Church-tézis

Algoritmus

Az algoritmus fogalmát nem definiáljuk, hanem intuitív fogalomnak tekintjük. Az algoritmus olyan matematikai eljárást jelent, mely valamely számítás, vagy konstrukció elvégzésére szolgál, s melyet gondolkodás nélkül, gépiesen lehet végrehajtani.

Church-tézis.

Algoritmikusan kiszámíthatóak azok az egész értékű függvények, amelyeknek az értékeit valamilyen algoritmussal elő tudjuk állítani.

Turing-kiszámíthatónak nevezünk egy $f:\mathbf{N}^m \rightarrow \mathbf{N}$ függvényt, ha van olyan Turing-gép, amelyik $x_1,\ldots, x_m$ bemenet esetén, ha $f(x_1,\ldots, x_m)$ létezik, akkor kiszámítja ezt az értéket, és leáll.

Church-tézis.

Az algoritmikusan kiszámítható függvények osztálya megegyezik a Turing-kiszámítható függvények osztályával.

Nyelvek

Nyelvek

Definíció.

Az $A=\{a_1,\ldots, a_n\}$ véges, nemüres halmazt *ábécé*-nek nevezzük, elemei a *betűk*, a belőlük képezhető véges hosszú sorozatok a *szavak*. Az összes véges hosszú sorozat halmazát A^* jelöli. Egy szóban előforduló betűk száma a *szó hossza*.

Definíció.

Legyen A véges ábécé. Az $L \subseteq A^*$ halmazt *nyelvnek* nevezzük.

Adjuk meg az A elemeinek valamilyen sorrendjét. Két azonos hosszú szót *lexikografikusan* rendezünk:

sorban összehasonlítva a betűket, az első eltérő betűt keressük, és az a szó előzi meg a másikat, amelyikben a korábbi betű szerepel.

Az L nyelv szavait *növekvően* rendezzük:

a rövidebb szó megelőzi a hosszabbat, az egyforma hosszúak között pedig lexikografikusan rendezünk.

Rekurzíve felsorolható nyelv, rekurzív nyelv

Definíció.

Rekurzíve felsorolható egy L nyelv, ha van olyan Turing-gép, amelyik L szavait felismeri.

Rekurzív egy L nyelv, ha van olyan Turing-gép, amelyik L szavait felismeri, és minden bemeneti szó esetén véges sok lépésben megáll.

Algoritmikusan megoldhatatlan problémák

A matematikában gyakran merülnek fel olyan kérdések, amelyeknél azt kell eldönteni, hogy egy osztály valamely eleme rendelkezik-e egy tulajdonsággal, vagy sem. Például:

- két természetes szám relatív prím-e,
- egy Turing-gép mindig megáll-e véges sok lépés után, stb.

Algoritmikus probléma: általános módszert kell megadnunk, amellyel minden konkrét esetben meg tudjuk válaszolni az adott típusú kérdést.

Az ilyen kérdések arra vezethetők vissza, hogy egy adott ábécéből alkotott szó benne van-e egy adott nyelvben, vagy sem.

Egy algoritmikus problémát megoldhatónak nevezünk, ha a hozzá tartozó formális nyelv rekurzív, és megoldhatatlannak, ha nem rekurzív. Tehát akkor mondunk egy ilyen problémát megoldhatatlannak, ha nem létezik olyan Turing-gép, amely bármely konkrét eset kódjáról véges sok lépésben eldönti, hogy az adott rögzített nyelvben benne van-e, vagy sem.

Tétel. Rekurzív nyelv komplementere is rekurzív.

Tétel. Ha L és L komplementere is rekurzívan felsorolható, akkor L rekurzív.

Tétel. Van olyan rekurzívan felsorolható nyelv, amelyik nem rekurzív.

Tétel. Van olyan formális nyelv, amelyik nem rekurzíven felsorolható.

Rekurzívan felsorolható nyelvek egy tulajdonsága *triviális*, ha vagy minden rekurzívan felsorolható nyelv rendelkezik ezzel a tulajdonsággal, vagy egyikük sem.

Tétel. Legyen adva a rekurzívan felsorolható nyelveknek valamilyen nem triviális tulajdonsága. Annak eldöntése, hogy egy nyelv rendelkezik-e az adott tulajdonsággal, algoritmikusan megoldhatatlan probléma.

Az utóbbi 5 tétel **Bizonyítása** megtalálható [3]-ban.

Annak felismerése, hogy egy rekurzívan felsorolható nyelv mikor üres, véges, rekurzív, stb., *algoritmikusan megoldhatatlan problémát* jelent.

Algoritmusok hatékonysága

Ordó, ordó

Legyen f és g két, természetes számokon értelmezett, valós értékű függvény. Nézzük a következő jelöléseket:

- $f=O(g)$, (f= nagy ordó g) ha van olyan $c>0$ konstans, hogy minden elég nagy n-re $|f(n)|\leq c|g(n)|$,
- $f=o(g)$, (f= kis ordó g) ha $g(n)$ csak véges sok helyen nulla, és $f(n)/g(n)\to 0$, ha $n\to\infty$.
- $f=\Omega(g)$, ha $g=O(f)$. (Ritkán használatos.)
- $f=\Theta(g)$, ha $f=O(g)$ és $f=\Omega(g)$, vagyis vannak olyan c_1, c_2 konstansok, hogy minden elég nagy n-re

$$c_1 g(n) \leq f(n) \leq c_2 g(n).$$

Az egyenlőség nem szimmetrikus. Pl. $O(n) = O(n^2)$, de $O(n^2) \neq O(n)$

Turing-gépek bonyolultságának mértékei

Jelölje t (ω) az M Turing-gép lépéseinek számát ω bemenet esetén, ha a számítás véges sok lépésben véget ér. Ha nem ér véget véges sok lépés után, akkor t ezen a helyen nincs értelmezve.

h(ω) az ω szó hossza.

Az M gép *időbonyolultsága*: t (n)= max(t (ω), ha h(ω)=n)

Jelölje l (ω) az M Turing-gép esetén azoknak a mezőknek a számát ω bemenet esetén, amelyeket M legalább egyszer leolvas, ha a számítás véges sok lépésben véget ér. Ha nem ér véget véges sok lépés után, akkor l ezen a helyen nincs értelmezve. (Mennyi külső memóriára van M-nek szüksége egy adott számítási folyamat során.)

Az M gép *szalagbonyolultsága*: l (n) = max(l (ω), ha h(ω)=n)

Gyakorlati kiszámíthatóság

Davidon észrevétele nyomán tételezzünk fel egy ultrasebességű szekvenciális gépet, amely annyi idő alatt végez el egy lépést, amíg a fény atomsugárnyi utat megtesz ($\sim 10^{-23}$ sec).Ha egy ilyen gép a világegyetem kezdete óta működne (kb. $1{,}4 \times 10^{10}$ éve), akkor még nem tett volna meg 2^{2^8} lépést, tehát még nem tudta volna megvizsgálni egy 2^8 elemű halmaz összes részhalmazát. (Lásd [1]).

Recski András továbbgondolta ezt a példát. Képzeljük el, hogy a gépnek annyi processzora van, ahány atom a világegyetem általunk ismert részén van ($\sim 2^{300}$). Ez a gép a Nagy Bumm óta a mai napig még nem végzett volna el 70! műveletet.

Vegyünk egy 70 pontú gráfot, amelyikben nincs Hamilton-kör. Ha nem ismerjük ezt a tényt, és az összes lehetőség megvizsgálásával keressük a választ, ez a szupergép a Nagy Bumm óta még nem tudott volna nekünk válaszolni.

Irodalomjegyzék

- [1] G. Birkhoff-T. C. Bartee: *A modern algebra a számítógép-tudományban* Műszaki Könyvkiadó, Budapest, 1974
- [2] Gonda János: *Bevezető fejezetek a matematikába III.* ELTE TTK, Bp. 1998
- [3] Demetrovics, Denev, Pavlov: *A számítástudomány matematikai alapjai* Tankönyvkiadó, Budapest, 1985
- [4] Dringó László-Kátai Imre: *Bevezetés a matematikába* Tankönyvkiadó, Budapest 1982
- [5] Járai Antal: *Bevezetés a matematikába* Eötvös Kiadó, Budapest, 2005
- [6] Lovász László: *Algoritmusok bonyolultsága* Tankönyvkiadó, Budapest 1991
- [7] B. A. Trahtenbrot: *Algoritmusok és absztrakt automaták* Műszaki Könyvkiadó, Budapest, 1978