

ALGORITMUSELMÉLET

Kiszámíthatóság-elmélet

12. előadás

A kiszámíthatóság-elmélet alapfogalmai

Az előadás során az algoritmussal való kiszámíthatósággal foglalkozunk. Azt a problémát fogjuk **algoritmikusan eldönthetőnek** ill. **kiszámíthatónak** nevezni, amely Turing-géppel is eldönthető, ill. kiszámítható.

A továbbiakban legyen Σ egy véges ábécé, amely nem tartalmazza a \square szimbólumot.

12.1 definíció: (rekurzívan felsorolható v. Turing-felismerhető nyelv)

Az $L \subseteq \Sigma^*$ nyelvet **rekurzívan felsorolhatónak** (vagy **Turing-felismerhetőnek**) nevezzük, ha létezik olyan M Turing-gép, melyre $L = L(M)$.

Egy L nyelv tehát akkor és csakis akkor rekurzívan felsorolható, ha létezik olyan algoritmus, amely az L nyelv szavait felsorolja.

Rekurzívan felsorolható nyelvek osztálya:

$$RE = \{ L \subseteq \Sigma^* : L \text{ rekurzívan felsorolható} \}$$

Amennyiben az L rekurzívan felsorolható, akkor előfordulhat, hogy valamely $w \in \Sigma^* \setminus L$ input szóra az M Turing-gép nem áll meg.

A végtelen ciklusok kizárásával kapjuk a következő fogalmat:

12.2 definíció: (rekurzív v. Turing-eldönthető nyelv)

Az $L \subseteq \Sigma^*$ nyelvet **rekurzív**nak (vagy **Turing-eldönthető**nek) nevezzük, ha létezik olyan M Turing-gép, melyre $L = L(M)$, és az M gép minden $w \in \Sigma^*$ input szóra megáll.

Egy L nyelv tehát akkor és csakis akkor rekurzív, ha létezik olyan eldöntő algoritmus, amelynek inputjára tetszőleges szót helyezve eldönti, hogy a szó beletartozik-e az L nyelvbe.

Megjegyzés: Minden rekurzív nyelv nyilván rekurzívan felsorolható. Elegendő ugyanis rendre megvizsgálni az összes $w \in \Sigma^*$ szót, s alkalmazni rájuk az eldöntési algoritmust. Egy w szót bele vesszük a felsorolásba, ha igen választ kapunk, egyébként elhagyjuk.

Rekurzív nyelvek osztálya:

$$R = \{ L \subseteq \Sigma^* : L \text{ rekurzív} \}$$

12.1 tétel:

Egy L nyelv akkor és csak akkor rekurzív, ha mind az L mind az \bar{L} nyelv rekurzívan felsorolható.

Bizonyítás:

- I. Legyen L rekurzív nyelv. Megmutatjuk hogy az L és \bar{L} nyelvek rekurzívan felsorolhatók.

Mivel L rekurzív, ezért tetszőleges $w \in \Sigma^*$ szóra a $w \in L$ probléma algoritmikusan eldönthető. Ismert, hogy $w \in L$ akkor és csakis akkor teljesül, amikor $w \notin \bar{L}$. Eszerint az \bar{L} nyelvre ugyanazt az eldöntési algoritmust használhatjuk, mint az L nyelvre, azzal a különbséggel, hogy amely szavakat az L nyelv esetén elfogadtunk, azokat most nem fogadjuk el, és fordítva.

Létezik tehát algoritmus, amely mind az L , mind az \bar{L} nyelv szavait felsorolja, vagyis ezek a nyelvek rekurzívan felsorolhatók.

II. Legyenek L és \bar{L} nyelvek rekurzívan felsorolhatók. Megmutatjuk, hogy az L nyelv rekurzív.

Mivel az L és \bar{L} nyelvek rekurzívan felsorolhatók, ezért létezik algoritmus az egyik ill. a másik nyelv szavainak felsorolására.

Kombináljuk ezeket az algoritmusokat úgy, hogy váltakozva hol az egyikkel, hol a másikkal állítunk elő egy-egy szót. Így olyan w_0, w_1, \dots felsorolást kapunk, ahol $w_{2i} \in L$, $w_{2i+1} \in \bar{L}$, minden $i = 0, 1, 2, \dots$ értékre.

Mivel a felsorolás teljes, ezért egy tetszőleges $w \in \Sigma^*$ szónak valahol elő kell fordulnia. Ezért a w szó L nyelvbe való tartozásának eldöntéséhez csak azt kell meghatároznunk, hogy páros vagy páratlan pozícióban fordul-e elő. Ezzel tulajdonképpen egy döntési algoritmust adtunk meg az L nyelvre, vagyis ez a nyelv rekurzív.

□

Mivel a Turing-gépek nemcsak nyelvek felismerésére, hanem függvények kiszámítására is használhatók, ezért analóg fogalmak definiálhatók (parciális) függvényekre is.

A kiszámítható függvények többsége parciális függvény, ami azt jelenti, hogy bizonyos helyeken nincsenek meghatározva, mert ott a kiszámításuk végtelen ciklusba kerül.

12.3 definíció: (parciálisan rekurzív v. parciálisan Turing-kiszámítható függvény)

Az $f: \Sigma^* \rightarrow \Sigma^*$ parciális függvényt **parciálisan rekurzív függvénynek** (vagy **parciálisan Turing-kiszámíthatónak**) nevezzük, ha létezik olyan M Turing-gép, melyre $f = f_M$.

Egy f függvény tehát akkor és csakis akkor parciálisan rekurzív, ha létezik olyan algoritmus, amely az f függvényt kiszámítja.

12.4 definíció: (rekurzív v. Turing-kiszámítható függvény)

Az $f : \Sigma^* \rightarrow \Sigma^*$ parciális függvényt **rekurzív függvénynek** (vagy **Turing-kiszámítható**nak) nevezzük, ha létezik olyan M Turing-gép, melyre $f = f_M$, és f minden $s \in \Sigma^*$ inputra értelmezve van.

Megjegyzés: Nyilvánvaló, hogy egy rekurzív függvény egyben parciálisan rekurzív is. A rekurzív elnevezés onnan ered, hogy a rekurzív függvények éppen azok a függvények, melyek bizonyos egyszerű függvényekből rekurzió segítségével felépíthetők.

A rekurzív nyelveket és a rekurzív függvényeket fogjuk algoritmussal kezelhető nyelveknek és függvényeknek tekinteni.

Felmerülhet a kérdés: **nem túl szűk-e ez a meghatározás?**

Nincsenek-e olyan algoritmusok, amelyek nem valósíthatók meg Turing-géppel? Például egy magasabb szintű programozási nyelven megírt programmal nem lehet-e több függvényt kiszámítani?

A választ a kérdésre ALONZO CHURCH adta meg 1936-ban, aki ekkor fogalmazta meg az ún. **Church tézist**, miszerint *az algoritmussal kiszámítható függvények osztálya megegyezik a parciálisan rekurzív függvények osztályával.*

Ennek a tézisnek az ekvivalens megfelelője a **Church-Turing tézis**, amely szerint *minden formalizálható probléma, ami megoldható algoritmussal, az megoldható Turing-géppel is*, illetve bármilyen, a Turing-gép fogalmával azonos számítási teljesítményű absztrakt modellel.

Church-Turing tézis:

- Egy $f: \Sigma^* \rightarrow \Sigma^*$ parciális függvény kiszámítható akkor és csakis akkor, amikor f parciálisan rekurzív.
- Egy $f: \Sigma^* \rightarrow \Sigma^*$ függvény kiszámítható akkor és csakis akkor, amikor f rekurzív.
- Egy $L \subseteq \Sigma^*$ nyelvre a nyelvbe tartozás problémája algoritmussal eldönthető akkor és csakis akkor, amikor L rekurzív.

Megjegyzés: A Church-Turing tézis nem tekinthető formálisan bizonyított állításnak. Azt a *tapasztalati tényt* fejezi ki csupán, hogy eddig nem sikerült Turing-gépek erejét meghaladó realisztikus számítási modellt (algoritmikus rendszert) találni.

Bizonyítottan *Turing-ekvivalensek* a C, Java, Prolog, Pascal, stb. programozási nyelven megírt nem interaktív programok.

A Church-Turing tézist elfogadva a következő két tétel állítása úgy értelmezhető, hogy nem lehet mindent algoritmussal eldönteni, illetve kiszámítani. Vannak tehát algoritmussal nem felismerhető nyelvek és nem kiszámítható függvények.

12.2 tétel:

Létezik olyan $L' \subseteq \Sigma^*$ nyelv, amely nem rekurzívan felsorolható.

Bizonyítás:

Egy Turing-gép leírható véges jelsorozattal, pl. magyar nyelven. Ezért az összes Turing-gépet tartalmazó halmaz számossága megszámlálhatóan végtelen. Ez annyit tesz, hogy az M_0, M_1, M_2, \dots felsorolás tartalmazza az összes Turing-gépet.

Mivel egy Turing-gép egyértelműen meghatározza az általa felismert nyelvet, ezért a rekurzívan felsorolható nyelvek is sorszámozhatók a természetes számokkal: $L(M_0), L(M_1), L(M_2), \dots$.

Elegendő megmutatni, hogy $L \subseteq \Sigma^*$ nyelveket nem lehet ilyen módon megszámozni. Ebből ugyanis azonnal következik, hogy létezik olyan $L' \subseteq \Sigma^*$ nyelv, ami nem nyelve egyetlen Turing-gépnek sem. Halmazelméleti fogalmakat használva: a rekurzívan felsorolható nyelvek halmaza *megszámlálhatóan végtelen*, az összes nyelv pedig *kontinuum* számosságú.

A Σ^* halmaz elemei, a véges hosszúságú Σ -beli szimbólumokból képzett szavak is megszámozhatók természetes számokkal. Hasznos számozást ad a kanonikus felsorolás: vegyük először az üres szót, majd soroljuk fel az 1 hosszúságú, majd a 2 hosszúságú szavakat, stb. Az azonos hosszúságú szavak a Σ^* lexikografikus rendezése szerint kövessék egymást.

A kanonikus felsorolás tehát a Σ^* szavainak egy w_0, w_1, w_2, \dots sorrendjét adja, s ebben a felsorolásban minden $w \in \Sigma^*$ szó pontosan egyszer szerepel.

Az L' nyelvnek a w_i szó pontosan akkor legyen eleme, ha $w_i \notin L(M_i)$, ahol $i = 1, 2, \dots$. Az L' nyelv definíciója korrekt, ugyanis tetszőleges $w \in \Sigma^*$ szóról egyértelműen rendelkezünk. Az így megszerkesztett L' nyelv nem lehet azonos egyetlen nyelvvel sem az $L(M_0), L(M_1), L(M_2), \dots$ nyelvek közül, hiszen a w_i szó az L' és $L(M_i)$ nyelvek közül az egyiknek eleme, a másiknak pedig nem. Az L' nyelv tehát nem rekurzívan felsorolható.

Az eljárást amellyel az L' nyelvet meghatároztuk, **Cantor-féle átlós módszer**nek nevezzük. Az L' nyelv sorozatát úgy kapjuk, hogy a táblázat főátlójában levő elemeket az ellenkezőjűkre változtatjuk.

	w_0	w_1	...	w_i	w_{i+1}	...
$L(M_0)$	nem	nem	...	nem	nem	...
$L(M_1)$	igen	nem	...	nem	nem	...
...						
$L(M_i)$	nem	igen	...	igen	nem	...
$L(M_{i+1})$	igen	nem	...	nem	nem	...
...						
L'	igen	igen	...	nem	igen	...

12.3 tétel:

Létezik olyan $f: \Sigma^* \rightarrow \Sigma^*$ parciális függvény, amely nem parciálisan rekurzív.

Eldönthetetlen nyelvek

12.5 definíció: (eldönthetetlen nyelv)

Az $L \subseteq \Sigma^*$ nyelvet **eldönthetetlen nyelv**nek nevezzük, ha L nem rekurzív.

12.6 definíció: (diagonális nyelv)

A **diagonális nyelv** azokat a w szavakat tartalmazza, melyek olyan Turing-gépeket írnak le, amelyek saját w kódjukat nem fogadják el.

$$L_d = \{ w : \text{létezik } M_w \text{ Turing-gép és } w \notin L(M_w) \}$$

12.5 tétel:

Az L_d nyelv nem rekurzívan felsorolható.

Bizonyítás: (indirekt módon)

Tételezzük fel, hogy a diagonális nyelv rekurzívan felsorolható. Ez azt jelenti, hogy létezik olyan M Turing-gép, amelyre $L_d = L(M)$.

Nézzük meg ennek az M gépnek a w kódját, vagyis azt a szót, amelyre $M = M_w$ teljesül. Az L_d nyelv és a w szó viszonyát illetően két lehetőség van: vagy $w \in L_d$ vagy $w \notin L_d$.

- Ha $w \in L_d$, akkor az L_d nyelv definíciója szerint $w \notin L(M_w)$. Mivel $L(M_w) = L(M) = L_d$, ezért kaptuk, hogy $w \notin L_d$, ami **ellentmondás**.
- Ha $w \notin L_d$, akkor az L_d nyelv definíciója szerint $w \in L(M_w)$, hiszen az M_w gép létezik. Mivel $L(M_w) = L(M) = L_d$, ezért kaptuk, hogy $w \in L_d$, ami ismét **ellentmondás**.

Mivel mindkét esetben ellentmondáshoz jutottunk, ezért az eredeti feltevésünk nem igaz, vagyis a diagonális nyelv nem rekurzívan felsorolható.

12.7 definíció: (univerzális nyelv)

Az **univerzális nyelv** $w\#s$ alakú szavakat tartalmaz, ahol w egy olyan Turing-gép kódja, amely elfogadja az s input szót.

$$L_u = \{ w\#s : \text{létezik } M_w \text{ Turing-gép és } s \in L(M_w) \}$$

Megjegyzés: Az elnevezés azon az egyszerű észrevételen alapszik, hogy ez a nyelv éppen az univerzális Turing-gépek nyelve: pontosan azokból a szavakból áll, amelyeket egy univerzális Turing-gép elfogad.

12.6 tétel: (Turing tétele)

Az L_u nyelv rekurzívan felsorolható, de nem rekurzív.

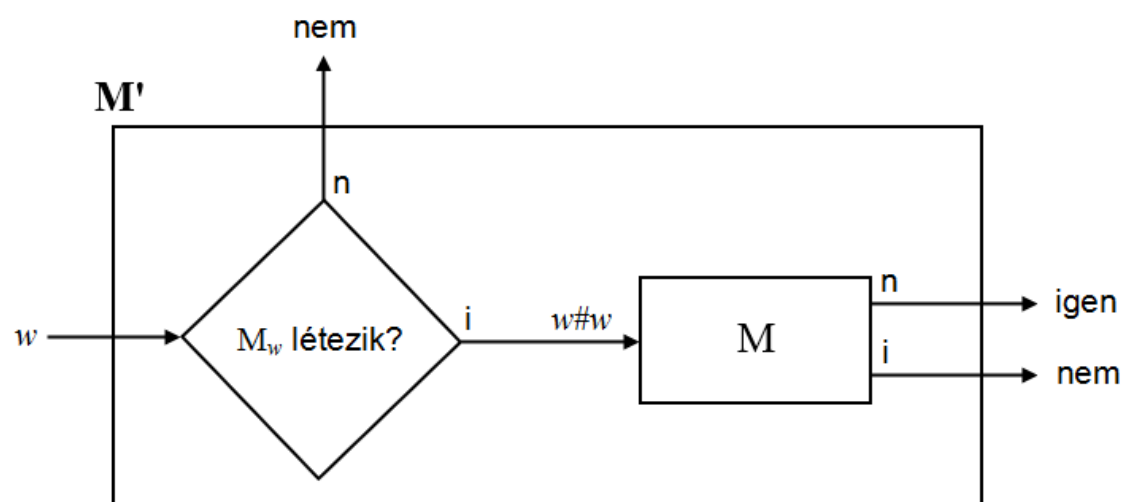
Bizonyítás: (indirekt módon)

Mivel az L_u nyelvet az univerzális Turing-gépek felismerik, ezért ez a nyelv rekurzívan felsorolható.

Megmutatjuk, hogy az L_u nyelv nem rekurzív.

Tételezzük fel, hogy az univerzális nyelv rekurzív. Ez azt jelenti, hogy létezik olyan M Turing-gép, amelyre $L_u = L(M)$, és az M gép minden input szóra megáll.

Legyen M' az a Turing-gép, amelynek működését az alábbi folyamatábra mutatja:



Az M' Turing-gép a $w \in \Sigma^*$ inputtal elindítva a következő lépéseket végzi:

- 1) Ellenőrzi, hogy az M_w gép létezik-e. Ha nem, akkor elutasító állapotban megáll.
- 2) Ha az M_w gép létezik, akkor elindítja az M gépet $w\#w$ inputtal. Ha M elfogadó állapotban áll meg, akkor az M' gép elutasító állapotban végez, egyébként elfogadó állapotban áll meg.

Az M' Turing-gép mindig megáll, mert az **1)** lépés tesztje algoritmussal elvégezhető, és a **2)** lépésben az M gép minden input szóra megáll.

Az ábrából látható, hogy az M' gép pontosan akkor fogadja el a w szót, ha az M_w gép létezik és $w \notin L(M_w)$. Más szóval, az M' gép éppen az L_d nyelvet fogadja el.

Ez azonban **ellentmondás**, mert az L_d nyelv nem rekurzívan felsorolható, és ezért nem lehet egyetlen Turing-gép nyelve sem. Ezért az M Turing-gép nem létezhet, vagyis az univerzális nyelv nem lehet rekurzív.

□

Nyelvek és függvények kapcsolata

A kiszámíthatóság szempontjából a rekurzívan felsorolható nyelvek a parciálisan rekurzív függvényekkel, míg a rekurzív nyelvek a rekurzív függvényekkel vannak kapcsolatban.

12.9 tétel:

Az $L \subseteq \Sigma^*$ nyelv akkor és csakis akkor rekurzívan felsorolható, ha van olyan $f : \Sigma^* \rightarrow \Sigma^*$ parciálisan rekurzív függvény, melynek értékkészlete éppen az L nyelv.

12.10 definíció: (nyelv karakterisztikus függvénye)

Legyen $0, 1 \in \Sigma$. Egy $L \subseteq \Sigma^*$ nyelv χ_L **karakterisztikus függvénye** a következőképpen van definiálva:

$$\chi_L(s) = \begin{cases} 1 \in \Sigma^*, & \text{ha } s \in L \\ 0 \in \Sigma^*, & \text{ha } s \notin L \end{cases}$$

12.10 tétel:

Az $L \subseteq \Sigma^*$ nyelv akkor és csakis akkor rekurzív, ha χ_L egy rekurzív függvény.

Bizonyítás:

I. Legyen L rekurzív nyelv. Megmutatjuk, hogy az χ_L karakterisztikus függvény rekurzív.

Mivel L rekurzív, ezért létezik olyan M Turing-gép, amely épp az L nyelvet ismeri fel, és minden input szóra megáll.

Legyen M' olyan Turing-gép, amely szimulálja az M gépet, majd az output szalagra 1-et ír, ha az M gép elfogadó állapotban állt meg, ill. 0-t, ha az M gép elutasító állapotban állt meg.

Nyilvánvaló, hogy az M' gép épp az L nyelv karakterisztikus függvényét számítja ki, vagyis χ_L rekurzív függvény.

II. Legyen χ_L rekurzív függvény. Megmutatjuk, hogy az L nyelv rekurzív.

Mivel χ_L rekurzív függvény, ezért létezik olyan M Turing-gép, amely épp az χ_L függvényt számítja ki, és minden input szóra megáll.

Legyen M' olyan Turing-gép, amely szimulálja az M gépet, majd elfogadó állapotban áll meg, ha az M gép output szalagján 1 a végeredmény, ill. elutasító állapotban áll meg, ha az M gép output szalagján 0 a végeredmény.

Nyilvánvaló, hogy az M' gép épp az L nyelvet ismeri fel, vagyis L rekurzív nyelv.