Algoritmusok és vizsgálatuk (GEMAK121M) 1. ZH elméleti fogalmak

Tartalomjegyzék

Lexikografikus rendezés: a lexikografikus rendezésben egy α szó megelőz egy β szót, ha vagy kezdőszelete (prefixe), vagy az első olyan betű, amely nem azonos a két szóban, az α szóban ABC szerint rendezve kisebb. Pl.: 0, 00, 000… 01, 010… 1, 10, 100… 11…

Növekvő rendezés: minden rövidebb szó megelőz minden hosszabb szót, az azonos hosszúságú szavak pedig lexikografikusan vannak rendezve. Pl.: 0, 1, 00, 01, 10, 11, 000, 001, 010…

$f = O(g)$: Legyen f és g két természetesen számokon értelmezett komplex értékű függvény:

$\boldsymbol{f} = \boldsymbol{O}(\boldsymbol{g})$, $ha\ \exists\ c > 0, \exists\ n_0 \in \mathbb{Z}^+$ küszöb, $hogy\ \forall n > n_0$: $|f(n)| \leq c \cdot |g(n)|$

$\boldsymbol{f} = \boldsymbol{o}(\boldsymbol{g})$, ha g(n) csak véges sok helyen nulla, és f(n)/g(n) → 0, ha n → ∞.

$\boldsymbol{f} = \boldsymbol{\Omega}(\boldsymbol{g})$, $ha\ g = O(f)$

$\boldsymbol{f} = \boldsymbol{\Theta}(\boldsymbol{g})$, $ha\ f = O(g)$ és $f = \Omega(g)$, vagyis $\exists\ c_1, c_2 > 0$ konstansok és $\exists\ n_0 \in \mathbb{N}$ küszöb, hogy $c_1 \cdot |g(n)| \leq |f(n)| \leq c_2 \cdot |g(n)|$.

Turing-gép: Matematikailag a Turing-gépet az alábbi adatok írják le: $T = (k, \Sigma, \Gamma, \alpha, \beta, \gamma)$, ahol $k \geq 1$, $k \in \mathbb{N}$, Σ és Γ véges halmazok, $* \in \Sigma$, $START, STOP \in \Gamma$, α, β, γ tetszőleges leképezések:

- α: $\Gamma \times \Sigma^k \rightarrow \Gamma$ megadja az új állapotot,
- β: $\Gamma \times \Sigma^k \rightarrow \Sigma^k$ megadja a szalagokra írt jeleket,
- γ: $\Gamma \times \Sigma^k \rightarrow \{-1, 0, 1\}^k$ megadja, hogy mennyit lép a fej.

T a p programmal szimulálja S-et: Azt mondjuk, hogy $T = (k+1, \Sigma, \Gamma_T, \alpha_T, \beta_T, \gamma_T)$ a $p \in \sum_0^*$ programmal szimulálja az $S = (k, \Sigma, \Gamma_S, \alpha_S, \beta_S, \gamma_S)$-t, ha tetszőleges $x_1, x_2, \ldots, x_k \in \sum_0^*$ szavakra T az $(x_1, x_2, \ldots, x_k, p)$ bemeneten akkor és csak akkor áll meg véges lépésben, ha S az $(x_1, x_2, \ldots, x_k)$ bemeneten megáll és megálláskor T első k szalagján rendre ugyanaz áll, mint S szalagjain.

Univerzális Turing-gép: Azt mondjuk, hogy a k+1 szalagos T Turing-gép univerzális, ha bármely k szalagos Σ fölötti S Turing-géphez létezik olyan p szó (program), mellyel a T szimulálja S-et.

Boole-függvény: Boole-függvénynek nevezünk egy f: $\{0, 1\}^n \rightarrow \{0, 1\}$ leképezést.

Boole-polinom: A konjunkció, a diszjunkció és a negáció műveleteivel felírt kifejezéseket Boole-polinomoknak nevezzük.

Diszjunktív normálforma: az olyan Boole-polinom, melynek diszjunkció művelettel összekapcsolt elemei konjunkciókból áll.

Logikai hálózat: Legyen G egy aciklikus irányított gráf. A gráf forrásait bemeneti csúcsoknak, nyelőit kimeneti csúcsoknak nevezzük. A gráf minden olyan v csúcsához, mely nem forrás ($d = d_+(v) > 0$), adjunk meg egy kaput (F_v: $\{0, 1\}^d \rightarrow \{0, 1\}$ $Boole-függvényt$). Az ilyen függvényekkel előállított irányított gráfot logikai hálózatnak nevezzük.

A logikai hálózat mérete a kapuk száma, a mélysége pedig egy bemeneti csúcstól egy kimeneti csúcsig vezető út maximális hossza.

Rekurzív függvény: Egy f: $\sum_0^* \rightarrow \sum_0^*$ függvényt kiszámíthatónak vagy rekurzívnak nevezünk, ha van olyan T Turing-gép (tetszőleges k számú szalaggal), amely bármely $x \in \sum_0^*$ bemenettel, véges idő után megáll, és az utolsó szalagjára az $f(x)$ szó lesz írva.

Rekurzív nyelv: Legyen $\mathcal{L} \subseteq \sum_0^*$ egy nyelv. Az $\mathcal{L}$ nyelvet rekurzívnak nevezzük, ha karakterisztikus függvénye: $f(x) = \begin{cases} 1, ha\ x \in \mathcal{L} \\ 0, ha\ x \in \sum_0^* - \mathcal{L} \end{cases}$ kiszámítható.

Rekurzíve felsorolható nyelv: Az $\mathcal{L}$ nyelvet rekurzíve felsorolhatónak nevezzük, ha vagy $\mathcal{L}=\emptyset$, vagy van olyan kiszámítható f: $\sum_0^* \to \sum_0^*$ függvény, amelynek értékkészlete $\mathcal{L}$.

Turing-gép leírása: Egy Turing-gép leírásának nevezzük a Σ és Γ halmazok felsorolását és az α,β,γ függvények táblázatát.

Megállási feladat: algoritmikusan nem lehet eldönteni, hogy egy univerzális Turing gép egy adott bemenettel véges időn belül leáll-e.

Nyelvek triviális tulajdonsága: Nyelvek egy tulajdonságát triviálisnak nevezzük, ha vagy minden $\mathcal{L}_T$ típusú (T tetszőleges Turing-gép) nyelvnek megvan, vagy egyiknek sem.

Formális rendszer: Egy F formális rendszer vagy más néven elmélet egy algoritmus, mely eldönti egy (P, T) párról, hogy P helyes bizonyítása-e T-nek.

Konzisztens elmélet: Egy elméletet konzisztensnek nevezünk, ha nincs olyan mondat, hogy ő is és a negáltja is tétel.

Teljes konzisztens elmélet: Egy S mondatot T elmélettől függetlennek hívunk, ha sem S, sem negáltja nem tétel T-ben. Egy konzisztens elmélet teljes, ha nincsen tőle független mondat.

Turing-gép időigénye: Egy T Turing-gép időigénye az a $time_T(n)$ függvény, amely a gép lépésszámának maximumát adja meg n hosszúságú bemenet esetén.

Turing-gép tárigénye: A $space_T(n)$ tárigény-függvényt úgy definiáljuk, mint a gép szalagjain azon különböző mezők maximális számát az n hosszúságú bemenetek esetén, melyekre a gép ír (a bemenet által elfoglalt mezőket nem számítjuk a tárba).

Turing-gép polinomiális: Azt mondjuk, hogy a T Turing-gép polinomiális, ha időigénye $O(f)$ valamely f polinomra, vagyis van olyan c>0 konstans, hogy T időigénye $O(n^c)$.

DTIME(f(n)): Azt mondjuk, hogy egy $\mathcal{L}\subseteq\sum_0^*$ nyelv *időbonyolultsága* legfeljebb $f(n)$, ha a nyelv egy legfeljebb $f(n)$ *időigényű* Turing-géppel eldönthető. A legfeljebb $f(n)$ *időbonyolultságú* nyelvek osztályát $DTIME(f(n))$-nel jelöljük.

DSPACE(f(n)): Azt mondjuk, hogy egy $\mathcal{L}\subseteq\sum_0^*$ nyelv *tárbonyolultsága* legfeljebb $f(n)$, ha a nyelv egy legfeljebb $f(n)$ *tárigényű* Turing-géppel eldönthető. A legfeljebb $f(n)$ *tárbonyolultságú* nyelvek osztályát $DSPACE(f(n))$-nel jelöljük.

PTIME (P): $PTIME$-mal vagy egyszerűen P-vel jelöljük mindazon nyelvek osztályát, melyek polinomiális Turing-géppel eldönthetők.

Teljesen időkonstruálható függvény: Egy f: $\mathbb{Z}_+\to\mathbb{Z}_+$ függvényt teljesen időkonstruálhatónak nevezünk, ha van olyan T Turing-gép, mely minden n hosszú bemeneten pontosan $f(n)$ lépést végez.

Jól számolható függvény: Egy f: $\mathbb{Z}_+\to\mathbb{Z}_+$ függvényt jól számolhatónak nevezünk, ha van olyan T Turing-gép, mely az $f(n)$-et az n bemeneten $O(f(n))$ idő alatt kiszámítja.

Tételek:

Kapcsolat a RAM és a Turing-gép között: Minden {0,1,2} fölötti Turing-géphez konstruálható olyan program a RAM-on, mely minden bemenetre ugyanazt a kimenetet számítja ki, mint a Turing-gép, és ha a Turing-gép lépésszáma N, akkor a RAM $O(N)$ lépést végez $O(\log N)$ jegyű számokkal.

Minden RAM programhoz van olyan Turing-gép, mely minden bemenetre ugyanazt a kimenetet számítja ki, mint a RAM, és ha a RAM futási ideje N, akkor a Turing-gép lépésszáma $O(N^2)$.

Kapcsolat a Turing-gépek és a Boole-hálózatok között: Minden Σ={0,1,∗} feletti T Turing-géphez és minden $1 \leq n \leq N$ számpárhoz van olyan n bemenetű, $O(N^2)$ méretű, $O(N)$ mélységű, legfeljebb 2 befokú Boole-hálózat, mely egy $(x_0,x_1,\ldots,x_{n-1}) \in \{0,1\}^n$ bemenetre akkor és csak akkor számol ki 1-et, ha az $x_0,x_1,\ldots,x_{n-1}$ bemenetre a T Turing-gép N lépés után az utolsó szalag 0. mezején 1 áll.

Church-tézis: Minden számítás az általa megadott rendszerben formalizálható.

A rekurzív és rekurzíve felsorolható nyelvek kapcsolata: Minden rekurzív nyelv rekurzíve felsorolható. Egy $\mathcal{L}$ nyelv akkor és csak akkor rekurzív, ha mind az $\mathcal{L}$ nyelv, mind a $\sum_0^* - \mathcal{L}$ nyelv rekurzíve felsorolható.

Rice tétele: Bármely nem-triviális nyelv-tulajdonságra algoritmikusan eldönthetetlen, hogy egy adott $\mathcal{L}_T$ nyelvnek megvan-e.

Algoritmikusan eldönthetetlen problémák: dominó-probléma; Diophantoszi-egyenlet; csoportok szóproblémája; poliéderek összehúzhatósága; Post szóproblémája.

Gödel nem-teljességi tétele: Minden minimálisan megfelelő elmélet nem-teljes.

Gödel teljességi tétele: Legyen P az összes olyan (B,T) pár halmaza, hogy B véges sok mondat és a T mondat minden olyan interpretációban igaz, melyben a B-beli mondatok igazak. Ekkor P rekurzív felsorolható.

Polinomiális idejű kombinatorikai algoritmusok: összefüggőségi teszt; legrövidebb út keresése; magyar módszer; maximális folyam keresése; Edmonds párosítás algoritmusa.

Polinomiális idejű aritmetikai algoritmusok: egész számok összeadása, kivonása, szorzása, maradékos osztása; két szám nagyság szerinti összehasonlítása; Euklideszi algoritmus; moduláris hatványozás.

Az euklideszi algoritmus polinomiális idejű (lemma): Az euklideszi algoritmus polinomiális idejű, pontosabban $O(\log a+\log b)$ aritmetikai műveletből áll, melyek a,b-nél nem nagyobb természetes számokon kell végezni.

A moduláris hatványozás polinomiális idejű (lemma): Legyen a,b és m három természetes szám. Ekkor $a^b(modulo\ m)$ kiszámítható polinomiális időben, pontosabban $O(\log b)$ aritmetikai művelettel, melyeket $O(\log m+\log a)$ jegyű természetes számokon végzünk.

Polinomiális idejű lineáris algebrai algoritmusok: vektorok összeadása, skaláris szorzása; mátrixok szorzása, invertálása; determinánsok kiszámítása; Gauss-elimináció.

Lineáris gyorsítási tétel: Minden T Turing-géphez és $c>0$ konstanshoz található olyan S Turing-gép, mely ugyanazt a nyelvet dönti el, és melyre $time_S(n) \leq c \cdot time_T(n)+n$.

Idő-hierarchia tétel: Ha $f(n)$ teljesen időkonstruálható és $g(n)\cdot\log g(n)=o(f(n))$, akkor van olyan nyelv $DTIME(f(n))$-ben mely nem tartozik $DTIME(g(n))$-be.

Hézag tétel: Minden rekurzív $\Phi(n) \geq n$ függvényhez van olyan rekurzív $f(n)$ függvény, hogy $DTIME(\Phi(f(n)))=DTIME(f(n))$, így például olyan is, amire: $DTIME(f(n))=DSPACE(f(n))$.

Gyorsítási tétel: Bármely rekurzív $g(n)$ függvényhez létezik olyan rekurzív $\mathcal{L}$ nyelv, hogy minden $\mathcal{L}$-et eldöntő T Turing-géphez létezik olyan $\mathcal{L}$-et eldöntő S Turing-gép melyre: $g(time_S(n))<time_T(n)$.