Snowball - jezik za implementaciju steming algoritama

Dragan Ivanović dragan.ivanovic@uns.ac.rs

Katedra za informatiku, Fakultet tehničkih nauka, Novi Sad

2015.

Šta je Snowball

- SNOBOL (StriNg Oriented and symBOlic Language) link
- Prost jezik za obradu stringova
- M. Porter 2001. godina
- String paterni odlučuju o toku izvršavanja programa

Stemeri zasnovani na algoritmu

- Skupom pravila, konvencija i definisanjem redosleda njihove primene se definiše steming
- Vrlo su značajni za IR sisteme, često se koriste jer ih je lakše kreirati nego stemere zasnovane na rečniku
- Na žalost nema puno opisanih stemera zasnovanih na algoritmima, pa čak i kada su opisani često je taj opis nejasan i tumači se pogrešno
- Rade brzo i začuđujuće dobro Why does it do so well? (Krovetz, 1995 - page 89)
- Snowball je pogodan za opis ovakvih algoritama, lako ga je razumeti, odnosno pravilno tumačiti
- Stemer opisan Snowball jezikom se lako može prevesti u programske jezike Java i ANSI C, a postoje načini i da se koriste u Python-u, Objektnom PASCAL-u i drugim jezicima

Zašto opisati stemer Snowball-om

- Kraaij-Pohlmann je stemer za Holandski opisao upotrebom C programskog jezika (open source je ovaj stemer)
 - teško za razumeti
 - može se koristi takvo kakvo jeste, ali ga je teško menjati, poboljšavati
 - pošto ga je teško razumeti onda su manje šanse da se koncepti iz ovog stemera primene za neke druge jezike
- "There are two main reasons for creating Snowball. One is the lack of readily available stemming algorithms for languages other than English. The other is the consciousness of a certain failure on my part in promoting exact implementations of the stemming algorithm described in (Porter 1980), which has come to be called the Porter stemming algorithm." M.F.Porter

Tipovi podataka i nazivi

- Tipovi podataka
 - strings
 - integers
 - booleans
- Nazivi
 - počinju slovom, u nastavku se mogu javljati slova, cifre i
 - naziv se vezuje za string, integer, boolean, routine, external, grouping
 - Ts (naziv1 naziv2 ...)
 - Sve ima deklaraciju na početku fajla u kojem je program i odvojenu definiciju

Literali i rutine

- Literali
 - hex 'DA' karakter čiji je kod hex DA
 - hex 'D A' dva karaktera čiji su kodovi hex D i A (13 i 10 carriage return i line feed)
 - decimal '13' carriage return
- Makroi
 - slovo ž stringdef zx hex '17E'
 - suglasnici define c 'bvgdzjklmnprstfhc{cx}{cy}{zx}{sx}{dx}'
 - da bi prethodna linija radila kako treba pre nje mora biti definisano stringescapes {}
- Rutine
 - define R as C
 - R je naziv rutine koja je prethodno deklarisana
 - C je komanda ili grupa komandi stavljenih u zagrade ()

Naredbe i signali

- Tok izvršavanja Snowball programa je određen signalima koje šalju naredbe - nema klasičnih naredbi za određivanje toka kao što su if, then, break, itd.
- Primer naredbe

$$$x = 1$$

Još jedan primer

- Sve naredbe vraćaju true ili false kao rezultat svog izvršavanja (to su signali)
- Niz naredbi

$$x > 0 y = 1$$

• N-ta naredba se izvršava ako je rezultat svih prethodnih naredbi true

Složene naredbe

- C1 or C2
 - Izvrši C1. Ako vrati true ignoriši C2, inače izvrši C2. Rezultujući signal je true ako i samo ako C1 ili C2 vraćaju true.
- C1 and C2
 - Izvrši C1. Ako vrati false ignoriši C2, inače izvrši C2. Rezultujući signal je true ako i samo ako C1 i C2 vraćaju true.
- o not C
 - Izvrši C. Rezultujući signal je true ako i samo ako C vraća false, inače je rezultujući signal false.
- try C
 - Izvrši C. Rezultujući signal je true šta god da vraća C.
- fail C
 - Izvrši C. Rezultujući signal je false šta god da vraća C.
- (\$x > 0 \$y = 1) or (\$y = 0)
 - postavi y na 1 ako je x veće od 0, inače postavi y na 0.
- try((\$x > 0) and (\$z > 0) \$y = 1)
 - postavi y na 1 ako su i x i z veći od 0 i vrati true (*try* nikada ne vraća false)

Aritmetičke operacije

- Operandi za aritmetičke operacije su +, -, *, / (postoji i unarni operator -)
- \$X op AE
 - umesto *op* može stojati jedan od sledećih elemenata koji porede vrednosti: ==, !=, >=, >=, <, ili jedan od sledećih elemenata za dodelu vrednosti: =, +=, -=, *=, /=
- Postoje i ugrađene promenjive koje se mogu koristiti u aritmetičkim izrazima
 - minint minimalni negativni broj
 - maxint maksimalni pozitivni broj
 - sizeof s broj karaktera u s, gde je s ime stringa
 - cursor pozicija do koje se došlo u procesiranju stringa
 - limit pozicija do koje se maksimalno može doći u stringu
 - size broj karaktera stringa (koji je rezultat obrade)

Naredbe vezane za stringove

- String može biti procesiran sa leva u desno (podrazumevani način) i sa desna u levo
- \$x = 'animadversion'
 - Postavljanje vrednosti, može se dodeli i rezultat neke komande, ne mora konstanta
- \$x ('anim', 'ad', 'vers')
 - Provera da li string x ima podreč anim, nakon čega ima i podreč ad, nakon čega ima i podreč vers. Nakon ove operacije vraća se rezultat true, a promenljiva cursor pokazuje iza podreči vers.
- C1 or C2
 - Ako su C1 i C2 naredbe vezane za stringove, pored standardnog ponašanja opisanog nekoliko slajdova ranije, ako prva naredba vrati false, vraća se i *cursor* na početak. Slično je i za ostale složene naredbe.
- o do C
 - Izvrši C i bez obzira na rezultat, vrati *cursor* gde je ranije bio i vrati *true*
- Više o ovim elementima sintakse na ovom linku

Naredbe vezane za stringove

• Postavljanje dela stringa u drugu promenljivu ili brisanje dela stringa

Ostali elementi sintakse

- Naredbe vezane za stringove
 - Markiranje dela stringa
 - Rad sa limit i cursor pokazivačima
 - Procesiranje stringa od nazad
 - substring i among
- Naredbe vezane za boolean promenljive
- Grupisanje karaktera
- Definicija kompletnog programa externals
- Komentari
- Više o ovim elementima sintakse na ovom linku

Porterov stemer za Engleski

- Osnovni Porterov stemer u Snowball programu je dostupan na ovom linku
- Unapređeni Porterov stemer u Snowball programu je dostupan na ovom linku

Stemer za Srpski 1/5

```
routines ( // deklaracija rutina, odnosno procedura
          prelude mark_regions
          R1 Step_1 )
externals ( stem ) // funkcije koje se mogu spolja pozvati, 'glavni program'
integers ( p1 ) // deklaracija integer varijabli
groupings ( v ca sa) // deklaracija grupa
stringescapes {} //izbacivanje odredjenih karaktera iz stringova
stringdef cx hex '10D' // definicija makroa koji se kasnije koristi u ostalim
// izrazima (cekic)
stringdef cy hex '107' // cevap
stringdef zx hex '17E' // zaba
stringdef sx hex '161' // suma
stringdef dx hex '111' // djak
//dzak je d{zx}ak
define v 'aeiou' // definicija grupe - samoglasnici - vowels
define sa '{cx}{cy}{zx}{sx}{dx}' //definicija grupe - suglasnici sa akcentima
define ca 'bvgdzjklmnprstfhc' + sa // definicija grupe - svi suglasnici - consona
//(osim lj, nj, dz jer su to dva slova)
```

Stemer za Srpski 2/5

```
define prelude as ( //definicija rutine koja ijekavicu i
           //jekavicu svodi na ekavicu
  do repeat goto ( //ponavljaj dok god ne dodjes do kraja
       //stringa, odnosno dokle god pronalazis 'ije' u stringu
    ca ['ije'] ca <- 'e' // ijekavica u ekavicu
  do repeat goto ( //ponavljaj dok god ne dodjes do kraja
       //stringa, odnosno dokle god pronalazis 'je' u stringu
     ca ['je'] ca <- 'e' // jekavica u ekavicu
```

Stemer za Srpski 3/5

```
define mark_regions as ( //definicija rutine koja postavlja p1
     //na odgovarajuce mesto
  $p1 = limit // p1 uzima vrednost limit, a limit je pokazivac
            //iza zadnjeg slova (brojna vrednost)
  gopast v gopast non-v setmark p1 // probaj da u reci od
       //pocetka preskocis jedan samoglasnik i nesto sto nije
       //samoglasnik i tu postavi p1, ako nisi uspeo onda
  //p1 ostaje na kraju reci
```

Stemer za Srpski 4/5

```
backwardmode (
  define R1 as $p1 <= cursor // definicija rutine R1 koja proverava
       //da li ce nakon odredjene radnje ostati dovoljno slova u reci
       //(onako kako je definisano u mark_regions)
  define Step_1 as ( //definicija glavne rutine koja radi steming,
       //poziva se u backwards rezimu
  [substring] //pronadji maskimalni podstring
   //(koji je nabrojan u listi nakon among)
   //krenuvsi od cursor pozicije, tj. od kraja
   R. 1
            // ako je podstring koji ostaje dovoljne duzine,
            //onda se moze uraditi stemming
      among ( //lista podstringova i njihovih zamena
             'lozi' 'loga' 'lozima' (<-'log')
             'pesi' 'pesima' (<-'peh')
             'a{sx}an' (<-'a{sx}ni')
             'skom' (<-'sko')
             'i{sx}tu' (<-'i{sx}te')
```

Stemer za Srpski 5/5

Naredbe i signali

- Neophodno je preuzeti Snowball iz Subversion repozitorijuma svn co svn://snowball.tartarus.org/snowball/trunk snowball
- Onda ga je potrebno kompajlirati upotrebom c kompajlera gcc -O -o Snowball compiler/*.c
- Prevođenje programa napisanog u Snowball jeziku u c
 ./Snowball serbianStemmer.sbl -o serbianStemmer
- Prevođenje programa napisanog u Snowball jeziku u javu
 ./Snowball serbianStemmer.sbl -java -o serbianStemmer