Lexikografikus rendezés

A lexikografikus rendezésben egy α szó megeloz egy β szót, ha vagy kezdőszelete (prefixe), vagy az első olyan betű, amely nem azonos a két szóban, az α szóban kisebb (az ábécé rendezése szerint).

Növekvő rendezés

Minden rövidebb szó megeloz minden hosszabb szót, az azonos hosszúságú szavak pedig lexikografikusan vannak rendezve.

$f = O(g)$

Legyen f és g két, természetes számokon értelmezett, komplex értéku függvény. Azt írjuk, hogy

$$f = O(g),$$

ha van olyan $c > 0$ konstans és olyan $n_0 \in Z_+$ küszöb, hogy minden $n > n_0$ esetén $\mid f(n) \mid \cdot\, c \mid g(n) \mid$.

Turing-gép

Az alábbi hatossal írható le $T = \langle k, \Sigma, \Gamma, \alpha, \beta, \gamma \rangle$:

- k : szalagok száma ($k \in \mathbb{N}, k \geq 1$)
- Σ ábécé ($* \in \Sigma$)
- Γ állapotok halmaza
- $\alpha : \Gamma \times \Sigma^k \to \Gamma$, új állapot (START, STOP $\in \Gamma$)
- $\beta : \Gamma \times \Sigma^k \to \Sigma^k$, szalagra írt jelek
- $\gamma : \Gamma \times \Sigma^k \to \{-1, 0, +1\}^k$, a fejek mozgását jelöli (-1 $\to$ balra, 0 $\to$ nem mozdul, $+1 \to$ jobbra)

T a p programmal szimulálja S-et

Legyen $T = \langle k+1, \Sigma, \Gamma_T, \alpha_T, \beta_T, \gamma_T \rangle$ és $S = \langle k, \Sigma, \Gamma_S, \alpha_S, \beta_S, \gamma_S \rangle$ kér Turing-gép ($k \geq 1$). Legyen $p \in \Sigma_0^*$. Azt mondjuk, hogy T a p programmal szimulálja S-et, ha tetszőleges $x_1, \ldots, x_k \in \Sigma_0^*$ szavakra T az $(x_1, \ldots, x_k, p)$ bemeneten akkor és csak akkor áll meg véges számú lépésben, ha S az $(x_1, \ldots, x_k)$ bemeneten megáll, és megálláskor T első k szalagján rendre ugyanaz áll, mint S szalagjain.

Univerzális Turing-gép

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

Boole-függvény

Boole-függvénynek nevezzük az $f : \{0,1\}^n \to \{0,1\}$ leképezést.

Boole-polinom

A konjunkció, diszjunkció és negáció műveleteivel fölírt kifejezéseket Boole-polinomoknak nevezzük.

Diszjunktív normálforma

Diszjunktív normálformának nevezzük az olyan Boole-polinomot, mely $\vee$ művelettel összekapcsolt elemi konjunkciókból áll.

Logikai hálózat

A gráf minden olyan v csúcsához, mely nem forrás, tehát melynek be-foka valamely $d = d_+(v) > 0$, adjunk meg egy „kaput”, vagyis egy $F_v : \{0,1\}^d \to \{0,1\}$ Boole-függvényt. (A függvény változói feleljenek meg a v-be befutó éleknek.) Az ilyen függvényekkel ellátott irányított gráfot logikai hálózatnak nevezzük.

Kiszámítható vagy rekurzív függvény

Egy $f : \sum_0^* \to \sum_0^*$ függvényt kiszámíthatónak vagy rekurzívnak nevezünk, ha van olyan T Turing-gép (tetszőleges k számú szalaggal), mely bármely $x \in \sum_0^*$ bemenettel (vagyis első szalagjára az x szót, a többire az üres szót írva), 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 hívjuk, ha karakterisztikus függvénye:

$$f(x) = \begin{cases} 1 & \text{, ha } x \in \mathcal{L} \\ 0 & \text{, ha } x \in \sum_0^* - \mathcal{L} \end{cases}$$

kiszámítható.

Rekurzíve felsorolható nyelv

Az $\mathcal{L}$ nyelvet rekurzíve fölsorolhatónak nevezzük, ha vagy $\mathcal{L} = \emptyset$, vagy van olyan kiszámítható $f : \Sigma_0^* \to \Sigma_0^*$ függvény, melynek értékkészlete $\mathcal{L}$.

Turing-gép leírása

Egy Turing-gép leírásának nevezzük a Γ és Σ halmazok felsorolását (ahol, mint eddig, Γ elemeit Σ fölötti szavak kódolják) és az α, β, γ függvények táblázatát.

Megállási feladat

Algoritmikusan nem lehet eldönteni, hogy egy univerzális Turing gép egy adott bemenettel véges idon belül leáll-e.

Nyelvek triviális tulajdonsága

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

F formális rendszer (elmélet)

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 hívunk, ha nincs olyan mondat, hogy ő is és a negáltja is tétel.

Teljes konzisztens elmélet

Egy konzisztens elmélet teljes, ha nincsen tole független mondat.

T Turing-gép időigénye

Egy T Turing-gép időigénye az a $\text{time}_T(n)$ függvény, mely a gép lépésszámának maximumát adja meg n hosszúságú bemenet esetén. Föltesszük, hogy $\text{time}_T(n) \geq n$.

T Turing-gép tárigénye

A $\text{space}_T(n)$ tárigény-függvényt úgy definiáljuk, mint a gép szalagjain azon különböző mezők maximális számát az n hosszúságú bemenetek esetén, melyekre a gép ír. Nyilván $\text{space}_T(n) \geq 1$.

T 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 $\mathcal{L} \subseteq \Sigma_0^*$ nyelv idobőnyolultsága legfeljebb $f(n)$, ha a nyelv egy legfeljebb $f(n)$ idoigényű Turing-géppel eldönthető. A legfeljebb $f(n)$ időbonyolultságú nyelvek osztályát DTIME($f(n)$)-nel jelöljük.

PTIME

Mindazon nyelvek osztályát, melyek polinomiális Turing-géppel eldönthetok, PTIME-mal vagy egyszeruen P-vel jelöljük.

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

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

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 a 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 T $\Sigma = \{0, 1, *\}$ feletti Turing-géphez és minden $N \geq n \geq 1$ számpárhoz van olyan n bemenetű, $O(N^2)$ méretű, $O(N)$ mélységű, legfeljebb 2 befokú Boole-hálózat, mely egy $(x_0, \ldots, x_{n-1}) \in \{0, 1\}^n$ bemenetre akkor és csak akkor számol ki 1-et, ha az $x_0 \ldots x_{n-1}$ bemenetre a T Turing gép N lépése után az utolsó szalag 0-ik mezején 1 áll.

Church-tézis

Minden „számítás” az általa megadott rendszerben formalizálható.

A rekurzív és a 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 fölsorolható.

Rice-tétel

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 szóprobléma

Gödel nem-teljességi tétele

Minden minimálisan megfelelő elmélet nem-teljes.

Gödel teljességi tétele

Legyen $\mathcal{P}$ az összes olyan $(\mathcal{B}, T)$ pár halmaza, hogy $\mathcal{B}$ véges sok mondat és a T mondat minden olyan interpretációban igaz, melyben a $\mathcal{B}$-beli mondatok igazak. Ekkor $\mathcal{P}$ rekurzív felsorolható.

Polinomiális idejű kombinatorikai algoritmusok

- összefüggőség-teszt
- legrövidebb út keresése
- maximális folyam keresése (Edmonds–Karp vagy Dinic–Karzanov módszerrel)
- „magyar módszer”
- Edmonds párosítás algoritmusa

Polinomiális idejű aritmetikai algoritmusok

- egész számok összeadása, kivonása, szorzása, maradékos osztása
- két szám nagyság szerinti összehasonlítása
- Euklideszi algoritmus két természetes szám legnagyobb közös osztójának megkeresésére

Az Euklideszi algoritmus polinomiális idejű

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

A moduláris hatványozás polinomiális idejű

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 szorozása, invertálása,
- determinánsok kiszámítása

Lineáris gyorsítási tétel

Minden T Turing-géphez és $c > 0$-hoz található olyan S Turing-gép, mely ugyanazt a nyelvet dönti el, és melyre $\text{time}_S(n) \leq c \cdot \text{time}_T(n) + n$.

Idő-hierarchia tétel

Ha $f(n)$ teljesen időkonstruálható és $g(n)(\log g(n)) = o(f(n))$, 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($f(n)$).

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 $g(\text{time}_S(n)) < \text{time}_T(n)$.