin{array}{cc|c} 1 & 2 & 3 \\ 0 & -3 & -6 \\ 0 & -6 & -12 \end{array}\right)$ mátrixhoz jutunk. Osszuk el a második sort –3-mal, majd adjuk hozzá ennek 6-szorosát a harmadik sorhoz. Az így kapott mátrix $\left(\begin{array}{cc|c} 1 & 2 & 3 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \end{array}\right)$ Itt a csupa nulla sor elhagyható, tehát

szorosát a harmadik sorhoz. Az így kapott mátrix Itt a csupa nulla sor elhagyható, tehát

marad az $\left(\begin{array}{cc|c} 1 & 2 & 3 \\ 0 & 1 & 2 \end{array}\right)$ mátrix. Itt a második sorból azonnal leolvasható, hogy $x_2=2$ (hiszen x_2 „ki van fejezve"). Ezt visszahelyettesíthetjük az első sornak megfelelő egyenletbe: $x_1+2x_2=x_1+2\cdot 2=3$, tehát $x_1=-1$. Azonban ez a lépés is „automatizálható". Ha a legutolsó mátrixnál az első sor második elemét kiejtjük, akkor x_1 is „ki lesz fejezve". Vonjuk ki ezért az első sorból a második sor 2-szeresét, ekkor az $\left(\begin{array}{cc|c} 1 & 0 & -1 \\ 0 & 1 & 2 \end{array}\right)$ mátrixot kapjuk, ahonnan valóban $x_1=-1$ is közvetlenül leolvasható.
Az egyenletrendszernek tehát egyetlen megoldása van: $x_1=-1$, $x_2=2$.

P2 példa:

$$x_1 + x_2 + 2x_3 = 3$$

$$4x_1 + 4x_2 + 5x_3 = 6$$

$$7x_1 + 7x_2 + 8x_3 = 10$$

A kiküszöbölés során a kibővített mátrix a következőképpen változik:

$$\left(\begin{array}{ccc|c} 1 & 1 & 2 & 3 \\ 4 & 4 & 5 & 6 \\ 7 & 7 & 8 & 10 \end{array}\right) \sim \left(\begin{array}{ccc|c} 1 & 1 & 2 & 3 \\ 0 & 0 & -3 & -6 \\ 0 & 0 & -6 & -11 \end{array}\right) \sim \left(\begin{array}{ccc|c} 1 & 1 & 2 & 3 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{array}\right)$$

Mivel az utolsó mátrix harmadik sora a lehetetlen $0x_1+0x_2+0x_3=1$ egyenletnek felel meg, ezért ennek az egyenletrendszernek nincs megoldása.

3.1.1 Tétel

I. Egy lineáris egyenletrendszer kibővített mátrixa elemi sorekvivalens átalakításokkal redukált lépcsős alakra hozható.

II. Az egyenletrendszer akkor és csak akkor oldható meg, ha a (redukált) lépcsős alakban nincs tilos sor.

III. Az egyenletrendszernek akkor és csak akkor egyértelmű a megoldása, ha (nincs tilos sor és) a vezéregyesek száma megegyezik az ismeretlenek számával.

IV. Ha több megoldás van, akkor a vezéregyest nem tartalmazó oszlopoknak megfelelő ismeretlenek szabad paraméterek (tetszőlegesen megválaszthatók), a többi ismeretlen pedig ezekkel egyértelműen kifejezhető. A megoldásszám ekkor végtelen test esetén végtelen, t elemű test esetén pedig t^s, ahol s a szabad paraméterek száma, és a(z összes) megoldás közvetlenül leolvasható a redukált lépcsős alakból.

3.1.2 Tétel

Ha egy k egyenletből álló n ismeretlenes lineáris egyenletrendszernek egyetlen megoldása van, akkor $n \leq k$. ❶
Bizonyítás: Egyértelmű megoldás esetén az RLA-ban a vezéregyesek száma n, másrészt a vezéregyesek különböző sorokban helyezkednek el, tehát számuk legfeljebb k. Innen valóban $n \leq k$. ❷
Ennek az észrevételnek egy egyszerű, de fontos következményét a későbbiekben sokszor fel fogjuk használni. Ehhez előbb bevezetjük a homogén lineáris egyenletrendszer fogalmát:

3.1.3 Definíció

Egy lineáris egyenletrendszert *homogén*nek nevezünk, ha a jobb oldali konstansok mindegyike nulla. ❶
Egy homogén egyenletrendszer biztosan megoldható, hiszen $x_1=\ldots=x_n=0$ mindig megoldás. Ezt *triviális megoldás*nak nevezzük. Így itt az az érdekes kérdés, hogy mikor létezik nemtriviális megoldás. Erre elégséges feltételt ad az alábbi

3.1.4 Tétel

Ha egy homogén lineáris egyenletrendszerben az ismeretlenek száma nagyobb, mint az egyenletek száma, akkor az egyenletrendszernek biztosan létezik nemtriviális megoldása. ❶

Bizonyítás: Indirekt, tegyük fel, hogy a triviálison kívül nincs más megoldás. Ekkor az egyenletrendszernek egyetlen megoldása van, tehát a 3.1.2 Tétel szerint az ismeretlenek száma nem lehet nagyobb az egyenletek számánál, ami ellentmond a feltételnek. ❷
A továbbiak előkészületeként az egyenletrendszerek két másik felírási módjával ismerkedünk meg. Ehhez szükségünk lesz az (oszlop)vektorok fogalmára.

3.1.5 Definíció

Az egy oszlopból álló mátrixokat *oszlopvektor*oknak nevezzük. Egy ilyen mátrix (egyetlen oszlopának) elemeit a vektor *komponensei*nek vagy *koordinátái*nak hívjuk. A T test elemeiből képzett q komponensű vektorok összességét ($T^{q\times 1}$ helyett röviden) T^q-val jelöljük. ❶
Ez a fogalom a sík-, illetve térvektorok (valós) számpárokként, illetve számhármasokként történő felírási módjának az általánosítása. Később még sokkal általánosabb értelemben fogjuk használni a „vektor" szót (lásd a 4.1 pontot).

T^q-ban — az általános mátrixműveleteknek megfelelően — beszélhetünk két vektor összegéről, illetve egy vektor skalárszorosáról (azaz T-beli elemmel vett szorzatáról), ezeket úgy kapjuk, hogy a megfelelő komponenseket összeadjuk, illetve a komponenseket a skalárral végigszorozzuk.
A vektorokat aláhúzott latin kisbetűkkel fogjuk jelölni.

Most rátérünk az egyenletrendszer egyik átírási módjára. Legyen

$$A=\begin{pmatrix}\alpha_{11} & \alpha_{12} & \cdots & \alpha_{1n}\\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2n}\\ \vdots & & & \\ \alpha_{k1} & \alpha_{k2} & \cdots & \alpha_{kn}\end{pmatrix}\in T^{k\times n}$$

$$\underline{x}=\begin{pmatrix}x_1\\ x_2\\ \vdots\\ x_n\end{pmatrix}\in T^n, \qquad \underline{b}=\begin{pmatrix}\beta_1\\ \beta_2\\ \vdots\\ \beta_k\end{pmatrix}\in T^k$$

azaz A az együtthatómátrix, x−−x_ az ismeretlenekből képezett vektor és a (már említett) b−b_ a jobb oldali konstansokból álló vektor. Ekkor a mátrixszorzás definíciójának megfelelően az egyenletrendszer felírható $A\underline{x}=\underline{b}$ alakban.
A másik átírási módhoz legyen

$$\underline{a}_j=\begin{pmatrix}\alpha_{1j}\\ \alpha_{2j}\\ \vdots\\ \alpha_{kj}\end{pmatrix}\in T^k, \quad j=1,2,\dots,n,$$

tehát az $\underline{a}_j$-k az A együtthatómátrix oszlopvektorai. Ekkor az egyenletrendszer a következőképpen írható fel: $x_1\underline{a}_1+x_2\underline{a}_2+\dots+x_n\underline{a}_n=\underline{b}$

Tétel. A mondatszerkezetű nyelvek osztálya egybeesik a Turing gépek által elfogadott nyelvek osztályával.

Amennyiben a mozgásfüggvény képhalmaza a $Q\times V\times\{$*Bal*, *Jobb*, *Helyben*$\}$ halmaz (és nem annak részhalmazainak halmaza), akkor determinisztikus Turing gépről beszélünk, és érvényes a következő tétel.

Tétel.

A determinisztikus Turing gépek által elfogadott nyelvek osztálya egybeesik a nemdeterminisztikus Turing gépek által elfogadott nyelvek osztályával.

A Turing-gépeknek több változata is ismert, most ezek közül mutatunk be néhányat.

Van olyan definíció, ahol a fej a **{***Bal*, *Jobb***}** irányokba léphet, és nem maradhat helyben. Belátható, hogy egy ilyen Turing-gép szimulálni tudja az eddigiekben ismertetett változatnak a fejet helyben

Determinisztikus Turing-gépek, lineárisan korlátozott automaták, eldönthetetlen problémák, tár és idő korlátok. Nemdeterminisztikus Turing-gépek, nevezetes nyelvosztályok, P, NP.

Determinisztikus Turing-gépek

Determinisztikus Turing-gép

1. Definíció. *Legyen* $k \geq 1$ *egy egész szám. Egy* k-szalagos Turing-gépet *egy* $M = (Q, \Sigma, \Gamma, q_0, *, F, \delta)$ *hetes ír le, ahol:*

- Q *egy véges, nem üres halmaz, a gép* állapot*ainak halmaza*
- Σ *egy véges, nem üres halmaz, a bemeneti* ábécé
- Γ *egy véges, nem üres halmaz, a* szalagábécé, $\Sigma \subset \Gamma$
- $q_0 \in Q$ *a* kezdő állapot
- $* \in \Gamma \setminus \Sigma$, *a szalagon az* üres jel
- $F \subseteq Q$ *az* elfogadó állapotok *halmaza*
- δ *az* átmeneti függvény,

$$\delta : (q, a_1, a_2, \cdots, a_k) \to (q', b_1, b_2, \cdots, b_k, D_1, D_2, \cdots, D_k),$$

ahol $q, q' \in Q$, $a_i, b_i \in \Gamma$ *és* $D_i \in \{B, J, H\}$ *(azaz* ***B****alra,* ***J****obbra vagy* ***H****elyben).*

Lineárisan korlátozott automaták

A következőkben egy másik, a szakirodalom által jól ismert olyan automata modellt mutatunk be, amely pontosan a környezetfüggő nyelveket fogadja el. A lineárisan korlátolt automatának (Linear Bounded Automaton (LBA), vagy lineárisan korlátozott Turing-gép) több változata is ismert, ezek közül ismertetünk egyet. Az automata egy véges vezérlővel rendelkezik és egy szalaggal, amelyen kezdetben az input szó áll. Az automata a működése során két lényeges eltérést mutat az eddig tárgyalt automatákhoz képest: az egyik, hogy a szalagon a fej előre és hátra is mozoghat, a másik, még lényegesebb, hogy nem csak olvashatja a szalagot, de annak tartalmát át is írhatja, ennek megfelelően nem olvasó, hanem író-olvasó fejről beszélünk.

Eldönthetetlen problémák

Eldönthetetlen, hogy egy M Turing-gép üres / véges / reguláris / környezetfüggetlen nyelvet ismer-e fel.

Már említettük, hogy eldöntési probléma egy olyan feladat, amelyre igen-nem válasz adható. Egy ilyen problémát eldönthetetlennek (vagy megoldhatatlannak) nevezünk, ha

Turinggéppel nem oldható meg. Pontosabban ez azt jelenti, hogy a probléma igaz példányait kódoló nyelv nem rekurzív. A kódolást általában nem adjuk meg pontosan, lényegében minden intuitíven algoritmikus kódolás megfelelő. A Church-Turing tézis szerint az, hogy egy probléma eldönthetetlen azt jelenti, hogy algoritmikusan megoldhatatlan.

Tár és idő korlátok

Idő- és tárbonyolultsági osztályok

Legyen $f(n): \mathbb{N} \to \mathbb{N}$függvény.

$TIME(f(n)) = \{L|L \text{ eldönthető } \mathcal{O}(f(n)) \text{ időigényű Turing} - \text{géppel}\}$

$P = \bigcup_{k>1} TIME(n^k)$

$NTIME(f(n)) = \{L|L \text{ eldönthető } \mathcal{O}(f(n)) \text{ időigényű nemdeterminisztikus Turing} - \text{géppel}\}$

$NP = \bigcup_{k>1} NTIME(n^k)$

Tipikus P-beli probléma az úgynevezett *elérhetőség* probléma, melynek bemenete egy G gráf, illetve két kitüntetett pontja. Válaszként megmondja van e út a két pont között.

Az NP-beli problémák közös tulajdonsága, hogy egy probléma egy példányának ellenőrzése polinom időben elvégezhető. Ennek megfelelően a nemdeterminisztikus Turing-gép általában „megsejt" egy megoldást és leellenőrzi azt.

A definíciókból következik, hogy $P \subseteq NP$. Sejtés, hogy a tartalmazás valódi.

A Turing-gép egyik előnye, hogy segítségével egyszerűen vizsgálhatjuk a számítások idő- és tárigényét. Az M Turing-gép **számolási ideje** az s inputon a megállásig végrehajtott lépések száma, **tárigénye** pedig a felhasznált szalagcellák száma.

Jelölje $T_M(n)$ az M gép maximális számolási idejét az n jelből álló bemeneteken. Az n hosszú szavakon a maximális tárigényt $S_M(n)$-nel jelöljük.

Ha van olyan n jelből álló $s \in I^*$ szó, amellyel elindítva M nem áll meg véges sok lépés után, akkor $T_M(n)$ értékét végtelennek tekintjük. Hasonló mondható az S_M függvényről is.

Van olyan L nyelv, amelyre igazak az alábbiak:

1. Az L felismerhető egy olyan M Turing-géppel, melyre $T_M(n)$ véges minden n-re.
2. Tetszőleges, az L-et felismerő N Turing-géphez van olyan N' Turing-gép, amelyre $L = L_{N'}$ szintén teljesül, továbbá $T_{N'}(n) = O(\log T_N(n))$.

Nemdeterminisztikus Turing-gépek

Nemdeterminisztikus Turing-gép

A nemdeterminisztikusság már ismert fogalom, itt is hasonlóan működik. Nemdeterminisztikus Turing-gép esetén az átmeneti függvény értéke egy halmaz, a gép akkor fogad el egy szót, ha van olyan számítási út, amelyik elfogadó állapotban ér véget.

Azt azért érdemes kiemelni, hogy egy bementhez tartozó számítási fának itt lehetnek véges és végtelen hosszú útjai is.

A véges automatákhoz hasonlóan itt is meg lehet szabadulni a nemdeterminizmustól,

3. Tétel. *Minden nemdeterminisztikus Turing-gép szimulálható determinisztikus Turing-géppel.*

Bizonyítás vázlat: Legyen M a nemdeterminisztikus Turing-gép, amihez egy M' determinisztikusat akarunk megadni. Az ötlet az, hogy M' egy tetszőleges x bemenetre az M számítási fáján egy szélességi bejárást végez (pontosabban föntről lefelé haladva generálja ezt a számítási fát). Ha talál egy elfogadó levelet (azaz, ahol M elfogadó állapotban elakadna), akkor M' megáll elfogadó állapotban. Ha a bejárás véget ér és nem talált ilyen levelet, akkor megáll egy nem elfogadó állapotban. Ha meg a fa végtelen nagy és nincs benne elfogadó levél, akkor M' nem fog megállni (ami definíció szerint azt jelenti, hogy nem fogadja el a bemenetet). $\square$

3. Megjegyzés. *Vegyük észre, hogy a mélységi bejárás erre a célra nem használható, mert az nem tér vissza, ha a számítási fában van egy végtelen hosszú út.*

Nevezetes nyelvosztályok

Nevezetes nyelvosztályok: P, PSPACE, EXPTIME, ezek kapcsolata.

$\mathrm{TIME}(t(n))=\mathrm{coTIME}(t(n))$

$\mathrm{SPACE}(s(n))=\mathrm{coSPACE}(s(n))$

Bizonyítás: fel kell cserélni a TG elfogadó és elutasító állapotait (SPACE-nél kell a tár-idő-tétel szimulációját is alkalmazni)

$\mathrm{EXPTIME}:=\cup_{k\geq 1}\mathrm{TIME}(2^{n^k})$

$\mathrm{PSPACE}:=\cup_{k\geq 1}\mathrm{SPACE}(n^k)$

EXPTIME osztály a gyakorlatban előforduló nyelvek univerzuma. Nemigen van olyan praktikus feladat, ami ezen kívül levő, még nehezebb nyelvhez vezetne.

$P\subseteq PSPACE\subseteq EXPTIME$

$TIME(t(n))\subset R$

$SPACE(s(n))\subset R$

$EXPTIME\subset R$

Utolsó 3 tény az átlós módszerrel bizonyítható.

P, FP, PSPACE, EXPTIME robosztus abban az értelemben, hogy nagymértékben függetlenek a gépmodelltől, amelynek a segítségével definiáltuk őket.

Elméleti értelemben hatékonyan kezelhető feladatok: P nyelvek és FP függvények.

Nevezetes nyelvosztályok

Definíció: **EXPTIME** = $\cup_{k\geq 1}$ TIME(2^{n^k}) az exponenciális időben felismerhető nyelvek osztálya.

Definíció: **PSPACE** = $\cup_{k\geq 1}$ SPACE(n^k) a polinom tárban felismerhető nyelvek osztálya.

Tétel: **P** $\subseteq$ **PSPACE** $\subseteq$ **EXPTIME**

Biz: Ha M TG t(n) időkorlátos, akkot t(n) időkorlátos is (lépésenként max. egy jelet ír a munkaszalagra).

Tehát TIME(n^k) $\subseteq$ SPACE(n^k), és így

P = $\cup_{k\geq 1}$ TIME(n^k) $\subseteq \cup_{k\geq 1}$ SPACE(n^k) = **PSPACE**

Nevezetes nyelvosztályok

Ha L $\in$ PSPACE, akkor valamely k-ra L $\in$ SPACE(n^k)

A tár-idő tétel szerint ekkor van alkalmas c > 0 hogy

L $\in$ TIME(c^{n^k}) $\subseteq$ TIME(2^{dn^k}) $\subseteq$ TIME($2^{n^{k+1}}$) $\subseteq$ EXPTIME

Tétel: **EXPTIME** $\subset$ **R** (=Rekurzív nyelvek osztálya)

Biz: $\subseteq$ következik EXPTIME időkorlátosságából.

A valódi tartalmazáshoz mutatunk egy L nyelvet, mely rekurzív, de nem **EXPTIME**–beli,

$\mathbb{P}$

A **P** osztály

Pl: TIME(n) a lineáris időben felismerhető nyelvek osztálya.

TIME(n^2) a négyzetes időben felismerhető nyelvek osztálya.

Definíció: **P** = $\cup_{k \geq 1}$ TIME(n^k) a polinom időben egyszalagos TG-pel eldönthető nyelvek osztálya.

1. Megj: Valamely, az alappal ekvivalens speciális TG modellt használva, ha az polinom időben szimulálható alap TG-pel, akkor P invariáns arra nézve, hogy az alap vagy a speciális TG-et használjuk számítási modellként.

2. Megj: **P** a praktikusan számítógéppel megoldható problémáknak felel meg. Általában találunk P-beli problémákra gyors (négyzetes, köbös,...) algoritmust

Tétel: Ha az L nyelv $n^{\log n}$ –nél rövidebb időben nem ismerhető fel, akkor L $\notin$ **P**.

NP

NP és coNP nyelvosztály, ezek kapcsolata P-vel és R-rel

[Algoritmusok – 257. oldal]

Az NTG-k segítségével fogalmazható meg kényelmesen a számításelmélet egyik legérdekesebb nyelvosztályának a definíciója.

$NP := \cup_{k\geq 1} NTIME(n^k)$.

Az NP nyelvosztály ugyanúgy épül fel az $NTIME(n^k)$ alakú nyelvhalmazokból, mint a P a $TIME(n^k)$ alakúakból. A definíciók közötti nyilvánvaló formai hasonlóság alapján az NP osztályt a P nemdeterminisztikus megfelelőjének tekinthetjük. Az NP nemdeterminisztikus polinom idejű Turing-gépekkel felismerhető nyelvekből áll.

$P \subseteq NP$.

Az NP nyelvosztály tartalmazza P-t. A két osztály közötti analógia korántsem teljes. Nem ismert például, hogy az NP osztály megegyezik-e a coNP nyelvosztállyal ($coNP = \{L \subseteq I^*; I^* \setminus L \in NP\}$). A (P=NP?) kérdés a számításelmélet egyik legfontosabb megoldatlan problémája.

$P \subseteq NP \cap coNP$.

Bizonyítás: $P \subseteq NP$ miatt $coP \subseteq coNP$. Ezután az állítás azonnal adódik a P=coP egyenlőségből.

$NP \cup coNP \subseteq PSPACE \subseteq EXPTIME \subseteq \Re$

Példák NP és coNP-beli nyelvekre

[Algoritmusok – 262. oldal]

Legyen 3-SZÍN a 3 színnel színezhető gráfok kódjaiból álló nyelv. Ekkor $3-SZÍN \in NP$. A bizonyítás során elég megmutatni, hogy a három színnel való színezhetőségnek van rövid és hatékonyan ellenőrizhető tanúja. Egy n szögpontú 3 színnel színezhető G gráf jó színezése alkalmas tanú lesz.

A G irányítatlan gráf egy köre Hamilton-kör, ha abban G minden csúcsa pontosan egyszer szerepel. Legyen H a Hamilton-kört tartalmazó gráfokból álló nyelv. Ekkor $H \in NP$. Az állítás rövid tanúja egy Hamilton-kör.

Egy gráf síkba rajzolható, ha a pontjainak kölcsönösen egyértelműen megfeleltethetőek síkbeli pontok, és az élek olyan szakaszoknak, amelyek legfeljebb csak a végpontjaikban találkozhatnak. Legyen SÍK a síkba rajzolható gráfok nyelve. Ekkor $SÍK \in coNP$. Az állítás igazolásához azt kell megmutatni, hogy $G \notin SÍK$ ténynek is van hatékonyan ellenőrizhető tanúja. Ez pedig következik a K. Kuratowski nevezetes tételéből: a G gráf nem síkba rajzolható $\Leftrightarrow$ G topologikusan tartalmaz teljes ötszöget vagy 3 ház – 3 kút gráfot ($K_5, K_{3,3}$).