Oponentský posudek diplomové práce Radovana Šestáka: Suffix Array for Large Alphabet

Předkládaná práce se zabývá implementací Burrows-Wheelerovy transformace v rámci většího projektu pro kompresi XML souborů s využitím rozkladu souboru na slabiky. Zde je potřeba pracovat s řetězci nad abecedou i značně větší než je běžných 256 znaku. To přináší nové problémy, které mnoho algoritmů rychlých na malých abecedách neřeší a nelze je tak přímočaře použít. Úkolem bylo prozkoumat existující algoritmy, porovnat je a navrhnout nejlepší algoritmus pro dané použití.

Celkově lze říci, že autor zřejmě věnoval nemalé úsilí studiu literatury, implementaci algoritmů a experimentům. Toto úsilí se bohužel autorovi nepodařilo v předložené práci zúročit, což je škoda. Stať velmi trpí rozhodnutím použít pro vypracování angličtinu a také velkým množstvím různě závažných chyb a nejasností. Problémy jsou natolik vážné, že bych autorovi doporučil jejich odstranění a zhodnocení vynaloženého úsilí na podzim.

Nyní k obsahu práce a výhradám podrobněji. Text lze v zásadě rozdělit na teoretický úvod (kapitoly 1 až 3), popis algoritmů (kapitoly 4 a 5) a experimentální závěr (kapitoly 6 a 7). Uvedené tři části se znatelně liší kvalitou zpracování, která má lehce stoupající tendenci. Například v první části jsou obrázky jen doleva zarovnané bloky textu, kterým schází názornost. Druhá část již přináší centrování a skutečné obrázky, ale ty jsou nepřiměřeně velké a většinou nejsou v textu zmíněny. Konečně, třetí část přináší přehledné tabulky a grafy, kterým lze ovšem vytknout občasné nezvýraznění nejlepšího výsledku a to, že v grafech symboly nesouhlasí s legendou. Stejně jako u obrázku lze pozorovat vývoj i v úrovni angličtiny, která v teoretické části působí autorovi značné potíže a dovádí ho až k tvrzení, že algoritmy pro porovnávání řetězců mají složitost $O(n^2 \cdot \log n)$ nebo k definici řetězce nad abecedou Σ jako uspořádané množiny znaků. V dalších částech jsou již problémy s jazykem výrazně menší.

Teoretická část práce začíná poněkud kostrbatým úvodem, který je následován pasáží obsahující základní stringologické a lingvistické definice. V těchto definicí lze nalézt řadu chyb, například v druhé části definice 2.1.6 chybí dva symboly a dva jsou špatně. Také v definici 2.1.10 se náhle místo R objevuje X. Konečně je tu i překvapivá změna v části 2.3, kde se od značení znaku abecedy a řetězců latinkou přechází k řecké abecedě, která není jinde v textu použita. V této sekci stojí za pozornost i kolize značení v definici 2.3.2, která zavádí označení Σ_{L_i} jak pro velká písmena, tak pro souhlásky.

Poslední kapitola teoretické části je věnovaná popisu Burrows-Wheelerovy transformace a její inverze na dvou příkladech. Tato část je velmi podobná odpovídající pasáži v dokumentaci projektu XBW a to včetně obrázku. Celkově je popis obou algoritmů srozumitelný, ale také zde jsou nějaké problémy. Na začátku oddílu 3.2 je ve výstupu transformace o jedno n více a v části 3.2.1 se náhle autor začne odkazovat na dosud nezmíněnou matici M. Algoritmus 1 postrádá inicializaci proměnné sum a konečně ve třetím odstavci na straně 15 se tvrdí, že F je posledním sloupcem M', ač je ve skutečnosti druhým.

Druhá část práce začíná kapitolou 4, která se zabývá popisem algoritmů pro třídění rotací nebo sufixů řetězce. Výběr algoritmů je celkem pestrý a jejich rozdělení do skupin je správné. Nicméně je škoda, že autor nevycházel například z přehledového článku trojice Puglisi, Smyth a Turpin. To by mu dle mého soudu poskytlo větší přehled a pravděpodobně změnilo výběr algoritmů k lepšímu. Například by autor nezkoumal Sadakaneho algoritmus a jeho vylepšení Larssonem, ale až finální algoritmus obou autorů. Vlastní popisy algoritmů by na některých místech potřebovaly více detailů. Například v popisu Sadakaneho algoritmu není jasné, co je iterace nebo jak je použit quicksort. Nejčastější chybou v této pasáži je, vzhledem k použití rotací, chybějící ochrana před přetečením či podtečením indexu. Jsou zde ovšem i další chyby. Ve třetím odstavci sekce 4.5.1 je tvrzení, že je-li rotace R_i první v přihrádce c_2c_2, pak je rotace R_{i+1} první v přihrádce c_2, ale například pro $X = aax$ a $i = 1$ to neplatí. Dále v algoritmu 3 zřejmě schází inicializace většiny buněk polí S a E. Jako poslední příklad chyb v kapitole 4 zmiňme i chaos značení v popisu kroku 1 na straně 28, kde pole SA mají většinou špatný počet apostrofů nebo jim alespoň chybí index 12 a většina i má být j.

Další kapitola se zabývá inverzí modifikované Burrows-Wheelerovy transformace, kde bylo namísto rotačního pole použito pole sufixové. Autor se pokouší vyhnout použití ukončovacího znaku pro vstup s odůvodněním, že by se tento znak navíc nemusel vejít do paměti vymezené pro původní řetězec. To je sice pravda, ale sotva by byl problém s tímto znakem navíc počítat předem, už jen proto, že významně urychluje porovnávání řetězců. Popisovaný algoritmus se zabývá odstraněním ukončovacího znaku ve výstupu transformace a jeho doplněním při inverzi, bohužel bez analýzy dopadu na rychlost a kompresi. Ani v této části se autor nevyvaroval chyb, když v lemma 5.0.3 opomíjí zdůraznit, že uvedené platí jen při použití sufixového pole, nebo v algoritmu 5 opět neinicializuje proměnnou sum a místo T[index] používá T_{index}.

Poslední část práce je z většiny tvořena kapitolou šest, jejíž název slibuje vylepšení výkonu Burrows-Wheelerovy transformace. Autor nejprve popisuje prostředí pro experimenty a celkem pestrou skupinu testovacích souborů. Zajímavé je, že v experimentech nebyla transformace měřena samostatně, ale v rámci kompresního programu XBW. Tento program na transformovaný výstup spouští move-to-front a jistou variantu kódování běhů. Zde by asi bylo namístě zdůvodnit toto složení kompresní metody a také absenci další fáze, například obvyklého Huffmanova kódování. Také je otázkou, zda či jak další fáze algoritmu ovlivňují naměřené časy. Část 6.4 se zabývá otázkou rychlého porovnávání rotací a sufixů a ukazuje, že použití rotací nemá ani v dosaženém kompresním poměru oproti sufixům žádnou výhodu. Autor popisuje dva algoritmy pro porovnávání rotací a jeden pro sufixy. Zde je škoda, že autor nevyzkoušel také implementace porovnávání obsažené ve standardní knihovně nebo vlastní implementace pracující přímo s ukazateli. Rovněž není vysvětleno, proč všechny navrhované funkce začínají testováním, zda jí nejsou předkládány dva shodné indexy. Může k tomu vůbec v implementovaných algoritmech dojít? Také by stálo za vysvětlení použití operátoru predekrementace místo prostého odečtení jedničky při vytváření návratové hodnoty funkce. Jde o styl psaní kódu nebo o optimalizaci? Je zajímavé, že naměřené rychlosti ukazují u algoritmu Sada vyšší rychlost při použití rotací než u sufixů. To při absenci jiného vysvětlení navozuje dojem možné chyby v implementaci. Konečně, ani tato část práce není bez chyb. V algoritmu 7 je na místě klíčového slova for použito while a v ukázkách kódu 6.1 je místo obvyklého predekrementu jen minus.

Další sekce kapitoly šest se zabývá výběrem vhodného quicksortu. Autor naznačuje experimentování s volbou pivota, ale neuvádí žádné výsledky, ani volbu pro finální implementaci tří testovaných variant. Následující podkapitola zkoumá vliv otočení vstupu před Burrows-Wheelerovou transformací. Výsledkem je drobné zhoršení komprese a drobné změny v rychlosti transformace. Autor se pokouší zdůvodnit změny rychlosti, ale pro algoritmus Kärkkäinena a Sanderse je vysvětlení chybné. Příčinou je to, že při třídění obráceného řetězce nejsou porovnávány obrácené přípony původního řetězce, jak tvrdí autor, ale prefixy původního řetězce odzadu. Část 6.7 celkem pěkně ukazuje vliv velikosti bloků vstupující do Burrows-Wheelerovy transformace na rychlost a kompresní poměr. Bohužel autor vynechává mnoho podstatných detailů jako jsou způsob dělení na bloky, velikost abecedy či použití rotací nebo sufixů. V navazující sekci se autor zabývá blahodárným vlivem velikosti abecedy na kompresi i rychlost. Opět schází detaily implementace dělení na bloky a výroby slovníku pro slabiky a slova. Dále považuji za dobré zdůraznit, že abeceda bajtů je zde zřejmě zbytečně znevýhodňována a to jak v rychlosti, tak i v kompresi. Důvodů je hned několik. Vzhledem k textové povaze lze očekávat skutečné využití zhruba 40 hodnot, což umožňuje lepší předzpracování v první fázi transformace. Také by bylo možné vytvoření slovníku bajtů nebo dodání Huffmanova kódování do kompresního algoritmu.

Poslední část kapitoly šest se zabývá výběrem nejrychlejšího algoritmu. Vybrán byl Itoh s tím, že v příliš repetitivních řetězcích, kde se tento algoritmus nechová dobře, dojde k detekci tohoto problému a přejde se k asymptoticky lepšímu algoritmu. Z dvou možností byl vybrán algoritmus Kärkkäinena a Sanderse jako rychlejší než Sadakanho, ač výsledky hovoří opačně. Bohužel autor opět neposkytuje detaily navrhovaného řešení a evidentně jej neimplementoval a neověřil svůj návrh.

V závěru autor jako hlavní přínos práce uvádí zlepšení komprese a zvýšení rychlosti při použití abecedy slov místo abecedy bajtů. Tento výsledek je ovšem poněkud pokažen tím, že u výsledků chybí řada informací a varianta používající abecedu bajtů není odpovídajícím způsobem optimalizována. Rovněž schází jakákoliv informace o rychlosti inverzní transformace, která by ukázala, zda nezaplatíme za zrychlení komprese pomalejší dekompresí.

Přiložené CD obsahuje vše potřebné a k dokonalosti mu schází snad jen testovací skripty nebo detailnější výsledky.

Závěrem bych chtěl zopakovat, že jde potenciálně o velmi dobrou práci, která je sražena mnoha zbytečnými chybami a nejasnostmi až pod hranici obhajitelnosti. Doporučuji práci opravit a odevzdat v nejbližším dalším termínu.

Praha, 11. května 2007

Martin Šeňtt