

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

SEMINAR

Implementacija FM-indeks algoritma

Ivan Borko, Sofia Čolaković, Florijan Stamenković

Voditelj: doc. dr. sc. Mirjana Domazet Lošo

Zagreb, siječanj 2015.

SADRŽAJ

1. Uvod i problematika	1
2. FM-indeks algoritam	2
2.1. Burrows-Wheeler transformacija (BWT)	2
2.1.1. LF-mapiranje	3
2.1.2. Rekonstrukcija originala	4
3. Implementacija i testiranje	6
4. Zaključak	7
5. Literatura	8
6. Sažetak	9

1. Uvod i problematika

Pretraživanje teksta česta je praktična potreba mnogih informacijskih sustava. Pod terminom "pretraživanje teksta" podrazumijevamo pronalazak svih pojavljivanja nekog niza znakova Q (engl. *query*) unutar drugog niza znakova S (engl. *string*). Tipično se rezultat pretraživanja R formulira kao niz indeksa (rednog broja znaka) unutar niza S na kojem počinje pojavljivanje niza Q . Primjerice, za niz $S = \text{"Žuti pas je opasan kad je opasan remenom oko pasa"}$ i niz $Q = \text{"pas"}$ rezultati pretraživanja su $R = \{6, 14, 28, 46\}$.

U području bioinformatike pretraživanje teksta koristi se u za pronalazak specifičnih sekvenci unutar zadanog genoma. Definicija pretraživanja je jednaka. Specifičnost bioinformatičkog pretraživanja jest da su nizovi koji se pretražuju iznimno velike duljine. Primjerice, ljudski genom tipično sadrži oko 3.3×10^9 znakova, što bi otisnuto na A4 stranice fontom veličine 10pt rezultiralo s otprilike milijun stranica.

Postoje mnogi algoritmi pretraživanja teksta koji na jednostavan način ispunjavaju definirane zahtjeve. Iz perspektive računalne složenosti algoritama, jednostavno je implementirati pretraživanje teksta koje radi u linearnom vremenu¹. Nažalost, za nizove vrlo velike duljine linearno vrijeme znači praktično predugo trajanje pretraživanja. Otud potreba, pogotovo u području bioinformatike, za vremenski sub-linearnim algoritmima pretraživanja.

U ovom projektu razmatramo implementaciju pretraživanja teksta koja se bazira na konceptu FM-indeksa. Konkretna implementacija bazira se na binarnim stablima valića (engl. *wavelet-trees*). Teoretsko razmatranje i praktično testiranje pokazuju da ovakva implementacija pretraživanja ima vremenski sub-linearnu složenost.

¹ Ako nije drukčije navedeno pri razmatranju složenosti pretraživanja uvijek govorimo o složenosti s obzirom na duljinu pretraživanog niza S .

2. FM-indeks algoritam

"Indeksiranje" teksta označava generiranje struktura podataka koje su podrška efikasnom pretraživanju. Za velike tekstove poželjno je da indeks bude memorijski efikasan. FM-indeks [1] pristup je indeksiranju koji ispunjava zahtjeve memorijske efikasnosti i sub-linearnog vremena pretraživanja. Prije nego definiramo FM-indeks, potrebno je razmotriti podatkovne strukture i algoritme koji ga sačinjavaju.

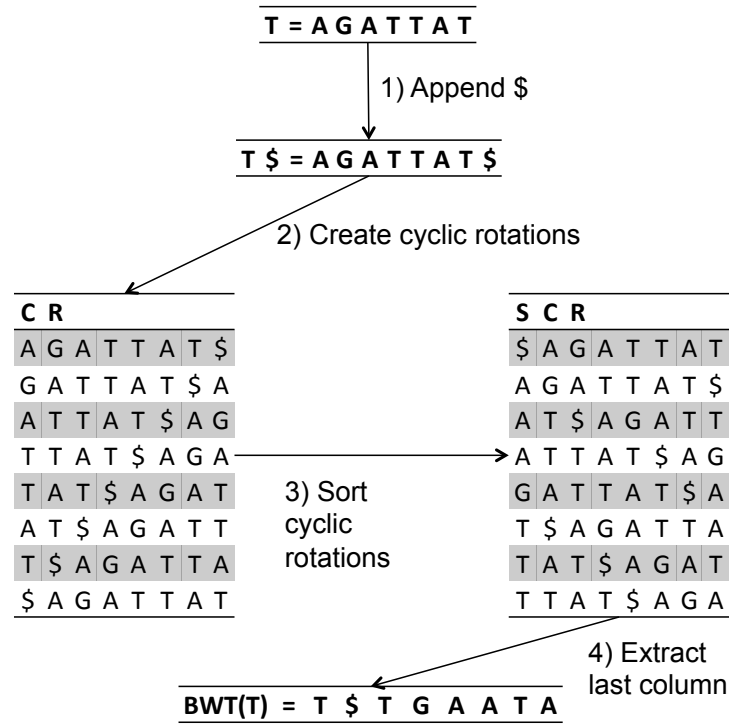
2.1. Burrows-Wheeler transformacija (BWT)

Burrows-Wheeler transformacija [2] transformira niz znakova na način koji će omogućiti efikasnu pohranu i brzo pretraživanje. BWT transformirani niz originalnog teksta T označavati ćemo sa T^{BTW} . Transformacija se provodi sljedećim koracima:

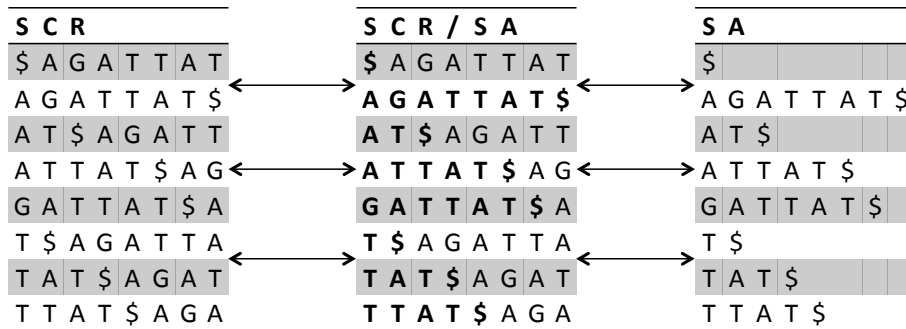
1. Poseban znak '\$' koji je leksikografski manji od svih ostalih znakova se dodaje na kraj niza T
2. Cikličkim rotacijama niza T dobiva se skup nizova koji čini tablicu CR (engl. *cyclic rotation*)
3. Tablica CR se leksikografski sortira u tablicu SCR
4. Posljednji stupac tablice SCR se ekstrahira u rezultat T^{BTW}

Opisani postupak ilustriran je za niz $T = "AGATTAT"$ na slici 2.1, preuzetoj iz rada [3].

Bitno je primjetiti kako je BWT transformacija niza srodna sufiksnoj listi SA (engl. *suffix array*). Sufiksna lista je struktura podataka koja se često koristi u algoritmima nad tekstom. Njenu formulaciju nećemo detaljno objašnjavati, materijali na temu su široko dostupni. Sličnost između BWT transformacije i SCR tablice korištene u BWT transformaciji ilustrirana je slikom 2.2.



Slika 2.1: Primjer algoritma BWT transformacije niza $T = AGATTAT$



Slika 2.2: SCR tablica BWT transformacije i sufiksna lista za niz $T = AGATTAT$

2.1.1. LF-mapiranje

LF-mapiranje (engl. *last-to-first mapping*) opisuje relaciju između posljednjeg stupca SCR tablice (označenog L) i prvog stupca te iste tablice (označenog F)¹. LF-mapiranje postulira da i -to pojavljivanje znaka c unutar stupca L korespondira i -tom pojavljivanju tog istog znaka unutar stupca F . Pri tome "korespondira" znači da se radi o istom znaku unutar originalnog niza T . Jednostavan dokaz ovog iskaza moguće je naći u [3].

Brza implementacija LF-mapiranja (konstantne vremenske složenosti) može se osvariti implementacijom dviju pomoćnih tablica. Tablica prefiksnih suma C (engl. *prefix-*

¹Primjetimo da je L isti stupac koji definira BWT transformaciju.

sum table) niza T za svaki znak c pohranjuje broj znakova u T koji su manji od c . Tablica pojavljivanja Occ (engl. *occurrence table*) pohranjuje informaciju koliko puta se neki znak c pojavio u nizu T do pozicije i (isključujući znak točno na poziciji i). Korištenjem tablica C i Occ LF mapiranje za T^{BWT} (koji odgovara stupcu L) računa se na sljedeći način, za znak c na poziciji i :

1. Pronađi broj pojavljivanja c u T^{BWT} do pozicije i unutar tablice Occ
2. Pronađi broj znakova manjih od c u T^{BWT} unutar tablice C
3. Zbroj pronađenih vrijednosti je indeks korespondirajućeg znaka u stupcu L

2.1.2. Rekonstrukcija originala

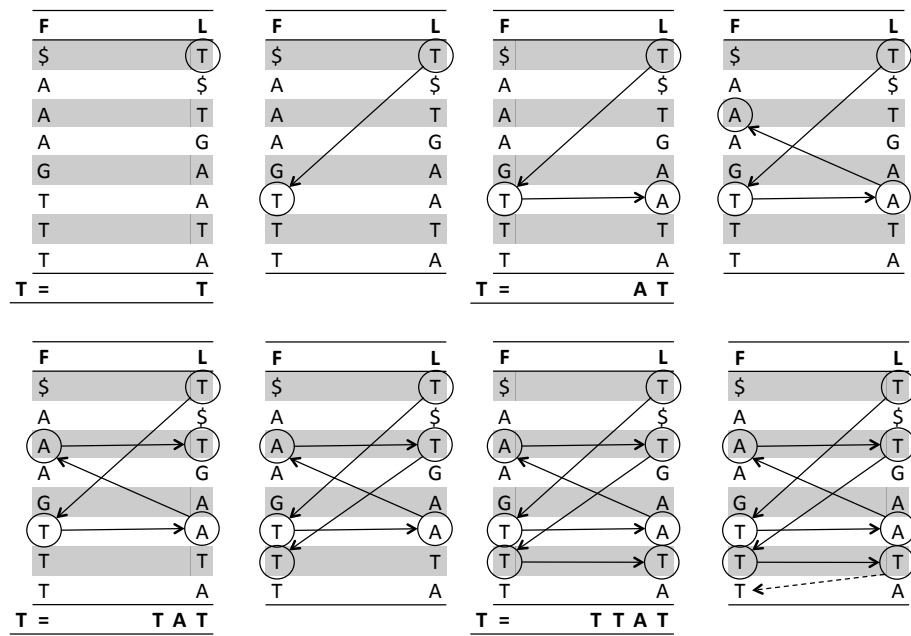
Na temelju transformiranog niza T^{BWT} moguće je rekonstruirati originalni niz T korištenjem LF-transformacije. Postupak je jednostavan, ako imamo na umu definiciju LF-transformacije i činjenicu da je prvi znak u T^{BWT} zasigurno posljednji znak niza T (ovo proizlazi iz činjenice da smo pri postupku BWT transformacije na početak niza umetnuli znak '\$').

Rekonstrukcija niza T obavlja se unatrag, od posljednjeg znaka prema prvom. Postupak je sljedeći:

1. Prvi znak iz T^{BWT} je posljednji znak iz T , zabilježimo ga
2. LF-transformacijom pronađimo F -indeks posljednjeg zabilježenog znaka ²
3. Dobiveni F -indeks u T^{BWT} nizu ukazuje na znak koji u nizu T prethodi posljednjem zabilježenom znaku (posljedica rotacije pri konstrukciji SCR)
4. Zabilježimo znak na F -indeks poziciji niza T^{BWT} u rekonstrukciju
5. Ako posljednji zabilježeni znak nije '\$', vraćamo se na korak 2.

Opisani postupak vizualiziran je na slici 2.3. Bitno je primjetiti kako pri rekonstrukciji originala nismo koristili ništa osim transformiranog niza T^{BWT} i LF-mapiranja (koje se ostvaruje tablicama C i Occ).

²U ovom trenutku nemamo tablicu SCR , zanima nas samo indeks



Slika 2.3: Primjer rekonstrukcije originala iz transformiranog niza T^{BWT}

3. Implementacija i testiranje

4. Zaključak

Zaključak.

5. Literatura

- [1] Paolo Ferragina and Giovanni Manzini. Opportunistic data structures with applications. pages 390–398, 2000.
- [2] M. Burrows, D. J. Wheeler, M. Burrows, and D. J. Wheeler. A block-sorting lossless data compression algorithm. Technical report, 1994.
- [3] J. Singer. A wavelet tree based fm-index for biological sequences in seqan. Master’s thesis, 2012.

6. Sažetak

Sažetak.