پردازش بهینه پرس و جوهای XML با استفاده از راهنمای تطبیق الگو

وحید گرکانی*، سید کامیار ایزدی†، مصطفی حق‌جو سانیجی‡

چکیده

با افزایش تعداد وحجم سندهای XML اهمیت پردازش بهینه پرس و جوهای XML بیشتر به چشم می‌خورد. پرس و جوهای XML شامل مجموعه‌ای از المان‌هایی هستند که تحت یک ساختار درختی به هم مربوط می‌باشند. در نتیجه مسأله انطباق شاخه‌ای§ مابین پرس و جو و سند، به عنوان هسته مرکزی پردازش پرس و جو مورد توجه است. روش‌های زیادی برای انطباق پرس و جوهای شاخه‌ای ارائه شده است ولی در تمامی این روش‌ها گره‌هائی که در نتیجه نهایی نیز شرکت ندارند، پردازش می‌شوند. این مسأله باعث پردازش بی‌ثمر گره‌ها و درنتیجه افزایش زمان پاسخگوئی به پرس و جو می‌شود.

در این مقاله با ارائه ایده راهنمای تطبیق الگو، روش جدیدی برای پردازش پرس و جوی XML مطرح شده است. در این روش تنها گره‌هائی پردازش می‌شوند که حتما در جواب نهایی شرکت دارند. راهنمای تطبیق الگو، رابطی مابین شما و محتوای سند است. راهنمای تطبیق الگو بر خلاف برخی از روشهای انطباق شاخه‌ای می‌تواند با بسیاری از شاخص‌های فعلی به خوبی کار کند و به انواع پرس و جوها به طور کارا پاسخ دهد. کارایی این روش با استفاده از محک DBLP در مقایسه با روش‌های مطرح در این زمینه نشان داده شده است.

کلمات کلیدی

انطباق شاخه‌ای، راهنمای تطبیق الگو، راهنمای پرس و جو، بهینه‌سازی جبری پرس و جو

Optimized XML Query Processing Using Pattern Matching Guide

Vahid Garakani, Kamyar Izadi, Mostafa Haghjoo
Faculty of Computer Engineering
Iran University of Science & Technology
Tehran, Iran

Abstract

With increasing of the number and size of XML documents, nowadays optimized processing of XML queries is more important. XML queries specify patterns of selection predicates on multiple elements related by a tree structure. Matching all occurrences of such a twig pattern in an XML document is a core operation for XML query processing. There are many methods for twig pattern matching, but all of them have useless accesses to many nodes which not used to construct the final result. Consequently it leads to increasing of the run time of the query processing.

In this paper, we propose a novel idea, called Pattern Matching Guide for XML query processing. In our proposed method only the nodes which are used to construct the final result are accessed. Pattern Matching Guide is a mediator between the schema and the content of XML document. Pattern Matching Guide, against some of twig pattern matching methods, could be used easily besides existing XML indexes and efficiently processes all kind of queries. The experimental results from DBPL benchmark show the efficiency of our method.

Keywords

Twig pattern matching, Pattern Matching Guide, Query Guide, Algebraic query optimization

* دانشجوی کارشناسی ارشد، رشته نرم افزار، دانشگاه علم و صنعت ایران، دانشکده مهندسی کامپیوتر، vahidgarakani@comp.iust.ac.ir
† دانشجوی دکتری، رشته نرم افزار، دانشگاه علم و صنعت ایران، دانشکده مهندسی کامپیوتر، izadi@iust.ac.ir
‡ عضو هیأت علمی دانشگاه، دانشگاه علم و صنعت ایران، دانشکده مهندسی کامپیوتر، haghjoom@iust.ac.ir
§ Twig pattern matching

۱- مقدمه

با افزایش تعداد و حجم سندهای XML، اهمیت پردازش کارای پرس وجوهای XML بیشتر به چشم می خورد. سندهای XML ذاتا درختی (گرافی شکل) هستند و این بدان معنی است که ساختار و محتوا در این سندها در کنار یکدیگر هستند. برای مثال پرس و جوی Q1 را در نظر بگیرید:

Q1: STUDENT//BOOK[TITLE='XML']

در Q1 از یک طرف رابطه ساختاری مابین *STUDENT*، *BOOK* و *TITLE* و از طرف دیگر رابطه مقداری *XML* در *TITLE* نیاز به وجود دو شاخص مقداری و ساختاری را نشان می‌دهد. امروزه شاخص‌های قدیمی مانند B^+Tree برای شاخص گذاری مقداری، کارائی قابل قبولی درسندهای XML دارند. در نتیجه مشکلی که همچنان توجه بسیاری از مقالات را به خود معطوف کرده است مشکل انطباق‌های ساختاری در این پرس وجوها است. به منظور معرفی ساختار یک سند راهنماهایی مانند XML SCHEMA، DTD، DATAGUIDE و SUMMARY STRUCTURAL ارائه شده است. از این راهنماها در هدایت پرس و جو استفاده می‌شود[۳-۱]. به علاوه شاخص‌های مسیری[۱] زیادی مانند Strong DataGuide، ToXin، Fabric، APEX، Index1، A(k) Index و F&B ارائه شده است. این شاخص‌ها روی مسیر شاخص گذاری می‌کنند و به پرس و جو های مسیری با سرعت بیشتری پاسخ می‌دهند [۹-۳]. از طرف دیگر برای پاسخ به پرس و جوهای ساختاری روش های زیادی مانند structural join و holistic twig join وجود دارد. در این روش‌ها برای پاسخ دادن به پرس وجوها از این راهنماها استفاده بهینه‌ای نمی‌شود و گره‌هائی را در مراحل میانی دخالت می‌دهندکه در جواب نهائی نیستند [۲۱-۱۱]. برای مثال در پاسخ به پرس و جوی Q1 با روش structural join، این پرس و جو به تعدادی از پیوندهای دودوئی تبدیل می‌شود *(student//book* و*book/title)* و با تجزیه پرس وجو به روابط پدر و فرزندی و یا جدی و نسلی[۲] حجم زیادی از داده‌های میانی تولید می‌شود. Holistic twig join سعی در برطرف کردن این مشکل دارد و پرس وجو را تجزیه نمی‌کند. این روش برای پاسخ به روابط جدی و نسلی بهینه است ولی برای روابط پدر و فرزندی مناسب نمی‌باشد. در ضمن برای پاسخ به پرس وجو تمامی گره‌های موجود در پرس و جوی Q1 را پردازش می‌کند.

در روش ارائه شده در [۲۲] فقط گره‌های برگ در پرس و جو (در Q1 گره *TITLE*)دخالت داده می‌شوند. هر گره دارای یک کد می‌باشد. با رمزگشائی پیشوند کد گره برگ، اطلاعاتی درباره اجداد آن *STUDENT*، *BOOK* مشخص می‌شود. این روش نیز به صورت کورکورانه و بدون راهنما به مقایسه گره‌ها می‌پردازد و در برخورد با اسناد حجیم کارائی مطلوبی ندارد. در این مقاله روش بهینه‌ای برای پردازش پرس و جوی XML ارائه نموده‌ایم. در روش ما با استفاده از راهنمای پرس و جو و راهنمای تطبیق الگو گره‌هائی پردازش می‌شوند که حتما (و یا در صورت پیدا شدن معادل در شاخه دیگر پرس وجو) قسمتی از جواب نهائی را برای ما تولید می‌کنند.کارهائی را که در این مقاله ارائه می‌دهیم، می‌توان به صورت زیر خلاصه کرد :

- در این مقاله یک راهنمای پرس و جو ارائه گردیده است. این راهنما در پیدا کردن گره‌های جواب به ما کمک می‌کند.
- روش ایجاد راهنمای تطبیق الگو به وسیله اعمال پرس و جو روی راهنمای پرس و جو شرح داده شده است.
- توسعه جدیدی از کدگذاری dewey بیان شده است. استفاده از این روش کدگذاری و استفاده از شاخص‌های مسیر، رسیدن به جواب نهائی را آسان‌تر می‌کند.
- روشی برای بهینه‌سازی راهنمای تطبیق الگو با بهره‌گیری از نمایش‌های جبری ارائه کرده‌ایم.
- کاربرد راهنمای تطبیق الگو در انواع عملگرها را شرح داده‌ایم.

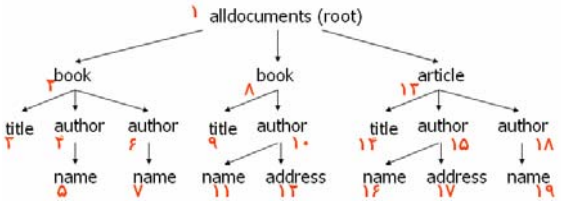
۲- کارهای مرتبط

Structural join اولین روشی است که برای پردازش عبارات مسیر[۳] ارائه شده است. در این روش یک عبارت مسیر به تعدادی رابطه پدر و فرزندی و یا جد و نسلی تبدیل می‌شود و در مرحله بعد ادغام دودوئی صورت می‌گیرد. مشکل این روش حجم زیاد گره‌های میانی است که باید در پردازش دخالت داده شوند[۱۱,۱۲,۱۳,۲۳]. اولین بار در [۱۵] holistic twig join ارائه شد.این روش سعی می‌کند حجم گره‌های میانی را کم کند و عمل شکستن پرس وجو را با استفاده از دو روش path stack و twig stack انجام ندهد. بهینه سازی‌های زیادی مانند [۱۶] روی این روش انجام گرفته است. برخی مقالات این روش را برای سندهای شاخص‌گذاری شده توسعه داده‌اند [۱۲,۱۶,۱۷,۱۸,۲۴]. در نهایت holistic twig optimal ارائه شد. ولی مشکلی که در این روش وجود دارد دخالت دادن گره‌های غیر برگی پرس و جو، در پردازش آن است. مشکل دیگر این روش بهینه نبودن آن در روابط پدر و فرزندی مابین گره‌های پرس وجو است(در روابط جدی ونسلی به صورت بهینه عمل می‌کند). در [۲۲] روشی ارائه شده است که سعی دارد فقط از طریق رمز گشایی پیشوند برگ‌ها مسیر آن را تشخیص دهد. این روش از راهنمای درخت استفاده نمی‌کند و گره‌هائی را با هم مقایسه می‌کند که هیچ تاثیری در رسیدن به جواب نهائی ندارند. تاکید ما در این مقاله بیشتر بر این است که:

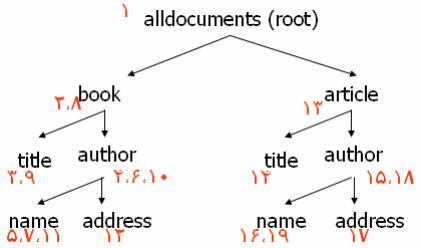
- تنها گره‌های برگ موجود در پرس و جو را پردازش کنیم.
- تعداد مقایسه‌هایی را که مابین گره‌های برگ پرس و جو انجام می‌دهیم، به کمترین حالت ممکن برسانیم به طوری که در حالت ایده‌آل هر مقایسه یک قسمت از جواب را تولید کند.

۳- راهنمای پرس و جو

راهنمای پرس وجوی ارائه شده در روش ما، توسعه‌ای از ایده ارائه شده توسط Goldman و Wildom در سال ۹۷ می‌باشد[3]. ایده کار بر این پایه است که به جای اینکه یک پرس وجو در پایگاه‌داده مستقیما اعمال شود، ابتدا پرس وجو با استفاده از شمای سند فرموله می‌گردد و با استفاده از این فرموله سازی پرس وجو، تعداد گره‌هائی که پردازش می‌شوند بسیار کمتر خواهند بود. بعد از آن تحقیقات زیادی مانند [۹-۳] که به شاخص‌های مسیری معروف شدند انجام شد. هدف این تحقیقات تسریع پرس و جوهای ساختاری بود. در این جا به طور مختصر شرحی از راهنمایی که در این روش استفاده شده است ارائه می‌کنیم. اگر فرض کنیم که شکل(۱) سند داده‌ای ما باشد، راهنمای پرس وجو چیزی مانند شکل(۲) خواهد بود.



شکل (۱): نمونه ای از سند داده ای



شکل(۲): راهنمای پرس وجوی سند

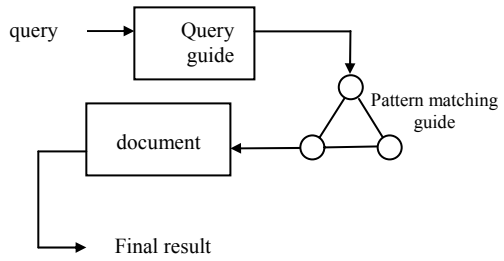
برای این راهنما می توان ویژگی‌های زیررا برشمرد:

- حدالامکان سعی می‌شود ساختار این راهنما به صورت درخت درآید. البته ما روش‌هایی را برای حذف و یا تبدیل حلقه‌هایی که ممکن است بوجود آیند داریم که بیان آنها خارج از حوصله این مقاله می‌باشد.
- در این راهنما تعداد و ترتیب گره‌ها اهمیت ندارد. این راهنما را می‌توان همان شمای سند دانست که در آن گره‌های برگ و گره‌های تکراری حذف شده‌اند.
- همراه هر گره در این راهنما یک لیست قرار دارد که به ترتیب گره‌های معادل در سند در آن ها قرار می‌گیرند.

این راهنما را می‌توان از DTD و یا XML schema ویا حتی خود سند نیز به دست آورد.

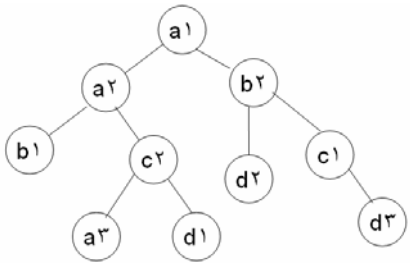
۴- تولید راهنمای تطبیق الگو

شکل(۳) یک شمای کلی از روش ما را نشان می‌دهد. ابتدا پرس وجو روی راهنمای پرس وجو اجرا می‌شود. نتیجه این اجرا تولید راهنمای تطبیق الگو است. در نهایت راهنمای تطبیق الگو با داده‌های واقعی در سند مقایسه می‌شود و جواب نهائی به دست می‌آید.



شکل(۳): روند کلی پردازش

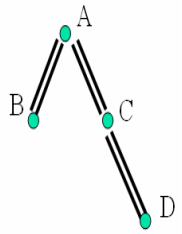
برای مثال فرض کنیم شکل(۴) راهنمای پرس وجو ما باشد و بخواهیم پرس وجوئی که در شکل (۵) آمده است را اجرا کنیم. راهنمای تطبیق الگوی حاصل در شکل(۶) نشان داده شده است. اگر بخواهیم راهنمای تطبیق الگو را بصورت ریاضی توصیف کنیم می‌توانیم آن را به صورت گراف $PMG=<V , E>$ نمایش دهیم. V مجموعه گره‌های برگ پرس و جو به شرطی که در راهنمای پرس و جو امکان رسیدن به قطعه‌ای از جواب را داشته باشند و E مجموعه یال‌های راهنمای تطبیق الگو است که به صورت $E=< n , m , L1 ,..., Ln>$ نمایش داده می‌شود که در آن n, m دو گره برگی که در شاخه‌های پرس و جو وجود دارند که باید با هم مقایسه شوند و $L1$ تا Ln نیز سطحی از کد dewey (این روش کد گذاری در ادامه شرح داده شده است) هستند که باید در این دو گره (m , n) یکسان باشد.



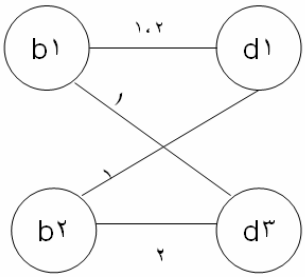
شکل(۴): راهنمای پرس وجوی سند

روش ساخت گراف بدین صورت است که ما فقط داده‌های برگ پرس وجو را در نظر می‌گیریم. برای مثال در شکل(۵) گره‌های b و d گره‌های برگ پرس جو می‌باشند، سپس در راهنمای پرس وجو به دنبال این گره‌ها (b , d) می‌گردیم که می‌توانند در سند واقعی جزء جواب باشند، برای مثال $d2$ به علت اینکه نسلی از هیچ Cای نیست نمی تواند جزء جواب پرس و جو باشد و در نتیجه در راهنمای تطبیق الگو نخواهد آمد. از آنجائی که پرس وجوی ما در این مثال یک پرس و جوی دو شاخه‌ای است از روی این راهنما نمی‌توان جواب قطعی را تنها با مشاهده یک گره بدست آورد، زیرا برای نمونه در مثال قبلی به دنبال a هائی هستیم که هر دو شاخه پرس و جو را داشته باشند و ممکن است بعضی ازآنها فقط یکی از شاخه‌ها را داشته باشند و این

چنین حالت‌ها را نمی‌توان از روی راهنمای پرس و جو متوجه شد ولی در صورت پیدا شدن گره‌های برگ پرس و جو(در این مثال *b,d*) در صورتی که هر دو جزئی از یک نقطه مشترک (در این مثال هردو زیر شاخه هائی از گره *a*) باشند، جز، جواب نهائی خواهند بود. با استفاده از این راهنما می‌توان روند مقایسه مابین گره‌های برگ در یک سند را تشخیص داد. برای مثال گره های *b1* در صورت پیدا شدن گره معادل *d1* حتما جزء جواب نهائی هستند.



شکل(۵): الگوی درختی پرس وجو



شکل(۶): راهنمای تطبیق الگو پرس و جو

در راهنمای تطبیق الگو مابین گره‌ها رابطه‌هایی قرار دارد، این روابط نحوه مقایسه مابین گره‌ها در سند را نشان می‌دهد. برای مثال از عدد ۲ که بالای رابطه *d3* , *b2* وجود دارد معلوم می‌شود که *b2* ها در صورتی شاخه دیگر *d3* ها هستند که در سطح ۲ از کد dewey خود شماره یکسانی داشته باشند. برای مثال اگر یکی از گره‌های *b2* کد dewey اش 1/5 و یکی از *d3* ها کد dewey اش 1/5/3/2 باشد هردو این گره‌ها شاخه‌هائی از یک *a* می‌باشند. در راهنمای تطبیق الگو حالت‌های خاصی وجود دارد که که بهتر است به آن‌ها دقت کنیم :

- گاهی در راهنمای پرس وجو گره‌ای وجود دارد که در سند واقعی وجود ندارد و در راهنمای تطبیق الگو تولید رابطه بی‌مورد می‌کند. این چنین گره‌هائی همانگونه که در مبحث بهینه‌سازی گفته خواهد شد به هیچ عنوان پردازش اضافه را به سیستم تحمیل نمی‌کنند، این بدان علت است که با مشاهده لیست خالی یک گره در راهنمای پرس وجو، رابطه‌های آن نیز از راهنمای تطبیق الگو پاک می‌شود و در نتیجه هیچ گره‌ای از طرف مقابل پردازش نمی‌شود.
- این روش برای پرس وجوهای *a//b/c* که تک مسیری هستند و به مقایسه با گره‌ای در شاخهٔ دیگر احتیاج ندارند، جواب حتمی را می‌دهد و راهنمای تطبیق الگو بدون رابطه خواهد بود و فقط شامل گره‌هائی است که در صورت وجود جزء جواب قطعی می‌باشند.
- با استفاده از راهنمای تطبیق الگو هیچ رابطه جدی و نسلی باقی نمی‌ماند و در نتیجه هیچ ابهامی در پرس وجو نخواهد بود و تمامی روابط و حالت های ممکن به صورت قطعی (پدر و فرزندی) وجود خواهند داشت.
- در [۲۱] از مفهوم extraction poin و بهینه‌سازی که با استفاده از این extraction poin ها انجام می‌شود استفاده شده است. در روش ما extraction poin ها را به عنوان برگ نتیجه در نظر می‌گیریم و مانند گره‌های راهنمای تطبیق الگو عمل می‌کنیم (در بخش بهینه‌سازی گراف‌ها بیشتر در این مورد شرح داده شده است).
- زمانی که می خواهیم دو گره *a , b* را در سند واقعی مقایسه کنیم، ممکن است تعداد گره‌های *b* بسیار کمتر از تعداد گره‌های *a* باشد و این می‌تواند تعدادی مقایسه بی‌مورد تولید کند. ولی در عمل در روش ما فقط تعداد مقایسه‌ها را به تعداد گره‌های لیست کمتر انجام می‌دهیم و از روی مابقی گره ها پرش می‌کنیم.
- برخی مواقع درجواب‌های ما، برای پاسخ به پرس و جوی چند شاخه ای احتیاج به مقایسه شاخه‌ها نداریم و این زمانی اتفاق می‌افتد که در راهنمای پرس وجو وجود گره برگ *a* وجود گره برگ *b* را الزامی می‌کند. در این گونه مواقع گراف گره *b* را شامل نمی‌شود و هر *a* که در سند باشد را جزء جواب محسوب می‌کند.

هدف از راهنمای تطبیق الگو رابطی مابین ساختار و داده است، به طوری که مقایسه و جستجو به صورت کورکورانه انجام نگیرد. راهنمای تطبیق الگو روشی را بیان می کند که هدف از مقایسه دو گره در آن بیان شده است. راهنمای تطبیق الگو فقط برگ هائی را از پرس و جو پردازش می‌کند که حتما و یا در صورت پیدا شدن معادل در شاخه دیگر جزء جواب ما باشند.

۵- نحوه تولید جواب های نهائی

همانگونه که در بخش های قبلی گفته شد، فرض ما بر این است که در راهنمای پرس‌وجو، هر گره یک شاخص یا یک لیست به همراه دارد که بر حسب شماره، گره‌های مرتبط در سند در آنها قرار داده شده‌اند. روش شماره گذاری در این جا همان روش dewey است. به طور خلاصه روش کد گذاری dewey به این صورت است که اگر گره *U* ، nامین فرزند گره *V* باشد، شماره کد dewey گره *U* همان شماره کد گره *V* را به عنوان پیشوند و n را در ادامه این کد دارد. برای مثال اگر کد dewey گره *V=<1/3/7>* باشد و گره *U* پنجمین فرزند گره *V* باشد کد *U=<1/3/7/5>* خواهد بود. همانگونه که در

شکل(۶) راهنمای تطبیق الگو دیدیم برای رسیدن به پاسخ باید ۴ مقایسه را انجام دهیم:

(b1 با d1) , (b1 با d3) , (b2 با d1) , (b2 با d3)

روش کار بدین صوت است که دو لیست را به ترتیب با هم مقایسه می‌کنیم. گره‌هائی جزو جواب اصلی ما به حساب می‌آیند که در سطحی که شماره آن در بالای خط رابط مابین دو گره در راهنمای تطبیق الگو آمده است با هم برابر باشند و در غیر این صورت گره‌ای که شماره کوچکتر را دارد از لیست حذف شده و گره بعدی از آن لیست پردازش می‌شود. برای مثال در زیر دو لیست *b2 , d3* نشان داده شده است. گره‌هائی در این مقایسه جزو جواب هستند که در سطح ۲ از کدشان با هم برابر باشند.

B2 = <1/3 , 1/5 , 1 ,7 >
D3 = <1/2/1/1 , 1/2/1/2 , 1/3/1/1 , 1/3/1/2 , 1/7/1/1 >
Result =< (1/3 , 1/3/1/1) , (1/3 , 1/3/1/2) , (1/7 , 1/7/1/1)>

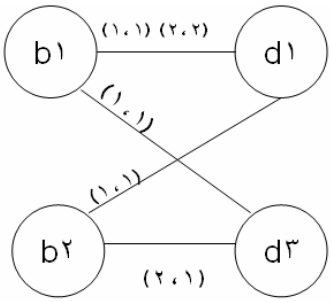
۶- بهینه‌سازی راهنمای تطبیق الگو

اگر طبق راهنمای تطبیق الگو دو لیست را از ابتدا تا انتها برای هر رابطه در راهنمای تطبیق الگو مقایسه کنیم، جوابهای تکراری زیادی تولید خواهد شد که باید از مجموعه جواب‌ها حذف شوند. در این مقاله سعی کرده‌ایم تعدادی از روش‌ها را برای کم کردن تعدا مقایسه گره‌ها ذکر کنیم و در کارهای آینده سعی می‌کنیم آنها را به صورت مدون ارائه کنیم. ما می‌توانیم راهنمای تطبیق الگو را به صورت جبری نیز بیان کنیم. برای مثال هدف پرس و جوی شکل(۴)پیدا کردن *a* هائی بود که شرط *//c//d , //b* به عنوان نسل را داشته باشند. پس ما می‌توانیم راهنمای تطبیق الگو را به صورت جبری به صورت زیر بیان کنیم :

PMG = (b1 Δ d1)U(b1 Δ d3) U (b2 Δ d1) U (b2 Δ d3)

عملگر Δ در اینجا همان عملگر مقایسه و عملگر اجتماع تمامی جواب‌ها (*a* ها) را با هم اجتماع می‌کند. حال ما می‌خواهیم تعدادی از بهینه سازی‌هائی را روی این عبارات جبری مطرح کنیم، تا عمگر Δ (مقایسه) به صورت بهینه به کار برده شود و هر گره‌ای که پردازش می‌شود حتما جواب غیر تکراری را تولید کند. در زیر تعدادی از این روش‌ها آورده شده است:

- برخی موارد پس از مقایسه دو گره به گره‌ای در سطح بالاتر می‌رسیم که این گره قبلا تولید شده است و عمگر مقایسه جوابی جدید به ما ارائه نکرده است. برای نمونه در مثال قبلی *(b1 Δ d3) U (b1 Δ d1)* ممکن است گره‌های *a1* تکراری به ما بدهد و این به دلیل این است که *b1* در هر دو مقایسه به صورت مستقل عمل می‌کند. پس ما می‌توانیم عبارت را بدین صورت بیان کنیم*(((b1 − (b1 Δ d1)) Δ d3)* یعنی *b1* هائی با *d3* مقایسه شوند که قبلا جوابی را با مقایسه *d1* تولید نکرده باشند.
- برخی موارد به مقایسه دو گره نیازی نداریم و وجود یکی از گره‌ها دیگری را قطعی می‌کند. برای مثال به رابطه *(d3 Δ b2)* دقت کنید، وجود *d3* وجود گره *d2* را حتمی می‌کند. پس می‌توان این رابطه‌ها را از گراف حذف کرد و تمامی *d3* ها را به عنوان پاسخ محسوب کرد. از طرف دیگر از آن جائی که تمامی *d3* ها به عنوان جواب قطعی در نظر گرفته می‌شوند پس در عبارت جبری راهنمای تطبیق الگو تمامی *d3* ها را از این گراف و عبارت حذف می‌کنیم.
- گاهی برای رسیدن به دو یا چند جواب فقط یک مقایسه انجام می‌دهیم. برای مثال در*(b1 Δ d1)* از هر مقایسه، ما به دو گره *a1* و *a2* می‌رسیم.
- برخی موارد ما برای رسیدن به یک پاسخ باید سه و یا بیشتر گره را با هم مقایسه کنیم و گره‌ای که در تمامی لیست‌ها معادل داشته باشد را به عنوان جواب بر می‌گزینیم *(a1 Δ a2 Δ .... Δ a_n)* . در اینگونه موارد ما می‌توانیم عملگر اشتراک را به صورت *((a1 Δ a2) ....) Δ a_n)* که به ترتیب تعداد گره‌ها در لیستها *a1 < a2 <…… < a_n* خواهد بود انجام دهیم.
- می‌توان راهنمای تطبیق الگو را طوری توسعه داد که extraction poin ها را نیز شامل شود. برای مثال از آن جائی که ما در پرس و جوی قبلی به دنبال a ها هستیم می‌توانیم راهنمای تطبیق الگو را به این صورت تغییر دهیم که در بالای هر رابطه جفت شماره قرار گیرد (شماره خروجی و شماره سطح مشترک)که این باعث می شود که تعریف یال‌ها در راهنمای تطبیق الگو به صورت *E=<n , m ,(L1 ,E1),….,(Ln ,En)>* تغییر کند(شکل ۷). دلیل این کار این است که عناصری که خروجی تکراری تولید می‌کنند را شناسائی و از پردازش آنها خودداری کنیم.



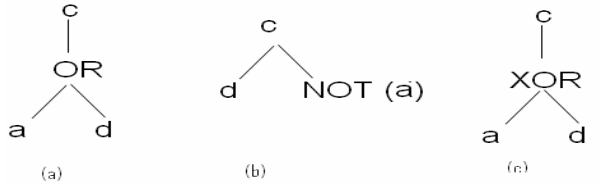
شکل(۷): راهنمای تطبیق الگو با شماره خروجی

برای نمونه در مثال قبلی برای مقایسه دو لیست b2 و d3 که *b2= <1/3>* و *d3 = <1/3/1/1 ,1/3/1/2 >* ما فقط به دنبال *a* ها هستیم و تنها سطح ۱ این شماره‌ها را می‌خواهیم، پس گره دوم در لیست *d3* را نباید پردازش کنیم چون جواب جدیدی برای ما تولید نمی‌کند.

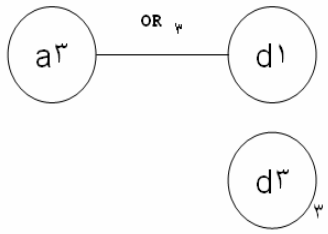
۷- راهنمای تطبیق الگو و عملگرها

تمامی مثال‌های قبلی برای پرس و جو از عملگر *AND* استفاده می‌کردند ولی در برخی موارد در پرس و جو از عملگرهایی مانند *OR* , *NOT* , *XOR* استفاده می‌شود. ما در این بخش به روش های پاسخ به این نوع پرس وجو ها و راهنمای تطبیق الگو آنها می‌پردازیم.

- در عملگر OR به دنبال گره‌هائی هستیم که یکی از شرط‌های شاخه OR را داشته باشد . برای مثال به پرس و جوی شکل (a-۸) دقت کنید، راهنمای تطبیق الگو حاصل را می‌توانیم فقط با سه گره برگ *a3* , *d1* , *d3* و بدون هیچ رابطه‌ای تشکیل دهیم ولی از آنجائی که مابین گرههای *d1* , *a3* رابطه برقرار است در نتیجه می‌تواند گره‌های تکراری *c2* را تولید بکند. پس بهتر است راهنمای تطبیق الگو را به صورت شکل(۹) رسم کنیم.



شکل (۸)



شکل (۹)

- در عملگر NOT ما به دنبال دونوع گره هستیم ۱) گره‌هائی که در صورت وجود جواب قطعی مسئله هستند ۲) گره‌هائی که در صورت پیدا نشدن معادل در شاخه *NOT* جزو جواب می‌باشند. برای مثال به پرس و جوی شکل (b-۸) دقت کنید. در اینجا *d3*ها جزء جواب‌های قطعی هستند ولی تعدادی از *d1* ها توسط *a3* حذف می‌شوند و در نهایت نمایش جبری راهنمای تطبیق الگو را می‌توانیم به صورت *d3 U (d1 – a3)* بنویسیم.
- عملگرهای XOR را بهتر است ابتدا تبدیل کنیم و بعد آنها را اجرا کنیم. برای مثال پرس و جوی شکل (c-۸) را در نظر بگیرید. بهتر است ما عبارت جبری راهنمای تطبیق الگو این پرس وجو را به صورت *d3 U (c2 – (d1 Δ a3))* نمایش دهیم.

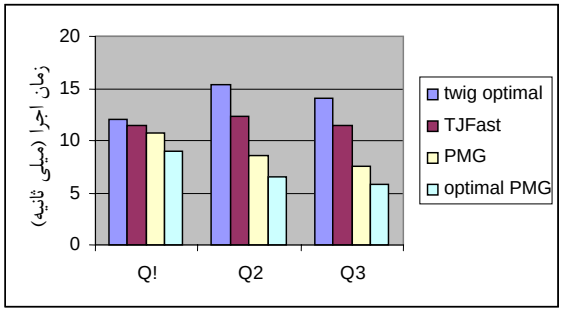
۸-نتایج آزمایشی

در این بخش مقایسه ای که از نتایج حاصل از روش راهنمای تطبیق الگو بدست آمده است را با دو روش *twig optimal* و *TJFast* (که امروزه جزء روش‌های متداول پردازش پرس و جو های شاخه‌ای محسوب می‌شوند) نشان داده شده است.تمامی نتایج و آمارهای ارائه شده حاصل اعمال پرس و جوهای جدول(۱) روی داده‌های بانک DBLP[10] می‌باشد. نتایج بدست آمده روی سیستمی با پردازنده 2.2 GHz ، اینتل پنتیوم ۴ ، سیستم عامل Linux Red Hat 8.0 و با حافظه اصلی 2G آزمایش شده است.

جدول (۱): پرس و جو های آزمایش شده

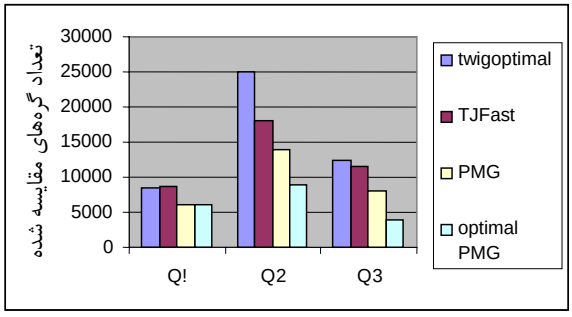
Q1 = dblp / article //year
Q2 = dblp / article[year AND month]
Q3= ./[article OR book]/title

ما به مقایسه دو مورد زمان اجرای الگوریتم و تعداد گره‌هائی که برای رسیدن به جواب نهائی پردازش می شوند برای ارزیابی روش خود پرداخته‌ایم. همانگونه که در شکل(۱۰) می‌بینیم ۴ حالت با هم مقایسه شده‌اند. دو میله سمت چپ زمان اجرای پرس و جو با استفاده از روش‌های twig optimal و TJFast می‌باشد. میله‌های سوم و چهارم به ترتیب استفاده از روش‌های راهنمای تطبیق الگو بصورت معمولی و بهینه می‌باشد.



شکل (۱۰): نمودار زمان اجرا

از طرف دیگر همانگونه که در شکل(۱۱) دیده می‌شود، این پرس و جوها را با استفاده از روش راهنمای تطبیق الگو با معیار تعداد گره‌های پردازش شده مورد آزمایش قرار داده‌ایم و این معیارها بر اساس تعداد مقایسه‌ها بیان شده‌اند. به طور میانگین با استفاده از روش راهنمای تطبیق الگو به صورت معمولی و بهینه به ترتیب به احتمال ۶۹ و ۹۷ درصد گره‌ای را که پردازش می‌کنیم جزء جواب قطعی خواهد بود و داده تکراری تولید نمی‌کند. سعی ما بر این است که این درصد ها را به صد نزدیک کنیم ولی چیزی که مشخص است بهبود غیر قابل انکاری است که این روش نسبت به روش‌های قبلی خود از لحاظ زمان پردازش و یا درصد گره پردازش شونده نسبت به روش‌های قبلی خود دارا است.



شکل (۱۱): نمودار تعداد مقایسه ها

۹- نتیجه

ما در این مقاله روشی جدید به نام راهنمای تطبیق الگو را ارائه کردیم. با استفاده از این گراف می‌توان عمل مقایسه گره‌ها را به صورت کورکورانه انجام نداد. تلاش می‌کنیم تمامی گره‌هائی که پردازش می‌شوند جوابی از مسئله را برای ما تولید کنند و یا به عبارت بهتر تعداد گره‌های پردازش شده با تعداد گره‌های جواب برابر باشند. ولی هنوز هم تا برابری کامل فاصله داریم. ما سعی می‌کنیم در مقالات و کارهای بعدی با بهبود در روش‌های زیر به این برابری برسیم :

- ارائه روشی جدید که برای پاسخ به پرس و جوهای شاخه‌ای، فقط یک شاخه را پردازش کنیم.
- ارائه روشی جدید برای اینکه که در زمان اجرای پرس و جو بتوانیم راهنمای پرس و جو را بسازیم و با استفاده از راهنمای پرس وجو بتوانیم به راهنمای تطبیق الگو برسیم.
- ارائه کامل روش‌های بهینه‌سازی روی راهنمای تطبیق الگو
- ارائه کد Dewey به صورت عددی یا محدوده‌ای به صوری که راهنمای تطبیق الگو بصورت لیستی از محدوده اعداد نمایش داده شود.

مراجع

[1] M. N. Garofalakis, A. Gionis, R. Rastogi, S. Seshadri, and K. Shim. "*XTRACT: A system for extracting document type descriptors from XML documents*", In Proc. ACM SIGMOD Conference: 165 - 176 (2000)

[2] S. Nestorov, J. Ullman, J. Wiener and S. Chawathe, "*Representative Objects : Concise Representations of Semi structured, Hierarchical Data*", In Proc. ICDE: 79–90(1997)

[3] R. Goldman and J. Widom. *"DataGuides: Enabling Query Formulation and Optimization in Semistructured Databases"*. In Proc. 23rd VLDB Conference: 436—445(1997)

[4] T. Milo and D. Suciu. *"Index Structures for Path Expressions"*. In Proc. 7th .ICDT: 277 - 295(1999)

[5] B. Cooper, N. Sample, M. Franklin, G. Hjaltason and M. Shadmon, "*A Fast Index for Semistructed Data*" , In Proc. 14th VLDB conference: 341 – 350(2001)

[6] F. Rizzolo and \A. Mendelzon, "*Indexing XML Data with ToXin*" , in Proc.5th. WebDB conference (2001)

[7] R. Kaushik, P. Shenoy, P. Bohannon, and E. Gudes. "*Exploiting Local Similarity for Indexing Paths in Graph-Structured Data*". In IEEE/ICDE, pages 129-140, San Jose, California, 2002.

[8] Chung, C., Min, J., Shim, K., "*Apex: An adaptive path index for XML data*", In Proc ACM Conference on Management of Data SIGMOD: 121 - 132(2005)

[9] R. Kaushik, P. Bohannon, J. Naughton and H. Korth, "*Covering Indexes for Branching Path Queries*", In Proc. 11rd SIGMOD Conference: 133 – 144(2005)

[10] Chael Ley. "*DBLP Computer Science Biblography*", http://www.informatik.unitrier.de/ley/db/index.html, February (2003)

[11] S. Al-Khalifa, H. V. Jagadish, N. Koudas, J. M. Patel, D. Srivastava, Y. Wu. *"Structural Joins: A Primitive for Efficient XML Query Pattern Matching.*" In Proc. ICDE: 141-152(2002)

[12] Chien et al. "*Efficient Structural Joins on Indexed XML* " , In Proc. VLDB Conference(2002)

[13] C. Zhang, J. Naughton, D. DeWitt, Q. Luo, and G. Lohman "*On supporting containment queries in relational database management systems*", In Proc. 11rd SIGMOD Conference: 133 – 144(2001)

[14] Y. Wu, J. M. Patel, and H. V. Jagadish. "*Structural join order selection for XML query optimization*", In Proceedings of VLDB(2003)

[15] Nicolas Bruno, Nick Koudas, and Divesh Srivastava, "*Holistic Twig Joins: Optimal XML Pattern Matching*", In Proc. SIGMOD Conference: 310–321(2002)

[16] Haifeng Jiang, Wei Wang, Hongjun Lu, and Jeffrey Xu Yu, "*Holistic Twig Joins on Indexed XML*", In Proc. VLDB Conference :273-284 (2003)

[17] R. Kaushik, R. Krishnamurthy, J. Naughton, and R. Ramakrishnan. "*On the integration of structure indexes and inverted lists*", In Proc SIGMOD Conference: 779 - 790 (2002)

[18] B. Yang, M. Fontoura, E. Shekita, S. Rajagopalan, and K. S. Beyer. "*Virtual Cursors for XML Joins*", In CIKM: 523-532(2004)

[19] T. Chen, J. Lu, and T. Ling. "*On boosting holism in XML twig pattern matching using structural indexing techniques*", In Proc. SIGMOD Conference: 455 – 466(2005)

[20] J. Lu, T. Chen, and T. W. Ling, "*Efficient processing of XML twig patterns with parent child edges: a look-ahead approach*". In Proc. CIKM Conference: 533 - 542(2006)

[21] Marcus Fontoura, Vanja Josifovski, Eugene Shekita, and Beverly Yang "*Optimizing Cursor Movement in Holistic Twig Joins*", In Proc. CIKM Conference: 784 – 791(2005)

[22] Jiaheng Lu, Tok Wang Ling, Chee-Yong Chan, and Ting Chen., "*From region encoding to extended dewey: On efficient processing of XML twig pattern matching*". In Proc VLDB Conference: 193 – 204(2005)

[23] Christian Mathis, Theo Härder, Michael Haustein "*Locking-Aware Structural Join Operators for XML Query Processing*", In Proc SIGMOD Conference: 467 - 478 (2006)

[24] Haifeng Jiang, Hongjun Lu, Wei Wang, and Beng Chin Ooi, "*XR-Tree: Indexing XML Data for Efficient Structural Joins.*" In Proc. ICDE Conference :253—264(2003)

زیرنویس‌ها

[1] Path index
[2] Ancestor-descendant
[3] Path expression