Félig strukturált adatbázisok indexei

$G = (V, E, \Sigma)$ címkézett irányított gráf gyökérrel

út (irányított) $n_0, n_1, \dots, n_p$ ahol n_0 = gyökér

címkesorozat l_0, l_1 l_p

(egyszerű kifejezés

út illeszkedik a címkesorozatra: $\text{címke}(n_i) = l_i$

Reguláris kifejezés:

$$R = \varepsilon \mid \Sigma \mid \_ \mid R.R \mid R|R \mid (R) \mid R^*$$

$L(R)$ R által meghatározott címkesorozatok (Reguláris nyelv)

n illeszkedik R-re, ha $L(R)$-ben van olyan címkesorozat amire illeszkedik

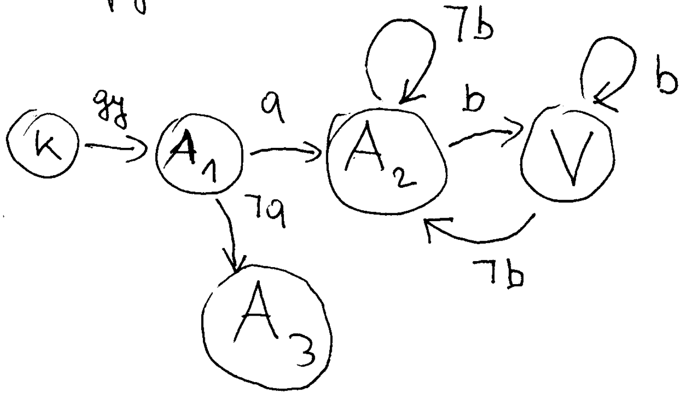
R kiértékelés eredménye G-n:

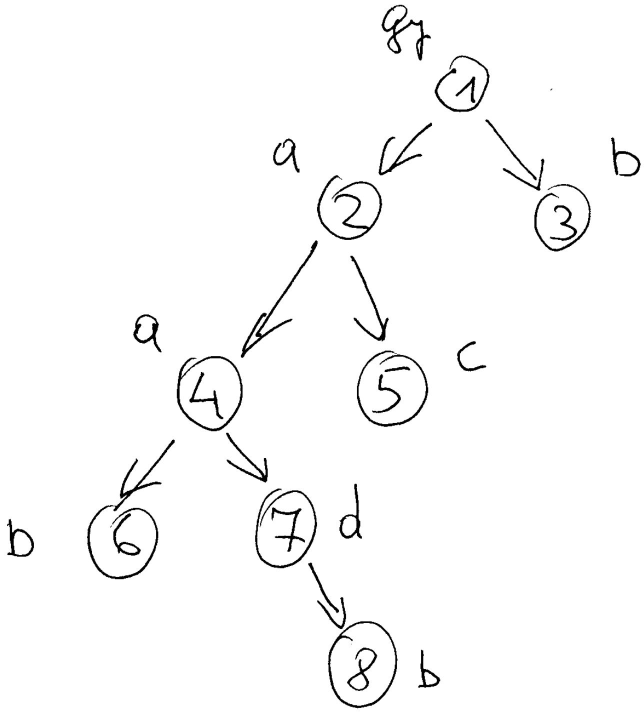
$$R(G) = \{ n \mid n \text{ illeszkedik } R\text{-re} \}$$

M: $-L(R)$ mérete végtelen is lehet

$- L(R) = \emptyset$ eldönthető!

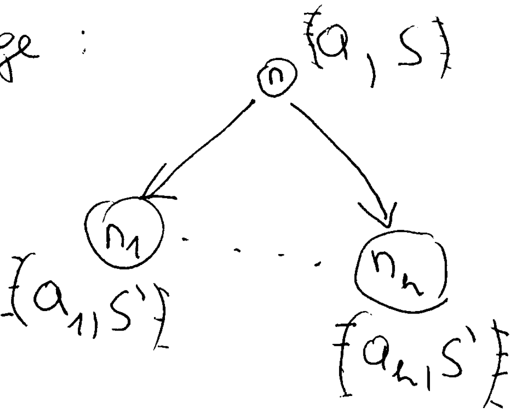
$- R \Leftrightarrow A_R$ véges Nem det. automata, gy . a . * . b





Naiv-kiértékelés (G, A_R)

lényege:



Ha $S \xrightarrow{a} S' \Rightarrow (n, S)$ [illegible]

és $S' \in Vég \Rightarrow n \in R(G)$

különben

n_i-ben folytatjuk

S' állapottal

Példában

$(1, K)$

$(2, A_1)$ $(3, A_1)$

$(4, A_2)$ $(5, A_2)$ $(3, A_3)$

$(6, A_2)$ $(7, A_2)$

$(6, V)$ $(8, A_2)$

$(8, V)$

$R(G) = \{6, 8\}$

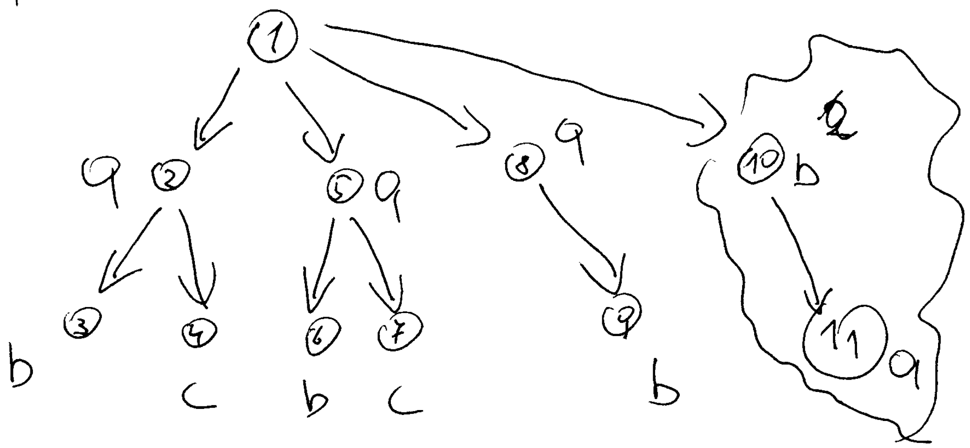
A: $R(G)$ költsége $O(|V| \cdot |A_R|)$ ahol

$|A_R|$ az állapotok száma

$$n_1 \overset{R}{=} n_2$$

n_1 és n_2 nem különböztethető meg reguláris kifejezéssel $\nexists R : n_1 \in L(R)$ de $n_2 \notin L(R)$ és fordítva

Példa

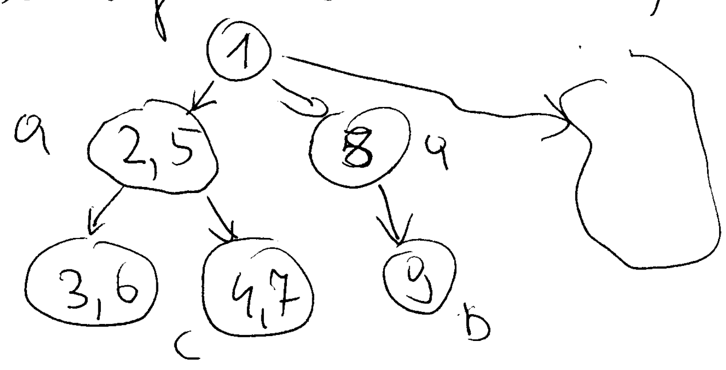


$$L(n) = \{ l_0, \dots, l_p \mid n \text{ illeszkedik } l_0, \dots, l_p\text{-re} \}$$

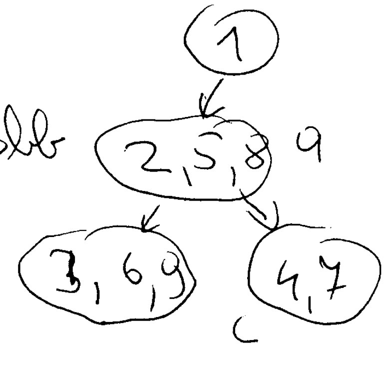
K: n_1 és n_2 nem különböztethető meg
$\iff L(n_1) = L(n_2)$

M: $n_1 \overset{R}{=} n_2$ esetén a gyökértől hozzájuk vezető úton is rendre megkülönböztethetetlen csúcsok szerepelnek

Osztályozás (Partíció)



durvább

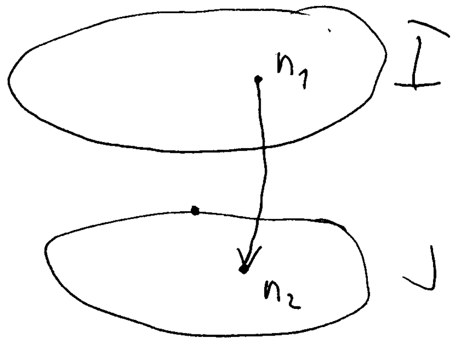


V-n alappartíció: címke szerinti osztályozás

I(G) index: (P, E(I))

partíció az alappartíció finomítása

$(I, J) \in E(I)$ akkor $\exists\, n_1 \in I,\ n_2 \in J:$ $(n_1, n_2) \in E$



Naiv index: $\overset{R}{\cong}$ ekvivalenciaosztályai

A: I naiv index $\Rightarrow$ $R(G) \cap I = \emptyset$ vagy $R(G) \supseteq I$

Naiv-indexes kiértékelés (G, R):

$\bigcup$ Naiv kiértékelés $(I(G), A_R)$

A: R(G) = Naiv-indexes kiértékelés(G, R)

A: Naiv indexes keresése PSPACE-COMPLETE probléma

I(G) biztonságos $n \mid \ell_0, \ldots, \ell_p$ ~~esetén~~

$\Downarrow$ illeszthető $\Uparrow$ pontos

$osztály(n) \mid \ell_0, \ldots, \ell_p$

biztonságos $R(G) \subseteq R(I(G))$

pontos $\supseteq$

A: 1. $\forall$ index biztonságos

2. Naív bizt és pontos

Biszimulációs V-n (ekvivalencia)

$u \approx v$ esetén

1) $\text{címke}(u) = \text{címke}(v)$

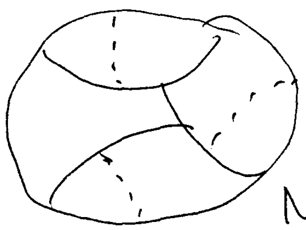
2) u' és v' $\exists$ és $u' \approx v'$

$$\begin{tikzcd} u' \arrow[d] & v' \arrow[d] \\ u \arrow[r, phantom, "\approx"] & v \end{tikzcd}$$

és fordítva.

1-index: biszimulációs ekvivalencia osztályai

K:



Naív-index finomítása az 1 index

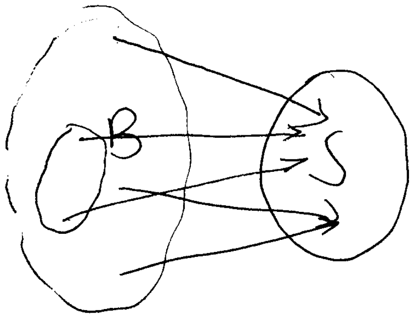
A: 1-index bizt és pontos

Keresés hatékonyan PT algoritmussal

Stabil partíciók. $G=(V,E)$, P

$B \subsetneq V$

B stabil S-re ha $B \subseteq E^{-1}(S)$ vagy
$B \cap E^{-1}(S) = \emptyset$



$E^{-1}(S)$

P stabil S-re $\forall$ blokkja stabil S-re

P stabil P'-re P stabil P' $\forall$ blokkjára

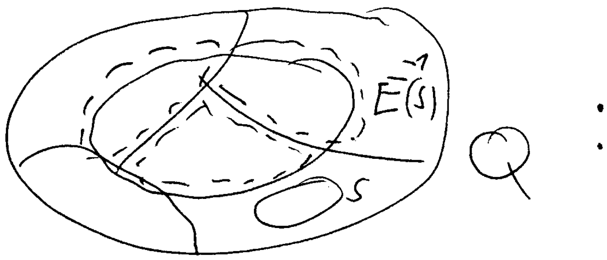
P stabil, ha P-re stabil

legdurvább stabil partíció problémája,
adott P, legdurvább stabil finomítását
keressük

M: $B \subseteq E(S)$ vagy $B \cap E(S) = \emptyset$
feltétel a G^{-1}-re vonatkozó stabilitás
(élek megfordítása)

hasít (S, Q) $E^{-1}(S)$-sel finomítunk

$S \subseteq V$ partíció



S hasítója Q-nak,
ha hasít$(S,Q) \neq Q$

A: 1) $Q \geq P$ (Q finomítása P-nek)
és P stabil S-re
$\Rightarrow$ Q stabil S-re

2) P stabil S-re és T-re
$\Rightarrow$ P stabil $S \cup T$-re

3) $P \geq Q \Rightarrow$ hasít$(S,P) \geq$ hasít(S,Q)

4) hasít$(S,$ hasít$(T,P)) =$ hasít$(T,$ hasít$(S,P))$

5) P legdurvább olyan partíciója, amely stabil S-re <u>és</u> T-re: hasít$(S,$ hasít$(T,P))$

Naiv PT(V, E, P)

lényege: ha nem stabil a partíció S blokkjára, akkor hasítsunk S-sel

~~Költség: O(k~~ $|E|$

A: legdurvább stabil finomítást kapjuk.

$O(|V|\cdot|E|)$

Javított változat (2 partíció kezelésével)

PT(V, E, P)

$Q := P$ invariáns — S

$X := \{V\}$ partíció $(Q \geq X)$

while $X \neq Q$

do S X blokkja, ami Q-nak nem blokkja

és S-ben B a Q blokkja

$|B| \leq \frac{|S|}{2}$

$X = (X - \{S\}) \cup (B, S-B)$

$Q = \text{hasít}(S-B, \text{hasít}(B, Q))$

return Q

Hatékonysági tétel — $O(|E|\log|V|)$

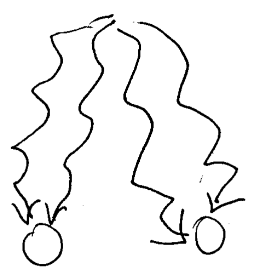
tárköltség $O(|E|+|V|)$

A: I, J 1-index csúcsai:

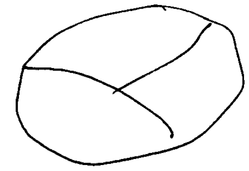
$\Rightarrow J \subseteq E(I)$ vagy $J \cap E(I) = \emptyset$

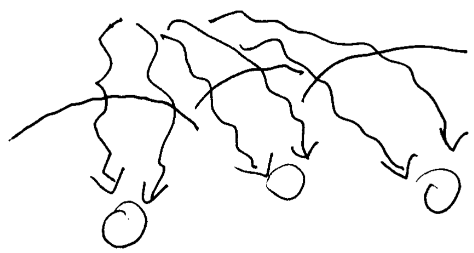
K: 1) J stabil I-re G^{-1}-ben

2) G legdurvább 1-indexe = G^{-1}-ben az alappartíció legdurvább stabil finomításával

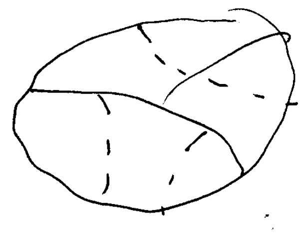


1-index összes ős alapján osztályoz





k-ik szintű ősök alapján osztályozunk



k-biszimuláció

1) $u \approx^0 v \Leftrightarrow$ címke(u) = címke(v)

2)
$$\begin{array}{ccc} u' & & v' \\ \downarrow & & \downarrow \\ u & \approx^k & v \end{array} \qquad \exists v' \quad u' \approx^{k-1} v'$$

és fordítva

$A(k)$-index k-biszimulációs osztályok

↑

approximatív

M: $A(0)$ alappartíció, $A(\infty) = 1$-index

$$L_k(u, G) = \{ \underbrace{l_0, \dots, l_p}_{\text{legfeljebb } k \text{ hosszú}} \mid p < k \text{ és } u \mid l_0, \dots, l_p \}$$

Á: 1) $u \approx^k v \Rightarrow L_k(u, G) = L_k(v, G)$

2) $I \in A(k)$-index csúcsa, $u \in I$ $\Rightarrow L_k(I, A(k)) = L_k(u, G)$

3) $A(k)$-index pontos a legfeljebb k hosszú kifejezésekre (és biztonságos)

4) ~~$A(k)$ az $A(k-1)$ nem feltt~~

$A(k) \geq A(k-1)$ ($A(k)$ finomítása $A(k-1)$-nek, nem felt. valódi)

D: $P = P_0, P_1, \dots, P_k$

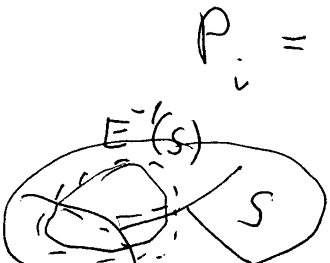
$P_{i+1} \geq P_i$ legdurvább olyan, ami stabil P_i-re

P_k a P legdurvább stabil finomításának k lépéses közelítése

Naiv-közelítés(V, E, P, k)

lényege $\forall S \in P_{i-1}$-re hasítunk:

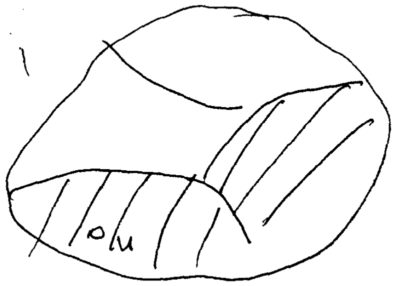
$$P_i = \text{hasít}(S, P_i)$$



$O(k|E|)$

Á: A(k) stabil A(k-1)-re a G^{-1}-ben

A(k)-indexes-kiértékelés (G, A_R, k)



A(k)

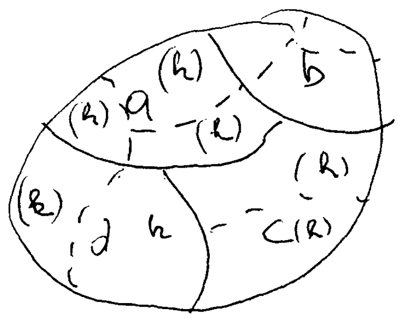
u ∉ L(R) kidobjuk

Indexes-kiértékelés $(G, A(k), A_R)$

Ha $|R| \le k$, akkor egyesítjük az osztályokat

$|R| \ge k$ esetén az eredmény csúcsait ellenőrizzük

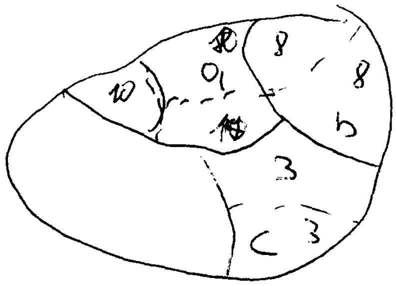
Dinamikus indexek (gyakori lekérdezések alapján)



A(k) ∀ csúcshoz k-hosszú utakat vizsgál

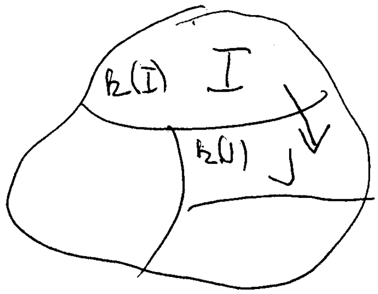
ÖTLET:

NEM FONTOS csúcsok esetén (ritkán szerepel lekérdezési eredményben) kisebb k is elég (lokális hasonlóság)



D(k)

D(k)-index



1) $k(I) \geq k(J) - 1$

2) I-ben a csúcsok $k(I)$ biszimulánsak

M: $A(k)$ is $D(k)$

Á: $I_1, I_2, \dots, I_s$ irányított út $D(k)$-ban

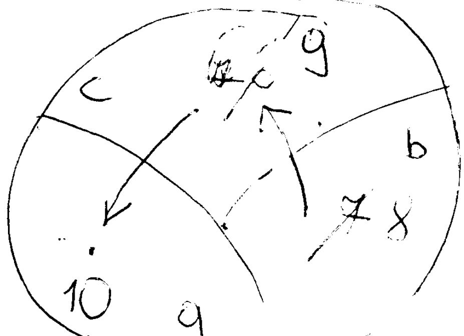
$k(I_j) \geq j-1$

($0 \quad \geq 1 \quad \geq s-2 \geq s-1$ biszimulánsak)

$\Rightarrow \forall u \mid$ címke$(I_1), \dots$ címke(I_s)

$\forall u \in I_s$ illeszkedik

K: $D(k)$ pontos $l_0, \dots, l_m$-re, ha minden $I \mid l_0, \dots, l_m$ esetén $k(I) \geq m$



Kezdeti súlyok: K_0

K= Súlymódosító(K_0)

Maximumból indulva

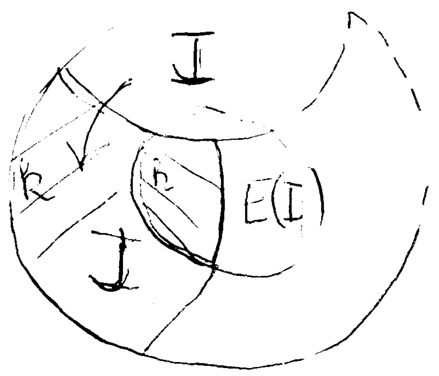
$I \to J$ esetén

$k(I) := \max(k(I), k(J)-1)$

$9 \quad = \max(6, 10-1)$

1) -t kielégíti

2) -t kisebbítésekkel érjük el



~~Ha~~

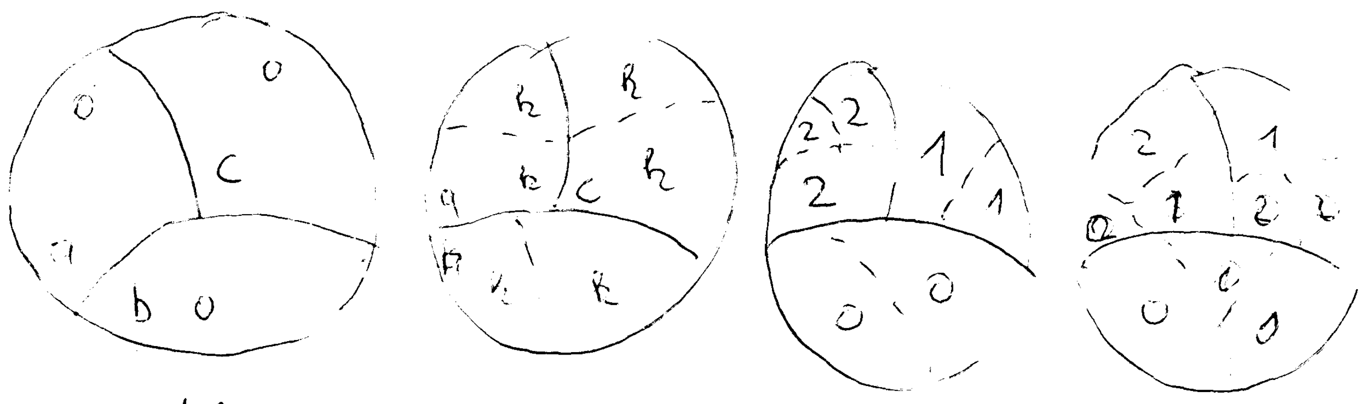
$i = 1 .. \max(K)$

$k(J) \geq i$

J helyett $E(I) \cap J$ és
$J - E(I)$

- súlyok: k, k
- újrarajzoljuk az éleket

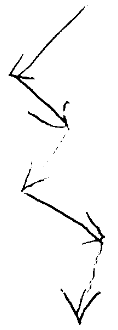
Minden hasítás 1-gyel növeli a kiszimulációs fokot ⇒ végén a <u>k</u> súlyú osztályok elemei k-kiszimulációsak lesznek $O(\max(K) \cdot |E|)$



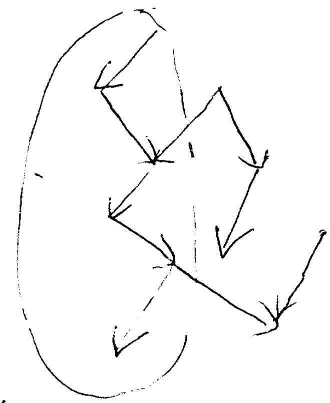
A(0) A(k) D(k) M(k)

GYAK

M(k) index olyan D(k) index, amely a GYAK={gyakori lekérdezések} -re nézve pontos



helyett

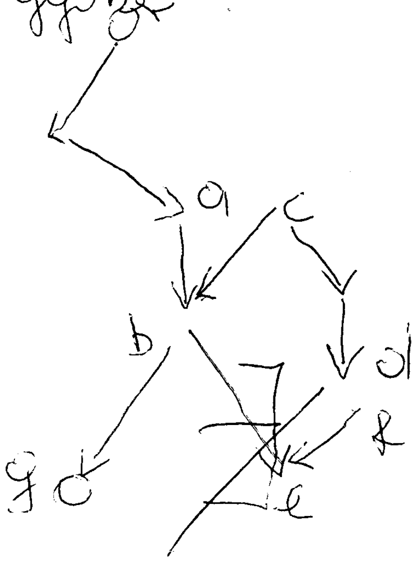


elágazó lekérdezések

főlekérdezés

Nyelvtan:

gyökér//a/b[\c//d and not \e/f]/g



Előd-stabil A a B-re

$A \subseteq E(B)$ vagy

$A \cap E(B) = \emptyset$

Utódstabil A a B-re

$A \subseteq E^{-1}(B)$ vagy

$A \cap E^{-1}(B) = \emptyset$

FB-index az alappartíció legdurvább olyan finomítása, amely egyszerre előd-stabil és utódstabil

FB-index készítés:

1-index → megfordítjuk az élek irányítását

amíg ~~kell~~ változik

(a kapott partíciót finomítjuk tovább)

Á: FB-index biztonságos és pontos az elágazó-lekérdezésekre $O(|E|\cdot|V|\cdot\log|V|)$

előre sorozat hossza $\leq f$
hátra –||– $\leq b$

d váltás (lekérdezés mélysége)

FB(f, b, d)-index pontos az ilyen lekérdezésekre

Spec	(∞,∞,∞)	FB
	$(\infty,0,1)$	1-index
	$(k,0,1)$	$A(k)$

Készítés:

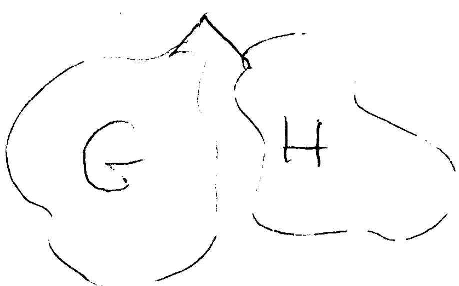
$\to$ $A(f)$-index $\longrightarrow$ megfordítjuk az irányítást

megfordítjuk az élek irányítását

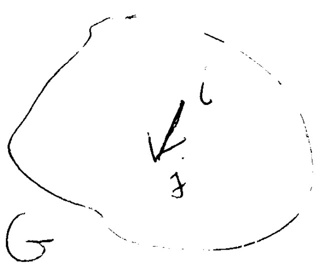
$A(b)$-index $\longleftarrow$ amíg váltás $\leq d$ [illegible]

M: Mindig a kapott partíciót finomítjuk tovább $O(d|E|\cdot\max(f,b))$

Indexek frissítése



$G+H$



(u,v) él

$G+$él

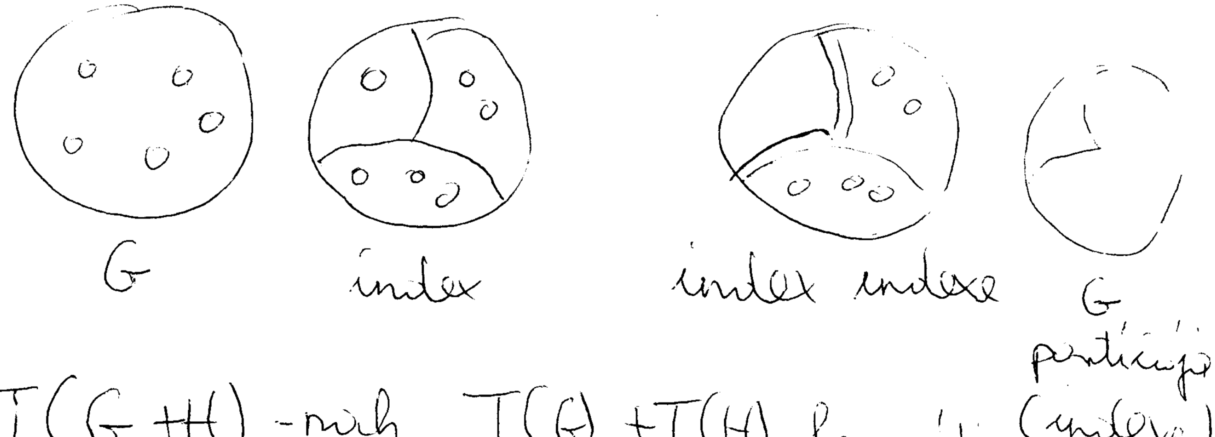
Hogy változnak az indexek?

A: $J \geq 1\text{-index}(G)$

$\Rightarrow 1\text{-index}(J) = 1\text{-index}(G)$

M: speciálisan $1\text{-index}(1\text{-index}(G)) = 1\text{-index}(G)$

Értelmezés:



M: $I(G+H)$-nak $I(G)+I(H)$ finomítása

így $I(I(G)+I(H)) = I(G+H)$

ahol I az 1-index

$G+(u,v)$:

Elromolhat a stabilitás

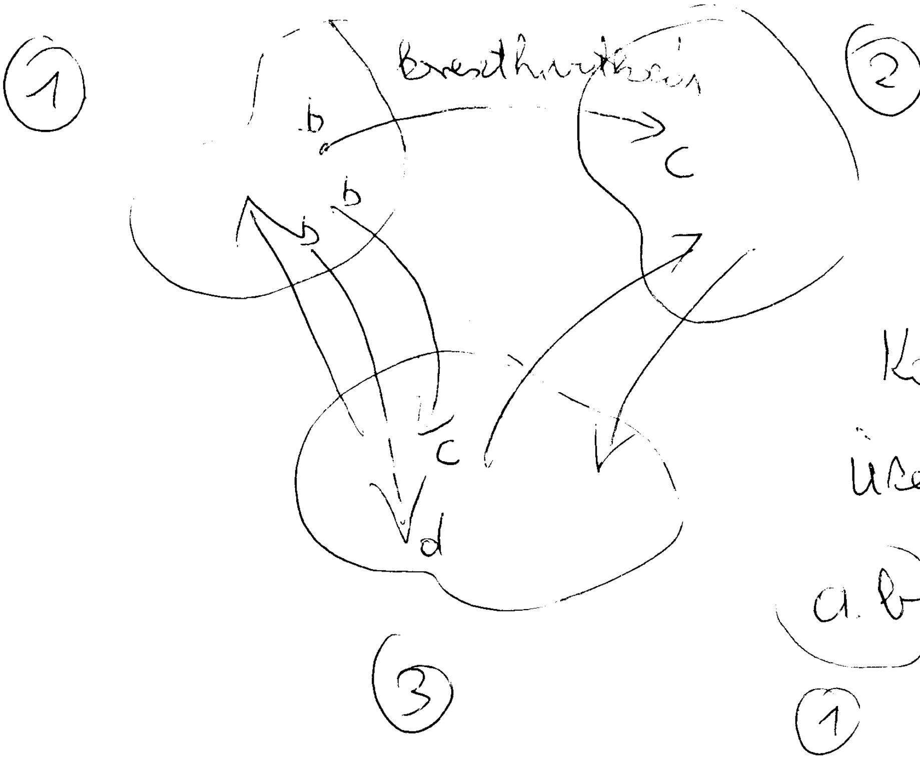
$I(u)$ osztályt

$I(v)$

PT-vel stabilizáljuk az őseikre visszamenőleg (nem kell az egész gráfot, csak $I(v)$ őseit)

Osztott adatbázisok

n db szerver ⟺ gráf n részes partíció-



Kommunikációs lépés üzenetszámással:

a. b. c

1

Hatékonyság:

a) – kommunikációs lépések száma konstans (független DB-től és R lekérdezéstől)

b) – átvitt adatok összmérete R(DB) méretétől és kereszthivatkozások számától függ (k)

A: reguláris lekérdezések hatékonyan kiértékelhetők:

a) 4

b) $O(|R(DB)|^2) + O(k)$