データベースシステム 114－15
（1998. 1. 20）

関係データベースに基づく半構造データの実現と管理

遠山 元道
toyama@ics.keio.ac.jp

慶應義塾大学 理工学部 情報工学科

Abstract

この論文で、関係データベースを中心に据えた半構造データ管理の可能性と限界を論じる。

半構造データは、無構造データと強構造データの中間に位置付けらる。これらの管理を論じるには、強構造データでは禁止される構造の不規則性をある程度許容しつつ、無構造データにはない構造の規則性を活用しなければならない。

著者等は、関係データベースからのHTML文書等の動的生成ソフトウェアの開発経験から、柔軟性に乏しい関係データベースを情報の格納媒体としながらも、ある種の不規則構造の取扱が可能であることを学んだ。特に、動的質問呼び出しをWWW上での操作と対応付けて遅延評価することで、データベース上の再帰的巡行 (recursive navigation) を実現することができる。

Relation-based Management of Semi-Structured Data

Motomichi Toyama
toyama@ics.keio.ac.jp

Department of Information and Computer Science, Keio University

Abstract

In this paper, we will discuss the possibilities and the limitations of relation based management of semi-structured data.

Semi-structured data are explained as something in the midway from unstructured data to highly structured data. On discussing management of semi-structured data, we should exploit the regularity that are not expected in unstructured data, while allowing irregularities forbidden in highly structured data.

From our past experiences on providing HTML or other kinds of appliational data as a dynamic heterogeneous view of a relational database, we have recognized that some sorts of irregularities could be handled and realized nicely in this architecture, leaving relational database itself as a rigid data repository. Especilly, the dynamic invocation of a query, along with the lazy evaluation nature of Web-like interaction, is presented as an elegant solution for navigational accesses in recursive data references.

1 はじめに

S. Abiteboul は [3] において半構造データを、画像のような無構造ではなく、またスキーマの固定した従来のデータベースよりは柔軟な構造をもつデータとゆるやかに規定し、例として HTML で書かれた Web ページや bibtex ファイルなどを挙げている。どんな Web ページでも半構造というわけではなく、単に文章を書き連ねたようなページは明らかに無構造である。Web を利用した商品の紹介や価格表など、同様の情報の繰り返しのあるページならば半構造といえるだろう。このことは、あえて LaTeX ソースでなく、bibtex ファイルが半構造データの例として挙げられていることからも分かる。

同様の情報の繰り返しとは言っても、bibitex ファイルで扱う文献情報などを見ると、データ毎に構造上の大きなばらつきがある。著者名が 1 つ (単著) であったり、複数であったり、また著者名の無いデータもある。論文などでは、それ自体のタイトルの他に、論文集の名称やページ数が不可欠である。このようなばらつきを許しつつ、なお全体的に共通のパターンをもつ「ゆるやかな規則性」が半構造を特徴づける最大の要因といえる。

半構造データの管理とは、このようなゆるやかな規則性をもったデータを対象とし、従来のデータベースシステムのように検索機能や、利用者ビューを提供する機能を実現しようとするものである。後者をデータの再利用または再構成と呼ぶ。例えばある Web サイトが、アルファベット順の会議録のリストから、その本に含まれる論文の一覧表へリンクする構成だったとしよう。この情報を逆転し、アルファベット順の著者名のリストから、その著者の論文のリストのページへリンクする Web サイトを提供しようとすれば再構成が必要である。このような要求を、半構造データの構造多様性への要求と呼ぶことにする。

一方、bibitex ファイルで提供されている半構造の文献データを Web ページや表計算のワークシートに変換するような要求を、半構造データのメディア多様性への要求と呼ぶことにする。このように複数のメディアに渡る半構造データの提供を自動化することにより、異なるメディアを使用する利用者への矛盾のない情報の提供が可能となる。

このような半構造データの検索／再構成を実現するにあたり、大きく分けて三通りのアプローチが考えられる。ひとつは HTML や bibtex ファイルなどの応用データそのものをオリジナルとする応用データアプローチ。第二に半構造データを扱う専用の DBMS やオブジェクト指向 DBMS を使う構造化 DB アプローチ。もう一つは、関係型 DBMS に情報を格納し、これから応用別データを生成する関係 DB アプローチである。

応用データアプローチは、多くの WEB ページのように応用の現場においてアドホックに作られたデータをそのままソースとし、ラッパー (Wrapper) によってその内容を解釈し、検索や再利用の対象とするものである。この立場には Harvest, TSIMMIS, Information Manifold など多くの研究例がある。あるがままのデータを利用するため、データの再利用性などはデータ自体の状態とラッパーの性能に大きく依存することになる。

一方、Stanford 大学の lore プロジェクト [2] や、Pensilvania 大の P. Buneman, D. Suciu (現 AT&T) など [5] は半構造データを格納するための専用のデータモデルに基づく DBMS を開発し、データはその半構造データベースに存在するという立場をとる。従来の DBMS 研究と同様に DBMS の閉じた世界での検索や再構成を論じ、実世界にすでに存在する応用データの利用や、処理結果の応用データへのマッピングに関しては別個の問題とする。

本論文で提案する方式では、関係データベースを情報の格納母体とする第三のアプローチをとる。第一、第二のアプローチと違って格納データが特定の構造を持たず、応用データの生成時に構造化を行うためデータの再構成が不要で、構造多様性の要求に自由に応じられる。Universita di Roma Tre の P. Atzeni 等の Araneus プロジェクトも関係 DB アプローチによっている。

著者等のグループは、過去数年間に渡って関係データベースの内容をソースとするデータベース出版のモデリングについて、多角的に研究を進めてきた。これは、関係データベースから応用データを生成することに相当し、現に HTML、Java などの WEB コンテンツ記述、LaTeX ソースによる印刷文書 (レポート) の生成、Visual Basic ソースの出力による表計算データの生成などを、SQL の単純な拡張によって統一的に実現してきた。

本論文ではこのような経験に基づき、異種メディアに渡る半構造データの統合管理アーキテクチャを提案し、その可能性と限界を議論する。

2 SQL+TFE=SuperSQL

本論文で提案する方式は、著者らが [8, 9, 10, 11] 等で提案してきた TFE と呼ばれる SQL の拡張に基づいている。TFE によって拡張された SQL を今後 SuperSQL と称することにする。

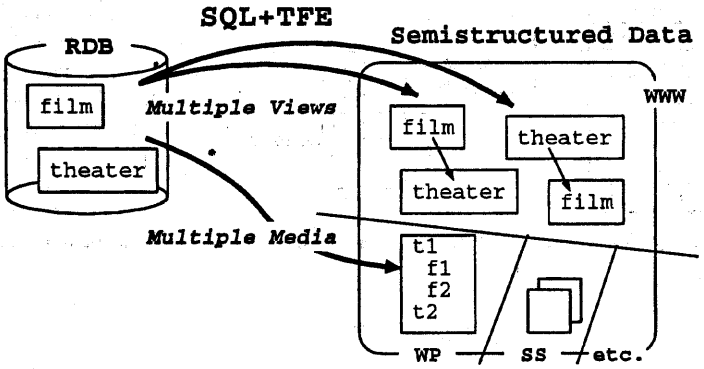

Figure 1: Overview

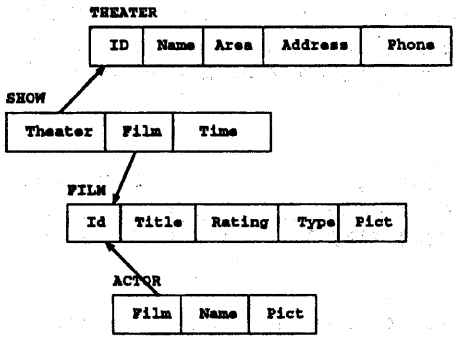

Figure 2: Sample Database Schema

SuperSQL により、関係データベースの質問結果を構造化し、さらにそれを Web, LaTeX, Excel ワークシートなど、各種の応用データに変換して出力することができる。この節では TFE の概要を説明し、後の節で半構造データに見られる不規則な構造の取り扱いについて論じる。

2.1 TFE: Target Form Expression の概要

TFE は、SQL をはじめとする関係データベース質問言語のターゲットリストの拡張である。ターゲットリストが単に属性をカンマで区切って並べたリストであるのに対し、TFE は生成される対象の構造を規定する接続子、反復子などの演算子をもつ一種の式表現である。個々の接続子と反復子には次元が対応する。質問の結果は本質的には多次元空間に埋め込まれた入れ子関係表である。それぞれの次元は、生成対象となるメディアのデータ構造の構成子に対応づけられている。例えば生成対象が HTML の場合、最初の 2 つの次元は<table>構造の行と列にそれぞれ対応し、第 3 次元はハイパーリンクに対応づけられている。

SuperSQL では、通常の SQL の SELECT <targe list> 句との違いをはっきりさせるために、GENERATE <media> <TFE> 句を導入している。今日までに、生成対象のメディアの指定として、LATEX, HTML, JAVA, EXCEL, TCLTK, O2C, SQL について試作を行っている。

2.2 接続子と反復子

コンマ (,)、感嘆符 (!)、パーセント記号 (%) の三つの二項演算子が、第一次元から第三次元の接続子である。これらは、オペランドによって生成された結果を、各々の次元の方向に接続する。

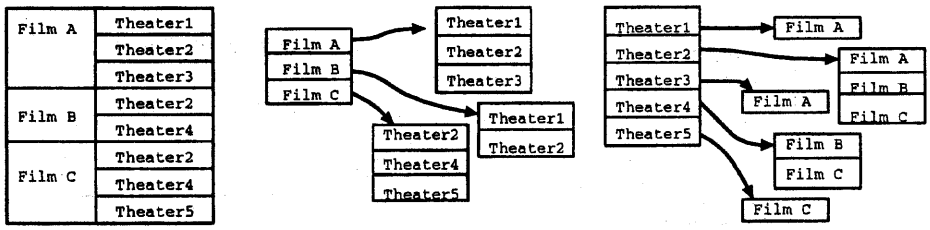

Figure 3: Multiple Views

一対の角括弧 ([ ]) の後に上記の接続子のいずれかを添えたものが、その接続子の次元の反復子である。反復子は、括弧内のオペランドによって生成されるものを、その次元の方向に連続的に接続する働きを持つ。反復子を入れ子にし、内側の反復子を何らかの単純属性と接続することにより、グルーピングを指定することができる。中括弧の対 ({ }) は演算子の結合順序の指定に使用する。

TFE は入れ子構造を抽象的な次元と一体化した点で、入れ子構造だけを扱う他の非第一正規形関係モデルの考え方と異なる。TFE の入れ子構造は、Abiteboul と Bidoit の VERSO モデル [1] や、Roth, Korth, Silberschatz らによる PNF [7] よりわずかに制約の少ないものとなっている。

2.3 例

TFE による構造化の例を示すため、図 2 のスキーマによる映画・映画館データベースを用いる。最初の 3 例は、TFE の基本的なレイアウト指定の表現力を説明している。これらの例で FROM 句と WHERE 句は共通で、また TFE に含まれる属性集合も同じである。したがって、これらの結果は同一の情報内容の異なる表示の例といえる。

```
Example 1:   GENERATE HTML [f.title, [t.name, s.time]!]!
             FROM  film f, theater t, show s
             WHERE f.id=s.film and s.theater=t.id

Example 2:   GENERATE HTML [f.title% [t.name, s.time]!]!
             FROM  film f, theater t, show s
             WHERE f.id=s.film and s.theater=t.id

Example 3:   GENERATE HTML [t.name% [f.title, s.time]!]!
             FROM  film f, theater t, show s
             WHERE f.id=s.film and s.theater=t.id
```

図 3 1), 2), 3) は、これらの質問によって生成される WEB ページの構造を表している。例 1 は 1 ページに結果を納めるレイアウトで、例 2 と例 3 は深さ方向 (第三次元) の結合子 (%) を利用して複数ページを一括生成するレイアウト指定である。例 2 の出力は木構造状に相互にリンクされた 4 ページからなる。%によって、それぞれの葉ページは根ページに表示される映画のタイトルの後ろ (奥側) に配置、すなわちリンクされる。例 3 では根ページは映画の索引の代わりに映画館の索引になっている。

プレゼンテーションの構成は、応用の目的に応じてこのように質問内の TFE によって指定することができる。

TFE を用いた質問は何段階もの索引をもつ互いにリンクされた複数の WEB ページを一括生成することができる。RDB/WWW 連係のためのソフトウェアは市場に数十も現れているが [6]、リンクを含む複数のページの生成をこのように簡単に宣言的に行えるものは他に存在しない。

なお、例 1 のメディア指定を HTML から LATEX に変えることにより、WEB ページではなく、同じ内容とレイアウトのレポートを印刷することができる。このようにしてメディア多様性への要求に応えることができる。

```
(titles.tfe)
GENERATE HTML
  [ verb(November Movies) !
    [ f.title %
        invoke(film, conc("title=",f.title)) ]!
  ]!
FROM  film f

(film.tfe)
GENERATE HTML
  [f.title !
    (imagefile(f.pict,GIF/),
     ((verb(Stars),[a.name %
        invoke(actor,conc("a.name=",a.name))]) !
      (verb(Type),f.type )!
      (verb(Theaters) %
        invoke(theaters,conc("f.title=",f.title))
      )) !
  ]!
FROM  film f, actor a
WHERE f.id=a.film
```

```
(actor.tfe)
GENERATE HTML
   [imagefile(a.pict,GIF/),
     (a.name !
      [f.title % invoke(film,conc("f.id=",f.id))]!
     ) ]!
FROM actor a, film f

(theater.tfe)
GENERATE HTML
   [t.name ! imagefile(t.map,GIF/) !
     [f.type, [f.title %
          invoke(film,conc("f.id=",f.id))]!]! ]!
FROM theater t, film f

(theaters.tfe)
GENERATE HTML
   [t.area, [t.name %
          invoke(theater,conc("t.id=",t.id))]!]!
FROM theater t
```

Figure 4: TFE Query Files for Dynamic Invocation

2.4 TFE 質問の動的呼出し

永藤、瀬戸、遠山は [8] で TFE に動的質問呼び出し関数 TFEfile を導入した。その後若干の仕様変更を加え、現在 **invoke(file, str1, att1, str2, att2,...)** として実装されている [10]。第一引数の file は呼び出される質問を記述したファイルを指定する。それ以降の引数 str_i, att_i は 2 つずつペアで文字列と属性を指定し、呼び出される質問に付加する条件を与える。**invoke** 関数はハイパーリンクを生成する深さ方向連結子 (%) の右辺にだけ許され、アンカーポイントのボタンを押した時に CGI スクリプトを通じて質問がデータベースに送られ、結果として生成される WEB ページがブラウザに送られる。

図 4 の質問 **titles.tfe** を実行すると、ブラウザには映画のタイトルの一覧が表示される。これらの内の一つを選んでボタンを押すと、**film.tfe** にタイトルを指定する条件が付加された上で実行される。

このようにして、図 5 に描かれた質問のネットワークが、ブラウザ上での利用者の選択によって次々に実行される。このようにして再帰的巡行が実現されるので、データベースシステム自体が再帰質問処理の機能を備える必要はない。

3 不規則性の取り扱い

前節で述べたように、動的質問呼び出しの行える SuperSQL では、関係データベースシステム自体に手を加えることなく再帰的巡行を実現することができる。

S. Abiteboul は [3] で関係 DBMS が半構造データに適さないと述べ、その理由として欠損値、多値、再帰的質問などを関係データベースが取り扱えないことを挙げている。これらは半構造データに頻繁に見られる構造上の不規則性である。再帰的質問については前節で見た通り、invoke 関数を用いて質問を再帰的に呼び出すことにより再帰的巡航を実現できることが分かった。実際、この使用感は半構造データを直接扱う Stanford 大学の lore システム上での巡航となんら変わるところは無い。以下で、多値と欠損値の扱いについて述べる。

3.1 多値の扱い

多値が常である属性、たとえば映画に対して出演俳優の名前などは明らかに 1 対多の関係にあるため、データベース設計の段階で独立したエンティティとして扱われる。この場合、多値の部分の TFE は自然に反復子で囲うことになる。

```
GENERATE HTML [f.title, f.director, [a.name]!]!
FROM film f, actor a, cast c
WHERE f.id=c.film and a.id=c.actor
```

問題は、ほとんどの場合に単値でありながら、時に多値となる属性で、たとえば映画における監督の名前の例が挙げられる。これを上記のような質問にしておくと、タイトルと監督名の組合せに対してグルーピングを行うため、監督が 2 人の連名の場合には同じ映画について 2 つのグループができてしまう。

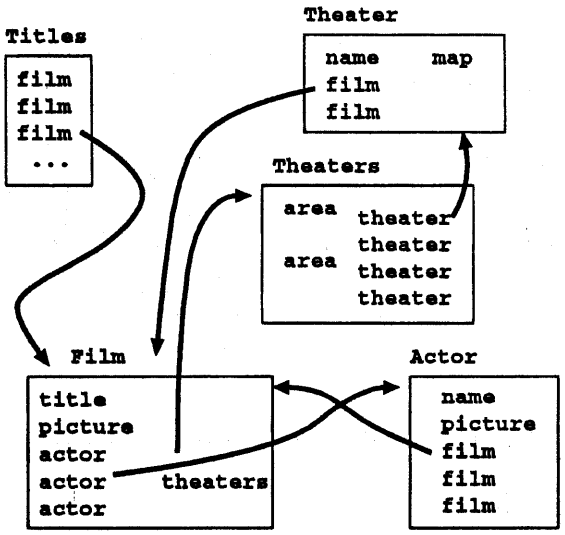

Figure 5: Dynamic Invocation

このような場合、下記のように監督名を単独で反復してやることにより、監督が何名の場合でも映画のタイトル毎に1つのグループが生成されることになる。

```
GENERATE HTML [f.title, [f.director]!, [a.name]!]!
FROM film f, actor a, cast c
WHERE f.id=c.film and a.id=c.actor
```

もちろんこのような例外がある場合、関係 film において属性 id が主キーであるように設計されていると、データの入力時に受け付けられない。設計時に予測できないこのような例外については一時的に主キーの指定をはずすか、id と director の複合キーに変更してデータを入力し、後で設計を変更(監督を独立のエンティティとする)して対処することになる。

3.2 欠損値の扱い

単純属性の欠損値の場合、SQL においても NULL 値を埋め草としておけば特に問題は生じない。問題が生じ得るのは、結合を含む質問で相手のタプルが存在しない場合である。上記の例で、出演俳優の無いアニメ映画の場合など、映画自体の情報が出力から欠落してしまう。このようなことを防ぐために、従となる関係との結合に際して左外結合 (left outer join) を使用し、相手が存在しない場合でも主となる関係のタプルが結果に現れるようにすればよい。従となる関係の部分を invoke 関数によって動的に生成する場合には、結果が空となっても問題は生じない。

3.3 アドホックな属性

関係データベースにおいては、スキーマ設計がデータ入力に先行することは必須条件である。このことはオブジェクト指向 DB でも同様で、スキーマを前提としない半構造専用の lore 等と大きく、しかも本質的に異なる点である。しかしながら、逆に lore 等ではスキーマが存在しないために、データ入力の度に属性名と属性値の両方を入力してやらなければならない。同様の手間をかけるならば、関係データベースにおいても逃げ道は存在する。

例えば、映画エンティティに対し、アドホックな属性が発生すると予測される場合、film_adhoc(id,att,val) のような関係を用意しておき、例外的な属性が生じた場合に、その映画の id と、一時的な属性名、その属性値の三つ組を入力しておくことにする。また film に対して film_adhoc を外結合することで、このような例外属性を出力に含めることが可能である。

4 むすび

本論文では、関係データベースを中心に据えた半構造データ管理の可能性と限界を論じた。関係データベースの情報に対してSuperSQL質問を実行することにより、各種の応用データを直接生成することができる。この際、グルーピングの仕方の指定により、多様な構造に対する要求に対称的に応じることができる。また、出力媒体の指定により、多様なメディアに対する要求にも応じることができる。

半構造データにおいては、WWWなどのハイパーテキストでよく見られるように、映画→俳優→(他の)映画→映画館→地図のように、データ間のつながりをとりとめもなくたどる使い方が必要となる。動的質問呼び出しをWWW上での操作と対応付けて遅延評価することで、データベース上の再帰的巡航(recursive navigation)を実現することにより、このような応用に対処することができる。

また、欠損値、多値、アドホックな属性などのデータの不規則性については、NULL値やTFEの反復子の利用、外結合の利用などによってある程度対処可能であることを示した。このことから、不規則性が常態ではなく、例外的に現れるような半構造データの応用については、SuperSQLの利用により、関係データベースを格納媒体とすることが適切であることを示した。関係データベースを使用することにより、従来から蓄積されたデータを容易に活用することができ、幅広い応用に適するものと考えられる。

謝辞

この研究の一部は、文部省COE形成プロジェクト「デジタルメディアの基礎と応用」によっている。

References

[1] S. Abiteboul, N. Bidoit, *Non First Normal Form Relations: An Algebra Allowing Data Restructuring, J. Comp. and Sys. Sci. 33, 361-393, (1986)*

[2] S. Abiteboul, D. Quass, J. McHugh, J. Widom, J. Wiener, *The Lorel Query Language for Semistructured Data*, ftp://db.stanford.edu//pub/papers/lorel96.ps

[3] S. Abiteboul, *Querying Semi-Structured Data*, in Proc. ICDT '97, 1997

[4] P. Atzeni, G. Mecca, P. Merialdo, Semistructured and Structured Data in the Web: Going Back and Forth, Proc. Workshop on Semistructured Data Management, in conjunction with SIGMOD 97, 1997

[5] P. Buneman, S. Davidson, M. Fernandez, D. Suciu, *Adding Structure to Unstructured Data*, in Proc. ICDT '97

[6] C. Lang, J. Chow, *Database Publishing on the Web and Intranets*, Coriolis Group Books (1996)

[7] M. A. Roth, H. F. Korth, A. Silbershatz, *Extended Algebra and Calculus for Nested Relational Databases*, in TODS 13, 4, ACM (1988), pp389-417

[8] 永藤, 瀬戸, 遠山, TFEによるHTMLソースの動的生成 Proc. 第7回データ工学ワークショップ(DEWS '96), 電子情報通信学会 データ工学研究会編, (1996), pp37-42

[9] T. Seto, T. Nagafuji, M. Toyama, *Generating HTML Sources with TFE Enhanced SQL*, in Proc. SAC '97, ACM (1997), pp96-105

[10] 遠山, TFEによるHTML生成質問における質問分割の効果と等価性, Proc. アドバンストデータベースシンポジウム '97, pp87-94

[11] M. Toyama, *Three Dimensional Generalization of Target List for Simple Database Publishing and Browsing*, in Proc. 3rd Australian Database Conference, Research and Practical Issues in Database, World Scientific Pub. Co.(1992), pp139-153

付録：TFEの構文と出力生成規則

<GENERATE 句> ::= GENERATE <メディア指定> <目的式>
<目的式> ::= <反復式> | <反復式><結合子><目的式> | {<目的式>}
<反復式> ::= [<TFE 式>]<結合子>
<TFE 式> ::= <基本項目> | {<TFE 式>} | <TFE 式><結合子><TFE 式>
<結合子> ::= , | ! | %

目的式をもつ質問文を実行する際、結果として得られる構造化(レイアウト)された表は、以下の概略手順によって構成する。なお、目的式(およびTFE式)の次数を、それに含まれる基本項目の数と定義する。

[整形出力生成規則]
入力：目的式　　出力：整形出力

1. 目的式中の基本項目を、出現順にコンマで連結してターゲットリストを作り、ホストとなる質問言語の文法に基づいた質問文を構成する。その実行結果を対象関係とする。この対象関係と与えられた目的式に、規則2または規則3を適用し、結果を得る。

2. 目的式1が単独の反復式([TFE式1]結合子1)の場合。対象関係を、TFE式1によって、規則2、3、4のいずれかで処理して得られる複数の出力を、結合子1で示される方向に逐次結合したものが結果である。

3. 目的式1が反復式1結合子1目的式2の形の場合。反復式1の次数をkとする。対象関係を、最初のk個の属性と残りの属性にそれぞれ射影して得られる対象関係R1とR2について、R1を反復式1で、規則2によって整形した結果と、R2を目的式2で、規則2、3、4のいずれかで整形した結果を、結合子1で結合したものが結果である。

4. 規則2、3が当てはまらない場合。対象関係Rを反復式でないTFE式1で整形する。このとき、一般に結果は複数の整形結果の集合となる。TFE式1中の最上位レベルにある基本項目(複数可)に対応する属性の実現値の組合せを基準としてRのタプルをグルーピングし、残りの属性からなる部分(商)関係の集合を、TFE式1中の反復式のリストによって規則2または3によって整形した結果を、グルーピング基準項目の値とともに結合する。このような整形結果は、グルーピング基準項目の実現値の組合せ数だけ得られるので、その集合を結果とする。