

# Porovnávanie dátových súborov

Autor: Michal Linhard

Diplomový vedúci: Mgr. Tibor Hegedüs

29. júna 2006

# Obsah

<b>1 Popis problému</b>	<b>3</b>
1.1 Základné pojmy . . . . .	3
1.2 Maximálne zhody . . . . .	3
1.3 Suffixové polia . . . . .	3
1.4 Strom LCP-Intervalov . . . . .	4
<b>Literatúra</b>	<b>5</b>

# Úvod

Táto práca sa zaobrá algoritmami ktoré slúžia na porovnávanie dátových alebo textových súborov, alebo všeobecnejšie povedané, ľubovoľných reťazcov nad nejakou konečnou abecedou. Poskytuje prehľad a klasifikáciu známych metód porovnávania textových alebo dátových reťazcov. Za jeden z najvšeobecnejších problémov považujeme hľadanie všetkých maximálnych zhôd (match) medzi porovnávanými retazcami. Tomuto problému sa tu venujeme bližšie a skúmame jeho súvislosť s ostatnými porovnávacími problémami ako je najdlhšia spoločná podpostupnosť, najdlhší spoločný podreťazec, editačná vzdialenosť, alignment problem a iné. Veľký výskum v tejto oblasti priniesli hlavne výskumníci z poľa bioinformatiky, kde je potrebné porovnávať veľké množstvo genetickej informácie. V poslednej dobe vzniklo veľa nových prístupov k porovnávaniu reťazcov, napríklad algoritmy založené na suffixových poliach, ktoré tu stručne zhrnieme. Väčina nástrojov na porovnávanie textov sa zameriava na vyhľadávanie rozdielov medzi dvoma verziami nejakého dokumentu. V práci argumentujeme, že aj hľadanie podobnosti, ako sú maximálne zhody môže byt užitočným nástrojom na analýzu textu alebo dát. Väčšina z prác na tému porovnávania reťazcov pomocou hľadania zhôd (matches) alebo opakovaní (repeats) je zameraná najmä na oblasť genetických informácií. Hlavným cieľom práce je implementácia jednoducho použiteľnej knižnice pokrývajúcej oblasť textových a dátových súborov, ktorá by využívala najmodernejšie metódy a algoritmy z tejto oblasti.

(voliteľná časť) Analyzujeme existujúce nástroje a knižnice a definujeme požiadavky na nový program využívajúci maximálne zhody dvoch súborov pre analýzu textovej alebo binárnej informácie a pokúsime sa ho implementovať.

# Kapitola 1

## Popis problému

### 1.1 Základné pojmy

Nech  $\Sigma$  je konečná usporiadaná abeceda.  $\Sigma^*$  je množina všetkých reťazcov nad  $\Sigma$ . Množinu  $\Sigma^* \setminus \{\varepsilon\}$  neprázdných reťazcov budeme označovať  $\Sigma^+$ . Nech  $S$  je reťazec dĺžky  $|S| = n$  nad  $\Sigma$ . Pre jednoduchosť budeme predpokladať, že veľkosť abecedy je konštantná a  $n < 2^{32}$ . Posledný predpoklad nám zaručí, že celé číslo z intervalu  $<0, n>$  môže byť uložené v 4 bajtoch. Špeciálny symbol  $\$$  je prvkom  $\Sigma$  (väčším ako všetky ostatné prvky) ale nevyskytuje sa nikde v  $S$ .  $S[i]$  označuje znak na pozícii  $i$  pre  $0 \leq i < n$ . Pre  $i \leq j$ ,  $S[i..j]$  označuje podreťazec  $S$  začínajúci znakom na pozícii  $i$  a končiaci znakom na pozícii  $j$ . Podreťazec  $S[i..j]$  môžeme reprezentovať párom pozícií  $(i, j)$ .

### 1.2 Maximálne zhody

**Definícia 1.2.1.** Nech  $S_1, S_2$  sú reťazce nad  $\Sigma$ . Trojica čísel  $(p_1, p_2, l)$  sa nazýva zhoda (match) reťazcov  $S_1, S_2$ , práve vtedy, keď

$$\begin{aligned} 0 \leq p_1 < |S_1| \quad \wedge \\ 0 \leq p_2 < |S_2| \quad \wedge \\ 0 \leq l \leq \min(|S_1|, |S_2|) \quad \wedge \\ p_1 + l \leq |S_1| \quad \wedge \\ p_2 + l \leq |S_2| \quad \wedge \\ S_1[p_1..p_1 + l - 1] = S_2[p_2..p_2 + l - 1]. \end{aligned}$$

Čísla  $p_1$  resp.  $p_2$  udávajú pozíciu zhody v  $S_1$  resp.  $S_2$  a číslo  $l$  jej dĺžku.

**Definícia 1.2.2.** Zhoda  $(p_1, p_2, l)$  reťazcov  $S_1, S_2$  sa nazýva ľavo resp. pravo maximálna práve vtedy, keď  $S_1[p_1 - 1] \neq S_2[p_2 - 1]$  resp.  $S_1[p_1 + l] \neq S_2[p_2 + l]$ . Zhoda sa nazýva maximálna keď je ľavo aj pravo maximálna.

**Definícia 1.2.3.** Nech  $(p_1, p_2, l)$  je zhoda medzi dvoma tými istými reťazcami  $S$ . Potom podreťazec  $\omega = S[p_1..p_1 + l - 1]$  sa nazýva opakovanie (repeat). Opakovanie je maximálne keď je maximálna daná zhoda. Opakovanie je supermaximálne, keď sa nevyskytuje ako podreťazec žiadneho iného maximálneho opakovania.

**Definícia 1.2.4.**

### 1.3 Suffixové polia

**Definícia 1.3.1.** Suffixový strom pre reťazec  $S$  je orientovaný strom s koreňom, ktorý má presne  $n + 1$  listov očíslovaných 0 až  $n$ . Každý vnútorný vrchol okrem koreňa má aspoň dvoch potomkov

a každá hrana je označená neprázdnym podreťazcom  $S$ . žiadne dve hrany vychádzajúce z toho istého vrcholu nemôžu byť označené reťazcami začínajúcimi na to isté písmeno. Kľúčovou črtou suffixového stromu je, že pre ľubovoľný list  $i$ , konkatenácia reťazcov na hranách cesty z koreňa do listu  $i$  je presne reťazec  $S_i$ , kde  $S_i = S[i..n-1]\$$  označuje  $i$ -ty neprázdný suffix reťazca  $S\$$

**Definícia 1.3.2.** *Suffixové pole suftab reťazca  $S$ , je pole celých čísel v rozmedzí 0 až  $n$ , špecifikujúce lexikografické usporiadanie  $n+1$  suffixov reťazca  $S\$$ . To znamená, že  $S_{suftab[0]}, S_{suftab[1]}, \dots, S_{suftab[n]}$  je postupnosť suffixov  $S\$$  v stúpajúcom lexikografickom poradí. Sufixové pole vyžaduje  $4n$  bajtov pamäte.*

**Definícia 1.3.3.** *Inverzné suffixové pole  $suftab^{-1}$  je tabuľka veľkosti  $n+1$  taká, že  $suftab^{-1}[suftab[q]] = q$  pre ľubovoľné  $0 \leq q \leq n$ .  $suftab^{-1}$  vieme vypočítať v lineárnom čase zo suffixového poľa a potrebuje  $4n$  bajtov pamäte.*

**Definícia 1.3.4.** *Tabuľka bwttab pre reťazec  $S$  obsahuje Burrows - Wheelerovu transformáciu známu z dátovej kompresie (TODO: referencia). Je to tabuľka veľkosti  $n+1$  taká, že pre každé  $i$ ,  $0 \leq i \leq n$ ,  $bwttab[i] = S[suftab[i]-1]$  ak  $suftab[i] \neq 0$ . bwttab má nedefinovanú hodnotu, ak  $suftab[i] = 0$ . Tabuľka bwttab sa dá uložiť do  $n$  bajtov a skonštruovať jedným prechodom cez suffixové pole v čase  $O(n)$ .*

**Definícia 1.3.5.** *Tabuľka LCP (longest common prefix) hodnôt lcptab je pole celých čísel v rozmedzí 0 až  $n$ . Definujeme  $lcptab[0] = 0$  a  $lcptab[i]$  je dĺžka najdlhšieho spoločného prefixu  $S_{suftab[i-1]} a S_{suftab[i]}$  pre  $1 \leq i \leq n$ . Kedže  $S_{suftab[n]} = \$$ , vždy máme  $lcptab[n] = 0$ . Tabuľka môže byť vypočítaná ako vedľajší produkt počas konštrukcie suffixového poľa alebo neskôr zo suffixového poľa v lineárnom čase [6, 6].*

## 1.4 Strom LCP-Intervalov

**Definícia 1.4.1.** *Interval  $[i..j]$ ,  $0 \leq i < j \leq n$  je lcp-interval LCP hodnoty  $l$ , ak*

1.  $lcptab[i] < l$
2.  $lcptab[k] \geq l$  pre všetky  $k$ ,  $i+1 \leq k \leq j$
3.  $lcptab[k] = l$  pre aspoň jedno  $k$ ,  $i+1 \leq k \leq j$
4.  $lcptab[j+1] < l$

Budeme používať aj skratku  $l$ -interval alebo aj  $l - [i..j]$ . Každý index  $k$ ,  $i+1 \leq k \leq j$  s  $lcptab[k] = l$  budeme nazývať  $l$ -index.

# Literatúra

- [1] BRENDA S. BAKER: On finding duplication in strings and software, technical report, AT&T Bell Laboratories, February, 1993
- [2] MANBER, U., & MYERS, G.: Suffix arrays: A new method for on-line string searches, SIAM Journal on Computing, 22(5), 935–948, 1993
- [3] E. UKKONEN: On-line construction of suffix-trees, Január Algorithmica 14 (1995), 249-260
- [4] M. CROCHMORE, W. RYTTER: Text Algorithms, Oxford University Press, New York, 1994
- [5] P. KO AND S. ALURU: Space-efficient linear time construction of suffix arrays, Combinatorial Pattern Matching, pp. 200-210, 2003
- [6] JUHA KÄRKKÄINEN AND PETER SANDERS: Simple linear work suffix array construction, in Proc. 30th International Colloquium on Automata, Languages and Programming (ICALP '03). LNCS 2719, Springer, 2003, pp. 943-955
- [7] S. KURTZ AND C. SCHLEIERMACHER: REPuter: Fast Computation of Maximal Repeats in Complete Genomes, Bioinformatics, 15(5):426-427, 1999
- [1] M.I. ABOUELHODA, S. KURTZ, AND E. OHLEBUSCH: Replacing Suffix Trees with Enhanced Suffix Arrays, Journal of Discrete Algorithms, 2:53-86, 2004
- [2] M.I. ABOUELHODA, S. KURTZ, AND E. OHLEBUSCH: The Enhanced Suffix Array and its Applications to Genome Analysis, In Proceedings of the Second Workshop on Algorithms in Bioinformatics, pages 449-463. Lecture Notes in Computer Science 2452, Springer-Verlag, 2002
- [3] R. DEMENTIEV, J. MEHNERT, AND J. KÄRKKÄINEN: Better External Memory Suffix Array Construction, In Workshop on Algorithm Engineering & Experiments, Vancouver, 2005
- [4] JENS MEHNERT: External Memory Suffix Array Construction, Diploma Thesis
- [5] R. DEMENTIEV, L. KETTNER, P. SANDERS: Stxxl: Standard Template Library for XXL Data Sets. Technical Report 2005/18, Fakultät für Informatik, University of Karlsruhe