Lineáris egyenletrendszerek

Definíció

Az

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

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:

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

Lineáris egyenletrendszerek megoldhatósága

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: $\text{rang}(A)$.

Tétel – rangkritérium

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

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 $n - \mathrm{rang}(A)$. 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 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 $(0 \ldots 0| \neq 0)$ sor adódik, akkor az egyenletrendszer ellentmondásos.
   - Ha az eljárás végén n sor marad, akkor az egyenletrendszer határozott, ha kevesebb, akkor határozatlan.

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

2021. június 7.

1. Determinisztikus Turing-gépek

1.1. Turing gép

A Turing gép (Alan Turing, 1936) egy potenciálisan végtelen *szalagmemóriával* és egy *író-olvasó fejjel* ellátott véges automata. A szalagmemória minden egyes pozíciója 1 *szalagábécébeli* elemet képes tárolni. Kezdetben a gép kezdőállapotban van és a szalagon egy véges hosszú input szó helyezkedik el. Az input szó előtti és utáni szalagpozíciók szóközzel vannak feltöltve. (**Feltételezzük, hogy az input szó utolsó karaktere nem szóköz**) Az input szó tehát az író-olvasó fej alatti betűtől (jobbra haladva) tart a szalag utolsó nem üres betűjéig. Speciálisan, üres input szó is elképzelhető. Ez esetben a szalag minden egyes pozíciója szóközzel van feltöltve, és az író-olvasó fej ezek egyikére mutat. Kezdetben az író-olvasó fej az input szó első betűjén áll. Egy Turing-gép elkülönített időpillanatokban hajt végre egy-egy elemi műveletet, aminek a részei a következőek:

1. író-olvasó fej alatti betű beolvasása,
2. ezen betű felülírása,
3. belső állapot megváltoztatása
4. író-olvasó fej mozgatása: balra, jobbra, helyben hagy.

Amennyiben a Turing-gép eljut egy végállapotba, megáll.

1.1.0.1. Definíció. A $TM = (Q, T, V, q_0, \#, d, F)$ rendezett hetest *Turing-gépnek* nevezzük, ahol

- Q a gép állapotainak véges halmaza,
- $q_0 \in Q$ a kezdő állapot,
- V a szalagábécé,
- $T \subseteq V$ az inputábécé,
- $\# \in (V \setminus Q)$ a szóköz betű,
- $F \subseteq Q$ a végállapotok halmaza,
- $d : Q \times V \to Q \times V \times \{Bal, Jobb, Helyben\}$
  a gép mozgásfüggvénye/átmenetfüggvénye

1.2. Konfiguráció

1.2.0.1. Definíció. A (u, q, av) hármast a TM Turing-gép **pillanatnyi konfigurációjának** nevezzük, ahol

- $a \in V \cup \{\lambda\}$,
- $u, v \in V^*$,
- $q \in Q$,
- és $u \in V^+$ esetén u nem kezdődhet, $v \in V^+$ esetén pedig v nem végződhet szóközzel.

Tehát az (u, q, av) konfiguráció azt jelenti, hogy a gép q belső állapotban van, miközben a feje éppen egy a jel felett áll, miközben a szalag "értelmes tartalma" éppen uav, vagyis a szalagon u áll a fejtől balra és v jobbra. Ha $q = q_0$ és $u = \lambda$, akkor **kezdőkonfigurációról** beszélünk, ha pedig $q \in F$, akkor pedig **végkonfigurációról**.

1.3. Konfigurációátmenet

1.3.0.1. Definíció. A W_1 pillanatnyi konfigurációból a W_2 **közvetlenül levezethető** (jele: $W_1 \vdash_{TM} W_2$), ha a $W_1 = (ub, q, av)$, ahol $q \in Q$, $u, v \in V^*$, $a \in V$, $b \in (V \setminus \{\#\}) \cup \{\lambda\}$, a következő feltételek valamelyike teljesülnek:

1. $W_2 = (u, q', ba'v)$ és $(q', a', Bal) \in d(q, a)$
2. $W_2 = (uba', q', v)$ és $(q', a', Jobb) \in d(q, a)$
3. $W_2 = (ub, q', a'v)$ és $(q', a', Helyben) \in d(q, a)$

1.3.0.2. Definíció. A W konfigurációból a W' konfiguráció **levezethető** a TM Turing gépben (jele: $W \vdash_{TM}^* W'$), ha létezik pillanatnyi konfigurációk egy olyan $W_0, ..., W_n$ sorozata, hogy $W_i \vdash W_{i+1}$, $i = 0, ..., n-1$ mellett a $W_0 = W$ és $W_n = W'$.

1.4. Szavak levezethetősége

1.4.0.1. Definíció. TM Turing gép **elfogadja** a $w \in T^*$ szót, ha van olyan W_1, W_2, W_3,..., W_k konfiguráció sorozat, hogy

1. $W_0 = (\#, q_0, w)$
2. $W_i \vdash W_{i+1}$, ahol $0 \leq i \leq n-1$
3. W_n elfogadó konfiguráció, azaz $W_n = (u, q_{acc}, v)$, ahol $u, v \in T^*$

Az $L(TM)$ nyelvet nevezzük a TM Turing gép által elfogadott nyelvnek. Ha a Turing gép nem fogad el egy a bemenetre írt szót, akkor

- nem elfogadó állapotban áll meg,
- vagy egyáltalán nem áll meg.

1.5. Többszalagos Turing gépek

Lényege, hogy egy szalag helyett k darab szalaggal rendelkezik. Az *input* mindig az első szalagon van. Többszalagos gépek esetén szokás egy szalagot az outputnak is fenntartani, ekkor a számítás végén azon a szalagon olvasható az eredmény, illetve sokszor az input szalag csak olvasható.

A már megismert *determinisztikus, egyszalagos* Turing gép átmenetfüggvényének definíciója a következőképpen módosul:

- $d : Q \times V^k \rightarrow Q \times V^k \times \{Bal, Jobb, Helyben\}^k$

1.5.0.1. Tétel. Minden k szalagos TM_1 Turing géphez létezik olyan 1 szalagos TM_2 Turing gép, hogy $L(TM_1) = L(TM_2)$.

1.6. Univerzális Turing gép

1.6.0.1. Tétel. Létezik egy TM_u Turing gép, amely a $< T, w >$ inputon szimulálja a T-et w-n. Azaz:

- $TM_u(< T, w >) = \begin{cases} elfogad & T\ elfogadja\ w - t \\ elutasit & T\ elutasitja\ w - t \\ végtelen & T\ végtelen\ futásba\ kerül \end{cases}$

Vegyünk egy 3 szalagos TM_u Turing gép, ami egy másik Turing gép működését szimulálni tudja. Az első szalagra a TM kódját és w kódját írva, TM_u szimulálja TM működését $w - n$.

1.6.1. Működése

1. TM_u átmásolja w kódját a 2. szalagra, T kezdőállapotát pedig a 3. szalagra.
2. A 2. szalagról a TM_u leolvassa az aktuális betűt, a 3. szalagról az aktuális állapotot.
3. Ennek megfelelően megkeresi az 1-es szalagon az ennek megfelelő átmenetet.
4. Ebből megadja az új állapotot és azt, hogy mit kell a 2. szalagra írni.

2. Lineárisan korlátozott automaták

Tulajdonképpen a Turing-gép egy olyan változata, ahol a számításra fordítható szalagterület az input által lefoglalt területre korlátozódik.

2.0.0.1. Definíció. Az $LBA = (Q, T, V, q_0, \#, d, F)$-t **lineárisan korlátozott automatának** hívjuk, ahol

- Q az állapotok halmaza
- T az inputábécé

- $V \supseteq T$ a szalagábécé
- q_0 a kezdőállapot
- $\# \in V \setminus T$ a szóköz karakter
- $d : \begin{cases} Q \times (V \setminus \{\#\}) & 2^{Q \times V \times \{Bal, Jobb, Helyben\}} \\ Q \times \{\#\} & 2^{Q \times \# \times \{Bal, Jobb, Helyben\}} \end{cases}$

A definícióból látható, hogy a szalagon levő # jelek nem írhatóak felül, ennek megfelelően az automata csak az eredeti input által elfoglalt területet használhatja számolásra.

2.0.0.1. Tétel. A lineárisan korlátozott automatákkal elfogadott nyelvek osztálya megegyezik a környezetfüggő nyelvek osztályával.

2.0.0.1. Megjegyzés. A nemdeterminisztikus lineárisan korlátozott automata az, amely a környezetfüggő nyelvosztály elfogadására alkalmas, az pedig, hogy a determinisztikus lineárisan korlátozott automaták által felismert nyelvek osztálya valódi részhalmaza-e ennek egy e jegyzet megírása idején is fennálló nevezetes megoldatlan probléma.

2.0.0.2. Megjegyzés. Ugyanezt a nyelvosztályt fogadják el azok a lineárisan korlátozott automaták, ahol az automatának egy előre rögzített k konstansszor annyi szalagpozíció (tárhely) áll rendelkezésre működése során, mint az input hossza (lásd Tárhely tétel).

2.0.0.3. Megjegyzés. Az is ismert (**Savitch tétele** (ejtsd: szévics)), hogy determinisztikus automatával, négyzetesen korlátolt tárral (vagyis, ahol az inputszó hosszának négyzetével arányos a megengedetten felhasználható szalagterület) minden környezetfüggő nyelv felismerhető.

3. Nevezetes nyelvosztályok

3.0.0.1. Definíció. Egy L nyelv **rekurzívan felsorolható**, ha van olyan Turing gép, ami minden $w \in L$ szó bemenetre elfogadó állapotban áll meg. A $w \notin L$ szavakra vagy nem elfogadó állapotban áll meg, vagy egyáltalán nem áll meg.

3.0.0.2. Definíció. Egy L nyelv **rekurzív**, ha van olyan Turing gép, ami minden bemenetre megáll, a $w \in L$ szavakra elfogadó állapotban áll meg, a $w \notin L$ szavakra nem elfogadó állapotban áll meg.

(A többi megtalálható a 6-os tételben!!)

4. Eldönthetetlen problémák

4.1. Turing gép, mint számítási modell

1. A Turing gép általános célú számítási modell.
2. Ráadásul programozható.
3. Matematikailag precíz.

A Turing gép alkalmas az intuitív algoritmus-fogalom formalizálására.

4.1.0.1. Tézis. Ha egy probléma algoritmikusan megoldható, akkor egy Turing gép is meg tudja oldani. Ha egy problémát Turing géppel nem lehet megoldani, akkor algoritmussal sem. (**Church-Turing tézis**)

4.2. Eldöntési problémák

Olyan problémák, melyekre igen-nem válasz adható. Például egy objektum benne van-e egy halmazban. Jelöljük az eldöntési problémákat a következőképpen: $(\mathbb{N}, M)$ vagy (V^*, L).

4.2.0.1. Megjegyzés. Ha van olyan algoritmus, ami megoldja $(\mathbb{N}, M)$,
akkor $(\mathbb{N}, M)$:

- algoritmikusan megoldható
- eldönthető.

Pl: $(\mathbb{N}, \mathbb{N}_{paros})$ eldönthető, $(\mathbb{N}, PRIM)$ eldönthető

4.2.0.2. Megjegyzés. Egy Turing gép elfogadja az M halmazt, ha:

- $n \in M$ bemenetre **elfogadó** állapotban áll meg
- $n \notin M$ bemenetre **nem-elfogadó** állapotban áll meg vagy egyáltalán meg sem áll.

Tehát egy probléma algoritmikusan megoldható, ha van olyan Turing gép, ami **eldönti** a hozzátartozó nyelvet, vagyis mindenképpen megáll és IGEN-t vagy NEM-at ad eredményként. Azaz egy probléma algoritmikusan megoldható, ha a hozzá tartozó nyelv rekurzív.

4.3. Megoldhatatlan problémák

4.3.1. Hilbert 10. problémája

(David Hilbert a II. Nemzetközi Matematikai Kongresszuson 1900-ban tartott előadást *Matematikai problémák* címmel. Sok általa felvetett probléma mind a mai napig megoldatlan.)

4.3.1.1. Probléma. Létezik-e olyan eljárás, ami eldönti, hogy egy egész együtthatós polinomnak van-e egész gyöke?

4.3.1.1. Állítás. Nem létezik olyan eljárás, ami képes eldönteni, hogy egy egész együtthatós polinomnak van-e egész gyöke. Azaz:

- L_p=$\{< p > | p$ polinomnak van egész gyöke$\}$ nyelv nem rekurzív, csak rekurzívan felsorolható.

4.3.1.1. Bizonyítás. Adott egy P polinom k változóval. Vegyünk egy Turing gépet, ami végigmegy az összes egész szám k-ason és kiszámolja, hogy gyöke-e p-nek. Ha talál ilyet, akkor elfogadja a P polinomot. Viszont mivel az egész számok halmaza végtelen, ezért nem fogja tudni elutasítani a P polinomot. (Nem fog megállni.)

4.3.2. A megállási probléma

4.3.2.1. Probléma. Van-e olyan algoritmus/program, ami tetszőleges másik programról megmondja, hogy helyesen működik és egy adott bemenetre megáll?

4.3.2.1. Állítás. Az $A_{TM} = \{< M, w > | M\ elfogadja\ w - t\}$[1] nyelv Turing-felismerhető, azaz rekurzívan felsorolható:

- Univerzális Turing gép: az $< M, w >$ bemenetre szimulálja M működését a w bemenetre:
  - megállhat $w \in L(M)$ és $w \notin L(M)$ esetén,
  - végtelen ciklusba kerülhet $w \notin L(M)$ esetén.

4.3.2.1. Bizonyítás. Tegyük fel, hogy A_{TM} rekurzív, azaz létezik olyan H Turing gép, hogy:

- $H(< M, w >) = \begin{cases} elfogad & M\ elfogadja\ w - t \\ elutasit & M\ nemfogadja\ el\ w - t \end{cases}$

[1] A "<,>" jelek kódolásra utalnak. Az $< X >$ jelöli az X objektum kódját.

Konfiguráljunk egy D Turing gépet, ami ellenkezőképp viselkedik, mint H. D az $< M >$ bemenetre:

1. Szimulálja H-t $< M, < M >>$ bemenetre.
2. Elfogadja $< M >$-et, ha H nem fogadja el $< M, < M >>$-et.
3. Elutasítja $< M >$-et, ha H elfogadja $< M, < M >>$-et.

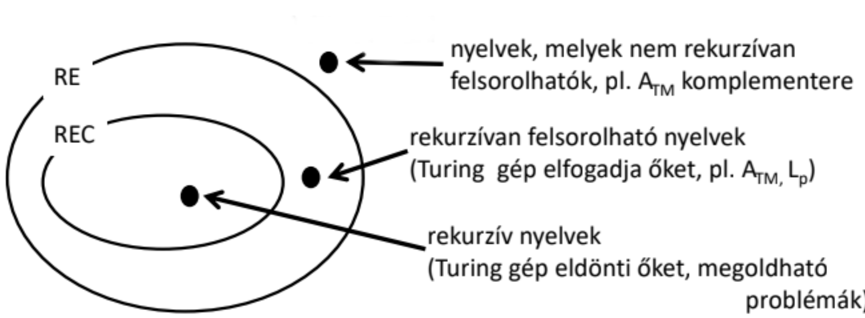
Indítsuk el D-t $< D >$ bemenettel.

- $D(< D >) = \begin{cases} elutasit & D\ elfogadja\ < D > -t \\ elfogad & D\ nem\ fogadja\ el\ < D > -t \end{cases}$

Ellentmondásra jutottunk, azaz az A_{TM} nyelv nem rekurzív.

4.3.2.1. Megjegyzés. Ha L és $\overline{L}$ is rekurzívan felsorolható, akkor L rekurzív[2]). Vagyis az $\overline{A_{TM}}$ nyelv nem rekurzívan felsorolható.

4.4. Összegzés



1. ábra. Algoritmikus megoldhatóság

5. Nemdeterminisztikus Turing gépek

A nemdeterminisztikus Turing gépek esetén az átmenet függvény a következő:

- $d : Q \times V \to 2^{Q \times V \times \{Balra, Jobbra, Helyben\}}$, ahol a $2^{Q \times V \times \{Balra, Jobbra, Helyben\}}$ a $Q \times V \times \{Balra, Jobbra, Helyben\}$ halmaz összes részhalmazának a halmaza.

[2]Bővebben: `https://gyires.inf.unideb.hu/KMITT/b24/ch09s03.html`

5.0.0.1. Állítás. Minden nemdeterminisztikus Turing gép átalakítható determinisztikussá úgy, hogy ne változzon meg a nyelv amit elfogad.

5.0.0.1. Definíció. Az TM nemdeterminisztikus Turing gép **elfogadja** az x inputot, ha a TM-et x bemenettel a kiinduló helyzetből indítva van legalább egy elfogadó (egy elfogadó állapotban véget érő) számítási út.

5.0.0.2. Állítás. Az $x \notin L(TM)$ pontosan akkor, ha a TM gépet x inputtal indítva nincs elfogadó számítási út.

5.0.0.1. Megjegyzés. Az NTG-k számításait néha hasznos egy gyökeres irányított faként elképzelni. A fa csúcsait a gép pillanatnyi helyzeteivel címkézhetjük. A gép pontosan akkor fogadja el az x inputot, ha a fában van olyan gyökértől levélig menő út, melynél a levélhez elfogadó állapot tartozik.

6. Tár és idő korlátok

Szeretnénk viszonylag pontos fogalmakat adni arra, hogy egy algoritmus gyors, illetve hogy hatékonyan bánik a tárral. Ezt úgy tesszük, hogy korlátozzuk az algoritmus (Turing-gép) számolási idejét vagy a felhasznált tárcellák számát. Az idő- és tárfelhasználást az *input* hosszának függvényében vizsgáljuk.

6.1. Időkorlátosság

Legyen $t : \mathbb{Z}^+ \to \mathbb{Z}^+$ egy függvény, melyre minden $n \in \mathbb{Z}^+$ esetén $t(n) \geq n$ teljesül.

6.1.1. Determinisztikus Turing gép

6.1.1.1. Definíció. A TM Turing gép $t(n)$ **időkorlátos**, ha n hosszú *inputokon* legfeljebb $t(n)$ lépést tesz.

6.1.1.1. Megjegyzés. A TM algoritmust/Turing gépet akkor tekintjük gyorsnak, ha $t(n)$ egy lassan növekedő függvény.

6.1.1.2. Definíció. A $TIME(t(n))$ egy olyan nyelvosztály, melyre teljesül, hogy

- $TIME(t(n)) := \{L \subseteq I^* | L$ felismerhető egy $\mathcal{O}(t(n))$ időkorlátos TM Turing géppel$\}$.

6.1.1.2. Megjegyzés. A $TIME(t(n))$ nyelvosztályba tehát azok az L nyelvek tartoznak, amelyekhez létezik $ct(n)$ időkorlátos Turing gép. A c állandó függhet L-től. A definíció lényeges eleme, hogy n hosszú x inputokon a számítás mindig befejeződik legfeljebb $ct(n)$ lépésben, tekintet nélkül arra, hogy $x \in L$ teljesül-e. Ennek következményeként a $TIME(t(n))$ **rekurzív nyelvekből** áll.

6.1.2. Nemdeterminisztikus Turing gép

6.1.2.1. Definíció. Egy NTM nemdeterminisztikus Turing gép $t(n)$ **időkorlátos**, ha n hosszúságú inputon NTM minden számítási út mentén legfeljebb $t(n)$ lépést téve megáll.

6.1.2.1. Megjegyzés. Úgy is fogalmazhatunk, hogy egy tetszőleges n hosszúságú x input esetén a számításhoz rendelt fa magassága legfeljebb $t(n) + 1$.

6.1.2.2. Definíció. Az $NTIME(t(n))$ egy olyan nyelvosztály, melyre teljesül, hogy

- $NTIME(t(n)) := \{L \subseteq I^* | L$ felismerhető egy $\mathcal{O}(t(n))$ időkorlátos NTM Turing géppel$\}$

6.2. Tárkorlátosság

Legyen $s : \mathbb{Z}^+ \rightarrow \mathbb{Z}^+$ egy függvény, melyre minden $n \in \mathbb{Z}^+$ számmal igaz, hogy $s(n) \geq log_2 n$

6.2.0.1. Definíció. A TM Turing gép $s(n)$ **tárkorlátos**, ha n hosszú inputokon legfeljebb $s(n)$ tárcellát használ a munkaszalagon (azaz $S_{TM}(n) \leq s(n)$)

6.2.0.1. Megjegyzés. Az $s(n) \geq log_2 n$ kikötés enyhe és értelmes feltevés. Ennyi hely kell ugyanis ahhoz, hogy egy n cellából álló szalagrészt – például az input szalag érdemi részét – címezni tudjunk. Emlékeztetünk még itt arra, hogy ha TM-nek csak egy szalagja van, akkor az a definíció szempontjából munkaszalagnak tekintendő.

6.2.0.2. Definíció. A $SPACE(s(n))$ egy olyan nyelvosztály, melyre teljesül, hogy:

- $\{L \subseteq I^* | L$ felismerhető egy $\mathcal{O}(s(n))$ tárkorlátos TM Turing géppel$\}$

6.3. Kiegészítés

A nyelvekhez hasonló módon kaphatunk idő-, illetve tárkorlátokkal meghatározott függvényosztályokat. Csupán annyit kell tennünk, hogy a definíciókban a nyelveket felismerő TG-k helyett függvényeket kiszámoló TG-ket szerepeltetünk.

6.3.0.1. Definíció. $FTIME(t(n)) :=$ az $\mathcal{O}(t(n))$ időkorlátos TG-k által kiszámítható $f : I^* \to I^*$ függvények osztálya.

6.3.0.2. Definíció. $FSPACE(s(n)) :=$ az $\mathcal{O}(s(n))$ tárkorlátos TG-k által kiszámítható $f : I^* \to I^*$ (parciális) függvények osztálya.

7. Bonyolultsági osztályok(P, NP)

7.1. P osztály

A determinisztikus módon, $t(n)$ időkorlátozással számoló Turing gépekkel kiszámítható nyelvek a $TIME(t(n))$ bonyolultsági osztályt alkotják. Általában az $t(n)$ nemnegatív egészekhez nemnegatív egészeket rendelő függvénytől megköveteljük, hogy monoton növekvő legyen. Ha $f(n) = c$ egy $c \in \mathbb{N}$ konstansra, akkor a nyelv bármely v szavát maximum $|v| + c$ lépésben eldönti a TM Turing-gép. Szokásos a lineáris függvény($t(n) = cn$), illetve az n tetszőleges polinomjának használata(ahol n a bemeneti szó hossza).

7.1.0.1. Definíció. Azon nyelvek (problémák) unióját, amelyekhez van olyan determinisztikus Turing gép, ami ezek szavait valamilyen polinomfüggvénnyel megadható időben eldönti (kiszámítja), **P** bonyolultsági osztálynak nevezzük.

7.2. NP osztály

Legyen most TM egy nemdeterminisztikus Turing gép. Azt mondjuk, hogy TM $t(n)$ időben eldönti/felismeri az L nyelvet, ha bármely $v \in L$ szóval indítva a kezdőkonfigurációból van olyan számítás, amely elfogadja v-t legfeljebb $t(n)$ lépés után ($n = |v|$).

7.2.0.1. Definíció. Azon nyelvek unióját, amelyekhez van olyan nemdeterminisztikus Turing gép, ami ezek szavait valamilyen polinomfüggvénnyel megadható időben eldönti (kiszámítja), **NP** bonyolultsági osztálynak nevezzük.

7.2.0.1. Megjegyzés. A számítógép-tudomány egyik legfontosabb nem tisztázott problémája a P és NP bonyolultsági osztályok viszonyának eldöntése, vagyis mivel P⊆NP, ezért a kérdés P≡?NP alakba írható. Általában elfogadott az a feltételezés, hogy a két osztály nem egyezik meg, egyelőre azonban nem ismert olyan feladat (nyelv) ami NP-ben van és bizonyítottan nincs P-ben.

7.2.0.2. Megjegyzés. Minden NP-beli nyelvre (problémára) igaz, hogy minden szavára létezik egy "tömör bizonyíték" (ami polinomiális időben ellenőrizhető) arra, hogy az adott szó benne van a nyelvben.

7.2.0.2. Definíció. Legyen X és Y két eldöntési probléma. Az X **Karp-redukciója (polinomiális visszavezetése)** az Y problémára egy olyan polinom időben számolható f függvény, amely X minden lehetséges bemenetéhez hozzárendeli Y egy lehetséges bemenetét úgy, hogy

$$x \in X \Leftrightarrow f(x) \in Y$$

Jelölése: $X \prec Y$, ha X-nek van Karp-redukciója Y-ra.

7.2.0.3. Definíció. Az X eldöntési probléma **NP-nehéz**, ha tetszőleges, $X' \in$ NP probléma esetén létezik $X' \prec X$ Karp-redukció.

7.2.0.4. Definíció. Az X eldöntési probléma **NP-teljes**, ha $X \in$ NP és X NP-nehéz.

7.2.0.3. Megjegyzés. Egy L problémát **NP-teljesnek** nevezünk, ha abból, hogy $L \in$P az következik, hogy P=NP.

7.2.0.1. Tétel. Ha az X probléma NP-teljes, $Y \in$ NP és $X \prec Y$, akkor Y is NP-teljes.

7.2.1. SAT probléma

Az egyik leghíresebb NP-teljes probléma a **SAT**(angol: satisfiability, kielégíthetőség szóból) probléma. A feladat maga röviden a következő:

- Adott egy propozicionális logikai formula, döntsük el, hogy kielégíthető-e (vagyis lehet-e a propozicionális változóknak úgy igaz-hamis értéket adni, hogy a formula igaz legyen).

A feladatnak több speciális megfogalmazása ismert:

1. A formula konjunktív normálformában adott. A feladat így is NP-teljes.
2. A konjunktív normálforma mellett az is adott, hogy az egyes tagok hány literált tartalmazhatnak. (n-SAT probléma). Az n-SAT $n \geq 3$ esetén NP-teljes, míg a 2-SAT probléma P-beli.

A "tömör bizonyíték" ez esetben egy kielégítő kiértékelés, amely megadja mely Boole-változó értéke legyen igaz és melyek legyenek hamisak.

7.2.2. Egyéb NP-teljes problémák

- 3-szín probléma
- Maximális méretű független pontrendszer gráfokban
- Maximális méretű klikk
- Részgráf izomorfia probléma
- Hamilton-kör probléma
- Hamilton-út probléma
- Részhalmaz összeg probléma
- Partíció probléma
- A háromdimenziós házasítás
- X3C

Kiegészítő anyag: `http://www.cs.bme.hu/~kiskat/algel/npcompl-2019.pdf`

7.3. Egyéb fontos bonyolultsági osztályok

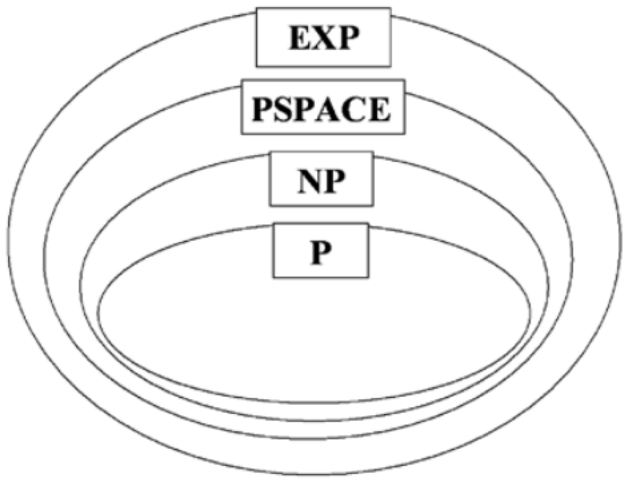
7.3.0.1. Definíció. PSPACE: determinisztikus Turing-géppel polinomiális szalagigénnyel kiszámolható problémák osztálya.

7.3.0.2. Definíció. NPSPACE: nemdeterminisztikus módon polinomiális szalagigénnyel kiszámolható problémák osztálya.

7.3.0.1. Megjegyzés. A PSPACE és az NSPACE problémaosztályok egybeesnek.

7.3.0.3. Definíció. EXP: determinisztikusan exponenciális időben kiszámolható problémák osztálya.

7.4. Összegzés



2. ábra. Bonyolultsági osztályok viszonya

7.5. További érdekességek

További érdekességeket lehet olvasni a bonyolultsági osztályokról, tár és időkorlátokról az *Ivanyos*, *Rónyai*, *Szabó - Algoritmusok* című könyv 8. fejezetéből: itt