A-004

ファクターオラクルの拡張と実験的評価
Extension of Factor Orcle and Experimental Evaluation

大井恒平† 山本博章‡ 藤原洋志‡
Kohei Ohi Hiroaki Yamamoto Hiroshi Fujiwara

1. はじめに

Allauzen 他[1]は、文字列に対し、ファクターオラクルと呼ばれる新たなデータ構造を考案し、文字列パターン照合に応用した。文字列 w に対するファクターオラクルは、少なくとも w のすべての部分文字列を受理するオートマトンであり（部分文字列でない語を受理する場合もある）、状態数が文字列の長さ+1 となる。複数文字列の検索や索引を考えた場合、複数文字列の集合はループのない決定性有限オートマトン（DFA）として表すことができる。したがって、このようなオートマトンにファクターオラクルを拡張することは有益であると思われる。Mohri[4]は、オートマトンのファクターオートマトンとして、ファクターオートマトンを文字列からオートマトンへ拡張している。ファクターオートマトンは DAWG (Directed Acyclic Word Graph)[2,3]としても知られている DFA で、受理する言語は、ちょうど w の部分文字列全体からなる集合となる。

本論文は、ファクターオラクルを構成する新たな構成法を与えることにより、ファクターオラクルをオートマトンに対するファクターオラクルに拡張する。提案法は、非決定性有限オートマトン（NFA）から DFA を構成する部分集合構成法に基づいている。我々は、入力が一つの文字列の場合、提案法で構成するファクターオラクルは Allauzen のファクターオラクルと同型であることを示す。これにより、提案法は Allauzen のファクターオラクルを、オートマトンへ拡張したものとみなすことができる。

2. 準備

Σ をアルファベットとする。任意の文字列 $w, x, y, z \in \Sigma^*$ に対し、$w=xyz$ と表すことができるとき、x を w の接頭辞、z を w の接尾辞、y を w の部分文字列と呼ぶ。$FACT(w)=\{x \mid x$ は w の部分文字列$\}$ と定義する。M を決定性有限オートマトン(DFA)とする。そのとき、L(M)を M によって受理される言語とし、$FACT(M)=\{ x \mid \exists w \in L(M), x$ は w の部分文字列$\}$ と定義する。一般に DFA はラベル付き有向グラフとして見ることができる。対応する有向グラフが閉路のない有向グラフとなるとき、その DFA を非巡回型 DFA と呼ぶ。明らかに、非巡回型 DFA が受理する言語は有限集合となる。逆に、文字列の有限集合は非巡回型 DFA で表すことができる。本論文では非巡回型 DFA を扱うため、これ以降、DFA といった場合、非巡回型を指すものとする。２つの DFA M_1 と M_2 に対し、状態の名前の付け替えだけで M_1 を M_2 に変えることができるとき、M_1 と M_2 は同型であるという。

DFA M に対するファクターオラクル M_{FO} は以下を満たす DFA である。

1. M_{FO} は少なくとも FACT(M)のすべての語を受理する。
2. M_{FO} の状態数は M の状態数以下である。

性質 1 は、M_{FO} は FACT(M)に入らない語を受理することも許している。

3. Allauzen のファクターオラクル

以下に Allauzen のファクターオラクルの構成法を示す。

構成法
入力 文字列 $p=p_1p_2 \ldots p_m$
1. for i = 0, …, m do
2. 状態 i を作成する
3. for i = 0, …, m-1 do
4. 状態 i から i+1 へ p_{i+1} による遷移を作成する
5. for i = 0, …, m-1 do
6. u を状態 0 から状態 i に到達できる最短の文字列とする
7. for all $\sigma \in \Sigma$, $\sigma \neq p_{i+1}$ do
8. もし $u\sigma \in FACT(p)$ ならば
9. 状態 i から最初に出現する $u\sigma$ の終端の位置となる状態 j へ σ による遷移を作成する。

この構成法により作られた DFA をファクターオラクルという。具体的に、文字列 aabcaac に対するファクターオラクルは図 1 のようになる。

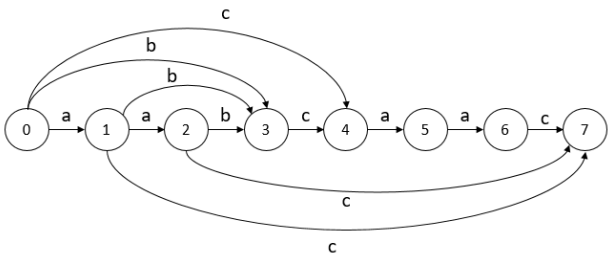

図 1 文字列 aabcaac に対するファクターオラクル

4. 提案法

4.1 アルゴリズム

本章では、新たに、オートマトンに対するファクターオラクルの構成法を示す。文字列の有限集合は DFA として表すことができるため、文字列のファクターオラクルの拡張になる。実際、一つの文字列の場合、Allauzen のファクターオラクルと提案法によるファクターオラクルは同型となることを示すことができる。以下に構成法を示す。入力は DFA であり、出力がファクターオラクルとなる。構成法は、入力となる DFA のすべての状態を初期状態と考え、NFA

† 信州大学大学院総合理工学研究科
‡ 信州大学工学部

から DFA を作成する手法を応用したものとなっている。なお、入力となる DFA M は非巡回型のため、各状態には、初期状態を 0 とし、トポロジカルオーダーで順に番号を付すことができる。したがって、各状態をそれに付された番号で表すこととする。さらに、$s=\{i_1,\ldots,i_j\}$ $(i_1<\cdots<i_j)$ を M の状態の部分集合としたとき、$\min(s)=i_1$ と定義する。

構成法

入力　非巡回型 DFA $M=(Q, \Sigma, \delta, q_0, F)$

出力　DFA $M_{FO}=(Q', \Sigma, \delta', s_0, F)$

1. $s_0 \leftarrow Q$, $m \leftarrow |Q|$, $Q' \leftarrow \{s_0\}$
2. for i =0, …, m do
3. s を min(s)=i なる Q’の状態とする。もしこのような状態がなければ次の i へ行く
4. for all $\sigma \in \Sigma$ do
5. s’ = δ (s, σ)を計算する
6. もし min(s’)=min(t)なる状態 t が Q’に存在すれば
7. δ ’(s, σ)=t と設定する
8. さもなければ、
9. δ ’(s, σ)=s’, Q’← Q’∪{s’}とする
10. End-for
11. End-for
12. F=Q’とする
13. M_{FO} を出力

提案法で構成した DFA に対し以下が成り立つ。

【定理 1】M_{FO} は次を満足する。

1. 少なくとも FACT(M)を受理する。
2. その状態数は M の状態数以下である。

さらに、入力を一つの文字列としたとき、次の定理を得る。

【定理 2】任意の文字列に対し、Allauzen のファクターオラクルと提案法で作成した DFA は同型である。

定理 2 は、我々のファクターオラクルは、Allauzen によって提案されたファクターオラクルを一般化したものであることを示唆している。

4.2 実行例

具体例を示す。まず、3 章で用いた文字列に対し、提案法により構成したファクターオラクルを図 2 に示す。これは、図 1 の Allauzen のファクターオラクルと同型になって

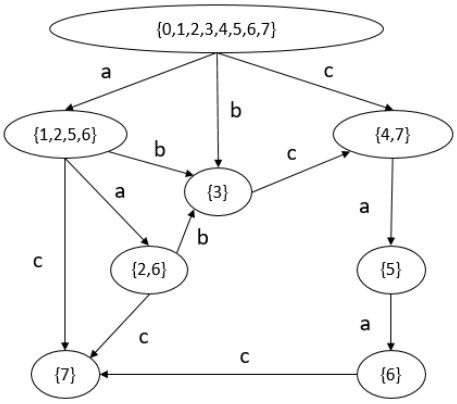

図 2　文字列 aabcaac に対する提案法によるファクターオラクル

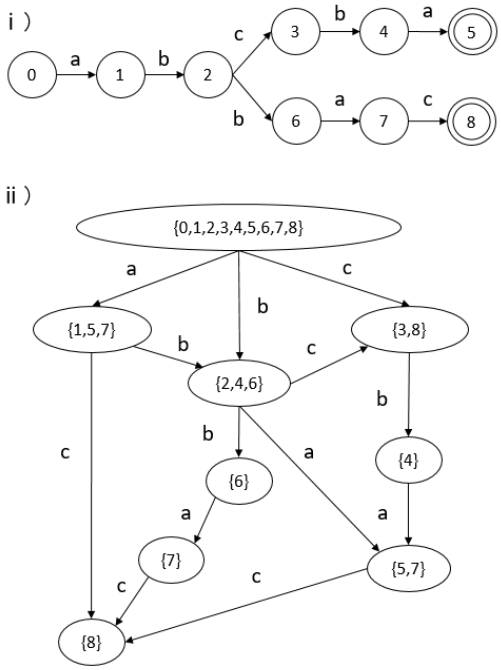

図 3　文字列集合{abcba,abbab}を受理する DFA(i)と i に対するファクターオラクル（ ii ）

いることがわかる。次に、図 3 に文字列の集合{abcba, abbab}を受理する DFA に対するファクターオラクルの作成例を与える。図 3(ii)のファクターオラクルは、すべての状態が受理状態となる。{abcba,abbab}のすべての部分文字列を受理するが、例えば、cbac のように部分文字列でない文字列も受理してしまう。これは、図3(ii)において状態{4}から a での実際の行先は状態{5}であるが、状態{5}と状態{5,7}とを統合したことにより発生したものである。

5. おわりに

本論文は、文字列に対して定義されるファクターオラクルを、オートマトンのファクターオラクルに拡張した。文字列の集合は DFA として表すことができるため、複数文字列のパターン照合への応用も可能である。今後の課題として、高速なアルゴリズムの開発や状態遷移数のより精密な評価があげられる。

謝辞

本研究は JSPS 科研費 JP17K00183 の助成を受けたものです。

参考文献

[1] C. Allauzen, M. Crochemore, M. Raffinot, “Factot Oracle: a new structure for pattern matching”, Proc. Of SOFSEM’99, LNCS, Vol.1725, pp.295-310 (1999).

[2] A. Blumer,J. Blumer, D. Haussler, and R. Mcconnell, “Cmplete Inverted Files for Efficient Text Retrieval and Analysis”, J. ACM, 34(3), pp. 578-595(1987).

[3] A. Blumer,J. Blumer, D. Haussler, “The smallest Automaton Recognizing the Subwords of a Text”, TCS, 40, pp. 31-55(1985).

[4] M. Mohri, P. Moreno, E. Weinstein, “Factot Automaton of Automaton and Application”, Proc. of CIAA2007, LNCS, Vol.4783, pp.168-179 (2007).