フィルタリング SQL におけるランキングについて

澤井里枝[†]　塚本昌彦[†]
寺田　努[††]　西尾章治郎[†]

近年，さまざまなデータ放送の普及により，ユーザが必要とするデータのみを自動的に抽出する情報フィルタリング技術に対する注目が高まってきた．そこで筆者らはこれまで，フィルタリングのためのユーザ要求記述言語であるフィルタリング SQL を提案し，ユーザの嗜好やデータ管理ポリシーを自由に表現できるようにした．しかし，これまでの研究では，主にフィルタリング SQL の記述構文を設計するのみであり，ある受信データ集合に対して実際にどのようなフィルタリング結果が得られるかは，完全には明確化されていない．そこで本稿では，フィルタリングポリシーの記述構文の中でも特に頻繁に利用されるランキングの構文について，その意味を明確にすることを目的とする．本研究では，フィルタリング SQL の意味を表現するために，フィルタリングを関数として表現したフィルタリング関数を用いる．筆者らがこれまで明らかにしてきたフィルタリング関数の数学的性質を利用することで，ユーザが記述した要求を実行するフィルタリングの特性が明らかにできるため，環境に応じた効率的な処理を実現できる．

On Ranking in FilteringSQL

Rie SAWAI,[†] Masahiko TSUKAMOTO,[†] Tsutomu TERADA[††]
and Shojiro NISHIO[†]

In recent years, due to the increasing popularization of data broadcasting, there is an increasing demand for filtering techniques that automatically extract only the necessary data. In our previous works, we have proposed FilteringSQL which is a user request description language for information filtering. FilteringSQL enables users to express their preferences and policy of data management. However, though we designed the syntax of FilteringSQL in previous works, it has not completely cleared what filtering results users will get from given data sets. In this paper, we clear the semantics of ranking sentence, which is frequently used to express filtering policy. We express the semantics of FilteringSQL using filtering function that represents information filtering as a function. By applying the mathematical properties of filtering function, we clarify the characteristics of filtering methods that carry out the request users described. From the properties of filtering methods, more efficient processing can be achieved according to environments.

1. はじめに

近年，ネットワークのブロードバンド化や，放送のデジタル化および多チャンネル化により，さまざまな放送型サービスが提供されるようになった[4),5)]．このような環境では，多様で膨大なデータを受信できるが，一般にユーザが必要とする情報はごく一部に限られているため，受信データから必要なデータを探し出すことはコストの高い作業である．そのため，自動的に受信データの取捨選択をする情報フィルタリング技術に対する要求が高まっている[1)~3),6)]．そこで筆者らは，フィルタリングに対するユーザの要求を記述するための**フィルタリング SQL** を提案した[7)]．フィルタリング SQL とは，データベースへの問合せ言語である SQL(Structured Query Language) をフィルタリングのために拡張した言語である．フィルタリング SQL により，ユーザはコンテンツの嗜好やデータ管理ポリシーなどを自由に表現できる．しかし，文献 7) は，フィルタリング SQL の記述構文と，それらの構文の用途を示したのみであるため，ある受信データ集合に対して実際にどのようなフィルタリング結果が得られるかは明確でなかった．そこで本稿では，**フィルタリング関数**を用いることで，フィルタリング SQL の意味を明確化することを目的とする．

フィルタリング関数とは，フィルタリングを関数として定義したものである．筆者らは，フィルタリングの性質をフィルタリング関数が満たす制約条件として定

† 大阪大学大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University
†† 大阪大学サイバーメディアセンター
Cybermedia Center, Osaka University

性的に表現することを可能にし[8),9)]，それらの制約条件を満たすフィルタリング関数の数学的性質を明らかにすることで，さまざまなフィルタリングの特性を明確にしてきた[10)~13)]．フィルタリングの数学的基盤を構築することで，キーワードマッチングや関連フィードバックなど，それぞれ独自の手法によってデータのフィルタリングを行っている従来のフィルタリング機構を定性的に表現できる．また，フィルタリングの性質の定性的な評価や処理手法の最適化，宣言的なフィルタリング言語の設計なども可能となる．

文献 14) では，フィルタリング関数により，フィルタリング SQL の基本的な構文の意味を明確化した．しかし，文献 14) が取り扱ったのは一部の基本的なプリファレンス構文のみであるため，フィルタリング SQL の意味論は未だ完全でない．そこで本稿では，フィルタリング SQL の構文の中でも特に頻繁に利用されるランキングの構文についてその意味を明確にする．ランキングとは,「最新のヒット曲ベスト 10 の音楽データが欲しい」や,「配信日時が最も新しい 100 個のニュースが欲しい」といったように，抽出するデータ数を定めたり，重要なデータを必要な順序に並べて提示する場合などに，頻繁に用いられるフィルタリングポリシーである．また，本稿では，単一のランキング記述だけでなく，他の構文と組合せた記述の意味も明確にする．記述の意味を表現するためにフィルタリング関数を用いることで，これまで筆者らが構築してきたフィルタリングの数学的基盤を適用し，ユーザの記述を実行するフィルタリングの性質を明らかにできる．フィルタリングが満たす性質に応じて，環境に適した効率的な処理が実現可能である．

以下，第 2 章でフィルタリング関数の概要を述べる．第 3 章では，ランキングの構文についてその意味を明確にし，記述内容を実行するフィルタリングの性質を明らかにする．第 4 章では，ランキングの構文とその他の構文を組合せた記述の意味を明確にする．第 5 章では，本稿で取り扱った構文の記述例とその意味を示し，各記述を実行するフィルタリングの特性について考察する．最後に第 6 章でまとめを行う．

2. フィルタリング関数

本章では，フィルタリング関数とその基本的な性質について述べる[8)~13)]．

2.1 フィルタリング処理の分類

あるフィルタリング手法が与えられたとき，実際の処理方法は以下に示すいくつかのパターンに分類できる．

データアイテムを受信する度に，新たな受信データと前回のフィルタリング結果を合せてフィルタリングする処理方法を**逐次処理**とよぶ．逐次処理では，一度蓄積されたデータも，データ受信時に再度フィルタリングする．それに対し，放送データを受信側にためておいてから一括してフィルタリングする処理方法を**一括処理**とよぶ．また，データ集合を 2 つ以上の任意の集合に分割して各々フィルタリングし，結果をマージしたものをフィルタリング結果とする処理方法を**分配処理**とよぶ．さらに，分配処理の結果を再びフィルタリングする処理方法を**並列処理**とよぶ．

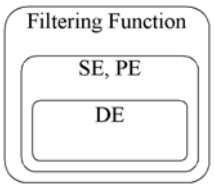

図 1 等価性間の関係

2.2 フィルタリング関数の性質

データアイテムの集合を $\mathbf{T}$ とする．フィルタリング関数とは，任意の $T \subset \mathbf{T}$ に対し☆，以下の 2 つの条件を満たす $2^{\mathbf{T}}$ 上の関数 f のことをいう[8),9)]．

減少性 (D: Decreasing)

$$f(T) \subset T$$

ベキ等性 (ID: Idempotent)

$$f(f(T)) = f(T)$$

減少性 D は，関数を適用した結果が元のデータ集合に含まれるデータアイテムのみであることを表す．ベキ等性 ID は，一度関数を適用すると，何度その関数を適用しても結果が変化しないことを表す．また，フィルタリング関数について以下のような性質が定義されている．

逐次等価性 (SE: Sequential Equivalence)

$$f(S \cup T) = f(S \cup f(T))$$

分配等価性 (DE: Distributed Equivalence)

$$f(S \cup T) = f(S) \cup f(T)$$

並列等価性 (PE: Parallel Equivalence)

$$f(S \cup T) = f(f(S) \cup f(T))$$

ここで，S, T は $\mathbf{T}$ の任意の部分集合とする．逐次等価性は一括処理と逐次処理の結果が等価であることを意味する．同様に，分配等価性は一括処理と分配処理の結果が等価であり，並列等価性は一括処理と並列処理の結果が等価であることを意味する．これまでに筆者らは，これらの等価性間に**図 1** に示すような包含関係があることを明らかにした[8),9)]．図 1 より，一括処理と分配処理の結果が等価であるフィルタリングは，逐次処理や並列処理の結果とも等価となることがわかる．また，一括処理と逐次処理の結果が等価であるフィルタリングは，並列処理の結果とも等価となり，一括処理

☆ 本稿では $A \subset B$ は A が B の部分集合である (A = B の場合を含む) ことを意味するものとする．

表 1　合成フィルタリング関数 $f \circ g$ の性質

$f \setminus g$	DE	SE, PE
DE	DE, SE, PE	¬DE, ¬SE, ¬PE
SE, PE	SE, PE, ¬DE	¬DE, ¬SE, ¬PE

表 2　合成フィルタリング関数 $f \circ g$ と $g \circ f$ の包含関係

$f \setminus g$	DE	SE, PE
DE	=	⊂, ¬ ⊃
SE, PE	⊃, ¬ ⊂	¬ ⊂, ¬ ⊃

表 3　和フィルタリング関数 $f^{\vee} g$ の性質

$f \setminus g$	DE	SE, PE
DE	DE, SE, PE	SE, PE, ¬DE
SE, PE	SE, PE, ¬DE	SE, PE, ¬DE

表 4　積フィルタリング関数 $f^{\wedge} g$ の性質

$f \setminus g$	DE	SE, PE
DE	DE, SE, PE	¬DE, ¬SE, ¬PE
SE, PE	¬DE, ¬SE, ¬PE	¬DE, ¬SE, ¬PE

と並列処理の結果が等価であるフィルタリングは，逐次処理の結果とも等価となる．図 1 に示す性質間の関係より，ある性質を満たすフィルタリングが他の性質を満たすかどうかが判断でき，環境に応じてより効率的な処理方法に変換できる．

2.3　ランキング関数

文献 10) において，**ランキング関数**を次のように定義した．

ランキングとは，ユーザの嗜好に応じて受信データを重要な順序に並べ，最も重要なデータを特定の数だけ選択する手法である．ある全順序 $R = (\mathbf{T}, <)$ に対する n ランキング関数 f とは，任意の $X \subset \mathbf{T}$ に対し，ある $a \in \mathbf{T}$ について $f(X) \stackrel{\triangle}{=} \{x \in \mathbf{T} | x < a\} \cap X$ と定義される**度数** n の関数である．関数 f が度数 n であるとは，任意の $X \subset \mathbf{T}$ に対して

- X が無限集合，あるいは X が有限集合であり $|X| \geq n$ ならば $|f(X)| = n$．
- X が有限集合であり $|X| < n$ ならば $f(X) = X$．

の 2 つの条件が成立することである．度数 n は蓄積するデータの個数を意味し，受信データの数がその数に満たない場合は，全ての受信データを蓄積する．また，全順序 $R = (\mathbf{T}, <)$ は，全データアイテムの重要度の順序を表す．“$<$” はデータアイテム間の順序を表し，“$d_1 < d_2$” は d_1 の方が d_2 よりも重要度が高いことを意味する．ランキング関数は，度数に合せてデータアイテム a よりも重要度が高いデータを蓄積するため，フィルタリングするデータ集合によって a は変化する．

ランキング関数に関して，以下の定理が成立する[10]．

定理 1　フィルタリング関数 f がある全順序 $(\mathbf{T}, <)$ に対する n ランキング関数であることと，f が度数 n で逐次等価性を満たすことは同値である．　□

2.4　合成フィルタリング関数と和積フィルタリング関数

実際のフィルタリングでは，複数の手法を組合せて用いるのが一般的である．そこで筆者らは，そのようなフィルタリングの特性を明確にするために，フィルタリング関数の合成関数や和積関数の性質を明らかにした．

合成フィルタリング関数が満たす性質を**表 1** に示す[11]．表 1 中の各要素は，フィルタリング関数 f, g がそれぞれ行，列の性質をもつとき，合成関数 $f \circ g$ が満たす性質を表す．また，“¬” はその性質を必ずしも満たさないことを表す．また，合成順序を交換したフィルタリング結果の包含関係を**表 2** に示す[12]．表 2 中の “=” は，フィルタリング関数 f, g がそれぞれの性質を満たすとき，任意の $S \subset \mathbf{T}$ に対して $f \circ g(S) = g \circ f(S)$ であることを示し，“⊂” は任意の $S \subset \mathbf{T}$ に対して $f \circ g(S) \subset g \circ f(S)$ であることを示す．また，“¬ ⊂” は，ある $S \subset \mathbf{T}$ に対して $f \circ g(S) \not\subset g \circ f(S)$ であることを示す．

さらに，任意の $S \subset \mathbf{T}$ に対して $f^{\vee} g(S) = f(S) \cup g(S)$ と定義される $f^{\vee} g$ を f と g の**和フィルタリング関数**とよび，$f^{\wedge} g(S) = f(S) \cap g(S)$ と定義される $f^{\wedge} g$ を f と g の**積フィルタリング関数**とよぶ．これらの和積フィルタリング関数が満たす性質を，それぞれ**表 3**，**表 4** に示す[13]．

3.　ランキングの意味論

本章では，フィルタリング SQL の 1 構文であるランキングの意味付けを行う．

まず，フィルタリング SQL の基本構文を以下に示す[7]．

```
EXTRACT   < 属性 >
FROM      < リソース >
WHERE     < プリファレンス >
          < 付加条件 >
```

EXTRACT 句では，抽出されたデータのうち蓄積する属性を指定する．FROM 句にデータ放送名などのデータリソースを指定し，WHERE 句にユーザの嗜好を表現するプリファレンス記述をする．また，必要に応じてデータの蓄積に関するオプションとなる条件を付加できる．

一般にデータ放送では，放送すべき全てのデータアイテムを一度に送信するのではなく，スケジュールにしたがっていくつもの部分集合に分けて送信する．したがって，部分集合の分け方，およびそれらを送信するスケジュールに応じて，ユーザが得られるフィルタリング結果は変化する．ユーザが常に記述した通りのフィルタリング結果を得るには，全てのデータアイテムを一度にフィルタリングする必要がある．そこで，あるフィ

ルタリング SQL 記述が，全てのデータアイテムを一括処理することを**宣言的意味論** (Declarative semantics) とよぶ．しかし，過去に受信したデータアイテムと，将来受信するデータアイテムを集めて一度にフィルタリングすることは現実的でない．そこで，あるフィルタリング SQL 記述が，データアイテムを受信する度に逐次処理することを**手続的意味論** (Procedural semantics) とよぶ．

以下，3.1節でランキングの宣言的意味論，3.2節で手続的意味論をそれぞれ明確にする．ただし，本稿ではデータアイテムの取捨選択に焦点を絞った議論を行うため，蓄積する属性やデータリソースの記述は省略する．

3.1 宣言的意味論

データアイテムの集合を $\mathbf{T}$，値の集合を V とすると，属性は

$a : \mathbf{T} \to V$

という関数として定義する．特に，受信日時 $received : \mathbf{T} \to D$，最終アクセス日時 $accessed : \mathbf{T} \to D$，蓄積期限 $expire : \mathbf{T} \to D$，メタデータにより予め設定された有効期限 $m_expire : \mathbf{T} \to D$，データサイズ $size : \mathbf{T} \to R$ を属性とする．D，R は，それぞれ日時 $D \subset V$，数値 $R \subset V$ であり，半順序 $<$ を含むものとする[14]．

ここで，属性 $a : \mathbf{T} \to V$，データアイテム $t \in \mathbf{T}$ に対し，$a(t) \in V$ を $t.a$ と記す．また，キーワード k を $k \subset \mathbf{T}$，キーワード全体の集合を $K \subset 2^{\mathbf{T}}$ とする．

次に，ランキングの構文を用いた記述 σ を以下に示す．

```
EXTRACT
WHERE
  PREFER   k1 TO k2
  PREFER   k3 TO k4
    ⋮
  best(n)
```

Σ をフィルタリング SQL 記述の全体，F をフィルタリング関数とすると，セマンティクス関数とは，関数 $S : \Sigma \to F$ である．また，$\sigma \in \Sigma$ である．

ここで，k_1，$k_2 \in K$ に対して，プリファレンス記述 p が

p： PREFER k_1 TO k_2

のとき，関数 θ を

$\theta(p) = (k_1, k_2) \in K \times K$

と定義する．さらに，

P： PREFER 文の集合

に対して，

$\theta(P) = \{\theta(p) | p \in P\}$

とおく．$\theta(P)$ にサイクルがないとき，P は**無矛盾**であるとよぶ．

P が無矛盾であるとき，$\theta(P)$ を含む最小の半順序が一意に存在する．これを $\Theta(P)$ と記す．ここで，$R_P \subset \mathbf{T}^2$ を

$$R_P = \{(d_1, d_2) | d_1 \in k_1,\ d_2 \in k_2,\ (k_1, k_2) \in \Theta(P)\} \quad (1)$$

とする．R_P に対して次の補題が成立する．

補題 1　R_P は擬順序である．　□

次に，x，$y \in \mathbf{T}$ に対して

$$x \sim y \overset{def}{\Longleftrightarrow} (x,\ y),\ (y,\ x) \in R_P$$

とする $\sim$ は同値関係である．また，$\mathbf{T}_\sim$ を $\mathbf{T}$ の $\sim$ による商集合とすると，$R_{P\sim} \subset \mathbf{T}_\sim{}^2$ を R_P から自然に決まる半順序とする．つまり，

$$R_{P\sim} = \{([x]_\sim, [y]_\sim) | x \in k_1,\ y \in k_2,\ (k_1, k_2) \in \Theta(P)\} \quad (2)$$

ここで，

$[x]_\sim,\ [y]_\sim \in \mathbf{T}_\sim,\ ([x]_\sim, [y]_\sim) \in R_{P\sim}$
$\Longleftrightarrow (d_1,\ d_2) \in R_{P\sim},\ d_1 \sim x,\ d_2 \sim y$

である．

同値類 $u \in \mathbf{T}_\sim$ の要素間に順序関係を導入する．u 上の a 順序とは

$D^a_u = \{(d_1, d_2) | d_1,\ d_2 \in u,\ d_1.a > d_2.a\}$

とする．$R_{P\sim}$ および $\mathbf{T}_\sim$ の各元上の a 順序を用いて，PREFER 文集合 P に対する a 順序 $Q^a_P \subset \mathbf{T}^2$ を

$$(x, y) \in Q^a_P \Longleftrightarrow \begin{cases} x \not\sim y \text{ ならば } ([x]_\sim,\ [y]_\sim) \in R_{P\sim} \\ x \sim y \text{ ならば } (x, y) \in D^a_x \end{cases}$$

と定義する．

ここで，Q^a_P に対し，ある $e \in \mathbf{T}$，$E \subset \mathbf{T}$ について

$U(e) = \{z | (z, e) \in Q^a_P\}$
$U(E) = \cup_{e \in E} U(e)$

とおく．

ある $E \subset \mathbf{T}$ に対して，

$X = U(E)$

のとき，X を $\boldsymbol{P}$ **における閉集合**とよぶ．また，X が P における閉集合，かつ $S \subset \mathbf{T}$ に対して

$$\begin{cases} |X \cap S| = n & (S \text{ が無限集合，あるいは } S \text{ が有限集合であり } |S| \geq n) \\ X = \mathbf{T} & (S \text{ が有限集合であり } |S| < n) \end{cases} \quad (3)$$

を満たすことを $\Box_\gamma X$ と記す．ここで，$S^a(\sigma) : \Sigma \to 2^F$ を

$$S^a(\sigma) = \{X | \Box_\gamma X\} \quad (4)$$

と定義し，これをγ^a**セマンティクス**とよぶ．任意の $S \subset \mathbf{T}$ に対して，$f \in F$ が

$f(S) = X \cap S$

とおけるとき，f は**γ適合**であるとよぶ．f がフィルタリング関数であることは，容易に確かめられる．

特に，

$$\sum_{y \in X \cap S} y.b$$

の値を最大とする f^M を **γ^b 最大適合**とよび，その値を最小とする f^m を **γ^b 最小適合**とよぶ．

γ 適合である f について以下の補題を示す．

補題 2　γ 適合である関数 f で，分配等価性 DE を満たさないものが存在する．

≪ 証明 ≫ データアイテム集合 $\mathbf{T} = \{d_1,\ d_2,\ d_3,\ d_4\}$ に対して，$(d_1, d_2),\ (d_2, d_3),\ (d_3, d_4) \in Q_P^a$ とする．また，$n = 2$ であるとする．

このとき，$S = \{d_1,\ d_2\},\ T = \{d_3,\ d_4\}$ とすると，

$$f(S) = \{d_1,\ d_2\}$$
$$f(T) = \{d_3,\ d_4\}$$
$$f(S \cup T) = \{d_1,\ d_2\}$$
$$f(S) \cup f(T) = \{d_1,\ d_2,\ d_3,\ d_4\}$$

となる．したがって，分配等価性 $f(S \cup T) = f(S) \cup f(T)$ を満たさない．　□

補題 3　任意の $S,\ T$ に対して，$S \subset T$ のとき $X_T \subset X_S$ ならば，γ 適合である関数 f は逐次等価性 SE を満たす．ただし，P における閉集合 $X_S,\ X_T$ は $f(S) = X_S \cap S,\ f(T) = X_T \cap T$ を満たすものとする．

≪ 証明 ≫ 任意の $S,\ T \subset \mathbf{T}$ に対して，f が

$$f(S \cup T) = f(S \cup f(T))$$

を満たすことを示せばよい．

i)　$|S \cup T| < n$ のとき

$S \subset S \cup T,\ T \subset S \cup T$ が成り立つため，$|S| < n,\ |T| < n$ である．したがって，

$$f(S \cup T) = S \cup T$$
$$f(S) = S$$
$$f(T) = T$$

となるため，$f(S \cup T) = f(S \cup f(T))$ が満たされる．

ii)　$|S \cup T| \geq n$ のとき

$X_{ST},\ X_S,\ X_T,\ X_{SfT} \in \mathbf{T}$ を

$$f(S \cup T) = X_{ST} \cap (S \cup T)$$
$$f(S) = X_S \cap S$$
$$f(T) = X_T \cap T$$
$$f(S \cup f(T)) = X_{SfT} \cap (S \cup f(T))$$

を満たす閉集合とする．

f の減少性より $S \cup f(T) \subset S \cup T$ であるため，

$$X_{ST} \subset X_{SfT} \tag{5}$$

となる．同様に，$T \subset S \cup T$ より

$$X_{ST} \subset X_T \tag{6}$$

である．

また，f のベキ等性より，$f(T) = f(f(T))$ である．したがって，X_T は

$$f(f(T)) = f(T) = X_T \cap T$$

を満たす．したがって，$f(T) \subset S \cup f(T)$ であるため，

$$X_{SfT} \subset X_T \tag{7}$$

が成り立つ．ここで，(5)，(7) より

$$\begin{aligned} & X_{SfT} \cap (S \cup f(T)) \\ &= X_{SfT} \cap (S \cup (X_T \cap T)) \\ &= (X_{SfT} \cap S) \\ &\quad \cup (X_{SfT} \cap X_T \cap T) \\ &= (X_{SfT} \cap S) \\ &\quad \cup (X_{SfT} \cap T) \quad (\because (7)) \\ &= X_{SfT} \cap (S \cup T) \\ &\supset X_{ST} \cap (S \cup T) \quad (\because (5)) \end{aligned}$$

となり，

$$f(S \cup T) \subset f(S \cup f(T)) \tag{8}$$

が導き出される．ここで，f は γ 適合であるため，X_{ST} は

$$|X_{ST} \cap (S \cup T)| = n$$

を満たすデータ集合である．したがって，

$$f(S \cup T) \neq f(S \cup f(T))$$

と仮定すると，(8) より，

$$|X_{SfT} \cap (S \cup f(T))| > n$$

となり，f が γ 適合であることに矛盾する．ゆえに，

$$f(S \cup T) = f(S \cup f(T))$$

が成立する．　□

特に，次のランキング記述 σ_r

```
best(n, attribute, ASC | DESC)
```

に対して，第 3 引数が ASC のとき

$$R_{attribute} = \{(d_1, d_2) | d_1, d_2 \in \mathbf{T},\ d_1.attribute \leq d_2.attribute\} \tag{9}$$

DESC のとき

$$R_{attribute} = \{(d_1, d_2) | d_1, d_2 \in \mathbf{T},\ d_1.attribute \geq d_2.attribute\} \tag{10}$$

とおく．$R_{attribute}$ が全順序であることは容易に確かめられる．したがって，σ_r の意味論は，全順序 $R_{attribute}$，任意の $X \subset \mathbf{T}$ に対し，ある $a \in \mathbf{T}$ について

$S(\sigma_r)(X) \triangleq \{x \in \mathbf{T} | x < a\} \cap X$

$$\begin{cases} |f(X)| = n \ (X \text{ が無限集合，あるいは} \\ \qquad X \text{ が有限集合であり } |X| \geq n \text{ のとき}) \\ f(X) = X \ (X \text{ が有限集合であり } |X| < n \text{ のとき}) \end{cases}$$

と定義する．これはランキング関数であるため，定理 1より，逐次等価性 SE を満たす．

3.2 手続的意味論

前節までは，宣言的意味論について述べたが，実際のフィルタリング処理では，逐次処理を行うのが一般的かつ現実的である．しかし，フィルタリング SQL 記述の要求を逐次処理で実行する，すなわち手続的意味論で解釈できるようにするには，宣言的意味論と手続的意味論が等しくなければならない．両者が等しくなるためには，

逐次等価性 $f(S \cup T) = f(S \cup f(T))$

を満たす必要がある．3.1節に示したランキングの記述 σ では，補題 3より，任意の S, T に対して $S \subset T$ のとき $X_T \subset X_S$ であるという条件を満たすならば，γ 適合であるフィルタリング関数 f が逐次等価性 SE を満たすことが示された．また，(9)，(10) の全順序を用いて意味付けされたランキング記述 σ_γ では，定理 1より必ず逐次等価性 SE を満たすことが示された．したがって，これらのフィルタリングでは，手続的意味論と宣言的意味論が一致する．

4. 複数の構文を組合せた記述

実際にユーザがフィルタリングの要求を記述する場合，複数のポリシーを組合せて記述するのが一般的である．そこで本章では，複数の構文を組合せた記述の意味を定義する．

まず，WHERE 句に複数の構文 σ_1，σ_2 を AND 演算子でつなげた記述 σ を以下に示す．

```
EXTRACT
WHERE      σ1
  AND      σ2
```

このような記述 σ の意味論を，任意の $X \subset \mathbf{T}$ に対して

$$S(\sigma)(X) \triangleq S(\sigma_1) \circ S(\sigma_2)(X) \tag{11}$$

あるいは

$$S(\sigma)(X) \triangleq S(\sigma_2) \circ S(\sigma_1)(X) \tag{12}$$

と定義する．ここで，$S(\sigma_1)$ と $S(\sigma_2)$ の実行順序は，それぞれが満たす性質に応じて変更できる場合がある．フィルタリング手法の実行順序を入れ換えた結果に包含関係があるかどうかを表 2 に示した．表 2 より，$S(\sigma_1)$ と $S(\sigma_2)$ が可換であるならば，(11) と (12) が一致するため，$S(\sigma_1)$ と $S(\sigma_2)$ の実行順序を自由に交換しても一貫したフィルタリング結果が得られる．しかし，$S(\sigma_1)$ と $S(\sigma_2)$ が可換でないとき，(11) と (12) が一致しないため，あらかじめ実行順序を決定しておく必要がある．

フィルタリング手法の実行順序を決定する際，フィルタリング手法の性質や環境を考慮することで，より効率的なフィルタリングを実現できる．例えば，簡単な論理演算を行うフィルタリングと複雑な論理演算を行うフィルタリングを組合せる場合，簡単なフィルタリングを先に行い，後の複雑な計算に適用するデータを減らすことで，フィルタリング全体の計算コストを軽減できる．また，蓄積条件にマッチするデータがより少ないフィルタリングを先に行い，なるべく初期の段階に必要なデータを絞り込むことで，2 回処理するデータを減らすことができる．一般に，適用するデータが多いほどその処理コストは高くなるため，受信データの内容や構造に応じて実行順序を変更することは有効である．

一方，以下のような記述 σ に対して，

```
EXTRACT
FROM       ( EXTRACT
             FROM
             WHERE     σ2)
WHERE      σ1
```

σ の意味論は，

$$S(\sigma)(X) \triangleq S(\sigma_1) \circ S(\sigma_2)(X) \tag{13}$$

と定義する．このように，記述が入れ子の形になっているものについては，$S(\sigma_1)$ と $S(\sigma_2)$ の実行順序を一意に定義する．

また，以下の記述は，複数のフィルタリング SQL 文を演算子で結び付けたものである．

```
EXTRACT
WHERE      σ1
 UNION
EXTRACT
WHERE      σ2
```

上のように，フィルタリング SQL 文を UNION 演算子で結び付けた記述の意味を

$$S(\sigma)(X) \triangleq S(\sigma_1) \vee S(\sigma_2)(X) \tag{14}$$

と定義する．同様に，

```
EXTRACT
WHERE      σ1
 INTERSECT
EXTRACT
WHERE      σ2
```

のように，INTERSECT 演算子を用いた記述の意味を

$$S(\sigma)(X) \triangleq S(\sigma_1) \wedge S(\sigma_2)(X) \tag{15}$$

とする．ここで，

$$S(\sigma_1) \vee S(\sigma_2)(X) = S(\sigma_2) \vee S(\sigma_1)(X) \tag{16}$$

$$S(\sigma_1) \wedge S(\sigma_2)(X) = S(\sigma_2) \wedge S(\sigma_1)(X) \tag{17}$$

が成立することは容易に確かめられる．

```
EXTRACT   *
FROM      A_Broadcast
WHERE     best(1000, Broadcast_Time, DESC)
  AND     best(50)
```

図 2　ユーザ要求の記述例 1

```
EXTRACT   *
FROM      A_Broadcast
WHERE     best(1000, Broadcast_Time, DESC)
  AND     PERIOD 1hour
```

図 3　ユーザ要求の記述例 2

5. 記　述　例

本章では，ランキングの構文と他の構文を組合せた記述例を示し，前節までに定義した意味論から，それらを実行するフィルタリングの特性について考察する．

図 2 の記述例は,「A_Broadcast により放送されたデータの中から，最近放送された 1000 個のデータで，ユーザに対する重要度の高さが 50 位以内であるデータが欲しい」というユーザの要求を表す．この記述の意味は，(10) で定義した全順序 $R_{attribute}$ に対するランキング関数 f と γ 適合であるフィルタリング関数 g との合成関数で表現される．f は逐次等価性を満たす．また，補題 3より，任意の S, T に対して $S \subset T$ のとき $X_T \subset X_S$ という条件を g が満たすならば，g は逐次等価性を満たす．したがって，g がこのような条件を満たすとき，図 2 の記述を実行するフィルタリングは，表 1 より逐次等価性を必ずしも満たさない．すなわち，宣言的意味論と手続的意味論が一致しない．また，表 2 より，処理の途中で実行順序を交換すると，一貫したフィルタリング結果が保証できなくなる．ゆえに，フィルタリング処理の実行前に，最適な実行順序を決定しておく必要がある．

図 3 の記述例は,「A_Broadcast により放送されたデータの中から，最近放送された 1000 個のデータで，1 時間以内に放送されたデータが欲しい」というユーザの要求を表す．“PERIOD” を用いた記述の意味論は，分配等価性を満たすことが既に明らかになっている[14]．したがって，図 3 の記述例の意味は，逐次等価性を満たすフィルタリング関数 f と分配等価性を満たすフィルタリング関数 g の合成関数で表現される．表 1 より，フィルタリングの実行順序を $f \circ g$ とすれば逐次等価性を満たすため，宣言的意味論と手続的意味論が一致する．逆に，フィルタリングの実行順序を $g \circ f$ とすると，逐次等価性を必ずしも満たさないため，宣言的意味論と手続的意味論は一致しない．また，表 2 より，2 つのフィルタリングは可換でないため，あらかじめ実行順序を決定しておく必要がある．

```
EXTRACT   *
FROM      A_Broadcast
WHERE     best(100)
 UNION
EXTRACT   *
FROM      A_Broadcast
WHERE     best(100, Broadcast_Time, DESC)
```

図 4　ユーザ要求の記述例 3

```
EXTRACT   *
FROM      A_Broadcast
WHERE     GENRE = Econimic News
  AND     PERIOD 1day
 INTERSECT
EXTRACT   *
FROM      A_Broadcast
WHERE     best(100, size, ASC)
```

図 5　ユーザ要求の記述例 4

次に，**図 4** の記述例は,「A_Broadcast により放送されたデータの中から，ユーザに対する重要度の高さが 100 位以内であるデータと，最近放送された 100 個のデータが欲しい」というユーザの要求を表す．このような記述の意味は，γ 適合であるフィルタリング関数 f と (10) で定義した全順序 $R_{attribute}$ に対するランキング関数 g の和フィルタリング関数で表される．したがって，f が任意の S, T に対して $S \subset T$ のとき $X_T \subset X_S$ という条件を満たすならば，表 3 より逐次等価性を満たすため，宣言的意味論と手続的意味論が一致する．

さらに，フィルタリング関数の合成と積によって意味が表現される記述例を**図 5** に示す．図 5 の記述は,「A_Broadcast により放送されたデータの中から，ここ 1 日以内の経済ニュースに関するもので，かつデータサイズの小ささが 100 位以内であるデータが欲しい」というユーザの要求を表す．INTERSECT 演算子よりも前の 2 構文の意味は，文献 14) より両者とも分配等価性を満たすフィルタリング関数で表現されるため，2 構文を組合せた記述の意味は，表 1 より分配等価性を満たす．一方，INTERSECT 演算子よりも後の構文の意味は，(9) により定義した全順序 $R_{attribute}$ に対するランキング関数で表される．したがって，この記述全体の意味は，分配等価性を満たすフィルタリング関数と逐次等価性を満たすフィルタリング関数の積フィル

タリング関数で表現できるため，表 4 より，宣言的意味論と手続的意味論が一致しない．

上記の性質から，フィルタリング SQL 記述の意味が逐次等価性を満たす場合，一括処理と逐次処理，並列処理のフィルタリング結果が等価となることが保証される．したがって，フィルタリング端末の処理能力が低い場合や，ネットワークの帯域が十分でない場合，複数の端末で並列処理することで負荷を分散できる．また，常に最新のフィルタリング結果が必要な場合は逐次処理を選択し，端末の処理能力が不足する場合は一括処理を選択するといった処理の等価変換が可能である．

6. おわりに

本稿では，フィルタリング SQL 構文の中で特に重要な構文であるランキングの構文の意味を明確にした．また，複数の構文を組合せて記述した場合の意味も定義した．構文の意味をフィルタリング関数で表現したことにより，ユーザの要求を実行するフィルタリングの性質を明らかにした．さらに，その性質から，環境に応じた効率的な処理方法が実現できることを述べた．

今後の課題として，フィルタリング SQL の拡張とその意味付けが挙げられる．実際のフィルタリングポリシーには，プリファレンスやランキング以外にも,「平日は仕事に関するデータ，休日は趣味に関するデータが欲しい」といったように時間の推移を考慮した要求や,「音楽データを蓄積するときは，必ずその作曲者の情報も一緒に蓄積したい」といったようにデータ間の相関性を考慮した要求が存在する．このようなポリシーを表現するために，フィルタリング SQL の構文を拡張し，本稿と同様にその意味を明確にする．また，それらの構文と，他の構文をさらに複雑に組合せたときの意味も明らかにする予定である．

また，文献 7) では，ユーザの嗜好を柔軟に表現するため，プリファレンス記述に “PREFER A TO B WITH Intensity”(B よりも A が非常に重要)，“PREFER A TO B WITH Littleness”(B よりも A が少しだけ重要) と表記することで，嗜好の度合いを表現できる構文を定義した．そこで，そのような構文の意味も明確にする必要がある．

謝　　辞

本研究は，文部科学省振興調整費「情報フィルタリングの数学的基盤の確立」,「モバイル環境向 P2P 型情報共有基盤の確立」，および文部科学省 21 世紀 COE プログラム「ネットワーク共生環境を築く情報技術の創出」，科学研究費補助金 (基盤研究 (B)(2))「大規模な仮想空間システムを構築する放送型サイバースペースに関する研究」(プロジェクト番号:15300033) の研究助成によるものである．ここに記して謝意を表す．

参 考 文 献

1) N. J. Belkin and W. B. Croft: “Information filtering and information retrieval: two sides of the same coin?,” *Communications of the ACM*, vol. 35, no. 12, pp. 29–38 (1992).
2) T. A. H. Bell and A. Moffat: “The design of a high performance information filtering system,” in *Proc. 19th International Conference on Research and Development in Information Retrieval (SIGIR1996)*, pp. 12–20 (1996).
3) 森田昌宏: “情報フィルタリングに関する研究動向,” JAIST Research Report, IS-RR-93-9I, 北陸先端科学技術大学院大学情報科学研究科 (1993).
4) 西正, 野村敦子: “多チャンネル放送の衝撃,” 中央経済社 (1997).
5) Satellite Magazine: http://www.satemaga.co.jp.
6) 澤井里枝: ミニサーベイ “ブロードバンド時代における情報フィルタリングの動向,” 電子情報通信学会第 13 回データ工学ワークショップ (DEWS2002) 論文集 (2002).
7) 澤井里枝, 寺田努, 塚本昌彦, 西尾章治郎: “フィルタリング SQL: フィルタリングのためのユーザ要求記述言語,” 電子情報通信学会第 11 回データ工学ワークショップ (DEWS2000) 論文集 (2000).
8) R. Sawai, M. Tsukamoto, Y. H. Loh, T. Terada, and S. Nishio: “Functional properties of information filtering,” in *Proc. 27th International Conference on Very Large Data Bases (VLDB2001)*, pp. 511–520 (2001).
9) 澤井里枝, 塚本昌彦, 寺田努, Loh Yin Huei, 西尾章治郎: “情報フィルタリングの関数的性質について,” 電子情報通信学会論文誌 D-I, vol. J85-D-I, no. 10, pp. 939–950 (2002).
10) 澤井里枝, 塚本昌彦, 寺田努, 西尾章治郎: “フィルタリング関数におけるセレクションとランキングについて,” 情報処理学会論文誌:データベース, vol.43, no. SIG12(TOD16), pp. 80–91 (2002).
11) 澤井里枝, 塚本昌彦, 寺田努, 西尾章治郎: “合成フィルタリング関数の性質について,” 情報処理学会論文誌:データベース, vol. 44, no. SIG3(TOD17), pp. 43–53 (2003).
12) 澤井里枝, 塚本昌彦, 寺田努, 西尾章治郎: “情報フィルタリングの実行順序に関する関数的性質について,” 情報処理学会論文誌:データベース, vol. 44, no. SIG3(TOD17), pp. 54–64 (2003).
13) 澤井里枝, 塚本昌彦, 寺田努, 西尾章治郎: “フィルタリング関数の和積について”, 情報処理学会論文誌:データベース, vol. 44, no. SIG12 (TOD19), pp. 86–97 (2003).
14) 澤井里枝, 塚本昌彦, 寺田努, 西尾章治郎: “情報フィルタリングのためのユーザ要求記述言語 FilteringSQL について”, 情報処理学会研究報告 (データベースシステム研究会 2003-DBS-131(I)), vol. 2003, no. 71, pp. 451–458 (2003).