INFORMATION RETRIEVAL USING HYBRID APPROXIMATE STRING MATCHING METHOD

Róbert Toth

Master Degree Programme (3), FIT BUT

E-mail: xtothr00@stud.fit.vutbr.cz

Supervised by: Jan Kaštil

E-mail: ikastil@fit.vutbr.cz

Abstract: In this paper we present an improved hybrid indexing method for approximate string matching, combining offline and online approach to problem of finding approximate occurences of pattern in text. Offline part of algorithm uses suffix automaton to store text indexed in linear time. During text preprocessing, another (pattern–independent) automaton is created to enable searching by blocks instead of individual characters. This online part of algorithm is based on modification of Four–Russians technique presented by Wu, Manber and Myers in 1996.

Keywords: EEICT, approximate string matching, approximate information retrieval, inexact searching, suffix trees, suffix automata, text indexing, offline algorithms

1. ÚVOD

Približné vyhľadávanie reťazcov v textoch je generalizáciou presného vyhľadávania a umožňuje v texte nájsť aj reťazce určitým spôsobom podobné hľadanému. Takéto vyhľadávanie je výpočtovo aj algoritmicky podstatne náročnejšie ako presné, no umožňuje užívateľovi nájsť hľadaný reťazec aj v prípade výskytu pravopisných chýb, preklepov či iných poškodení dát. Použitie presného vyhľadávania v širšom spektre úloh je totiž limitované nutnosťou mať k dispozícii dáta bez akýchkoľvek chýb, no v reálnom aj digitálnom svete sa prirodzene vyskytujú údaje v určitej miere poškodené alebo nepresné–či už spôsobené nepresným prepisom, informačným šumom, zaokrúhľovaním hodnôt alebo chybou parity. Práve umožnenie výskytu chýb pri približnom vyhľadávaní rozširuje možnú aplikáciu takto upravených algoritmov na mnohé ďalšie oblasti.

Cieľom tejto práce je navrhnúť kombinovaný algoritmus, ktorý pre umožnenie rýchlejšieho vyhľadávania využije znalosť (statického) textu na jeho predspracovanie, zároveň však po zadaní vyhľadávaného reťazca predspracuje aj ten, čo umožní urýchliť vyhľadávanie v texte ešte viac. Pre tento účel bude pre indexáciu textu použitý suffixový automat a vyhľadávaný reťazec bude predspracovaný pomocou špeciálneho univerzálneho deterministického konečného automatu.

2. PROBLÉM PRIBLIŽNÉHO VYHĽADÁVANIA

Problém približného vyhľadávania reťazcov môžeme chápať ako úlohu nájsť v zadanom *texte* $T_{1\ldots n}$ všetky pozície hľadaného reťazca (*patternu*) $P_{1\ldots m}$, umožňujúc pritom výskyt určitého počtu k *editačných chýb* (inzercií, delécií alebo substitúcií) v každej zhode. Algoritmy používané k približnému vyhľadávaniu môžeme pritom rozdeliť na dve základné skupiny. *Online algoritmy*, používané v prípade predom neznámeho textu, si pred samotným vyhľadávaním väčšinou predspracujú pattern. Na opačnej strane stoja *offline algoritmy*, v literatúre nazývané tiež *indexovacie*, pretože pred vyhľadávaním vytvoria nad textom určitú dátovú štruktúru (tzv. index).

Klasické riešenie tohoto problému je založené na použití tzv. *dynamickej programovacej tabuľky*, čo je matica veľkosti $(m+1)\times(n+1)$, kde každá hodnota $C_{i,j}$ reprezentuje editačnú vzdialenosť

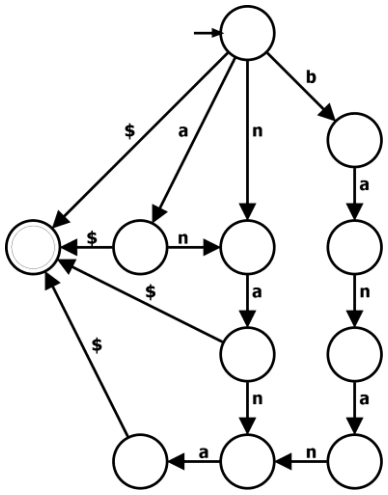
$P_{1..i}$ od $T_{1...j}$. Detailný popis výpočtu je možné nájsť v [3] a [5]. Po vypočítaní všetkých hodnôt tabuľky sa približné zhody nachádzajú na tých pozíciách v texte T_j, kde hodnota $C_{m,j} \leq k$.

3. PREDSPRACOVANIE TEXTU

Rýchlosť vyhľadávania klasického algoritmu v rozsiahlych textoch môže byť veľmi malá, pretože výpočet celej matice veľkosti $(m+1)\times(n+1)$ má zložitosť $\mathrm{O}(m\times n)$. Z tohoto dôvodu je potrebné nad textom vytvoriť index, ktorý umožní vyhľadávanie rapídne zrýchliť. V našom algoritme použijeme tzv. suffixový automat.

3.1. SUFFIXOVÝ AUTOMAT

Suffixový automat (*suffix automaton*) je konečný automat, ktorý prijíma práve a len všetky suffixy textu, nad ktorým je postavený. Jeho konštrukcia má časovú zložitosť $\mathrm{O}(n)$ a je popísaná v [2]. Obrázok 1 zobrazuje automat vytvorený nad textom „banana“.



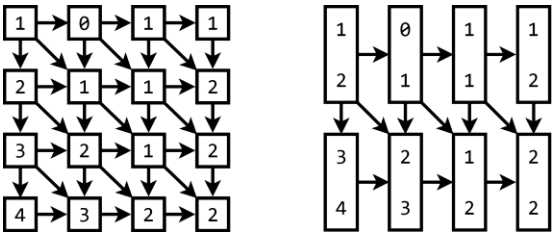
Obrázok 1: Suffixový automat vytvorený nad textom „banana“ (znak „$“ je ukončovací)

3.2. VYHĽADÁVANIE V TEXTE

Pre približné vyhľadávanie v suffixovom strome použijeme Cobbsov algoritmus popísaný v [1]. Ten nám umožní dosiahnuť časovú zložitosť približného vyhľadávania $\mathrm{O}(m\times q)$, kde $0 < q \leq n$. Postup vyhľadávania je pritom taký, že algoritmus suffixovým automatom prijíma jednotlivé znaky vyhľadávaného patternu a v každom stave počíta konkrétny stĺpec dynamickej programovacej tabuľky. Po prijatí posledného znaku patternu sa podobne ako v pôvodnom algoritme detekuje zhoda tým, že sa overí, či je posledná hodnota stlpca matice $C_{m,j} \leq k$.

4. SPRACOVANIE PATTERNU

Online časť algoritmu spočíva v predspracovaní vyhľadávaného patternu takým spôsobom, aby jeho následné vyhľadávanie v texte bolo rýchlejšie. K tomuto účelu využijeme zaujímavú vlastnosť dynamickej programovacej matice, že každá jej hodnota $C_{i,j}$ závisí len na troch najbližších hodnotách $C_{i-1,j}$, $C_{i,j-1}$ a $C_{i-1,j-1}$ (odvodenie v [3]). Vďaka tomuto faktu je možné výpočet jednotlivých hodnôt nahradiť výpočtom po celých blokoch hodnôt, ako to ukazuje Obrázok 2. Prvýkrát bol takýto výpočet aplikovaný v [4], my použijeme vylepšený algoritmus WMM z [6].



Obrázok 2: Transformácia výpočtu jednotlivých hodnôt za výpočet blokov

5. KOMBINOVANÝ ALGORITMUS

Jemným prispôsobením oboch uvedených algoritmov, pôvodne určených na samostatné využitie pri úplne opačných situáciách (Cobbsov pre offline vyhľadávanie, kým WMM pre urýchlenie online vyhľadávania) je možné udržať lineárny čas predspracovania textu a zároveň dosiahnuť násobné urýchlenie vyhľadávania podľa zvolenej veľkosti bloku algoritmu WMM. Jeho najväčšia nevýhoda (pomalé predspracovanie patternu) sa navyše v tejto kombinácii s Cobbsovým algoritmom stráca, pretože automat ním vytvorený je univerzálny, tj., nie je závislý od patternu a je ho teda možné vytvoriť len raz pri predspracovaní textu v lineárnom čase.

6. ZÁVER

V rámci práce sa podarilo navrhnúť nový hybridný algoritmus kombinujúci dva rozdielne existujúce prístupy. Autorova predchádzajúca implementácia algoritmu WMM v [5] ukazuje, že je takto možné dosiahnuť až štvornásobné zrýchlenie už tak rýchleho Cobbsovho algoritmu, ktorý má pri konštantnom m alebo k časovú zložitosť nezávislú od dĺžky textu. Zároveň sa preukázalo, že algoritmy z oboch tried je možné veľmi vhodne kombinovať, čo otvára dvere ďalšiemu skúmaniu takéhoto prístupu. Hoci je totiž metóda WMM jednou z často využívaných optimalizácií v online algoritmoch, podľa našich informácií nebola nikdy použitá na urýchlenie offline vyhľadávania.

REFERENCIE

[1] COBBS, Archie L. Fast approximate matching using suffix trees. *Combinatorial Pattern Matching.* 1995, vol. 937, s. 41-54. ISBN: 978-3-540-60044-2.

[2] CROCHEMORE, Maxime. Transducers and Repetitions. *Theoretical Computer Science.* 1986, vol. 45, s. 63-86. Dostupné z: <http://www-igm.univ-mlv.fr/~mac/Articles-PDF/Cro86tcs-trans-rep.pdf>.

[3] GUSFIELD, Dan. *Algorithms on Strings, Trees and Sequences : Computer Science and Computational Biology*. Davis (California): Cambridge University Press, 1997. ISBN 9780521585194.

[4] MASEK, William Joseph; PATERSON, Michael S. A Faster Algorithm Computing String Edit Distances. *Journal of Computer and System Sciences*. February 1980, vol. 20, no. 1, s. 18-31. Dostupný z WWW: <http://www.cs.ust.hk/mjg_lib/bibs/DPSu/DPSu.Files/MaPa80.pdf>. ISSN 0022 0000.

[5] TOTH, Róbert. *Približné vyhľadávanie reťazcov.* Brno, 2011. Dostupné z WWW: <http://www.fit.vutbr.cz/study/DP/BP.php.cs?id=12745&file=t>. Bakalárska práca. FIT VUT v Brně. Vedúci práce Ing. Jan Kaštil.

[6] WU, Sun; MANBER, Udi; MYERS, Gene. A Sub-quadratic Algorithm for Approximate Limited Expression Matching. *Algorithmica*. January 1996, vol. 15, no. 1, s. 50-67. Dostupný z WWW: <ftp://ftp.cs.arizona.edu/reports/1992/TR92-36.pdf>. ISSN 0178-4617.