

## Dodavanje reči u trie

Za svaku reč koja se ubacuje, beleži se i putanja HTML dokumenta gde je reč nadjena.

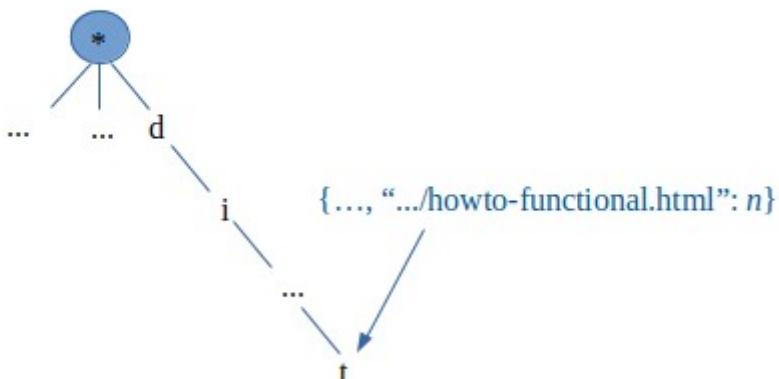
Algoritam za ubacivanje reči u trie slovo po slovo:

1. Proveri da li deca trenutnog cvora sadrže trenutno slovo reči koja se ubacuje
2. Ako postoji, to slovo ne treba da se ubacuje
  - Predji na čvor iz dece koji odgovara tom slovu
  - Predji na sledeće slovo reči koja se ubacuje
3. Ako ne postoji, slovo treba ubaciti u listu dece
  - Predji na dodati cvor
  - Predji na sledeće slovo reči koja se ubacuje
4. Kada su sva slova ubacena, u poslednji dodati cvor treba dodati informaciju u kom dokumentu se nalazi ubacena reč i povećati brojač te reči za taj dokument

Path: *.../howto-functional.html*

“... In a large program, different sections might ...”

---



## Traženje reči u trie-u

Za svaku traženu reč, vraća se rečnik koji sadrži putanje HTML dokumenata gde je reč nadjena i odgovarajući broj nadjenih reči.

Algoritam radi slično ubacivanju reči, s tim što se ne vrši ubacivanje slova, već se nakon nađenog poslednjeg slova vraća rečnik koji se nalazi na tom čvoru. U koliko reč ne postoji biće vraćen prazan rečnik.

## Izdvajanje HTML dokumenata iz direktorijuma

Algoritam radi rekurzivno.

1. Za trenutni direktorijum proveri redom fajlove:
  - Da li je fajl koji se završava sa “.html”? Ako jeste dodaj u listu.
  - Da li je direktorijum? Ako jeste, uradi 1. za njega

## Parsiranje upita pretrage

1. Razdvoj string pretrage po razmacima
2. Proveri da li je neka od reči rezervisana (AND, OR ili NOT), ako jeste:
  - Proveri da li su unete tri reči i,
  - da li je rezervisana reč na drugoj poziciji i koja je rezervisana reč u pitanju
    - Za OR, posmatraj obe reči uobičajeno
    - Za AND, drugu reč posmatraj kao obaveznu
    - Za NOT, drugu reč posmatraj kao isključivu
  - ako neki od uslova nije zadovoljen, string pretrage se smatra pogrešnim
3. Ako nema rezervisanih reči, sve reči posmatraj uobičajenu reč pretrage

Algoritam vraća listu reči pretrage i, po potrebi, jednu obaveznu reč ili jednu isključivu reč

Primeri upita pretrage:

"python AND programming"  
"python NOT programming"  
"python OR programming"  
"python and programming"

Vraćaju:

[python], programming, None  
[python], None, programming  
[python, programming], None, None  
[python, and, programming], None, None

## Pretraga dokumenata

Algoritam pretrage koristi parsirani string upita pretrage i popunjeno trie stablo.

1. Za svaku reč iz liste reči pretrage pretraži trie i rezultat smesti u [rezultujući rečnik](#)
  - U slučaju da dokument već postoji u [rezultujućem rečniku](#), saberi broj nađenih reči za taj dokument
  - U suprotnom, samo ubaci novi dokument i broj nađenih reči za taj dokument
2. Ako postoji obavezna reč, pretraži trie i rezultat smesti u rečnik obaveznih dokumenata
  - Uraditi presek i sabrati broj nadjenih reči u odgovarajućih dokumenata
3. Ako postoji isključiva reč, pretraži trie iz [rezultujućeg rečnika](#) obrisati dobijene dokumente

## Parser – napredna pretraga

Koristi se Lark biblioteka za parsiranje, konkretno implementacija *Earley* algoritma za parsiranje stringova. Gramatika parsiranja je opisana u kodu:

```
start : exp
exp : WORD -> word
      | "(" exp ")"
      | exp exp -> or_op
      | exp "||" exp -> or_op
      | exp "&&" exp -> and_op
      | "!" exp -> not_op
```

Parser vraća stablo parsiranja koje opisuje prioritet operacija.