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 hatékonynak?

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?*
   algoritmikus 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?* **algoritmikus megoldás:** szélességi keresés **Hatékonysága:** élszám *lineáris* függvénye

Bevezetés

Példák problémákra

3. A nulladrendű logika eldöntésproblémája. Egy formulahalmaznak következménye-e egy adott formula? *Tanultuk, hogy elég egy klózhalmaz (vagy KNF) kielégíthetőségét vizsgálni.* SAT probléma: kielégíthető-e adott KNF. Egy **algoritmikus megoldás:** ítélettábla. Exponenciális futási idő. Nem ismeretes (és majd látni fogjuk, hogy nem is várható) polinom idejű algoritmikus megoldás.
4. Egy város neveztességeit (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 (~ 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.

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 eldöntésproblémája (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 ~ UA létezése).

Bevezetés

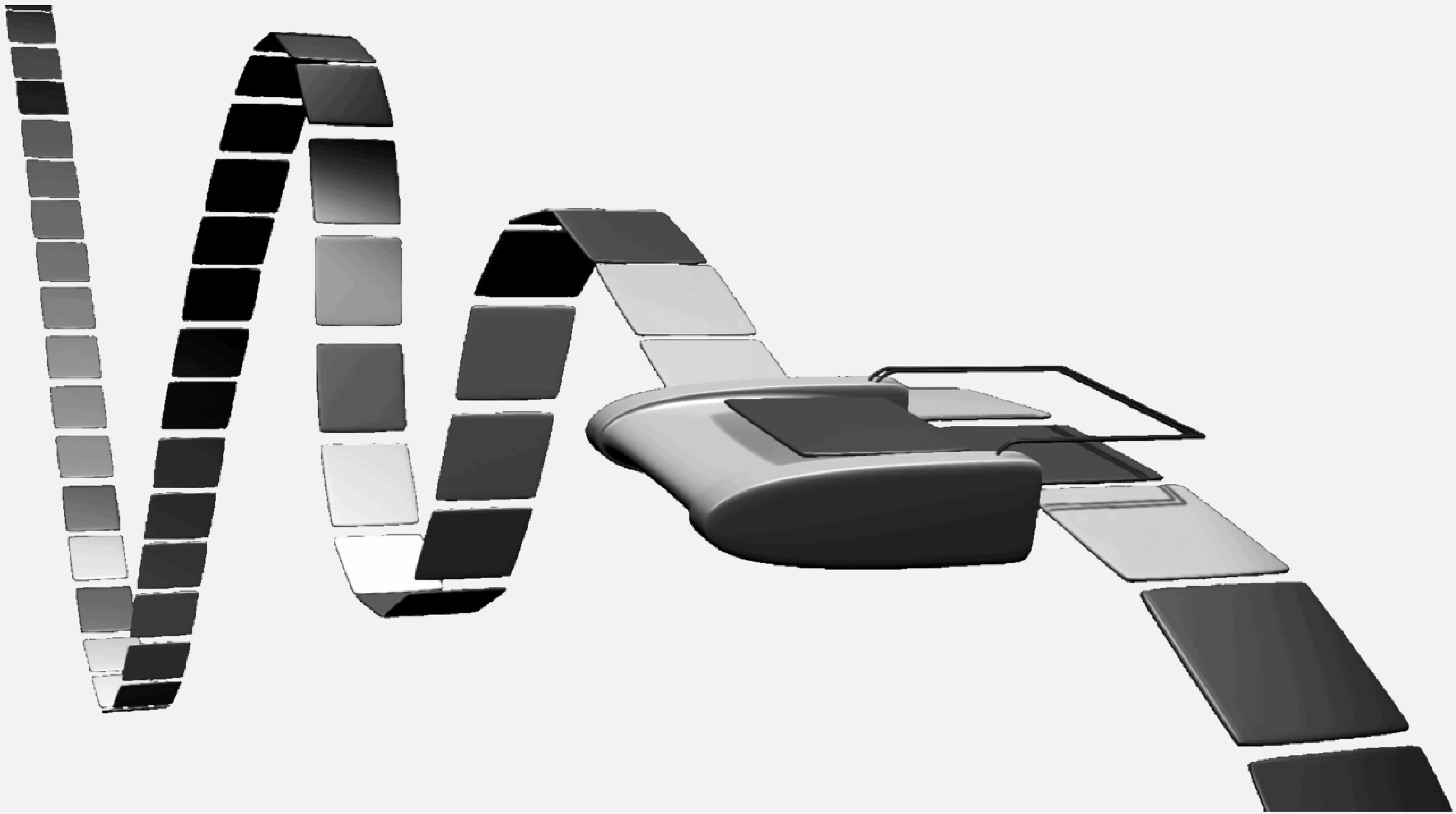
Példák problémákra

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 diophanthoszi egyenletek problémájára nincs algoritmikus megoldás.

Tematika

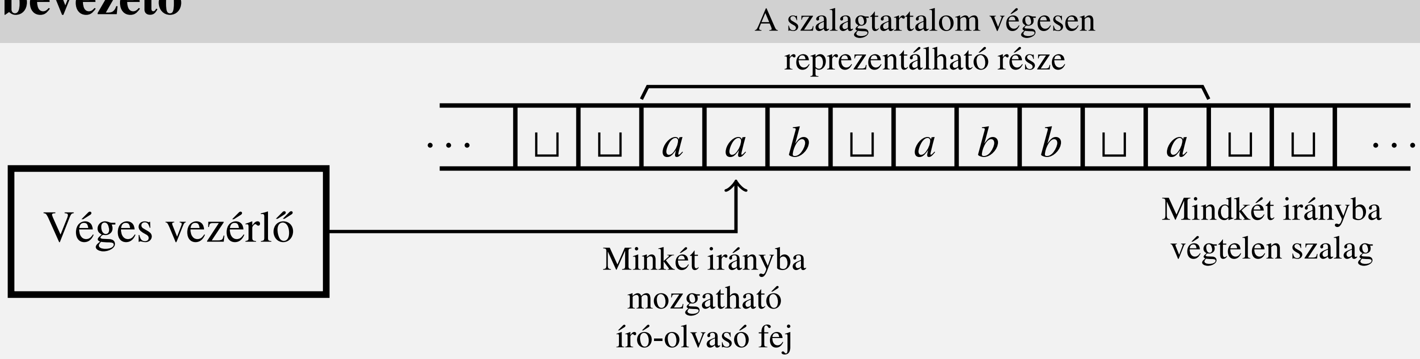
1. Turing gép és változatai, mint algoritmusmodell
2. Algoritmikusan eldönthetetlen problémák
3. Eldönthető problémák hatékonyságáról: bevezetés a bonyolultságelméletbe

Turing gépek



Turing gépek

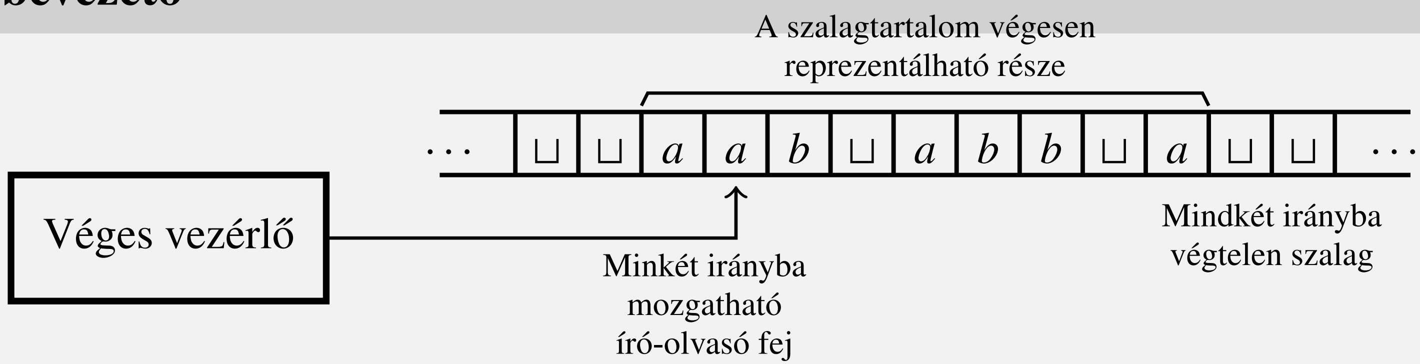
Informális bevezető



- 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 példányait 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 $\sqcup \in \Gamma \setminus \Sigma$.
- $\delta : (Q \setminus \{q_i, q_n\}) \times \Gamma \to Q \times \Gamma \times \{L, S, R\}$ 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.

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

Konfigurácóátmenet

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 $\exists\, n > 0 \wedge C_1, C_2, \ldots \in C_n \in C_M$, 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

$L(M) = \{u \in \Sigma^* \mid q_0 u \sqcup \vdash^* x q_i y$ 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 felsorolható**nak (vagy *parciálisan rekurzívnak*, vagy *félig eldönthetőnek*) az eldönthető nyelveket pedig **rekurzív**nak is nevezni.

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

Nyilván $R \subseteq RE$. 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.

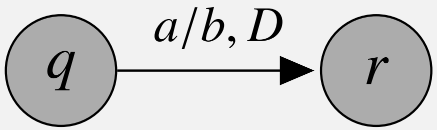
Legyen $f : \mathbb{N} \to \mathbb{N}$ egy függvény. Azt mondjuk, hogy M **időigénye** $f(n)$ (vagy, hogy M egy $f(n)$ időkorlátos gép), ha minden $u \in \Sigma^*$ input szóra, M időigénye az u szón legfeljebb $f(|u|)$.

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

Turing gépek

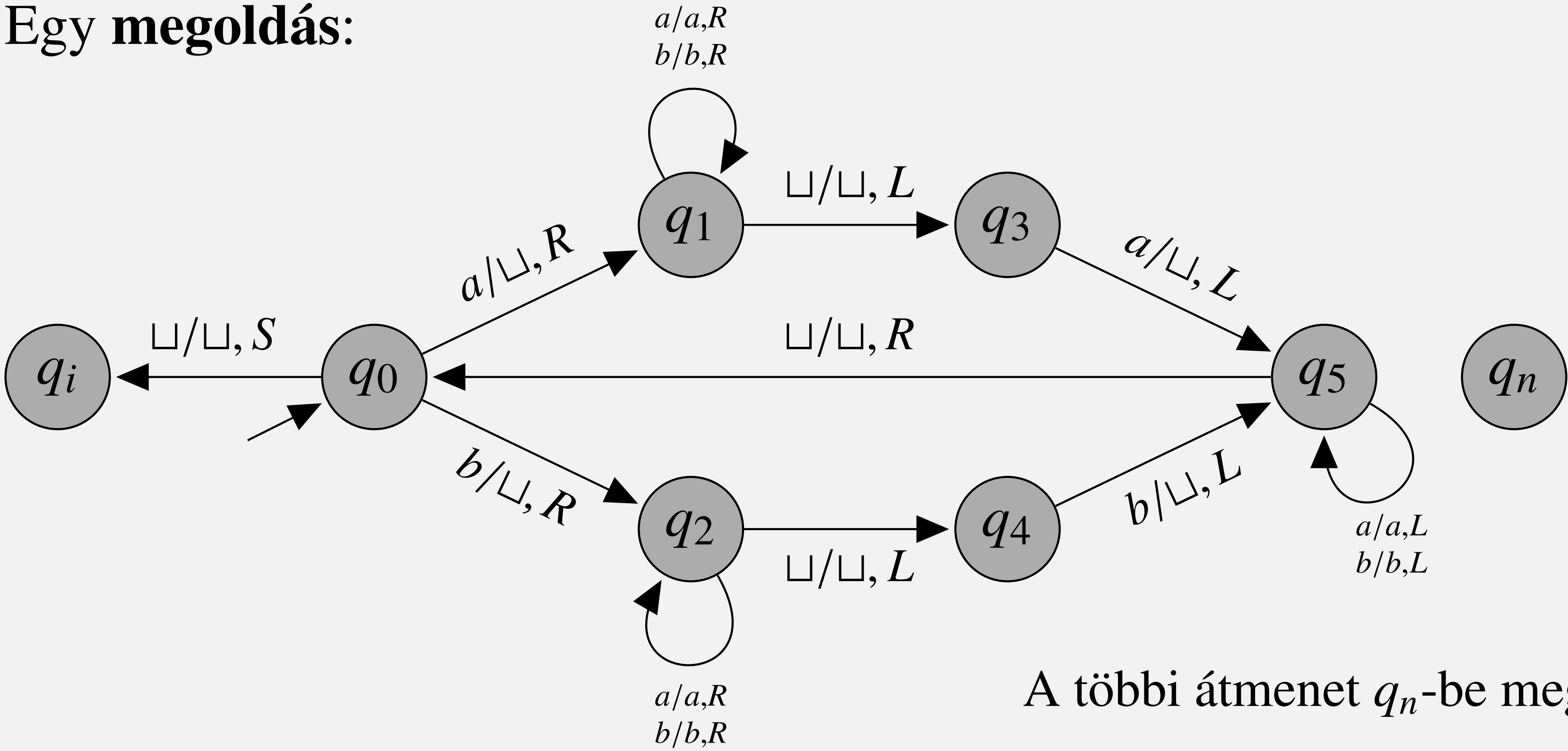
Egy példa

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

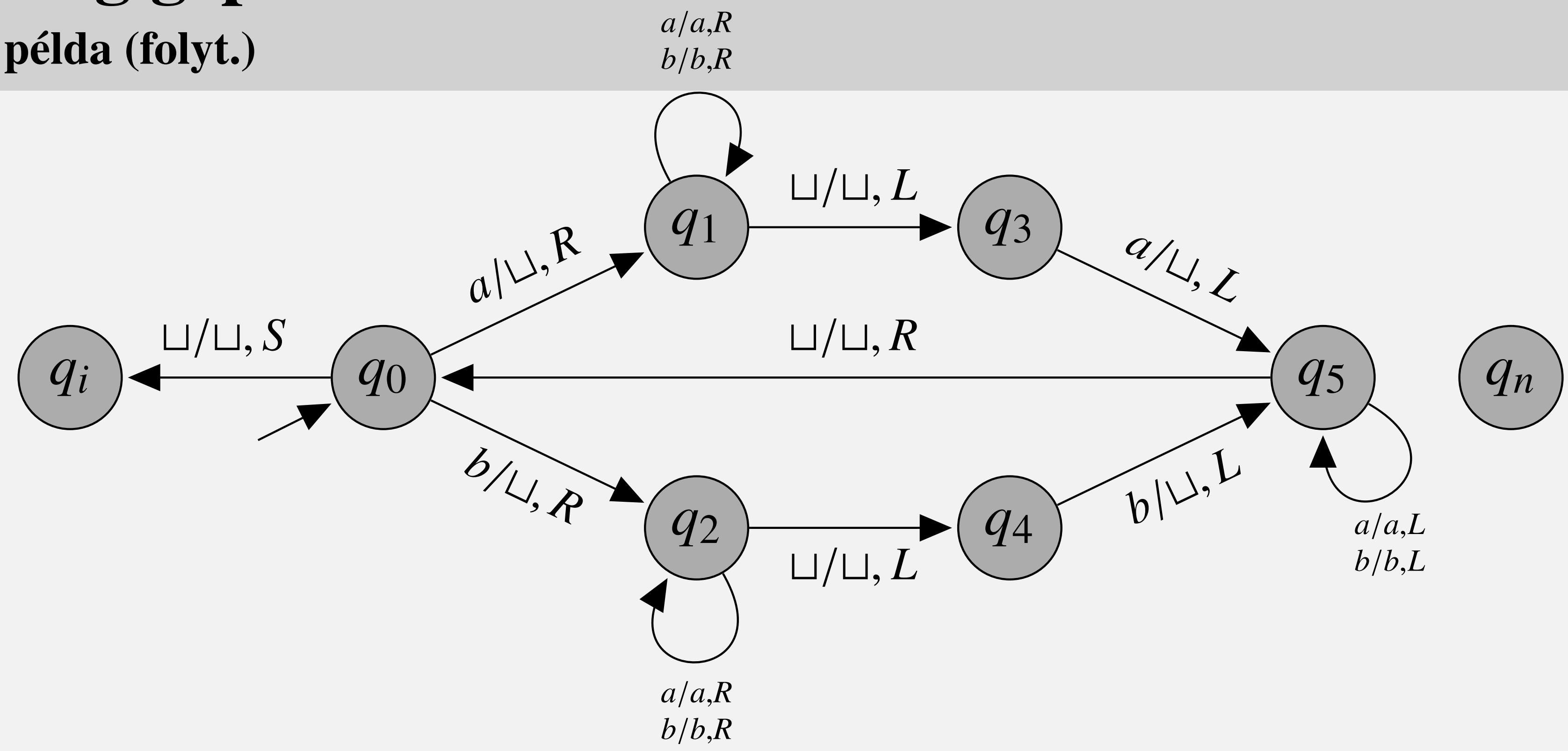
Egy **megoldás**:



A többi átmenet q_n-be megy.

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

Definició

Legyenek $f, g : \mathbb{N} \to \mathbb{R}_0^+$ 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 $\mathbb{N} \to \mathbb{R}_0^+$ 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 $\mathbb{N} \to \mathbb{R}_0^+$ 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 $f(n)/g(n) \to +\infty \Rightarrow f(n)=\Omega(g(n))$ és $f(n)\neq O(g(n))$

  ha $f(n)/g(n) \to c \quad (c > 0) \Rightarrow f(n) = \Theta(g(n))$

  ha $f(n)/g(n) \to 0 \quad \Rightarrow f(n) = O(g(n))$ és $f(n) \neq \Omega(g(n))$

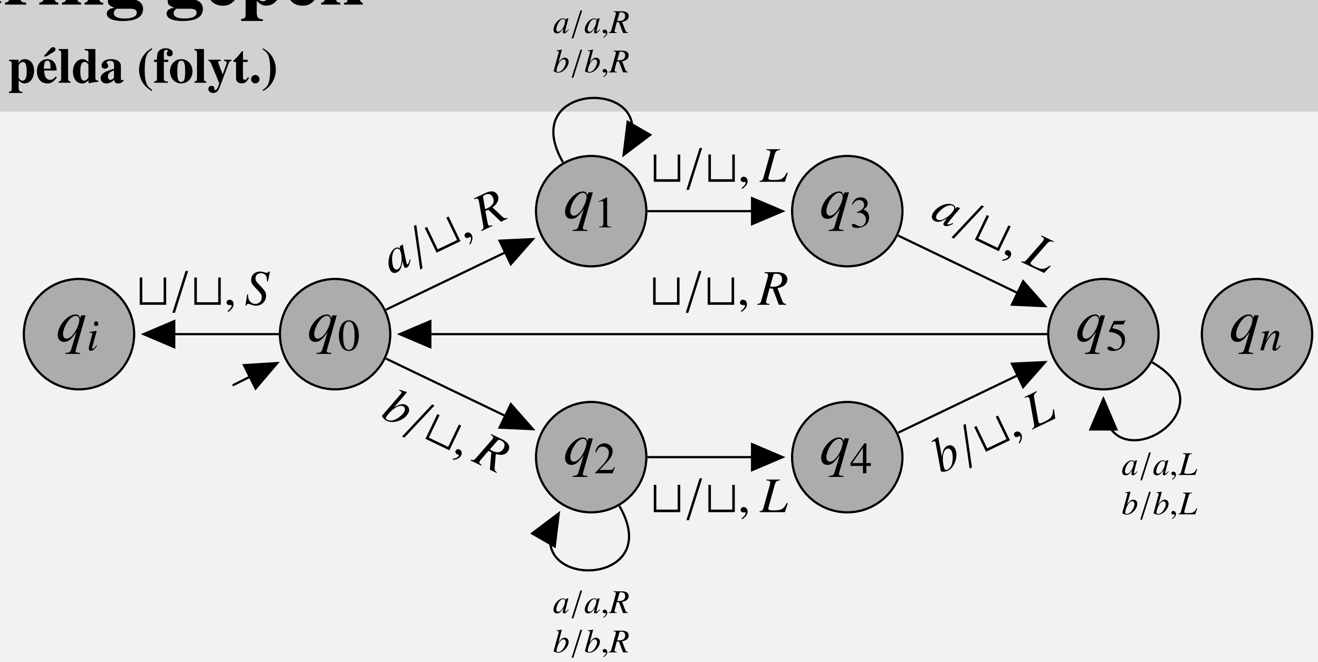
Függvények aszimptotikus nagyságrendje

Konkrét függvények

- $p(n) = a_k n^k + \cdots + a_1 n + a_0$ ($a_k > 0$), 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.