Bonyolultságelmélet

Monday 26th September, 2016, 18:28

A kurzus teljesítési követelményei

Gyakorlat

- Három kisdolgozat 6–6 pontért kb. a 4., 7. és 10. gyakorlaton
- Egy nagydolgozat 28 pontért utolsó héten előadáson
- Pontszám: **két legjobb** kis- plusz a nagydolgozat pontszáma
- Ha ≥ 16, a gyakorlat teljesítve. Ponthatárok: 16, 22, 28, 34.
- Egyébként a nagydolgozat javítása első vizsgaidőpontban
- Ha így már ≥ 16, a gyakorlati jegy elégséges
- Ha így sem, a gyakorlat és a kurzus nem teljesített

Vizsga

- Tíz kiskérdés $3-3$ pontért, minimum: 12
- Egy esszé 30 pontért, minimum: 7 (szóbeli)
- Ponthatárok: 19, 30, 41, 51

Az előadás felépítése

- A kiszámíthatóság- és a bonyolultságelmélet kialakulása
- Turing-gépek. A kiszámítás $\textsc{Ram}$ modellje.
- Problémák egymáshoz viszonyított nehézsége: a (hatékony) visszavezetés
- Eldönthetetlen problémák
- Bonyolultsági osztályok
- Teljesség, nehézség; $\mathbf{NP}$-teljes problémák
- Approximáció, pszeudopolinomiális algoritmusok, parametrizált komplexitás
- $\mathbf{PSPACE}$-nehéz problémák
- Randomizált algoritmusok
- Párhuzamosítás
- Kriptográfia

A kiszámíthatóság elméletének kialakulása

1900: Hilbert 10. problémája (a 23-ból)

Adott $p(x_1, \ldots, x_n)$ **egész együtthatós polinom**, létezik-e (egész értékű) zérushelye.

1928: Hilbert

Találjunk olyan algoritmust, amellyel a **predikátumkalkulus** tetszőleges formulájáról eldönthetjük, hogy **tautológia**-e.

Megjegyzés

Algoritmusból sokkal kevesebb (megszámlálhatóan végtelen) van, mint eldöntési problémából (kontinuum) – Cantor, 1874 –, emiatt a problémák **túlnyomó többsége** megoldhatatlan.

A kiszámíthatóság elméletének kialakulása

Mi az, hogy valami kiszámítható?

1931, Gödel: Primitív rekurzív függvények

- $f(n) = 0$;
- $s(n) = n + 1$;
- $\pi_k^i(x_1, \ldots, x_k) = x_i$;
- $f/n,\ g_1/k, \ldots, g_n/k \Rightarrow h(\boldsymbol{x}) = f(g_1(\boldsymbol{x}), \ldots, g_n(\boldsymbol{x}))$;
- $f/k,\ g/(k+2) \Rightarrow$

$$h(0, \boldsymbol{x}) = f(\boldsymbol{x}),$$
$$h(n+1, \boldsymbol{x}) = g(n, h(n, \boldsymbol{x}), \boldsymbol{x}).$$

a primitív rekurzív függvények.

Néhány primitív rekurzív függvény

Példák

- $\mathrm{add}(0, x) = x$, $\mathrm{add}(n + 1, x) = \mathrm{add}(n, x) + 1$ – összeadás
- $\mathrm{mult}(0, x) = 0$, $\mathrm{mult}(n + 1, x) = \mathrm{add}(\mathrm{mult}(n, x), x)$ – szorzás
- $\mathrm{pow}(0, x) = 1$, $\mathrm{pow}(n + 1, x) = \mathrm{mult}(\mathrm{pow}(n, x), x)$ – hatványozás
- faktoriális, különbség, egyenlőség tesztelés, előjel, egészosztás, maradék, stb

A kiszámíthatóság elméletének kialakulása

Mi az, hogy valami kiszámítható?

1934, Gödel: Általános rekurzív függvények
A primitív rekurzív függvények konstrukciói, plusz:

- $f/(k+1) \Rightarrow h(\boldsymbol{x}) = \min_{n \geq 0}\{n : f(n, \boldsymbol{x}) = 0\}$.

az általános (parciális) rekurzív függvények.

Példa: Ackermann-függvény

$$A(0, n) = n + 1$$
$$A(m + 1, 0) = A(m, 1)$$
$$A(m + 1, n + 1) = A(m, A(m + 1, n))$$

pl.
$A(1, 1) = A(0, A(1, 0)) = A(1, 0) + 1 = A(0, 1) + 1 = 2 + 1 = 3.$
2 pontért jövő hétre számítsuk ki $A(5, 1)$-et.
Az Ackermann-függvény nem primitív, de általános rekurzív függvény.

A kiszámíthatóság elméletének kialakulása

Mi az, hogy valami kiszámítható?

- 1931, Gödel: Primitív rekurzív függvények
- 1934, Gödel: Általános rekurzív függvények
- 1930-as évek eleje, Church, Kleene, Rosser (Princeton): λ-definiálhatóság

1935: AMS New York-i összejövetele

Church tézise: Az általános rekurzív függvények (matematikai fogalom) megfelelnek az algoritmikusan kiszámítható függvény fogalmának (intuitív fogalom).

1936: Kleene

Az általános rekurzív függvények megegyeznek a λ-definiálható függvényekkel.

A kiszámíthatóság elméletének kialakulása

1936: Turing

Bebizonyítja, hogy a Turing-gép

- `https://www.youtube.com/watch?v=gJQTFhkhwPA`
- `https://www.youtube.com/watch?v=E3keLeMwfHY`
- `https://www.youtube.com/watch?v=FTSAiF9AHN4`

ekvivalens a λ-definiálhatósággal, és megfogalmazza azt a tézist, hogy a **Turing-gép megfelel az algoritmussal kiszámítható függvényeknek**.

A **Church-Turing tézist** tapasztalati tények és matematikai eredmények támasztják alá:

- Minden intuitív értelemben megoldható problémáról sikerült kimutatni, hogy azok megoldhatók a matematikai modellekben is.
- A matematikai modellek ekvivalensek.

A kiszámíthatóság elméletének kialakulása

Ezek után...

1936, '37: Church, Turing

Nem létezik olyan algoritmus, mely a predikátumkalkulus tetszőleges formulájáról eldöntené, hogy tautológia-e.

1971: Matijasevič

Hilbert 10. problémája algoritmikusan **megoldhatatlan**.

Kiszámíthatóság és bonyolultság

Kiszámíthatóságelmélet

Melyek az **algoritmikusan** megoldható problémák?

Bonyolultságelmélet

Melyek a **gyakorlatilag** megoldható problémák? (erőforrásigény)

1971: Cook

P és **NP** osztályok, **NP**-teljesség, SAT **NP**-teljes

1972: Karp

Rámutat az **NP**-teljes problémák változatosságára.

1973: Levin

Több kombinatorikus probléma „univerzális a kimerítő keresésre”.

További bonyolultsági osztályok és számítási módok

- **L**, **NL**, **PSPACE**, **EXP**, ...
- Valószínűségi modellek
- Párhuzamos számítás

stb.

Mit jelent az időigény?

Az alábbi táblázat megadja, hogy adott futásidejű algoritmus adott számítási kapacitású architektúrán mekkora inputra fut még le **egy napon belül**.

	 C64 1Mflops	 Cray Y-MP 1Gflops	 mai GPU 5TFlops	 Tianhe-2 34PFlops	 Föld bolygó 10EFlops
n	86.4G	8.64×10^{13}	4.32×10^{17}	2.94×10^{21}	8.64×10^{23}
$n \log n$	2.75G	2.1×10^{12}	8.1×10^{15}	4.5×10^{19}	1.17×10^{22}
n^2	300k	9.3M	657M	54G	929G
n^3	4420	44.208	756k	14.3M	95M
2^n	36	46	58	71	79
1.1^n	264	336	426	518	578
$n!$	14	16	19	22	24

Amíg Moore törvénye (nagyjából) igaz, addig a polinomidejű algoritmusok egy-egy újabb év elteltével konstans**szor** több adattal tudnak adott időn belül elbánni.

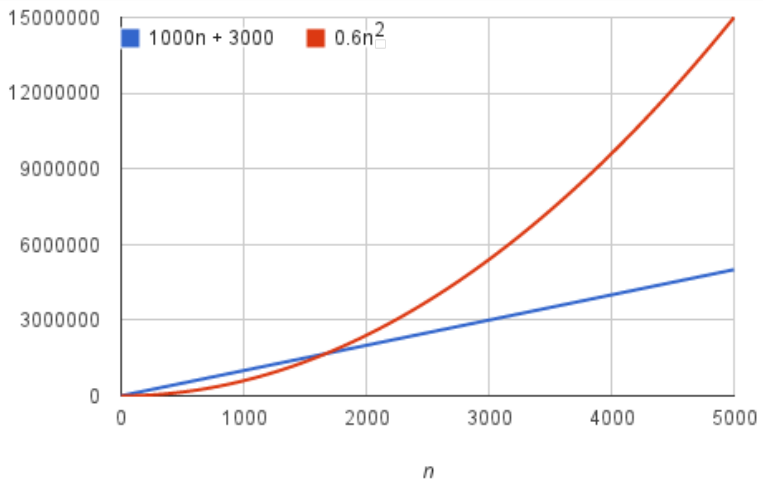
Aszimptotikus jelölések

Definíció

Legyenek $f, g : \mathbb{N} \to \mathbb{N}$ függvények.

- $\boldsymbol{f(n) = \mathcal{O}(g(n))}$, ha létezik olyan $c > 0$, hogy $f(n) \leq c \cdot g(n)$ majdnem minden n-re.
- $\boldsymbol{f(n) = \Omega(g(n))}$, ha $g(n) = \mathcal{O}(f(n))$.
- $\boldsymbol{f(n) = \Theta(g(n))}$, ha $f(n) = \mathcal{O}(g(n))$ és $g(n) = \mathcal{O}(f(n))$.

Ha $f(n) = \mathcal{O}(g(n))$, de $g(n) \neq \mathcal{O}(f(n))$, annak jele $\boldsymbol{f(n) = o(g(n))}$. (és ω hasonlóan.)



$$1000n + 3000 = \mathcal{O}(0.6n^2)$$

Aszimptotikus jelölések

Határértékkel

Ha $\lim_{n\to\infty} \frac{f(n)}{g(n)} \ldots$

- $\ldots = 0$, akkor $f(n) = o(g(n))$
- $\ldots < \infty$, akkor $f(n) = \mathcal{O}(g(n))$

Példa

Ha $p(n)$ és $q(n)$ polinomok (pozitív főegyütthatóval), akkor

- $p(n) = \mathcal{O}(q(n))$ pontosan akkor, ha p fokszáma legfeljebb akkora, mint q-é;
- $p(n) = \Theta(q(n))$ pontosan akkor, ha fokszámaik megegyeznek.

Legyen $a \in \mathbb{N}$, $a > 1$. Ekkor $p(n) = o(a^n)$.

A $\mathbf{P}$ osztály, informálisan

Definíció

Egy probléma a $\mathbf{P}$ **osztályba** esik, ha megoldható polinom időigényű (vagy $\mathcal{O}(n^k)$ időigényű) algoritmussal.

A Cobham – Edmonds tézis

- Egy algoritmus akkor ad **gyakorlatilag kielégítő** megoldást egy problémára, ha polinom időigényű.
- Egy problémát akkor tekintünk **gyakorlatilag megoldhatónak**, ha a $\mathbf{P}$ osztályba esik.

Néhány ellenvetés

- Elfogadható-e egy $\mathcal{O}(n^{100})$ időigényű algoritmus?
- A konstansok nagysága.
- Kisméretű adatokon egy exponenciális algoritmus is jobban viselkedhet, mint egy polinomiális.
- Legrosszabb eset helyett a várható viselkedés vizsgálata is indokolt lehet.
- A $\mathbf{P}$ osztályba eső egyes problémák hatékonyan párhuzamosíthatók, míg mások (valószínűleg) nem.

Ugyanakkor

- A gyakorlatban előforduló $\mathbf{P}$-beli problémák rendje kicsi. Pl. a lineáris programozás alapfeladatát megoldó első polinomidejű algoritmus, az **ellipszoid módszer** (1979, $\mathcal{O}(n^6 \cdot L)$) a **szimplex módszer** (1947) exponenciális futásidejű algoritmusánál a **gyakorlatban** lassabban futott. A szintén polinomidejű **projektív módszer** (1984, $\mathcal{O}(n^{3.5} L^2 \log L \log\log L)$) azonban már a gyakorlatban előforduló problémák esetében is gyorsabbnak bizonyult a szimplex módszernél.
- A $\mathbf{P}$ osztály elegáns matematikai elmélethez vezet.

Összefoglalás

- Megismertük Hilbert tizedik problémáját.
- Megismertük a primitív rekurzív és az általános rekurzív függvényeket.
- Megismertük az Ackermann-függvényt, ami általános rekurzív, de nem primitív rekurzív.
- Kleene tétele: általános rekurzió $=$ λ-kalkulus.
- Turing tétele: λ-kalkulus $=$ Turing-gép.
- Church-Turing tézis: algoritmus $=$ Turing-gép, általános rekurzió, λ-kalkulus.
- Church tétele: nincs algoritmus, mely input elsőrendű logikai formuláról eldöntené, hogy tautológia-e.
- Matijasevič: Hilbert tizedik problémája eldönthetetlen.
- Megismertük az O, Ω, Θ, o, ω jeleket.
- Cobham-Edmonds tézis: hatékony algoritmus $=$ polinomidejű.
- $\mathbf{P}$: a polinomidőben eldönthető problémák osztálya.