2004－DBS－134　(I)　(34)
2004／7／13

構造化文書における関連部分の検索手法とその実装の枠組み

向　井　直　人[†]
黒　田　英　嗣[††]　プラダン スジット[†]

現在，キーワードによる文書の検索手法が最も一般的である．その一方で，論理的要素からなる文書に対しキーワードによる問い合わせを行った際に結果として返される解の単位の問題について，種々の研究が進められている．我々はこの問題について，木構造に基づく代数を用いた新しい問い合わせモデルを提案してきた．本研究においては，これまで提案してきた問い合わせモデルを従来のリレーショナルデータベース管理システムを用いて実装することを目的とする．ある一つの XML 文書の論理的要素を示す各木節点に範囲ラベルづけを行い，それらのラベルを一つのリレーションとして格納しておくことにより，問い合わせモデルで定義された基本的な演算を，このリレーションに対する単純な SQL クエリとして変換することが可能になる．

Retrieval of Relevant Fragments from Structured Documents and a Framework for its Relational Implementation

Naoto Mukai ,[†] , Eiji Kuroda [††] and Sujeet Pradhan [†]

Naive users typically query documents with keywords. The problem of retrieval unit when keyword queries are posed against a structured document consisting of several logical components has been studied in the past. We developed a new query model based on tree algebra, which successfully resolves this problem. However, one important issue any such effective theoretical model has to deal with, is the difficulty in its equally effective implementation. In this paper, we overview our query model and explore how this model can be successfully implemented using an existing relational database technology. Tree nodes representing logical components of a structured document are indexed with their pre-order and post-order rankings and stored as a relation. We then show how the basic algebraic operation defined in our query model can be transformed into a simple SQL query against this relation.

1. はじめに

大部分のデジタル文書 (もしくは単に文書) は多様な要素により構成され，それらの要素には文書内にある内容間の自然な階層関係が存在する．一般にこのような文書は，その階層的構造と内容の順序を保つ木構造 (木) として表すことができる．また，それぞれの要素間の意味的な関連は，それ自身で充足する可能性もあれば，多数により一つの意味をなす可能性もある．

そこで我々は，ある問い合わせに対して文書内の関連部分を解として返す，木構造に基づく代数を用いたモデルを提案および検討してきた[10]．このモデルは，ユーザに文書の構造を意識させないよう，幾つかのキーワードを単に指定することにより問い合わせが可能である一方で，文書の構造を意識した解を結果として返すことができる．さらに，問い合わせに際し用意されたフィルタを用いることにより，問い合わせの解を多様に操作することができる．このようなモデルにより我々は，文書内の関連部分を動的に検索する問い合わせのための，理論的かつ基礎的な手法の提供を行った．

本研究の主題であるモデルの実装においては，文書を XML データであると規定し，その XML データの木 (XML 木) について各節点の位置情報を仮想的な 2 次元平面上に写像，および 1 つのリレーションに格納する．そのうえで，モデル中に定義された演算を，2 次元平面への簡単な領域演算として変換を行う．さらに領域演算を等価な SQL クエリに変換することによって，従来のリレーショナルデータベース管理システム (RDBMS) を用いたモデルの実装が可能であることを示し，なおかつその有効性について述べる．

以下第 2 章では，本研究の主たる動機について述べ，

† 倉敷芸術科学大学　〒 712–8505 岡山県倉敷市連島町西之浦 2640
†† 両備システムズ株式会社　〒 700–8504　岡山県岡山市豊成二丁目 7 番 16 号

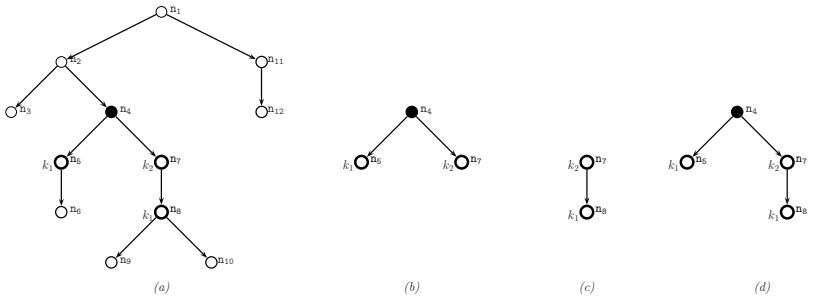

図 1 任意の文書の木と，問い合わせ $\{k_1, k_2\}$ に対し考えられる 3 つの解
Fig. 1 A document tree and three possible answers to the query $\{k_1, k_2\}$

3 章では，我々がこれまで提案してきた問い合わせモデルの概観の説明を行う．そして 4 章では，3 章で説明を行った問い合わせモデルについて，従来の RDBMS を用いて実装するその枠組みを示す．さらに 5 章では，本研究に関連する主要な研究を述べ，最後に 6 章で本論文をまとめる．

2. 検索単位の問題

典型的な構造化文書の意味のある部分は，必ずしも物理的に索引づけられた 1 つの論理的な要素であるとは限らない．このことについて我々は，以下に二つの理由を示す．

(1) 同一文書内に粒度の異なる要素や異種の要素が混在する．さらに各要素は，映画の一場面を表す画像のように自己充足的である可能性があれば，文章の段落のようにそうでない可能性もある．

(2) ある問い合わせに対して，関連する解は複数の要素にわたり分割されている可能性がある．このことについて，例えば問い合わせ $\{k_1, k_2\}$ を図 1 中の文書 (a) に対し行った場合，相互関連した要素からなる多くの直感的な解 (b)(c)(d) が，この問い合わせの解として適切であると考えられる．

このような理由により，キーワードによる問い合わせの単純なブール変換が必ずしも所望の直感的な解を発行できるとは限らないことは明確である．このような純粋なアプローチでは，問い合わせの解を全く発行しないか，あるいは文脈を無視した多数の問い合わせの解を発行するであろう．そのため我々は，問い合わせ中に指定されたキーワードに基き，多様かつ多数の論理的要素からなる全ての直感的な解を動的に構成するような問い合わせモデルを必要とした．

3. モデルの概観

本研究の主題を述べる前に，これまで我々が提案したきた問い合わせモデルにおいて基礎をなすデータモデルの定義，および，そのデータモデルに基づいた問い合わせ手法の定義について，その概観の説明を行う．はじめに，データモデルの説明を行う．

定義 1 (デジタル文書) デジタル文書，もしくは単に文書は，節点集合 $\mathtt{N}$ と枝集合 $\mathtt{E} \subseteq \mathtt{N} \times \mathtt{N}$ を持つ順序木 $\mathcal{D} = (\mathtt{N}, \mathtt{E})$ であり，その他のあらゆる節点に経路 (パス) を持つ根が唯一存在する木である．

文書木の各節点 $\mathtt{n}$ は，文書の各論理的要素と対応づけられる．また，各節点 $\mathtt{n}$ と対応する論理的要素中の代表するキーワードを返す関数 $\mathtt{keywords}(\mathtt{n})$ が別に存在する．さらに，これらの節点は文書のトポロジが失われないように順序づけられる．ここで，全ての節点 $\mathtt{N}$ の代わりに $\mathtt{nodes}(\mathcal{D})$ と表記し，全ての枝 $\mathtt{E}$ の代わりに $\mathtt{edges}(\mathcal{D})$ と表記する．

定義 2 (文書の断片) $\mathcal{D}$ が任意の文書であるとすると，$\mathtt{nodes}(\mathtt{f}) \subseteq \mathtt{nodes}(\mathcal{D})$ および $\mathcal{D}$ の $\mathtt{nodes}(\mathtt{f})$ により誘導された部分グラフが木ならば，$\mathtt{f} \subseteq \mathcal{D}$ は文書の断片，もしくは単に断片である．すなわち，誘導された部分グラフは接続され，特徴づけられた根を持つ木である．

断片は，文書木の節点の部分集合により表すことができる．図 1 中の節点集合 $\langle \mathtt{n4}, \mathtt{n5}, \mathtt{n7}, \mathtt{n8} \rangle$ を含むツリーはサンプル文書の断片であるとし，その根は $\mathtt{n4}$ である．以降，部分集合の表記により示される断片のはじめの節点を，誘導された木の根とする．さらに本論文では，唯一の節点からなる断片を単なる節点であると解釈する．

本章の残りの部分では，上記データモデルに基づいた問い合わせ手法の説明を行う．ここでユーザは，任意のキーワードセット (キーワード集合) を単に指定

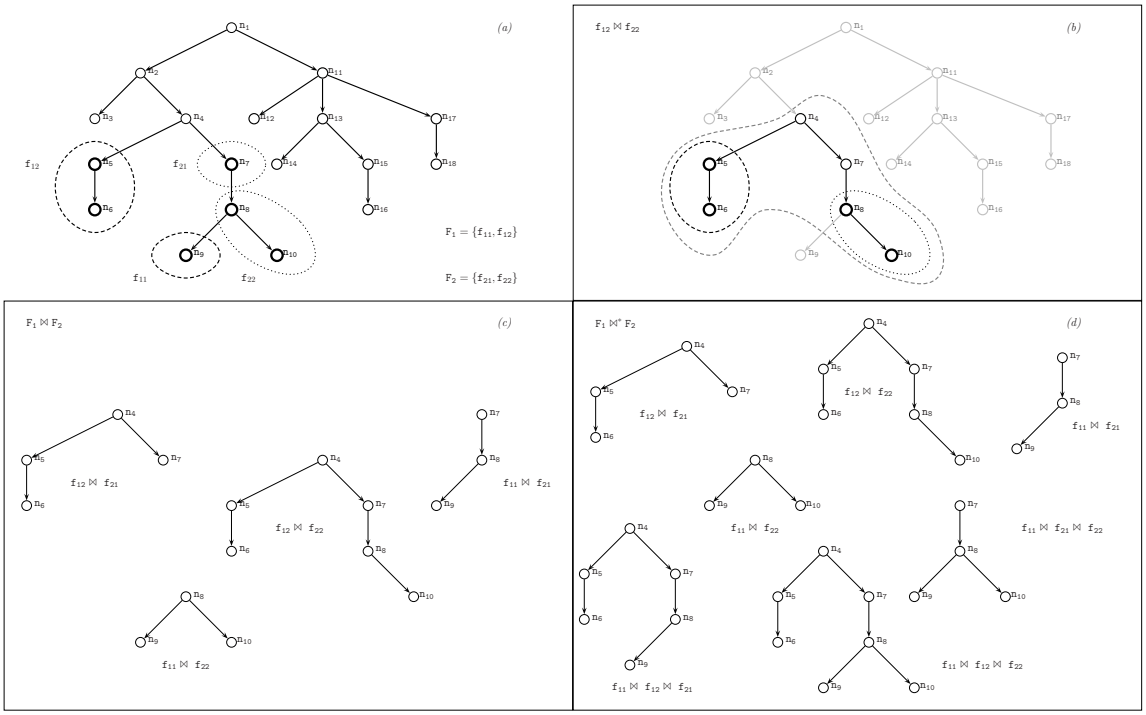

図 2 (a) 任意の文書木 (b) 断片結合 (c) ペアワイズ断片結合 (d) パワーセット断片結合
Fig. 2 (a) A Document Tree (b) Fragment Join (c) Pairwise Fragment Join and (d) Powerset Fragment Join Operations

することにより問い合わせを明確化する．

定義 3 (問い合わせ) ある問い合わせは，$j = 1, 2, \ldots, m$ について $\mathtt{Q} = \{\mathtt{k}_1, \mathtt{k}_2, ..., \mathtt{k}_m\}$ とする．このとき，$\mathtt{k}_j$ は問い合わせの各キーワードである．

ここでは，節点 $\mathtt{n}$ に対応づけられた文書の内容に問い合わせの任意のキーワード $\mathtt{k}$ が現れることを，$\mathtt{k} \in \mathtt{keywords(n)}$ と表記する．

定義 4 (問い合わせの解) ある問い合わせ $\mathtt{Q} = \{\mathtt{k}_1, \mathtt{k}_2, ..., \mathtt{k}_m\}$, を考えるとき，この問い合わせに対する解 $\mathtt{A}$ は $\{\mathtt{f} \mid (\forall \mathtt{k} \in \mathtt{Q}).\exists \mathtt{n} \in \mathtt{f} : \mathtt{k} \in \mathtt{keywords(n)}\}$ を満足する断片の集合である．

次に，直感的な解を求めるために用いる，断片集合に対する演算の定義をいくつか示す．

定義 5 (選択) 任意の文書の断片集合を $\mathtt{F}$ であるとし，$true$ あるいは $false$ に各断片を写像する述語を $\mathtt{P}$ であるとすると，述語 $\mathtt{P}$ による F からの選択は，$\mathtt{F}$ の部分集合 $\mathtt{F}'$ となる．$\mathtt{F}'$ は $\mathtt{P}$ を満足する全てかつ唯一の断片を含む．また，$\mathtt{F}$ からの選択は $\sigma_{\mathtt{P}}$ により示され，$\sigma_{\mathtt{P}}(\mathtt{F}) = \{\mathtt{f} \mid \mathtt{f} \in \mathtt{F}, \mathtt{P}(\mathtt{f}) = true\}$ となる．

以降，述語 $\mathtt{P}$ は選択 $\sigma_{\mathtt{P}}$ のフィルタと呼ばれる．最も単純なフィルタはキーワード 'k' のみを持つ断片を選択する 'keyword $= k$' 形式のフィルタであり，キーワード選択のために存在する．別の 'size $< c$' 形式のフィルタは断片のサイズのコントロールを行う．ここで断片のサイズは，それ自身に含まれる節点の数により測定される．我々はこの他に，より実用的ないくつかのフィルタの提案を行ったが，本論文では省略する．

定義 6 (断片結合) 仮に $\mathtt{f}_1, \mathtt{f}_2, \mathtt{f}$ を文書 $\mathcal{D}$ の任意の断片であるとする．このとき，$\mathtt{f}_1 \bowtie \mathtt{f}_2$ により表される，$\mathtt{f}_1$ と $\mathtt{f}_2$ の間で結合された断片が $\mathtt{f}$ であれば，以下が全て成り立つ．

(1) $\mathtt{f}_1 \subseteq \mathtt{f}$,
(2) $\mathtt{f}_2 \subseteq \mathtt{f}$,
(3) $\mathtt{f}' \subseteq \mathtt{f} \wedge \mathtt{f}_1 \subseteq \mathtt{f}' \wedge \mathtt{f}_2 \subseteq \mathtt{f}' \Rightarrow \nexists \mathtt{f}'$

ここで，任意の断片 $\mathtt{f}_1$，$\mathtt{f}_2$，$\mathtt{f}_3$ について断片結合は以下のような代数的特性を持つ．

等冪性 $\mathtt{f}_1 \bowtie \mathtt{f}_1 = \mathtt{f}_1$
可換性 $\mathtt{f}_1 \bowtie \mathtt{f}_2 = \mathtt{f}_2 \bowtie \mathtt{f}_1$
結合性 $(\mathtt{f}_1 \bowtie \mathtt{f}_2) \bowtie \mathtt{f}_3 = \mathtt{f}_1 \bowtie (\mathtt{f}_2 \bowtie \mathtt{f}_3)$
吸収性 $\mathtt{f}_1 \bowtie (\mathtt{f}_2 \subseteq \mathtt{f}_1) = \mathtt{f}_1$

これらの特性は演算の容易な実装を可能にするだけではなく，先に述べる問い合わせの計算コストを軽減する．

定義 7 (ペアワイズ断片結合) 文書 $\mathcal{D}$ の任意の断片集合 $\mathtt{F}_1$ と $\mathtt{F}_2$ があるとすると，$\mathtt{F}_1$ と $\mathtt{F}_2$ のペアワイズ断片結合は $\mathtt{F}_1 \bowtie \mathtt{F}_2$ により表され，それは $\mathtt{F}_1$ と

F_2 それぞれの要素同士のあらゆる組み合わせの断片結合を取ることにより誘導される断片集合であり，形式的には次の通りである．

$$\mathtt{F}_1 \bowtie \mathtt{F}_2 = \{\mathtt{f}_1 \bowtie \mathtt{f}_2 \mid \mathtt{f}_1 \in \mathtt{F}_1,\ \mathtt{f}_2 \in \mathtt{F}_2\}$$

ここで，任意の断片集合 $\mathtt{F}_1, \mathtt{F}_2, \mathtt{F}_3$ についてペアワイズ断片結合は以下のような代数的特性を持つ．

可換性 $\mathtt{F}_1 \bowtie \mathtt{F}_2 = \mathtt{F}_2 \bowtie \mathtt{F}_1$

結合性 $(\mathtt{F}_1 \bowtie \mathtt{F}_2) \bowtie \mathtt{F}_3 = \mathtt{F}_1 \bowtie (\mathtt{F}_2 \bowtie \mathtt{F}_3)$

不動点 $\mathtt{F}_1 \bowtie \mathtt{F}_1 = (\mathtt{F}_1 \bowtie \mathtt{F}_1) \bowtie \mathtt{F}_1$

ただし，等冪性は成り立たないことに注意する．

定義 8 (パワーセット断片結合) 文書 $\mathcal{D}$ の任意の断片集合 $\mathtt{F}_1$ と $\mathtt{F}_2$ があるとすると，$\mathtt{F}_1$ と $\mathtt{F}_2$ のパワーセット断片結合は $\mathtt{F}_1 \bowtie^* \mathtt{F}_2$ により表され，それは $\mathtt{F}_1$ 内および $\mathtt{F}_2$ 内に含まれる要素の任意数 (ただし，0 ではない) に対し断片結合を適用することにより誘導される断片集合である．形式的には次の通りである．

$$\mathtt{F}_1 \bowtie^* \mathtt{F}_2 = \{\bowtie (\mathtt{F}'_1 \cup \mathtt{F}'_2) \mid \mathtt{F}'_1 \subseteq \mathtt{F}_1, \mathtt{F}'_2 \subseteq \mathtt{F}_2, \mathtt{F}'_1 \neq \phi, \mathtt{F}'_2 \neq \phi\}$$

$$\text{ただし，} \bowtie \{\mathtt{f}_1, \mathtt{f}_2, \ldots, \mathtt{f}_n\} = \mathtt{f}_1 \bowtie \ldots \bowtie \mathtt{f}_n$$

上記の定義は次のように展開できる．

$$\begin{aligned}\mathtt{F}_1 \bowtie^* \mathtt{F}_2 \quad = \quad & (\mathtt{F}_1 \bowtie \mathtt{F}_1) \cup \\ & (\mathtt{F}_1 \bowtie \mathtt{F}_1 \bowtie \mathtt{F}_2) \cup (\mathtt{F}_1 \bowtie \mathtt{F}_2 \bowtie \mathtt{F}_2) \cup \\ & (\mathtt{F}_1 \bowtie \mathtt{F}_1 \bowtie \mathtt{F}_1 \bowtie \mathtt{F}_2) \cup \\ & (\mathtt{F}_1 \bowtie \mathtt{F}_1 \bowtie \mathtt{F}_2 \bowtie \mathtt{F}_2) \cup \\ & (\mathtt{F}_1 \bowtie \mathtt{F}_2 \bowtie \mathtt{F}_2 \bowtie \mathtt{F}_2) \cup \ldots\end{aligned}$$

さらにパワーセット断片結合は，断片結合とペアワイズ断片結合双方が持つ代数的特性を用いることにより，次の等価表現に変形することができる．

$$\mathtt{F}_1 \bowtie^* \mathtt{F}_2 = (\mathtt{F}_1 \bowtie \mathtt{F}_1) \bowtie (\mathtt{F}_2 \bowtie \mathtt{F}_2)$$

図 2 に，本章中でこれまで示した演算のいくつかを示す．(a) は演算の対象である，断片集合 $\mathtt{F}_1$ と $\mathtt{F}_2$ を持つ任意の文書木を示す．そして (b) は，(a) 中の文書木において，(a) 中の断片集合 $\mathtt{F}_1$ と $\mathtt{F}_2$ を入力とした断片結合を行った際の解を示す．同様に，(c) はペアワイズ断片結合を行った際の解を示し，(d) はパワーセット断片結合を行った際の解を示す．

ここで，上記の変形後のパワーセット断片結合は，ある問い合わせに対し関連する全ての断片を計算するために実際に用いる演算であることを特に注意されたい．このことは，多項式時間内の解の計算を上記の変形が可能にすることを明らかにしている．

本章でこれまで説明を行った本モデルを用いることにより，例えば $\{\mathtt{k}_1, \mathtt{k}_2\}$ により表される問い合わせは次式により評価することができる．

$$\begin{aligned}\mathtt{Q} = \{\mathtt{k}_1, \mathtt{k}_2\} &= \sigma_{\mathsf{keyword}=k_1}(\mathtt{F}) \bowtie^* \sigma_{\mathsf{keyword}=k_2}(\mathtt{F}) \\ &= (\mathtt{F}_1 \bowtie \mathtt{F}_1) \bowtie (\mathtt{F}_2 \bowtie \mathtt{F}_2)\end{aligned}$$

ただし，$\mathtt{F}_1 = \sigma_{\mathsf{keyword}=k_1}(\mathtt{F})$ および $\mathtt{F}_2 = \sigma_{\mathsf{keyword}=k_2}(\mathtt{F})$.

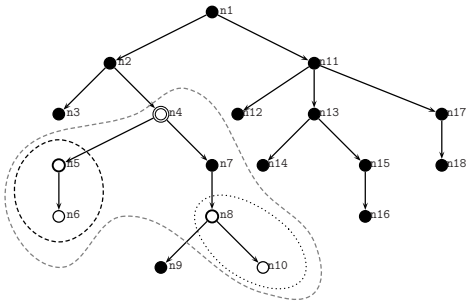

図 3 任意の構造化文書を表す XML 木
Fig. 3 An XML Tree representing a structured document

4. モデルの実装

本論文の主題である本章では，これまで説明を行ったモデルについて，その実装を従来の RDBMS を用いて行う枠組みについて詳述する．

はじめに，本研究においては扱う文書を具体的に XML データであると規定する．XML データは至るところに存在し，そして幅広く研究が進められている．XML データを扱う利点として，実装に際しデータを統一的に扱えること，そのデータ構造が我々の提案してきたモデルに合致することが挙げられる．

具体的に，本研究では XML データを範囲ラベルづけ手法[3] を用いて，XML 木の各節点に先行順木走査 (preorder traversal) と後行順木走査 (preorder traversal) それぞれの順序によりラベルづける．このような手法を用いることにより，XML 木のある任意の節点は，位置情報として X 軸に先行順木走査によるラベル，Y 軸に後行順木走査によるラベルを持つ仮想的な 2 次元平面上に対して容易に写像できる．さらに 2 次元平面は，ある任意の節点に着目したとき，その節点を通るよう X 軸 Y 軸双方から垂線を引くことにより 4 つ領域が形成され，その節点の先祖の節点，子孫の節点，データ内の順序においてその節点より前にあり，なおかつ先祖を除く節点 (preceding)，データ内の順序においてその節点より後にあり，なおかつ子孫を除く節点 (following) をそれぞれ含む領域を持つ．

ここで，各節点につけられた 2 種類のラベルは，RDBMS における一つのリレーションとして格納される．以降では，このリレーションに対して SQL クエリによる問い合わせを行い，モデルの演算結果と同

様の解を求める．

これまでの説明について図解する．図 3 におけるサンプルの XML 木に対して範囲ラベルづけを行い，そのラベルに基づいて 2 次元平面上への写像を行った結果が図 4 に分布する点である．ここで，上記にて説明を行った 4 つの領域について注意する．図 5 は，図 3 のそれぞれの節点と，対応するラベルの情報を格納したリレーションである．

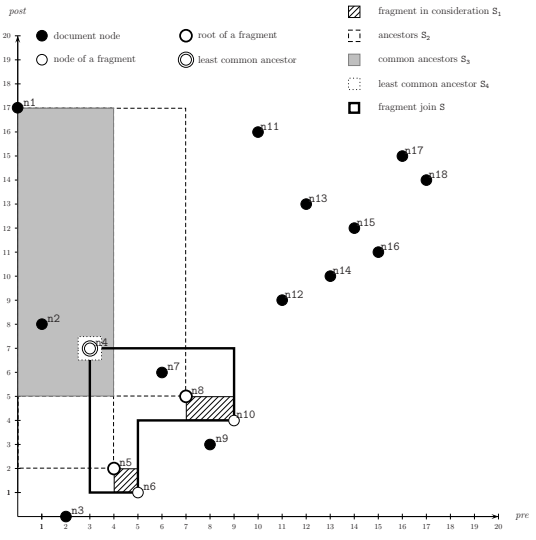

図 4 先行順/後行順木走査による平面上の断片結合
Fig. 4 Fragment Join Operation on Pre-post plane

4.1 等価な SQL 表現

範囲ラベルづけ手法を用いることは，問い合わせに対する解の計算コストを軽減することに繋がる．もしもこれらの手法を用いずモデルの実装を行った場合，複雑な木走査アルゴリズムを用いて解を求める必要が出るであろう．逆に，これらの手法を用いてモデルの実装を行えば，以下のような理解しやすい解釈ができ，モデルの各演算に等価な SQL クエリへの変換も容易に行える．

さて，これまで説明を行ってきたモデルのキーワード選択演算は理解しやく実装も容易であるが，結合演算は複雑であるため，実装に際し特別の注意を払う必要がある．そのため我々は，モデルの主要な演算である断片結合についてのみ，それと等価な 2 次元平面上の領域演算への変換方法を述べる．ペアワイズ断片結合とパワーセット断片結合については，双方が断片結合の変形であることを理由に説明を省略し，本章末にて実装に際し注意すべき事柄を述べるのみにとどめる．

基本的に任意の断片間の断片結合は，最小かつ共通の先祖に行き着くまで，それぞれの先祖の節点を検出していくことを意味する．したがって断片結合は，2 次元平面上において，入力する 2 つの断片共通の左上の領域に節点を検出することにより定義できる (図 4 参照)．断片結合を評価するため，我々は以下の，3 つの領域と 1 つの節点に対する問い合わせを行う必要がある．

(1) 2 つの入力断片の節点を全て含む最小領域を集合 $\mathtt{S}_1$ により表す．図 4 中の斜線の領域.

(2) 2 つの入力断片それぞれの先祖となる全ての節点を含む最小領域を集合 $\mathtt{S}_2$ により表す．図 4 中の破線により囲まれた領域.

(3) 2 つの入力断片双方の共通の先祖となる全ての節点を含む最小領域を集合 $\mathtt{S}_3$ により表す．図 4 中の灰色の領域.

(4) 2 つの入力断片から最も近い双方共通の先祖となる節点を集合 $\mathtt{S}_4$ により表す．図 4 中の点線により囲まれた領域.

	pre	*post*
n1	0	17
n2	1	8
n3	2	0
n4	3	7
n5	4	2
n6	5	1
n7	6	6
n8	7	5
n9	8	3
n10	9	4
n11	10	16
n12	11	9
n13	12	13
n14	13	10
n15	14	12
n16	15	11
n17	16	15
n18	17	14

図 5 ラベルのリレーション
Fig. 5 Relation of pre-post

以上により，断片結合と等価な SQL クエリは，和集合と差集合を用いた一般的な集合演算によって次のように表現できる．

$$\mathtt{S} = ((\mathtt{S}_1 \cup \mathtt{S}_2) - \mathtt{S}_3) \cup \mathtt{S}_4$$

例として，2 つの入力断片 $\langle \mathtt{n5}, \mathtt{n6} \rangle$ と $\langle \mathtt{n8}, \mathtt{n10} \rangle$ は図 4 における斜線の領域となり，2 つの断片間の断片結合の結果は (図 4 の) 太線により囲まれた領域に含まれる $\langle \mathtt{n4}, \mathtt{n5}, \mathtt{n6}, \mathtt{n7}, \mathtt{n8}, \mathtt{n10} \rangle$ となる．ここで，擬似的な断片結合である SQL クエリは，図 5 のリレーションに対し問い合わせを行い，その際に用いられる SQL クエリは，例えば図 6 のようなものとなるであろう．また，このときの $\mathtt{S}_1$,$\mathtt{S}_2$,$\mathtt{S}_3$,$\mathtt{S}_4$ は以下となる．

(1) $\mathtt{S}_1 = \{4, 5, 7, 9\}$.

(2) $S_2 = \{0, 1, 3, 6\}$.
(3) $S_3 = \{0, 1, 3\}$.
(4) $S_4 = \{3\}$.

これらに対し先に示した集合演算式を適用すると以下となる．

$$S = ((S_1 \cup S_2) - S_3) \cup S_4 = \{3, 4, 5, 6, 7, 9\} \equiv \langle n4, n5, n6, n7, n8, n10 \rangle$$

ここで，図 2(b) と図 3 中に示された断片結合の結果と上記の結果を見比べると，より直感的に等価性を理解できるであろう．

以上のような手法により，モデルの断片結合と等価な SQL クエリへの変換は容易に行えた．そして，このような断片結合と等価な SQL クエリを用いて，さらにペアワイズ断片結合およびパワーセット断片結合と等価な SQL クエリへの変換を行える．以上により，従来の RDBMS を用いたモデルの実装が可能であり，同時にその枠組みが示されたこととする．

最後に，ペアワイズ断片結合とパワーセット断片結合それぞれに等価な SQL クエリへの変換を行ったとし，パワーセット断片結合と等価な SQL クエリによる問い合わせを行う場合に，その結果として返されるものが断片の集合であることに注意する必要がある．このとき，どの断片が所望するものであるかを選択する必要があると思われるが，それは単純な述語あるいは SQL 標準の組み込み関数を用いることにより可能であることを述べておく．

5. 関 連 研 究

近年，従来のデータベース管理システムに XML データを格納し索引づけることについて，その関心が高まりつつある．多くの研究は，XML データのコーディングを行うための多様な手法を提案している[13)5)8)14)]．[8)] による研究は，正則経路式のシーケンスに対しての効率的な利用方法に焦点を合わせる一方で，[13)] はその順序付けに重点をおく．また，リレーショナルデータベースに XML データを格納するための，経路に基づく索引付け，および SQL クエリに相当する XPath 問い合わせ表現への変換は[14)] に提案されてきた．我々の問い合わせ手法におけるモデル実装は主として[5)6)] に紹介された研究により端を発するが，同文献中で検討された主要な点は，XPath による問い合わせ評価を利用可能にする，データベースの索引付け構造である．このことは，我々が本論文中でアドレスを用いてきたこととの相違点を明らかにしている．さらに[15)] による研究は，構造化文書からキーワード間の関係を考慮した部分検索を行うが，文書木内のパスを用いる点で我々の研究と異なる．

構造化文書に関するデータベースシステムの研究は，これまで[11)] で行われてきた．また多数の研究は，領域代数に基づいた，構造化文書からの論理的単位の検索に関して行われてきた[12)1)2)9)7)]．しかし，これらの研究の多くは，テキスト検索における内容と構造の統合に関連し，さらに従来の XML 問い合わせにキーワード問い合わせを統合する提案である[4)]．データベースの形式的なアプローチにより構造化文書の適切な断片を検索するとき，キーワードのみに基づく問い合わせ形態をとる研究は少ない．

6. ま と め

本論文では，これまで我々が提案してきた問い合わせモデルの説明を行い，さらに従来の RDBMS を用いてモデルを実装する枠組みを示し，その実現性および有効性についても述べた．

今後は，本論文で述べたことについてさらなる検討および改良を重ね，そのうえで RDBMS を用いた実際の実装を行う予定である．

参 考 文 献

1) Forbes J. Burkowski. Retrieval activities in a database consisting of heterogeneous collections of structured text. In *Proc. of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 112–125. ACM Press, 1992.
2) Charles L. A. Clarke, G. V. Cormack, and F. J. Burkowski. An algebra for structured text search and a framework for its implementation. *The Computer Journal*, 38(1):43–56, 1995.
3) Edith Cohen, Haim Kaplan, and Tova Milo. Labeling dynamic XML trees. In *PODS*, 2002.
4) D. Florescu, D. Kossman, and I. Manolescu. Integrating keyword search into XML query processing. In *International World Wide Web Conference*, pages 119–135, 2000.
5) Torsten Grust. Accelerating XPath location steps. In *Proc. of the 2002 ACM SIGMOD International Conference on Management of Data*, pages 109–120. ACM, June 2002.
6) Torsten Grust, Maurice van Keulen, and Jens Teubner. Staircase join: Teach a relational DBMS to watch its (axis) steps. In *Proc. of the 29th VLDB Conference*, pages 524–535, September 2003.
7) Jani Jaakkola and Pekka Kilpelainen. Nested text-region algebra. Technical Report C-1999-2, Department of Computer Science, Univer-

```
1  SELECT pre
2  FROM tree
3  WHERE pre IN
4    (
5      (nodes(f1)) OR (nodes(f2))
6    )
7  OR
8    (
9      (pre < pre(root(f1)) AND post > post(root(f1)))
10   OR
11     (pre < pre(root(f2)) AND post > post(root(f2)))
12   )
13 AND pre NOT IN
14   (
15   SELECT pre
16   FROM tree
17   WHERE
18         (pre < pre(root(f1)) AND post > post(root(f1)))
19     AND
20         (pre < pre(root(f2)) AND post > post(root(f2)))
21   )
22 OR pre IN
23   (
24   SELECT max(pre)
25   FROM tree
26   WHERE
27         (pre < pre(root(f1)) AND post > post(root(f1)))
28     AND
29         (pre < pre(root(f2)) AND post > post(root(f2)))
30   )
31 ORDER BY pre
```

図 6 任意の 2 つの断片`f1` と`f2` 間の断片結合と等価な SQL 表現
Fig. 6 SQL equivalent expression for fragment join operation between two arbitrary fragments `f1` and `f2`

sity of Helsinki, January 1999. Available at http://www.cs.helsinki.fi/TR/C-1999/2/.

8) Quanzhong Li and Bongki Moon. Indexing and querying XML data for regular path expressions. In *Proc. of 27th International Conference on Very Large Data Bases*, pages 361–370. Morgan Kaufmann, September 2001.

9) G. Navarro and R.A. Baeza-Yates. Proximal nodes: A model to query document databases by content and structure. *ACM Transactions on Information Systems*, 15(4):400–435, 1997.

10) Sujeet Pradhan and Katsumi Tanaka. Retrieval of relevant portions of structured documents. In *DEXA*, 2004.

11) R. Sacks-Davis, T. Arnold-Moore, and J. Zobel. Database systems for structured documents. In *International Symposium on Advanced Database Technologies and Their Integration*, pages 272–283, 1994.

12) A. Salminen and F. Tompa. Pat expressions: an algebra for text search. *Acta Linguistica Hungar*, 41(1-4):277–306, 1992.

13) I. Tatarinov, S. Viglas, K. S. Beyer, J. Shanmugasundaram, E. J. Shekita, and C. Zhang. Storing and querying ordered XML using a relational database system. In *Proc. of the 2002 ACM SIGMOD International Conference on Management of Data*, pages 204–215. ACM, June 2002.

14) Masatoshi Yoshikawa, Toshiyuki Amagasa, Takeyuki Shimura, and Shunshuke Uemura. XRel: a path-based approach to storage and retrieval of XML documents using relational databases. *ACM Transactions on Internet Technology*, 1(1):110–141, August 2001.

15) 依田平, 大月一弘, 清光英成, and 森下淳也. ツリー型不定形文書からの部分文書の検索手法の検討. In **第** *14* **回データ工学ワークショップ** *(DEWS2003)*, 2003.