正規表現の貪欲な部分照合と Morris-Pratt アルゴリズム *

奥居　哲　　増田　拓也

1　はじめに

本プロジェクトでは正規表現の貪欲な部分照合を効率的に行う手法について研究を進めており，これまでに決定性有限オートマトン（DFA）を用いる手法を提案している．本稿では提案手法が古典的な Morris-Pratt 文字列照合アルゴリズム（あるいはそれと等価な Aho-Corasick オートマトンの構築）の正規表現への一般化になっていることを述べる．

2　DFA に基づく貪欲な部分照合

本プロジェクトの提案手法は，基本的には正規表現から Thompson 構成 [4] で構築される非決定性有限オートマトン（NFA）に部分集合構成（subset construction）を適用して（漸進的に）DFA を構築するものであるが，幾つかの点で既存の標準的な手法とは異なる．第 1 に，クリーネ閉包を実現する NFA の構築に Thompson の原論文にある古典的な方法を用いることで，貪欲な照合の意味論に合致しない照合を除外している．第 2 に，与えられた NFA の状態の列と終状態に到達済みか否かを示す真偽値のフラグとの組を DFA 状態とみなしている．列中の NFA 状態は ε-遷移で最終的に到達する状態だけを含み，これらの出現順位が貪欲な照合における複数の解を探索する際の優先順位を反映している．第 3 に，フラグが偽値である DFA 状態から新たな DFA 状態を生成する際には，NFA の始状態から ε-遷移で最終的に到達する状態も含めている．これにより部分照合の開始位置の効率的なシフト操作を実現している．第 4 に，ε-閉包を計算する際に途中の経路を記憶している．照合に成功した場合には（NFA の）終状態から始状態までこの経路を逆にたどることで部分式に捕獲される文字列の位置を得ることができる．

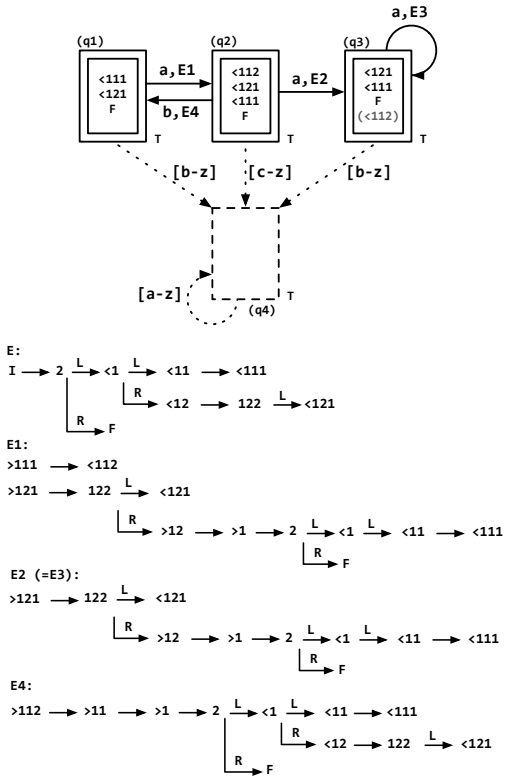

図 1　$(ab{+}a*)*$ から構築した貪欲 DFA

図 1 は本手法を用いて正規表現 $(ab{+}a*)*$ から図 2 の Thompson NFA を経由して生成された貪欲な DFA を例示している（NFA の各状態にはタグ（部分式の位置情報）を付与している）．この DFA は $q1, q2, q3$ に破線で示された沈下状態（sink）q_4 を加えた 4 つの状態からなる．q_1, q_2, q_3 はすべて終状態であり二重線で示されている．4 つの遷移における ε-閉包の経路が $E1, E2, E3, E4$ で示されている．E は NFA の始状態からの ε-閉包の経路である．

この DFA に文字列 *abaaabaa* を与えると，以下のよ

* Partial greedy regular expression matching and Morris-Pratt algorithm by S. Okui and T. Masuda

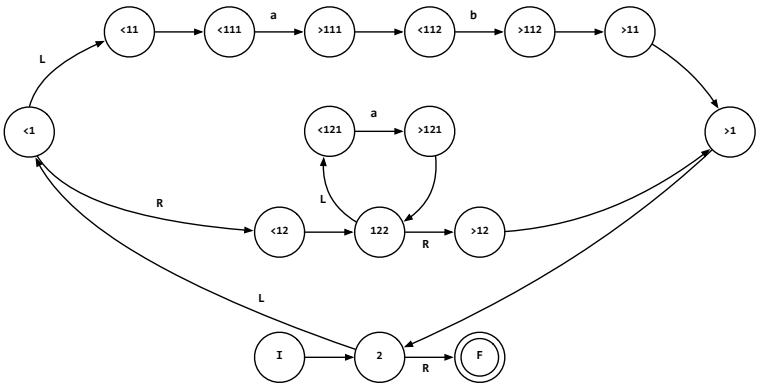

図 2　$(ab{+}a*)*$ から構築した Thompson NFA

うに決定的に遷移し沈下状態に至る．

$$q1 \overset{a,E1}{\to} q2 \overset{b,E4}{\to} q1 \overset{a,E1}{\to} q2 \overset{a,E2}{\to} q3 \overset{a,E3}{\to} q3 \overset{b}{\to} q4$$

このことから与えられた文字列の最初から 5 文字目までの範囲 $[0,5)$ に部分照合することが分かる．$E3$ $(= E2)$ の末尾にある終状態 F から始め $E3, E2, E1, E4$ を順に経由して $E1$ の頭の I までたどることで，以下のようなタグと文字の系列を得る．

$$\begin{aligned}\langle 1, \langle 11, \langle 111, &a, \rangle 111, \langle 112, \\ &b, \rangle 112, \rangle 11, \rangle 1, \langle 1, \langle 12, \langle 121, \\ &a, \rangle 121, \langle 121, \\ &a, \rangle 121, \langle 121, \\ &a, \rangle 121, \rangle 12, \rangle 1\end{aligned}$$

これから各部分式に捕獲される文字列の範囲が以下のように判明する．（部分式 1 と 121 が複数の範囲を捕獲しているのは，それぞれ 2，3 回反復して用いられているからである．）

1: $[0,2)$, $[2,5)$	112: $[1,2)$
11: $[0,2)$	12: $[2,5)$
111: $[0,1)$	121: $[2,3)$, $[3,4)$, $[4,5)$

3　Morris-Pratt アルゴリズムとの関連

パターンが一般的な正規表現ではなく単なる文字列として与えられる場合については，部分照合を線形時間で行う様々なアルゴリズムが知られている．Morris-Pratt (MP) アルゴリズム [2, 3] はそのひとつで，照合を再試行するパターン上の位置を予め計算しておく（後戻り表）．MP アルゴリズムと関連するものに Aho-Corasick（AC）オートマトン [1] がある．一般にパターンが単一文字列で与えられる場合，AC オートマトンは MP アルゴリズムの後戻り表から生成できる．逆に，AC オートマトンから MP アルゴリズムの後戻り表を再現できる．図 3 は $ababbaaa$ からつくられる MP アルゴリズムの後戻り位置と AC オートマトンである（ただし AC オートマトンにおける終状態からの遷移は取り除いてある）．アルファベットは $\{a, b\}$ である．

一方，図 4 は正規表現 $ababbaaa$ からつくられる NFA と，それから生成される DFA を例示している．この図の DFA と図 3 の AC オートマトンとを比較すると両者は一見して同じ構造をしており，提案手法と MP アルゴリズム，あるいは AC オートマトンとの関連を示唆している．

DFA$< \Sigma, Q_i, \mathrm{I}_i, \mathrm{F}_i, \delta_i >$（$i = 1, 2$）は，$Q_1$ から Q_2 への全単射 ϕ が存在し以下を満たすとき**同型**（isomorphic）であると言われる．

- $\phi(\mathrm{I}_1) = \mathrm{I}_2$
- $\phi(\mathrm{F}_1) = \mathrm{F}_2$
- $\phi(\delta_1(q, a)) = \delta_2(\phi(q), a)$

DFA C と D が同型であることを $C \cong D$ と記す．図 3 の AC オートマトンと図 4 の貪欲 DFA は同型である．以下，パターン文字列 p から作られる AC オートマトンあるいは貪欲 DFA をそれぞれ $AC(p)$，$GD(p)$ で表し，

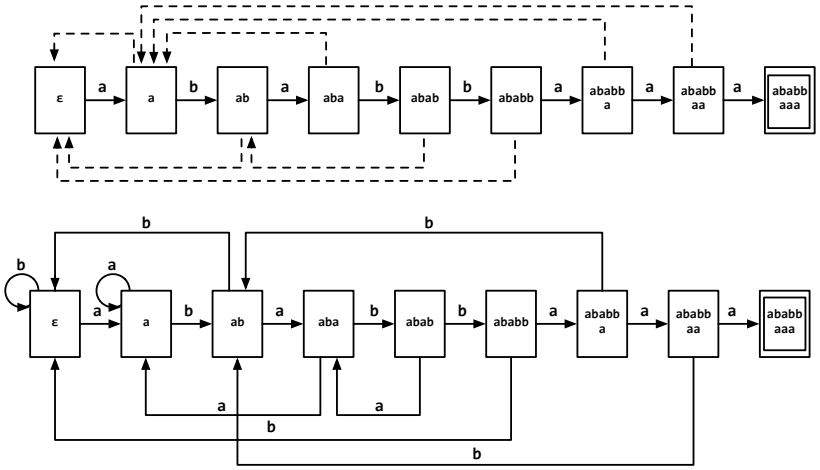

図 3　MP アルゴリズムにおけるパターン文字列 *ababbaaa* の戻り位置と AC オートマトン

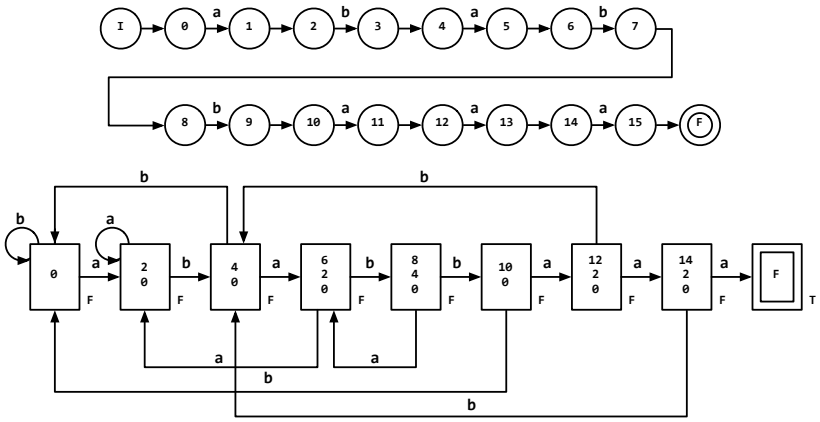

図 4　正規表現 *ababbaaa* から構築した Thompson NFA と貪欲 DFA

$AC(p) \cong DG(p)$ が任意の $p \in \Sigma^+$ に関して成り立つことの証明の概略を述べる．

$AC(p)$ あるいは $DG(p)$ を考え $\langle \Sigma, Q, \mathrm{I}, \mathrm{F}, \delta \rangle$ と置く．それぞれの構築法より任意の状態 $q \in Q$ にたいして $q = \delta(\mathrm{I}, x)$ となる p の接頭辞 x がひとつに決まることが分かる．つまり，$AC(p)$ あるいは $GD(p)$ の状態を p の接頭辞と同一視しラベル付けできる．特に $\mathrm{I} = \delta(\mathrm{I}, \varepsilon)$, $\mathrm{F} = \delta(\mathrm{I}, p)$ である．図 3 の AC オートマトンの状態はこの方法でラベル付けされている．

$AC(p) = \langle \Sigma, Q, \mathrm{I}, \mathrm{F}, \delta \rangle$ に対し，$Q \backslash \{\mathrm{I}, \mathrm{F}\}$ から Q への写像 $Bord$ を $Bord(\delta(\mathrm{I}, x)) = \delta(\mathrm{I}, y)$，ただし y は x の接尾辞でもあるような最長の真の接頭辞（すなわち x の最長の真のボーダ），と定義する．このとき，$AC(p)$ の作り方から以下のことが言える．

補題 1　パターン文字列 $p \in \Sigma^+$ から作られる AC オートマトン $\langle \Sigma, Q, \mathrm{I}, \mathrm{F}, \delta \rangle$ が与えられている．xa $(x \neq \varepsilon, a \in \Sigma)$ を p の真の接頭辞とするとき

$$\delta(q_x, b) = \begin{cases} q_{xa} & (b = a) \\ \delta(Bord(q_x), b) & (b \neq a) \end{cases}$$

$$Bord(q_{xa}) = \delta(Bord(q_x), a)$$

が成り立つ．ただし q_s は $\delta(\mathrm{I}, s)$ の略記である．

一方，$GD(p) = \langle \Sigma, Q, \mathrm{I}, \mathrm{F}, \delta \rangle$ に対し，$Q \backslash \{\mathrm{I}, \mathrm{F}\}$ から Q への写像 $Tail$ を $Tail(\langle [q_1, \ldots, q_n], f \rangle) =$

$\langle [q_2, \ldots, q_n], f \rangle$ ($n > 1$) と定義する．このとき，$GD(p)$ の作り方から以下のことが言える．

補題 2 パターン文字列 $p \in \Sigma^+$ から作られる貪欲な DFA $\langle \Sigma, Q, \mathrm{I}, \mathrm{F}, \delta \rangle$ が与えられている．xa ($x \neq \varepsilon, a \in \Sigma$) を p の真の接頭辞とするとき

$$\delta(q_x, b) = \begin{cases} q_{xa} & (b = a) \\ \delta(Tail(q_x), b) & (b \neq a) \end{cases}$$

$$Tail(q_{xa}) = \delta(Tail(q_x), a)$$

が成り立つ．ただし q_s は $\delta(\mathrm{I}, s)$ の略記である．

定理 1 $p \in \Sigma^+$ にたいして $AC(p) \cong GD(p)$ が成り立つ．

証明 $AC(p)$ を $\langle \Sigma, Q_{AC}, \mathrm{I}_{AC}, \mathrm{F}_{AC}, \delta_{AC} \rangle$ と置き，$GD(p)$ を $\langle \Sigma, Q_{GD}, \mathrm{I}_{GD}, \mathrm{F}_{GD}, \delta_{GD} \rangle$ と置く．x を p の接頭辞とするとき，$AC(p)$ の状態 $\delta_{AC}(\mathrm{I}_{AC}, x)$ を $GD(p)$ の状態 $\delta_{GD}(\mathrm{I}_{GD}, x)$ に写す写像 ϕ が同型写像になることを p の長さに関する数学的帰納法で示す．帰納法が期待通りはたらくように，証明したい命題に $\phi(Bord(\delta_{AC}(\mathrm{I}_{AC}, x))) = Tail(\phi(\delta_{AC}(\mathrm{I}_{AC}, x)))$（ただし x は p の任意の空でない真の接頭辞）を付加したものを証明する（いわゆる induction loading）．以下 $\delta_{AC}(\mathrm{I}_{AC}, x)$ を q_x と略記する．

まず $|p| \leq 2$ の場合，題意は容易に示せる．それ以外の場合，$p = xa$ （$a \in \Sigma, |x| \geq 2$）と置くと帰納法の仮定より，ϕ は $AC(x)$ から $GD(x)$ への同型写像であり，x の任意の空でない真の接頭辞 y に対して $\phi(Bord(q_y)) = Tail(\phi(q_y))$ が成り立つ．y として特に $yb = x$ となる y ($b \in \Sigma$) を考えると，補題 1，2 と帰納法の仮定より

$$\begin{aligned} \phi(Bord(q_x)) &= \phi(\delta_{AC}(Bord(q_y), b)) \\ &= \delta_{GD}(\phi(Boad(q_y)), b) \\ &= \delta_{GD}(Tail(\phi(q_y)), b) \\ &= Tail(\phi(q_x)) \end{aligned}$$

が言えるので，これより induction loading で付加した命題が成立する．$AC(p)$ の状態 q_{xa} と $GD(p)$ の状態 $\phi(q_{xa})$ は $AC(x)$ と $GD(x)$ に含まれないので，ϕ は Q_{AC} から Q_{GD} への全単射である．さらに q_x，$\phi(q_x)$ からの遷移に関して，上で証明した等式と補題 1，2，帰納法の仮定より以下の二つの等式が成り立つ．

$$\phi(\delta_{AC}(q_x, a)) = \phi(q_{xa}) = \delta_{GD}(\phi(q_x), a)$$

$b \in \Sigma$ （$b \neq a$）に対して

$$\begin{aligned} \phi(\delta_{AC}(q_x, b)) &= \phi(\delta_{AC}(Bord(q_x), b)) \\ &= \delta_{GD}(\phi(Bord(q_x)), b) \\ &= \delta_{GD}(Tail(\phi(q_x)), b) \\ &= \delta_{GD}(\phi(q_x), b) \end{aligned}$$

$AC(p)$ の始状態 q_ε と終状態 q_p は，ϕ によってそれぞれ $GD(p)$ の始状態と終状態に写される．以上により ϕ は $AC(p)$ から $DG(p)$ への同型写像である．□

この証明は結局，$AC(p)$ と $GD(p)$ にそれぞれ $Bord$ と $Tail$ という演算子を含めて拡張した代数系が同型であることを述べている．つまり，AC オートマトンにおける $Bord$ 関数のはたらきを貪欲 DFA では $Tail$ 関数が担っており，Morris-Pratt アルゴリズムにおける後戻り先は，貪欲 DFA における NFA 状態の列から先頭の状態（つまり，もとの NFA における最も優先順位の高い遷移先）を削除する単純な操作により得られる．

4 おわりに

本プロジェクトで提案した貪欲 DFA が Morris-Pratt アルゴリズムの後戻り表，およびそれと等価な Aho-Corasick オートマトンの任意の正規表現への一般化であり，正規表現が単なる文字列の場合には両方のオートマトンの構造が一致することを述べた．

参考文献

[1] A. V. Aho and M. Corasick. Efficient string matching: An aid to bibliographic search. *Comm. ACM*, 18:333–340, 1975.

[2] J. H. Morris Jr. and V. R. Pratt. A linear pattern-matching algorithm. Technical Report 40, University of California,Berkeley, 1970.

[3] D. E. Knuth, J. H. Morris Jr., and V. R. Pratt. Fast pattern matching in strings. *SIAM J. Comput.*, 6:323–350, 1977.

[4] K. Thompson. Regular expression search algorithm. *Communication of the ACM*, 11(6):419–422, 1969.