Web Surfing Algebra における関数および演算子の性質について

On Properties of Functions and Operators in Web Surfing Algebra

片山 聡一郎♠ 遠山 元道♦
Soichiro KATAYAMA Motomichi TOYAMA

Web から自分が欲しい情報を取り出すために,いくつもの Web ページをネットサーフィンする様な複雑な検索が必要な場合もある.この様な検索におけるユーザの負担を軽減するためにユーザがインターネットを利用した情報探索行動を定性的に表現する数学的基盤として,Web Surfing Algebra(WSA)を提案する.また,ユーザに WSA 式をわかりやすく提示するために WSA ナビゲーショングラフを提案する.本論文では Web Surfing Algebra として定義した関数および演算子に関して,等価性などの観点からその性質を明らかにする.さらに検索の効率化や共同作業についての考察を行う.

In Web, many users have to do a complicated search that is to do netsurf of many Web pages for gainning information which I want. We have proposed Web Surfing Algebra(WSA) which is mathematical representation of action of collecting information with internet,so that users can do netsurf easily. We have also proposed WSA navigation graph which show a WSA expression intelligibly. We reveal the characteristics of functions and operators in WSA in this paper. In addition, We examined promotion and coaction of search.

1. はじめに

Webは巨大かつ多様性に富んだ情報メディアであり，自分が欲しい情報を取り出すためにgoogle[1]，Yahoo![2]などWeb検索エンジンが用いられている．またユーザの検索要求が多様化複雑化するのに応じて検索結果の質を向上させるための研究も盛んに行われている[3]，[4]．

Web検索においてユーザは検索クエリを通じて自分の情報要求を検索エンジンに伝えている．例えば「沖縄の駅弁について調べたい」とユーザが思った場合，検索エンジンに「沖縄 AND 駅弁」というクエリを与える．多くの場合，情報要求に含まれている単語を組み合わせる事によって作られる検索クエリを1回検索エンジンに与えれば自分が知りたい情報を得る事ができる．

しかしながらユーザの要求は様々であり，例えば「宜野湾の名物を体験できる新宿の店を調べたい」といった事をユーザが思った場合，この情報要求を従来の検索エンジンで満たすのは簡単ではない．

この場合ユーザの情報要求に含まれる単語を組み合わせて「宜野湾 AND 名物 AND 新宿」といった検索クエリを検索エンジンに与えても，最終的に知りたい新宿のお店の情報は検索結果からは得にくい．この原因は，調べたい店を紹介しているWebページの文章中に「宜野湾」や「名物」といった単語が含まれているとは限らないからである．

従来の検索エンジンを用いてこの情報要求を満たすためには，まず「宜野湾 AND 名物」で検索を行い一旦宜野湾の名物を調べ，その後改めてその名物を扱っている新宿の店について検索を行うといった必要があり，ユーザが行わなければならない検索行動は先程の情報要求の例より多い．

1回検索クエリを検索エンジンに与えるだけでは知りたい情報を得る事ができず，複数のWebページの情報を必要とする検索においてユーザの負担を軽減し，検索を効率よく行う事ができる様にする事はWeb検索の利便性の向上につながる．この様な検索を支援するシステムを構築するためには，検索の手順及び検索結果の履歴を管理し扱う事ができる必要がある．そこでユーザのネットサーフィン行為に対応した代数Web Surfing Algebra (WSA)を提案する．WSAを利用する事により，ユーザのWebにおける情報探索行為を代数として扱う事が可能となる。そして，複雑な検索に対する検索支援システムについて議論でき，その様にして構築された検索支援システムにより，情報を得るための手順を案内したり，ユーザの検索負担を減らしたり，検索を効率よく行ったり，検索を共同作業で行ったりする事ができると考えられる．

2. Web Surfing Algebra(WSA)

2.1 Web Surfing Algebra の定義

Web Surfing Algebra(WSA)とはユーザのネットサーフィン行為に対応した代数である．ユーザのネットサーフィン行為及び，ネットサーフィン中システムで行われる処理について1つ1つを以下に述べる関数，演算子に対応させた代数である．WSAの対象はユーザのWeb上の情報・データに対する情報探索行動である．

2.2 Web Surfing Algebra における用語の定義

WSAにおいて扱われる用語を以下に紹介，説明する．

・word：単語
名詞，動詞，形容詞など，意味を持った文字列．

・words：単語集合
0個以上の単語を要素とする順序つきの集合と定める．

・query：クエリ
検索エンジンにおける検索クエリを指すものとする．単語,"　","AND","OR","NOT","(",")"からなる文字列．

・queries：クエリ集合
0個以上のqueryを要素とする順序つき集合と定める．

・result：検索結果
検索エンジンに対して検索クエリを与える事によって得られる検索結果．Webページのタイトル，URL，ページの要約からなる組と定める．

・results：検索結果集合
0個以上のresultsを要素とする順序つき集合と定める．

・URLs：URL集合
0個以上のURLを要素とする順序つき集合と定める．

・attribute：属性
名詞，形容詞など特徴を示す文字列．

・WSA式
WSA式はユーザのネットサーフィンにおける情報に対する

♠ 学生会員　慶應義塾大学大学院理工学研究科修士課程
katasou@db.ics.keio.ac.jp
♦ 正会員　慶應義塾大学理工学部情報工学科
toyama@ics.keio.ac.jp

要求及びそのためのプロセスを以下に述べるWSAの関数，演算子によって対応付けしたものである．WSA式の処理は括弧の一番内側にある関数，演算子の処理から行っていき，一番外側の関数，演算子が処理し終わった段階でユーザの目標が達せられた状態となる．

なお式を処理していく途中で，出力が0件であるなど十分な情報が得られない場合には適宜式の処理を遡る事となる．

2.3 WSAで使用される関数及び演算子

WSAで使用される関数及び演算子は情報探索行動における行為及びその目的と対応付けされている． WSAで使用される関数及び演算子を紹介する．

なお演算子と関数の違いはユーザが選択する行為が含まれているものを演算子，含まれていないものを関数と定める．

2.3.1 ユーザ及びシステムのWeb検索行為に対応した関数，演算子

・Q(queries)：Q関数

queryを検索エンジンで検索し，検索結果の集合を得る．入力が複数のqueryからなる場合1つずつ処理していく．

入力･･･検索クエリ(queries)

出力･･･検索結果集合(results)

・σ(results)：σ演算子

入力された検索結果からユーザは0個以上のURLを選択する．

入力･･･検索結果集合(results)

出力･･･URL集合(URLs)

・$\pi_{\{attribute\}}$(URLs)：π演算子

入力されたURLのWebページを開きユーザに提示する．ユーザは指定された属性を持つ単語をピックアップする．

入力･･･URLの集合(URLs)

出力･･･指定された属性(attribute)を持つとユーザが選択した単語集合(words)

2.3.2 集合に対して操作する関数，演算子

・dis（queries）：disjunction

入力されたクエリ（単語含む）集合間を"OR"で結合する．

入力･･･クエリ集合

出力･･･入力された集合の要素を"OR"でつないだ結果

・con（queries）：conjunction

入力されたクエリの集合間を"AND"で結合する．

入力･･･単語集合またはクエリ集合

出力･･･入力された集合の要素を"AND"でつないだ結果

・words ∪ words：union（和集合）

２つの単語集合の和集合をとり，新たな単語集合を得る．

入力･･･２つの単語集合

出力･･･２つの単語集合の和集合をとり，その中から重複するものを取り除いた結果得られる単語集合

・words ∩ words：intersection（積集合）

２つの単語集合の積集合をとり，新たな単語集合を得る．

入力･･･２つの単語集合

出力･･･２つの単語集合の積集合をとる事により得られる単語集合

・words ⊔ words：選択union

ユーザは２つの単語集合を見て，それらの中から出力に含めたくない要素を選択．それらを除き残った要素の和集合をとり新たな単語集合を得る．

入力･･･２つの単語集合

出力･･･２つの単語集合の単語の中からユーザが選択された要素を取り除き，残った単語の和集合をとり，その中から重複する要素を取り除いた結果得られる単語集合

・words ⊓ words：選択intersection

ユーザが２つの単語集合を見比べ，それらの中から全く同じまたはほぼ同じ要素を選択．それらを要素とする新たな単語集合を得る．

入力･･･２つの単語集合

出力･･･ユーザが選択した単語集合

・ asc（words）：asc関数

入力された単語の集合を昇順に並び替える．

入力･･･単語の集合(words)

出力･･･入力された単語の集合を昇順に並び替える事によって得られる単語の集合

・ dec（words）：dec関数

入力された単語の集合を降順に並び替える．

入力･･･単語の集合(words)

出力･･･入力された単語の集合を降順に並び替える事によって得られる単語の集合

検索において通常ユーザは結果の上位数件しか利用しない．そこで集合の全要素の上位n件のみを出力とする場合，関数，演算子に添え字nを付ける事によって表現する．但し出力された集合の要素数がn件に満たない場合は全てとする．

2.4 WSA式の例

「宜野湾の名物を調べる」という場合，WSA式で表すとその一例として以下の様に表せる．

$\pi_{\{名物\}}$（σ(Q(宜野湾 AND 名物))）

この式を処理の流れは，始めにQ関数を処理し「宜野湾 AND 名物」を検索エンジンにかける．次にσ演算子を処理し，その検索結果をユーザに提示し，URLを選択してもらう．最後にπ演算子を処理し，選択されたURLのページを開き，ユーザに名物を探し出してもらう．

3. WSAナビゲーショングラフ

情報検索支援システムにおいてWSA式をそのままユーザに提示してもユーザは理解するのが困難である．

そこでWSA式をグラフでユーザに提示する手法の1つとしてWSAナビゲーショングラフ(以下ナビグラフ)を提案する．

ナビグラフはWSA式から生成される．先程の検索例「宜野湾の名物を調べる」に対応するWSA式をナビグラフで表すと図1の様になる．

4. WSA の関数および演算子の性質

4.1 性質を表現するための演算子の定義

集合でないものに対しては全く同一のものについては=と記す．query において OR や AND などにより結合の順が違うものに関しては～で表す事にする．

ある集合Sに対して，その要素の数を｜S｜と表記する．順序つき集合S_1,S_2に対して，その要素が全く等しい時にはS_1=S_2で表す．さらに，それらの要素の順番も全く等しい時には$S_1 \equiv S_2$で表す．また，S_2の要素がS_1と全く一致する，またはS_1の要素の部分集合であるとき，集合S_2は集合S_1の部分集合であると言い，$S_1 \supset S_2$と表す．また集合S_1, S_2で共通する要素の順番を比較してみた時に，ねじれの関係が無い時には$S_1 // S_2$と表す．$S_1 \supset S_2$かつ$S_1 // S_2$の時$S_1 \supseteq S_2$と表す．

Web 上のデータは常に変化し続けているが，本論文では前提として同一のクエリに対しては同一の検索結果が返って

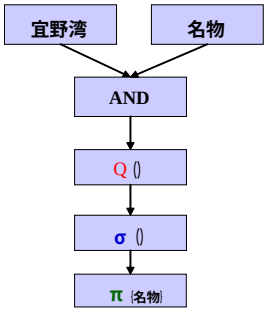

図 1　WSA ナビゲーショングラフ

Fig. 1　WSA navigation graph

くるものとする.

4.2　関数，演算子の性質

・ Q(query)

query として q1 が存在する時，以下の式が成り立つ.

Q(q1 AND q1)≡Q(q1 OR q1)≡Q(q1)

Q_n (q1 AND q1)≡Q_n (q1 OR q1)≡Q_n (q1)

query として q1, q2(q1≠q2)が存在するならば以下の式が成り立つ.

Q(q1 AND q2)⊂Q(q1)⊂Q(q1 OR q2)＝Q(q1)∪Q(q2)

Q(q1)⊇Q_n (q1)

Q(q1 AND q2)⊇Q_n (q1 AND q2)

Q(q1 OR q2)⊇Q_n (q1 OR q2)

が成り立つ．なお

Q(query1 OR query2)と Q(query1 AND query2)

の関係に関しては検索エンジンに依存する.

・σ (results)

この演算子では入力された検索結果集合からユーザが URL を選択をするが，その基準が常に一定かどうかで議論が異なる.

基準が常に一定の場合，検索結果集合R_1，R_2（但し$R_1 \supseteq R_2$）が存在するならば

$\sigma(R_1) \supseteq \sigma(R_2)$

基準が常に一定でない場合にはある URL が選ばれるか，選ばれないかが定まらないため上記の式は成り立たなくなる.

・$\pi_{\{attribute\}}$ (URLs)

この演算子では入力された URL の Web ページから該当する単語をユーザが選ぶため，ユーザが単語を選択する基準が一定の場合とそうでない場合とで分けて議論する必要がある.

・$\pi_{\{attribute\}}(\sigma(URLs))$

σ 演算子，π 演算子に関して基準が常に一定の場合について考える．検索結果集合R_1，R_2（但し$R_1 \supseteq R_2$）が存在するならば以下の式が成り立つ.

$\pi_{\{attribute\}}(\sigma(R_1)) \supseteq \pi_{\{attribute\}}(\sigma(R_2))$

・ dis(queries)/con(queries)

これらの関数では入力された単語集合の要素を集合の要素の順番に従って順番に 1 つずつ要素を OR/AND でつないでいくと考える.

クエリ q1, q2 が存在するならば以下の式が成り立つ.

dis({q1, q2})～dis({q2, q1})

con({q1, q2})～con({q2, q1})

・ ∪/∩

単語集合W_1, W_2が存在する時$W_1 \cup W_2$と$W_2 \cup W_1$, $W_1 \cap W_2$と　$W_2 \cap W_1$では，集合の要素は同じだが順序は一致しないので以下の式が成り立つ.

$W_1 \cup W_2 = W_2 \cup W_1$, $W_1 \cap W_2 = W_2 \cap W_1$

・ ⊔/⊓

ユーザの基準が一定の場合単語の集合W_1, W_2が存在する時∪及び∩の時と同様に以下の式が成り立つ.

$W_1 \sqcup W_2 = W_2 \sqcup W_1$, $W_1 \sqcap W_2 = W_2 \sqcap W_1$

ユーザの基準が一定でない場合には，上記は成り立たない.

・ asc(words)/dec(words)

単語の集合 W が存在する時，次の式が成り立つ.

asc(W)≡asc(asc(W))，dec(W)≡dec(dec(W))

asc(W)≡asc(dec(W))，dec(W)≡dec(asc(W))

また単語 word1, word2 が存在するならば

asc({word1, word2})≡asc({word2, word1})

dec({word1, word2})≡dec({word2, word1})

・ Q(query)と OR と∪の関係

検索クエリとして q1，q2(q1≠q2)が存在するならば

Q(q1 OR q2)=Q(q1) ∪ Q(q2)

|Q(q1 OR q2)|=|Q(q1) ∪ Q(q2)|

であり，Q 関数の処理回数は∪の場合の方が多い.

また nQ()について OR と∪との違いを見てみると

|Q_n (q1 OR q2)|≤|Q_n (q1) ∪ Q_n (q2)|

また，関数，演算子の処理回数も∪の場合の方が多い.

それぞれの検索結果について見てみるとQ_n (q1 OR q2)は q1, q2 のどちらかの検索結果が上位n件のうちのほとんどを占めてしまい，もう一方の結果は上位にはほとんど反映されない可能性がある．一方，Q_n (q1) ∪ Q_n (q2)の検索結果は 2n個以内にQ(q1), Q(q2)のそれぞれ上位n個の結果を含む事になる.

4.3　式を遡る際の各関数，演算子における処理

次に何らかの理由により WSA 式の処理の途中で遡る際，各関数，演算子でどの様な処理方法があるか述べていく.

処理方法については次の 4 種類が考えられる.

(1)1 つ前の関数，演算子からの入力で処理していないものに対して処理を行う.

(2)1 つ前の関数，演算子に戻る.

(3)改めてユーザに処理しなおしてもらう.

(4)AND による条件を緩める.

関数，演算子により行う事のできる処理の種類が異なる．表にまとめると表 1 の様になる．なお△は Q()は関数であるが検索クエリをユーザに入力しなおしてもらう例外処理である.

なお，n が定められている場合には n の値を大きくする事も 1 つの処理方法となる.

4.4　追加された入力に対する各関数，演算子の処理

・Q(query)

検索クエリとして q1，q2(q1≠q2)が存在する時，式の同じ部分で Q(q1)，Q(q2)が逐次的に処理された場合，その出力を入力とする関数，演算子においてそれらを処理する順番は，基本的な方針としては次の 3 種類が考えられる.

1．Q(q1)の結果，Q(q2)の結果の順に処理する.

2．Q(q2)の結果，Q(q1)の結果の順に処理する.

3．Q(q1)の結果，Q(q2)の結果を交互に処理する.

・σ (results)

検索結果集合をR_1，R_2とする.

始めにσ(R_1)の処理を行ない，その後R_2が入力された際，

表 1　式を遡る際の各関数，演算子における処理
Table 1　Process of functions and operators in Track Back

関数演算子	(1)	(2)	(3)	(4)
Q(query)	○	○	△	
σ(results)	○	○	○	
$\pi_{\{attribute\}}$ (URLs)	○	○	○	
dis(queries)	○	○		
con(queries)	○	○		○
∪	○	○		
∩	○	○		
⊔	○	○	○	
⊓	○	○	○	

ユーザが選び出す対象としては次の 2 種類が考えられる．

１．R_2のみ

２．R_2とR_1のうち σ(R_1)を処理した際選ばれなかったもの

この出力を次の関数，演算子においてどの様な順で処理するかは Q 関数で述べた様に 3 種類考えられる．

π 演算子においても σ 演算子と同様の事が考えられる．

・dis(queries)/con(queries)/ ∪/∩/⊔/⊓

これらに関しては逐次的に入力された場合には，これまでの結果と合わせて処理した結果を出力とする．

・asc(words)/dec(words)/arr(words)

これらに関しては逐次的に入力された場合には，新たに入力された単語の集合を既に並び替えられている単語集合に加え改めて並び替える事となる．

5. 考察

WSA の利用の仕方としてはまず検索履歴の保存のための利用が考えられる．検索履歴を WSA 式によって保存でき，それを利用して例えば検索行為の類型化が考えられる．

また検索行為を定式化する事により検索の最適化や共同作業の可能性について議論する事ができる．以下その 2 点について述べる．

5.1　検索の最適化

WSA の観点から検索の最適化について考えると以下の点が挙げられる．

・関数，演算子の結合の仕方，順番について

関数，演算子の性質において等価性について述べてきたが，式の結合の仕方を変える事や式を処理する順番を変える事が可能である．

・Q_n ()などにおけるnの設定

最適な n の値を各関数，演算子毎に定める事が検索の最適化につながる．

・演算子の処理回数

ユーザに対して応答を求める回数を必要最小限とする事が最適化につながる．

・WSA 式を遡る際の対応

WSA 式を処理していく過程で，遡る回数はできるだけ少なく，また遡る際は，最適な方法を選択する事が検索の最適化につながる．

・追加された入力に対する処理

追加された入力に対する各関数，演算子での処理には，いくつかの方法が考えられる．よって，その中から最適な方法を選択する事が検索の最適化につながる．

5.2　検索の共同作業

複数人による並列処理を考える場合に以下に述べる 2 種類の並列処理を考える必要がある．

・パイプライン並列性

各作業者が別々に式中の別の関数，演算子を並列処理していく事を考える．WSA では定義されている関数，演算子は全て入力，出力の型が定められているため，WSA を利用してパイプライン並列処理を行う事は可能だと考えられる．問題点はユーザの選択が必要となる演算子において基準が一定でなくなる可能性がある点である．

・パラレル並列性

複数人が同一の関数を共同作業で処理していく事を考える．この際パイプライン並列性での問題に加えて問題となるのが，各人にどの様に作業を分配するか，各人の結果をどの様に統合するかである．作業の分配については，入力が複数あるのであれば入力を分配する事により，それぞれ別々に作業をする事が可能である．但しデータの同期は必要である．

6. まとめと今後の課題

本研究ではユーザのネットサーフィン行為に対応した代数 Web Surfing Algebra を提案し，使用される関数および演算子を定義した．また，ユーザへの提示手法として WSA ナビゲーショングラフを提案した．

そして各関数および演算子の性質について述べ，検索の最適化，並列性についての考察を行った．この際，ユーザのネットサーフィン行為を WSA により最適化や並列性における問題点をはっきりとさせる事ができた点は，WSA の大きな利点であると言える．

今後の課題としては以下の事が考えられる．

- WSA の関数，演算子の網羅性
- 検索の効率化のための最適な関数，演算子の処理の選択
- WSA に基づく情報検索支援システムの構築

[文献]

[1] L.Page, S.Brin, R.Motwani, T.Winograd: “The PageRank Citation Ranking”, Stanford Digital Library Technologies, Working Paper SIDL-WP1999, pp.414-425 (1999).

[2] Yahoo! home page, http://www.yahoo.com

[3] Natalie S. Glance: “Community Search Assistant”, Proceedings of the 6th international conference on Intelligent user interfaces, pp. 91-96 (2001).

[4] Mark Levene and Richard Wheeldon: “Navigating the World-Wide-Web”, Web Dynamics.Springer-Verlag, (2003).

片山 聡一郎　Soichiro KATAYAMA

慶應義塾大学大学院理工学研究科開放環境科学専攻修士課程在学中．2004 慶應義塾大学理工学部情報工学科卒業．データベースの研究に従事．情報処理学会学生会員．日本データベース学会学生会員．

遠山 元道　Motomichi TOYAMA

慶應義塾大学理工学部情報工学科専任講師．博士（工学）．1984 慶應義塾大学大学院博士課程単位取得退学．主にデータベースシステムの研究に従事．IEEE Computer Society, ACM，情報処理学会，日本ソフトウェア科学会，電子情報通信学会，日本データベース学会会員．