データベースシステム 116－68
（1998. 7. 10）

DB要素検索のための実体インデクス

村田 美友紀† 掛下 哲郎††
† 八代工業高等専門学校 情報電子工学科
†† 佐賀大学理工学部 知能情報システム学科

情報化技術の発展に伴い，データベース (DB) を構成するクエリー，ビューなどのDB要素も複雑化している．DBシステムの開発コストや変更コストを低減するためにはDB要素の再利用が不可欠である．このために，DB要素の検索機構が必要になる．我々は実体をサンプルとして用いるDB要素の検索機構を提案している．本手法は，仕様に基づいた検索に対応している．本稿では，各DB要素を唯一に識別できる極小の実体集合を用いて実体インデクスを定義し，実体インデクスの効率的な構成法を提案する．実体インデクスはメモリに常駐できるため，サンプル構築を高速化する．また，検索対象のDB要素を唯一に識別するサンプル構築が常に行えるため，DB要素検索の手間を低減する．最小の実体インデクス構成問題はNP完全になるが，DB要素の集合をGとしたとき，サイズ$|G|-1$以下の実体インデクスが多項式時間で構成できる．また，Gに対する要素の追加，削除に対応した実体インデクスの更新アルゴリズムを提案する．

Entity Index for Database Component Retrieval

Miyuki Murata† Tetsuro Kakeshita††
†Department of Infomation and Electronics Engineering,
Yatsushiro National College of Technology
††Department of Information Science, Saga University

Database components, such as database query and view, are getting complicated according to the advancement of the information technology. Database components must be reused in order to decrease development cost of database systems. Then a retrieval mechanism becomes necessary for such components. We have proposed a mechanism using entities as samples to describe specification of a database component. In this paper, we define entity index as a minimal set of entities such that each component can be uniquely identified by the elements of the set. We also propose an efficient development algorithm of the entity index. Sample constraction becomes fast since the entity index can be kept in main memory. Furthermore retieval cost of a database component can be reduced since a sample can be constructed to uniquely identify each database component. Although the construction problem of the minimum entity index is NP complete, an entity index of size $|G|-1$ can be constructed in polymonial time for an arbitrary set G of database components. We also develop reorganization algorithms of the entity index when a database component is added to or deleted from G.

1 はじめに

情報化社会の発展に伴いDBの複雑化が進行する中で，質問，ビュー，一貫性制約，プログラム等のDB要素も複雑になりつつある．DBの開発や保守，既存DB上でのシステム開発を効率的に行なうためにはDB要素の再利用が不可欠である．DB要素を再利用する際にはその検索機構が必要となる．DB要素は機能に対応して開発されるため，仕様に基づいたDB要素の検索機能を提供する必要がある．そこで，我々は集合を用いて各DB要素の仕様を統一的に表現し，集合を検索するためにサンプルを活用する機構を提案した[1, 2, 3]．サンプルは実体の集合であり，サンプル中の実体は正例か負例かが決定されている．サンプルは正例をすべて含み，いずれの負例も含まない集合を検索する．サンプルは検索の目的である集合の特徴を表現できるため，本機構は仕様に基づいたDB要素の検索に対応している．利用者はサンプルの改良を繰り返し目的の集合を検索する．集合を絞り込むためには適切な実体をサンプルに追加すれば良い．また，サンプルに追加する実体を検索する方法として，論理式による検索と実体の重複度別分類を提案している．

DBに格納されている実体は変更される可能性を常に持つ．このため，サンプルを作成する際に，目的の集合を唯一に識別できる実体がDB上に常に存在するとは限らない．また，DB上のすべての実体をサンプルへの追加候補とすると，必要な実体を検索するための手間が大きくなる可能性もある．これに対応するために，サンプル構築に必要な実体集合を作成し，DBとは別に保存する方法が考えられる．

我々は，このような実体の集合を実体インデクスと呼ぶ．本稿では，実体インデクスの性質と構成方法について考察する．実体インデクスを用いると，各DB要素を唯一に識別するサンプルを構成できるため，検索漏れを防げる．実体インデクスとDBを分離することによって，実体インデクス中の実体はDBの操作の影響を受けない．実体インデクスが小さければメモリに常駐でき，検索の高速化を図ることができる．また，実体インデクスを用いることにより，サンプルに追加する実体の検索領域を縮小できる．

本稿は以下のように構成されている．2節では，集合を用いてDB要素をモデル化した後，サンプルを用いた集合検索機構を示す．3節では，実体インデクスを定義した後，その性質について考察する．4節では，複数の集合を識別するための最小の実体インデクス構成問題のNP完全性を示す．5節では，検索対象の集合を追加，削除する際に適切な実体を実体インデクスに追加または削除するための多項式時間アルゴリズムを提案する．このアルゴリズムを用いると，DB要素の集合 G に対して，要素数が高々$|G|-1$の実体インデクスを構成できる．

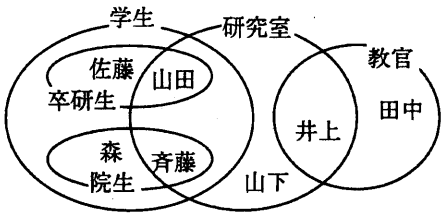

図 1: サンプルを用いた集合検索の例

2 サンプルを用いた集合検索機構

DB要素には，質問，ビュー，データマイニング[4, 5]によって得られた知識，プログラムなどがある．質問やビューは，それらを満足する実体の集合によって表現できる．知識はDBの部分集合に対応する．このため，DB要素の仕様は集合を用いて表現できる．

サンプル Sは実体の集合であり，各要素は正例または負例のいずれかが明示されている．サンプル Sが集合 gを検索するとき，S中のすべての正例 eに対し $e \in g$，かつ S中のすべての負例$\bar{e}$に対し$\bar{e} \notin g$である．候補集合 $G = \{g_1, g_2, \cdots, g_n\}$ について，$Set(G,S) = \{g \in G | S$は gを検索する $\}$ を定義する．$Set(G,S)$ は Sによる G の絞り込みの結果である．

例 1 図1に集合 G ={ 学生, 卒研生, 院生, 研究室, 教官 } と各集合に属する実体を示している．G に対して，サンプル S_1 ={ 山田 } を用いて検索を行なうと，$Set(G,S_1)$ ={ 学生, 卒研生, 研究室 } となる．G に対して，S_1に'斉藤'を追加したサンプル S_2 = { 山田, 斉藤 } を用いて検索すると，$Set(G,S_2)$ = { 学生, 研究室 } となる．さらに負例'$\overline{\text{森}}$'を追加したサンプル S_3 = { 山田, 斉藤, $\overline{\text{森}}$ } を用いて検索すると，$Set(G,S_3)$ = { 研究室 } となり，集合が唯一に決定する． □

サンプル Sによって検索される集合 $Set(G,S)$ には以下の性質が成り立つ．ここで，Gは集合，S_1, S_2はサンプルとする．

性質 1

$Set(G, S_1 \cup S_2) = Set(G, S_1) \cap Set(G, S_2 - S_1)$ □

性質 2

$Set(G, S_1 \cup S_2) = Set(Set(G, S_1), S_2 - S_1)$ □

性質1は，サンプルに含まれる実体数が多いほど絞り込みの効果が高いことを示している．さらに追加する実体集合と追加前のサンプルが互いに素ならば，絞り込みの効果が最も高いことも分かる．性質2より，サンプルに実体集合を追加した後の集合検索は，前段階で絞り込まれた集合に対してのみ行えばよいことが分かる．

本機構を用いた一般的な集合検索はサンプルを改良することにより，対話的に行なう．性質1より，サンプルに実体を追加することによって，候補集合数が減少する．よって検索の目的とする集合 (目的集合) を唯一に決定するためのサンプルの改良法には，サンプルへの実体の追加を用いる．追加する実体を適切に選択する方法として論理式による実体検索と実体の重複度別分類がある．

論理式による実体検索では，サンプルに追加する実体を論理式を用いて検索する．論理式 L によって定義される集合を S_L，候補集合を $g_1, \cdots, g_n$，目的集合を gとする．正例を追加する時には $S_L \subseteq (g - \cap g_i)$，負例を追加する時には，$S_L \subseteq (\cup g_i - g)$ を満足する L を作成すればよい．L によって検索される実体を追加することにより，候補集合を絞り込むことができる．ここで，$(g - \cap g_i) \cup (\cup g_i - g) = \phi$ の場合には，追加するのに適切な実体が存在しない．

実体 e_k について G に対する e_kの重複度 $W_k = |\{g \in G | e_k \in g\}|$ を定義する．実体の重複度別分類では，$\cup g_i$の要素中から複数個の実体を選択し，それを重複度順に並べ替えたリストを実体リストと呼ぶ．重複度0および $|G|$ の実体は集合を絞り込めないため，実体リストに含めない．利用者は実体リストの中から適当な実体を1つ選択し，サンプルに追加する．ここで，実体リストが空の場合には，実体の選択ができない．また，実体リストに含まれる実体を検索するのに要するコストは $O(|\cup g_i|)$ となるため，$\cup g_i$が大きくなるに従ってサンプル構成コストが大きくなる．

集合	サンプル
学生	{ 山田, 斉藤, 井上}
卒研生	{ 山田, 斉藤, 井上}
院生	{山田, 斉藤, 井上}
研究室	{ 山田, 斉藤, 井上 }
教官	{山田, 斉藤, 井上 }

表 1: 実体インデクスの例

3 実体インデクス

論理式による実体検索や実体の重複度別分類をDB中の実体に直接適用すると，サンプルを構築するために適切な実体が存在しない場合や，実体検索コストが高くなる場合がある．これを解決するために，DB化された任意の集合を唯一に識別できる実体集合 (実体インデクス) を定義し，その性質について考察する．

定義 1 実体インデクス E_Gは集合 $G = \{g_1, \cdots, g_n\}$ に対して定義される．E_Gは実体の集合である．G 中の任意の集合を唯一に識別するサンプルは，E_G 中の要素を用いて作成できる． □

以下に実体インデクスの例を示す．

例 2 図1の集合 G を考える．G に対して定義される実体インデクスは，{ 山田, 斉藤, 井上 } である．集合とそれを識別するサンプルを表1に示す． □

G の実体インデクスは一般に複数存在する．すべての $E \subset E_G$が G の実体インデクスでないとき，E_Gを極小の実体インデクスと呼ぶ．また，集合 G の領域集合 A_Gを以下のように定義する．各領域は，特定の $G' \subseteq G$ に属する集合のみに含まれる要素の集合である．

$$A_G = \left\{ \bigcap_{g \in G'} g - \bigcup_{g \in G - G'} g \;\middle|\; G' \subseteq G \right\}$$

実体インデクスに対して以下の補題が成立する．

補題 1 極小の実体インデクス E_Gの各要素は A_Gにおいて互いに異なる領域に属する． □

[証明] E_Gの要素 e_i, e_jがともに同一領域に含まれると仮定する．このとき $e_i \in g_k$ $(e_i \notin g_k)$ ならば $e_j \in g_k$ $(e_j \notin g_k)$ であるため，e_iと e_iが区別する集合は等しい．よって，$E_G - \{e_i\}$，$E_G - \{e_j\}$ も G の実体インデクスとなり仮定に矛盾する． □

補題 2 極小の実体インデクス E_Gに属する実体の G に対する重複度は 1 以上，$|G|-1$ 以下である．□

[証明] 重複度 0 の実体はすべての集合 $g \in G$ に含まれない．また，重複度 $|G|$ の実体はすべての集合 $g \in G$ に含まれる．このため，これらの要素は G の要素を区別できないため，E_Gより削除できる．□

集合 g, g'について，$diff(g,g') = (g-g') \cup (g'-g)$ を定義する．

補題 3 G の任意の 2 つの要素 g_i, g_j $(g_i \neq g_j)$ について，実体インデクス E_Gは実体 $e \in diff(g_i, g_j)$ を含む．□

[証明] G の要素 g_i, g_jについて，E_Gが $e \in diff(g_i, g_j)$ を含まないと仮定する．仮定より，E_G のすべての要素について $e \in g_i$ならば $e \in g_i \cap g_j$ より $e \in g_j$である．$e \notin g_i$ならば $e \in \cup g_k - (g_i \cup g_j)$ より $e \notin g_j$である．よって，E_Gを用いて作成された g_iを検索するサンプルはすべて g_j を検索する．これは定義に反する．□

補題3は逆も成立する．すなわち集合 $E = \{e | e \in diff(g_i, g_j), g_i, g_j \in G, g_i \neq g_j\}$ は G の実体インデクスである．ただし Eの要素数は $O(|G|^2)$ となる．

補題 4 実体インデクス E_Gのすべての要素を用いて作成されたサンプルは G 中で高々1 個の集合を検索する．□

[証明] 実体インデクスの定義より，任意の集合は E_Gの要素を用いたサンプルによって唯一に識別できる．g_iを唯一に識別するサンプル Sが E $(\subseteq E_G)$ のすべての要素を用いて構成されるとする．このとき，$E_G - E$ の各要素について g_i の正例か負例かが一意に決定される．よって，g_i を識別する E_G の要素をすべて用いたサンプル S'がただ 1 個作成できる [1]．□

補題 4より，E_G中のすべての実体を用いて作成されたサンプル Sに対して，$|Set(G,S)| = 0$ または 1 であることが分かる．すなわち，実体インデクス中のすべての実体から構成されたサンプルは，目的集合をただ 1 個検索するか，いずれの集合も検索しない．

補題 5 実体インデクス E_Gのすべての要素を用いて作成されたサンプル S_1, S_2が G 中で識別する集合は互いに異なる．□

[1] g_i を識別するサンプルが複数存在する場合も，E_Gの各要素が g_iの正例か負例かは一意に決定されるため，すべてのサンプルは同一のサンプル S' で表現できる．

[証明] E_G中のすべての実体を用いて作成された異なるサンプル S_i, S_jが，同一の集合 g_kを唯一に識別すると仮定する．$S_i \neq S_j$より，S_iでは正例（負例）であるが S_jでは負例（正例）となる実体 e が存在する．このとき，S_iが g_kを検索するならば $e \in g_k$ $(e \notin g_k)$，S_jが g_kを検索するならば $e \notin g_k$ $(e \in g_k)$ である．よって，S_iと S_jで識別される集合は異なるため，仮定に反する．□

以上の補題を用いると実体インデクスの大きさについて以下の定理が証明できる．

定理 1 任意の集合 G に対して，G の実体インデクス E_Gは少なくとも $\log_2 |G|$ 個の要素を含む．□

[証明] 実体集合 Eのすべての要素を用いたサンプルは $2^{|E|}$個作成できる．よって，補題 4, 5より，E によって識別可能な集合数の最大値は $2^{|E|}$ である．故に定理が成立する．□

定理 1で示した実体インデクスの要素数の下限を満足する集合の例を以下に示す．

例 3 実体集合 Eのべき集合 2^E を考える．2^Eの要素 $E'(\subseteq E)$ は，サンプル $\{e \in E'\} \cup \{\bar{e} \in E - E'\}$ によって唯一に識別できる．ここで，実体 $e \in E$を含まないサンプルを用いた場合，2^E中で Eと $E-\{e\}$ が識別不能になる．従って，Eは集合 2^Eの極小の実体インデクスである．このとき，$|E| = \log_2 |2^E|$ が成立する．□

定理 2 任意の集合 G に対して，要素数が $|G|-1$ 以下の実体インデクス E_Gが存在する．□

[証明] $|G| = 1$ のとき $E_G = \phi$より定理は成立する．$|G| = k$のとき，$|E_G| \leq |G|-1$ の実体インデクスが存在すると仮定する．ここで，$G' = G \cup \{g_{k+1}\}$ を考える．E_Gは Gのすべての要素を識別するため，G の要素のうち g_{k+1} と区別できない要素 g_lは高々 1 個である．g_{k+1} と g_lを区別するためには，実体 $e \in diff(g_k, g_l)$ を Eに追加すればよい．このため，G'の実体インデクス $E_{G'}$の要素数は $|G'|-1$ 以下となる．□

以下に実体インデクス E_Gの要素数が $|G|-1$ となる集合の例を示す．

例 4 集合 G において．$g_i \cap g_j = \phi$ $(i \neq j)$ とする．G に対して，$|G|-2$ 個の実体から構成される実体インデクス E_Gが存在すると仮定する．このとき，E_Gのいずれの実体も含まない集合が G 中に少なくとも 2 個存在する．これを g_i, g_jとすると，補題 3より矛盾が発生する．

集合	サンプル
g_1	$\{e_1, \overline{e_2}, \cdots, \overline{e_{\|G\|-2}}, \overline{e_{\|G\|-1}}\}$
g_2	$\{\overline{e_1}, e_2, \cdots, \overline{e_{\|G\|-2}}, \overline{e_{\|G\|-1}}\}$
$\vdots$	$\vdots$
$g_{\|G\|-1}$	$\{\overline{e_1}, \overline{e_2}, \cdots, \overline{e_{\|G\|-2}}, e_{\|G\|-1}\}$
$g_{\|G\|}$	$\{\overline{e_1}, \overline{e_2}, \cdots, \overline{e_{\|G\|-2}}, \overline{e_{\|G\|-1}}\}$

表 2: 実体インデクスの要素数が $|G|-1$ となる例

G に対して $e_1 \in g_1, \cdots, e_{|G|-1} \in g_{|G|-1}$ からなる実体集合 Eを考える．このとき $|E| = |G| - 1$ である．表 2に示すように，E中の実体を用いると各 g_i を唯一に識別できるサンプルが作成できる． □

4 最小実体インデクスの構成問題

本節では，最小実体インデクス構成問題を定式化し，その NP 完全性を証明する．

最小実体インデクス構成問題

インスタンス： 実体集合 Eの部分集合を要素とする集合 G，正の整数 $K < |E|$.

質問： G に対して要素数が K以下の実体集合 $E' \subset E$が構成できるか．ここで，E'は G の任意の要素 g_i, g_jについて実体 $e \in diff(g_i, g_j)$ を含む．

□

最小実体インデクス構成問題がクラス NP に属することは容易に示せる．すなわち，互いに異なる $g_i, g_j \in G$ に対して，与えられた $E' \subset E$が g_i, g_jのいずれか一方のみに属する実体を含むことを検査すればよい．このために，$\forall e \in E, g \in G$ について eが gに属することが多項式時間で検査できることを仮定する．この仮定を満たさない集合によって仕様が記述されている DB 要素は，多項式時間 $O(F(|e|))$ で計算不可能なため再利用に適さない[2]．

節点被覆問題は，各節点の次数が 4 以下の平面グラフに限定した場合でも NP 完全になることが知られている [6]．本節ではこの問題を最小実体インデクス構成問題に帰着することで NP 完全性を示す．

節点被覆問題

インスタンス: グラフ $\hat{G} = (\hat{V}, \hat{E})$，正の整数 $\hat{K} < |\hat{V}|$．ただし $\hat{G}$の各節点の次数は 4 を超えない．

質問： $\hat{G}$について，要素数が $\hat{K}$以下の節点集合 $\hat{V}' \subset \hat{V}$が存在するか．ここで，$\hat{V}'$は各枝 $(u, v) \in \hat{E}$ について vまたは u の少なくとも一方を含む．

□

節点被覆問題のインスタンス $\hat{G}$と $\hat{K}$に対して，以下の手順で最小実体インデクス構成問題のインスタンス E, G, Kを構成する．

$$E = \{v_1, v_2, v_3, v_4 | v \in \hat{V}\}$$
$$K = 3|\hat{V}| + \hat{K}$$

各 $v \in \hat{V}$に対応して以下の集合を定義する．

$$g_v^1 = \{v_2, v_3\}$$
$$g_v^2 = \{v_1, v_3\}$$
$$g_v^3 = \{v_1, v_2\}$$
$$g_v^4 = \{v_1, v_2, v_3\}$$

各 $e = (u, v) \in \hat{E}$に対応して以下の集合を定義する．

$$g_e = g_v^k \cup \{u_4, v_4\}$$

ここで，$k \in \{1, 2, 3, 4\}$ は v (または u) を始点とする各枝について互いに異なるように選択する．$\hat{G}$ の各節点の次数は 4 以下であることより，これは常に可能である．以上に基づいて集合 G を以下のように定義する．

$$G = \{g_v^1, g_v^2, g_v^3, g_v^4 | v \in \hat{V}\} \cup \{g_e | e \in \hat{E}\}$$

E, G, Kが多項式時間で構成可能なことは容易に示せる．G について，以下の補題が成り立つ．

補題 6 グラフ $\hat{G} = (\hat{V}, \hat{E})$ がサイズ $\hat{K}$以下の節点被覆を持つならば，集合 G はサイズ K以下の実体インデクス $E_G \subset E$を持つ． □

[証明] 仮定を満たす $\hat{G}$の節点被覆を $\hat{V}'$とし，以下の実体集合 $S \subseteq E$を考える．

$$S = \{v_1, v_2, v_3 | v \in \hat{V}\} \cup \{v_4 | v \in \hat{V}'\}$$

明らかに $|S| = 3|\hat{V}| + \hat{K} = K$である．以下，互いに異なる任意の $g, g' \in G$ について，$diff(g, g')$ と Sが共通要素を持つことを示す．$g \neq g'$より $diff(g, g') \neq \phi$である．

$g = g_v^i, g' = g_v^j$ $(i \neq j)$ の場合，$diff(g, g') \subseteq \{v_1, v_2, v_3\} \subseteq S$である．

[2] $|e|$ は実体 e の情報量とする．

$g = g_v^i, g' = g_u^j$ $(u \neq v)$ の場合，$diff(g,g') \subseteq \{v_1, v_2, v_3, u_1, u_2, u_3\} \subseteq S$である．

$g = g_e, g' = g_v^i$の場合，$e = (u,v)$ とすると $\{u_4, v_4\} \subseteq diff(g,g')$ となる．ここで，$\hat{V}'$は節点被覆のため u または vが$\hat{V}'$に属する．従って，u_4 または v_4は Sに属する．

最後に $g = g_e, g' = g_{e'}$ $(e \neq e')$ の場合を考え，$e = (u,v), e' = (u',v')$ とする．$v = v'$ならば定義より $g \cap \{v_1, v_2, v_3\} = g_v^k$, $g' \cap \{v_1, v_2, v_3\} = g_v^{k'}$ $(k \neq k')$ となる．従って，$diff(g_v^k, g_v^{k'}) \subseteq diff(g,g')$ となるが，$diff(g_v^k, g_v^{k'}) \subseteq \{v_1, v_2, v_3\} \subseteq S$より $S \cap diff(g,g') \neq \phi$である．$v \neq v'$ならば，$g \cap \{v_1, v_2, v_3\} = g_v^k$, $g' \cap \{v_1, v_2, v_3\} = \phi$より $S \cap diff(g,g') \neq \phi$となる． □

補題 7 $\{v_1, v_2, v_3 | v \in \hat{V}\} \subset E_G$ □

[証明] 任意の $v \in \hat{V}$と $i \in \{1,2,3\}$ に対して定義より，$diff(g_v^i, g_v^4) = \{v_i\}$ が成立する．よって補題3より $v_i \in E_G$. □

補題 8 $(u,v) \in \hat{E}$ならば，$\{u_4, v_4\} \cap E_G \neq \phi$ □

[証明] $e = (u,v)$ に対応する集合 g_eに対して $g_v^k \in G$ が常に存在し，$diff(g_e, g_v^k) = \{u_4, v_4\}$ が成立する．よって補題3より $\{u_4, v_4\} \cap E_G \neq \phi$. □

補題 9 集合 G がサイズ K以下の実体インデクス $E_G \subset E$を持つならば，グラフ$\hat{G} = (\hat{V}, \hat{E})$ はサイズ$\hat{K}$以下の節点被覆を持つ． □

[証明] 補題7より $\{v_1, v_2, v_3 | v \in \hat{V}\} \subset E_G$が成立する．ここで$\hat{V}' = \{v | v_4 \in E_G\}$ を考えると，$\hat{V}' = E_G - \{v_1, v_2, v_3 | v \in \hat{V}\}$ より $|\hat{V}'| = \hat{K}$である．補題8より$\hat{V}'$は$\hat{G}$の節点被覆である． □

以上より所望の定理が証明された．

定理 3 最小実体インデクス構成問題は NP 完全である． □

5 近似的に最適な実体インデクスの構成

本節では，任意の集合 G に対して要素数が高々 $|G|-1$ の実体インデクスを構成するために実体インデクス木を導入する．実体インデクス木のデータ量は $O(|G|)$ である．また，G に対する要素の追加，削除が行なわれた場合に，実体インデクスおよび実体インデクス木を多項式時間で更新するためのアルゴリズムを提案する．

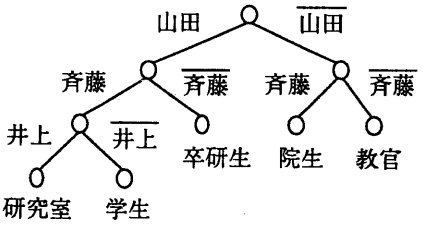

図 2: 実体インデクス木の例

5.1 実体インデクス木

集合 G に対して実体インデクス木 T_Gを以下のように定義する．

定義 2 集合 G に対する実体インデクス木 T_Gは以下の条件を満足する木である．

1. 各中間節点 vは実体 e に対応しており，2 個の子節点 v_p, v_nを持つ．枝 (v, v_p) のラベルは e, (v, v_n) のラベルは$\bar{e}$である．
2. 各葉節点 u は $g \in G$ と一対一に対応しており，根節点から u に至る経路上のラベル集合を Sとすると，$Set(G,S) = \{g\}$ である． □

実体インデクス木は必ずしも平衡しない．図1の集合 G に対して構成される実体インデクス木を図2に示す．実体インデクス木の定義より，以下の補題が成り立つ．

補題 10 実体インデクス木 T_Gのデータ量は $O(|G|)$ である． □

[証明] 定義2の条件1より T_Gにおける葉以外の節点数は葉節点数 -1 である．条件2より葉節点数は $|G|$ である．従って T_Gの節点数は $2|G|-1$ である．また T_Gは連結した木であることより，枝数は $2|G|-2$ である．以上より補題が成り立つ． □

補題 11 実体インデクス木 T_Gの中間節点に対応する実体の集合 Eは Gの実体インデクスであり，$|E| \leq |G|-1$ である． □

[証明] 定義 2の条件 2 より，T_Gを用いると集合 $g \in G$ を唯一に識別するサンプル Sが作成できる．また，T_Gの中間節点数は $|G|-1$ であることから $|E| \leq |G|-1$ である． □

なお，補題 11で構成された実体インデクスは必ずしも極小ではないことに注意されたい．

5.2 集合の追加

実体インデクスが定義されている既存の集合 G に新たに要素 g を追加する際に，実体インデクスの既存要素では g が識別できない場合がある．このような場合には適切な実体を求め，実体インデクス木を再構成する必要がある．本節では，T_G を用いた集合追加アルゴリズムを提案する．ここで，T_G の中間節点に対応する実体集合を E とする．

1. 追加する集合 g とは識別できない集合 g_k を求める．
2. $e \in diff(g, g_k)$ を E 中で検索する．
3. ステップ 2 の検索に失敗した場合，$e \in diff(g, g_k)$ を DB 中で検索する．
4. 以上の検索が失敗した場合，g と g_k を示して利用者に $e \in diff(g, g_k)$ を作成させる．
5. $E \leftarrow E \cup \{e\}$.
6. g_k に対応する葉節点 u_k に対して，子節点 u_p, u_n を作成する．u_k に対応する実体は e とし，枝 (v_k, u_p) と (u_k, u_n) のラベルはそれぞれ $e, \bar{e}$ とする．$e \in g$ ($\notin g$) ならば，u_p と u_n をそれぞれ $g(g_k)$ および $g_k(g)$ と対応づける．

ステップ 1 において，g_k を求めるアルゴリズムを以下に示す．

1. 節点 v を T_G の根節点とする．
2. v が葉節点ならば v に対応する集合を g_k とする．
3. v に対応する実体 e について，$e \in g$ ならば $v \leftarrow v_p$，そうでなければ $v \leftarrow v_n$ とする．
4. 2 へ戻る．

ここで，2 個の集合 g_k, g_l が検索されたと仮定すると，g_k, g_l は E_G によって識別できない．これは実体インデクスの定義と矛盾する．従って、g_k はただ 1 個求まる．

ステップ 2 において e が存在すれば g は E_G によって唯一に識別でき，そうでなければ識別できない．E によって，g と g_k を除くすべての集合は識別可能である．このため，g と g_k を区別する実体 e は 1 個だけ求めればよい[3]．

ステップ 3 では，E_G によって g と g_k が識別できない場合に DB 中より e を検索する．検索に失敗した場合には，g を追加した利用者が $e \in diff(g, g_k)$ を与える必要がある．ステップ 5 では実体インデクスの更新，ステップ 6 では実体インデクス木の再構成をそれぞれ行っている．

[3] ステップ 3, 4 においても同様である．

補題 12 集合追加アルゴリズムを用いた T_G および E の再構成 (ステップ 4 を除く) は多項式時間で行える． □

[証明] 定義 2 より，T_G の中間節点数は $|G|-1$ であるため，T_G の高さは $|G|$ を超えない．また，$e \in g$ の判定は 4 節の仮定により，$O(F(|e|))$ 時間で行える．従って，ステップ 1 は $O(|G| \cdot F(|e|))$ 時間で実行できる．ステップ 2 は $O(|E| \cdot F(|e|))$ 時間で行なえる．また，DB 中の実体数を N とすると，ステップ 3 の実行時間は $O(N \cdot F(|e|))$ である．ステップ 5, 6 は定数時間で行なえる．よって，ステップ 4 を除くアルゴリズムの実行時間は $O(F(|e|) \cdot (|E| + N))$ である．$F(|e|)$ は多項式のため，補題が成立する． □

ここで，g をクエリーとすると $N \cdot F(|e|)$ は g の実行時間に対応する．このため，実行時間はそれほど大きくならないと考えられる．

補題 13 実体インデクス木に対して，集合追加アルゴリズムを適用して再構成された木は定義 2 を満足する． □

[証明] ステップ 6 では，新たに中間節点となった u_k に対して実体 e を，u_k から出る枝に対してラベル $e, \bar{e}$ をそれぞれ対応づけている．従って条件 1 を明らかに満足する．根節点から u_k に至る経路上のラベル集合を S_k とすると $Set(G \cup \{g\}, S_k) = \{g, g_k\}$ である．u_k に対応する実体 e について，$e \in g$ ならば $Set(G \cup \{g\}, S_k \cup \{e\}) = \{g\}$ かつ $Set(G \cup \{g\}, S_k \cup \{\bar{e}\}) = \{g_k\}$ である．このとき，u_p と u_n にはそれぞれ g, g_k が対応する．従って，再構成後の実体インデクス木は条件 2 を満足する．$e \notin g$ の時も同様に証明できる． □

5.3 集合の削除

集合 G から要素 g を削除する際には，$|E_G| \leq |G|-1$ を保つために集合の識別に不要な実体を削除し，実体インデクス木を再構成する必要がある．本節では，T_G を用いた集合削除アルゴリズムを提案する．

例 5 集合 $G = \{g_1, \cdots, g_3\}$ に対して図 3 の実体インデクス木 T_G が定義されている．G より g_1 を削除する．実体 e_1 は g_1 と g_2 を識別するためにのみ必要なため削除できる．実体 e_2 は g_2 と g_3 を識別するために必要なため削除できない． □

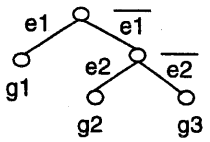

図 3: 実体インデクスからの実体の削除

T_Gを用いた集合削除アルゴリズムを以下に提案する.

1. 削除する集合gに対応する葉節点をuとし，その親節点をv_aとする.
2. v_aに対応する実体eと同一の実体が対応する節点$v_b(v_a \neq v_b)$がT_G中に存在しないならばeをEより削除する.
3. u以外のv_aの子節点をv_cとする.
4. 節点v_a, uおよび枝(v_a, u), (v_a, v_c)を削除する.
5. v_aがT_Gの根節点ならばv_cを新たな根節点とする.
6. v_aが根節点でなければ，v_aの親節点v_Aについて枝(v_A, v_a)を(v_A, v_c)に変更する.

ステップ1で求めたuの親節点v_aに対応する実体eが削除の候補となる．ステップ2では，eがEから削除可能であることを確認した後，削除を実行する．ステップ3から6では実体インデクス木の再構成を行なう．ステップ6では枝のラベルは変更しない.

補題 14 集合削除アルゴリズムを用いたT_GおよびEの再構成は，$O(|G|)$時間で行える. □

[証明] ステップ1は定数時間で行なえる．中間節点数が$|G|-1$であることより、ステップ2は$O(|G|)$時間で行なえる．ステップ3から6まではGの要素数に関わらず定数時間で行なえる. □

補題 15 実体インデクス木に対して，集合削除アルゴリズムを適用して再構成された木は定義2を満足する. □

[証明] v_aを除く中間節点と枝のラベルは変更されないため，明らかに条件1を満足する．v_Aを先祖とする各葉節点u_iについて，根節点とu_iの経路上にあるラベル集合から構成されるサンプルをS_i，u_iに対応する集合をg_iとする．削除された枝(v_a, v_c)のラベルがeならば，すべてのg_iについて$e \in g_i$が成立する．よって，$Set(G-\{g\}, S_i-\{e\}) = Set(G-\{g\}, S_i) = Set(G, S_i) = \{g_i\}$が成立する．また，$u$を除く各葉節点は変更されないため，再構成後の実体インデクス木は条件2を満足する．(v_a, v_c)のラベルが$\overline{e}$の場合も同様に証明できる. □

補題 11，13，15より，実体インデクスの要素数は$|G|-1$以下であることが保証される.

6 おわりに

本稿では，サンプルを用いた集合検索を効率よく行うための実体インデクスを定義し，その性質を調べた．最小の実体インデクス構成問題はNP完全になるが，DB要素の集合Gに対して，サイズ$|G|-1$以下の実体インデクスが多項式時間で構成できる.

実体インデクスとサンプルを用いた集合検索を併用することで，集合によって記述される各種の情報を効率よく検索できる．この中には各種DB要素の他にも，キーワード集合で特徴づけられる文書等が含まれる．従って，実体インデクス構成問題は文書検索のためのキーワード設計問題にも対応している.

本稿では実体インデクスの要素数に着目した考察を行った．今後，実体インデクスを用いて構成されたサンプルに対する評価を行い，最適なサンプルを効率よく構築できる実体インデクスに関する考察を進める予定である．これと併せて，各種DB要素およびソフトウエア部品の再利用に向けた汎用かつ実用的な枠組みを構築する予定である.

謝辞 本研究の一部は文部省科学研究費特定領域研究（#08244105）の援助を受けている.

参考文献

[1] 村田，掛下，“データベース化されたビューに対するサンプルを用いた検索”，電子情報通信学会DEWS'97論文集，pp.67–72, 1997.

[2] 村田，掛下，“サンプルを用いた論理式検索機構の評価”，情報処理学会DBS研究会 133–3, 1997.

[3] 掛下，村田，“論理式による内包情報のモデル化とその操作”，情報処理学会全国大会 6AA-5, 1997.

[4] U. Fayyad, R. Uthurusamy, Ed. , “Data mining and knowledge discovery in databases”, *Comm. ACM*, Vol. 39, No. 11, pp.24–68, Nov. 1996.

[5] M-S. Chen, J. Hans, P. S. Yu, “Data mining: an overview from a database perspective”, *IEEE Trans. Knowledge and Data Engineering*, Vol. 8, No. 6, pp. 866–883, 1996.

[6] M. R. Garey, D. S. Johnson, "Computers and Intractability: A Guide to the Theory of NP Completeness", Freeman, 1979.