A_006

正規表現関数を用いた文字列照合アルゴリズムの高速化に関する研究

石田 俊一 †, 大塚 寛 ‡, 本多 和正 †
九州大学大学院システム情報科学府 † 愛媛大学理学部 ‡

1 概要

正規表現による文字列照合問題では、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。一方で、この問題はキーワードによる文字列照合問題の一種の拡張とみなす方法もある [4]。我々はこの方法に正規表現関数 [5] を適用した手法について研究を行っている。

キーワードによる文字列照合問題を解くアルゴリズムである、Boyer-Moore 法 (BM 法) や Knuth-Morris-Platt 法 (KMP 法) では、照合が失敗した際に次回の照合開始位置を効率的に決定することで高速な照合を実現している。今回の研究ではそのような開始位置を決定する手法を、正規表現関数を用いた手法に導入することで、文字列照合の高速化を考察した。

キーワードによる文字列照合問題を解くアルゴリズムはいくつかの型に分類できるが、ここでは Knuth-Morris-Pratt 型 (KMP 型) について述べる。

2 正規表現と正規表現関数

2.1 記号列、言語

ここでは以下の議論で必要となる定義と性質を挙げる。$\Sigma(\neq \phi)$ を記号の有限集合、$\Sigma^* = \bigcup_{i\geq 0} \Sigma^i$ を有限の長さの記号列全体の集合とし、$\varepsilon \in \Sigma^*$ を空記号列、$a \in \Sigma$、$x, y, z, ... \in \Sigma^*$, $I, J, ... \subseteq \Sigma^*$ として用いる。x の y による左商を $x \setminus y$、右商を x/z を次で定義する。

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

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

正規表現 r、正規言語 $||r||$ を以下で定義する。なお s, t は正規表現、α は変数である。

$r ::= \phi|\varepsilon|a|st|s|t|s^*$

$||\phi|| = \phi$, $||\varepsilon|| = \{\varepsilon\}$,$||a|| = \{a\}$,

$||s|t|| = ||s|| \cup ||t||$,$||st|| = ||s|| \cdot ||t||$, $||s^*|| = ||s||^*$

2.2 正規表現関数

Σ 上の正規表現 r に対し、$[r]$ で表した正規表現関数 $[r] : \wp(\Sigma^*) \to \wp(\Sigma^*)$ を次のように定義する。

$[\mathbf{a}](I) = I \setminus \{a\}$, $[\varepsilon](I) = I$, $[\phi](I) = \phi$,

$[(s|t)](I) = [s](I) \cup [t](I)$, $[(st)](I) = [t]([s](I))$,

$[(s^*)](I) = \cup_{i\geq 0}[s]^i(I)$,

すなわち、正規表現関数 r とは入力された言語 I の各記号列から r と照合する接頭語を除いた記号列の集合を出力する関数である。

r を Σ 上の正規表現関数、$x \in \Sigma^*, I \subseteq \Sigma^*$ とするとき、次の性質が成り立つ。以下で $pre(x)$ は x の接頭語の集合を表す。

$x \in ||r|| \Leftrightarrow \varepsilon \in [r](x)$

$pre(x) \cap ||r|| \neq \phi \Leftrightarrow [r](x) \neq \phi$

3 移動距離について

Knuth-Morris-Platt 法では、照合が失敗した際に次回の照合開始位置を効率的に決定することで高速な照合を実現している。現在の照合開始位置から次回の照合開始位置までの文字数を**移動距離**と呼ぶ。以下でキーワードによる文字列照合問題を解く KMP 法での移動距離の計算を紹介する [1]。

3.1 KMP 法での移動距離の計算

KMP 法では文字列照合問題を解く際、単一キーワード p の先頭から一致した部分記号列 v を用い、次の式を解くことで移動距離 $N(v)$ を求める。

$N(v) = min\{n|1 \leq n, v\Sigma^* \cap \Sigma^n p \neq \phi\}$

KMP 法では各 $v(v \in pre(p))$ に対する移動距離を最初に作り、照合を調べる際に使うことで高速化を図っ

Faster algorithms for Regular Expression Pattern Matching based on Regular Expression Function
†Graduate School of Information Science, Kyushu University
‡Faculty of Science, Ehime University

ている。この移動距離は複数キーワードによる文字列照合問題に対する移動距離に拡張出来る [1]。

3.2 正規表現関数を使った解法への導入

上で紹介した移動距離を、さらに正規表現 r による文字列照合問題に対する移動距離 $N(r)$ へ拡張する。

r' を r の部分表現、$v \in ||r'||$ として、

$N(r') = min\{n | 1 \leq n, ||r'||\Sigma^* \cap \Sigma^n||r|| \neq \phi\}$

しかし、このままでは計算出来ないため、条件を緩めることで以下の移動距離 $N'(r')$ へ変更する。

$(N'(r') \leq N(r'))$

$N'(r')$ を次の順に求める。

各 r' に対し、以下の操作を行う

1 $long(r') = min\{|v| \ |v \in ||r'||\}$

2 $str(r', long(r')) =$
$\{q||q| = X, q \in pre(p), p \in ||r'||\}$

3 $N'(r') = min\{n | 1 \leq n, Q\Sigma^* \cap \Sigma^n||r|| \neq \phi\}$

ここでは上の 3 の計算手順について述べる。

これは $I\Sigma^* \cap ||r|| \neq \phi$ を計算するアルゴリズムと $min\{n | 1 \leq n, Q\Sigma^* \cap \Sigma^n||r|| \neq \phi\}$ を計算し、移動距離の表を作成するアルゴリズムの 2 つからなる。(ここで $shift(x, aw) = (xa, w)$ とする)

Algorithm 3.1 $I\Sigma^* \cap ||r|| \neq \phi$ の計算

$acc([r], I) =$ **begin switch** (r, I)
case $\varepsilon \in I \ \rightarrow \ \{\varepsilon\}$
case $r = \varepsilon \ \rightarrow \ I$
case $r = a \ \rightarrow \ I \setminus \{a\}$
case $r = st \ \rightarrow \ acc(t, (acc(s, I)))$
case $r = s|t \rightarrow \ acc(s, I) \cup acc(t, I)$
case $r = s^* \rightarrow \ T \quad s.t.$
$T = \phi, \ U = I;$
while$(U - T \neq \phi) \ \rightarrow$
$T, U = T \cup U, acc(s, U)$
end

Algorithm 3.2 移動距離 $N'(r')$ の計算

$N'(r')$=**begin**
for each$(r')\{$
$z, D = long(r'), \phi;$
$W = str(r', z);$
foreach$(w \in W)\{$
$x = \varepsilon;$
while$(w \neq \varepsilon)\{$
$(x, w) = shift(x, w);$
if$(acc([r], w) = \phi) \ D = D \cup \{|x|\}$, **break**;$\}$
$\}$
return $min(D)$;
$\}$
end

ここで $N'(r')$ に対し、r' が単一キーワード p のときは、$Q = pre(p)$ より KMP 法で求まる移動距離 $N(v)$ と同じである。

移動距離を利用した照合方法の概要は以下の通りである。

上の手順を用い、照合を調べる前に $(r', N'(r'))$ の表を作成する。そして、照合を調べる際に失敗したとき、失敗するまでの部分表現 r' と対応する $N'(r')$ だけ次の照合先頭位置をずらし照合を調べる。

4 まとめ

これまでの研究で KMP アルゴリズム、BM アルゴリズムで使われる移動距離を正規表現関数を使った解法に導入することが出来た。しかし、BM アルゴリズムでは、照合が失敗した際のテキストの文字を使った移動距離の計算も高速化を図るために使われている。今後の課題としては、この文字に対する移動距離の導入が挙げられる。

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] Jeffrey 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)