

# Paradigma upitnih jezika

Luka Hadži-Đokić  
mr16269@alas.matf.bg.ac.rs

1. novembar 2018.

## 1 Uvod

Upitni jezici služe za potraživanje informacija i/ili podataka iz nekog informacionog sistema uz pomoć upita (eng. *query*). Upiti, kao njihova glavna sintaksna jedinica, kreću se od prirodnog jezika do posebno i precizno formuliranih rečenica kontekstno-slobodne gramatike, u zavisnosti od strukture sistema koji se pretražuje i potreba i očekivanja njegovog korisnika. Zbog toga se obično dele<sup>1</sup> na upitne jezike baza podataka (eng. *database query language*) i upitne jezike za pronalaženje informacija (eng. *information retrieval query language*). Obe kategorije su većim delom deklarativne, mada neki upitni jezici baza podataka podržavaju i proceduralni i deklarativni stil, kao što je Gremlin [3] za grafovske baze podataka.

Kada posmatramo problem koji ovakvi jezici rešavaju, jasno je i zašto je to slučaj. Naime, kada pokušavamo da pronađemo određenu informaciju u nekom informacionom sistemu, bio on obična baza podataka ili ceo indeksiran veb [2], dobro znamo kako rešenje treba da izgleda. Sa druge strane, nismo uvek upoznati sa detaljima implementacije samog sistema, pa nam proceduralno pisanje algoritma u najgorem slučaju nije izvodljivo, a u opštem slučaju ne donosi željene rezultate sa pogleda brzine izvršavanja programa. Zato deklarativna paradigma, koja nam omogućava da detalje algoritma pretrage ostavimo projektantima sistema, dominira kako upitnim jezicima baza podataka, tako i upitnim jezicima za pronalaženje informacija.

## 2 Upitni jezici baza podataka

Podela iz odeljka 1 je zasnovana na nivou strukturisanosti sistema nad kojim radi upitni jezik. Sa jedne strane te podele nalaze se čvrsto povezani i dobro strukturisani sistemi kao što su baze podataka. Glavni predstavnik je SQL, koji je dizajniran po uzoru na relacioni model Edgara Koda [1]. Jedan primer SQL upita može izgledati ovako:

---

```
SELECT naziv, sifra, espb  
FROM predmeti  
WHERE espb > 6  
ORDER BY sifra;
```

---

<sup>1</sup>Po pomenutoj strukturi sistema koji se pretražuje.

Ovaj SQL upit se odnosi na unapred definisanu bazu podataka koja sadrži tabelu “predmeti”. Relacione baze podataka možemo zamišljati kao skup tabela. Ova tabela ima kolone “naziv”, “sifra”, “espb”, i možda (verovatno) još neke. Jasno je šta taj upit postiže - vraća nam informacije o predmetima sa više od 6 ESPB, sortirane po polju “sifra”.

SQL je, kao i većina ostalih upitnih jezika, primer domenskog ili oblasno-specifičnog jezika (eng. *domain-specific language*). Koristi se i namenjen je za pretraživanje, menjanje i kreiranje baza podataka. Za razliku od programskih jezika opšte primene, od kojih očekujemo da budu primenjivi u velikom broju različitih situacija, domenski ne moraju da budu Tjuring-kompletni (eng. *Turing-complete*), i često nisu<sup>2</sup>. Ipak, SQL jeste Tjuring-kompletna, i ovo svojstvo dokazano je za SQL implementiranjem Cikličnog tag sistema (eng. *Cyclic Tag System*) [4]. To znači da je moguće u SQL-u implementirati “bilo koji algoritam”, kao na primer vizuelizaciju Mandelbrotovog skupa [5]. Stoga, vidimo da Tjuring-kompletnost nije dovoljna kako bi neki jezik mogli da nazovemo jezikom za opštu upotrebu. Granica između domenskih jezika i jezika opšte upotrebe nije u potpunosti jasna, ali za pojedinačne slučajeve možemo lako da odredimo kojoj grupi pripadaju - SQL je domenski jezik.

Ostali upitni jezici baza podataka mogu se razvrstati u nekoliko grupa. Datalog je, kao podskup Prologa, predstavnik upitnih jezika za deduktivne baze podataka. Za grafovske baze podataka koriste se GraphQL i Gremlin. Korisnici velikog broja ostalih nerelacionih (ili NoSQL) baza podataka koriste MapReduce model ili REST API bez upotrebe posebnih upitnih jezika.

### 3 Upitni jezici za pronalaženje informacija

Upitni jezici za pronalaženje informacija koriste se u raznovrsnim sredinama gde je struktura podataka manje jasna. Radi se o pretraživanju biblioteke, veba ili, malo opštije, kolekcije (tekstualnih) datoteka. Upitnim jezicima za pronalaženje informacija (često) ne tražimo egzaktno rešenje, već sve dokumente slične onom koji nam je potreban. Rešenje je, zato, najčešće u obliku niza referenci na relevantne datoteke. Izračunavanje ili odlučivanje relevantnosti je glavni problem upitnih jezika za pronalaženje informacija [7].

Ovi jezici se udaljavaju od kontekstno slobodne gramatike u korist prirodnog jezika<sup>3</sup>. Upiti se interpretiraju na nekoliko različitih načina. Najjednostavniji od njih je pretraga po ključnim rečima - upit se posmatra kao niz reči, a relevantnost se određuje po broju ključnih ili njima semantički sličnih reči koje se nalaze u nekom dokumentu. Malo bolje strukturisane ili indeksirane datoteke omogućavaju dodatnu preciznost (npr. pretraga knjiga po godini izdavanja). Sa druge strane, postoje sistemi koji omogućavaju upite na prirodnom jeziku i pokušavaju da što bolje “razumeju” upit pomoću tehnika obrade prirodnog jezika (eng. *Natural language processing*), dok relevantnost računaju mašinskim učenjem i drugim probabilističkim metodama [2].

CQL (eng. *Contextual Query Language*) je jedan od glavnih predstavnika. Po svojoj sintaksi i semantici nastao je kao spoj kompleksnih i moćnih jezika iz grupe upitnih jezika baza podataka kao što su SQL i XQuery i jednostavnih

<sup>2</sup>Jezik, (tj. apstraktna mašina koja ga predstavlja) je Tjuring-potpun ako može da simulira Univerzalnu Tjuringovu Mašinu (ili neku drugu, ekvivalentnu mašinu).

<sup>3</sup>Mada postoje izuzeci, kao što je CQL.

i intuitivnih jezika za pronalaženje informacija kao što je CCL ili upitni jezici pretraživača veba. Rezultat je jezik koji je čitljiv i lak za razumevanje, ali i izražajan i koristan u uslovima (manje ili više) ograničene strukture dokumenata [6].

## 4 Zaključak

Problem koji upitni jezici rešavaju je na prvi pogled prost, ali je posle dubljeg posmatranja jasno da su podproblemi koji ga sačinjavaju izuzetno kompleksni i teški za implementaciju. Uprkos tome, ne očekuje se od njihovih korisnika da budu tehnološki obrazovani, pa se veliki značaj stavlja i na intuitivnost i jednostavnost gramatike upitnih jezika. Zbog toga su upitni jezici interesantni i sa pogleda projektovanja sistema, i dizajna programskih jezika, i interakcije čoveka i računara.

## Literatura

- [1] E. F. Codd. A relational model of data for large shared data banks. *Commun. ACM*, 13(6):377–387, June 1970.
- [2] Bruce Croft, Donald Metzler, and Trevor Strohman. *Search Engines: Information Retrieval in Practice*. Addison-Wesley Publishing Company, USA, 1st edition, 2009.
- [3] Apache Software Foundation. Gremlin, 2009. online at: <https://tinkerpop.apache.org/gremlin.html>.
- [4] Andrew Gierth. CTS implemented in SQL, 2009. online at: [https://wiki.postgresql.org/wiki/Cyclic\\_Tag\\_System](https://wiki.postgresql.org/wiki/Cyclic_Tag_System).
- [5] Andrew Gierth. Mandelbrot set in SQL, 2009. online at: [https://wiki.postgresql.org/wiki/Mandelbrot\\_set](https://wiki.postgresql.org/wiki/Mandelbrot_set).
- [6] Sally H McCallum. A look at new information retrieval protocols: Sru, opensearch/a9, cql, and xquery. In *World Library and Information Congress: 72nd IFLA General Conference and Council*, 2006.
- [7] C. J. Van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979.