

Logika és számításelmélet

7. előadás

Elérhetőség, fóliasorok, ajánlott irodalom

Előadó: Tichler Krisztián

Elérhetőség: 2-708, ktichler@inf.elte.hu

Előadások itt lesznek: www.cs.elte.hu/~tichlerk

Ajánlott irodalom:

- ▶ Gazdag Zsolt: Jegyzet és fóliasor a weben:
people.inf.elte.hu/gazdagzs
- ▶ Papadimitriou: Számítási bonyolultság
- ▶ Sipser: Introduction to the Theory of Computation

Bevezetés

Algoritmikus megoldás

Ha adott egy való életben előforduló probléma akkor első feladatunk ezt a matematika absztrakt nyelvén megfogalmazni.

Általában **algoritmikus megoldást** keresünk, azaz olyan általános megoldást, mely ugyanúgy helyesen működik az inputparaméterek változtatása esetén is.

A válasz típusa alapján léteznek **eldöntési problémák** (a válasz igen/nem) illetve **kiszámítási problémák** (bármilyen lehet a válasz típusa, pl. egy szám).

Sokszor sikerül a problémára hatékony algoritmust találni, de előfordulhat az is hogy nem találunk ilyet.

Kérdés 1: Mikor nevezhetünk egy algoritmust hatékonnak?

Kérdés 2: Van-e egyáltalán mindig algoritmikus megoldás?

Bevezetés

Mikor tekinthető hatékonynak egy algoritmus?

Adott egy problémára 5 algoritmus, melyek egy n méretű input esetén rendre $\log_2 n$, n , n^2 , n^3 , 2^n erőforrást (pl. időt, tárat) használnak. Az erőforrásból legfeljebb K -t használhatnak fel. Egy kétszeres javítás (pl. egy dupla olyan gyors számítógép beszerzése) a maximális inputméretre rendre a következő hatással van: Négyzetes, kétszeres, $\sqrt{2}$ -szeres, $\sqrt[3]{2}$ -szeres, +1-es javulás.

Ezek alapján szokás a (legfeljebb) polinomiális algoritmusokat hatékonynak tekinteni.

Megjegyzés 1: A tárral gyakran hajlamosak vagyunk fukarabbul bánni, sokszor konstans vagy logaritmikus tárat használó algoritmust keresünk.

Megjegyzés 2: Előfordulhat, hogy a gyakorlat mást mutat. pl. n^{80} vs. $2^{n/100}$ vagy várható értékben polinomiális, de legrosszabb esetben exponenciális algoritmusok.

Bevezetés

Példák problémákra

1. $12322+4566=?$ *És általában?*

algorithmikus megoldás: az általános suliból jól ismert összeadó algoritmus.

Az algoritmus minden számpárra kiszámítja a megoldást, mindig terminál. Hatékonysága: számjegyek számának *lineáris* függvénye.

2. Egy közösségi oldalon elérhetem-e tőlem ismerősről ismerősre haladva Roger Federert? Valószínűleg igen, hacsak nem frissen regisztráltam és nincs, vagy alig van ismerősöm. Ha igen, mekkora a minimális lépésszám? **ELÉR** probléma: *Egy G gráfban elérhető-e u -ból v ?* **algorithmikus megoldás:** szélességi keresés **Hatékonysága:** élszám *lineáris* függvénye

Bevezetés

Példák problémákra

3. A nulladrendű logika eldöntéskérdése. Egy formulahalmaznak következménye-e egy adott formula?
Tanultuk, hogy elég egy klóz halmaz (vagy KNF) kielégíthetőségét vizsgálni. SAT probléma: kielégíthető-e adott KNF. Egy **algoritmikus megoldás**: ítélet tábla. Exponenciális futási idő. Nem ismeretes (és majd látni fogjuk, hogy nem is várható) polinom idejű algoritmikus megoldás.
4. Egy város nevezetességeit (n db.) úgy szeretné végiglátogatni egy turista (hoteléből indulva és oda visszatérve), hogy semelyik nevezetességet se látogatja meg kétszer. Ez megfelel Hamilton kör keresésének egy gráfban. Egy **algoritmikus megoldás**: minden lehetséges sorrend kipróbálása. Hatékonyság: $n!$, nem ismeretes (és majd látni fogjuk, hogy nem is várható) polinom idejű algoritmikus megoldás.

Bevezetés

Példák problémákra

5. Utazó ügynök probléma (TSP): Egy utazó ügynök szeretne repülővel végiglátogatni n várost. Bizonyos városok között van repülőjárat, a használatának pedig egy költsége (\sim jegyár). Mennyi a körút minimális költsége? **algoritmikus megoldás:** minden körút kipróbálása, és az addig talált minimum nyilvántartása. *A Hamilton kör általánosítása.*
6. generatív grammatikák szóproblémája: adott szót generálja-e a grammatika?
algoritmikus megoldás:
 3. típus esetén lineáris,
 2. típus esetén köbös,
 1. típus esetén exponenciális,
 0. típus esetén ??? (folyt. köv.)

Bevezetés

Megoldható-e minden probléma algoritmikusan?

Eddigi tanulmányaink során számos olyan problémát láttunk, melyre létezik hatékony algoritmus (pl. ELÉR), másokra (pl. TSP és SAT) nem ismert hatékony, azaz polinomiális megoldás.

Úgy tűnhet, hogy ha nem is mindig hatékonyat, de minden problémára találunk algoritmikus megoldást.

David Hilbert 1920-ban meg is hirdette programját.

Axiomatizáljuk a matematika összes elméletét egy végesen reprezentálható axiómarendszerrel!

ennek részeként: *Adjunk meg egy olyan Univerzális Algoritmust (UA), mely az összes matematikai állításról el tudja dönteni, hogy igaz-e vagy hamis!*

Bevezetés

A számítástudomány (számításelmélet) születése

Tétel (Gödel első nemteljességi tétele, 1931) Az elemi aritmetikát tartalmazó effektíven kiszámítható elmélet nem lehet egyszerre helyes és teljes. \Rightarrow a Hilbert program megvalósíthatatlan.

És az UA? Mi is pontosan az algoritmus? 1930-as évek: különböző algoritmus modellek bevezetése

- ▶ Gödel: rekurzív függvények
- ▶ Church, Kleene, Rosser: λ -kalkulus
- ▶ Turing: Turing gép

Melyik az "igazi"?

Az 1930-as évek második felétől sorra születtek olyan tételek, melyek ezen modellek megegyező számítási erejét mondták ki.

Bevezetés

A Church-Turing tézis

Church-Turing tézis

A kiszámíthatóság különböző matematikai modelljei mind az effektíven kiszámítható függvények osztályát definiálják.

Nem tétel!!!

Ha elfogadjuk a tézis igazságát, ezek bármelyike tekinthető az algoritmus matematikai modelljének. Mi a Turing gépet fogjuk választani.

Az algoritmus néhány további modellje, melyek szintén a Turing géppel egyező erejűek.

- ▶ 0. típusú grammatika
- ▶ veremautomata 2 vagy több veremmel
- ▶ C, Java, stb.

Bevezetés

A negatív válasz

Church és Turing egymástól függetlenül a következőkre jutottak

Tétel (Church, 1936)

Két λ -kalkulusbeli kifejezés ekvivalenciája algoritmikusan eldönthetetlen.

Tétel (Turing, 1936)

A Turing-gépek megállási problémája algoritmikusan eldönthetetlen.

Bevezetés

Példák problémákra

6. (folyt.) szóprobléma,
0. típus: algoritmikusan eldönthetetlen
csak parciálisan rekurzív (igen esetben termináló, nem esetben nem feltétlen termináló) algoritmus ismeretes.
7. Az elsőrendű logika eldöntéskérdésproblémája
(Entscheidungsproblem).
Church és Turing (1936) bizonyították, hogy nem adható algoritmus annak eldöntésére, hogy egy elsőrendű logikai formula logikailag igaz-e illetve hogy kielégíthető-e.
(Entscheidungsproblem \sim UA létezése).

Bevezetés

Példák problémákra

8. Hilbert 23 problémája (1900) közül a 10. a következő volt:

Diophantoszi egyenletek. Keressük meg egy tetszőleges egész együtthatós többváltozós egyenlet egész gyökeit. Pl. input:

$3xy - 2x^2 + 2z + 4$, output: $x = y = 0, z = -2$

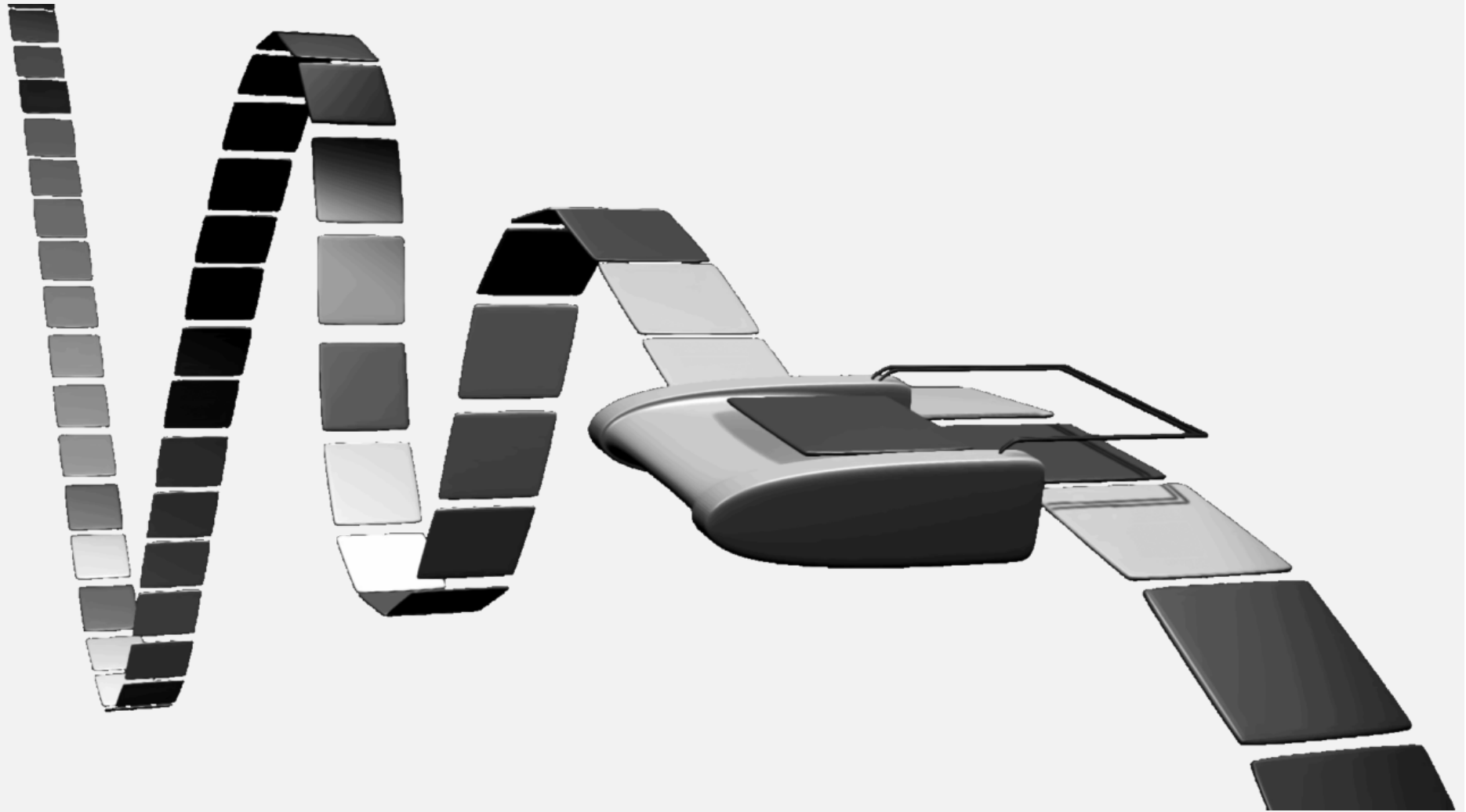
A feladat tartalmazza a "nagy Fermat tételt" is. Wiles tétele (1995): igaz a "nagy Fermat tétel", az $x^n + y^n = z^n$ diophantoszi egyenletnek nincs pozitív egész megoldása $n \geq 3$ -ra.

Matiyasevich tétele (1970): A diophantoszi egyenletek problémájára nincs algoritmikus megoldás.

Tematika

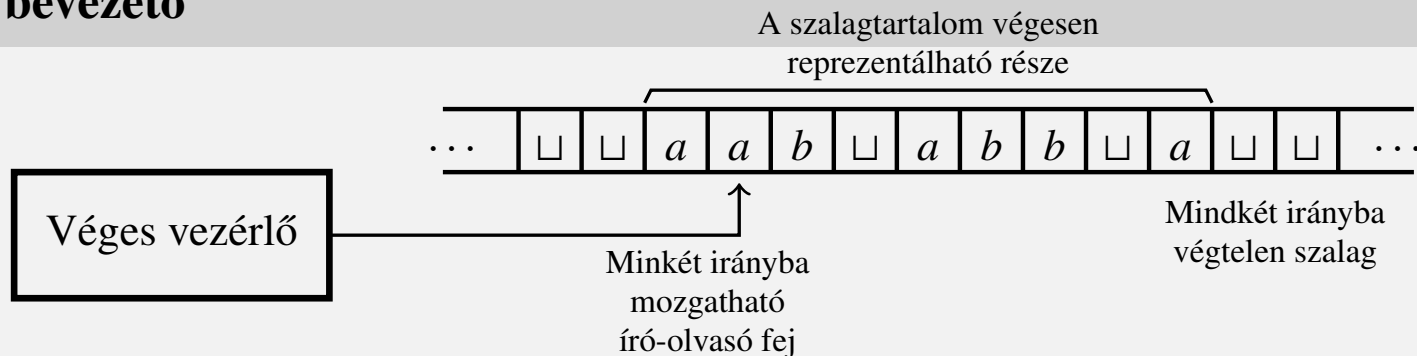
1. Turing gép és változatai, mint algoritmusmodell
2. Algoritmikusan eldönthetetlen problémák
3. Eldönthető problémák hatékonyságáról: bevezetés a bonyolultságelméletbe

Turing gépek



Turing gépek

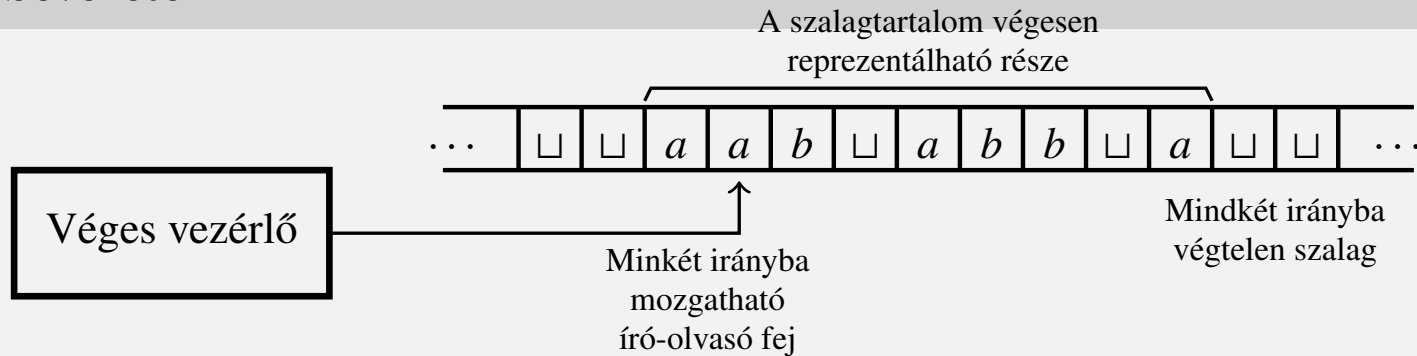
Informális bevezető



- ▶ a Turing gép (TG) az algoritmus egyik lehetséges modellje
- ▶ a TG egyetlen programot hajt végre (de bármely inputra!!!), azaz tekinthető egy célszámítógépnek.
- ▶ informálisan a gép részei a vezérlőegység (véges sok állapottal), egy mindkét irányba végtelen szalag, és egy mindkét irányba lépni képes író-olvasó fej
- ▶ kezdetben egy input szó van a szalagon (ε esetén üres), a fej ennek első betűjéről indul, majd a szabályai szerint működik. Ha eljut az elfogadó állapotába elfogadja, ha eljut az elutasító állapotába elutasítja az inputot. Van egy harmadik lehetőség is: nem jut el soha a fenti két állapotába, "végtelen ciklusba" kerül.

Turing gépek

Informális bevezető



- ▶ a gép determinisztikus, továbbá minden esetben definiált az átmenet
- ▶ a végtelen szalag potenciálisan végtelen tár
- ▶ egy \mathcal{P} probléma példányaait egy megfelelő ábécé felett elkódolva a probléma igen-példányai egy $L_{\mathcal{P}}$ formális nyelvet alkotnak. $L_{\mathcal{P}}$ (és így a probléma maga is) eldönthető, ha van olyan mindig termináló gép, mely pontosan $L_{\mathcal{P}}$ szavait fogadja el.
- ▶ a Church-Turing tézis értelmében úgy gondolhatjuk tehát, hogy éppen a TG-pel eldönthető problémák (nyelvek) az algoritmikusan eldönthető problémák.

Turing gépek

Definíció

Turing gép

A **Turing gép** (továbbiakban röviden TG) egy $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ rendezett hetes, ahol

- ▶ Q az állapotok véges, nemüres halmaza,
- ▶ $q_0, q_i, q_n \in Q$, q_0 a kezdő- q_i az elfogadó- és q_n az elutasító állapot,
- ▶ Σ és Γ ábécék, a bemenő jelek illetve a szalagszimbólumok ábécéje úgy, hogy $\Sigma \subseteq \Gamma$ és $\sqcup \in \Gamma \setminus \Sigma$.
- ▶ $\delta : (Q \setminus \{q_i, q_n\}) \times \Gamma \rightarrow Q \times \Gamma \times \{L, S, R\}$ az átmenet függvény.
 δ az egész $(Q \setminus \{q_i, q_n\}) \times \Gamma$ -n értelmezett függvény.

Az $\{L, S, R\}$ halmaz elemeire úgy gondolhatunk mint a TG lépéseinek irányai (balra, helyben marad, jobbra). Valójában elég 2 irány:

Minden helyben maradó lépés helyettesíthető egy jobbra és egy balra lépéssel egy új, csak erre az átmenetre használt új állapot bevezetése által.

Turing gépek

Konfiguráció

A TG működtetését a gép konfigurációival írhatjuk le.

Konfiguráció

A TG **konfigurációja** egy uqv szó, ahol $q \in Q$ és $u, v \in \Gamma^*$, $v \neq \varepsilon$.

Az uqv konfiguráció egy tömör leírás a TG aktuális helyzetéről, mely a gép további működése szempontjából minden releváns információt tartalmaz: a szalag tartalma uv (uv előtt és után a szalagon már csak \sqcup van), a gép a q állapotban van és az író-olvasó fej a v szó első betűjén áll. Két konfigurációt azonosnak tekintünk, ha csak balra/jobbra hozzáírt \sqcup -ekben térnek el egymástól.

A gép egy $u \in \Sigma^*$ -beli szóhoz tartozó **kezdőkonfigurációja** a $q_0u\sqcup$ szó. (Vagyis q_0u , ha $u \neq \varepsilon$ és $q_0\sqcup$, ha $u = \varepsilon$).

Elfogadó konfigurációi azon konfigurációk, melyre $q = q_i$.

Elutasító konfigurációi azon konfigurációk, melyre $q = q_n$.

Az elfogadó és elutasító konfigurációk közös elnevezése **megállási konfiguráció**.

Turing gépek

Konfigurációátmenet

Jelölje C_M egy M TG-hez tartozó lehetséges konfigurációk halmazát.
 $M \vdash \subseteq C_M \times C_M$ **konfigurációátmenet-relációját** az alábbiak szerint definiáljuk.

$\vdash \subseteq C_M \times C_M$ egylépéses konfigurációátmenet

Legyen $uqav$ egy konfiguráció, ahol $a \in \Gamma$, $u, v \in \Gamma^*$.

- ▶ Ha $\delta(q, a) = (r, b, R)$, akkor $uqav \vdash ubrv'$, ahol $v' = v$, ha $v \neq \varepsilon$,
különben $v' = \sqcup$,
- ▶ ha $\delta(q, a) = (r, b, S)$, akkor $uqav \vdash urbv$,
- ▶ ha $\delta(q, a) = (r, b, L)$, akkor $uqav \vdash u'rcbv$, ahol $c \in \Gamma$ és $u'c = u$,
ha $u \neq \varepsilon$, különben $u' = u$ és $c = \sqcup$.

Turing gépek

Többlépéses konfigurációátmenet, felismert nyelv

Többlépéses konfigurációátmenet: \vdash reflexív, tranzitív lezártja, azaz:

$\vdash^* \subseteq C_M \times C_M$ **többlépéses konfigurációátmenet**

$C \vdash^* C' \Leftrightarrow$

- ▶ ha $C = C'$ vagy
- ▶ ha $\exists n > 0 \wedge C_1, C_2, \dots, C_n \in C_M$, hogy $\forall 1 \leq i \leq n - 1$ -re $C_i \vdash C_{i+1}$ valamint $C_1 = C$ és $C_n = C'$.

Az M TG által felismert nyelv

$L(M) = \{u \in \Sigma^* \mid q_0 u \sqsubset \vdash^* x q_i y \text{ valamely } x, y \in \Gamma^*, y \neq \varepsilon\}.$

Figyeljük meg, hogy $L(M)$ csak Σ feletti szavakat tartalmaz.

Turing gépek

A TG-ek és a nyelvek

Egy $L \subseteq \Sigma^*$ nyelv **Turing-felismerhető**, ha $L = L(M)$ valamely M TG-re.

Egy $L \subseteq \Sigma^*$ nyelv **eldönthető**, ha létezik olyan M TG, mely minden bemeneten megállási konfigurációba jut és $L(M) = L$.

A Turing-felismerhető nyelveket szokás **rekurzívan felsorolhatónak** (vagy *parciálisan rekurzívnak*, vagy *félíg eldönthetőnek*) az eldönthető nyelveket pedig **rekurzívnak** is nevezni.

A rekurzívan felsorolható nyelvek osztályát RE -vel, a rekurzív nyelvek osztályát pedig R -rel jelöljük.

Nilván $R \subseteq RE$. Igaz-e hogy $R \subset RE$?

Turing gépek

Futási idő

Egy M TG futási ideje (**időigénye**) az u szón n ($n \geq 0$), ha M az u -hoz tartozó kezdőkonfigurációból n lépésben (konfigurációátmenettel) jut el megállási konfigurációba. Ha nincs ilyen szám, akkor M futási ideje az u -n végtelen.

Legyen $f : \mathbb{N} \rightarrow \mathbb{N}$ egy függvény. Azt mondjuk, hogy M **időigénye** $f(n)$ (vagy, hogy M egy $f(n)$ időkorlátos gép), ha minden $u \in \Sigma^*$ input szóra, M időigénye az u szón legfeljebb $f(|u|)$.

Gyakran megelégszünk azzal, hogy a pontos időkorlát helyett jó aszimptotikus felső korlátot adunk az időigényre.

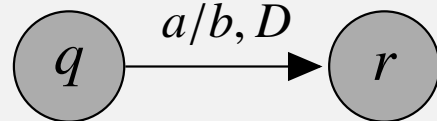
Turing gépek

Egy példa

Feladat: Készítsünk egy M Turing gépet, melyre

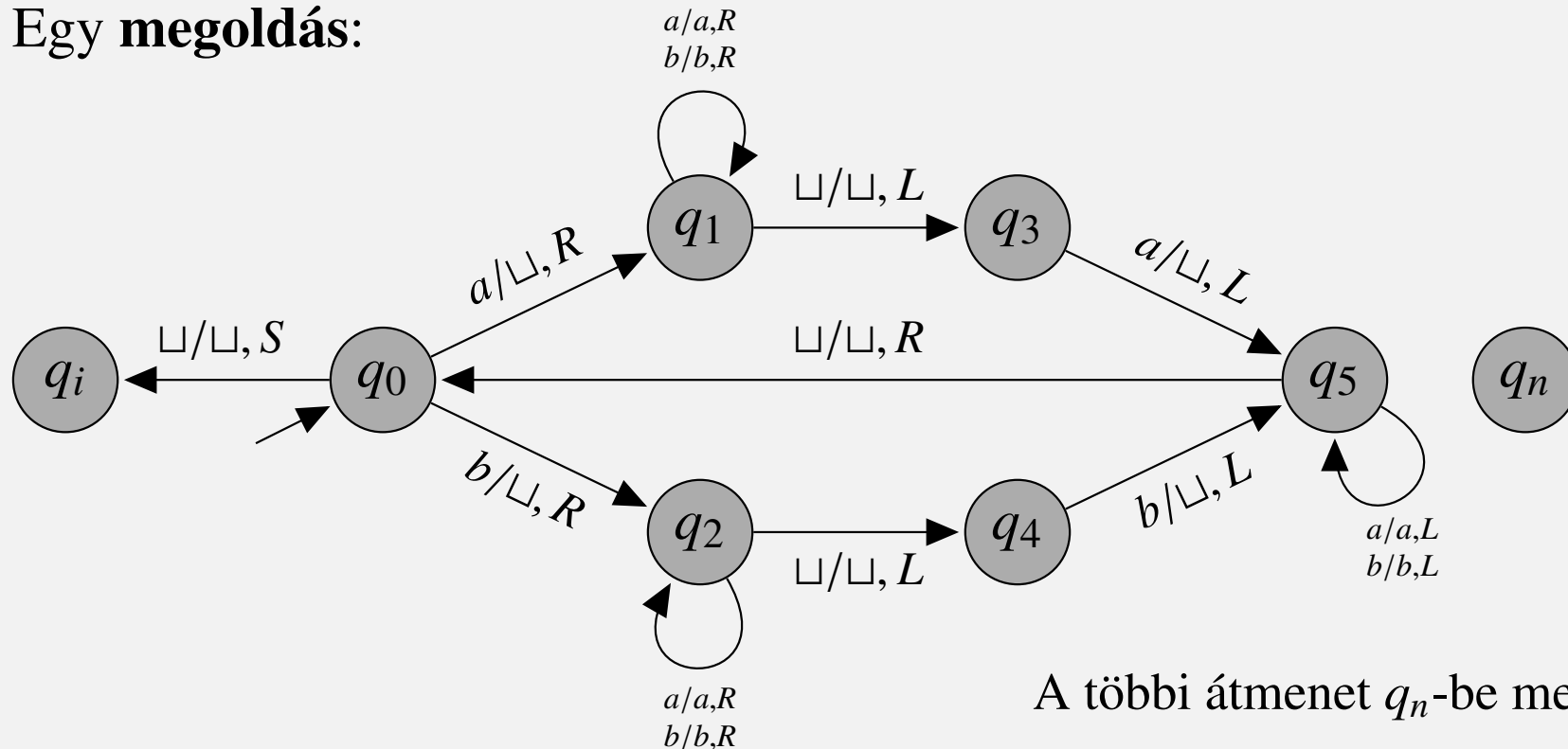
$$L(M) = \{ww^{-1} \mid w \in \{a, b\}^*\}!$$

Az átmenetdiagram.



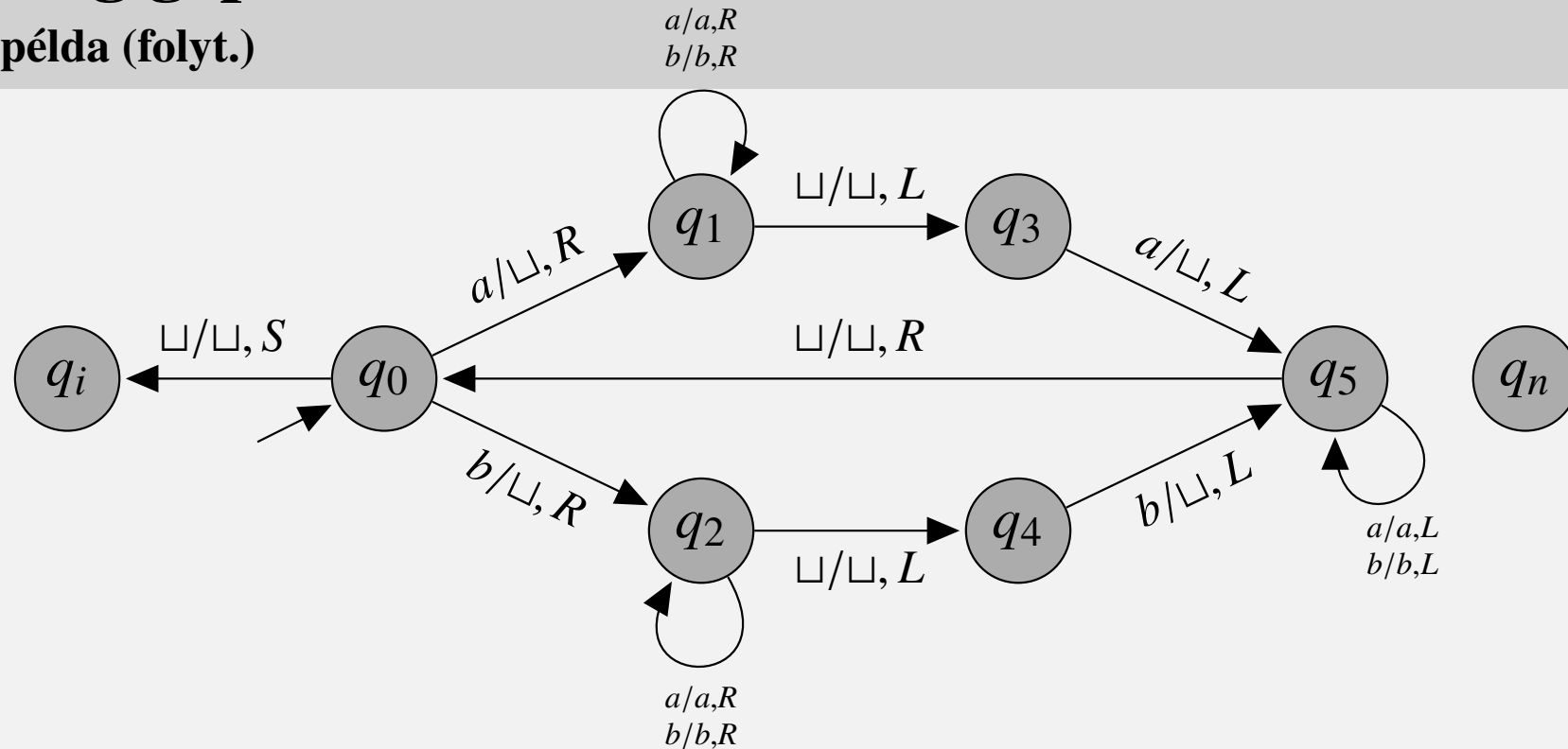
$\delta(q, a) = (r, b, D)$ jelölése
($q, r \in Q, a, b \in \Gamma, D \in \{L, S, R\}$)

Egy megoldás:



Turing gépek

Egy példa (folyt.)



Példa. Konfigurációátmenetek sorozata az *aba* inputra:

$q_0aba \vdash q_1ba \vdash bq_1a \vdash baq_1 \sqcup \vdash bq_3a \vdash q_5b \vdash q_5 \sqcup b \vdash q_0b \vdash q_2 \sqcup \vdash q_4 \sqcup \vdash q_n \sqcup$.

Az *aba* inputra 10 lépésben jut a gép megállási konfigurációba. Ebben a példában tetszőleges n -re ki tudjuk számolni a pontos időigényt is, de egyszerűbb (és gyakran elegendő) egy jó aszimptotikus felső korlát megadása.

Függvények aszimptotikus nagyságrendje

Definíció

Legyenek $f, g : \mathbb{N} \rightarrow \mathbb{R}_0^+$ függvények, ahol \mathbb{N} a természetes számok, \mathbb{R}_0^+ pedig a nemnegatív valós számok halmaza.

- ▶ f -nek g aszimptotikus felső korlátja (jelölése: $f(n) = O(g(n))$); ejtsd: $f(n)$ = nagyordó $g(n)$) ha létezik olyan $c > 0$ konstans és $N \in \mathbb{N}$ küszöbindex, hogy $f(n) \leq c \cdot g(n)$ minden $n \geq N$ -re.
- ▶ f -nek g aszimptotikus alsó korlátja (jelölése: $f(n) = \Omega(g(n))$) ha létezik olyan $c > 0$ konstans és $N \in \mathbb{N}$ küszöbindex, hogy $f(n) \geq c \cdot g(n)$ minden $n \geq N$ -re.
- ▶ f -nek g aszimptotikus éles korlátja (jelölése: $f(n) = \Theta(g(n))$) ha léteznek olyan $c_1, c_2 > 0$ konstansok és $N \in \mathbb{N}$ küszöbindex, hogy $c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$ minden $n \geq N$ -re.

Megjegyzés: a definíció könnyen kiterjeszthető aszimptotikusan nemnegatív, azaz egy korlát után nemnegatív függvényekre.

Függvények aszimptotikus nagyságrendje

Függvények aszimptotikus nagyságrend szerinti osztályozása

O , Ω , Θ 2-aritású relációnak is felfogható az $\mathbb{N} \rightarrow \mathbb{R}_0^+$ függvények univerzumán, ekkor

- ▶ O , Ω , Θ tranzitív (pl. $f = O(g)$, $g = O(h) \Rightarrow f = O(h)$)
- ▶ O , Ω , Θ reflexív
- ▶ Θ szimmetrikus
- ▶ O , Ω fordítottan szimmetrikus ($f = O(g) \Leftrightarrow g = \Omega(f)$)
- ▶ (köv.) Θ ekvivalenciareláció, az $\mathbb{N} \rightarrow \mathbb{R}_0^+$ függvények egy osztályozását adja. Az egyes függvényosztályokat általában "legegyszerűbb" tagjukkal reprezentáljuk. Pl. 1 (korlátos függvények), n (lineáris függvények), n^2 (négyzetes függvények), stb.

Függvények aszimptotikus nagyságrendje

Tételek

- ▶ $f, g = O(h) \Rightarrow f + g = O(h)$, hasonlóan Ω -ra, Θ -ra.
(Összeadásra való zártság)
- ▶ Legyen $c > 0$ konstans $f = O(g) \Rightarrow c \cdot f = O(g)$, hasonlóan Ω -ra, Θ -ra. (Pozitív konstanssal szorzásra való zártság)
- ▶ $f + g = \Theta(\max\{f, g\})$ (szekvencia tétele). A domináns tag határozza meg egy összeg aszimptotikus nagyságrendjét.
- ▶ Ha létezik az f/g határérték
 - ha $f(n)/g(n) \rightarrow +\infty \Rightarrow f(n) = \Omega(g(n))$ és $f(n) \neq O(g(n))$
 - ha $f(n)/g(n) \rightarrow c \quad (c > 0) \Rightarrow f(n) = \Theta(g(n))$
 - ha $f(n)/g(n) \rightarrow 0 \Rightarrow f(n) = O(g(n))$ és $f(n) \neq \Omega(g(n))$

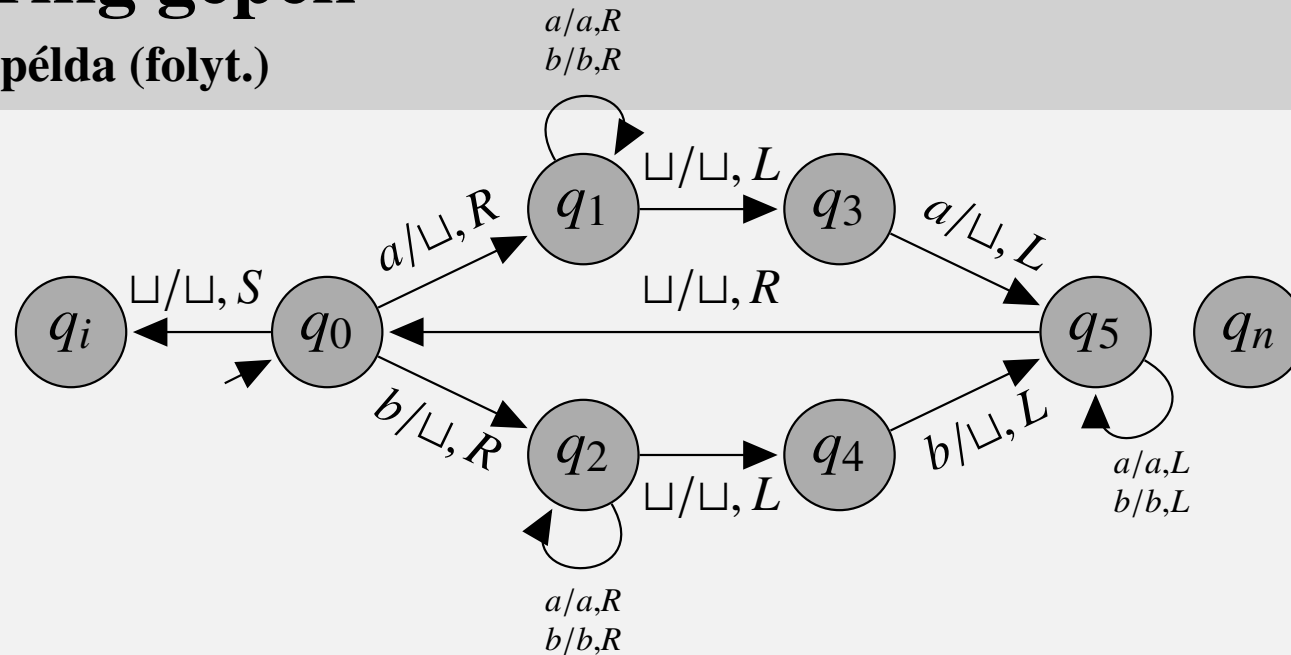
Függvények aszimptotikus nagyságrendje

Konkrét függvények

- ▶ $p(n) = a_k n^k + \dots + a_1 n + a_0$ ($a_k > 0$), ekkor $p(n) = \Theta(n^k)$,
- ▶ Minden $p(n)$ polinomra és $c > 1$ konstansra $p(n) = O(c^n)$, de $p(n) \neq \Omega(c^n)$,
- ▶ Minden $c > d > 1$ konstansokra $d^n = O(c^n)$, de $d^n \neq \Omega(c^n)$,
- ▶ Minden $a, b > 1$ -re $\log_a n = \Theta(\log_b n)$,
- ▶ Minden $c > 0$ -ra $\log n = O(n^c)$, de $\log n \neq \Omega(n^c)$.

Turing gépek

Egy példa (folyt.)



A TG időigénye $O(n^2)$, hiszen $O(n)$ iteráció mindegyikében $O(n)$ -et lépünk, +1 lépés q_i -be vagy q_n -be.

Van-e jobb aszimptotikus felső korlát? Nincs, mert van végtelen sok szó, melyre $\Omega(n^2)$ -et lép.

Eldönti-e az $L = \{ww^{-1} \mid w \in \{a, b\}^*\}$ nyelvet vagy "csak" felismeri? Eldönti.

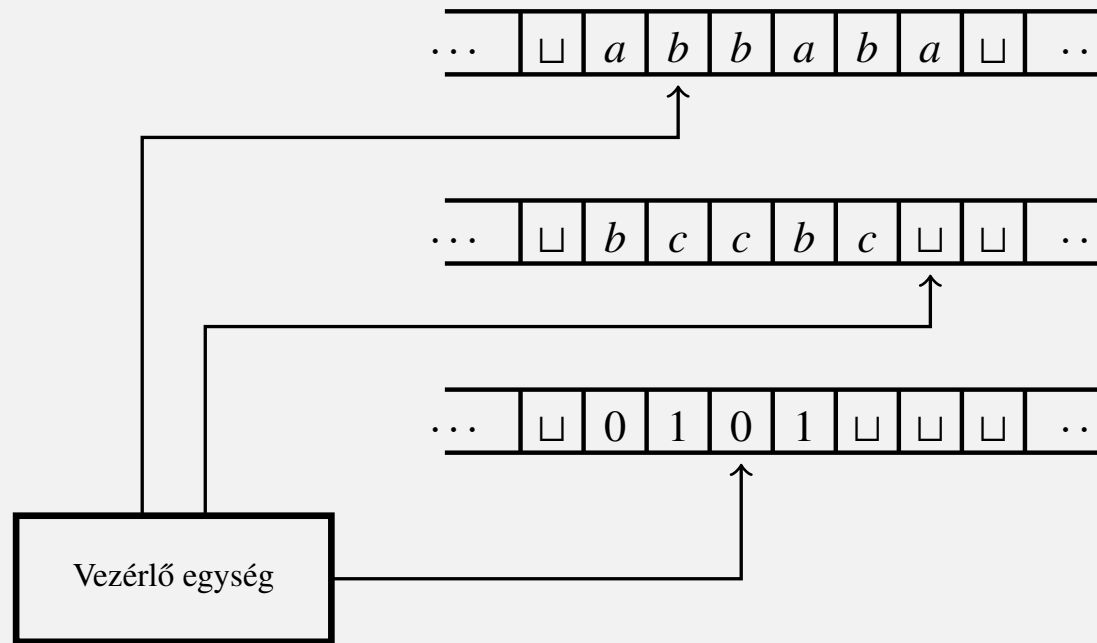
Van-e olyan TG, ami nem dönti el, de azért felismeri L -et? Igen, a q_n -be menő átmeneteket vezessük végtelen ciklusba.

Logika és számításelmélet

8. előadás

k -szalagos Turing-gép

Informális kép



- ▶ Egy ütem: Mind a k szalag olvasása, átírása és a fejek léptetése egyszerre, egymástól függetlenül.
- ▶ Az egyszalagos géppel analóg elfogadásfogalom.
- ▶ Az egyszalagos géppel analóg időigény fogalom.

k -szalagos Turing-gép

Definíció

k -szalagos Turing-gép

A k -szalagos Turing-gép egy olyan $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ rendszer, ahol

- ▶ Q az állapotok véges, nemüres halmaza,
- ▶ $q_0, q_i, q_n \in Q$, q_0 a kezdő- q_i az elfogadó- és q_n az elutasító állapot,
- ▶ Σ és Γ ábécék, a bemenő jelek illetve a szalagszimbólumok ábécéje úgy, hogy $\Sigma \subseteq \Gamma$ és $\sqcup \in \Gamma \setminus \Sigma$,
- ▶ $\delta : (Q \setminus \{q_i, q_n\}) \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{L, S, R\}^k$ az átmenet függvény.

k -szalagos Turing-gép

Konfigurációk

Konfiguráció

k -szalagos TG konfigurációja egy $(q, u_1, v_1, \dots, u_k, v_k)$ szó, ahol $q \in Q$ és $u_i, v_i \in \Gamma^*$, $v_i \neq \varepsilon$ ($1 \leq i \leq k$).

Ez azt reprezentálja, hogy az aktuális állapot q , az i . szalag tartalma $u_i v_i$ és az i . fej v_i első betűjén áll ($1 \leq i \leq k$).

Kezdőkonfiguráció

Az u szóhoz tartozó **kezdőkonfiguráció**: $u_i = \varepsilon$ ($1 \leq i \leq k$), $v_1 = u\sqcup$, és $v_i = \sqcup$ ($2 \leq i \leq k$).

[v_1 miért $u\sqcup$ és nem u ? Azért, hogy $u = \varepsilon$ ne legyen külön eset, és ugyanazt a szalagtartalmat reprezentálják.]

Elfogadó/elutasító/megállási konfiguráció

A $(q, u_1, v_1, \dots, u_k, v_k)$ konfiguráció, ahol $q \in Q$ és $u_i, v_i \in \Gamma^*$, $v_i \neq \varepsilon$ ($1 \leq i \leq k$), **elfogadó konfiguráció**, ha $q = q_i$, **elutasító konfiguráció**, ha $q = q_n$, **megállási konfiguráció**, ha $q = q_i$ vagy $q = q_n$.

k -szalagos Turing-gép

Egy- és többlépéses konfigurációátmenet

A k -szalagos Turing-gépek **egylépéses konfigurációátmenetét** az egyszalagos esettel analóg módon definiálhatjuk. Mivel túl sok eset van (a lehetséges 3^k irány- k -as miatt) ezért ezt csak egy példán keresztül nézzük meg. Jelölés: \vdash .

Legyen $k=2$ és $\delta(q, a_1, a_2) = (r, b_1, b_2, R, S)$ a TG egy átmenete. Ekkor $(q, u_1, a_1v_1, u_2, a_2v_2) \vdash (r, u_1b_1, v'_1, u_2, b_2v_2)$, ahol $v'_1 = v_1$, ha $v_1 \neq \varepsilon$, különben $v'_1 = \sqcup$.

Vegyük észre, hogy a fejek nem kell hogy szinkronban lépjenek, egymástól függetlenül mozoghatnak.

Ezek után a **többlépéses konfigurációátmenet** definíciója megegyezik az egyszalagos esetnél tárgyalttal. Jelölés: \vdash^* .

k -szalagos Turing-gép

Felismert nyelv, időigény

k -szalagos Turing-gép által felismert nyelv

$$L(M) = \{u \in \Sigma^* \mid (q_0, \varepsilon, u\sqcup, \varepsilon, \sqcup, \dots, \varepsilon, \sqcup) \vdash^* (q_i, x_1, y_1, \dots, x_k, y_k), x_1, y_1, \dots, x_k, y_k \in \Gamma^*, y_1, \dots, y_k \neq \varepsilon\}.$$

Azaz, csakúgy mint az egyszalagos esetben, azon inputábécé feletti szavak halmaza, melyekkel a TG-et indítva az az elfogadó, q_i állapotában áll le.

A k -szalagos TG-ek által **felismerhető** illetve **eldönthető** nyelv fogalma szintén analóg az egyszalagos esettel.

k -szalagos Turing-gép futási ideje adott szóra

Egy k -szalagos Turing-gép **futási ideje** egy u szóra a hozzá tartozó kezdőkonfigurációból egy megállási konfigurációba megtett lépések száma.

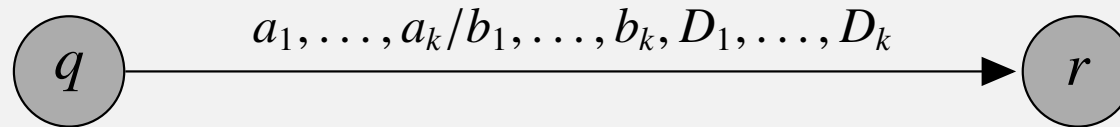
Ezek után az **időigény** definíciója megegyezik az egyszalagos esetnél tárgyalttal.

k -szalagos Turing-gép

Egy példa

Feladat: Készítsünk egy M kétszalagos Turing gépet, melyre
 $L(M) = \{ww^{-1} \mid w \in \{a, b\}^*\}$!

Az átmenetdiagram.



$\delta(q, a_1, \dots, a_k) = (r, b_1, \dots, b_k, D_1, \dots, D_k)$ jelölése
($q, r \in Q, a_1, \dots, a_k, b_1, \dots, b_k \in \Gamma, D_1, \dots, D_k \in \{L, S, R\}$)

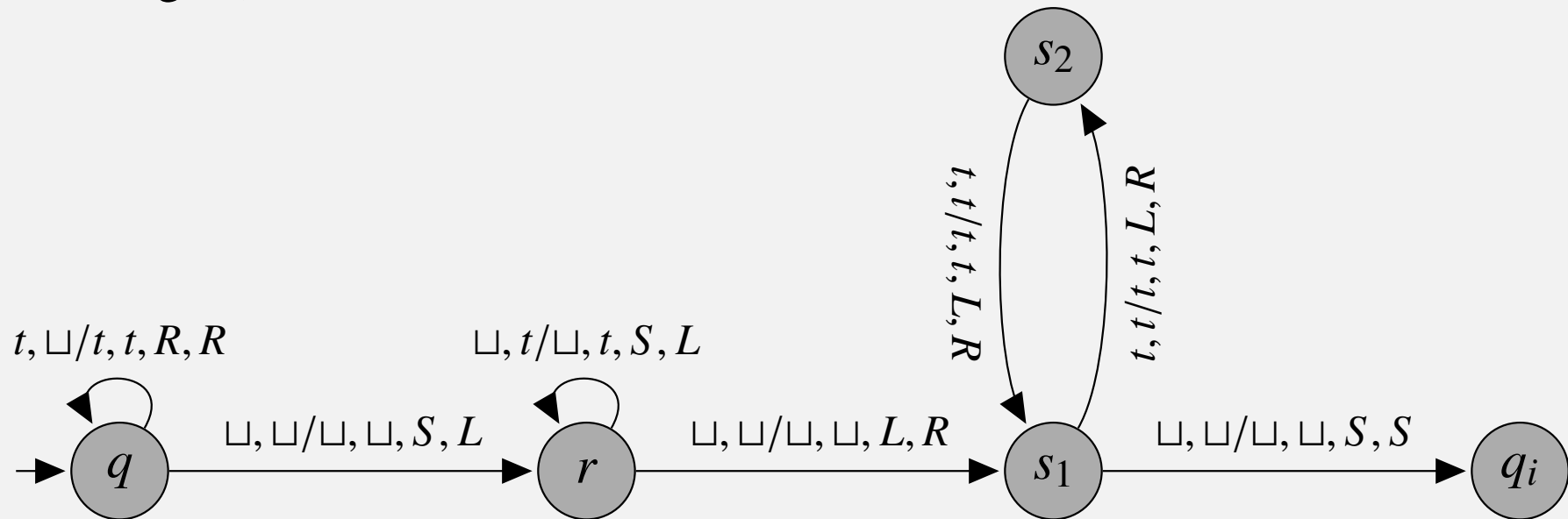
k -szalagos Turing-gép

Egy példa

Feladat: Készítsünk egy M kétszalagos Turing gépet, melyre $L(M) = \{ww^{-1} \mid w \in \{a, b\}^*\}$!

Egy megoldás:

(A többi átmenet q_n -be megy. Minden átmenetre $t \in \{a, b\}$ tetszőleges.)



Mennyi a TG időigénye? Ez egy $O(n)$ időkorlátos TG.

k -szalagos Turing-gép

Szimulálás egy szalaggal

Ekvivalens TG-ek

Két TG **ekvivalens**, ha ugyanazt a nyelvet ismerik fel.

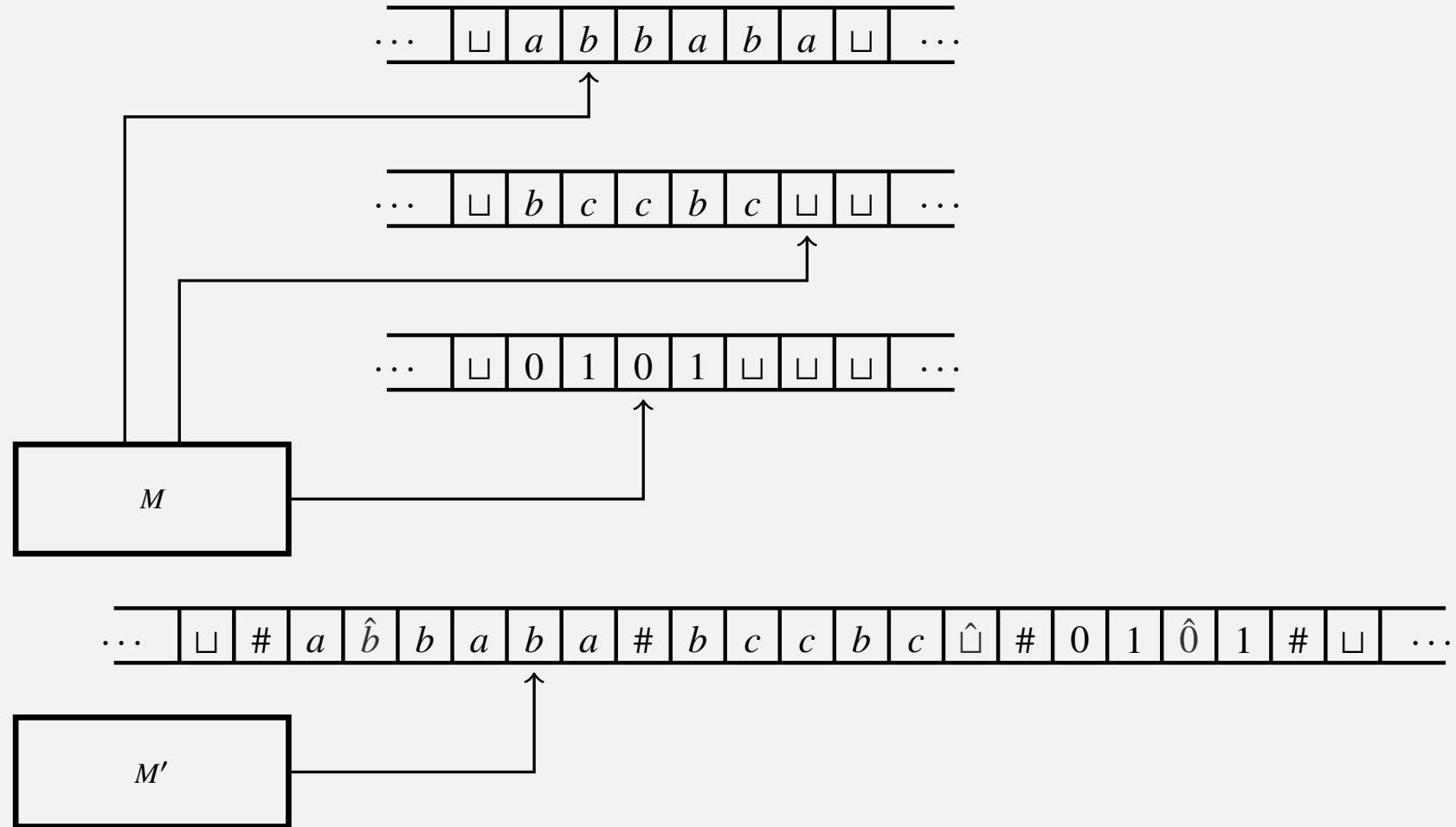
Tétel

Minden M k -szalagos Turing-géphez megadható egy vele ekvivalens M' egyszalagos Turing-gép. Továbbá, ha M legalább lineáris időigényű $f(n)$ időkorlátos gép (azaz $f(n) \geq n$), akkor M' $O(f(n)^2)$ időkorlátos.

k -szalagos Turing-gép

Szimulálás egy szalaggal

Bizonyítás (vázlat): A szimuláció alapötlete



k -szalagos Turing-gép

Szimulálás egy szalaggal

A szimuláció menete egy $a_1 \cdots a_n$ bemeneten:

1. M' kezdőkonfigurációja legyen $q'_0 \# \hat{a}_1 a_2 \cdots a_n \# \hat{\sqcup} \# \cdots \hat{\sqcup} \#$
2. M' először végigmegy a szalagon (számolja a #-okat) és eltárolja a $\hat{\cdot}$ -pal megjelölt szimbólumokat az állapotában
3. M' mégegyszer végigmegy a szalagján és M átmenetfüggvénye alapján aktualizálja azt
4. ha M valamelyik szalagján nő a szó hozza, akkor M' -nek az adott ponttól mozgatnia kell a szalagja tartalmát jobbra
5. Ha M elfogadó vagy elutasító állapotba lép, akkor M' is belép a saját elfogadó vagy elutasító állapotába
6. Egyébként M' folytatja a szimulációt a 2-ik ponttal

k -szalagos Turing-gép

Szimulálás egy szalaggal – időigény

Meggondolható, hogy M egyetlen lépésének szimulálásakor

- ▶ a lépések számára aszimptotikus felső korlát az M' által addig felhasznált cellaterület (tár). (Kétszer végigmegy M' szalagján, legfeljebb k -szor kell egy \sqcup -nek helyet csinálni, ami szintén $O(\text{felhasznált cellaterület})$)
- ▶ a felhasznált cellaterület $O(1)$ -el nőtt. ($\leq k$ -val, hiszen $\leq k$ -szor kell egy \sqcup -t beszúrni)

Az M' által felhasznált cellaterület mérete kezdetben $\Theta(n)$, lépésenként $O(1)$ -gyel nőhet, így $O(n + f(n)O(1)) = O(n + f(n))$ közös, minden lépés után igaz aszimptotikus felső korlát az M' által felhasznált cellaterület méretére.

Tehát M minden egyes lépésének szimulációja $O(n + f(n))$ M' -beli lépés.

Így M' összesen $f(n) \cdot O(n + f(n))$ időkorlátos, ami $O(f(n)^2)$, ha $f(n) = \Omega(n)$.

Turing-gép egy irányban végtelen szalaggal

- ▶ Az egy irányban végtelen szalagos Turing-gép egy, a bal oldalán zárt szalaggal rendelkezik
- ▶ A fej nem tud "leesni" a bal oldalon, még ha az állapot-átmeneti függvény balra lépést ír is elő a legbaloldalibb cellán, ilyenkor a fej helyben marad.

Tétel

Minden egyszalagos M Turing-géphez van vele ekvivalens egyirányban végtelen szalagos M'' Turing-gép.

Turing-gép egy irányban végtelen szalaggal

Tétel

Minden egyszalagos M Turing-géphez van vele ekvivalens egyirányban végtelen szalagos M'' Turing-gép.

Bizonyítás (vázlat):

1. Szimuláljuk M -et egy olyan M' TG-pel, ami két darab egy irányban végtelen szalaggal rendelkezik:
 M' megjelöli mindkét szalagjának első celláját egy speciális szimbólummal. Ezután M
 - az első szalagján szimulálja M -et akkor, amikor az a fej kezdőpozícióján vagy attól jobbra dolgozik,
 - a második szalagján pedig akkor, amikor az M a fej kezdőpozíciótól balra dolgozik (ezen a szalagon az ettől a pozíciótól balra lévő szó tükörképe van)
2. Szimuláljuk M' -t egy egyirányban végtelen szalagos M'' Turing-géppel (az előző tételben látott bizonyításhoz hasonlóan)

Nemdeterminisztikus Turing-gép

definíció, egylépéses konfigurációátmenet

Jelölje $\mathcal{P}(X) = \{Y \mid Y \subseteq X\}$ az X halmaz hatványhalmazát.

Nemdeterminisztikus Turing-gép (NTG)

A nemdeterminisztikus Turing-gép (NTG) olyan $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ rendszer, ahol

- ▶ $Q, \Sigma, \Gamma, q_0, q_i, q_n$ ugyanaz, mint eddig
- ▶ $\delta : (Q \setminus \{q_i, q_n\}) \times \Gamma \rightarrow \mathcal{P}(Q \times \Gamma \times \{L, S, R\})$

Konfigurációk C_M halmazának fogalma azonos.

$\vdash \subseteq C_M \times C_M$ egylépéses konfigurációátmenet

Legyen $uqav$ egy konfiguráció, ahol $a \in \Gamma$, $u, v \in \Gamma^*$.

- ▶ Ha $(r, b, R) \in \delta(q, a)$, akkor $uqav \vdash ubrv'$, ahol $v' = v$, ha $v \neq \varepsilon$, különben $v' = \sqcup$,
- ▶ ha $(r, b, S) \in \delta(q, a)$, akkor $uqav \vdash urbv$,
- ▶ ha $(r, b, L) \in \delta(q, a)$, akkor $uqav \vdash u'rcbv$, ahol $c \in \Gamma$ és $u'c = u$, ha $u \neq \varepsilon$ különben $u' = u$ és $c = \sqcup$

Nemdeterminisztikus Turing-gép

többlépéses konfigurációátmenet, felismert nyelv

Többlépéses konfigurációátmenet: \vdash reflexív tranzitív lezártja, azaz:

$\vdash^* \subseteq C_M \times C_M$ **többlépéses konfigurációátmenet**

$C \vdash^* C' \Leftrightarrow$

- ▶ ha $C = C'$ vagy
- ▶ ha $\exists n > 0 \wedge C_1, C_2, \dots, C_n \in C_M$, hogy $\forall 1 \leq i \leq n - 1$ -re $C_i \vdash C_{i+1}$ valamint $C_1 = C$ és $C_n = C'$.

NTG által felismert nyelv

$L(M) = \{u \in \Sigma^* \mid q_0 u \sqcup \vdash^* x q_i y \text{ valamely } x, y \in \Gamma^*, y \neq \varepsilon\}.$

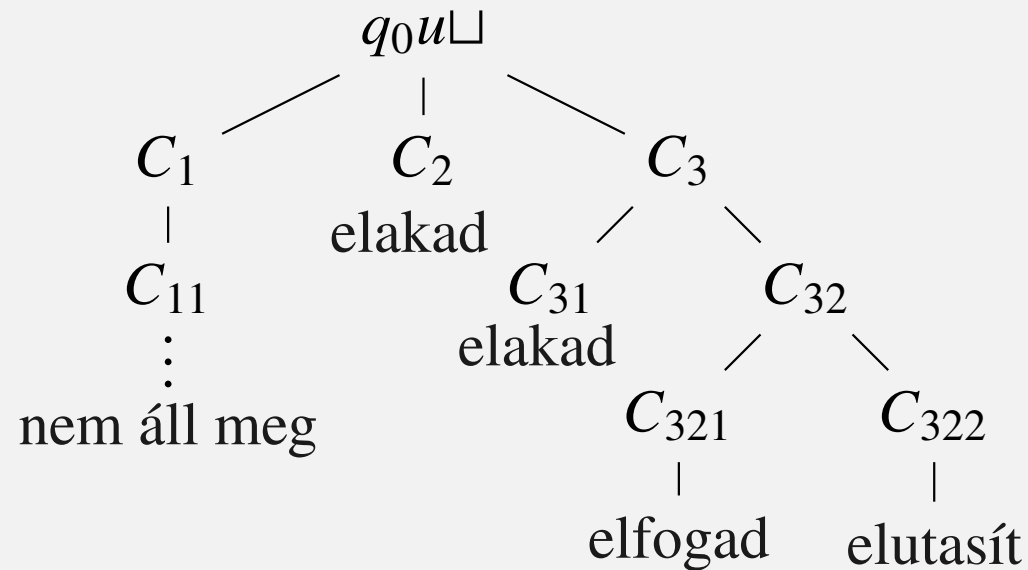
Egy NTG-re úgy gondolhatunk, hogy több számítása is lehet ugyanarra a szóra. Akkor fogad el egy szót, ha legalább egy számítása q_i -ben ér véget.

Nemdeterminisztikus Turing-gép

Nemdeterminisztikus számítási fa

$u \in \Sigma^*$ nemdeterminisztikus számítási fája

Irányított fa, melynek csúcsai konfigurációkkal címkézettek. $q_0u\sqcup$ a gyökér címkéje. Ha C egy csúcs címkéje, akkor $\{|C' \mid C \vdash C'\}$ gyereke van és ezek címkéi éppen $\{C' \mid C \vdash C'\}$ elemei.



Példa:

Elfogadja u -t, hiszen a $q_0u\sqcup \vdash C_3 \vdash C_{32} \vdash C_{321}$ számítása elfogadó konfigurációba visz. Egy ilyen számítás is elég az elfogadáshoz.

Nemdeterminisztikus Turing-gép

NTG-vel való eldönthetőség, időigény

M **eldönti** az $L \subseteq \Sigma^*$ nyelvet, ha felismeri és minden $u \in \Sigma^*$ szóra az M számítási fája véges és minden levele elfogadó vagy elutasító konfiguráció.

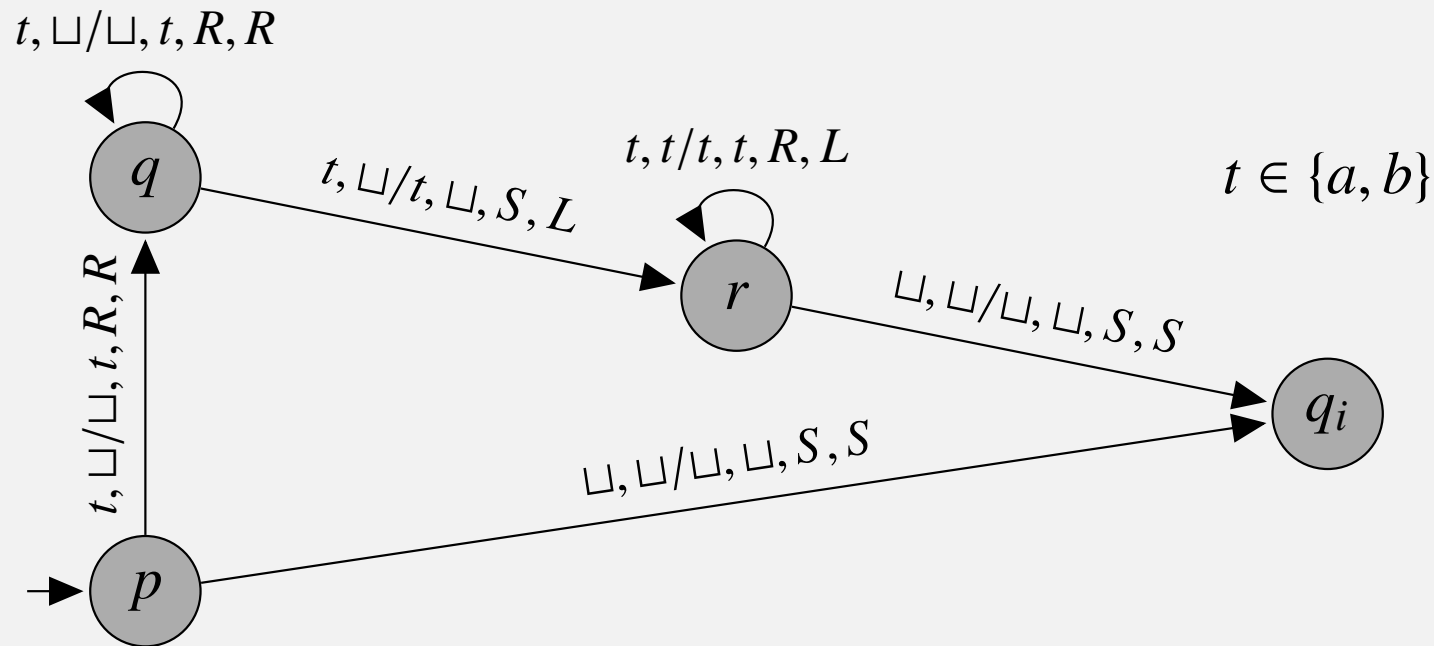
M $f(n)$ **időkorlátos** (időigényű), ha minden $u \in \Sigma^*$ n hosszú szóra u számítási fája legfeljebb $f(n)$ magas.

Megjegyzés: a nemdeterminisztikus Turing-gép definíciója értelemszerűen kiterjeszthető k -szalagos gépekre is, így beszélhetünk k -szalagos nemdeterminisztikus Turing-gépekről is.

Nemdeterminisztikus Turing-gép

Példa

Feladat: Készítsünk egy M nemdeterminisztikus Turing-gépet, melyre $L(M) = \{ww^{-1} \mid w \in \{a, b\}^*\}$!



$(p, \varepsilon, abba, \varepsilon, \sqcup) \vdash (q, \varepsilon, bba, a, \sqcup) \vdash (r, \varepsilon, bba, \varepsilon, a) \vdash (q_i, \varepsilon, bba, \varepsilon, a)$

$(p, \varepsilon, abba, \varepsilon, \sqcup) \vdash (q, \varepsilon, bba, a, \sqcup) \vdash (q, \varepsilon, ba, ab, \sqcup) \vdash (r, \varepsilon, ba, a, b) \vdash (r, b, a, \varepsilon, ab) \vdash (r, ba, \sqcup, \varepsilon, \sqcup ab) \vdash (q_i, ba, \sqcup, \varepsilon, \sqcup ab)$

Nemdeterminisztikus Turing-gép

Szimulálás determinisztikus TG-pel

Tétel

Minden $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ $f(n)$ idejű NTG-hez megadható egy ekvivalens, $2^{O(f(n))}$ idejű M' determinisztikus TG.

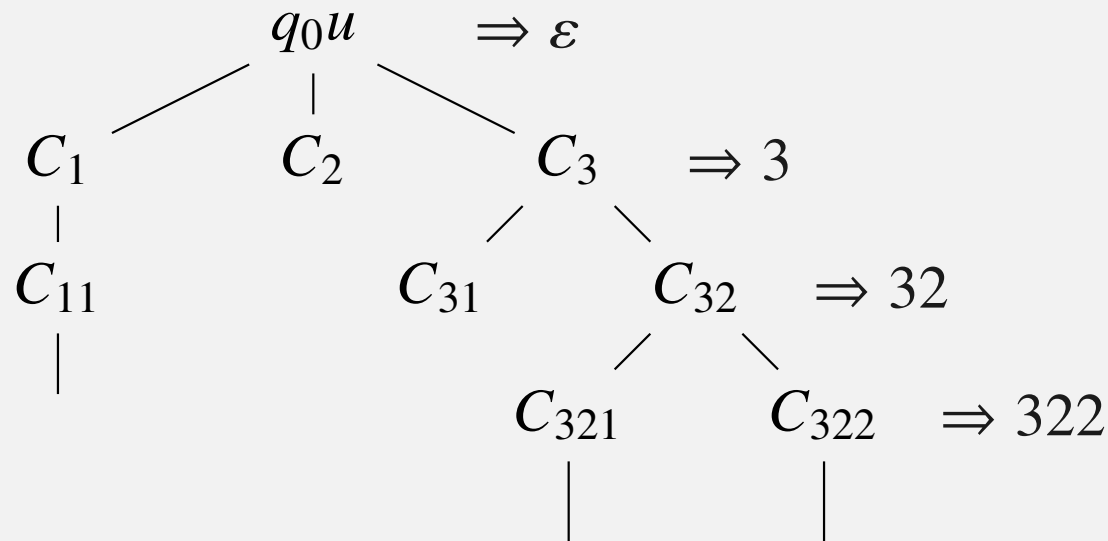
Bizonyítás (vázlat): M' egy adott $u \in \Sigma^*$ bemeneten szimulálja u M -beli összes számítását számítási fájának szélességi keresése által.

- ▶ Legyen d az M átmenetfüggvényének jobb oldalán szereplő halmazok számosságának a maximuma, azaz
$$d = \max_{(q,a) \in Q \times \Gamma} |\delta(q, a)|.$$
- ▶ Legyen $T = \{1, 2, \dots, d\}$ egy ábécé.
- ▶ minden $(q, a) \in Q \times \Gamma$ esetén rögzítsük le $\delta(q, a)$ elemeinek egy sorrendjét

Nemdeterminisztikus Turing-gép

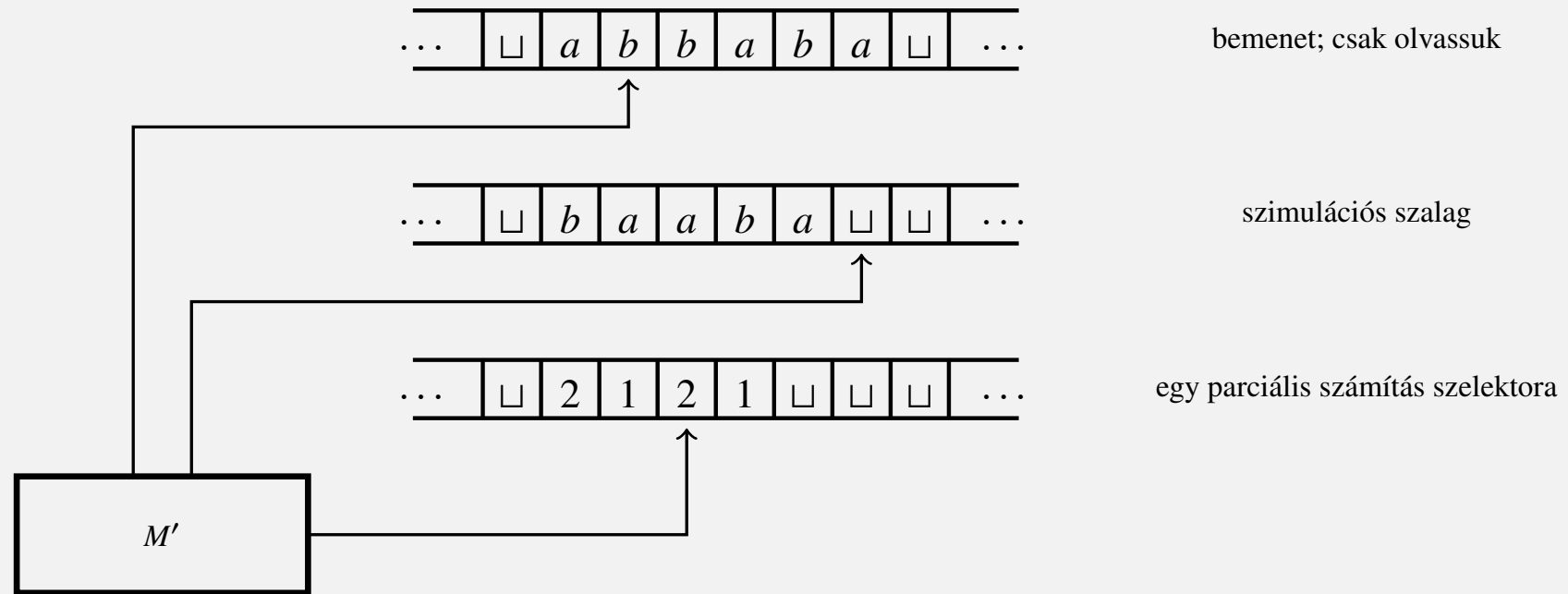
Szimulálás determinisztikus TG-pel

A számítási fa minden csúcsához egyértelműen hozzárendelhető egy T^* -beli szó, az adott konfigurációhoz tartozó parciális konfigurációátmenet-sorozat szelektora.



Nemdeterminisztikus Turing-gép

Szimulálás determinisztikus TG-pel



M' működése:

Nemdeterminisztikus Turing-gép

Szimulálás determinisztikus TG-pel

- ▶ M' kezdőkonfigurációja: az 1-es szalag tartalmazza a bemenetet, a 2-es és 3-as szalagok üresek.
- ▶ Amíg nincs elfogadás
 - M' rámásolja az 1-es szalag tartalmát a 2-esre
 - Amíg a 3-ik szalagon a fej nem \sqcup -re mutat
 - Legyen k a 3-ik szalagon a fej pozíciójában lévő betű
 - Legyen a 2-ik szalagon a fej pozíciójában lévő betű a és a szimulált M aktuális állapota q
 - Ha $\delta(q, a)$ -nak van k -ik eleme, akkor
 - M' szimulálja M egy lépését ezen elem szerint
 - Ha ez q_i -be vezet, akkor M' is elfogad
 - Ha ez q_n -be vezet, akkor M' kilép ebből a ciklusból
 - M' a 3-ik szalagon eggyel jobbra lép
 - M' törli a 2. szalagot és előállítja a 3. szalagon a hossz-lexikografikus (shortlex) rendezés szerinti következő szót T felett

Nemdeterminisztikus Turing-gép

Szimulálás determinisztikus TG-pel

- ▶ M' akkor és csak akkor lép elfogadó állapotba, ha a szimulált M elfogadó állapotba lép, azaz a két gép ekvivalens
- ▶ M' -nek $f(n)$ -ben exponenciálisan sok számítást kell megvizsgálnia (\leq egy $f(n)$ magasságú teljes d -áris fa belső csúcsainak száma, ami $O(d^{f(n)})$), azaz M' időigénye $2^{O(f(n))}$

Megjegyzés:

- ▶ Abból, hogy a bizonyításban alkalmazott szimuláció exponenciális időigényű még nem következik, hogy nincs hatékonyabb szimuláció.
- ▶ Az a *sejtés*, hogy nem lehet a nemdeterminisztikus Turing-gépet az időigény drasztikus romlása nélkül determinisztikus Turing-géppel szimulálni.

Logika és számításelmélet

9. előadás

Számosság

(ismétlés), definíció

A véges halmazok fontos tulajdonsága a méretük (\Rightarrow *természetes számok* fogalma). Cél: ennek kiterjesztése végtelen halmazokra. Egy ilyen általánosítás a **számosság** (*G. Cantor, 1845-1918*).

Halmazok számossága

- ▶ A és B halmazoknak megegyezik a számossága, ha létezik bijekció köztük. Jelölése: $|A| = |B|$.
- ▶ A számossága legalább annyi, mint B számossága, ha van B -ből injekció A -ba. Jelölése: $|A| \geq |B|$.
- ▶ A számossága nagyobb, mint B számossága, ha van B -ből injekció A -ba, de bijekció nincs. Jelölése: $|A| > |B|$.

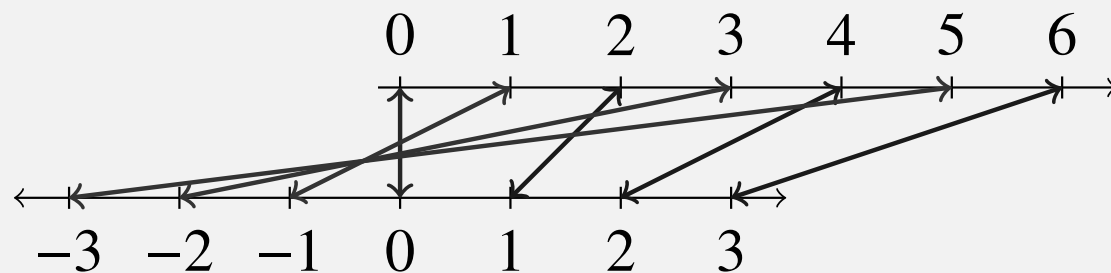
Cantor-Bernstein tétel

Ha A -ból B -be van injekció és B -ből A -ba is van, akkor A és B között bijekció is van, azaz ha $|A| \leq |B|$ és $|A| \geq |B|$, akkor $|A| = |B|$.

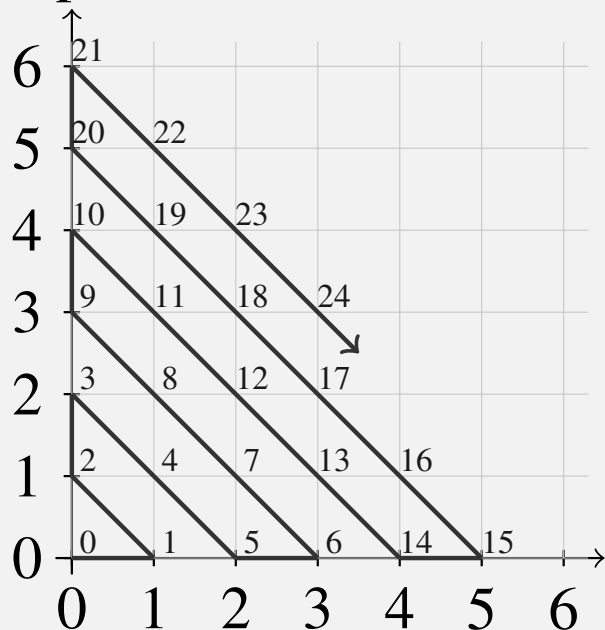
Számosság

Példák

1. Példa: $|\mathbb{N}| = |\mathbb{Z}|$.



2. példa: $|\mathbb{N}| = |\mathbb{N} \times \mathbb{N}|$.



Számosság

További példa; a megszámlálhatóan végtelen számosság

3. példa: $|\mathbb{N}| = |\mathbb{Q}|$.

Bizonyítás:

$\mathbb{N} \subset \mathbb{Q}$, ezért $|\mathbb{N}| \leq |\mathbb{Q}|$.

$\mathbb{Q}^+ := \{\frac{p}{q} \mid p \in \mathbb{N}^+, q \in \mathbb{N}^+, \text{ a tört nem egyszerűsíthető}\}$.

$\mathbb{Q}^- := \{-\frac{p}{q} \mid p \in \mathbb{N}^+, q \in \mathbb{N}^+, \text{ a tört nem egyszerűsíthető}\}$.

$\frac{p}{q} \in \mathbb{Q}^+ \mapsto (p, q) \in \mathbb{N} \times \mathbb{N}$ injektív, tehát $|\mathbb{Q}^+| \leq |\mathbb{N} \times \mathbb{N}| = |\mathbb{N}|$.

Legyen $\mathbb{Q}^+ = \{a_1, a_2, \dots\}$, $\mathbb{Q}^- = \{b_1, b_2, \dots\}$, ekkor

$\mathbb{Q} = \{0, a_1, b_1, a_2, b_2, \dots\}$

Megszámlálhatóan végtelen számosság

\mathbb{N} számosságát **megszámlálhatóan végtelennek** nevezzük. Egy halmaz **megszámlálható**, ha véges vagy megszámlálhatóan végtelen.

Tétel

Megszámlálható sok megszámlálható halmaz uniója megszámlálható.

Számosság

A continuum számosság

Van-e más végtelen számosság a megszámlálhatóan végtelenen kívül?

Igen, $|\mathbb{R}| > |\mathbb{N}|$.

Continuum számosság

\mathbb{R} számosságát **continuumnak** nevezzük.

4. példa: $|\mathbb{R}| = |(0, 1)|$.

$\tan(\pi(x - \frac{1}{2}))\big|_{(0,1)} : (0, 1) \rightarrow \mathbb{R}$ bijekció $(0, 1)$ és \mathbb{R} között.

Megjegyzés: $|\mathbb{R}| = |(a, b)| = |[c, d]|$ és $|\mathbb{R}| = |\mathbb{R}^n|$.

Számosság

Szavakkal kapcsolatos számosságok

5. Példa: $|\{0, 1\}^*| = |\mathbb{N}|$.

A hossz-lexikografikus (shortlex) rendezés egy bijekció:

$\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, 0000, \dots$

6. Példa

$$|\{L \mid L \subseteq \{0, 1\}^*\}| = |\{(b_1, \dots, b_i, \dots) \mid b_i \in \{0, 1\}, i \in \mathbb{N}\}|$$

Természetes bijekció van köztük:

Soroljuk fel a bináris szavakat a hossz-lexikografikus rendezés szerint.

Egy nyelvhez rendeljük azt a megszámlálhatóan végtelen hosszúságú bitsorozatot, melynek 1 az i . bitje, ha benne van az i . szó, 0 ha nem (a nyelv *karakterisztikus vektorát*).

Jelöljük a jobboldali halmazt $\{0, 1\}^{\mathbb{N}}$ -nel.

Számosság

Szavakkal kapcsolatos számosságok

7. Példa $|\{0, 1\}^{\mathbb{N}}| = |[0, 1)|$.

Bizonyítás (vázlat):

$x \in [0, 1)$ -hez rendeljük hozzá x kettedestört alakjának "0." utáni részét (ha kettő van akkor az egyiket). Injektív, így $|[0, 1)| \leq |\{0, 1\}^{\mathbb{N}}|$.

$\mathbf{z} \in \{0, 1\}^{\mathbb{N}}$ minden 1-esét helyettesítsük 2-essel, írjuk elé "0."-t és tekintsük harmadostörtnek. Meggondolható, hogy ez injektív megfeleltetés, így $|\{0, 1\}^{\mathbb{N}}| \leq |[0, 1)|$.

A Cantor-Bernstein tétel alapján $|\{0, 1\}^{\mathbb{N}}| = |[0, 1)|$.

Számosság

Cantor-féle átlós módszer

Állítás: $|\{0, 1\}^{\mathbb{N}}| > |\mathbb{N}|$

Bizonyítás:

$|\{0, 1\}^{\mathbb{N}}| \geq |\mathbb{N}|$:

$H_0 := \{(1, 0, 0, 0, \dots), (0, 1, 0, 0, \dots), (0, 0, 1, 0, \dots), \dots\}$

$H_0 \subset \{0, 1\}^{\mathbb{N}}$, és $|H_0| = |\mathbb{N}|$.

Kell: $|\{0, 1\}^{\mathbb{N}}| \neq |\mathbb{N}|$.

Számosság

Cantor-féle átlós módszer

Állítás: $|\{0, 1\}^{\mathbb{N}}| > |\mathbb{N}|$

Indirekt tegyük fel, hogy $|\{0, 1\}^{\mathbb{N}}| = |\mathbb{N}|$. Ez azt jelenti, hogy bijekcióba lehet állítani $\{0, 1\}^{\mathbb{N}}$ elemeit \mathbb{N} elemeivel, azaz $\{0, 1\}^{\mathbb{N}} = \{u_i \mid i \in \mathbb{N}\} = \{u_1, u_2, \dots\}$ a $\{0, 1\}^{\mathbb{N}}$ elemeinek egy felsorolása (a természetes számokkal való megindexelése).

Legyen $u_i = (u_{i,1}, u_{i,2}, \dots, u_{i,j}, \dots)$, ahol minden $i, j \in \mathbb{N}$ -re $u_{i,j} \in \{0, 1\}$.

Tekintsük az $u = (\overline{u_{1,1}}, \overline{u_{2,2}}, \dots, \overline{u_{i,i}}, \dots)$ megszámlálhatóan végtelen hosszúságú bináris (azaz $\{0, 1\}^{\mathbb{N}}$ -beli) szót, ahol $\overline{b} = 0$, ha $b = 1$ és $\overline{b} = 1$, ha $b = 0$.

Mivel, minden megszámlálhatóan végtelen hosszúságú bináris szó fel van sorolva, ezért létezik olyan $k \in \mathbb{N}$, melyre $u = u_k$.

Ekkor u k .bitje $u_{k,k}$ (így jelöltük u_k k . bitjét), másrészt $\overline{u_{k,k}}$ (így definiáltuk u -t).

De ez nem lehetséges, tehát az indirekt feltevésünk, azaz hogy $|\{0, 1\}^{\mathbb{N}}| = |\mathbb{N}|$ hamis.

Számosság

Cantor-féle átlós módszer

1. Következmény

A continuum számosság nagyobb, mint a megszámlálhatóan végtelen számosság.

2. Következmény

Több $\{0, 1\}$ feletti nyelv van mint $\{0, 1\}$ feletti szó. (Számosság értelemben.)

Számosság

Cantor-féle átlós módszer

Megjegyzés $\{L \mid L \subseteq \{0, 1\}^*\} = \mathcal{P}(\{0, 1\}^*)$. Igaz-e általában, hogy $|\mathcal{P}(H)| > |H|$?

Tétel

Minden H halmazra $|\mathcal{P}(H)| > |H|$.

Bizonyítás: $|\mathcal{P}(H)| \geq |H|$, hiszen $\{\{h\} \mid h \in H\} \subseteq \mathcal{P}(H)$.

$|\mathcal{P}(H)| \neq |H|$: Cantor-féle átlós módszerrel:

Indirekt $f : \mathcal{P}(H) \leftrightarrow H$ bijekció. Definiálunk egy $A \subseteq H$ halmazt:

$$\forall x \in H : x \in A \Leftrightarrow x \notin f^{-1}(x)$$

$f(A) \in A$ igaz-e? Ha igen, $f(A) \notin A$, ha nem $f(A) \in A$, tehát $f(A)$ se az A halmazban, se azon kívül nincs, ellentmondás.

Számítási feladatok megoldása TG-pel

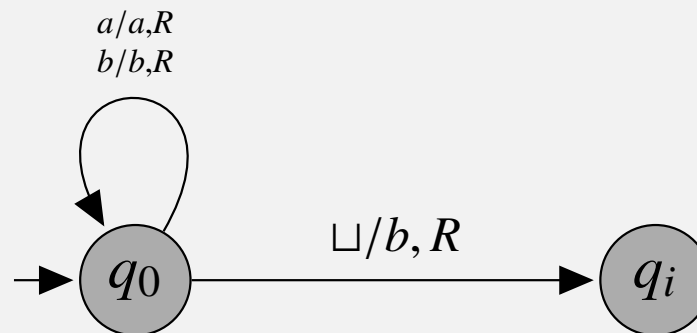
Használhatjuk a TG-eket szófüggvények kiszámítására is. A számítási feladatok megadhatók szófüggvényként.

Szófüggvényt kiszámító TG

Azt mondjuk, hogy az $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, (q_n) \rangle$ TG kiszámítja az $f : \Sigma^* \rightarrow \Delta^*$ szófüggvényt, ha minden $u \in \Sigma^*$ -beli szóra megáll, és ekkor $f(u) \in \Delta^*$ olvasható az utolsó szalagján.

Megjegyzés: Nincs szükség q_i és q_n megkülönböztetésére, elég lenne egyetlen megállási állapot. [Ezért van q_n ()-ben.]

Példa: $f(u) = ub$ ($u \in \{a, b\}^*$).



Problémák, mint formális nyelvek

Ha egy problémának megszámlálható sok lehetséges bemenete van (a hétköznapi problémák gyakorlatilag ilyenek), akkor a bemeneteket elkódolhatjuk egy véges ábécé felett.

Fontos-e, hogy mekkora ezen ábécé mérete? Egy d méretű ábécé esetén az első n bemenet elkódolásához nagyjából $\log_d n$ hosszú szavak kellenek. Mivel $\log_d n = \Theta(\log_{d'} n)$, ha $d, d' \geq 2$, ezért a válasz az, hogy nem igazán számít.

De! Ne kódoljunk unárisan! Pl. 2 szám összeadása.

Ha I egy bemenet, jelölje $\langle I \rangle$ az I kódját.

Eldöntési probléma:

$L = \{\langle I \rangle \mid I \text{ a probléma igen példánya}\}$ eldönthető-e Turing géppel.

Kiszámítási probléma:

Van-e olyan TG, ami f -t illetve $\langle I \rangle \mapsto \langle f(I) \rangle$ -t számítja ki.

A Turing gépek elkódolása

Tegyük fel, hogy $\Sigma = \{0, 1\}$. A fentiek szerint minden input hatékonyan kódolható Σ felett.

Egy M Turing-gép **kódja** (jelölése $\langle M \rangle$) a következő:

Legyen $M = (Q, \{0, 1\}, \Gamma, \delta, q_0, q_i, q_n)$, ahol

- ▶ $Q = \{p_1, \dots, p_k\}$, $\Gamma = \{X_1, \dots, X_m\}$, $D_1 = R$, $D_2 = S$, $D_3 = L$
- ▶ $k \geq 3$, $p_1 = q_0$, $p_{k-1} = q_i$, $p_k = q_n$,
- ▶ $m \geq 3$, $X_1 = 0$, $X_2 = 1$, $X_3 = \sqcup$.
- ▶ Egy $\delta(p_i, X_j) = (p_r, X_s, D_t)$ átmenet kódja $0^i 10^j 10^r 10^s 10^t$.
- ▶ $\langle M \rangle$ az átmenetek kódjainak felsorolása 11-el elválasztva.

Észrevétel: $\langle M \rangle$ 0-val kezdődik és végződik, nem tartalmaz 3 darab 1-t egymás után.

$$\langle M, w \rangle := \langle M \rangle 111w$$

Létezik nem Turing-felismerhető nyelv

Jelölés: Minden $i \geq 1$ -re,

- ▶ jelölje w_i a $\{0, 1\}^*$ halmaz i -ik elemét a hossz-lexikografikus rendezés szerint.
- ▶ jelölje M_i a w_i által kódolt TG-t (ha w_i nem kódol TG-t, akkor M_i egy tetszőleges olyan TG, ami nem fogad el semmit)

Tétel

Létezik nem Turing-felismerhető nyelv.

Bizonyítás: Két különböző nyelvet nem ismerhet fel ugyanaz a TG. A TG-ek számossága megszámlálható (a fenti kódolás injekció $\{0, 1\}^*$ -ba, ami volt, hogy megszámlálható). Másrészt viszont a $\{0, 1\}$ feletti nyelvek számossága continuum (volt).

Azaz valójában a nyelvek "többsége" ilyen. Tudnánk-e konkrét nyelvet mutatni? Igen, $L_{\text{átló}} = \{\langle M \rangle \mid \langle M \rangle \notin L(M)\}$ például ilyen.

$L_{\text{átló}}$ Turing-felismerhetetlen

Tétel

$L_{\text{átló}} \notin RE$.

A Cantor-féle átlós módszerrel adódik:

Bizonyítás: Tekintsük azt a mindkét dimenziójában megszámlálhatóan végtelen méretű T bittáblázatot, melyre $T(i, j) = 1 \Leftrightarrow w_j \in L(M_i) \ (i, j \geq 1)$.

Legyen \mathbf{z} a T átlójában olvasható végtelen hosszú bitsztring, $\bar{\mathbf{z}}$ a \mathbf{z} bitenkénti komplementere. Ekkor:

- ▶ minden $i \geq 1$ -re, T i -ik sora az $L(M_i)$ nyelv karakterisztikus függvénye
- ▶ $\bar{\mathbf{z}}$ az $L_{\text{átló}}$ karakterisztikus függvénye
- ▶ Minden TG-pel felismerhető, azaz RE-beli nyelv karakterisztikus függvénye megegyezik T valamelyik sorával
- ▶ $\bar{\mathbf{z}}$ különbözik T minden sorától
- ▶ Ezek alapján $L_{\text{átló}}$ különbözik az összes RE-beli nyelvtől

Az univerzális TG

Felismerhetőség

Univerzális nyelv: $L_u = \{\langle M, w \rangle \mid w \in L(M)\}$.

Tétel

$$L_u \in RE$$

Bizonyítás: Konstruálunk egy 4 szalagos U "univerzális" TG-et, ami minden TG minden bementére szimulálja annak működését.

- 1. szalag:** U ezt csak olvassa, itt olvasható végig $\langle M, w \rangle$.
- 2. szalag:** M aktuális szalagtartalma (elkódolva a fentiek szerint)
- 3. szalag:** M aktuális állapota (elkódolva a fentiek szerint)
- 4. szalag:** segédzalag

Az univerzális TG

Felismerhetőség

Univerzális nyelv: $L_u = \{\langle M, w \rangle \mid w \in L(M)\}$.

Tétel

$$L_u \in RE$$

U működése vázlatosan:

- ▶ Megnézi, hogy a bemenetén szereplő szó első része kódol-e TG-t; ha nem elutasítja a bemenetet
- ▶ ha igen felmásolja w -t a 2., q_0 kódját a 3. szalagra
- ▶ Szimulálja M egy lépését:
 - Leolvassa a második szalagról M aktuálisan olvasott szalagszimbólumát
 - Leolvassa a harmadik szalagról M aktuális állapotát
 - Szimulálja M egy lépését (ha kell, használja a segédzalagot) M első szalagon található leírása alapján.
- ▶ Ha M aktuális állapota elfogadó vagy elutasító, akkor U is belép a saját elfogadó vagy elutasító állapotába

Az univerzális TG

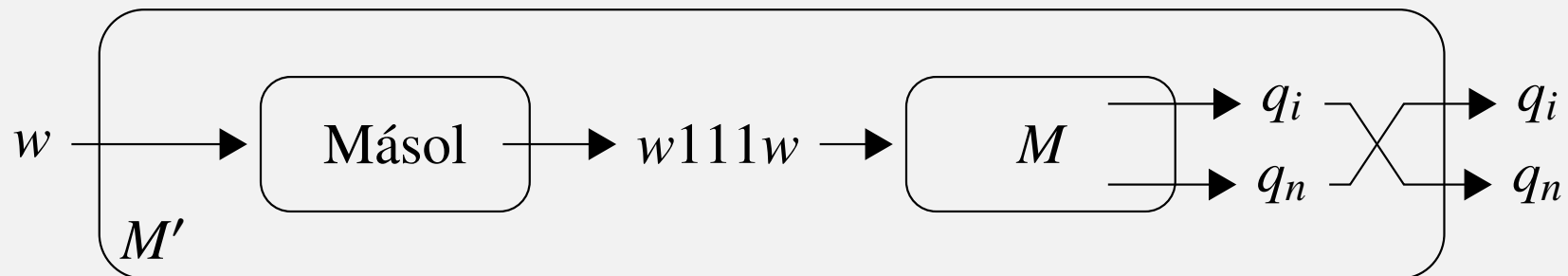
Eldönthetetlenség

Megjegyzés: Ha M nem áll meg w -n, akkor U se áll meg $\langle M, w \rangle$ -n, így U nem dönti el L_u -t.

Tétel

$L_u \notin R$.

Bizonyítás: Indirekt, tegyük fel, hogy létezik L_u -t eldöntő M TG. M -et felhasználva készítünk egy $L_{\text{átló}}$ -t felismerő M' TG-et.



$w \in L(M') \Leftrightarrow w111w \notin L(M) \Leftrightarrow$ a w által kódolt TG nem fogadja el w -t $\Leftrightarrow w \in L_{\text{átló}}$.

Tehát $L(M') = L_{\text{átló}}$, ami lehetetlen egy előző tétel miatt.

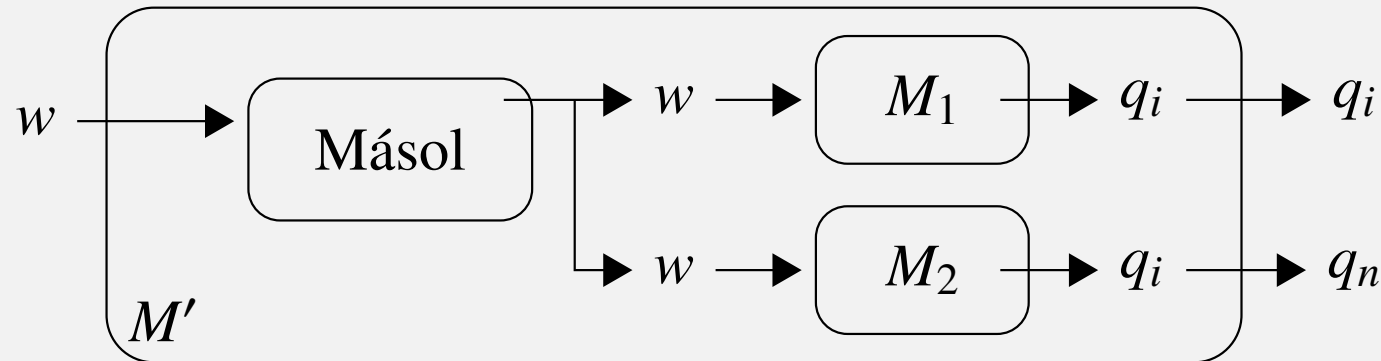
RE és R tulajdonságai

Jelölés: Ha $L \subseteq \Sigma^*$, akkor jelölje $\bar{L} = \{u \in \Sigma^* \mid u \notin L\}$.

Tétel

Ha L és $\bar{L} \in RE$, akkor $L \in R$.

Bizonyítás: Legyen M_1 és M_2 rendre az L -t és \bar{L} -t felismerő TG.
Konstruáljuk meg az M' kétszalagos TG-t:



M' lemásolja w -t a második szalagjára, majd felváltva szimulálja M_1 és M_2 egy-egy lépését addig, amíg valamelyik elfogadó állapotba lép. Így M' az L -et ismeri fel, és minden bemeneten meg is áll, azaz $L \in R$.

RE és R tulajdonságai

Következmény

RE nem zárt a komplementer-képzésre.

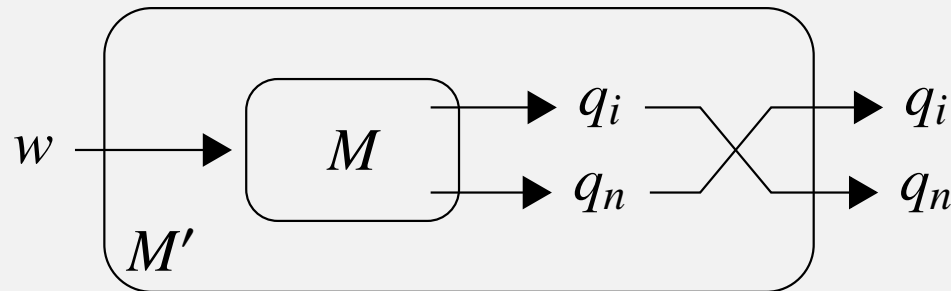
Bizonyítás:

Legyen $L \in RE \setminus R$ (L_u pl. egy ilyen nyelv) Ekkor $\bar{L} \notin RE$, hiszen ha $\bar{L} \in RE$ lenne, akkor ebből az előző tétel miatt $L \in R$ következne, ami ellentmondás.

Tétel

R zárt a komplementer-képzésre

Bizonyítás: Legyen $L \in R$ és M egy TG, ami az L -t dönti el. Akkor az alábbi M' \bar{L} -t dönti el:



Visszavezetés

Kiszámítható szófüggvény

Az $f : \Sigma^* \rightarrow \Delta^*$ szófüggvény **kiszámítható**, ha van olyan Turing-gép, ami kiszámítja. [lásd szófüggvényt kiszámító TG]

Visszavezetés

$L_1 \subseteq \Sigma^*$ **visszavezethető** $L_2 \subseteq \Delta^*$ -ra, ha van olyan $f : \Sigma^* \rightarrow \Delta^*$ kiszámítható szófüggvény, hogy $w \in L_1 \Leftrightarrow f(w) \in L_2$. Jelölés:
 $L_1 \leq L_2$

[Emil Posttól származik, angolul many-one reducibility]

Tétel

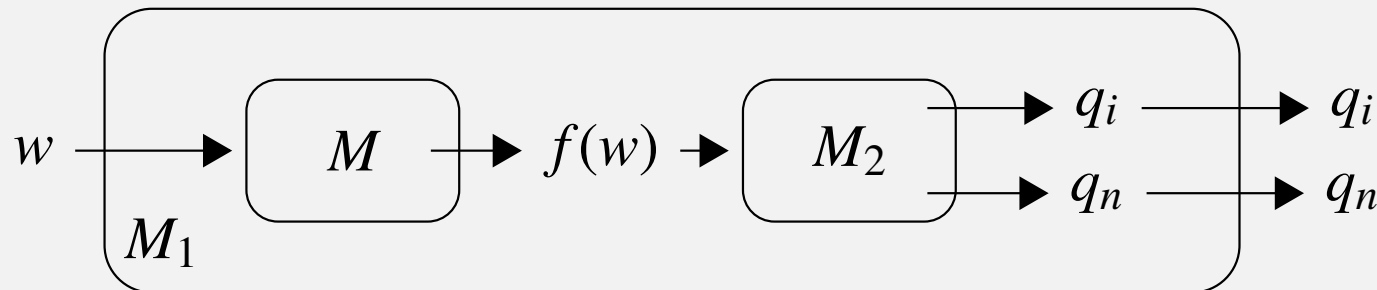
- ▶ Ha $L_1 \leq L_2$ és $L_1 \notin RE$, akkor $L_2 \notin RE$.
- ▶ Ha $L_1 \leq L_2$ és $L_1 \notin R$, akkor $L_2 \notin R$.

Visszavezetés

Bizonyítás:

Legyen $L_2 \in RE$ ($\in R$) és tegyük fel, hogy $L_1 \leq L_2$. Legyen M_2 az L_2 -t felismerő (eldöntő), M pedig a visszavezetést kiszámító TG.

Konstruáljuk meg M_1 -et:



Ha M_2 felismeri L_2 -t M_1 is fel fogja ismerni L_1 -t, ha el is dönti, akkor M_1 is el fogja dönteni.

Következmény

- ▶ Ha $L_1 \leq L_2$ és $L_2 \in RE$, akkor $L_1 \in RE$.
- ▶ Ha $L_1 \leq L_2$ és $L_2 \in R$, akkor $L_1 \in R$.

Bizonyítás: Indirekten azonnal adódik a fenti tételből.

A Turing gépek megállási problémája

Megállási probléma:

$L_h = \{\langle M, w \rangle \mid M \text{ megáll a } w \text{ bemeneten}\}.$

[Megjegyzés: más jegyzetekben L_{halt} néven is előfordulhat.]

Észrevétel: $L_u \subseteq L_h$

Igaz-e ha $A \subseteq B$, és A eldönthetetlen akkor B is az? Nem.

Tétel

$L_h \notin R.$

Bizonyítás: Az előző tétel alapján elég megmutatni, hogy $L_u \leq L_h$, hiszen tudjuk, hogy $L_u \notin R$.

Tetszőleges M TG-re, legyen M' az alábbi TG

M' tetszőleges u bemeneten a következőket teszi:

1. Futtatja M -et u -n
2. Ha M q_i -be lép, akkor M' is q_i -be lép
3. Ha M q_n -be lép, akkor M' végtelen ciklusba kerül

A Turing gépek megállási problémája

Bizonyítás: (folyt.)

Belátható, hogy

- ▶ $f : \langle M, w \rangle \rightarrow \langle M', w \rangle$ kiszámítható függvény
- ▶ Tetszőleges (M, w) (TG,input)-párra $\langle M, w \rangle \in L_u \Leftrightarrow M$ elfogadja w -t $\Leftrightarrow M'$ megáll w -n $\Leftrightarrow \langle M', w \rangle \in L_h$

Tehát f által L_u visszavezethető L_h -ra. Így $L_h \notin R$.

Megjegyzés: Visszavezetések megadásakor jellemzően csak azon szavakra térünk ki, amelyek ténylegesen kódolnak valamilyen nyelvbeli objektumot (TG-t, (TG,szó) párt, stb.)

Pl. a fenti esetben nem foglalkoztunk azzal, hogy f mit rendeljen olyan szavakhoz, melyek nem kódolnak (TG, szó) párt. Ez általában egy könnyen kezelhető eset, most:

$$f(x) = \begin{cases} \langle M', w \rangle & \text{ha } \exists M \text{ TG, hogy } x = \langle M, w \rangle \\ \varepsilon & \text{egyébként,} \end{cases} \quad (x \in \{0, 1\}^*)$$

hiszen ε nem kódol (TG,szó) párt (L_h elemei (TG,szó) párok).

A Turing gépek megállási problémája

Tétel

$$L_h \in RE.$$

Bizonyítás: Az előző tétel következménye alapján elég megmutatni, hogy $L_h \leq L_u$, hiszen tudjuk, hogy $L_u \in RE$. Tetszőleges M Turing-gépre, legyen M' az alábbi TG: M' tetszőleges u bemeneten a következőket teszi:

1. Futtatja M -et u -n
2. Ha M q_i -be lép, akkor M' is q_i -be lép
3. Ha M q_n -be lép, akkor M' q_i -be lép

Belátható, hogy

- ▶ $f : \langle M, w \rangle \rightarrow \langle M', w \rangle$ kiszámítható függvény
- ▶ Tetszőleges (M, w) (TG,input)-párra $\langle M, w \rangle \in L_h \Leftrightarrow M$ megáll w -n $\Leftrightarrow M'$ elfogadja w -t $\Leftrightarrow \langle M', w \rangle \in L_u$

Tehát f által L_h visszavezethető L_u -ra.

Logika és számításelmélet

10. előadás

Rice tétel

Rekurzíve felsorolható nyelvek tulajdonságai

Tetszőleges $\mathcal{P} \subseteq RE$ halmazt a rekurzívan felsorolható nyelvek egy tulajdonságának nevezzük. \mathcal{P} **triviális**, ha $\mathcal{P} = \emptyset$ vagy $\mathcal{P} = RE$.

$$L_{\mathcal{P}} = \{\langle M \rangle \mid L(M) \in \mathcal{P}\}.$$

Rice tétele

Ha $\mathcal{P} \subseteq RE$ egy nem triviális tulajdonság, akkor $L_{\mathcal{P}} \notin R$.

Rice tétel

Bizonyítás

Bizonyítás:

1. eset $\emptyset \notin \mathcal{P}$.

Mivel tudjuk, hogy $L_u \notin R$, elég belátni, hogy $L_u \leq L_{\mathcal{P}}$.

Mivel \mathcal{P} nem triviális, ezért létezik $L \in \mathcal{P}$. ($L \neq \emptyset$).

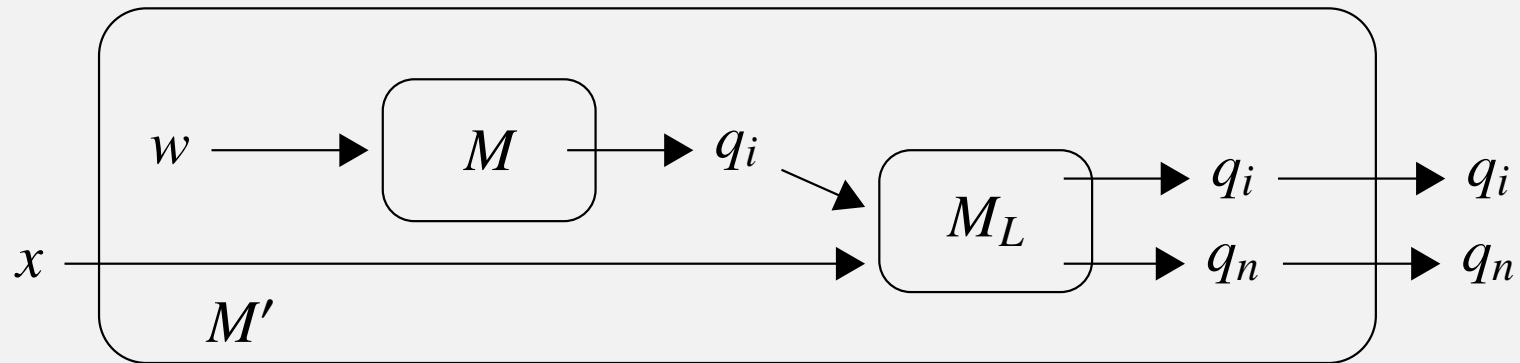
$L \in RE$, ezért van olyan M_L TG, melyre $L(M_L) = L$.

Egy tetszőleges $\langle M, w \rangle$ TG – bemenet pároshoz elkészítünk egy M' (valójában $M'_{\langle M, w \rangle}$) kétszalagos TG-t, mely egy x bemenetén a következőképpen működik:

1. Bemenetétől függetlenül először szimulálja M -et w -n
2. Így, ha M nem áll meg w -n, M' se áll meg semelyik inputján
 $\Rightarrow L(M') = \emptyset$.
3. Ha M elutasítja w -t, akkor M' q_n -be lép és leáll (azaz nem fogadja el x -et $\Rightarrow L(M') = \emptyset$).
4. Ha M elfogadja w -t, akkor M' szimulálja M_L -et x -en (azaz $L(M') = L$).

Rice tétel

Bizonyítás



Összefoglalva

- ▶ $\langle M, w \rangle \in L_u \Rightarrow L(M') = L \Rightarrow L(M') \in \mathcal{P} \Rightarrow \langle M' \rangle \in L_{\mathcal{P}}.$
- ▶ $\langle M, w \rangle \notin L_u \Rightarrow L(M') = \emptyset \Rightarrow L(M') \notin \mathcal{P} \Rightarrow \langle M' \rangle \notin L_{\mathcal{P}}.$

Azaz:

$\langle M, w \rangle \in L_u \Leftrightarrow \langle M' \rangle \in L_{\mathcal{P}},$ tehát $L_u \leq L_{\mathcal{P}}$ és így $L_{\mathcal{P}} \notin R.$

Rice tétel

Bizonyítás

2. eset $\emptyset \in \mathcal{P}$.

- ▶ Alkalmazhatjuk az 1. eset eredményét $\overline{\mathcal{P}} = RE \setminus \mathcal{P}$ -re, hiszen ekkor $\overline{\mathcal{P}}$ szintén nem triviális és $\emptyset \notin \overline{\mathcal{P}}$.
- ▶ Azt kapjuk, hogy $L_{\overline{\mathcal{P}}} \notin R$.
- ▶ $\overline{L_{\mathcal{P}}} \notin R$, hiszen ha R -beli lenne akkor a nem TG-kódokat elutasítva $L_{\overline{\mathcal{P}}}$ -t eldöntő TG-t kapnánk.
- ▶ $\overline{L_{\mathcal{P}}} \notin R \Rightarrow L_{\mathcal{P}} \notin R$ (tétel volt).

Rice tétel

Alkalmazások

Következmények:

Eldönthetetlen, hogy egy M TG

- ▶ az üres nyelvet ismeri-e fel. ($\mathcal{P} = \{\emptyset\}$)
- ▶ véges nyelvet ismer-e fel ($\mathcal{P} = \{L \mid L \text{ véges } \}$)
- ▶ környezetfüggetlen nyelvet ismer-e fel
($\mathcal{P} = \{L \mid L \text{ környezetfüggetlen } \}$)
- ▶ elfogadja-e az üres szót ($\mathcal{P} = \{L \in RE \mid \varepsilon \in L\}$)
- ▶ ...

Post Megfelelkezési Probléma

Legyenek $u_1, \dots, u_n, v_1, \dots, v_n \in \Sigma^+$ ($n \geq 1$).

A $D = \{d_1, \dots, d_n\}$ halmazt **dominókészletnek** nevezzük ha $d_i = \frac{u_i}{v_n}$ ($1 \leq i \leq n$).

A $d_{i_1} \cdots d_{i_m}$ sorozat ($m \geq 1$) a D egy **megoldása**, ha $d_{i_j} \in D$ ($1 \leq j \leq m$) és $u_{i_1} \cdots u_{i_m} = v_{i_1} \cdots v_{i_m}$.

Példa: Az $\left\{\frac{b}{ca}, \frac{a}{ab}, \frac{ca}{a}, \frac{abc}{c}\right\}$ egy megoldása $\frac{a}{ab} \frac{b}{ca} \frac{ca}{a} \frac{a}{ab} \frac{abc}{c}$.

Megjegyzés: Tehát a megoldáshoz a dominók többször felhasználhatók és nem kell mindet felhasználni.

Post Megfelelkezési Probléma (PMP):

$L_{\text{PMP}} = \{\langle D \rangle \mid D\text{-nek van megoldása}\}.$

Tétel

$L_{\text{PMP}} \in RE.$

Bizonyítás: Ha D -t egy ábécének tekintjük, akkor éppen a D feletti szavak a potenciális megoldások. Egy TG, mely ezen D feletti szavakat a hosszlexikografikus sorrendben sorra kipróbálja és ha megoldást talál q_i -ben leáll éppen L_{PMP} -t ismeri fel.

Post Megfelelkezési Probléma

Tétel

$$L_{\text{PMP}} \notin R.$$

Bizonyítás:

Definiáljuk a PMP egy módosított változatát, MPMP-t. Az MPMP probléma igen-példányai olyan (D, d) (dominókészlet, dominó) párok, melyre D -nek van d -vel kezdődő megoldása.

$$L_{\text{MPMP}} = \{\langle D, d \rangle \mid d \in D \wedge D\text{-nek van } d\text{-vel kezdődő megoldása}\}.$$

Először megmutatjuk, hogy $L_{\text{MPMP}} \leq L_{\text{PMP}}$.

Jelölés: ha $u = a_1 \cdots a_n \in \Sigma^+$ és $*$ $\notin \Sigma$ akkor legyen

$$\text{balcsillag}(u) := * a_1 * a_2 \cdots * a_n$$

$$\text{jobbcsillag}(u) := a_1 * a_2 * \cdots a_n *.$$

$$\text{mindkétcsillag}(u) := * a_1 * a_2 * \cdots a_n *.$$

Post Megfelelkezési Probléma

Legyen $D = \{d_1, \dots, d_n\}$ egy tetszőleges dominókészlet, ahol $d_i = \frac{u_i}{v_i}$ ($1 \leq i \leq n$).

D' legyen a következő $|D| + 2$ méretű készlet:

$$d'_0 = \frac{\text{balcsillag}(u_1)}{\text{mindkétsillag}(v_1)}, \quad d'_i = \frac{\text{balcsillag}(u_i)}{\text{jobbcsillag}(v_1)} \quad (1 \leq i \leq n), \quad d'_{n+1} = \frac{* \#}{\#}.$$

Állítás: $\langle D, d_1 \rangle \in L_{\text{MPMP}} \iff \langle D' \rangle \in L_{\text{PMP}}.$

Az állítás bizonyítása:

- ▶ ha $d_{i_1} \cdots d_{i_m}$ MPMP egy (D, d_1) bemenetének egy megoldása, akkor $d'_0 d'_{i_2} \cdots d'_{i_m} d'_{n+1}$ megoldása D' -nek, mint PMP inputnak.
- ▶ ha $d'_{i_1} \cdots d'_{i_m}$ D' -nek, mint PMP inputnak egy megoldása, akkor az első illetve az utolsó betű egyezése miatt ez csak úgy lehetséges, hogy $d'_{i_1} = d'_0$ és $d'_{i_m} = d'_{n+1}$. Ekkor viszont $d_{i_1} \cdots d_{i_{m-1}}$ megoldása a (D, d_1) MPMP bemenetnek.

Ezzel az állítást bizonyítottuk. Mivel a megfeleltetés TG-pel kiszámítható, ezért $L_{\text{MPMP}} \leq L_{\text{PMP}}.$

Post Megfelelkezési Probléma

Most megmutatjuk, hogy $L_u \leq L_{\text{MPMP}}$.

Minden $\langle M, w \rangle$ (TG, szó) párhoz megadunk egy $\langle D, d \rangle$ (dominókészlet, kezdődődominó) párt, úgy hogy

$w \in L(M) \iff D$ -nek van d -vel kezdődő megoldása.

Legyen $M = (Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n)$ és $w = a_1 \cdots a_n \in \Sigma^*$.
(D, d) konstrukciója:

- $d := \frac{\#}{\#q_0a_1\cdots a_n\#}$ (ahol $\# \notin \Sigma$) $d \in D$
- – ha $\delta(p, a) = (q, b, R)$, akkor $\frac{pa}{bq} \in D$
– ha $\delta(p, a) = (q, b, L)$, akkor $(\forall c \in \Gamma :) \frac{cpa}{qcb} \in D$
– ha $\delta(p, a) = (q, b, S)$, akkor $\frac{pa}{qb} \in D$
- $(\forall a \in \Gamma :) \frac{a}{a} \in D$
- $\frac{\#}{\#}, \frac{\#}{\square\#}, \frac{\#}{\#\square} \in D$
- $(\forall a \in \Gamma :) \frac{aq_i}{q_i}, \frac{q_ia}{q_i} \in D$
- $\frac{q_i\#\#}{\#} \in D$.

Post Megfelelkezési Probléma

Példa:

Ha M -nek $\delta(q_0, b) = (q_2, a, R)$ és $\delta(q_2, a) = (q_i, b, S)$ átmenetei, akkor $q_0bab \vdash aq_2ab \vdash aq_i b b$ egy bab -ot elfogadó konfigurációátmenet.

Az $\langle M, bab \rangle$ -hoz tartozó dominókészlet tartalmazza többek között a

$\frac{\#}{\#q_0bab\#}$ kezdő-, $\frac{q_0b}{aq_2}$ és $\frac{q_2a}{q_ib}$ átmenet-, $\frac{a}{a}$, $\frac{b}{b}$, $\frac{\sqcup}{\sqcup}$ és $\frac{\#}{\#}$ identikus dominókat valamint a befejezéshez szükséges $\frac{aq_i}{q_i}$, $\frac{q_ib}{q_i}$ és $\frac{q_i\#\#}{\#}$ dominókat.

Ekkor egy kirakás (|-al blokkokra osztva):

$$\frac{\#}{\#q_0bab\#} \mid \frac{q_0b}{aq_2} \frac{a}{a} \frac{b}{b} \frac{\#}{\#} \mid \frac{a}{a} \frac{q_2a}{q_ib} \frac{b}{b} \frac{\#}{\#} \mid \frac{aq_i}{q_i} \frac{b}{b} \frac{b}{b} \frac{\#}{\#} \mid \frac{q_ib}{q_i} \frac{b}{b} \frac{\#}{\#} \mid \frac{q_ib}{q_i} \frac{\#}{\#} \mid \frac{q_i\#\#}{\#}$$

Post Megfelelkezési Probléma

$$\frac{\#}{\#q_0bab\#} \mid \frac{q_0b}{aq_2} \frac{a}{a} \frac{b}{b} \frac{\#}{\#} \mid \frac{a}{a} \frac{q_2a}{q_1b} \frac{b}{b} \frac{\#}{\#} \mid \frac{aq_i}{q_i} \frac{b}{b} \frac{b}{b} \frac{\#}{\#} \mid \frac{q_ib}{q_i} \frac{b}{b} \frac{\#}{\#} \mid \frac{q_ib}{q_i} \frac{\#}{\#} \mid \frac{q_i\#\#}{\#}$$

A fenti példán szemléltetjük, hogy $w \in L(M) \iff D$ -nek van d -vel kezdődő megoldása.

Az első blokk csak a $d = \frac{\#}{\#q_0bab\#}$ kezdődődominóból áll.

A következő két blokkban alul és felül is konfigurációk következnek, felül mindig eggyel "lemaradva".

A 4.-6. blokkokban a $\frac{aq_i}{q_i}$ (és $\frac{q_ia}{q_i}$) típusú dominókkal egyesével behozható a felső szó lemaradása, egészen addig, amíg az alsó rész már csak $q_i\#$ -al hosszabb.

Végül a 7. blokkban csak egy (záró)dominó szerepel, melynek az a szerepe, hogy behozza a még megmaradt lemaradást.

Post Megfelelkezési Probléma

A fenti példa alapján meg lehet általános esetben is konstruálni egy megoldást, így $w \in L(M) \Rightarrow$ van $\langle D, d \rangle$ -nak megoldása.

Másrészt ha van d -vel kezdődő megoldás, akkor ez a dominósorozat két szavának hosszára vonatkozó megfontolások alapján csak q_i -t tartalmazó dominók használatával lehetséges. Meggondolható, hogy minden kirakás gyakorlatilag konfigurációátmenetek sorozata kell legyen az első q_i megjelenéséig, és így a w szóhoz tartozó kezdőkonfigurációból el lehet jutni elfogadó konfigurációba, azaz $w \in L(M)$.

Nyilván $\langle D, d \rangle \langle M, w \rangle$ -ből TG-pel kiszámítható, így beláttuk, hogy $L_u \leq L_{\text{MPMP}}$.

Innen a tétel bizonyítása: $L_u \leq L_{\text{MPMP}}$, $L_{\text{MPMP}} \leq L_{\text{PMP}}$ és tudjuk már, hogy $L_u \notin R$. Ebből a visszavezetés tranzitivitása és korábbi tételünk alapján $L_{\text{PMP}} \notin R$.

CF nyelvtanokkal kapcsolatos eldönthetetlen problémák

Egyértelmű nyelvtan

Egy G környezetfüggetlen (CF, 2-es típusú) nyelvtan **egyértelmű**, ha minden $L(G)$ -beli szónak pontosan egy baloldali levezetése van G -ben. (Baloldali levezetés: mindig a legbaloldalibb nemterminálist írjuk át a mondatformában.)

$$L_{\text{ECF}} = \{\langle G \rangle \mid G \text{ egy egyértelmű CF nyelvtan}\}.$$

Tétel

$$L_{\text{ECF}} \notin R$$

Bizonyítás: Megmutatjuk, hogy $L_{\text{PMP}} \leq \overline{L_{\text{ECF}}}$.

Legyen $D = \left\{ \frac{u_1}{v_1}, \dots, \frac{u_n}{v_n} \right\}$ egy tetszőleges dominókészlet.

$\Delta := \{a_1, \dots, a_n\}$ úgy, hogy $\Gamma \cap \Delta = \emptyset$.

$$P_A := \{A \rightarrow u_1 A a_1, \dots, A \rightarrow u_n A a_n, A \rightarrow \varepsilon\}.$$

$$P_B := \{B \rightarrow v_1 B a_1, \dots, B \rightarrow v_n B a_n, B \rightarrow \varepsilon\}.$$

CF nyelvtanokkal kapcsolatos eldönthetetlen problémák

Egyértelmű nyelvtan

Bizonyítás: (folyt.)

$$G_A = \langle A, \{A\}, \Gamma \cup \Delta, P_A \rangle. \quad G_B = \langle B, \{B\}, \Gamma \cup \Delta, P_B \rangle.$$

$$G_D = \langle S, \{S, A, B\}, \Gamma \cup \Delta, \{S \rightarrow A, S \rightarrow B\} \cup P_A \cup P_B \rangle.$$

$f : \langle D \rangle \rightarrow \langle G_D \rangle$ visszavezetés, mert:

* ha $\frac{u_{i_1}}{v_{i_1}} \cdots \frac{u_{i_m}}{v_{i_m}}$ megoldása D -nek, akkor $u_{i_1} \cdots u_{i_m} = v_{i_1} \cdots v_{i_m}$.

De ekkor $u_{i_1} \cdots u_{i_m} a_{i_m} \cdots a_{i_1} = v_{i_1} \cdots v_{i_m} a_{i_m} \cdots a_{i_1}$ kétféleképpen is levezethető, így G_D nem egyértelmű.

* ha G_D nem egyértelmű, akkor van olyan szó, aminek két baloldali levezetése van. De ezek $S \rightarrow A$ -val illetve $S \rightarrow B$ -vel kell kezdődjenek, hiszen G_A és G_B egyértelmű. A generált szavak xy , $x \in \Gamma^*$, $y \in \Delta^*$ alakúak, így ugyanaz a generált Γ feletti prefix is. Így a két levezetés D egy megoldását adja.

f nyilván TG-pel kiszámítható. Mivel $L_{\text{PMP}} \notin R$, következik, hogy $L_{\text{ECF}} \notin R$, amiből kapjuk, hogy $L_{\text{ECF}} \notin R$.

CF nyelvtanokkal kapcsolatos eldönthetetlen problémák

Közös metszet, ekvivalencia, tartalmazás

Tétel

Eldönthetetlenek az alábbi CF nyelvtanokkal kapcsolatos kérdések.
Legyen G_1 és G_2 két CF nyelvtan.

- ▶ $L(G_1) \cap L(G_2) \neq \emptyset$
- ▶ $L(G_1) = \Gamma^*$ valamely Γ -ra
- ▶ $L(G_1) = L(G_2)$
- ▶ $L(G_1) \subseteq L(G_2)$

Csak az elsőit bizonyítjuk. L_{PMP} -t vezethetjük vissza rá. Legyen $D = \left\{ \frac{u_1}{v_1}, \dots, \frac{u_n}{v_n} \right\}$ a dominókészlet. Készítsük el a fenti G_A és G_B nyelvtanokat. Könnyen látható, hogy D -nek akkor és csak akkor van megoldása, ha $L(G_A)$ -nak és $L(G_B)$ -nek van nemüres metszete.
(A másik 3 állítás: biz. nélkül)

Eldönthetetlen problémák az elsőrendű logikában

Tétel

Eldönthetetlen, hogy A elsőrendű logikai formulára

(1) $\models A$ teljesül-e (logikailag igaz-e).

(biz. nélkül). L_{PMP} -t lehet visszavezetni rá. Azaz minden D dominókészlethez megadható egy A_D elsőrendű formula, melyre van D -nek megoldása $\Leftrightarrow \models A_D$. (részletek l. pl. Gazdag jegyzet)

Következmény

Legyen \mathcal{F} egy elsőrendű formulahalmaz és A egy elsőrendű formula.

Eldönthetetlen, hogy

(2) A kielégíthetetlen-e

(3) A kielégíthető-e

(4) $\mathcal{F} \models A$ teljesül-e

Bizonyítás: (2) $\models \neg A \Leftrightarrow A$ kielégíthetetlen. (3) Eldönthetetlen nyelv komplementere. (4) Kielégíthetlenségre visszavezethető (l. logika rész).

Eldönthetetlen problémák az elsőrendű logikában

Logikából tanultuk, hogy van olyan algoritmus, ami egy tetszőleges A elsőrendű formulára pontosan akkor áll meg igen válasszal, ha A kielégíthetetlen (például a elsőrendű logika rezolúciós algoritmus). Ezért a kielégíthetlenség eldöntése RE-beli probléma.

\Rightarrow a kielégíthetőség eldöntése nem RE-beli probléma.

Mi a helyzet nulladrendű logika esetén?

A fenti kérdések mindegyike eldönthető. (ítélettábla). Véges sok interpretáció van, elsőrendben végtelen.

Nulladrendű logikában, az a kérdés van-e hatékony megoldás.

A továbbiakban az R nyelvosztályt vizsgáljuk. (Bonyolultságelmélet.)

BONYOLULTSÁGELMÉLET

Determinisztikus és nemdeterminisztikus időbonyolultsági osztályok

A továbbiakban eldönthető problémákkal foglalkozunk, ilyenkor a kérdés az, hogy milyen hatékonyan dönthető el az adott probléma.

- ▶ $\text{TIME}(f(n)) = \{L \mid L \text{ eldönthető } O(f(n)) \text{ időigényű determinisztikus TG-pel}\}$
- ▶ $\text{NTIME}(f(n)) = \{L \mid L \text{ eldönthető } O(f(n)) \text{ időigényű NTG-pel}\}$
- ▶ $P = \bigcup_{k \geq 1} \text{TIME}(n^k)$.
- ▶ $NP = \bigcup_{k \geq 1} \text{NTIME}(n^k)$.
- ▶ Észrevétel: $P \subseteq NP$.
- ▶ Sejtés: $P \neq NP$ (sejtjük, hogy igaz, de bizonyítani nem tudjuk).

NP

A P tartalmazza a gyakorlatban is hatékonyan megoldható problémákat.

Milyen problémákat tartalmaz NP?

Egy L NP-beli problémához definíció szerint létezik öt polinom időben eldöntő NTG ami gyakran a következőképpen működik: a probléma minden I bemenetére polinom időben „megsejti” (azaz nemdeterminisztikusan generálja) I egy lehetséges m megoldását és polinom időben leellenőrzi (determinisztikusan), hogy m alapján $I \in L$ -e.

A következőkben a P és NP bonyolultsági osztályok közötti kapcsolatot vizsgáljuk.

Polinom idejű visszavezetés

Polinom időben kiszámítható szófüggvény

Az $f : \Sigma^* \rightarrow \Delta^*$ szófüggvény **polinom időben kiszámítható**, ha van olyan Turing-gép, ami polinom időben kiszámítja.

Visszavezetés polinom időben

$L_1 \subseteq \Sigma^*$ **polinom időben visszavezethető** $L_2 \subseteq \Delta^*$ -ra, ha van olyan $f : \Sigma^* \rightarrow \Delta^*$ polinom időben kiszámítható szófüggvény, hogy $w \in L_1 \Leftrightarrow f(w) \in L_2$. Jelölés: $L_1 \leq_p L_2$.

A polinom idejű visszavezetést Richard Karpról elnevezve *Karp-redukciónak* is nevezik.

Tétel

- ▶ Ha $L_1 \leq_p L_2$ és $L_2 \in P$, akkor $L_1 \in P$.
- ▶ Ha $L_1 \leq_p L_2$ és $L_2 \in NP$, akkor $L_1 \in NP$.

Az elsőt bizonyítjuk, a második analóg.

Polinom idejű visszavezetés

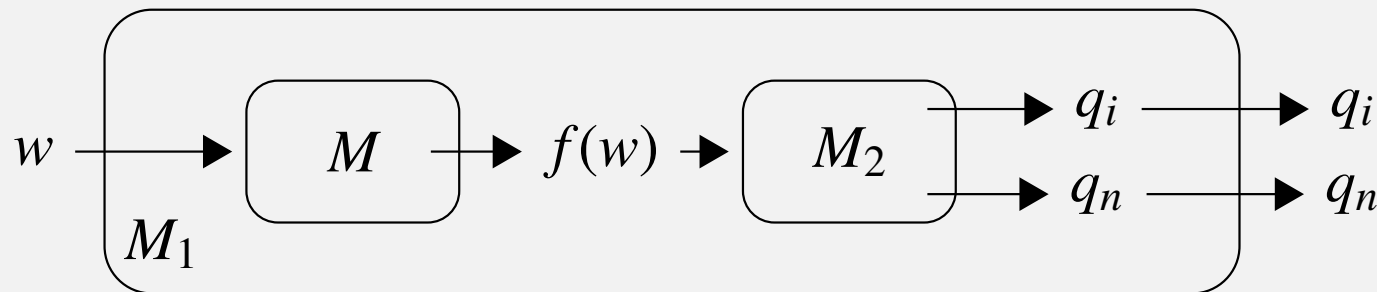
Bizonyítás:

Legyen $L_2 \in P$ és tegyük fel, hogy $L_1 \leq_p L_2$.

Legyen M_2 az L_2 -t eldöntő, míg M a visszavezetést kiszámító TG.

Feltehetjük, hogy M $p(n)$ és M_2 $p_2(n)$ polinom idejű TG-ek.

Konstruáljuk meg M_1 -et:



- ▶ M_1 eldönti az L_1 nyelvet
- ▶ ha w n hosszú, akkor $f(w)$ legfeljebb $p(n)$ hosszú lehet
- ▶ M_1 időigénye $p_2(p(n))$, ami szintén polinom

Polinom idejű visszavezetés

Adott problémaosztályra nézve nehéz nyelv

Legyen \mathfrak{C} egy problémaosztály. egy L probléma **\mathfrak{C} -nehéz** (a polinom idejű visszavezetésre nézve), ha minden $L' \in \mathfrak{C}$ esetén $L' \leq_p L$.

Adott problémaosztályban teljes nyelv

Egy \mathfrak{C} -nehéz L probléma **\mathfrak{C} -teljes**, ha $L \in \mathfrak{C}$.

Tétel

Legyen L egy NP-teljes probléma. Ha $L \in P$, akkor $P = NP$.

Bizonyítás: Elég megmutatni, hogy $NP \subseteq P$.

Legyen $L' \in NP$ egy tetszőleges probléma.

Ekkor $L' \leq_p L$, hiszen L NP-teljes.

Mivel $L \in P$, ezért az előző tétel alapján $L' \in P$.

Ez minden $L' \in NP$ -re elmondható, ezért $NP \subseteq P$.

Logika és számításelmélet

11. előadás

NP-teljesség

Emlékeztetőül:

NP-teljes nyelv

Egy L probléma **NP-teljes** (a polinom idejű visszavezetésre nézve), ha

- ▶ $L \in \text{NP}$
- ▶ L NP-nehéz, azaz minden $L' \in \text{NP}$ esetén $L' \leq_p L$.

Azaz az NP-teljes problémák (ha vannak) az NP-beli problémák legnehezebbjei. Egyikre sem ismeretes polinomiális algoritmus és nem túl valószínű, hogy valaha is lesz ugyanis láttuk, hogy elég egyetlen NP-teljes problémáról bizonyítani, hogy P-beli, ez implikálná, hogy $P = \text{NP}$.

Van-e NP-teljes probléma egyáltalán?

$\text{SAT} = \{ \langle \varphi \rangle \mid \varphi \text{ kielégíthető nulladrendű KNF} \}$

Tétel (Cook)

SAT NP-teljes.

Cook tétel bizonyítás

Bizonyítás:

- ▶ $\text{SAT} \in \text{NP}$: Adott egy φ input. Egy NTG egy számítási ágán polinom időben előállít egy I interpretációt. Majd szintén polinom időben ellenőrzi, hogy ez kielégíti-e φ -t.
- ▶ SAT NP-nehéz: ehhez kell, $L \leq_p \text{SAT}$, minden $L \in \text{NP}$ -re.
 - Legyen $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_i, q_n \rangle$ egy L -et eldöntő $p(n)$ polinom időkorlátos NTG. (Feltehető, hogy $p(n) \geq n$.)
 - Legyen továbbá $w = a_1 \cdots a_n \in \Sigma^*$ egy szó.
 - M segítségével megadunk egy polinom időben előállítható φ_w nulladrendű KNF formulát, melyre $w \in L \Leftrightarrow \langle \varphi_w \rangle \in \text{SAT}$.
 - M egy számítása w -n leírható egy T táblázattal, melynek
 - első sora $\# \sqcup^{p(n)} C_0 \sqcup^{p(n)-n} \#$, ahol $C_0 = q_0 w M$ kezdőkonfigurációja w -n
 - T egymást követő két sora M egymást követő két konfigurációja (elegendő \sqcup -el kiegészítve, elején és a végén egy $\#$ -el). Minden sor $2p(n) + 3$ hosszú.

Cook tétel bizonyítás (folyt.)

- $p(n) + 1$ sor van. Ha hamarabb jut elfogadó konfigurációba, akkor onnantól kezdve ismételjük meg az elfogadó konfigurációt.

kezdőkonf.

1. konf.

2. konf.

•
•
•

$p(n)$. konf.

[illegible]
$$p(n) + 1 \text{ sor}$$
$$2p(n) + 3 \text{ oszlop}$$

- a konfigurációátmenet definíciója miatt bármely két sor közötti különbség belefér egy 2x3-as "ablakba"
- T magassága akkora, hogy minden, $\leq p(n)$ lépéses átmenetet tartalmazhasson. A \sqcup -ek számát ($\Rightarrow T$ szélességét) pedig úgy, hogy az ablakok biztosan "ne eshessenek le" egyik oldalon se.

Cook tétel bizonyítás (folyt.)

- φ_w ítéletváltozói $X_{i,j,s}$ alakúak, melynek jelentése: T i -ik sorának j -ik cellájában az s szimbólum van, ahol $s \in \Delta = Q \cup \Gamma \cup \{\#\}$.
- φ_w a w bemenetre M minden lehetséges legfeljebb $p(n)$ lépésű működését leírja. Felépítése: $\varphi_w = \varphi_0 \wedge \varphi_{\text{start}} \wedge \varphi_{\text{move}} \wedge \varphi_{\text{accept}}$.
- φ_0 akkor és csak akkor legyen igaz, ha minden cellában pontosan 1 betű van:

$$\varphi_0 := \bigwedge_{\substack{1 \leq i \leq p(n)+1 \\ 1 \leq j \leq 2p(n)+3}} \left(\left(\bigvee_{s \in \Delta} X_{i,j,s} \right) \wedge \bigwedge_{s,t \in \Delta, s \neq t} (\neg X_{i,j,s} \vee \neg X_{i,j,t}) \right)$$

- φ_{start} akkor és csak akkor legyen igaz, ha T első sora a \sqcup -okkal és $\#$ -ekkel a fent említett módon adott hosszúságúra kiegészített kezdőkonfiguráció.

$$\varphi_{\text{start}} := X_{1,1,\#} \wedge X_{1,2,\sqcup} \wedge \cdots \wedge X_{1,2p(n)+2,\sqcup} \wedge X_{1,2p(n)+3,\#}$$

Cook tétel bizonyítás (folyt.)

– φ_{move} akkor és csak akkor legyen igaz, ha minden ablak legális, azaz δ szerinti átmenetet ír le:

$$\varphi_{\text{move}} := \bigwedge_{\substack{1 \leq i \leq p(n) \\ 2 \leq j \leq 2p(n)+2}} \psi_{i,j},$$

ahol $\psi_{i,j} \sim \bigvee_{\substack{(b_1, \dots, b_6) \\ \text{legális ablak}}} X_{i,j-1,b_1} \wedge X_{i,j,b_2} \wedge X_{i,j+1,b_3} \wedge X_{i+1,j-1,b_4} \wedge X_{i+1,j,b_5} \wedge X_{i+1,j+1,b_6}$

b_1	b_2	b_3
b_4	b_5	b_6

De: $\psi_{i,j}$ nem elemi diszjunkció!!! Ezért e helyett:

$$\psi_{i,j} := \bigwedge_{\substack{(b_1, \dots, b_6) \\ \text{illegális ablak}}} \neg X_{i,j-1,b_1} \vee \neg X_{i,j,b_2} \vee \neg X_{i,j+1,b_3} \vee \neg X_{i+1,j-1,b_4} \vee \neg X_{i+1,j,b_5} \vee \neg X_{i+1,j+1,b_6}$$

Cook tétel bizonyítás (folyt.)

– végezetül: φ_{accept} akkor és csak akkor legyen igaz, ha az utolsó sorban van q_i -t:

$$\varphi_{\text{accept}} = \bigvee_{j=2}^{2p(n)+2} X_{p(n)+1,j,q_i}$$

.

- $w \in L \Leftrightarrow$ az M NTG-nek van w -t elfogadó számítása $\Leftrightarrow T$ kitölthető úgy, hogy ϕ_w igaz $\Leftrightarrow \phi_w$ kielégíthető $\Leftrightarrow \langle \varphi_w \rangle \in \text{SAT}$,

- hány literált tartalmaz a φ_w formula? Legyen $k = |\Delta|$.

$\phi_0 : (p(n) + 1)(2p(n) + 3)(k + k(k - 1)) = O(p^2(n))$,

$\varphi_{\text{start}} : 2p(n) + 3 = O(p(n))$,

$\varphi_{\text{move}} : \leq p(n)(2p(n) + 1)k^6 \cdot 6 = O(p^2(n))$,

$\varphi_{\text{accept}} : 2p(n) + 1 = O(p(n))$,

azaz φ_w $O(p^2(n))$ méretű, így polinom időben megkonstruálható

- tehát $w \mapsto \langle \varphi_w \rangle$ pol. idejű visszavezetés, így $L \leq_p \text{SAT}$.

- Ez tetszőleges $L \in \text{NP}$ nyelvre elmondható. Így SAT NP-nehéz.

Mivel NP-beli, ezért NP-teljes is.

További NP-teljes problémák, kSAT

Tétel

Ha L NP-teljes, $L \leq_p L'$ és $L' \in \text{NP}$, akkor L' NP-teljes.

Bizonyítás: Legyen $L'' \in \text{NP}$ tetszőleges. Mivel L NP-teljes, ezért $L'' \leq_p L$. Mivel a feltételek szerint $L \leq_p L'$, ezért a polinom idejű visszavezetések tranzitívítása miatt ($p_1(p_2(n))$ is polinom!) L' NP-nehéz. Ebből és a 3. feltételből következik az állítás.

Tehát polinom idejű visszavezetéssel további nyelvek NP-teljessége bizonyítható.

$k\text{SAT} = \{\langle \varphi \rangle \mid \varphi \text{ kielégíthető KNF és minden tagban pontosan } k \text{ különböző literál van}\}.$

Az ilyen formulákat $k\text{KNF}$ -nek nevezzük a továbbiakban.

3SAT NP-teljesége

Tétel

3SAT NP-teljes.

► 3SAT NP-beli: lásd SAT

► $\text{SAT} \leq_p 3\text{SAT}$

Kell $f : \varphi \mapsto \varphi'$, φ KNF, φ' 3KNF, φ' kielégíthető $\Leftrightarrow \varphi$ kielégíthető, f polinom időben kiszámolható.

$\varphi \mapsto \varphi'$:

ℓ	$\ell \vee X \vee Y, \ell \vee X \vee \neg Y, \ell \vee \neg X \vee Y, \ell \vee \neg X \vee \neg Y$
$\ell_1 \vee \ell_2$	$\ell_1 \vee \ell_2 \vee X, \ell_1 \vee \ell_2 \vee \neg X$
$\ell_1 \vee \ell_2 \vee \ell_3$	$\ell_1 \vee \ell_2 \vee \ell_3$
$\ell_1 \vee \ell_2 \vee \ell_3 \vee \ell_4$	$\ell_1 \vee \ell_2 \vee X, \neg X \vee \ell_3 \vee \ell_4$
$\ell_1 \vee \dots \vee \ell_n \ (n \geq 5)$	$\ell_1 \vee \ell_2 \vee X_1, \neg X_1 \vee \ell_3 \vee X_2, \dots, \neg X_{n-2} \vee \ell_{n-1} \vee \ell_n$

$X, Y, X_1, \dots, X_{n-2}$ új ítéletváltozók. φ' ezek konjunkciója.

Megjegyzés: HORNSAT: mint SAT, de klózonként max. 1 pozitív literál lehet. HORNSAT és 2SAT $\in P$.

3 színezhetőség

Egy gráf k -**színezhető**, ha csúcsai k színnel színezhetők úgy, hogy a szomszédos csúcsok színei különbözőek.

$3\text{SZÍNEZÉS} = \{\langle G \rangle \mid G \text{ 3-színezhető}\}$

Tétel

3SZÍNEZÉS NP-teljes.

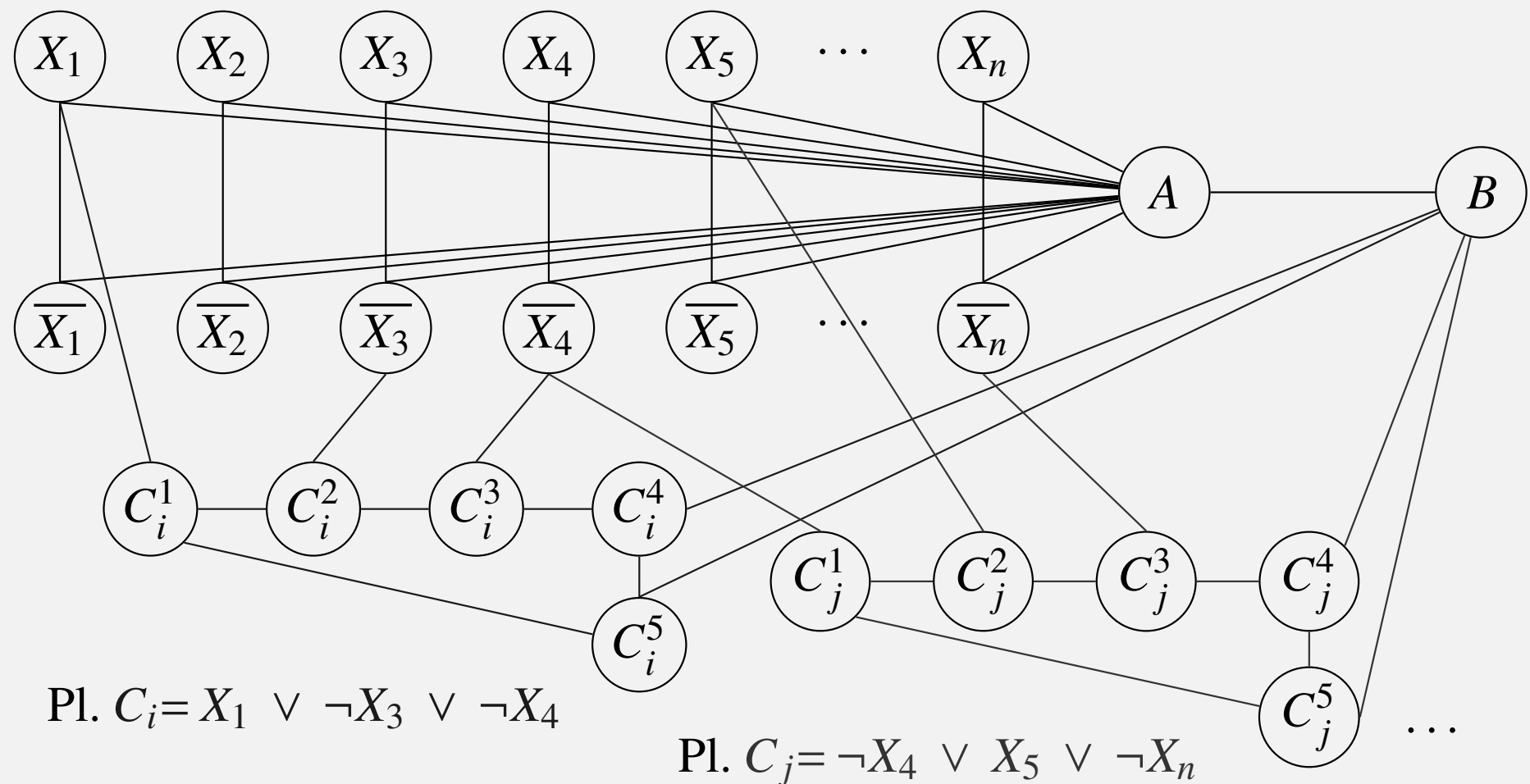
- ▶ 3SZÍNEZÉS NP-beli: egy NTG számítási ágai színezzék ki a gráfot. Egy konkrét színezésről ellenőrizni, hogy helyes-e polinom időben megtehető.
- ▶ $3\text{SAT} \leq_p 3\text{SZÍNEZÉS}$

Elegendő minden φ 3KNF formulához polinom időben elkészíteni egy G_φ gráfot úgy φ kielégíthető $\Leftrightarrow G_\varphi$ 3 színezhető.

3 színezhetőség

Legyenek X_1, \dots, X_n a φ -ben előforduló ítéletváltozók. Továbbá $\varphi = C_1 \wedge \dots \wedge C_m$, azaz C_1, \dots, C_m φ pontosan 3 literálból álló klózai.

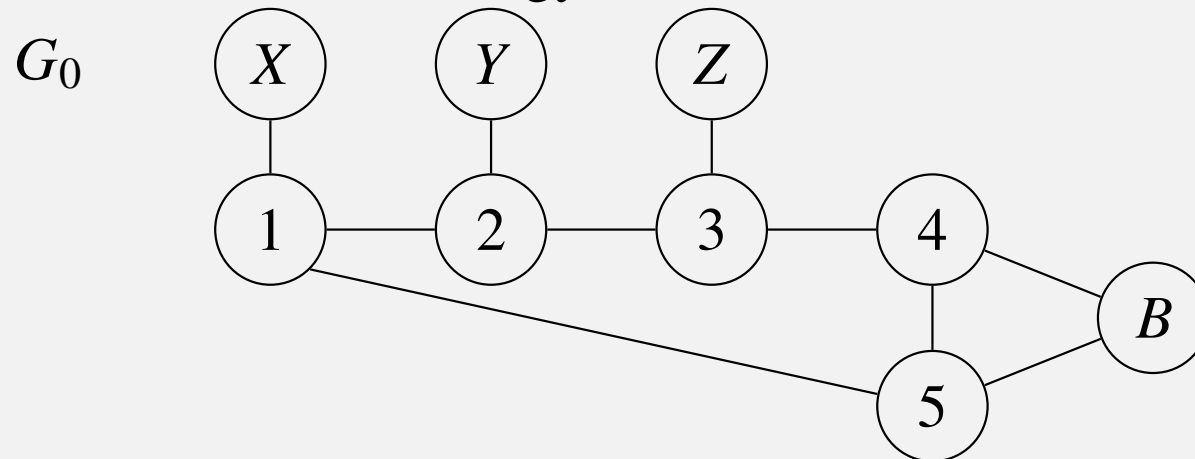
G_φ konstrukciója:



Minden klózhoz tartozik egy ötszög a fenti módon.

3 színezhetőség

Lemma: Legyen G_0 az alábbi gráf és tegyük fel, hogy az X, Y, Z, B csúcsokat 2 színnel kiszíneztük. Akkor és csak akkor létezik ehhez a parciális színezéshez az egész G_0 -ra kiterjeszthető 3-színezés, ha X, Y, Z, B nem mind egyszínű.



A lemma bizonyítása:

- ▶ Ha X, Y, Z, B egyszínű, akkor a maradék 2 színnel kéne az ötszöget kiszínezni, amit nem lehet.
- ▶ Egyébként megadunk egy színezést. *1. lépés:* első körben csak 2 színt használunk, 1,2,3,4,5-öt színezzük az $\{X, Y, Z, B\}$ -beli szomszédjával ellentétes színűre.

3 színezhetőség

Ez persze még nem jó, mert 1,2,3,4,5 között lehetnek azonos színű szomszédok.

2. lépés: bevetjük a 3. színt: ha 1,2,3,4,5 között van ciklikusan, az óramutató járása szerint valahány egymás utáni egyszínű csúcs, akkor minden párosadikat színezzük át a 3. színre.

A visszavezetés bizonyítása:

- Tegyük fel hogy φ kielégíthető, ekkor meg kell adnunk G_φ egy 3-színezését. Legyenek a színek piros, zöld és kék. Ha X_i igaz, akkor legyen az X_i csúcs zöld, az $\overline{X_i}$ csúcs piros. Ha hamis, akkor épp fordítva. A legyen kék és B legyen piros. Mivel minden klóz ki van elégítve, így minden ötszöghöz van zöld (az igaz literál) és piros szomszéd (B) is, így a lemma miatt a színezés minden ötszögre kiterjeszthető.

3 színezhetőség

- ▶ Tegyük fel most, hogy G_φ ki van színezve 3 színnel. Ámnfth. A kék. Mivel $X_1, \dots, X_n, \overline{X_1}, \dots, \overline{X_n}$ mind A szomszédai, így egyikük se lehet kék. Továbbá az $(X_i, \overline{X_1})$ párok össze vannak kötve, így minden párban pontosan egy piros és pont egy zöld csúcs van. Ámnfth. B piros (a zöld eset analóg). Mivel az ötszögek ki vannak színezve, ezért a lemma miatt minden ötszögnek van zöld szomszédja. Az " $X_i := \text{igaz} \Leftrightarrow X_i$ csúcs zöld" interpretáció tehát kielégíti φ -t.

Tehát beláttuk, hogy $\varphi \mapsto G_\varphi$ visszavezetés. Mivel G_φ φ ismeretében az input méretének polinomja időben legyártható, ezért a visszavezetés polinom idejű.

Mivel 3SAT NP-teljes korábbi tételünk miatt 3SZÍNEZÉS is az.

Megjegyzés: 2SZÍNEZÉS $\in P$

3 irányítatlan gráfokkal kapcsolatos probléma

Az alábbi nyelvek esetén G egyszerű, irányítatlan gráf k pedig egy nemnegatív egész. G egy teljes részgráfját **klikknek**, egy üres részgráfját **független ponthalmaznak** mondjuk.

$\text{KLIKK} = \{\langle G, k \rangle \mid G\text{-nek van } k \text{ méretű klikkje}\}$

$\text{FÜGGETLEN PONT HALMAZ} =$

$\{\langle G, k \rangle \mid G\text{-nek van } k \text{ méretű független ponthalmaza}\}$

Legyen $S \subseteq V(G)$ és $E \in E(G)$. Ha $S \cap E \neq \emptyset$, akkor a csúcshalmaz **lefogja** E -t. Ha S minden $E \in E(G)$ élt lefog, akkor S egy **lefogó ponthalmaz**.

[Megjegyzés: LEFOGÓ PONT HALMAZ a Gazdag jegyzetben CSÚCSLEFEDÉS néven szerepel]

$\text{LEFOGÓ PONT HALMAZ} = \{\langle G, k \rangle \mid G\text{-nek van } k \text{ méretű lefogó ponthalmaza}\}$

Ha G -nek van k méretű klikkje/független ponthalmaza, akkor bármely kisebb k -ra is van. Ha van k méretű lefogó ponthalmaz, akkor bármely nagyobb k -ra is van ($k \leq |V(G)|$).

FÜGGETLEN PONTALMAZ

Tétel

KLIKK, FÜGGETLEN PONTALMAZ, LEFOGÓ PONTALMAZ NP-teljes.

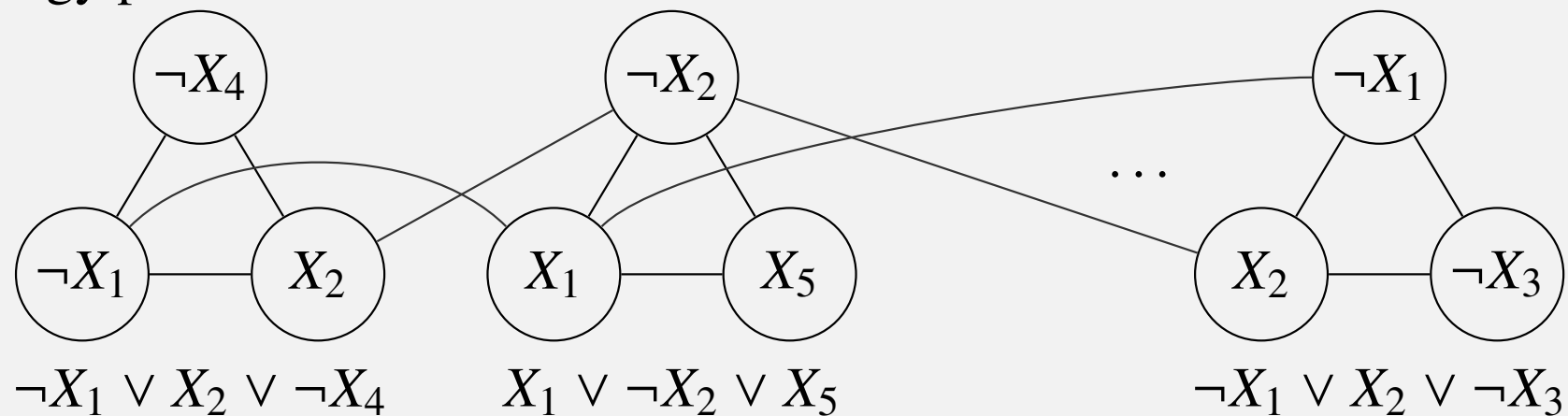
- ▶ Egy NTG egy számítási ágán vizsgáljon meg egy k elemű pontalmazt. Mindhárom esetben az ellenőrzés polinomiális.
- ▶ $3\text{SAT} \leq_p \text{FÜGGETLEN CSÚCSHALMAZ}$

Kell: $f : \varphi \mapsto (G_\varphi, k)$, φ 3KNF, G_φ -ben van k független akkor és csak akkor ha φ kielégíthető.

(G_φ, k) konstrukciója: minden egyes $L_1 \vee L_2 \vee L_3$ klózhoz vegyünk fel egy a többitől diszjunkt háromszöget, a csúcsokhoz rendeljük hozzá a literálokat. Így m darab klóz esetén $3m$ csúcsot kapunk. Kössük össze éllel ezen felül a komplement párokat is.
 $k := m$.

FÜGGETLEN PONTHALMAZ

Egy példa:



* Ha φ kielégíthető, akkor minden klózban van kielégített literál, válasszunk klózonként egyet, ezeknek megfelelő csúcsok m elemű független csúcshalmazt alkotnak.

* Ha G_φ -ben van m független csúcs, akkor ez csak úgy lehet, ha háromszögenként 1 van. Vegyünk egy ilyet, ezen csúcsoknak megfelelő literálok között nem lehet komplementens pár, hiszen azok össze vannak kötve. Így a független halmaznak megfelelő, (esetleg csak parciális) interpretáció kielégít minden klózt. Ha nincs minden változó kiértékelve, egészítsük ki tetszőlegesen egy interpretációvá.

KLIKK, LEFOGÓ PONTALMAZ

- FÜGGETLEN PONTALMAZ \leq_p KLIKK

$$f : (G, k) \mapsto (\bar{G}, k)$$

Ez jó visszavezetés, mert ami G -ben klikk az \bar{G} -ban független pontalmaz és viszont.

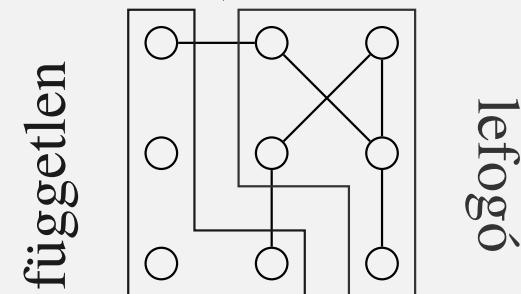
- FÜGGETLEN PONTALMAZ \leq_p LEFOGÓ PONTALMAZ

$$f : (G, k) \mapsto (G, |V(G)| - k)$$

Ha G -ben van k méretű F független pontalmaz, akkor van $|V(G)| - k$ méretű lefogó pontalmaz (F komplementere).

Ha G -ben van $|V(G)| - k$ méretű L lefogó pontalmaz, akkor van k méretű független pontalmaz (L komplementere).

Mindkét visszavezetés polinom időben kiszámítható.



Lefogó ponthalmaz hipergráfokban

S egy **hipergráf** (vagy halmazrendszer), ha $S = \{A_1, \dots, A_n\}$, ahol $A_i \subseteq U$, $(1 \leq i \leq n)$ valamely U alaphalmazra. $H \subseteq U$ egy **lefogó ponthalmaz**, ha $\forall 1 \leq i \leq n : H \cap A_i \neq \emptyset$.

HIPERGRÁF LEFOGÓ PONTALMAZ = $\{\langle S, k \rangle \mid S \text{ egy hipergráf és } S\text{-hez van } k \text{ elemű lefogó ponthalmaz}\}$.

Tétel

HIPERGRÁF LEFOGÓ PONTALMAZ NP-teljes.

Bizonyítás: HIPERGRÁF LEFOGÓ PONTALMAZ NP-beli, hiszen polinom időben ellenőrizhető, hogy U egy részhalmaza minden S -beli halmazt metsz-e.

LEFOGÓ PONTALMAZ a HIPERGRÁF LEFOGÓ PONTALMAZ speciális esete, hiszen a gráf a hipergráf speciális esete: egy gráf éleire úgy gondolunk, mint 2-elemű halmazokra, így a gráf éleinek halmaza egy hipergráf. (A visszavezetés az identikus leképezés.

$U := V(G)$, $S := E(G)$, k ugyanaz).

[Megjegyzés: a Gazdag jegyzetben HITTING SET]

Logika és számításelmélet

12. előadás

Irányítatlan/irányított Hamilton út/kör

Hamilton út/kör

Adott egy $G = (V, E)$ irányítatlan / irányított gráf ($|V| = n$). Egy $P = v_{i_1}, \dots, v_{i_n}$ felsorolása a csúcsoknak **Hamilton út** G -ben, ha $\{v_{i_1}, \dots, v_{i_n}\} = V$ és minden $1 \leq k \leq n - 1$ -re $\{v_{i_k}, v_{i_{k+1}}\} \in E$ (illetve irányított esetben $(v_{i_k}, v_{i_{k+1}}) \in E$). Ha $\{v_{i_n}, v_{i_1}\} \in E$ (illetve irányított esetben $(v_{i_n}, v_{i_1}) \in E$) is teljesül, akkor P **Hamilton kör**.

Jelölés: H-út/ H-kör Hamilton út/ Hamilton kör helyett.

$HÚ = \{\langle G, s, t \rangle \mid \text{van a } G \text{ irányított gráfban } s\text{-ből } t\text{-be H-út}\}.$

$IHÚ = \{\langle G, s, t \rangle \mid \text{van a } G \text{ irányítatlan gráfban } s\text{-ből } t\text{-be H-út}\}.$

$IHK = \{\langle G \rangle \mid \text{van a } G \text{ irányítatlan gráfban H-kör}\}.$

Irányított $s - t$ -Hamilton út NP teljessége

Tétel

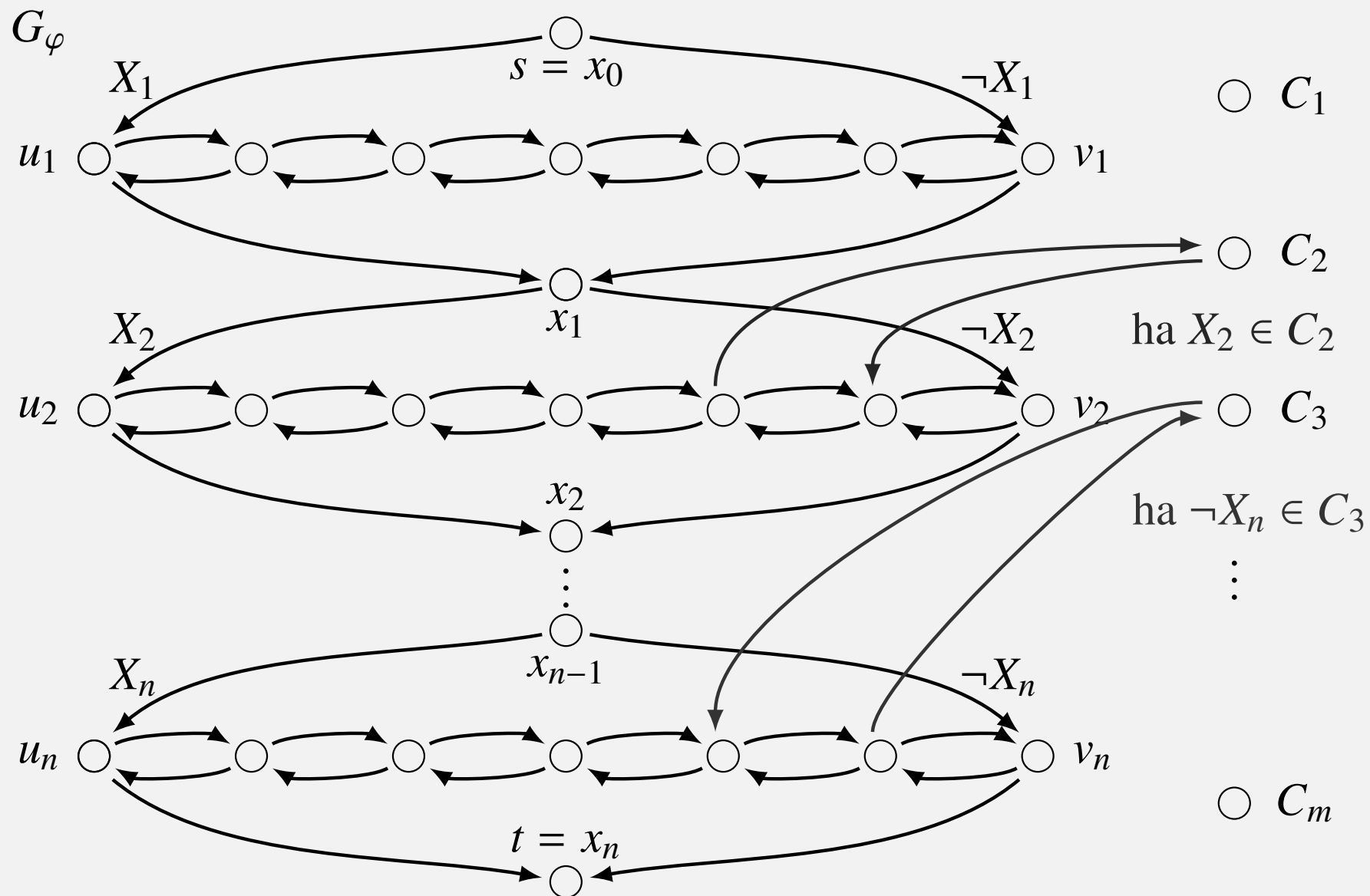
HÚ NP-teljes

Bizonyítás: NP-beli, hiszen polinom időben előállítható egy n darab csúcs egy P felsorolása. P -ről polinom időben ellenőrizhető, hogy a csúcsok egy permutációja-e és hogy tényleg H-út-e.

$\text{SAT} \leq_p \text{HÚ}$. Elég bármely φ KNF-hez konstruálni (G_φ, s, t) -t azzal a tulajdonsággal, hogy φ kielégíthető \Leftrightarrow a G_φ -ben van s -ből t -be H-út.

Legyenek X_1, \dots, X_n a φ -ben előforduló ítéletváltozók és C_1, \dots, C_m φ klózai.

Irányított $s - t$ -Hamilton út NP teljessége



Irányított $s - t$ -Hamilton út NP teljessége

G_φ konstrukciója

- ▶ $\forall 1 \leq i \leq n : (x_{i-1}, u_i), (x_{i-1}, v_i), (u_i, x_i), (v_i, x_i) \in E(G_\varphi)$
- ▶ $s := x_0, t := x_n$
- ▶ $\forall 1 \leq i \leq n$ -re u_i és v_i között $2m$ belső pontú kétirányú út $w_{i,1}, \dots, w_{i,2m}$.
- ▶ Minden $w_{i,k}$ legfeljebb egy C_j -vel lehet összekötve.
- ▶ Ha $X_i \in C_j$, akkor $(w_{i,k}, C_j)$ és $(C_j, w_{i,k+1}) \in E(G_\varphi)$. (pozitív bekötés)
- ▶ Ha $\neg X_i \in C_j$, akkor $(w_{i,k+1}, C_j)$ és $(C_j, w_{i,k}) \in E(G_\varphi)$. (negatív bekötés)

Az $u_i v_i$ út pozitív bejárása: $u_i \rightsquigarrow v_i$.

Az $u_i v_i$ út negatív bejárása: $u_i \leftrightsquigarrow v_i$.

Irányított $s - t$ -Hamilton út NP teljessége

- ▶ Egy $s - t$ H-út $\forall 1 \leq i \leq n$ -re az (x_{i-1}, u_i) és (x_{i-1}, v_i) közül pontosan egyiket tartalmazza, előbbi esetben az $u_i v_i$ utat pozitív, utóbbi esetben negatív irányban járja be.
- ▶ Egy $s - t$ H-út minden C_j -t pontosan egyszer köt be, meggondolható, hogy az $u_i v_i$ út pozitív bejárása esetén csak pozitív, negatív bejárása esetén csak negatív bekötés lehetséges.
- ▶ Ha van H-út, akkor az $u_i v_i$ utak pozitív/negatív bejárása meghatároz egy változókiértékelést, a $\forall 1 \leq j \leq m : C_j$ klóz bekötése mutat C_j -ben egy igaz literált.
- ▶ Ha φ kielégíthető, válasszunk egy őt igazra kiértékelő interpretációt és ebben minden klózhoz egy igaz literált. Az $u_i v_i$ utaknak válasszuk igaz változók esetén a pozitív, egyébként a negatív bejárását. Ha a kiválasztott literálokhoz rendre bekötjük a C_j csúcsokat H-utat kapunk.

G_φ polinom időben megkonstruálható így $\text{SAT}_{\leq p}$ HÚ, azaz HÚ NP-nehéz, de láttuk, hogy NP-beli, így NP-teljes is.

Irányítatlan $s - t$ -Hamilton út NP teljessége

Megjegyzés: IHÚ és IHK NP-belisége az előzőekhez hasonlóan adódik.

Tétel

IHÚ NP-teljes

Bizonyítás: $HÚ \leq_p IHÚ$. Adott G, s, t , ahol G irányított. Kell G', s', t' , ahol G' irányítatlan és akkor és csak akkor van G -ben s -ből t -be H-út, ha G' -ben van s' -ből t' -be.

G minden v csúcsának feleljen meg G' -ben 3 csúcs v_{be} , $v_{közép}$ és v_{ki} . és G' élei közé vegyük be a $\{v_{be}, v_{közép}\}$ és $\{v_{közép}, v_{ki}\}$ éleket. Továbbá minden $E = (u, v)$ G -beli él estén adjuk hozzá $E(G')$ -hez $\{u_{ki}, v_{be}\}$ -t.

$s' := s_{be}, t' := t_{ki}$.

Könnyű látni, hogy ha P H-út G -ben, akkor P' H-út G' -ben, ahol P' -t úgy kapjuk P -ből, hogy minden v csúcsot v_{be} , $v_{közép}$ és v_{ki} -vel helyettesítünk, ebben a sorrendben.

Irányítatlan Hamilton út/kör NP teljes

Fordítva, könnyen látható, hogy ha P egy H-út G' -ben akkor v_{be} , $v_{közép}$, v_{ki} sorrendű 3-asok követik egymást (különben a $v_{közép}$ -eket nem tudnánk felfűzni). Ezeket a 3-asokat v -vel helyettesítve egy G -beli utat kapunk.

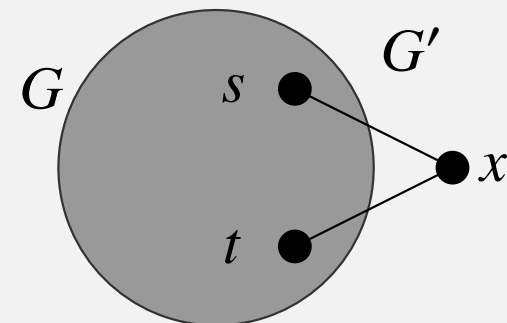
Az utak kezdetére és végére vonatkozó feltételek is teljesülnek.

Tétel

IHK NP-teljes

Bizonyítás: $IH\dot{U} \leq_p IHK$. Adott G, s, t . G' konstrukciójában adjunk hozzá egy új x csúcsot és két új élt $\{s, x\}$ -et és $\{t, x\}$ -t G -hez.

Könnyen meggondolható, hogy akkkor és csak akkor van G -ben s - t H-út, ha G' -ben van H-kör.



Az utazóügynök probléma

Számítási (optimalizálási) verzió: Adott egy G élsúlyozott irányítatlan gráf nemnegatív élsúlyokkal. Határozzuk meg a legkisebb összsúlyú H-kört (ha van).

Eldöntési verzió:

$\text{TSP} = \{ \langle G, K \rangle \mid G\text{-ben van } \leq K \text{ súlyú H-kör} \}.$

Tétel

TSP NP-teljes

Bizonyítás: $\text{TSP} \in \text{NP}$, hasonló érvek miatt, mint HÚ, az összköltségfeltétel is polinom időben ellenőrizhető.

$\text{IHK} \leq_p \text{TSP}$. Adott egy G gráf. G függvényében konstruálunk egy G' élsúlyozott gráfot és megadunk egy K számot. $G' := G$, minden élsúly legyen 1 és $K := |V|$. Könnyen látható, hogy G -ben van H-kör $\Leftrightarrow G'$ -ben van legfeljebb K összsúlyú H-kör.

NP szerkezete

NP-köztes nyelv

L NP-köztes, ha $L \in \text{NP}$, $L \notin \text{P}$ és L nem NP-teljes.

Ladner tétele

Ha $\text{P} \neq \text{NP}$, akkor létezik NP-köztes nyelv.

(biz. nélkül)

NP-köztes jelöltek (persze egyikről se tudhatjuk):

- ▶ GRÁFIZOMORFIZMUS = $\{\langle G_1, G_2 \rangle \mid G_1 \text{ és } G_2 \text{ irányítatlan izomorf gráfok}\}$.
- ▶ PRÍMFAKTORIZÁCIÓ: adjuk meg egy egész szám prímtényezős felbontását [számítási feladat],

Egy új eredmény: Babai László, magyar matematikus (még nem lektorált) eredménye: GRÁFIZOMORFIZMUS \in QP, ahol

$$\text{QP} = \bigcup_{c \in \mathbb{N}} \text{TIME}(2^{(\log n)^c})$$

a "kvázipolinom időben" megoldható problémák osztálya.

co \mathfrak{C} bonyolultsági osztály

Ha \mathfrak{C} egy bonyolultsági osztály $\text{co}\mathfrak{C} = \{L \mid \bar{L} \in \mathfrak{C}\}$.

Bonyolultsági osztály polinom idejű visszavezetésre való zártsága

\mathfrak{C} **zárt a polinomidejű visszavezetésre nézve**, ha minden esetben ha $L_2 \in \mathfrak{C}$ és $L_1 \leq_p L_2$ teljesül következik, hogy $L_1 \in \mathfrak{C}$.

Volt: P és NP zártak a polinomidejű visszavezetésre nézve.

Tétel

Ha \mathfrak{C} zárt a polinomidejű visszavezetésre nézve, akkor $\text{co}\mathfrak{C}$ is.

Bizonyítás: Legyen $L_2 \in \text{co}\mathfrak{C}$ és L_1 tetszőleges nyelvek, melyekre $L_1 \leq_p L_2$. Utóbbiból következik, hogy $\bar{L}_1 \leq_p \bar{L}_2$ (ugyananaz a visszavezetés jó!). Mivel $\bar{L}_2 \in \mathfrak{C}$, ezért a tétel feltétele miatt $\bar{L}_1 \in \mathfrak{C}$. Azaz $L_1 \in \text{co}\mathfrak{C}$.

Igaz-e, hogy $P=coP$? Igen. (L -et polinom időben eldöntő TG q_i és q_n állapotát megcseréljük: \bar{L} -t polinom időben eldöntő TG.)

Igaz-e, hogy $NP=coNP$? A fenti konstrukció NTG-re nem feltétlen \bar{L} -t dönti el.

Következmény

coNP zárt a polinom idejű visszavezetésre nézve.

Tétel

$L \in \mathfrak{C} \iff \bar{L} \in co\mathfrak{C}$.

Bizonyítás:

- ▶ Ha $L \in \mathfrak{C}$, akkor $\bar{L} \in co\mathfrak{C}$.
- ▶ Legyen $L' \in \mathfrak{C}$, melyre $L' \leq_p L$. Ekkor $\bar{L}' \leq_p \bar{L}$.
Ha L' befutja \mathfrak{C} -t akkor \bar{L}' befutja $co\mathfrak{C}$ -t. Azaz minden $co\mathfrak{C}$ -beli nyelv polinom időben visszavezethető \bar{L} -re.

Tehát \bar{L} $co\mathfrak{C}$ -beli és $co\mathfrak{C}$ -nehéz, így $co\mathfrak{C}$ -teljes.

Példák coNP teljes nyelvekre

$\text{UNSAT} := \{\langle \varphi \rangle \mid \varphi \text{ kielégíthetetlen nulladrendű formula}\}.$

$\text{TAUT} := \{\langle \varphi \rangle \mid \text{a } \varphi \text{ nulladrendű formula tautológia}\}.$

Tétel

UNSAT és TAUT coNP-teljesek.

Bizonyítás: $\text{ÁLTSAT} = \{\langle \varphi \rangle \mid \varphi \text{ kielégíthető nulladrendű formula}\}$ is NP-teljes (NP-beli és SAT speciális esete neki.)

$\text{ÁLTSAT} = \text{UNSAT}$, az előző tétel alapján UNSAT coNP-teljes.

$\text{UNSAT} \leq_p \text{TAUT}$, hiszen $\varphi \mapsto \neg \varphi$ polinom idejű visszavezetés.

Informálisan: coNP tartalmazza a polinom időben cáfolható problémákat.

Megjegyzések: Sejtés, hogy $\text{NP} \neq \text{coNP}$. Egy érdekes osztály ekkor a $\text{NP} \cap \text{coNP}$. Sejtés: $\text{P} \neq \text{NP} \cap \text{coNP}$. Bizonyított, hogy ha egy coNP-teljes problémáról kiderülne, hogy NP-beli, akkor $\text{NP} = \text{coNP}$.

Tárbonyolultság

Probléma a tárbonyolultság mérésénél: Hiába "takarékos" a felhasznált cellákkal a gép, az input hossza mindig alsó korlát lesz a felhasznált tárterületre. Egy megoldási lehetőség: A valódi tárigény az **ezen felül** igénybevett cellák száma. A csak az input területét használó számításoknak 0 legyen a tárigénye? Ez se az igazi.

Off-line Turing-gép

Az **off-line Turing-gép** egy legalább 3 szalagos gép, amelynek az első szalagja csak olvasható, az utolsó szalagja csak írható. További szalagjait munkaszalagoknak nevezzük.

Off-line TG-ek tárigénye

Az off-line TG **tárigénye** egy adott inputra a munkaszalagjain felhasznált cellák száma. Egy TG $f(n)$ **tárkorlátos**, ha bármely u inputra legfeljebb $f(|u|)$ tárat használ.

Determinisztikus és nemdeterminisztikus tárnyolultsági osztályok

Így az off-line TG-pel **szublineáris** (lineáris alatti) tárnyolultságot is mérhetünk.

- ▶ $\text{SPACE}(f(n)) := \{L \mid L \text{ eldönthető } O(f(n)) \text{ tárkorlátos determinisztikus off-line TG-pel}\}$
- ▶ $\text{NSPACE}(f(n)) := \{L \mid L \text{ eldönthető } O(f(n)) \text{ tárkorlátos nemdeterminisztikus off-line TG-pel}\}$
- ▶ $\text{PSPACE} := \bigcup_{k \geq 1} \text{SPACE}(n^k).$
- ▶ $\text{NPSPACE} := \bigcup_{k \geq 1} \text{NSPACE}(n^k).$
- ▶ $\text{L} := \text{SPACE}(\log n).$
- ▶ $\text{NL} := \text{NSPACE}(\log n).$

Az ELÉR probléma

$\text{ELÉR} = \{ \langle G, s, t \rangle \mid \text{A } G \text{ irányított gráfban van } s\text{-ből } t\text{-be út} \}.$

$\text{ELÉR} \in \text{P}$ (valójában $O(n^2)$, lásd Algoritmusok és adatszerk. II., szélességi/mélységi bejárás)

Tétel

$\text{ELÉR} \in \text{SPACE}(\log^2 n).$

Bizonyítás:

- ▶ Rögzítsük a csúcsok egy tetszőleges sorrendjét.
- ▶ $\text{ÚT}(x, y, i) := \text{igaz}$, ha létezik x -ből y -ba legfeljebb 2^i hosszú út.
- ▶ s -ből van t -be út G -ben $\iff \text{ÚT}(x, y, \lceil \log n \rceil) = \text{igaz}$.
- ▶ $\text{ÚT}(x, y, i) = \text{igaz} \iff \exists z (\text{ÚT}(x, z, i - 1) = \text{igaz} \wedge \text{ÚT}(z, y, i - 1) = \text{igaz})$.
- ▶ Ez alapján egy rekurzív algoritmust készítünk, melynek persze munkaszalagján tárolnia kell, hogy a felsőbb szinteken milyen (x, y, i) -kre létezik folyamatban lévő hívás.

ELÉR: az $\text{ÚT}(x, y, i)$ algoritmus

- ▶ ha $i = 0$, akkor $2^0 = 1$ hosszú út (megnézi az inputot).
- ▶ A munkaszalagon (x, y, i) típusú hármasok egy legfeljebb $\lceil \log n \rceil$ hosszú sorozata áll. A hármasok 3. attribútuma 1-esével csökkenő sorozatot alkot $\lceil \log n \rceil$ -től
- ▶ $\text{ÚT}(x, y, i)$ meghívásakor az utolsó hármas (x, y, i) a munkaszalagon. Az algoritmus felírja az $(x, z, i - 1)$ hármaszt a munkaszalagra (x, y, i) utáni helyre majd kiszámítja $\text{ÚT}(x, z, i - 1)$ értékét.
- ▶ Ha hamis, akkor kitörli $(x, z, i - 1)$ -et és z értékét növeli.
- ▶ Ha igaz, akkor is kitörli $(x, z, i - 1)$ -et és $(z, y, i - 1)$ -et ráírja, (y -t tudja az előző (x, y, i) hármasból).
 - Ha igaz, akkor $\text{ÚT}(x, y, i)$ igaz, visszalép ((x, y, i) és $(z, y, i - 1)$ 2. argumentumának egyezéséből látja)
 - Ha hamis akkor kitörli és z értékét eggyel növelve $\text{ÚT}(x, z, i - 1)$ -en dolgozik tovább.
- ▶ Ha egyik z se volt jó, akkor $\text{ÚT}(x, y, i)$ hamis.

Konfigurációs gráf

Az $\text{ÚT}(s, t, \lceil \log n \rceil)$ algoritmus a munkaszalagján $O(\log n)$ darab tagból álló egyenként $O(\log n)$ hosszú (x, y, i) hármast tárol, így $\text{ELÉR} \in \text{SPACE}(\log^2 n)$.

Konfigurációs gráf

Egy M TG G_M konfigurációs gráfjának csúcsai M konfigurációi és $(C, C') \in E(G_M) \Leftrightarrow C \vdash_M C'$.

Elérhetőségi módszer: bonyolultsági osztályok közötti összefüggéseket lehet bizonyítani az $\text{ELÉR} \in P$ vagy $\text{ELÉR} \in \text{SPACE}(\log^2 n)$ tételeket alkalmazva a konfigurációs gráfra, vagy annak egy részgráfjára.

Savitch tétele

Savitch tétele

Ha $f(n) \geq \log n$, akkor $\text{NSPACE}(f(n)) \subseteq \text{SPACE}(f^2(n))$.

Bizonyítás: Legyen M egy $f(n)$ tárigényű NTG és w az M egy n hosszú bemenete.

Ekkor M egy konfigurációját $O(f(n) + \log n)$ tárral eltárolhatjuk (aktuális állapot, a munkaszalagok tartalma, fejek pozíciója, az első szalag fejének pozíciója n féle lehet, ezért $\geq \log n$ tár kell ennek eltárolásához). Ha $f(n) \geq \log n$, akkor ez $O(f(n))$.

Feltehető, hogy M -nek csak egy elfogadó konfigurációja van.

(Törölje le a TG a munkaszalagjait, mielőtt q_i -be lép!)

A legfeljebb ekkora méretű konfigurációkat tartalmazó konfigurációs gráf mérete $2^{df(n)}$ valamely $d > 0$ konstansra. Így az előző tétel szerint $O(\log^2(2^{df(n)})) = O(f^2(n))$ tárral egy determinisztikus TG eldönti, hogy

$\text{ÚT}(c_{\text{kezdő}}, c_{\text{elfogadó}}, \lceil \log(2^{df(n)}) \rceil)$ igaz-e.

Savitch tétele

Következmények

Következmény

$$\text{PSPACE} = \text{NPSPACE}$$

Bizonyítás: polinom négyzete is polinom.

Tétel

$$\text{NL} \subseteq \text{P}$$

Bizonyítás

Legyen $L \in \text{NL}$ és M L -et $f(n) = O(\log n)$ tárral eldöntő NTG.

Meggondolható, hogy egy n méretű inputra M legfeljebb $f(n)$ méretű szalagtartalmakat tartalmazó konfigurációinak a száma legfeljebb $cnd^{\log n}$ alkalmas c, d konstansokkal, ami egy $p(n)$ polinommal felülről becsülhető. Így a G konfigurációs gráfnak legfeljebb $p(n)$ csúcsa van. G polinom időben megkonstruálható.

Feltehető, hogy G -ben egyetlen elfogadó konfiguráció van. G -ben a kezdőkonfigurációból az elfogadó konfiguráció elérhetősége $O(p^2(n))$ idejű determinisztikus TG-pel eldönthető, azaz $L \in \text{P}$.

L és NL

ELÉR fontos szerepet tölt be az $L \stackrel{?}{=} NL$ kérdés vizsgálatában is.

Tétel

$ELÉR \in NL$

Bizonyítás: Az M 3-szalagos NTG a (G, s, t) inputra ($n = |V(G)|$) a következőt teszi:

- ▶ ráírja s -t a második szalagra
- ▶ ráírja a 0 -t a harmadik szalagra
- ▶ Amíg a harmadik szalagon n -nél kisebb szám áll
 - Legyen u a második szalagon lévő csúcs
 - Nemdeterminisztikusan felírja u helyére egy v ki-szomszédját a második szalagra
 - Ha $v = t$, akkor elfogadja a bemenetet, egyébként növeli a harmadik szalagon lévő számot binárisan eggyel
- ▶ Elutasítja a bemenetet
- ▶ Mindkét szalag tartalmát $O(\log n)$ hosszú kóddal tárolhatjuk.

L és NL

Log. táras visszavezetés

Egy $L_1 \subseteq \Sigma^*$ nyelv **logaritmikus tárral visszavezethető** egy $L_2 \subseteq \Delta^*$ nyelvre $L_1 \leq_\ell L_2$, ha $L_1 \leq L_2$ és a visszavezetéshez használt függvény kiszámítható logaritmikus táras determinisztikus (off-line) Turing-géppel

NL-nehéz, NL-teljes nyelv

Egy L nyelv **NL-nehéz** (a log. táras visszavezetésre nézve), ha minden $L' \in \text{NL}$ nyelvre, $L' \leq_\ell L$; ha ráadásul $L \in \text{NL}$ is teljesül, akkor L **NL-teljes** (a log. táras visszavezetésre nézve)

Tétel

L zárt a logaritmikus tárral való visszavezetésre nézve

Bizonyítás: Tegyük fel, hogy $L_1 \leq_\ell L_2$ és $L_2 \in \text{NL}$.

Legyen M_2 az L_2 -t eldöntő, M pedig a visszavezetésben használt f függvényt kiszámoló logaritmikus táras determinisztikus TG.

L és NL

Az M_1 TG egy tetszőleges u szóra a következőképpen működik

- ▶ A második szalagján egy bináris számlálóval nyomon követi, hogy M_2 feje hányadik betűjét olvassa az $f(u)$ szónak; legyen ez a szám i (kezdetben 1)
- ▶ Amikor M_2 lépne egyet, akkor M_1 az M -et szimulálva előállítja a harmadik szalagon $f(u)$ i -ik betűjét (de csak ezt a betűt!!!)
- ▶ Ezután M_1 szimulálja M_2 aktuális lépését a harmadik szalagon lévő betű felhasználásával és aktualizálja a második szalagon M_2 fejének újabb pozícióját
- ▶ Ha eközben M_1 azt látja, hogy M_2 elfogadó vagy elutasító állapotba lép, akkor M_1 is belép a saját elfogadó vagy elutasító állapotába, egyébként folytatja a szimulációt a következő lépéssel

Belátható, hogy M_1 L_1 -et dönti el és a működése során csak logaritmikus méretű tárat használ, azaz $L_1 \in L$.

ELÉR NL-teljesége

Következmény

Ha L NL-teljes és $L \in L$, akkor $L = NL$.

Bizonyítás: Legyen $L' \in NL$ tetszőleges, ekkor $L \leq_\ell L$ és $L \in L$, így L logaritmikus tárral való visszavezetésre nézve zártsága miatt $L' \in NL$.
Tehát $L = NL$.

Tétel

ELÉR NL-teljes a logaritmikus tárral történő visszavezetésre nézve.

Bizonyítás:

- ▶ Korábban láttuk, hogy $ELÉR \in NL$
- ▶ Legyen $L \in NL$, megmutatjuk, hogy $L \leq_\ell ELÉR$
- ▶ Legyen M egy L -et eldöntő $O(\log n)$ táras NTG és $|u| = n$
- ▶ Az $O(\log n)$ tárat használó konfigurációk $\leq c \cdot \log n$ hosszúak (alkalmas c -re)

ELÉR NL-teljesége; Immerman-Szelepcsényi

- ▶ A G_M konfigurációs gráfban akkor és csak akkor lehet a kezdőkonfigurációból az elfogadóba jutni (feltehető, hogy csak 1 ilyen van), ha $u \in L(G)$. Így $L \leq \text{ELÉR}$.

Kell még, hogy a visszavezetés log. tárat használ, azaz G_M megkonstruálható egy log. táras N determinisztikus TG-pel:

- ▶ N sorolja fel a hossz-lexikografikus rendezés szerint az összes legfeljebb $c \cdot \log n$ hosszú szót az egyik szalagján, majd tesztelje, hogy az legális konfigurációja-e M -nek, ha igen, akkor a szót írja ki a kimenetre
- ▶ Az élek (konfiguráció párok) hasonlóképpen felsorolhatók, tesztelhetők és a kimenetre írhatók

Immerman-Szelepcsényi tétel

$\text{NL} = \text{coNL}$

(biz. nélkül)

Hierarchia tétel

$$\text{EXPTIME} := \bigcup_{k \in \mathbb{N}} \text{TIME}(2^{n^k}).$$

Tétel

$\text{NL} \subset \text{PSPACE}$ és $\text{P} \subset \text{EXPTIME}$.

(biz. nélkül)

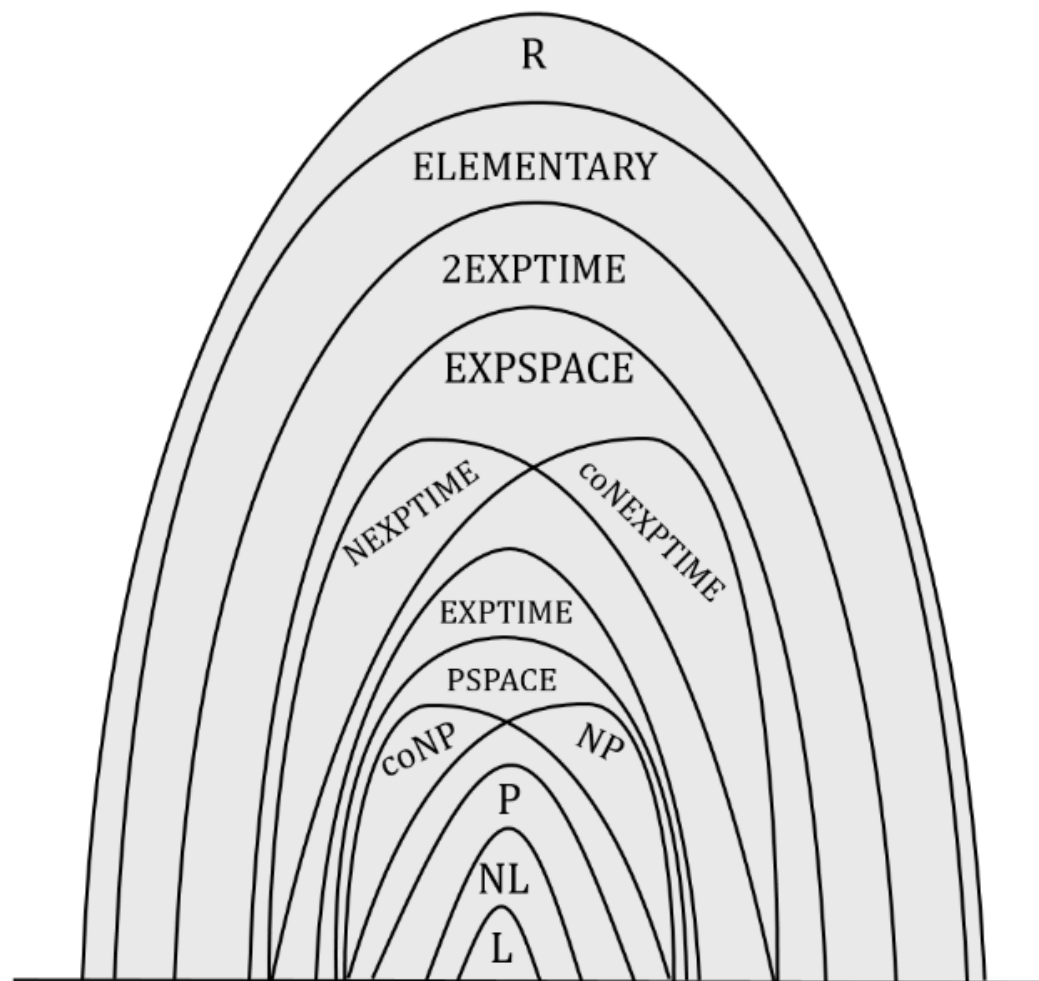
Tétel

$\text{L} \subseteq \text{NL} = \text{coNL} \subseteq \text{P} \subseteq \text{NP} \subseteq \text{PSPACE} \subseteq \text{EXPTIME}$

Bizonyítás: 1 és 4 definíció szerint, 2-es Immerman-Szelepcsényi, 3-ast előbb bizonyítottuk. 5-ös: egy TG-nek "nincs ideje" több tárat használni, mint időt. 6-os: elérhetőségi módszerrel. A használt tárban exponenciális méretű lesz a konfigurációs gráf, a gráf méretében négyzetes (azaz összességében a tár méretében exponenciális) időben tudjuk az elérhetőséget tesztelni a kezdőkonfigurációból az elfogadóba konfigurációba.

Sejtés: Utóbbiban minden tartalmazás valódi.

R szerkezete



R szerkezete $P \neq NP$ esetén [ábra: Gazdag Zs. jegyzet]