

1. Pretraživanje teksta u relacijskim sustavi za upravljanje bazama podataka

Pretraživanje teksta (Full Text Search - **FTS**)

- Traženje dokumenata koji zadovoljavaju postavljeni **uvjet** i njihovo rangiranje u skladu sa **sličnošću** dokumenta s postavljenim uvjetom.
- **Uvjet** je obično niz riječi, a **sličnost** je brojčana vrijednost (rang) koja je u najjednostavnijoj implementaciji povezana s frekvencijom riječi iz uvjeta u pretraživanom dokumentu.
- **Rang** – relevantnost dokumenata/znakovnog niza za postavljeni upit (pretpostavka je da je relevantniji dokument u kojem se traženi uzorak riječi pojavljuje na bliskim pozicijama), **ts_rank()** u `psql`

Pristupi pri pretraživanju teksta

- **Točno podudaranje**
 - uvjet i tekst se u cijelosti podudaraju, identični su
 - uvjet je sadržan u dijelu teksta (djelomično podudaranje)
 - =, LIKE, NOT LIKE, SIMILAR TO, regularni izrazi
- Podudaranje temeljom **morfologije, sintakse i semantike** jezika
 - Koriste se tzv. gramatički algoritmi koji pronalaze podudarnost između uvjeta i teksta temeljem **normaliziranog** oblika riječi (korijen, leksem), temeljem sinonima i itd.
 - TSVector, TSQuery tipovi podataka, @@ operator
- **Približno podudaranje (fuzzy)**
 - **Algoritmi temeljni** na „udaljenosti“ između znakovnih nizova – Levenshtein funkcija, Hamming
 - **Q-Gram algoritmi** - % operator, similarity funkcija
 - **Fonetsko podudaranje** – Soundex, Metaphone funkcije

Nedostaci standardnog relacijskog SUBP u pretraživanju teksta

- Ne postoji efikasni indeksi pa pri procesiranju svakog upita treba „obraditi“ svaki dokument – **sporo**
- Pronalaženje riječi koje ne želimo – **točnost**
- Ne postoji **lingvistička podrška** čak niti za engleski jeziknema načina da se
 - prepoznaju **izvedene** riječi, npr. podatak i podatkovni
 - nemoguće je zanemariti veznike, prijedloge i ostale riječi koje se često pojavljuju a nemaju semantičko značenje
 - nemoguće je dovesti u vezu riječi istog značenja (**sinonimi**)
- Ne postoji **rangiranje** rezultata što rezultat s mnogo dokumenata čini neupotrebljivim

Pretraživanje teksta temeljem **morfologije, sintakse i semantike** jezika

- Tekst/dokument je (multi) skup riječi (bag of words)
- **Efikasna pretraga** teksta podrazumijeva prethodnu obradu teksta (dokumenta):
 - **parsiranje** teksta i rastavljanje na tokene (riječi, brojevi, tagovi, razmak, url,...)
 - uklanjanje riječi koje nemaju semantičko značenje u tekstu (**stop** riječi)
 - utvrđivanje **korijena** za svaki token
 - identificiranje **sinonima**
 - pohrana obrđenog teksta u oblik prikladan za pretragu

TSVector – podatkovni tip za reprezentaciju dokumenta

TSQuery – podatkovni tip za reprezentaciju tekstualnog upita

@@ - FTS operator za rad sa TSVector i TSQuery

Parser – prvi korak u FTS temeljem morfologije, sintakse i semantike

- razdvaja izvorni tekst na **tokene** i utvrđuje tip tokena
- ne modificira izvorni tekst
- tipovi tokena koje parser prepoznaje su unaprijed definirani

Riječnici – ulaz je tokeniziran tekst koji je dao parser (Simple Dictionary, Synonym Dictionary, Snowball Dictionary, ...)

- definiraju stop riječi, sinonime, stvaranje veza između fraza i pojedinih riječi
- **normaliziraju** tekst radi prepoznavanja različitih riječi jednakog značenja
- **reduciraju** veličinu reprezentacije dokumenta - TSVector
- vraća polje leksema ako prepozna token, prazno polje ako je stop riječ i riječnik prepozna leksem, u protivnom NULL
- mogu se **ulančavati**, ako riječnik vrati NULL token se proslijeđuje sljedećem riječniku (obično se na početak liste stavlja najspecifičniji)

Brzina FTS-a temeljem morfologije, sintakse i semantike (**kvalitetna priprema za efikasno pretraživanje FTS**)

- ubrzava se spremanjem TSVectora u bazu (caching), ujedno i reprezentacija teksta koja je manja od originalnog (uklanjanje stop riječi)
- izgradnja **indeksa**
 - **GIN** (generalized inverted index) – atribut mora biti tip TSVector (brža pretraga ali dulja izgradnja, sporiji UPDATE i zauzima više mjesta od GIST-a)

CREATE INDEX idxName ON tableName USING gin(attrName)

- **GIST** (generalized search tree) – atribut mora biti tipa TSVector ili TSQuery

CREATE INDEX idxName ON tableName USING gist(attrName)

Približno pretraživanje teksta (**Fuzzy text search**)

- Tehnika pronalaženja dokumenta/niza znakova koji se približno podudara s traženim uzorkom
- Kvaliteta podudaranja se mjeri ovisno o vrsti primijenjenog algoritma

1. Algoritmi temeljeni na **udaljenosti** znakovnih nizova

- Udaljenost uređivanja (**edit distance**) znakovnih nizova s_1 i s_2 je minimaln broj operacija potrebnih da se jedan niz transformira u drugi. Moguće operacije su: izmjena, umetanje i brisanje znaka.
- Autor ideje Vladimir Levenshtein 1965. – **Levenshteinova** udaljenost
- Riješenje mora biti jednoznačno

$$d(\text{sreća, sretna}) = 2$$

2. **Q-Gram** algoritmi

- Tekst/dokument je (multi) skup q-grama
- Ako je riječ A slična riječi B, one vjerojatno sadrže barem jedan podudaran (zajednički) podniz duljine Q
- Manji q znači veća senzitivnosti – q-gram algoritam ne radi uvijek dobro ovisno o vrijednosti q (prevelika ili premala vrijednost q)
- Prikladan i za veća odstupanja

Primjer: $q = 2$ (bigrami)

happiness: haappppiin ne es ss

happily: haappppiil ly

- 4 podudarana od ukupno 10 bigrama (bez ponavljanja podudaranih)
- mjera sličnosti bi se mogla odrediti kao kvocijent: $\frac{4}{10} = 0,4$

- **PostgreSQL** relalizira q-gram kao trigrame ($q = 3$) i dodaje 2 praznine na početak i 1 prazninu na kraj svakog niza

Primjer:

happiness	<div>__h</div>	<div>_ ha</div>	<div>hap</div>	<div>app</div>	<div>ppi</div>	pin	ine	nes	ess	ss_
happily	<div>__h</div>	<div>_ ha</div>	<div>hap</div>	<div>app</div>	<div>ppi</div>	pil	ily	ly_		

- 5 podudarnih od ukupno 13 trigrama (bez ponavljanja podudarnih)
- Sličnost: $\frac{5}{13} = 0,384615$
- Može biti jako sporo za puno velikih dokumenata – mogućnost kreiranja specijalnog (invertiranog) indeksa

CREATE INDEX title_trigram_idx **ON** movies **USING gist**(title gist_trgm_ops)

Fonetsko podudaranje – Soundex i Metaphone

- Akutalno u jezicima u kojima se izgovor riječi razlikuje od zapisa
- Ideja je dovesti u vezu riječi koje se jednako ili slično izgovaraju ali drugačije zapisuju
- Algoritam riječ predstavlja znakovnim nizom koji prezentira izgovor (zvučenje) riječi
 - **Soundex** je ograničen na vlastita imena u engleskom jeziku
 - **Metaphone** koristi ideju algoritma Soundex ali nije ograničen na vlastita imena

2. Napredni SQL

Prozor – tranzijentni skup n-torki pomoću kojeg se definiraju

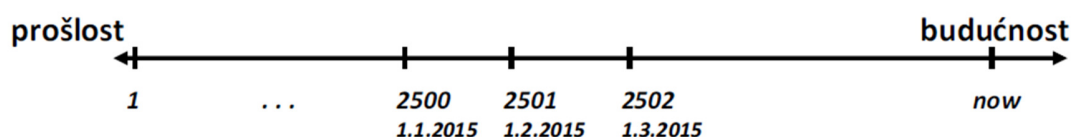
- **Particija** (partition)
 - Definira se pomoću **PARTITION BY** dijela (bez PARTITION BY cijela relacija je jedna particija)
 - Ne može biti pomičan
 - Može sadržavati okvir(e)
- **Okvir** (frame)
 - Definira se pomoću frame i **ORDER BY** dijelova – uvijek relativno u odnosu na tekuću n-torku
 - Pomiće se unutar particije
 - Ne može zahvatiti više particija
 - Dvije vrste okvira postoje: **ROWS** i **RANGE** okvir
- **ROWS** okvir
 - n-torka pripada okviru tekuće n-torke ako je u specificiranom rasponu – navodi se broj n-torki koje prethode ili slijede tekuću (PRECEDING, FOLLOWING, ...)
 - poredak ne mora biti definiran (bez ORDER BY)
- **RANGE** okvir
 - n-torka pripada okviru tekuće n-torke ako je vrijednost odgovarajućih atributa (navedenih u ORDER BY) u specificiranom rasponu vrijednosti
 - važan je poredak n-torki u particiji (ORDER BY) jer on definira n-torke koje prethode/slijede tekuću

4. Temporalne baze podataka

Vrijeme se uglavnom shvaća kao jednodimenzionalni kontinuum koji se prostire od prošlosti prema budućnosti.

Vrijeme se modelirati (definira) kao:

- **konačno/beskonačno**
 - Početak – početak vremenske linije (PostgreSQL – 4713 p.n.e.)
 - Zauvijek – kraj vremenske linije (PostgreSQL – 294276 n.e.)
- **diskretno/kontinuirano**
 - Diskretni modeli izomorfni su prirodnim ili cijelim brojevima (kalendarske vrijednosti preslikavamo u cjelobrojne konstante)



- ♦ Svaki prirodan ili cijeli broj odgovara osnovnoj jedinici vremena – **chrononu** (kvantum vremena, diskretna i nedjeljiva jedinica vremena kao dio hipoteze da vrijeme nije kontinuirano)
- ♦ **Chronone** možemo grupirati u veće jedinice vremena (sati, dani ...)
- Kontinuirani modeli izomorfni su realnim brojevima
Svaki realan broj odgovara jednom vremenskom trenutku
- **apsolutno/relativno**
 - 31. listopada 2015. 9:15
 - Dva tjedna, 5 sati

Fenomeni iz stvarnog svijeta se

1. dogode u određenom trenutku (*chrononu*) i nemaju trajanje ili
2. traju/istinite su u nekom vremenskom intervalu ili periodu

Fenomeni iz stvarnog svijeta čija su nam vremenska svojstva zanimljiva mogu se opisati kao **stanja** ili kao **dogadjaji**

- **Stanja** opisuju činjenice vezane uz neki objekt u bazi podataka koje su istinite u nekom vremenskom intervalu ili periodu. Te se činjenice ne smatraju tačnima izvan pridruženog perioda.
- **Događaji** opisuju činjenice vezane uz neki objekt u bazi podataka koje su se dogodile u određenom trenutku (*chrononu*) i nemaju trajanje.

tekRacunLimit		stanje		uplataIsplata			dogadaj	
brTekRacun	iznosLimit	vrijediOd	vrijediDo	brTekRacun	iznos	datum		
2341906787	5000	1.1.2015.	1.5.2015.	2341906787	5000.00	10.06.2015.		
2341906787	8000	2.6.2015.	NULL	2341906787	-3500.00	12.06.2015.		
2211906783	10000	1.1.2005.	NULL	2192345667	-1000.00	01.07.2015.		
2192345667	9000	1.1.2015.	1.7.2015.	2192345667	-5000	20.05.2015.		
2192345667	5000	2.7.2015.	NULL					
5361906785	7000	20.5.2015.	NULL					

Osnovni vremenski tipovi podataka:

- **instant**
Određeni chronon na vremenskoj liniji diskretnog modela ili točka na vremenskoj liniji kontinuiranog modela (npr. 12. studeni 2015, 8:50:59)
- **interval**
Neusidreni interval na vremenskoj liniji, ima samo trajanje (npr. 2 sata, 3 dana,...)
- **period**
Usidreni (apsolutno definirani) interval na vremenskoj liniji (npr. zimski semestar 2015/16: 05.10.2015 – 29.1.2016)
- **periods**
Skup disjunktivnih usidrenih intervala, naziva se još i vremenskim elementom (engl. temporal element)

DateRange tip (uvijek) koristi oblik koji uključuje donju i isključuje gornju granicu: [] (tzv. **closed-open** notaciju)

U kontekstu temporalnih baza podataka značajne su dvije ortogonalne (međusobno nezavisne) **dimenzije** vremena:

- **Vrijeme valjanosti (valid time)**
 - Vrijeme u stvarnom svijetu kada se neki događaj dogodio ili period u kojem neka činjenica važeća, nezavisno od trenutka kada je informacija o tom događaju/činjenici zapisana u bazu podataka
- **Transakcijsko vrijeme (transaction time ili system time)**
 - Vrijeme kada je određena promjena zabilježena u bazi podataka ili vremenski interval tijekom kojeg se baza podataka nalazi u određenom stanju

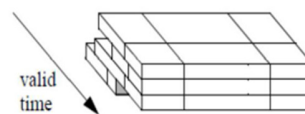
S obzirom na sposobnost upravljanja vremenom valjanosti i transakcijskim vremenom, razlikujemo četiri vrste relacija:

- Trenutačne relacije (snapshot tables)
- Relacije vremena valjanosti (application-time period tables)
- Relacije transakcijskog vremena (system versioned tables)
- Bitemporalne relacije (system-versioned application-time period tables)

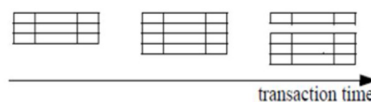
Trenutačne relacije (snapshot)



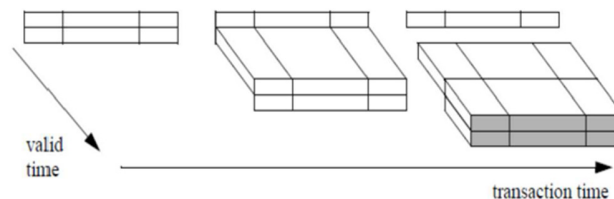
Relacije vremena valjanosti



Relacije transakcijskog vremena



Bitemporalne relacije



Prednosti relacija vremena valjanosti

- Većina poslovnih podataka je podložna promjenama tj. zahtijeva praćenje perioda u kojem je određena vrijednost valjana
- Današnji SUBP uglavnom ne pružaju potporu za:
 - Definiranje perioda valjanosti za n-torku
 - Definiranje ograničenja tipa „osoba može prebivati samo u jednom mjestu u određenom vremenskom periodu”
 - UPDATE/DELETE n-torke za dio perioda valjanosti
- Trenutno, najčešće aplikacije pružaju podršku za gornje zahtjeve
- Glavni problemi:
 - Kompleksnost programskog koda
 - Loše performance
- Relacije vremena valjanosti omogućavaju:
 - Pojednostavljenje programskog koda
 - Poboľjšane performance
 - Transparentne su u odnosu na naslijeđene aplikacije

Tri osnovne vrste upita (u sustavima koji upravljaju vremenom)

- **trenutni** upiti – gledaju kakvo je trenutno stanje tj. stanje u trenutku pokretanja upita
- **slijedni** upiti – rade s podacima iz niza vremenskih trenutaka tj. iz različitih trenutaka u prošlosti, gledaju što se dogodilo i kada
- **neslijedni** upiti – koriste podatke iz niza vremenskih trenutaka tj. iz različitih trenutaka u prošlosti, gledaju da li se nešto uopće dogodilo, ne zanima ih kada

5. Geoprostorne baze podataka

GIS (Geoinformacijski sustav) – formalna definicija:

Informacijski sustav za upravljanje, analizu, vizualiziranje i distribuiranje informacija o objektima i pojavama, čiji referentni sustav je definiran na površini Zemlje.

Marble & Peuquet - četiri podsustava GIS-a:

- **Sustava za unos podataka** skuplja i/ili procesira geoprostorne podatke prikupljene iz postojećih mapa, senzora, itd.
- **Sustav za pohranjivanje i dohvat podataka** organizira i pohranjuje podatke u obliku koji omogućuje brz dohvat za buduće analize, kao i brza i točna ažuriranja podataka u geoprostornoj bazi podataka
- **Sustav za rukovanje i analizu podataka** koji obavlja razne zadaće kao npr. Promjenu formata podataka koristeći korisnički definirana agregacijska pravila ili procjena parametara različitih geoprostornih simulacija i sl.
- **Sustav za izvještavanje** koji može prikazati cijelu ili dio baze podataka, ako i upravljati podatcima i ostvariti izlaz bilo u tabličnom ili kartografskom obliku.

GIS i povezne tehnologije mogu otvoriti na **5 tipova geografskih upita**:

- Što se nalazi na nekoj lokaciji?
- Gdje je nešto?
- Što se promijenilo od...?
- Koji geoprostorni obrasci postoje?
 - Koji su odnosi između dva ili više skupa podataka koji se odnose na istu lokaciju?
 - Koje geografske varijacije postoje s obzirom na prostor?
- Što ako...? "What if?"

SUGP – programski modul:

- Implementiran proširenjem objektno-relacijskog
- Posjeduje skup geoprostornih apstraktnih tipova podataka (GeoATP) kao i upitni jezik koji podržava te tipove podataka i operacije nad njima
- Prostorno indeksiranje
- Djelotvorni algoritmi za operacije nad geoprostornim tipovima podataka
- Specifična pravila za optimiranje upita

Topologija se definira kao geoprostorni odnosi između susjednih objekata. Topologija govori gdje su objekti s obzirom na jedan drugoga, te kako se odnose. Ti odnosi mogu biti jednostavni (npr. udaljenost), ali uključuju i složenije pojmove kao što su susjednost i povezanost.


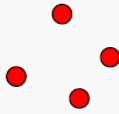
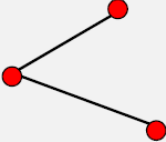
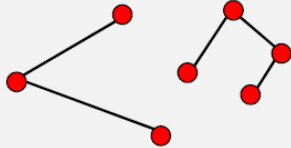
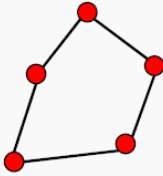
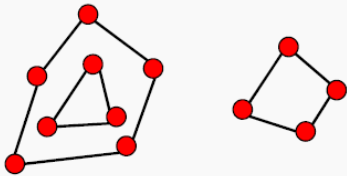
Temeljna obilježja geoprostornih objekata:

- Geometrijska
 - Metrička
 - ◆ Oblik (apstrakcija geometrijske strukture: točka, linija, poligon)
 - ◆ Položaj (u odnosu na referentni koordinatni sustav)
 - ◆ Veličina (0D, 1D, 2D ili 3D)
 - Topološka
 - ◆ Relacije među objektima (susjedstvo, povezanost i sl.)
- Tematska
 - ◆ Atributi čije su domene jednostavni tipovi podataka (integer, char ...)

PostGIS tipovi podataka:

- **Geometry:** puno bogatiji skup funkcija, provjerava odnosa su u pravilu brža, bolja podrška u postojećim alatima (s ovime dalje radimo)
- **Geography:** jednostavna mjerenja i odnosi na relativno velikom području
- **Raster**
- **Topology**

PostGIS podtipovi podataka:

Point 	Multipoint 
Linestring 	Multilinestring 
Polygon 	Multipolygon 

GeoATP (Geoprostorni apstraktni tipovi podataka)

- Operacije: funkcije čiji su argumenti geoprostorni tipovi podataka, a rezultat je ili geoprostorni tip ili prosti tip (skalarna vrijednost)
- Radimo sa 2D prostorom – mali skup operacija, izražajnost, konzistentnost, jezična I spoznajna temeljitost

GeoATP – Operacija:

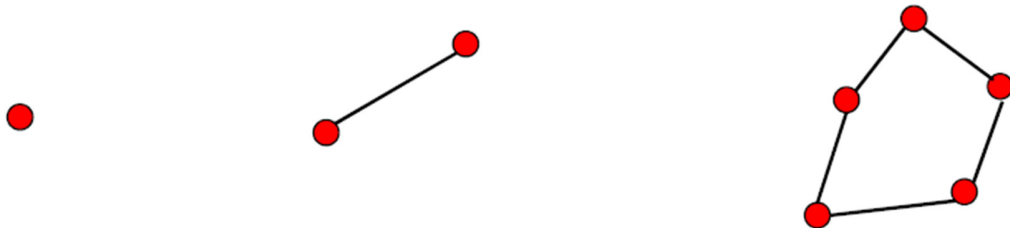
- **Geometrijske operacije**
 - Skupovne (unija, presjek)
 - Aritmetičke (duljina krivulja, površina poligona)
 - Druge (buffer, convex hull)
- **Topološke relacije** (touches, within, disjointed ...)
- **Operacije nad grafovima** (traženje najkraćeg puta)

GeoATP – topološke relacije (Model 9 presjeka - 9IM)

- Binarna topološka relacija **R** između dva prostorna objekta **A** i **B** opisuje se usporedbom unutrašnjosti (A^0), granice (∂A) i vanjštine (A^-) dvaju objekata.
- Tih 6 komponenata moguće je kombinirati tako da oblikuju 9 temeljnih vrijednosti (presjeka) za opis topoloških relacija
- Svaki presjek može poprimiti vrijednosti \emptyset (prazan skup) i $\neg\emptyset$ (ne prazan skup)
- Uređeni skup 9 presjeka može se prikazati matricom:

$$R(A,B) = \begin{pmatrix} A^0 \cap B^0 & A^0 \cap \partial B & A^0 \cap B^- \\ \partial A \cap B^0 & \partial A \cap \partial B & \partial A \cap B^- \\ A^- \cap B^0 & A^- \cap \partial B & A^- \cap B^- \end{pmatrix}$$

Osnovni tipovi podataka: **točka**, **linija**, **poligon**:



Unutrašnjost granica i vanjština:

- Unutrašnjost i granica **poligona** su jasne
- Granica linije su **točke** na njenim krajevima dok je unutrašnjost sve ostalo
- **Točka** nema granicu već samo unutrašnjost i vanjštinu

DE-9IM – Dimenzijski prošireni model 9 presjeka (9IM)

- Pored operatora unutrašnjosti ($^{\circ}$), granice (∂) i vanjštine ($-$), uvodi se i operator dimenzije

$$\dim(s) = \begin{cases} - \text{ ako je } S = \emptyset \\ 0 \text{ ako } S \text{ sadrži barem točku, ali ne i linije i površine} \\ 1 \text{ ako } S \text{ sadrži barem liniju, ali ne površinu} \\ 2 \text{ ako } S \text{ sadrži barem površinu} \end{cases}$$

*S – opći skup točaka

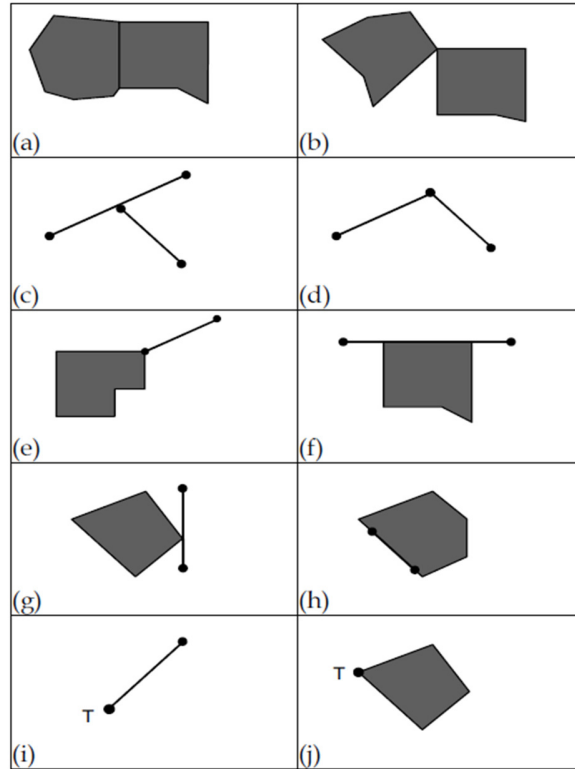
- Svaki element matrice proširuje se dimenzijom
 - Novu matricu moguće je zapisati ovako:

$$DE9I = \begin{pmatrix} \dim(\partial\lambda_1 \cap \partial\lambda_2) & \dim(\partial\lambda_1 \cap \lambda_2^0) & \dim(\partial\lambda_1 \cap \lambda_2^-) \\ \dim(\lambda_1^0 \cap \partial\lambda_2) & \dim(\lambda_1^0 \cap \lambda_2^0) & \dim(\lambda_1^0 \cap \lambda_2^-) \\ \dim(\lambda_1^- \cap \partial\lambda_2) & \dim(\lambda_1^- \cap \lambda_2^0) & \dim(\lambda_1^- \cap \lambda_2^-) \end{pmatrix}$$

Na temelju dimenzijski proširenog modela 9 presjeka, definiraju se topološke relacije:

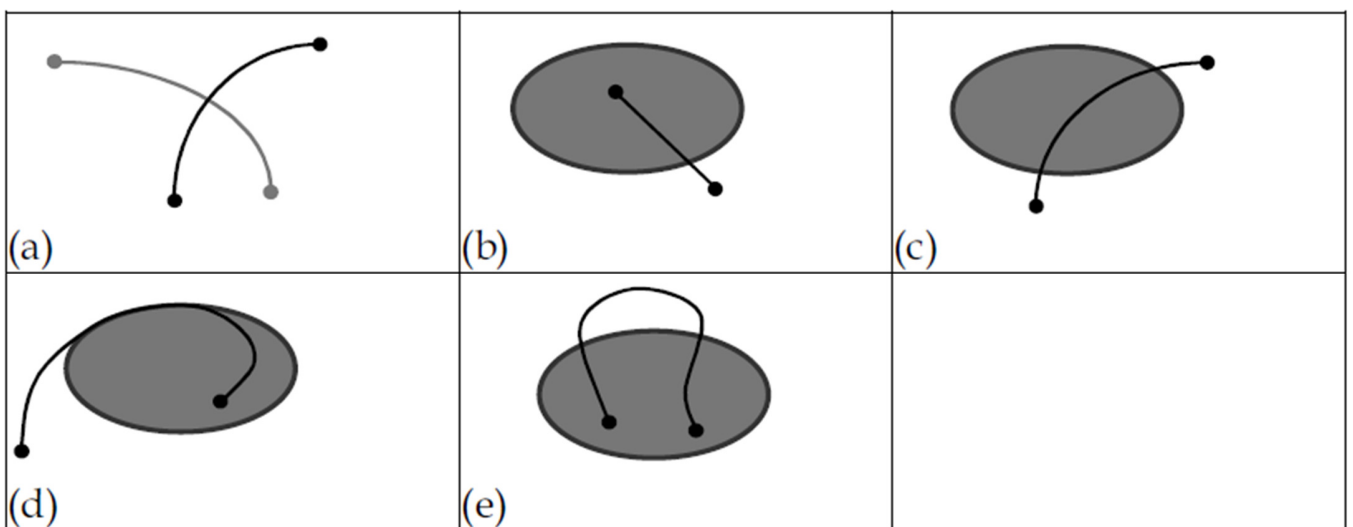
- **Touch** – relacija dodirivanja (vrijedi za parove objekata poligon/polygon, poligon/linija, linija/linija, točka/polygon i točka/linija – sve osim točka/točka)

$$\langle \lambda_1, touch, \lambda_2 \rangle \Leftrightarrow (\lambda_1^0 \cap \lambda_2^0 = \emptyset) \wedge (\lambda_1 \cap \lambda_2 \neq \emptyset)$$



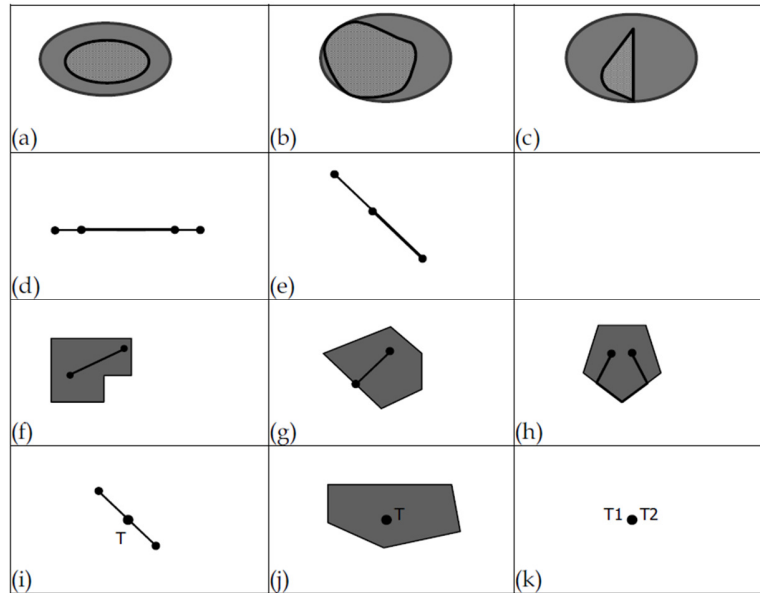
- **Cross** – relacija presijecanja (vrijedi za parove linija/linija i linija/poligon)

$$\langle \lambda_1, cross, \lambda_2 \rangle \Leftrightarrow (\dim(\lambda_1^0 \cap \lambda_2^0) = \max(\dim(\lambda_1^0), \dim(\lambda_2^0)) - 1) \wedge (\lambda_1 \cap \lambda_2 \neq \lambda_1) \wedge (\lambda_1 \cap \lambda_2 \neq \lambda_2)$$



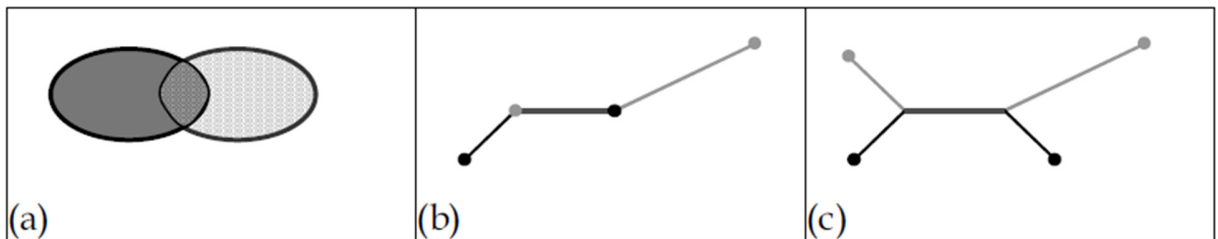
- **In** – relacija pripadanja (vrijedi za sve kombinacije objekata)

$$\langle \lambda_1, in, \lambda_2 \rangle \Leftrightarrow (\lambda_1 \cap \lambda_2 = \lambda_1) \wedge (\lambda_1^0 \cap \lambda_2^0 \neq \emptyset)$$



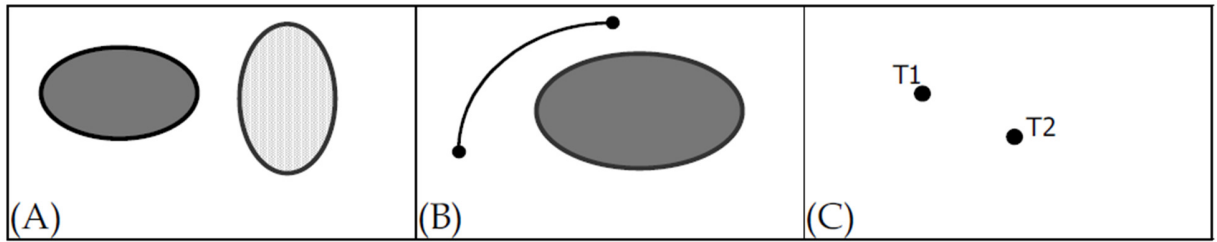
- **Overlap** – relacija preklapanja (vrijedi za homogene parove poligon/poligon i linija/linija)

$$\langle \lambda_1, overlap, \lambda_2 \rangle \Leftrightarrow (\dim(\lambda_1^0) = \dim(\lambda_2^0) = \dim(\lambda_1^0 \cap \lambda_2^0)) \wedge (\lambda_1 \cap \lambda_2 \neq \lambda_1) \wedge (\lambda_1 \cap \lambda_2 \neq \lambda_2)$$

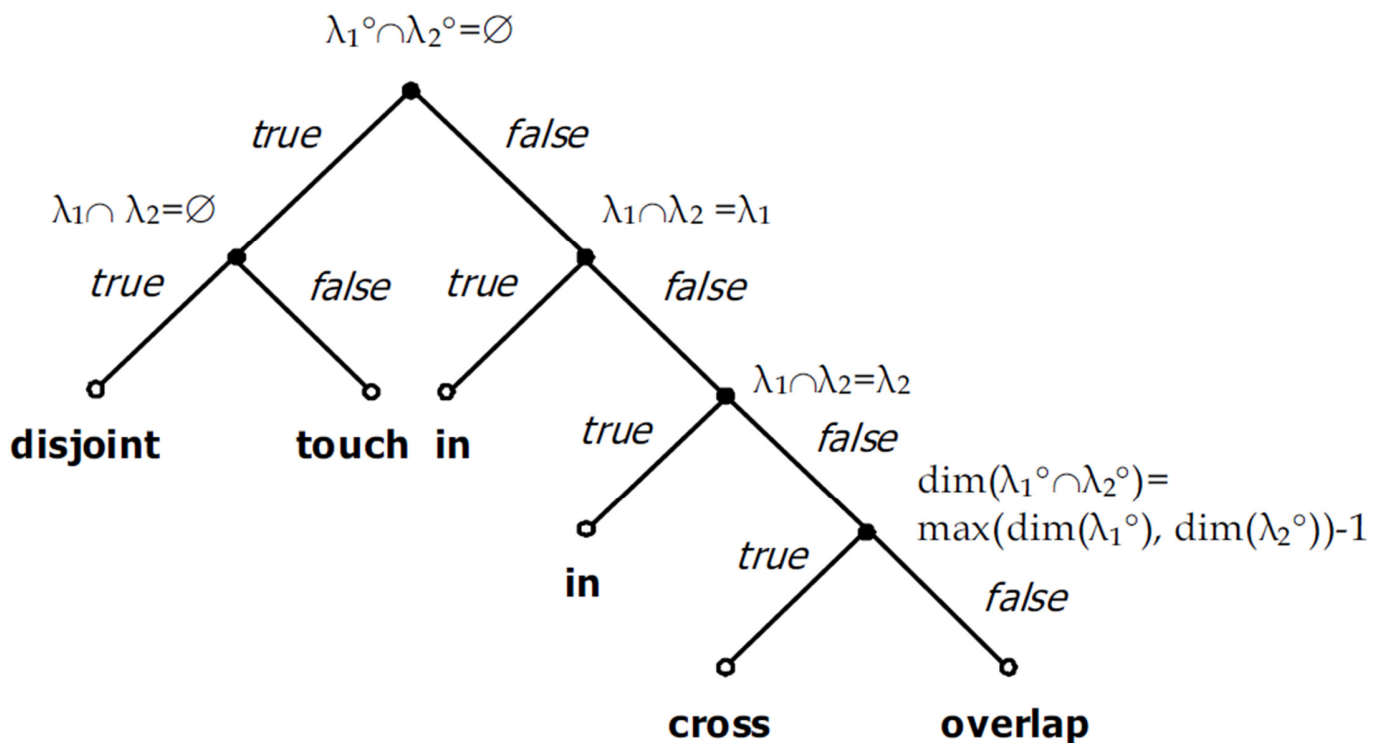


- **Disjoint** – relacija odvojenosti (vrijedi za sve kombinacije objekata)

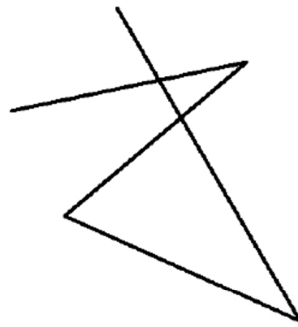
$$\langle \lambda_1, \text{disjoint}, \lambda_2 \rangle \Leftrightarrow \lambda_1 \cap \lambda_2 = \emptyset$$



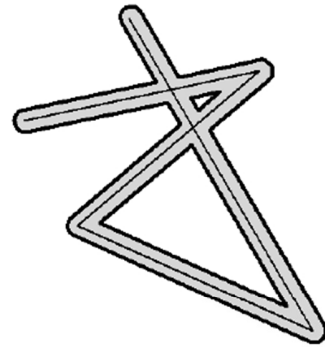
Stablo odlučivanja - međusobna **isključivost** i **zatvorenost** skupa topoloških relacija (sve gornje relacije se daju izvesti iz ovog stabla)



Buffer



(a)

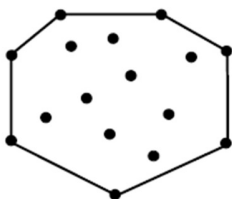


(b)

$$\text{buffer}(\delta) = \begin{cases} \delta > 0 : \{x \in \mathbb{R}^2 \mid d(x, g) \leq \delta\} \\ \delta < 0 : \{x \in \mathbb{R}^2 \mid x \in g \wedge d(x, \partial g) > \delta\} \end{cases}$$

Konveksna ljuska (**convex hull**)

- Konveksna ljuska skupa točaka **S** jest najmanji konveksni skup točaka (poligon) za koji je svaka točka skupa **S** ili na granici ili u unutrašnjosti tog poligona
- Konveksni skup točaka (poligon) – linija povučena između bilo koje dvije točke skupa u potpunosti se nalazi u tom skupu



(a)

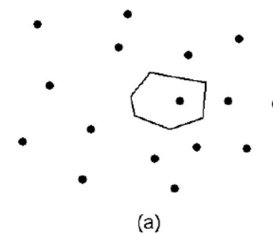


(b)

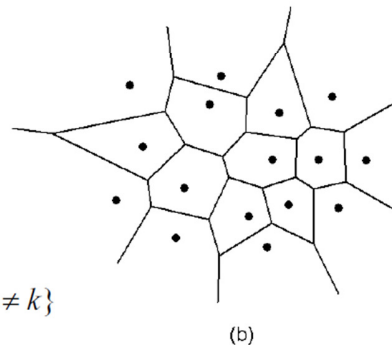


(c)

- Voronoijev dijagram (***voronoi***)
- Geoprostorni upit korisnika
 - Odredi područja u Hrvatskoj koja gravitiraju pojedinim gradovima sa više od 50000 stanovnika



- Geometrijska operacija
 - Za svaku točku ravnine odrediti koja joj je od ***N*** točaka iz skupa ***S*** najbliža



$$\text{voronoi}(t_k) \Leftrightarrow \{x \in \mathbb{R}^2 \mid d(t_k, x) < d(t_i, x), \forall i \neq k\}$$

Indeksi prostornih podataka (R, R*, R+ i Hilbertovo R-stablo)

R stablo

- Slično B stablima (svaki čvor može sadržavati više elemenata)
- Indeksira se položaj objekta u bazi (koordinata)
- Dijeli prostor na MBR (minimal bounding box)
- Elementi unutar čvora sadrže pokazivače na svoju djecu te MBR unutar kojeg se njegova djeca nalaze (listovi sadrže id objekta i MBR unutar kojeg se nalaze)

