1N-4

正規表現関数による文字列照合問題と照合位置に関する考察

石田 俊一† 大塚 寛‡
愛媛大学大学院理工学研究科† 愛媛大学理学部‡

1. 概要

正規表現による文字列照合問題(RPM)では、有限オートマトン(FA)を用いたアルゴリズムがよく知られている。一方、前方参照付き正規表現による RPM では参照元となる部分正規表現に照合する部分テキストを特定出来なければならない。ところが決定性有限オートマトン(DFA)と非決定性有限オートマトン(NFA)では、言語の受理能力に関しては同等であるにも関わらず、部分テキストの特定に関しては NFA のほうが容易であり、DFA では困難という事実がある[3]。

そこで我々は正規表現中の部分正規表現に照合する部分テキストが照合したテキストのどの位置に現れるかと言う問題を考察した。なお前方参照付き正規表現は言語の受理能力に関し、正規表現によるそれを越えているが、それでも参照元に照合する部分テキストの位置を特定することは正規表現の範囲内での問題と考えられる。ここでは FA を用いる替わりに、正規表現によるテキストの左商を基礎とする正規表現関数[5]を用いた RPM の解法を利用し、[2]による照合アルゴリズムの分類/詳細化に沿い、[4]の手法に従って考察した。

2. 正規表現と正規表現関数

2.1. 正規表現

まず以下の議論で必要となる定義を挙げる。

$a,b,c,...\in\Sigma$ を記号の有限集合、$\Sigma^*=\bigcup_{n\geq 0}\Sigma^n$ を有限の長さの記号列の集合とし、ε を空記号列、$x,y,z,...\in\Sigma^*$, $I,J,...\subseteq\Sigma^*$ として用いる。x の y による左商を $x\setminus y$、右商 x/y を次で定義する。

$$x\setminus y=\begin{cases} z & x=yz \\ \perp & otherwise \end{cases},\quad x/y=\begin{cases} z & x=zy \\ \perp & otherwise \end{cases}$$

言語では左商を $I\setminus J=\{y\mid xy\in I, x\in J\}$、右商を $I/J=\{x\mid xy\in I, y\in J\}$ とする。

(前方参照付き)正規表現 r を以下で定義する。なお s,t は正規表現、α は変数である[1]。

$r ::= \phi\mid\varepsilon\mid a\mid s+t\mid s\cdot t\mid s^*\mid s\%\alpha\mid\alpha$

正規表現 r に対する正規言語 $\|r\|$ は通常どおりだが、変数定義に対しては $\|s\%\alpha\|=\|s\|$、変数参照に対しては参照元の正規表現に照合した文字列で、正規言語の範囲では記述できないが、以下では扱わないため、ここには挙げない。

2.2. 正規表現関数

正規表現 r に対し同じ記号で表わした正規表現関数 $r:\wp(\Sigma^*)\to\wp(\Sigma^*)$ を次のように定義する。

$\phi(I)=\phi,\quad \varepsilon(I)=I,\quad a(I)=I\setminus\{a\},$

$(s+t)(I)=s(I)\cup t(I),\quad (s\cdot t)(I)=t(s(I)),$

$(s^*)(I)=\bigcup_{i\geq 0}s^i(I),\quad (s\%\alpha)(I)=s(I)$

正規表現関数は以下の性質を持つ。

$r(I)=I\setminus\|r\|,\qquad x\in\|r\|\Leftrightarrow\varepsilon\in r(\{x\})$

$\|r\|=\{x\mid x\in\Sigma^*,\varepsilon\in r(\{x\})\}$

$H(x)\cap\|r\|\neq\phi\ \Leftrightarrow r(\{x\})\neq\phi$

3. 正規表現による文字列照合問題

RPM とは正規表現 r と入力テキスト $S\in\Sigma^*$ に対し次のような集合 O を求めることである。

$O=\{(x,y,z)\mid S=xyz, y\in\|r\|\}$

以下にこの問題を解く抽象的なアルゴリズムを挙げる。なお表記法は[2,4]に従う。

```
begin x,w = ε,S ;
  if ε∈‖r‖ → O = {(ε,ε,S)}
     ε∉‖r‖ → O = φ        fi;
  do w ≠ ε →
     y,z = ε,w ;
     do z ≠ ε cand y(z↑1) ∈ H(‖r‖) →
        y,z = y(z↑1),(z↓1) ;
        if y∈‖r‖ → O = O∪{(x,y,z)}        fi;
     od
     x,w = x(w↑1),(w↓1) ;
  od
end
```

Regular Expression Pattern Matching Problem and Finding its Matching Positions based on Regular Expression Function
†Graduate School of Science and Engineering, Ehime University
‡Faculty of Science, Ehime University

3.1. 正規表現関数を用いた解法

上の抽象的なアルゴリズムを正規表現関数を使った解法に詳細化する。まず言語 I に対し $r(I)$ は正規表現関数の定義に従い再帰的に計算する。この計算は $I=\{w\}$ から始まり、正規言語で左商をとることから、必ず終了することが保証されることを注意しておく。これを利用して以下のような詳細化が得られる。

```
begin x, w = ε, S;
  if ε∈‖r‖ → O = {(ε, ε, S)}
     ε∉‖r‖ → O = φ      fi;
  do w ≠ ε →
     I = r({w});
     while I ≠ φ →
        z ∈ I, y = w / z, I = I − z;
        O = O ∪ {(x, y, z)};
     end
     x, w = x(w↑1), (w↓1);
  od
end
```

枠で囲まれたコード部分が詳細化を行った部分である。後で述べる FA による詳細化でも、基本的に同じコード部分を詳細化する[2,4]。これらはいずれもテキストの接尾語の各接頭語に対し照合を行う KMP 型(Knuth-Morris-Pratt の照合アルゴリズムに由来)に分類されるアルゴリズムである[2]。

4. 照合位置の特定

RPM により得られた O の各元 (x,y,z) と正規表現 r のある部分表現(前方参照つき正規表現 $r=\cdots s\%\alpha\cdots$ の定義部分 $s\%\alpha$)に対し、s に照合する(α に束縛される) y の部分文字列を特定する。ただし+は容易、*は・に帰着されるので、以下では・、特にその中でも $r=t\cdot s\%\alpha\cdot u$ となる場合を考察する。s に照合する y の部分文字列は、先にあげた正規表現関数の性質と以下の性質

- $\forall w\ x\in\{w\}/r(\{w\})\Rightarrow x\in\|r\|$
- $r(\{y\})=\{z\mid xz=y, x\in\|r\|\}$

より、$(\{y\}\backslash\|t\|)/\|u\|$ あるいは $t(\{y\})/\|u\|$ を求めることである。これを求めるアルゴリズムの概略は以下の通りである。

$r(y)$ を再帰的に計算する中で $(s\%\alpha)(\{w_1,\cdots,w_n\})$ を計算するが、各 $s(w_i)=\{v_1,\cdots v_{m_i}\}$ の各元 v_j に t を適用し、$\varepsilon\in t(v_j)$ であれば、s に照合する文字列として w_i/v_j が取れることになる。この操作をすべての $i\,(1\le i\le n)$, $j(1\le j\le m_i)$ に対して行うことで照合位置が特定できる。

4.1. FA との比較

FA および正規表現関数による照合アルゴリズムでは以下の事実が知られている[1,5]。なお m は正規表現の長さ、n はテキストの長さを表す。

計算量	時間	空間
DFA	$O(2^m+n)$	$O(2^m)$
NFA	$O(m^n)$	$O(m)$
正規表現関数	$O(mn^2)$	$O(n)$

前方参照については、正規表現から導出される標準的な NFA では、その状態と部分表現が対応するので、容易に求まる。他方正規表現から導出される DFA では、一般にその状態と部分表現との対応が取れないので、困難であることが知られている[3]。正規表現関数の場合は、正規表現による微分により[5]

- 正規表現のどの部分を計算しているのか把握できる
- 右商を用いることで、部分表現に照合する文字列を求めることができる

ことで、照合位置の特定が可能になる。

5. 今後の課題

今回のアルゴリズムは KMP 型であったが、[4]では BM 型(Boyer-Moore の照合アルゴリズムに由来)が示されている。今後は正規表現関数を使った BM 型の考察と照合位置の特定、両方の型で異なる性質について調べたい。

参考文献

[1] A. V. Aho: Algorithms for Finding Patterns in Strings, HANDBOOK OF THEORETICAL COMPUTER SCIENCE, Elsevier Science Publishers B.V., pp.257-295 (1990)

[2] B. W. Watson, G. Zwaan: A taxonomy of sublinear multiple keyword pattern matching algorithms, Science of Computer Programming 27 pp.85-118 (1996)

[3] J. E. F. Friedle: Mastering Regular Expressions, O'reilly & Associates (1997), 歌代昭正(訳): 詳説正規表現, オライリー・ジャパン(1999)

[4] B. W. Watson, R. E. Watson: A Boyer-Moore-style algorithm for regular expression pattern matching, Science of Computer Programming 48 pp.99-117 (2003)

[5] 山本篤, 山口和紀: 正規表現関数による正規表現の拡張とそのパターンマッチングへの応用, 情報学会論文誌, Vol.44, No.7, pp.1756-1764 (2003)