Burrows-Wheeler algoritm

Konstantin Tretjakov

7. mai 2003. a.

Burrows-Wheeler algoritm on tekstipakkimisalgoritm, mis oli esimesena kirjeldatud Michael Burrows and David Wheeleri poolt kirjutatud artiklis "A Block-sorting Lossless Data Compression Algorithm" mis ilmus aastal 1994. Tegelikult oli algoritmi põhiline transformatsioon avastatud Wheeleri poolt aastal 1983.

Algoritm on lubab kiirt lahtipakkimist, ning tema pakkimissuhe on võrreldav paljude kommertspakkijate (nagu näiteks PKZIP) omadega. Samas ei ole see algoritm patenteeritud.

Algoritm on plokkidepõhine — tekst pakitakse plokkide kaupa. Põhiidee seisneb selles, et alguses tervele plokkile rakendatakse mingit kindlat sümbolite ümberjärjestust *(Burrows-Wheeler transformation)*, mis "lükaks" sarnased sümbolid tekstis rohkem kokku. Edasi saadud tekstile rakendatakse *Move-To-Front* kodeerimine, mis annab seda efekti, et pikad jadad, milles esineb ainult väike arv erinevaid sümboleid (ja nad tekkisid esimese transformatsiooni käigus), teisenduvad pikadeks väikeste numbrite (nagu 0, 1) jadadeks. Tulemuseks on sama pikkusega sõne, kus on eriti palju väikesid arve, ning on vähe suuri arve. Ilmselt sellist sõnet saab juba eriti hästi kokkupakkida mingi tavalise statistilise pakkimise meetodiga, nagu aritmeetiline kodeerimine või Huffmani kodeerimine. Nüüd vaatleme algoritmi erinevaid samme.

1 Burrows-Wheeler transformatsioon

Algoritmi põhisamm seisneb transformatsioonis, mis lühidalt on kirjeldatav järgmiselt: moodustatakse teksti kõikvõimalikud tsüklilise nihkega saadud ümberjärjestused, sorteeritakse neid, ning väljastatakse saadud järjestuses sõnede viimased tähed. Väljastatakse ka nn. primaarne indeks *(primary index)* — esialgse sõne indeks sorteeritud massiivis. Näiteks: olgu meil sõne `abraca`. Moodustame sellest kõikvõimalikud tsüklilised ümberjärjestused ning kirjutame need maatriksi ridadena:

```
  012345
 +------
0|abraca
1|bracaa
2|racaab
```

```
3|acaabr
4|caabra
5|aabrac
```

Nüüd sorteerime saadud maatriksi read:

```
  012345
 +------
0|aabrac
1|abraca
2|acaabr
3|bracaa
4|caabra
5|racaab
```

Tähistame saadud maatriksi M-ga. Selle maatriksi viimane veerg `caraab` ongi transformeeritud sõne, ning primaarne indeks on 1, sest just selles reas esineb esialgne sõne `abraca`. Näitame nüüd, et teisendus on pööratav. Olgu meil antud kodeerimisega saadud sõne `caraab` ning primaarne indeks $p = 1$. Märkime, et esialgse maatriksi M iga veerg oli antud sõne sümbolite mingi ümberjarjestus, kusjuures esimene veerg koosnes sümbolitest alfabeetilises järjekorras. Seega sorteerides sõnes `caraab` sümboleid, me saame kohe maatriksi M veeru 0 — `aaabcr`. Moodustame maatriksi M', nihutades kõik M read ühe sümboli võrra paremale:

```
M:              M':
  012345           012345
 +------          +------
0|aabrac         0|caabra
1|abraca         1|aabrac
2|acaabr    ==>  2|racaab
3|bracaa         3|abraca
4|caabra         4|acaabr
5|racaab         5|bracaa
```

Lihtne on aru saada, et maatriksi M' read moodustavad maatriksi M ridade mingit ümberjärjestust T. Olgu $T(i)$ — maatriksi M i-nda rea indeks maatriksis M'. (näiteks $T(0) = 1$). Näitame, et teisendust T saab taastada, teades ainult maatriksite esimesed veerud. Fikseerime mingit sümbolit, näiteks `a`. Kõik `a`-ga algavad sõned maatriksis M on leksikograafilises järjekorras, sest selles maatriksis üldse kõik sõned on järjestatud. Maatriksis M' on kõik `a`-ga algavad sõned ka omavahel leksikograafilises järjestuses, sest nad kõik algavad sama tähega, ning teisest tähest alates järjestatud. Järelikult kui me vaatleme kõiki sõnesid, mis algavad sama tähega, siis nad on mõlemates maatriksites M ja M' samas järjekorras. Näiteks teades, et maatriksi M read $0, 1, 2$ algavad `a`-ga, ning maatriksi M' read $1, 3, 4$ algavad `a`-ga, saame järeldada, et $T(0) = 1$, $T(1) = 3$, $T(2) = 4$. Niimoodi saame taastada terve teisenduse T. Me teame, et maatriksi M' rea i esimene sümbol eelneb esialgses sõnes maatriksi M rea

i esimesele sümbolile. Me teame ka, et maatriksi M' rea p esimene sümbol on esialgse sõne viimane sümbol. Temale "järgneb" esialgse sõne esimene sümbol, mida me saame leida maatriksi M p-ndas reas. Seega meie näite puhul me leiame, et algse sõne esimene sümbol on $M_{1,0}$=`a`. Edasi, kuna me leidsime T me teame, et maatriksi M rida 1 teisendub maatriksi M' reaks $T(1) = 3$, ning järelikult leitud sümbolile järgnev sümbol asetseb maatriksi M reas 3. Siin on see `b`. Edasi leiame järgmise sümboli maatriksi M reast $T(3) = 5$ ja nii leiame terve esialgse sõne. Seega Burrows-Wheeleri teisendus on tõepoolest pööratav.

Proovime veel põhjendada, miks antud teisendus kasulik on. Oletame et me pakkime tavalist ingliskeelset teksti, kus palju kordi esineb sõne `the`. Kuna pärast teisenduse käigus moodustatud maatriksi sorteerimist, read, mis algavad tähtedega `he` koonduvad kokku, siis ilmselt paljud need read lõppevad tähega `t` (kuna see on kõige tõenäolisem täht, mis saab sümbolite komplekti `he` eelneda). Seega kodeeritud sõnes paljud `t`-tähed koonduvad grupidesse. Veel ühe illustratsioonina saab tuua näite, et sõne `blablabla` transformeeritakse sõnesse `lllaaabbb`.

2 Move-To-Front encoding

Antud kodeerimismeetod toimub järgmiselt: alguses on meil olemas list tüüpi struktuur, kuhu on sisse kantud kõik tähestiku tähed. Kodeerimine toimub tähe kaupa — iga järgmist tähte kodeeritakse tema indeksiga listis, ning seejärel liigutakse selle tähe listi algusesse. Sellega saavutakse seda, et pikad sümbolite jadad, mis koosnevad ainult mõnedest erinevatest sümbolitest kodeeruvad jadadega väikestest numbritest . Meie näite puhul sõne `caraab`. kodeeruks järgmiselt:

List	Kodeeritav sümbol	Väljastatav kood
a b c r	c	2
c a b r	a	1
a c b r	r	3
r a c b	a	1
a r c b	a	0
a r c b	b	3

3 Pakkimine

Algoritmi lõpetamiseks teostatakse tegeliku pakkimist mingi tavalise pakkimismeetodiga nagu näiteks aritmeetiline kodeerimine või Huffmani kood. Kerge on näha et meie näite puhul ei saanud me Burrows-Wheeleri transformatsioonist eriti palju kasu, kuid tegelikkuses suurte tekstide korral kasu on märgatav.