

仕様に基づいた RDB クエリ検索システム

掛下 哲郎[†] 村田 美友紀^{††}

DB アプリケーションを構築する際には数多くのクエリが必要になる。これらのクエリの中には、単純な SQL 文で記述できない等の理由により実現が難しいものも含まれる。したがって、効率的なアプリケーション開発を促進するためには、クエリの再利用が不可欠である。我々は、各種コンポーネントを仕様に基づいて検索するために、サンプルを用いた方式を提案した。本稿では、サンプルを用いた検索方式を適用して、仕様に基づいた RDB クエリ検索システムを実装する。本システムは、クエリの仕様を属性、組、組のペアを要素とする集合を用いて表現し、集合要素から構成されるサンプルを用いて検索を行う。ここで、クエリを特定するサンプルを作成するために要素辞書を用いる。効率的にサンプルを作成するための利用者支援機構として、フィルタリング、検索、重複度別分類を組み合わせた方式を実装する。評価実験の結果、94%のクエリがサイズ $\log n$ 以下のサンプルで特定された (n はクエリ数)。

A Specification Based Retrieval System for RDB Queries

TETSURO KAKESHITA[†] and MIYUKI MURATA^{††}

A DB application contains number of queries some of which cannot be described by a simple SQL statement. Reuse of such query is thus necessary for effective development of a DB application. We have proposed a sample based mechanism to retrieve components. In this paper, we utilize this mechanism to implement a specification based RDB query retrieval system. Query specification is represented by a set whose members are projection attributes, tuples and tuple pairs. A sample is composed of some members of the sets. We use the element dictionary in order to construct a sample to identify a query in the repository. Furthermore we implement a method by combining three mechanisms in order to assist sample construction. They are filtering, element retrieval and element weight. 94% of the queries are identified by a sample of size less than $\log n$, where n is the number of queries.

1. はじめに

情報化社会の高度化により、アプリケーションを構成する関数やモジュール等のソフトウェアコンポーネントは複雑化している。ソフトウェア工学の分野では、アプリケーション開発コストを低減するために、関数やモジュールの再利用が行われている。

また、データベース (DB) の分野においても DB アプリケーション構築の際には、多様な利用目的に対応したフォームやレポートを作成するため、多くのクエリが必要になる。これらのクエリ等を DB コンポーネントと呼ぶ。クエリの中には複数のクエリを組み合

わせなければ作成できないものもある。図 1 に示すクエリ q は、党派別の当選者数と得票総数を求め、当選者数順に並べ替えを行う。 q は 3 個のテーブルを基に、2 個の選択クエリと 3 個の集計クエリを用いて構成されている。複雑な構造を持つクエリは単純な SQL 文では記述できない。したがって、既存クエリの再利用が不可欠である。

コンポーネントを再利用するためには、リポジトリ中から所望のコンポーネントを検索する機構が必要になる。このとき、コンポーネント仕様に基づいて検索できることが望まれている。我々は、サンプルを用いたコンポーネント検索方式を提案した¹⁾。本方式は、集合を用いてコンポーネント仕様を表現する。利用者は、集合要素を用いてサンプルを作成する。本方式を用いたコンポーネント検索は多項式時間で終了する。また、サンプル作成コストを低減するために、要素辞書²⁾とサンプル作成支援機構¹⁾を活用する。これらを組み合わせるとリポジトリ中のコンポーネントをつね

[†] 佐賀大学理工学部知能情報システム学科

Department of Information Science, Faculty of Science and Engineering, Saga University

^{††} 八代工業高等専門学校情報電子工学科

Department of Information and Electronics Engineering, Yatsushiro National College of Technology

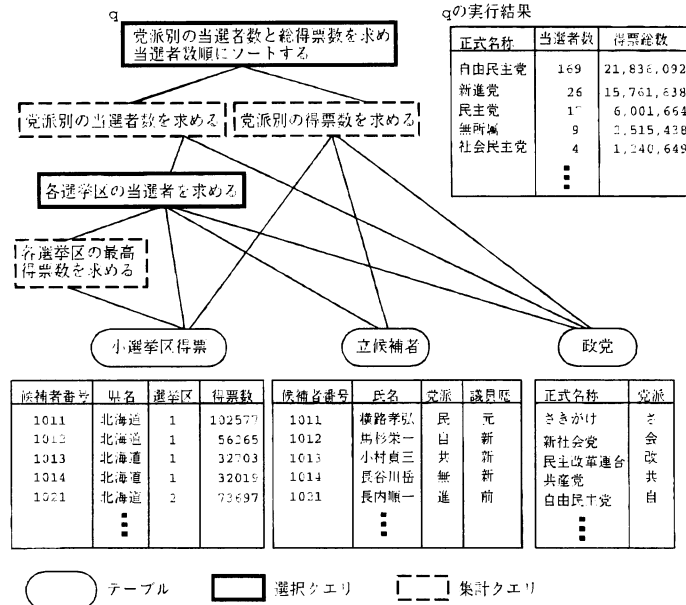


図1 複数クエリから構成されるRDBクエリの例

Fig. 1 Example RDB query consisting of multiple queries.

に特定できる。また、サンプルサイズはほとんどの場合 $\log_2 n$ 以下になる。ここで n はリポジトリ中のコンポーネント数である。

本稿では、サンプルを用いた検索方式を適用して、仕様に基づいたRDBクエリ検索システムを実装する。さらに、クエリ検索システムを用いて要素辞書およびサンプル作成支援機構の有効性を評価する。

ソフトウェア工学分野では、ソフトウェアコンポーネントを再利用するために、仕様に基づいた検索方式が研究されている。Rittri³⁾やZaremskiら⁴⁾によって、引数や戻り値の型から構成されるシグネチャを用いた関数検索方式(シグネチャマッチング)が提案されている。この方式では詳細な検索条件の記述が難しい。Fisherら⁵⁾やMiliら⁶⁾は、利用者が記述した仕様を検索条件として用いるコンポーネント検索方式を提案している。この方式では、利用者が仕様記述言語を習得する必要がある。また、定理証明器を用いて検索を実現するため、大規模リポジトリに対する検索の効率化が難しい。Podgurskiら⁷⁾は入力や出力の値をサンプルとして用いた関数検索機構を提案しているが、利用者に対する支援機構は提供されていない。

本稿は以下のように構成されている。2章ではサンプルを用いたコンポーネント検索方式を定義する。3章ではクエリ検索方式を提案する。4章では、クエリ検索システムの実装について述べる。5章では実装したシステムを用いて要素辞書およびサンプル作成支援

機構の有効性を検証する。最後にまとめと今後の課題について述べる。

2. サンプルを用いたコンポーネント検索方式

本章では、サンプルを用いたコンポーネント検索方式を定義する。また、サンプル作成コストを低減する方式として、要素辞書とサンプル作成支援機構を導入する。

2.1 サンプルを用いた検索

本方式では、集合を用いてコンポーネントの仕様を表現する。コンポーネント c の仕様を表現する集合を $g(c)$ と表記する。

コンポーネントの集合を $C = \{c_1, \dots, c_n\}$ とする^{*}。サンプル S は、リテラル $e \in U_{i=1}^n g(c_i)$ の集合である。 e は正例か負例かが決定されている。 $|S|$ をサンプルサイズと呼ぶ。コンポーネント $c \in C$ と S について、すべての正例 $e \in S$ が $e \in g(c)$ を満足し、すべての負例 $\bar{e} \in S$ が $\bar{e} \notin g(c)$ を満足するとき S は c を検索する。

コンポーネント集合 C とサンプル S について、 $Set(C, S) = \{c | c \in C, S \text{ は } c \text{ を検索する}\}$ を定義する。 $Set(C, S) = \{c\}$ が成り立つとき、 S は c を特定するという。

C 中で検索の対象とするコンポーネントを目的コン

* リポジトリは全コンポーネントの集合を格納する。

ポーネントと呼ぶ。また、サンプルへのリテラル追加によって絞り込まれたコンポーネントを候補コンポーネントと呼ぶ。

2.2 要素辞書

要素辞書を用いると、リポジトリ中で目的コンポーネントを特定するサンプルがつねに構成できる。また、コンポーネント間の相違に関する質問にも答えることができる。

定義 1 コンポーネント集合 C の要素辞書 D_C は $U = \cup_{i=1}^n g(c_i)$ の部分集合であり、任意の $c, c' \in C$ ($g(c) \neq g(c')$) に対して $g(c) \cap D_C \neq g(c') \cap D_C$ を満足する。 □

要素数が最小の要素辞書構成問題は NP 完全である。我々は、 n 個の要素を持つ任意のコンポーネント集合に対し、要素数がたかだか $n-1$ の要素辞書を構成するアルゴリズムを考案した。本アルゴリズムはコンポーネントの追加、削除の際に実行され、要素辞書を再構成する。以上の詳細については文献 2) を参照されたい。

2.3 サンプル作成支援機構

サンプルを作成するための利用者支援機構として、要素検索と重複度別分類が提案されている。これらを用いると、コンポーネントを特定するサンプルサイズの期待値は $\log_2 n$ になる¹⁾。

要素検索

α は、要素 e とコンポーネント c について $e \in g(c)$ が成立する確率の平均値である。 α の値はリテラル追加前後の候補コンポーネント数の比によって推定できる。システムは、 $\alpha > 0.5$ ならば負例を、 $\alpha \leq 0.5$ ならば正例を、それぞれサンプル追加に適したリテラルの種類として利用者に提示する。利用者はこれに従って目的コンポーネントの正例または負例を検索する要素検索クエリを作成する。システムは、要素検索クエリを用いて要素辞書から要素を検索する。検索された要素は、システムが指定した種類のリテラルとしてサンプルに追加される。

重複度別分類

要素 e の重複度は、 e を正例とするコンポーネントの数であり、要素の絞り込み効果に対応する。システムは、重複度順にソートした要素のリストを利用者に提示する。利用者は提示された要素リストの中から適切な要素を選択し、リテラルとしてサンプルに追加する。重複度の小さな要素は正例として、大きな要素は負例として追加することが望ましい。これにより、候補コンポーネントの絞り込み効果を考慮して要素を選択できる。

3. RDB クエリ検索方式

本章では、クエリをコンポーネントとするリポジトリに対して、サンプルを用いた検索方式を適用する。そのために、集合を用いてクエリの仕様を表現する。次に、サンプルを用いたクエリ検索アルゴリズムにサンプル作成支援機構を組み込む。本稿では選択クエリ (SPJ 質問)、集計クエリ、これらを組み合わせたクエリを検索の対象とする。

3.1 RDB クエリ仕様の集合表現

RDB クエリ記述言語には SQL や QBE 等がある。いずれの言語においても、作成されたクエリは選択条件および結合条件を満足する組について、射影属性で指定された属性値の組を検索する。また、ソート条件を指定することで組の表示順序を指定できる。よって、クエリの仕様はクエリ記述言語の種類にかかわらず、クエリの実行結果を用いて統一的に表現できる。以上の考察より、クエリ q の仕様を次の集合で表現する。

$$\{q \text{ の射影属性} \} \cup \{q \text{ が検索する組} \}$$

$$\cup \{q \text{ が検索する組のペア} \}$$

$\{q \text{ の射影属性} \}$ を q のシグネチャと呼ぶ。 q のシグネチャには、関係スキーマ上で定義された属性だけでなく、関数や式を用いて計算された属性も含まれる。スキーマ上で定義されている属性には、スキーマ設計者によって名前が付けられる。これに対して、関数等を用いて計算された属性には、クエリ作成者が名前を付ける。このため、意味的に異なる属性に同一の名前が付けられる可能性がある。しかし、同一名の属性が異なる意味を持つと、検索の際にあいまいさが生じる。本稿では、再利用を促進する観点から以下を仮定する。**仮定 1** 2つの属性 a と a' は、両者が意味的に等しい場合にのみ同一の名前を持つ。 □

$\{q \text{ が検索する組} \}$ はクエリの検索条件を表現する。ソート条件を含むクエリ仕様を表現する場合には組の表示順序を指定する必要がある。これを指定するために $\{q \text{ が検索する組のペア} \}$ を用いる。なお、関数を用いたソート条件の指定等を考慮すると、検索したクエリに対するソート条件の追加は必ずしも容易でない。したがって、ソート条件を考慮したクエリ検索も必要になる。

クエリ仕様の表現例を以下に示す。

例 1 図 2(A) に示す 2つの関係を考える。これらに対して、図 2(B) のクエリ q_1 を実行した結果が図 2(C) である。よって、 q_1 の仕様は以下の集合を用いて表現できる。

講義		
科目	教官	時間数
英語	山田	16
DB	山下	16
SW	田中	15
英会話	山田	10

成績			
学生	科目	点	欠課数
森	英語	43	0
斎藤	英語	90	0
鈴木	英語	65	2
森	DB	76	5
斎藤	DB	78	3
森	英会話	49	0
鈴木	英会話	45	2

(A)

学生	科目	点
森	英語	43
鈴木	英会話	45
森	英会話	49

(C)

```

q1
SELECT  学生, 科目, 点
FROM    成績, 講義
WHERE   (講義.科目 = 成績.科目)
        AND (教官 = '山田') AND (点 < 50)
ORDER BY 点
  
```

(B)

図2 サンプルスキーマ

Fig. 2 Sample schema.

{ 学生, 科目, 点, < 森, 英語, 43>, < 森, 英会話, 49>, < 鈴木, 英会話, 45>, [
< 森, 英語, 43>, < 森, 英会話, 49>], [
< 森, 英語, 43>, < 鈴木, 英会話, 45>], [
< 森, 英会話, 45>, < 鈴木, 英会話, 49>] } □

RDBクエリはDBの状態に依存して実行結果が決まるため、DB状態が異なると仕様を表現する集合が異なる。そこで、クエリ実行前のDB状態を仮定する。このDB状態を検索用DBと呼ぶ。検索用DBは通常のDBとは独立しており、更新操作の影響を受けない。クエリ集合 C に対する検索用DBについて以下の仮定を設ける。

仮定2 検索用DBにおけるRDBクエリ仕様を表現する集合は、任意の2個のクエリ $q, q' (\in C)$ について、 $g(q) \neq g(q')$ となる。 □

仮定2を満足する検索用DBが存在した場合、2.2節で示したように要素辞書をつねに構成できる。また、5章で使用する検索用DBでは意図的なデータ編集を行っていないが、上記の仮定は満足される。しかし、一般には任意のクエリ集合 C に対して仮定2が満足されるとは限らない。ただし、検索用DBとして複数のバージョンを許した場合には、以下の定理が成立する。

定理1 クエリ集合 C において $q \in C$ の仕様が互いに異なるならば、たかだか $|C| - 1$ 個のバージョンを持つ検索用DBが存在して、任意の2個のクエリ $q, q' (\in C)$ について、いずれかのバージョンで $g(q) \neq g(q')$ となる。 □

[証明] $|C|$ に関する帰納法を用いる。 $|C| = 1$ の場合には検索用DBは不要なので、明らかに定理が成立する。 $|C| = k$ の場合に定理が成立したと仮定する。

仮定より、 $C \cup \{q_{k+1}\}$ において、検索用DBのすべてのバージョンで $g(q_{k+1}) = g(q)$ となる $q \in C$ はたかだか1個しかない。したがって、 q, q_{k+1} の実行結果が異なる検索用DBを新たなバージョンとして追加することで定理の条件を満たせる。 □

本稿では、議論を単純化するために単一バージョンの検索用DBに関する検討を主に行う。このために仮定2を設ける。

3.2 サンプルを用いたRDBクエリ検索

2章で提案した検索方式を用いてRDBクエリを検索する際には、サンプル S 中に含まれるリテラル e がクエリ q の仕様を表現する集合 $g(q)$ に含まれるか否かを判定する必要がある。

e が属性のとき、 q の射影属性の集合に e が含まれるならば $e \in g(q)$ である。 e が組のとき、 q を検索用DB上で実行し、検索された組の集合に e が含まれるならば $e \in g(q)$ である。 e が組のペアのとき、 q を検索用DB上で実行した結果に e が示す順序で e を構成する2つの組が含まれるならば $e \in g(q)$ である。

クエリを検索するためのアルゴリズムを以下に示す。ここで下線部は利用者が行う。

アルゴリズム1 RDBクエリ検索

- (1) サンプルを空にする。
- (2) リテラルをサンプルに追加する。
- (3) サンプルを用いてクエリを検索する。
- (4) クエリが特定されていなければ(2)へ戻る。 □

アルゴリズム1を用いたクエリ検索は、利用者がサンプルを改良しながら対話的に行う。サンプルの改良は利用者が行うため、クエリ検索のための利用者の負担はステップ(2)~(4)の繰返し回数が多い(サンプル

q_2 : 山田教官が担当する教科の得点が 50 点以下の学生を検索する。

q_3 : ある科目の欠課率が 10% 以上の学生を検索する。

q_4 : 英語の成績が平均点以下の学生を検索する。

q_5 : 得点が 50 点以下の学生が存在する科目を検索する。

学生	科目	点
森	英語	43
森	英会話	49
鈴木	英会話	45

学生	科目	欠課率
鈴木	英語	12.5
森	DB	31.3
斎藤	DB	18.8
鈴木	英会話	20.0

学生	点
森	43
鈴木	65

科目	教官
英語	山田
数学	田中
英会話	山田

図 3 クエリの例
Fig. 3 Sample queries.

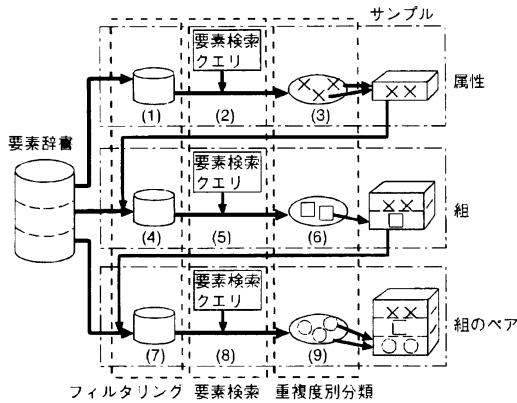


図 4 サンプル作成手順
Fig. 4 Sample construction steps.

サイズが大きい)ほど増大する。なお、ステップ (3) の検索は候補クエリを対象に行えばよい。

以下にサンプルを用いたクエリ検索の例を示す。

例 2 図 2 に示す検索用 DB とクエリ q_1 、図 3 に示すクエリ q_2, \dots, q_5 を考える。クエリ集合を $C = \{q_1, \dots, q_5\}$ とする。まず、属性を追加したサンプル $S_1 = \{点\}$ を用いて検索すると、 $Set(C, S_1) = \{q_1, q_2, q_4\}$ となり射影属性に点を含むクエリが検索される。 S_1 に組 $\langle 森, 英語, 43 \rangle$ を追加したサンプル S_2 を用いてクエリを検索すると、 $Set(C, S_2) = \{q_1, q_2\}$ となる。さらに組のペア $\{\langle 鈴木, 英会話, 45 \rangle, \langle 森, 英会話, 49 \rangle\}$ を正例として追加したサンプル S_3 を用いて検索すると $Set(C, S_3) = \{q_1\}$ となり、 q_1 が特定される。 □

3.3 サンプル作成手順

本節では、RDB クエリ検索のための効率的なサンプル作成手順を提案する (図 4)。この図では標準的な手順を示しているが、利用者は必要に応じて各手順をスキップしたり、繰り返し実行したりできる。

C' を候補クエリの集合とする。クエリ絞り込みにもとない要素辞書は再構成される。利用者はこの要素辞書を用いてサンプルを作成する。これにより C' 中の任意のクエリが特定できる。

表 1 要素検索クエリで指定する検索条件
Table 1 Retrieval condition for element retrieval query.

	条件
属性	利用者が指定した属性 a を用いて値が計算されている属性 利用者が指定した関数 f を用いて値が計算されている属性 利用者が指定した関係 r に含まれている属性
組	利用者が指定した式 exp を満足する組。ただし、 exp は SQL の WHERE 句に記述できる式である。
組のペア	利用者が指定した組 t (現在のサンプルの正例) を含むペア

クエリの仕様を表現する集合には 3 種類の要素が含まれる。この中で、サンプルに対するリテラルの追加は属性から開始する。その後、組、組のペアの順に追加する。属性のみで構成されるサンプルはシグネチャマッチングを用いたクエリ検索を実現する。組の追加は、検索条件に基づいたクエリの絞り込みに対応する。組のペアはクエリが表示する組の順序を明示するために追加する。この順序でリテラルを追加する方式は検索条件の詳細化に対応するため、利用者にも分かりやすい。

リテラルの追加は、フィルタリング (図 4 (1), (4), (7)), 検索 (図 4 (2), (5), (8)), 重複度別分類 (図 4 (3), (6), (9)) の 3 つから構成される。

フィルタリング

フィルタリングは、要素辞書から所望の種類 (属性、組、組のペア) の要素を抽出する。フィルタリングはシステムが自動的に実行する。要素 e は要素辞書から抽出されるため、 e をリテラルとしてサンプルに追加すると、候補クエリ数はつねに減少する。

要素検索

本手法は、フィルタリングで求められた要素の中から、利用者が作成する要素検索クエリを用いてサンプルに追加するリテラルを検索する。要素検索クエリは通常のクエリとは異なり、組だけでなく、属性や組のペアも検索対象とする。また、要素検索クエリはスキーマが異なる組を検索する場合もある。表 1 に要素

検索クエリで指定できる条件を要素の種類ごとに示す。

重複度別分類

重複度別分類を用いると、サンプルへの追加候補要素の中から絞り込み効果が最も高い要素を選択できる。システムは、検索された要素について重複度を計算し、重複度順にソートした要素のリストを利用者に提示する。利用者は提示されたリストの中から要素を選択し、正例/負例を指定する。

要素検索と重複度別分類の使い分け

要素の重複度を計算するためには、候補クエリを実行して、実行結果に当該要素が含まれるか否かを判定する必要がある。候補クエリ数が多い場合には、重複度別分類の適用は計算コストの面から困難になる。したがって、大規模リポジトリでクエリ検索を行う場合、初期段階では要素検索のみを用いてサンプルに追加する要素を求める方が合理的である。また、要素検索を用いることで複数リテラルを一括してサンプルに追加できるため、利用者が作成する要素検索クエリ数は $\log n$ より小さくなる。

一方、クエリの絞り込みが進行した段階では候補クエリ間の相違が少なくなるため、要素検索クエリの作成は困難になる。しかし、候補クエリ数が少なくなっている可能性が高いため、重複度別分類が効果的に活用できる。

このように、要素検索と重複度別分類は相補的に利用する。

4. 実装

本章では RDB クエリ検索システムの全体構成を示した後、データ構造、ユーザインタフェースについて述べる。

4.1 全体構成

クエリ検索システムの全体構成（データフロー図）を図 5 に示す。本システムは、クエリの追加削除、サンプル作成支援、クエリ検索の各モジュールで構成される。リポジトリは検索対象のクエリを格納する。利用者がクエリの追加削除を行うとリポジトリが更新される。これと同時に要素辞書も再構成される。サンプルを作成する際には、サンプル作成支援を用いる。サンプル作成支援は、属性リテラル追加、組リテラル追加、組のペアリテラル追加に詳細化されている。さらにそれぞれについて、フィルタリング、要素検索、重複度別分類に詳細化される。クエリ検索では、作成したサンプルを用いてクエリの検索を行う。

4.2 データ構造

本システムは Microsoft Access を用いて実装した。

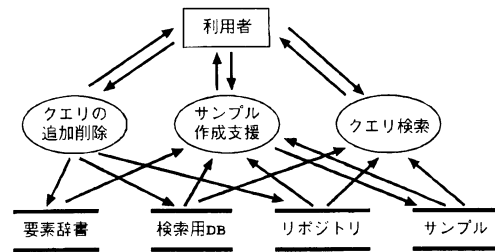


図 5 システム構成図

Fig. 5 System organization.

主なテーブルのスキーマを表 2 に示す。* はキー属性を示す。要素辞書には要素の識別番号を登録し、要素の実体は種類別のテーブルに格納する。組テーブルは組を直接格納するのではなく、属性値ごとに分割して格納する。改良後のサンプルを用いたクエリの検索は、候補クエリを対象に行えばよい。これを実現するために、リポジトリの属性‘候補’に各クエリが候補クエリであるか否かを記憶する。また、3.3 節で導入したサンプル作成手順に従うとフィルタリング、要素検索、重複度別分類の順で実行することにより、サンプル追加の候補となる要素の絞り込みが行われる。この情報は要素辞書テーブル中の属性‘候補’に格納する。

4.3 ユーザインタフェース

RDB クエリ検索システムのユーザインタフェースを図 6 に示す。メインウィンドウで、クエリ追加、クエリ削除、サンプル作成、クエリ検索ボタンを押すと対応する機能が実行される。サンプル作成ボタンを押すと 2 つのウィンドウ（サンプルとサンプル作成支援）が開く。サンプルウィンドウは現在のサンプルを表示する。利用者はサンプル作成支援で、適切なステップに対応するボタンを押す。たとえば、属性/フィルタリングボタンを押すと属性のフィルタリングに結果が表示される。属性/検索ボタンを押すと属性の検索が開く。利用者はここで要素検索クエリを作成する。属性の検索ボタンまたはサンプル作成支援の属性/重複度別分類ボタンを押すと属性の重複度別分類が開き、要素リストが表示される。利用者が正負フィールドで正または負を選択すると、要素は利用者が指定した種類のリテラルとしてサンプルに追加される。クエリの検索は、メインウィンドウでクエリ検索ボタンを押すことで実行される（サンプルを用いたクエリ検索結果）。クエリの一覧より所望のクエリを実行することもできる。

5. 評価実験

本章では、4 章で実装したプロトタイプを用いて、

表 2 テーブルの構造
Table 2 Table schema.

テーブル	属性	データ型	説明
リポジトリ	クエリ番号*	数値	クエリ識別番号
	クエリ名	テキスト	クエリ名
	候補	Yes/No	既存のサンプルで検索されるならば Yes, そうでなければ No
要素辞書	種類*	数値	要素の種類 (属性, 組, 組のペアのいずれか)
	要素番号*	数値	要素の識別番号
	候補	Yes/No	候補要素ならば Yes, そうでなければ No
サンプル	種類*	テキスト	要素の種類 (属性, 組, 組のペアのいずれか)
	要素番号*	数値	要素の識別番号
	正負	テキスト	正, 負, 未 (サンプルに含まれない) のいずれか
属性	属性番号*	数値	属性の識別番号
	属性名	テキスト	属性名
	データ型	数値	データ型を示す数値
組	組番号*	数値	組の識別番号
	属性番号*	数値	属性の識別番号
	属性値	テキスト	属性値
組のペア	組のペア番号*	数値	組のペアの識別番号
	前組番号	数値	先に表示される組の識別番号
	後組番号	数値	後に表示される組の識別番号



図 6 ユーザインタフェース

Fig. 6 User interface.

要素辞書およびサンプル作成支援の有効性を確認するための評価実験を行う。

5.1 実験方法

本実験には 96 年総選挙 DB を用いた。対象 DB は 8 個のテーブルから構成されており、属性の総数は 47 である。リポジトリに格納されたクエリ総数は 47 である。このうち、入れ子クエリは 20 個で、クエリを構成するサブクエリ数の最大値は 4 である。検索対象クエリのうち射影属性が互いに一致するクエリ集合が 7 個ある。射影属性が一致するクエリの最大数は 6 であり、そのうちの 4 個はソート条件だけが異なる。また、計算によって求められる属性を含むクエリ数は 22

である。したがって、リポジトリは利用者が直接作成するのは困難なクエリを含む。また、類似したクエリが多く存在する。

本実験はリポジトリ中の各クエリを特定するサンプルを作成し、そのサンプルサイズについて考察する。サンプルは以下の 3 通りの方法で作成する。

Case 1 要素辞書、サンプル作成支援を用いない。リポジトリ中よりランダムに選択したクエリ q について、 $g(q)$ から 1 個の要素をランダムに選択し、サンプルに追加する。

Case 2 要素辞書は用いるがサンプル作成支援を用いない。要素辞書から 1 個の要素をランダムに選

択し、サンプルに追加する。

Case 3 要素辞書とサンプル作成支援（フィルタリングと重複度別分類）を用いる。要素辞書から既存のサンプルを用いてフィルタリング、重複度別分類を行い、最も絞り込み効果が高い要素をサンプルに追加する。

Case 1 は Podgurski ら⁷⁾が提案する方式に対応する。属性はクエリのシグネチャに対応するため、シグネチャマッチングは Case 1 において属性のみでサンプルを作成する場合に対応する。提案方式は Case 3 に対応する。Case 2 は要素辞書とサンプル作成支援機構の効果を分離するために定義した。

サンプル作成支援機構における要素検索クエリは利用者が作成するため、適切なクエリの作成は利用者の経験や知識に依存する。評価実験ではサンプル作成支援機構自身の有効性を評価するために、要素検索クエリはつねに True とする。つまり要素検索クエリはフィルタリングによって抽出されたすべての要素を検索する。

要素辞書は、47 個のクエリをランダムな順序でリポジトリに追加することで作成した。要素辞書は、属性 18 個、組 11 個、組のペア 4 個の計 33 個の要素から構成されている。

5.2 実験結果

図 7、図 8、図 9 に、各場合におけるサンプルサイズと候補クエリ数の分布を示す。破線は平均値を示す。Case 2 と Case 3 ではサンプルサイズが大きくなるに従って候補クエリ数が単調減少しているのに対し、Case 1 ではサンプルサイズが 20 を超えると絞り込みが進まなくなる。候補クエリの仕様が互いに類似した状況でクエリを特定するためには、目的クエリを他のクエリと区別するリテラルをサンプルに追加する必要がある。Case 1 では $\cup_{i=1}^n g(q_i)$ ($q_i \in C$) からランダムに要素を選択しているため、適切な要素の選択が困難である。一方、Case 2、Case 3 では、サイズが 33 (=要素辞書のサイズ) 以下のサンプルですべてのクエリが特定できる。これは、要素辞書を用いることでクエリを区別する要素を選択できるためである。

図 10 にサンプルサイズとそのサンプルサイズにおいて特定されたクエリ数の度数分布を示す。Case 1 ではサイズ 30 以上のサンプルでなければ特定できないクエリ数が 27 であった。これに対し、Case 2 では 39 個のクエリがサイズ 30 以下のサンプルで特定された。さらに、Case 3 では、94%のクエリがサイズ 5 ($< \log_2 47$) 以下のサンプルで特定された。2.3 節の理論的成果とあわせて考えると、実際のサンプルサ

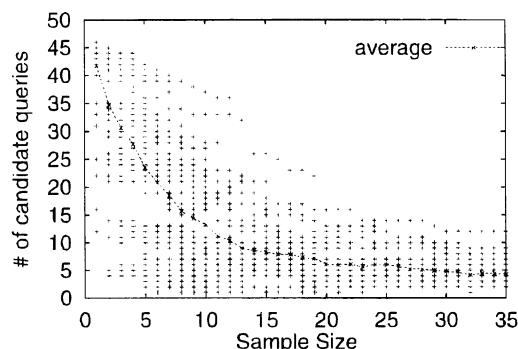


図 7 サンプルの絞り込み効果：Case 1

Fig. 7 # of candidate queries at each sample size: Case 1.

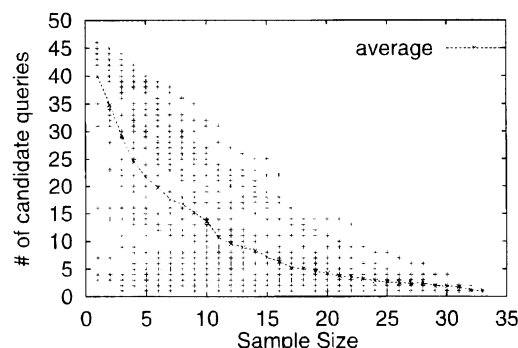


図 8 サンプルの絞り込み効果：Case 2

Fig. 8 # of candidate queries at each sample size: Case 2.

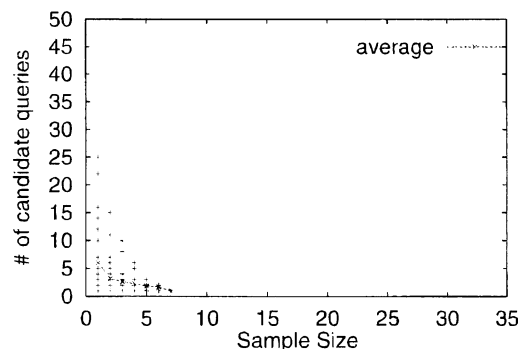


図 9 サンプルの絞り込み効果：Case 3

Fig. 9 # of candidate queries at each sample size: Case 3.

イズは期待値 $\log n$ を下回る可能性が高い。よって、RDB クエリ検索におけるサンプル作成支援機構の有効性が示された。

6. おわりに

本稿では、仕様に基づいて RDB クエリを検索するシステムを実装し、その評価を行った。本システムを

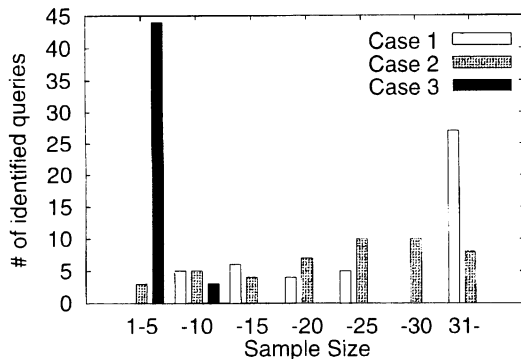


図 10 サンプルによって特定されたクエリ数

Fig. 10 # of identified queries at each sample size.

用いることにより、所望のクエリを多項式時間で特定できる。本システムは要素辞書、サンプル作成支援機構の導入により、利用者のサンプル作成負担を軽減できる。

本システムは検索/集計クエリのみを検索対象としている。このほかに、パラメータクエリ、追加クエリ、削除クエリ、更新クエリ等がある。このうち、パラメータクエリの仕様は射影属性だけでなく値入力を要求する属性を含む。組および組のペアの各要素は入力値を含むように拡張される。追加/削除/更新クエリの仕様は、変更対象の組や変更後の組を要素とする集合で表現できる。したがって、サンプルを用いた検索方式は様々なクエリ検索に応用できる。

仮定 1 を満たすためにはテーブル作成やクエリ作成時における属性の名前付けガイドラインや検証機構が必要になる。

仮定 2 は本稿で使用した総選挙 DB 等では成立する。しかし、サンプルを用いた検索方式の適用範囲をより広めるためには、定理 1 で示したように検索用 DB のバージョン管理を導入する方法が考えられる。そのためにはバージョン管理が必要になる条件や、検索用 DB の最小化問題に関する考察等が必要になる。

これらについても今後研究を行う予定である。

謝辞 投稿版原稿に対して丁寧なコメントを書いていただいた匿名の査読者に感謝いたします。

参考文献

- 1) Kakeshita, T. and Murata, M.: Specification-based component retrieval by means of examples, *International Symposium on Database Applications in Non-Traditional Environments*

(DANTE'99), pp.389–398 (1999).

- 2) 村田美友紀, 掛下哲郎: 集合間の相違を明確にする要素辞書, *情報処理学会論文誌: データベース*, Vol.40, No.SIG3(TOD 1), pp.60–67 (1999).
- 3) Rittri, M.: Using types as search keys in function libraries, *Proc. Conf. on Functional Programming Languages and Computer Architectures*, Reading, Mass., Addison-Wesley (1989).
- 4) Zaremski, A.M. and Wing, J.M.: Signature matching: A tool for using software libraries, *ACM Trans. Softw. Eng. and Methodology*, Vol.4, No.2, pp.146–170 (1995).
- 5) Fischer, B., Kievernagel, M. and Snelting, G.: Deduction-based software component retrieval, *IJCAI Workshop on Reuse of Proofs, Plans and Programs*, Montreal, Canada (1995).
- 6) Mili, R., Mili, A. and Mittermeir, R.T.: Storing and retrieving software components: a refinement based system, *IEEE Trans. Softw. Eng.*, Vol.23, No.7, pp.445–460 (1997).
- 7) Podgurski, A. and Pierce, L.: Retrieving reusable software by sampling behavior, *ACM Trans. Softw. Eng. and Methodology*, Vol.2, No.3, pp.286–303 (1993).

(平成 12 年 3 月 20 日受付)

(平成 12 年 7 月 11 日採録)

(担当編集委員 吉川 正俊)



掛下 哲郎 (正会員)

昭和 37 年生。昭和 59 年九州大学工学部情報工学科卒業。平成元年同大学院博士後期課程修了。工学博士。同年佐賀大学理工学部講師を経て、現在、助教授。データベースおよびソフトウェア工学の研究に従事。電子情報通信学会、ACM、IEEE 等会員。



村田美友紀 (正会員)

昭和 46 年生。平成 6 年佐賀大学工学部情報科学科卒業。平成 8 年同大学院博士前期課程修了。同年八代工業高等専門学校助手。平成 12 年佐賀大学大学院博士後期課程修了。博士(工学)。非手続的データベース操作言語の研究に従事。