

Számításelmélet szigorlat

Digitális számításelmélet tárgy

Kidolgozott tételek

Készítette:
Srancsik Bálint
2006 április

Megj.: Az anyag a Pázmány Péter Katolikus Egyetem Információs Technológiai Karának Digitális számításelmélet tantárgyának szigorlati tételeinek kidolgozását tartalmazza. Az anyag teljességért és helyességért felelőséget nem vállalok, ebből kifolyólag mindenki saját felelősségre használja. (Ám hozzáteszem, pusztán ebből az anyagból felkészülve sikeres vizsgát tettem.) Az anyag nem áll szerzői jogvédelem alatt, így szabadon terjeszthető.

1. Formális nyelv, megadási módjai. Generatív grammatika, levezetés, generált nyelv fogalma. Grammatikák ekvivalenciája. Chomsky-féle nyelvosztályok.

Formális nyelv:

Def: Σ tetszőleges jelek egy nemüres, véges halmaza $\rightarrow abc$

Σ^* az összes Σ -beli jelekből álló véges sorozatok halmaza $\rightarrow$ szavak

$$\Sigma^* = \{w = a_1 \dots a_n \mid a_i \in \Sigma, i = 1, \dots, n, n \geq 0\}$$

$w = a_1 \dots a_n \rightarrow n$ a szó hossza

ϵ : 0 hosszú sorozat $\rightarrow$ üres szó

I : 0 és 1 jeleket tartalmazza $\rightarrow$ bináris abc

L egy formális nyelv Σ felett, ha $L \subseteq \Sigma^*$

Megadási módok:

- elemek felsorolása (ha azok száma véges)
- $L = \{w \in \Sigma^* \mid w \text{ P tulajdonságú}\}$
- $L = L(M)$ ahol M egy automata
- Operátorokkal
- Generáló nyelvtanokkal

Generatív grammatika

Def: $G = (V, \Sigma, R, S)$ egy generatív grammatika, ahol

V : egy véges halmaz $\rightarrow$ nemterminálisok (gram. változók)

Σ : egy véges halmaz, diszjunkt V -vel $\rightarrow$ terminálisok

R : véges halmaz $\rightarrow$ levezetési szabályok

$\alpha \rightarrow \beta$ α és β terminálisok és nemterminálisok sorozata

$S \in V$: kezdőszimbólum (mondatszimbólum)

Jelölés: $\kappa \Rightarrow_G \mu$ vagy $\kappa \Rightarrow \mu$

Levezetés:

Egy $G = (V, \Sigma, R, S)$ grammatika alapján a $\kappa \in (V \cup \Sigma)^*$ szóból közvetlenül levezethető $\mu \in (V \cup \Sigma)^*$ szó, ha léteznek olyan $\gamma, \delta \in (V \cup \Sigma)^*$ szavak és $\alpha \rightarrow \beta$ szabály R -ben, hogy $\kappa = \gamma\alpha\delta, \mu = \gamma\beta\delta$

Def: Egy $w_1, \dots, w_n (V \cup \Sigma)^*$ -beli szavak sorozatára azt mondjuk, hogy ez egy levezetése w_n -nek w_1 -ből a $G = (V, \Sigma, R, S)$ alapján.

$$W_1 \Rightarrow_G W_2 \Rightarrow_G \dots \Rightarrow_G W_n$$

Def: u és $v (V \cup \Sigma)^*$ -beli szavak esetén v levezethető u -ból G alapján,

ha $u=v$

vagy létezik $n>0$, $u=w_1, \dots, w_n=v$ minden $(V \cup \Sigma)^*$ -beli szóra

úgy hogy: $w_1 \Rightarrow_G W_2 \Rightarrow_G \dots \Rightarrow_G W_n$

Jelölés: $u \Rightarrow_G^* v$ vagy $u \Rightarrow^* v$

Generált nyelv:

Def: A G grammatika generálja az $L \subseteq \Sigma^*$ nyelvet, ha $L = \{w \in \Sigma^* \mid S \Rightarrow^* w\}$

$L(G)$ a G grammatika által generált nyelv

Grammatikák ekvivalenciája:

Def: G és G' grammatikák ekvivalensek, ha $L(G) = L(G')$

Chomsky-féle nyelvosztályok

A nyelvet generáló grammatika szabályai alapján:

Reguláris nyelv (R) 3-as nyo.: $A \rightarrow a$, $A \rightarrow aB$ szabályok

Környezetfüggetlen (CF) 2-es nyo.: $A \rightarrow \alpha$

Környezetfüggő (CS) 1-es nyo.: $\kappa = \beta A \gamma \rightarrow \beta \alpha \gamma$

$\alpha \rightarrow \beta$, ahol $|\alpha| \leq |\beta|$

0-s nyo.: tetszőleges alakú szabályok!

2. Reguláris kifejezések, reguláris nyelvek. Determinisztikus és nemdeterminisztikus véges automata fogalma, ekvivalenciájuk. Reguláris nyelvek felismerése automatával. Egy reguláris nyelvet felismerő minimális dva.

Reguláris kifejezések:

Definíció: Egy Σ abc feletti reguláris kifejezések:

1. Σ minden eleme.
2. ε
3. $\emptyset$
4. Ha α és β reguláris kifejezések, akkor $(\alpha\beta)$ is az.
5. Ha α és β reguláris kifejezések, akkor $(\alpha\cup\beta)$ is az.
6. Ha α reguláris kifejezés, akkor α^* is az.

Minden reguláris kifejezés reprezentál egy nyelvet.

Reguláris nyelvek:

Azon nyelvek halmaza, melyek felismerhetők determinisztikus véges automatával (dva)

L reguláris nyelv zárt a reguláris operátorokra nézve.

Azaz ha A_1 és $A_2 \in \alpha$, akkor:

1. $A_1 \cup A_2 \in \alpha$.
2. $A_1 \circ A_2 \in \alpha$.
3. $A_1^* \in \alpha$.

Determinisztikus véges automata (dva vagy va)

$M = (Q, \Sigma, \delta, q_0, F)$ ahol:

1. Q , egy véges halmaz $\rightarrow$ az állapotok halmaza
2. Σ , egy véges halmaz $\rightarrow$ az abc
3. $\delta: Q \times \Sigma \rightarrow Q$ az állapotátmenet függvény
4. $q_0 \in Q$ a kezdeti állapot
5. $F \subseteq Q$ a végállapotok halmaza

Definíció: Felismert (elfogadott) jelsorozat

$M = (Q, \Sigma, \delta, q_0, F)$ egy dva, $w = x_1 x_2 \dots x_n$ Σ -beli jelek sorozata.

M felismeri w -t, ha létezik $r_0, r_1 \dots r_n$ Q -beli állapotok sorozata, hogy:

1. $r_0 = q_0$
2. $\delta(r_i, x_{i+1}) = r_{i+1}, i = 1, \dots, n-1$
3. $r_n \in F$

Definíció: M dva felismeri az A nyelvet, ha $A = \{w \mid M \text{ felismeri } w\}$. Jelölés: $A = L(M)$

Nemdeterminisztikus véges automata (nva)

$N = (Q, \Sigma, \delta, q_0, F)$ ahol:

1. Q, egy véges halmaz $\rightarrow$ az állapotok halmaza
2. Σ , egy véges halmaz $\rightarrow$ az abc
3. $\delta: Q \times (\Sigma \cup \{\epsilon\}) \rightarrow P(Q)$ az állapotátmenet függvény
4. $q_0 \in Q$ a kezdeti állapot
5. $F \subseteq Q$ a végállapotok halmaza

Definíció: Felismert (elfogadott) jelsorozat

$N = (Q, \Sigma, \delta, q_0, F)$ egy nva, $w = w_1, w_2 \dots w_n \Sigma^*$ -beli jelek sorozata.

N felismeri w-t, ha létezik $r_0, r_1 \dots r_n$ Q-beli állapotok sorozata, hogy:

1. $r_0 = q_0$
2. $r_{i+1} \in \delta(r_i, w_{i+1}) i = 0, \dots, n-1$
3. $r_n \in F$

Definíció: N nva felismeri az A nyelvet, ha $A = \{w \mid N \text{ felismeri } w\}$. $A = L(N)$

Ekvivalencia

Definíció: M_1 és M_2 ekvivalensek, ha $L(M_1) = L(M_2)$.

Tétel: Minden $N = (Q, \Sigma, \delta, q_0, F)$ nva-hoz létezik vele ekvivalens M dva.

3. Reguláris és CF nyelvek jellemzése a megfelelő pumpáló lemmával.

Pumpáló lemma: Reguláris nyelv

Ha $L \subseteq \Sigma^*$ egy reguláris nyelv, akkor létezik $p \in \mathbb{N}$ szám, hogy $\forall s \in L$ esetén, ha s hossza legalább p , akkor s felbontható $s = xyz$ alakba úgy, hogy:

1. $xy^n z \in L$ minden $n \geq 0$ esetén,
2. $|y| > 0$,
3. $|xy| \leq p$.

Bizonyítás:

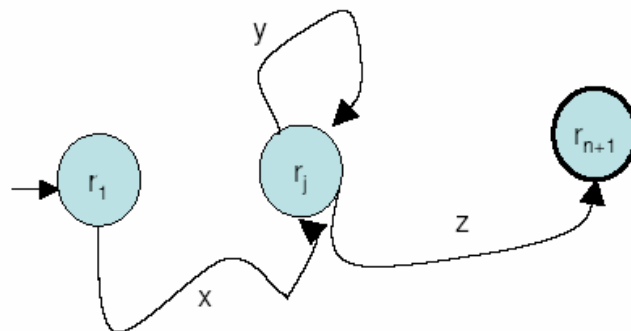
- $p = |Q|$ legyen.
- Legyen $s \in L$, $s = s_1 s_2 \dots s_n$, $p \leq n$. Az s -et elfogadó állapotok sorozata legyen $r = r_1, r_2 \dots r_{n+1}$, ekkor $p < n+1$. Tehát r tagjai között kell lennie legalább két azonosnak, a **skatulya elv** miatt. Jelölje ezeket r_j és r_k .

$$x = s_1 s_2 \dots s_{j-1},$$

$$y = s_j s_{j+1} \dots s_{k-1},$$

$$z = s_k s_{k+1} \dots s_n.$$

1. $xy^n z \in L$ minden $n \geq 0$ esetén valóban,
2. $|y| > 0$, mert $j \neq k$,
3. $|xy| = k - j + 1 \leq p$.



Pumpáló lemma: CF nyelv

Ha $L \in \Sigma^*$ egy CFG, akkor létezik $p \in \mathbb{N}$ szám, hogy $\forall s \in L$ esetén, ha s hossza legalább p , akkor s felbontható $s = uvxyz$ alakba úgy, hogy:

1. $uv^nxy^nz \in L$ minden $n \geq 0$ esetén,
2. $|vy| > 0$,
3. $|vxy| \leq p$.

Bizonyítás:

Legyen G egy, az adott L -t generáló CFG, és b legyen a szabályok jobb oldalán szereplő szimbólumok max. száma, feltételezzük hogy $b \geq 2$. Tetszőleges m mélységű levezetési fában max. b^m levél szerepel.

Legyen $p = b^{|V|+2}$, ahol $|V|$ a G -beli változók száma. Mivel $b \geq 2$, így $p > b^{|V|+1}$, ezért ha s L -beli legalább p hosszú sorozat, akkor minden levezetési fája legalább $|V|+2$ mély kell hogy legyen.

Legyen s . L -beli legalább p hosszú sorozat, legyen τ s egy legkevesebb csomópontú levezetési fája. τ legalább $|V|+2$ mély, ezért a leghosszabb út a gyökértől egy levélig is legalább $|V|+2$.

Ez az út legalább $|V|+1$ változót tartalmaz, ekkor a galambdúc elv miatt van olyan változó, mely legalább 2-szer szerepel az út végső, $|V|+1$ hosszú szakaszán. Legyen ez A.

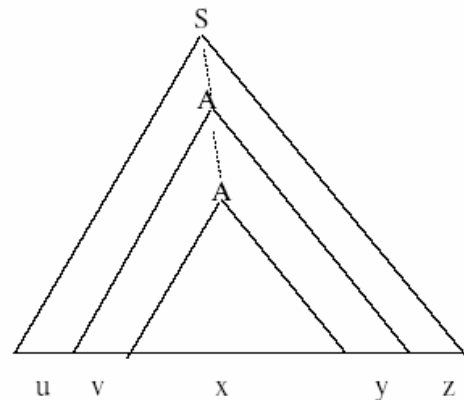
$$s = uvxyz$$

$uv^nxy^nz \in L$ $n \geq 0$ esetén valóban v és y közül

legalább az egyik nem ε .

Különben τ -nál kevesebb csomópontú levezetési fa létezne s -hez.

v_{xy} legfeljebb p hosszú, mivel A -ból egy maximum $|V|+2$ mélységű fa vezet le, melynek legfeljebb $b^{|V|+2} = p$ levele lehet.



4. Környezetfüggetlen grammatikák, egyértelmű ill. többértelmű CF grammatika Chomsky-féle normálalak, BNF. Műveletek CF nyelveken. Verem automata, kapcsolata CF nyelvekkel.

Környezetfüggetlen nyelvtan (CFG)

Definíció: A $G = (V, \Sigma, R, S)$ grammatika környezetfüggetlen, ha a levezetési szabályainak bal oldalán egyetlen nemterminális szerepel, azaz a szabályok

$$A \rightarrow \beta \text{ alakúak, ahol } A \in V, \beta \in (V \cup \Sigma)^*$$

Definíció: Egy környezetfüggetlen grammatika szerinti levezetés baloldali, ha a levezetés egyes lépéseiben a legbaloldalibb nemtermimálist helyettesítjük be.

Egyértelmű-többértelmű

Definíció: Egy nyelvtan egyértelmű, ha minden generált szavának egyetlen baloldali levezetése van. Különben többértelmű.

Megj: Minden baloldali levezetés meghatároz egy levezetési fát. Tehát egyértelmű nyelvtan esetén minden generált szóhoz egyetlen levezetési fa létezik.

Definíció: Egy CF nyelv eredendően többértelmű, ha nem generálható egyértelmű grammatikával.

Chomsky-féle normálalak (CNF)

Definíció: A $G = (V, \Sigma, R, S)$ CF grammatika Chomsky-féle normál alakban adott, ha a szabályok az alábbi alakúak:

$$A \rightarrow BC$$

$$A \rightarrow a$$

$$S \rightarrow \varepsilon$$

ahol $A, B, C \in V$, $a \in \Sigma$, B és C nem kezdőszimbólumok.

Tétel: Tetszőleges CF nyelvhez megadható öt generáló grammatika Chomsky-féle normál alakban.

Backus-Naur Form (BNF)

programozási nyelvek szintaxisának definiálására szolgál.

Műveletek CF nyelveken

Tétel: Legyen L a CF nyelvek halmaza.

L zárt a reguláris operátorokra nézve.

Azaz ha $A_1 \in L$ és $A_2 \in L$, akkor:

1. $A_1 \cup A_2 \in L$.
2. $A_1 \circ A_2 \in L$.
3. $A_1^* \in L$.

A metszetre és a komplementerre nézve L nem zárt!

Egy rekurzív és egy CF nyelv metszete CF nyelv lesz

Verem automata

$M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ ahol:

1. Q , egy véges halmaz az állapotok halmaza
2. Σ , egy véges halmaz a bemeneti abc
3. Γ , egy véges halmaz a verem abc
4. $\delta: Q \times \Sigma_\epsilon \times \Gamma_\epsilon \rightarrow P(Q \times \Gamma_\epsilon)$ az állapotátmenet függvény
5. $q^0 \in Q$ a kezdeti állapot
6. $F \subseteq Q$ a végállapotok halmaza

Jelölés: $\Sigma_\epsilon = \Sigma \cup \{\epsilon\}$, $\Gamma_\epsilon = \Gamma \cup \{\epsilon\}$ ('nem olvasunk-írunk')

Definíció: Felismert (elfogadott) jelsorozat

$M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ egy verem automata, $w = w_1, w_2 \dots w_n$ Σ_ϵ -beli jelek sorozata.

M felismeri w -t, ha létezik:

$r_0, r_1 \dots r_n$ Q -beli állapotok sorozata, és

$s_0, s_1 \dots s_n$ Γ_ϵ -beli jelek sorozata hogy:

1. $r_0 = q_0$ és $s_0 = \epsilon$
2. $(r_{i+1}, b) \in \delta(r_i, w_{i+1}, a)$ $i = 0, \dots, n-1$, ahol:
 $s_i = at$ és $s_{i+1} = bt$ valamely $a, b \in \Gamma_\epsilon$ és $t \in \Gamma^*$
3. $r_n \in F$

Definíció: M verem automata felismeri az A nyelvet, ha $A = \{w \mid M \text{ felismeri } w\}$.

Minden L CF nyelvhez létezik őt felismerő verem automata.

5. Turing gép, ekvivalens fajtái. Rekurzív és rekurzíve felsorolható nyelvek, számosságuk. Példa rekurzíve felsorolható, de nem rekurzív nyelvre, illetve nem rekurzíve felsorolható nyelvre. Turing-gépek és nyelvtanok kapcsolata.

Turing gép

$M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{elfogad}}, q_{\text{elutasít}})$ ahol:

1. Q , egy véges halmaz az állapotok halmaza
2. Σ , egy véges halmaz a bemeneti abc, mely nem tartalmazza a $_$ jelet
3. Γ , egy véges halmaz a szalag abc, ahol $_ \in \Gamma$ és $\Sigma \subseteq \Gamma$
4. $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{J, B\}$ az állapotátmenet függvény
5. $q_0 \in Q$ a kezdeti állapot
6. $q_{\text{elfogad}} \in Q$ az elfogadó állapot
7. $q_{\text{elutasít}} \in Q$ az elutasító állapot

Definíció: M Turing gép elfogad egy w sorozatot, ha létezik:

$C_1, C_2 \dots C_n$ konfigurációk sorozata, hogy:

1. C_1 a kiindulási konfiguráció,
2. C_i rákövetkezője C_{i+1} , $i = 1, \dots, n-1$,
3. C_n elfogadó konfiguráció.

Jelölés: $L(M)$ jelöli az M Turing gép által elfogadott sorozatok halmazát. Ez a gép által felismert nyelv.

Turing gép variációk, ekvivalenciájuk az alap géppel

- fej helyben maradhat
- több szalag
- nem determinisztikus állapotátmenet
- felsoroló (enumerator)

Ezek mind ekvivalensek az alapgéppel, többet nem tudnak, esetleg kényelmesebbek lehetnek bizonyos szempontból.

Definíció: M és S Turing gép változatok ekvivalensek, ha $L(M) = L(S)$.

Rekurzív és rekurzíve felsorolható nyelvek:

Definíció: Egy $L \subseteq \Sigma^*$ nyelv Turing felismerhető vagy rekurzíve felsorolható, ha van olyan M Turing gép, melyre $L(M) = L$.

Megj: $s \in \Sigma^* - L$ esetén nem biztos, hogy megáll a gép, így csak az L elemeire kapunk igenlő választ.

Definíció: Egy $L \subseteq \Sigma^*$ nyelv Turing eldönthető vagy rekurzív, ha van olyan M Turing gép, melyre $L(M) = L$, és M minden $s \in \Sigma^*$ szóra megáll.

Megj: Minden rekurzív nyelv rekurzíve felsorolható is. (Az érdekes kérdés a fordított tartalmazás.)

Példa rekurzíve felsorolható, de nem rekurzív nyelv:

Az olyan TG kód (inputszó párok L_u nyelve), melyekre a TG elfogadja az inputot az univerzális nyelv.

$L_u = \{ w\#s \in \{0, 1\}^*, M_w \text{ létezik, és } s \in L_{M_w} \}$

Biz.: indirekten.

Példa nem rekurzíve felsorolható nyelv:

Def: Diagonális nyelv: azon TG kódja, mely TG-ek nem fogadják el a saját kódjukat

$L_d = \{ w \in \{0, 1\}^*, \text{ az } M_w \text{ gép létezik, és } w \notin L_{M_w} \}$

Biz.: indirekten.

6. Church-Turing tézis, példa algoritmikusan nem eldönthető problémára. TG kódja, az univerzális TG fogalma, használata.

Church-Turing tézis

(Algoritmus: véges sok lépéssel kiszámít egy fv-t, eldönt egy kérdést.)

Megvalósítható-e minden algoritmus Turing géppel?

Church-Turing tézis: Ami algoritmussal kiszámítható/eldönthető, az Turing kiszámítható/eldönthető. Nevezetesen:

- Egy $f : \Sigma^* \rightarrow \Sigma^*$ parciális fv algoritmussal kiszámítható $\Leftrightarrow$ f parciálisan rekurzív.
- Egy $f : \Sigma^* \rightarrow \Sigma^*$ teljes fv algoritmussal kiszámítható $\Leftrightarrow$ f rekurzív.
- Egy L nyelvre a nyelvbe való tartozás problémája algoritmussal eldönthető $\Leftrightarrow$ a nyelv rekurzív.

Algoritmikusan nem eldönthető probléma

Hilbert 10 problémája: Diofantikus egyenletek általános megoldhatósága: adjunk algoritmust, mely tetszőleges d . e -re eldönti, hogy az megoldható-e.

$f(x_1, \dots, x_n) = 0$ egész megoldását keressük, ahol $f(x_1, \dots, x_n)$ egész együtthatós, n változós polinom.

Felismerhető, de nem dönthető el.

Dominóprobléma: k féle dominóval lefedhető-e a sík? Nem felismerhető, nem eldönthető.

TG kódja

Feltesszük az $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{elfogad}}, q_{\text{elutasít}})$ TG-ről, hogy:

- $Q = \{0, 1, \dots, q\}$, 0 a kezdő, q az elfogadó, $q-1$ az elutasító állapot
- $\Sigma = \{0, 1\}$ bemeneti abc,
- $\Gamma = \{0, 1, \dots, s\}$ szalag abc, s felel meg az üres jelnek, $_$ -nek
- $J=1, B=0$

Ekkor M egyértelműen megadható egy $\{0, 1\}^*$ -beli w szóval, ez M kódja.

Univerzális TG fogalma

Képes szimulálni tetszőleges, a w kódjával adott M_w TG-et.

Tétel: Van olyan U 3 szalagos TG, melyre teljesül, hogy ha $w, s \in \{0, 1\}^*$, és M_w létezik, akkor az U gép pontosan akkor fogadja el (utasítja el, kerül végtelen ciklusba) az $w\#s$ bemenetet, amikor M_w az s bemenetet elfogadja (elutasítja, ciklusba kerül).

A 3 szalag szerepe:

w#s	nem változik, M_w működését adja meg
s	a szalag tartalma, M_w állapota szerint
0	M_w pillanatnyi állapota

Univerzális TG működése

0. Ellenőrzi, hogy w TG-t ír-e le, ha nem, elutasít. Ha igen, beírja s-t a 2., és 0-t (a kezdőállapot kódját) a 3. szalagra.

1. Az első szalag alapján meghatározza, hogy M_w mit lépne, ha a 3. szalagon megadott állapotban lenne, és az olvasófej alatt a 2. szalagon szereplő jelet látná.

U megteszi M_w lépését, és megfelelően módosítja a 2. és 3. szalag tartalmát.

U megáll elfogadó ill. elutasító állapotban, ha az M_w szerinti új állapot éppen elfogadó ill. elutasító. Ha nem, akkor az 1. lépést ismétli.

7. Primitív rekurzív függvények, μ -rekurzív függvények, Turing kiszámítható függvények.

Primitív rekurziós függvények

Def: Az alap függvények a következők:

- a) $N \rightarrow N$ rákövetkező fv: $\text{succ}(n) = n+1$.
- b) $N^k \rightarrow N$ ($k \geq 0$) k-változós zero fv: $\text{zero}_k(n_1, \dots, n_k) = 0$.
- c) $N^k \rightarrow N$ ($k \geq 0$) k-változós j-dik identitás fv: $\text{id}_{k,j}(n_1, \dots, n_k) = n_j$.

Bonyolultabb függvényeket a következő módokon kombinálunk:

Def: Legyen $k, m \geq 0$, és adottak a $g: N^k \rightarrow N$ illetve $h_1, \dots, h_k: N^m \rightarrow N$ függvények. Ekkor g kompozíciója a $h_1, \dots, h_k$ függvényekkel az az $f: N^m \rightarrow N$ fv, melyre:

$$f(n_1, \dots, n_m) = g(h_1(n_1, \dots, n_m), \dots, h_k(n_1, \dots, n_m)).$$

Def: Legyen $k \geq 0$, és adottak a $g: N^k \rightarrow N$ illetve $h: N^{k+2} \rightarrow N$ függvények. Ekkor g és h által rekurzívan definiált függvény az az $f: N^{k+1} \rightarrow N$ fv, melyre:

$$f(n_1, \dots, n_k, 0) = g(n_1, \dots, n_k).$$

$$f(n_1, \dots, n_k, m+1) = h(n_1, \dots, n_k, m, f(n_1, \dots, n_k, m)).$$

Def: A primitív rekurzív függvények az alap függvények, és a belőlük tetszőleges számú kompozícióval és rekurzióval származtatott függvények.

- A primitív rekurzív függvények számossága megszámlálhatóan végtelen.
- Nem minden kiszámítható fv. primitív rekurzív.

μ -rekurzív függvények

Def: Legyen $k \geq 0$. A $g: N^{k+1} \rightarrow N$ függvény minimalizációja az az $f: N^k \rightarrow N$ függvény, melyet a következő módon értelmezünk:

$$f(n_1, \dots, n_k) = \text{a legkisebb } m, \text{ melyre } g(n_1, \dots, n_k, m) = 1, \text{ ha van ilyen } m$$

$$f(n_1, \dots, n_k) = 0 \text{ különben.}$$

Legyen $k \geq 0$. A $g: N^{k+1} \rightarrow N$ függvény minimalizálható, ha a fenti eljárás mindig megáll, azaz minden $n^1, \dots, n^k \in N$ esetén van olyan $m \in N$, hogy $g(n^1, \dots, n^k, m) = 1$.

Def: Egy f fv μ -rekurzív, ha előállítható az alap függvényekből a kompozíció, a rekurzió és a minimalizáció operátorokkal, az utóbbit minimalizálható függvényre alkalmazva.

Tétel: Az $f: \mathbb{N}^k \rightarrow \mathbb{N}$ függvény μ -rekurzív akkor és csak akkor, ha rekurzív, azaz kiszámítható Turing géppel.

Turing kiszámítható függvények

Definíció: $x \in \Sigma^*$ kezdeti bemeneti sorozatra az M Turing gép **outputja** az $y \in \Sigma^*$ sorozat, ha M megáll az x inputra, és megállásakor a szalagon éppen w szerepel (az üres szimbólumok előtt), valamint a fej ismét az első pozíción áll.

Ezt a tényt $y = M(x)$ jelöli.

Definíció: Az M által kiszámított fv az a $f_M: \Sigma^* \rightarrow \Sigma^*$ parciálisan értelmezett fv, mely minden x -en értelmezett, melyre van y , hogy $y = M(x)$.

Definíció: Egy $f: \Sigma^* \rightarrow \Sigma^*$ parciális függvény parciálisan rekurzív, ha van M TG, hogy $f = f_M$. Ha ezen felül f minden Σ^* -ra értelmezve van, akkor f rekurzív. A rekurzív függvények Turing értelemben kiszámíthatók

8. Algoritmusok (TG-ek) idő és tárigénye. A **P** nyelvosztály jellemzése, zártsága nyelvek feletti műveletekre. Nevezetes P-beli problémák. Tár-idő tétel. A P, NP, PSPACE, EXPTIME és a co operátorral kapott nyelvosztályok, ezek tartalmazási viszonyai.

TG idő- és tárigénye

M TG számolási ideje az s inputon a megállásig végrehajtott lépések száma.

A tárigény pedig az olvasott (felhasznált) szalagcellák száma. Ebbe nem feltétlenül számoljuk bele az inputot és az outputot.

M TG, n jelből álló input, akkor

$T_M(n)$: max számolási idő

$S_M(n)$: max tárigény

Ha van olyan n jelből $\in \Sigma^*$ amellyel M-et elindítva az véges sok lépés után nem áll meg $\Rightarrow$

$T_M(n) = \infty$

Ha N és M TG-ek melyekre $T_M(n) < T_N(n)$ teljesül minden elég nagy n-re $\Rightarrow$ M algoritmus gyorsabbnak mondjuk N-nél.

Def: f, g: $\mathbb{N} \rightarrow \mathbb{R}^+$ függvények. $f(n) = O(g(n))$, ha találhatók c és n_0 pozitív egészek, hogy $f(n) \leq c g(n)$ ha $n \geq n_0$.

Azt mondjuk, hogy g(n) asszimptotikus felső korlátja f(n)-nek.

Definíció: Legyen t: $\mathbb{N} \rightarrow \mathbb{N}$ függvény olyan, hogy $t(n) \geq n$ minden n-re.

Az M TG **t(n) időkorlátos**, ha az n hosszú bemenetek esetén legfeljebb t(n) lépést tesz meg, azaz $T_M(n) \leq t(n)$.

Definíció: $TIME(t(n))_- = \{L \subseteq I^* \mid L \text{ felismerhető egy } O(t(n)) \text{ időkorlátos M TG-pel.}\}$.

Definíció: Legyen s: $\mathbb{N} \rightarrow \mathbb{N}$ függvény olyan, hogy $s(n) \geq \log_2 n$ minden n-re. Az M TG **s(n) tárkorlátos**, ha az n hosszú bemenetek esetén legfeljebb s(n) db cellát használ a munkaszalagon, azaz $S_M(n) \leq s(n)$.

Definíció: $SPACE(s(n))_- = \{L \subseteq I^* \mid L \text{ felismerhető egy } O(s(n)) \text{ tárkorlátos M TG-pel.}\}$.

P nyelv osztály

Definíció: $\mathbf{P} = \cup_{k \geq 1} \text{TIME}(n^k)$ a polinom időben egyszalagos TG-pel eldönthető nyelvek osztálya.

Nevezetes P-beli problémák:

$\text{PATH} = \{ \langle G, s, t \rangle \mid G \text{ irányított gráf, és van benne irányított út } s\text{-ből } t\text{-be} \}.$

$\text{RELRPIME} = \{ \langle x, y \rangle \mid x \text{ és } y \text{ relatív prímek} \}.$

$\text{GRAPH-2-COLOR} = \{ G \mid G \text{ gráf csúcsai kiszínezhetők 2 színnel} \}.$

$\text{EULER-CYCLE} = \{ G \mid G \text{ gráfban van Euler kör} \}.$

$\text{PRIME} = \{ x \mid x \text{ prím} \}.$

Tétel: Ha az L nyelv $n^{\log n}$ -nél rövidebb időben nem ismerhető fel, akkor $L \notin \mathbf{P}$.

Tár-idő tétel

Tétel: Ha $L \in \text{SPACE}(s(n))$, akkor van olyan L -től függő c konstans, hogy $L \in \text{TIME}(c^{s(n)})$.

P, NP, PSPACE, EXPTIME és a co operátorokkal kapott nyelv osztályok

Jelölés: L I feletti nyelv esetén $L^C = I^* \setminus L$ az L komplementere.

Def: $\mathcal{L}$ nyelv osztály esetén $\text{co}\mathcal{L} = \{ L \mid L^C \in \mathcal{L} \}$ az nyelv osztály komplementere.

Tétel:

a) $\text{co}(\text{co}\mathcal{L}) = \mathcal{L}$

b) Ha $\mathcal{A} \subseteq \mathcal{B}$ nyelv osztályok, akkor $\text{co}\mathcal{A} \subseteq \text{co}\mathcal{B}$.

$\text{TIME}(t(n)) = \text{coTIME}(t(n))$, azaz $L \in \text{TIME}(t(n))$ akkor és csak akkor, ha $L^C \in \text{TIME}(t(n))$.

$\text{SPACE}(s(n)) = \text{coSPACE}(s(n))$, azaz $L \in \text{SPACE}(s(n))$ akkor és csak akkor, ha

$L^C \in \text{SPACE}(s(n))$.

Definíció: $\mathbf{EXPTIME} = \cup_{k \geq 1} \text{TIME}(2^{n^k})$ az exponenciális időben felismerhető nyelvek osztálya.

Definíció: $\mathbf{PSPACE} = \cup_{k \geq 1} \text{SPACE}(n^k)$ a polinom tárban felismerhető nyelvek osztálya.

Tétel: $\mathbf{P} \subseteq \mathbf{PSPACE} \subseteq \mathbf{EXPTIME}$

9. Az **NP** nyelv osztály jellemzése, nevezetesen NP-beli nyelvek, tanukkal. Az **NP**-teljes nyelvek jellemzése. Nevezetesen NP-teljes problémák.

Nem determinisztikus Turing gép

$M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{elfogad}}, q_{\text{elutasít}})$ ahol:

1. Q , egy véges halmaz az állapotok halmaza
2. Σ , egy véges halmaz a bemeneti abc, mely nem tartalmazza a $_$ jelet
3. Γ , egy véges halmaz a szalag abc, ahol $_ \in \Gamma$ és $\Sigma \subseteq \Gamma$
4. $\delta: Q \times \Gamma \rightarrow \mathcal{P}(Q \times \Gamma \times \{J, B\})$ az állapotátmenet függvény
5. $q_0 \in Q$ a kezdeti állapot
6. $q_{\text{elfogad}} \in Q$ az elfogadó állapot
7. $q_{\text{elutasít}} \in Q$ az elutasító állapot

Definíció: Legyen $t: \mathbb{N} \rightarrow \mathbb{N}$ függvény olyan, hogy $t(n) \geq n$ minden n -re. Az M NDTG **$t(n)$ időkorlátos**, ha az n hosszú bemenetek esetén M minden számítási út mentén legfeljebb $t(n)$ lépést megtéve megáll. Azaz n hosszú bemenetek esetén legfeljebb $t(n)+1$ mély a számítási fa.

Definíció: $\text{NTIME}(t(n)) = \{L \subseteq I^* \mid L \text{ felismerhető egy } O(t(n)) \text{ időkorlátos } M \text{ NDTG-pel.}\}$.

Az NP nyelv osztály

Definíció: $\text{NP} = \bigcup_{k \geq 1} \text{NTIME}(n^k)$ a polinom időben egyszalagos NDTG-pel eldönthető nyelvek osztálya. Megj: Nem ismert olyan fizikai megvalósítás, mely $t(n)$ -nel arányos időben szimulálná egy $t(n)$ időkorlátos NDTG működését.

$P \subseteq \text{NP}$; $P \subseteq \text{NP} \cap \text{coNP}$

NP-be tartozás tanukkal

Def: Legyen L egy Σ feletti nyelv. L polinomiálisan bizonyítható, ha van olyan V algoritmus, hogy $L = \{w \mid V \text{ elfogadja } \langle w, c \rangle -t \text{ valamely } \Sigma^* \text{-beli } c\text{-re}\}$, és V polinom idejű w hosszát tekintve.

NP azokat a nyelveket tartalmazza, melyek polinomiálisan bizonyíthatók

Egy NP-beli nyelv komplementere nem mindig NP-beli.

Pl.:

Merlin lovagjai és udvarhölgyei nem házashatók össze / tanu adható

Nincs Hamilton kör a gráfban / tanu nem adható

G gráf nem síkba rajzolható / tanu adható: valamelyik Kuratowski-gráf mint G részgráfja

Nem ismert, hogy $coNP = NP$ teljesül-e.

NP-teljes nyelvek és problémák

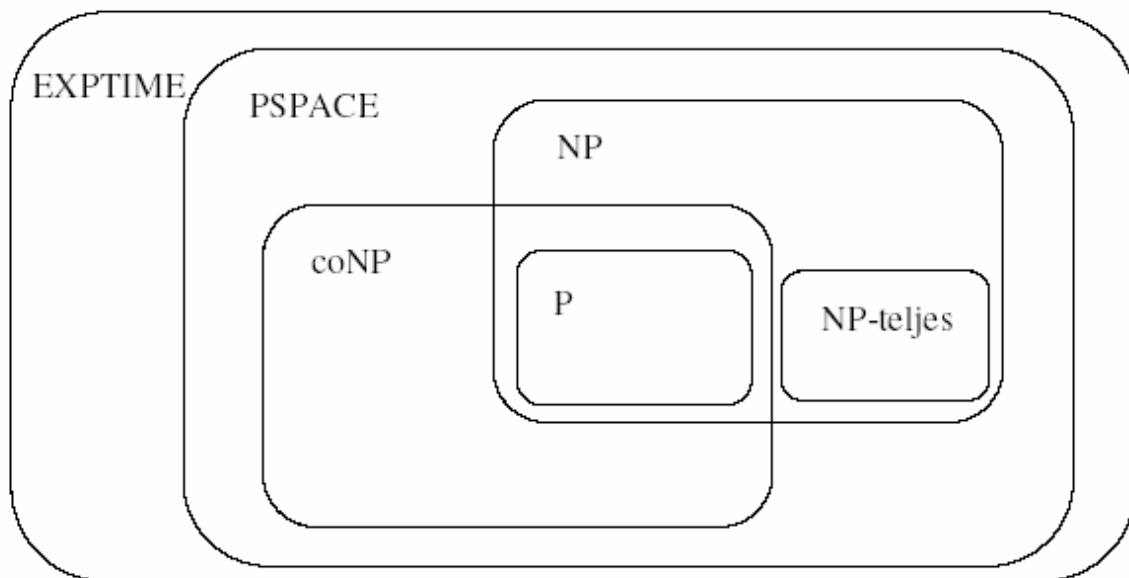
Egy problémát NP-nehéznek nevezünk, ha minden NP-beli probléma visszavezethető rá. Ha probléma maga is része NP-nek, akkor azt mondjuk, hogy a probléma NP-teljes. Ha NP-teljes problémát meg tudnánk oldani polinom időben, akkor minden NP-beli probléma is megoldható lenne polinom időben.

Nevezetes NP-teljes problémák

Tétel: VERTEX COVER (lefogható-e G minden éle legfeljebb k ponttal) NP-teljes.

Tétel: k-COLORABILITY (kiszínezhető-e G minden éle legfeljebb k színnel) NP-teljes $k \geq 3$ esetén.

Tétel: HAMPATH NP-teljes.



10. Kolmogorov bonyolultság, összenyomhatóság, összenyomhatatlanság. Invariancia tétel és következményei. A Kolmogorov bonyolultság függvény nem rekurzív.

Összenyomhatóság

Def: $I = \{0, 1\}$, M egy olyan TG, melynek bemeneti és szalag abc-je I .

M az $f_M: I^* \rightarrow I^*$ parciális függvényt számolja ki. Ekkor egy I^* -beli x szó esetén

$$C_M(x) = \begin{cases} \min \{ |y| : y \in I^*, f_M(y) = x \} & \text{ha van ilyen } y \\ \infty & \text{Különben} \end{cases}$$

$C_M(x)$ azt adja meg, hogy M milyen mértékben nyomja össze x -et.

Invariancia tétel

Legyen U egy univerzális TG. Ekkor tetszőleges M TG-hez $c_M \in \mathbb{Z}^+$ szám, hogy minden I^* -beli x szó esetén

$$C_U(x) \leq C_M(x) + c_M.$$

Kolmogorov-bonyolultság

Def: Egy I^* -beli x szó esetén a szó Kolmogorov-bonyolultsága $C(x)$, ahol $C(x) = C_U(x)$.

U univerzális TG.

Összenyomhatatlanság

Def: Egy I^* -beli x szó összenyomhatatlan, ha $C(x) \geq |x|$.

Tétel: Legyen $k \in \mathbb{Z}^+$ szám. Ekkor

- 1) Legfeljebb $2^{k+1} - 1$ olyan x szó van, melyre $C(x) \leq k$.
- 2) Minden n természetes számra van n hosszú, összenyomhatatlan szó.

Nem rekurzív

Tétel: A Kolmogorov bonyolultságot megadó $C: I^* \rightarrow \mathbb{Z}^+$ függvény nem rekurzív.