繰り返し構造からの文字列推測の困難さ

松原 渉* 篠原 歩*

2009 年 7 月 22 日

概 要

本論文は，文字列の繰り返し構造に関する理解を深めることを目的に，文字列の繰り返し構造からの文字列推測問題を提案し，その問題の時間計算量および，アルファベットサイズ依存性について考察する．まず，アルファベットサイズの制約がないときや，出力がバイナリであるときに，高速に文字列推測ができることを示す．その一方で，アルファベットサイズが 4 以上のときや，最小アルファベットサイズを求める問題は NP 困難であることを示す．

1 はじめに

1.1 文字列の繰り返し構造

繰り返し構造とは文字列におけるもっとも基本的な性質の一つであり，文字列処理アルゴリズムやゲノム情報学の分野で数多くの応用研究がある．中でも，周期性が延長不可能な部分文字列を連とよび，数多くの研究がなされている．

最近では組み合わせ学の観点から，連の最大数に関する研究が積極的に行われている．これまでの解析のなかで，長さ n の文字列に含まれる連の最大数 $\rho(n)$ の値として，次の上限 [3]，下限 [8, 9] が知られている．

$$0.944575 < \frac{\rho(n)}{n} < 1.029$$

連の最大数について研究は進んでいるが，全容解明には至っておらず，繰り返し構造の本質的理解が求められている．なかでも [3] では，文字列に含まれる連のパターンを網羅的に生成し，解があるかどうかを大規模計算機によって検算することにより，もっともよい上限を与えている．その中では，繰り返し構造に関する組み合わせ的性質を活用することで，探索の効率化がなされている．

本論文では文字列中の連の集合が与えられたとき，文字列にどのような制約が加わるかに興味をもち，文字列の繰り返し構造に関する情報から，文字列を推測する問題を定式化し，その計算量を解析する．

1.2 文字列推測問題

文字列推測問題とは，ある文字列に関するデータ構造が与えられたとき，そのデータ構造を満たすような文字列を推測せよという問題である．データ構造の構築に関する逆操作に相当することから，**逆問題**ともよばれる．このような逆問題を考える意義の一つとして，必要十分条件 (if-and-only-if) を明確にすることにより，そのデータ構造を完全に特徴付られることがあげられる．また，応用として，あるデータベースの索引を構築したときに，正しく索引が構築できたかどうかを検査することにも役立つ．

これまで，いくつかのデータ構造に対する逆問題についての研究がなされている．グラフについては，坂内ら [1] が DASG や DAWG から文字列を線形時間で推測するアルゴリズムを提案している．配列に関する研究として，Franek ら [5] は，ボーダー配列の逆問題を線形時間で解くアルゴリズムを提案している．ボーダー配列とは，文字列照合における KMP アルゴリズムの失敗関数の遷移先を表現する配列としてよく知られている．Duval ら [4] は，同様の問題

*東北大学大学院情報科学研究科

に対して，整数アルファベットに対してオンラインの線形時間アルゴリズムを提案している．坂内ら [1] は，接尾辞配列の逆問題を解く線形時間アルゴリズムを提案している．また，Crochemore ら [2] は，文字列照合における BM アルゴリズムに用いられる，接頭辞配列の逆問題を線形時間で解くアルゴリズムを提案している．

1.3 本論文の成果

本論文では，文字列の繰り返し構造が与えられたとき，元の文字列を推測する問題について，時間計算量やアルファベットサイズ依存性について考察する．構成は以下の通りである．

4 章ではアルファベットサイズを制約しない問題について，$O(n^2)$ 時間で解けることを示す．また 5 章ではバイナリアルファベットについて，連の逆問題が $O(n \log n)$ で解けることを示す．

6 章では，この問題の困難性を示すために，グラフの点彩色問題から，連の逆問題に多項式時間帰着を示す．これにより，最小アルファベットサイズを求める問題が NP 困難であること，およびアルファベットサイズが 4 以上について，判定問題が NP 完全であることが導かれる．

2 準備

2.1 文字列

自然数の集合を $\mathcal{N}$ とかく．文字の有限集合を**アルファベット**とよぶ．$\Sigma = \{1, 2, \dots\}$ とし，特に $\Sigma_k = \{1, 2, \dots, k\}$ とする．アルファベット Σ に対して，Σ^* の要素を**文字列**という．文字列 w の長さを $|w|$ と表す．文字列 $w = XYZ$ について，X, Y および Z をそれぞれ，w の**接頭辞**, **部分文字列**, **接尾辞**と呼ぶ．$1 \le i \le |w|$ に対し，w の i 番目の文字を $w[i]$ と表す．また $1 \le i \le j \le |w|$ に対し，T の i 文字目から j 文字目までの部分文字列を $w[i:j]$ と表す．

文字列 w は $1 \le i \le |w|-p$ について $w[i] = w[i+p]$ が成り立つとき，**周期** p をもつという．任意の整数 $k \ge 2$ として，$w = u^k$ となる u が存在しないとき，w は**素**な文字列であるという．長さ n の文字列 w について，次の条件を満たす 3 つ組 $\langle i, j, p \rangle$ を w **に含まれる連**という．

条件 1 $w[i:j]$ は周期 $p \le \frac{j-i+1}{2}$ をもつ,

条件 2a $i = 1$ または $w[i-1] \ne w[i+p-1]$,

条件 2b $j = n$ または $w[j+1] \ne w[j-p+1]$,

条件 3 $w[i:i+p-1]$ は素な文字列である．

また，文字列 w について，w に含まれる連の集合を $Runs(w)$ とかく．すなわち任意の文字列の連集合は $\mathcal{N}^3$ の部分集合で表すことができる．

定理 1 ([7])**.** 長さ n の文字列 w が与えられたとき，w に含まれる連の集合 S は $O(n)$ 時間で求まる．

2.2 グラフ

$G = (V, E)$ を無向グラフとする．すべての辺 $(i, j) \in E$ について，$f(i) \ne f(j)$ であるような写像 $f : V \to \mathcal{N}$ を G の**点彩色**という．整数 k として，グラフ G が k 点彩色可能であるとは，k 色以下の点彩色が存在することを意味する．

問題 1. [点彩色問題]
入力: 無向グラフ $G = (V, E)$
出力: G に対する最小の k の点彩色 $f : V \to \{1, 2, \dots k\}$ を求めよ．

問題 2. [k 点彩色問題]
入力: 無向グラフ G, 整数 k
出力: G が k 点彩色可能か決定せよ．

定理 2. 問題 1 は NP 困難である．

定理 3 ([6])**.** $k \ge 3$ のとき，問題 2 は NP 完全である．

3　連の情報による制約

この章では，連の情報 $S \subseteq \mathcal{N}^3$ が与えられた際に，$Runs(w) = S$ を満たす文字列 w にどのような制約が加わるのかを考察する．

w を長さ n の文字列であるとして，連による制約を各文字に対応する n 個の点をもつグラフ G で表現することを考える．点集合 $V = \{1, 2, \ldots, n\}$ とし，次の 2 つの辺集合を考える．$w[i] = w[j]$ となる点の組 (i, j) を辺集合 R で表し，$w[i] \neq w[j]$ となる点の組 (i, j) を辺集合 $\bar{R}$ で表す．

ここで連の定義から，$\langle i, j, p\rangle \in Runs(w)$ であるとき，条件 1 より，

$$R \supseteq \{(k, k+p) \mid i \leq k \leq j-p\}$$

となる．また，条件 2 から

$$\begin{aligned} \bar{R} &\supseteq \{(i-1, i-1+p) \quad | \quad i \neq 1\}, \\ \bar{R} &\supseteq \{(j+1, j+1-p) \quad | \quad j \neq n\} \end{aligned}$$

と表せる．さらに，連が存在しないという情報からも制約がかかることがある．周期 1 の連が存在する区間を取り出すと，

$$R_1 = \{k \mid i \leq k < j, \langle i, j, 1\rangle \in S\}$$

となるので，$k \in R_1$ のとき，かつそのときに限り，$w[k] = w[k+1]$ が成り立つ．すなわち，連が存在しないなら，別の文字が割り当たることから，

$$\bar{R} \supseteq \{(k, k+1) \mid k \notin R_1\}$$

がいえる．ここで，同じ文字が割り当たる点同士を縮約してひとつの点と見なしても，辺集合 $\bar{R}$ のもつ性質を失わないので，辺集合 R に基づいてグラフ G を縮約できる．G を縮約して得られたグラフを $G_S = (V_S, E_S)$ とすると，連情報 S を満たすための必要条件を表すグラフとなる．V_S は集合 $\{1, 2, \ldots, n\}$ を分割した $\{V_1, V_2, \ldots, \}$ からなり，ある点 V_k について $\{i, j\} \subseteq V_k$ ならば，$w[i] = w[j]$ となる．また，ある辺 $(V_x, V_y) \in E_S$ について，$i \in V_x, j \in V_y$ ならば，$w[i] \neq w[j]$ となる．すなわち，$Runs(w) = S$ を満たす文字列 w はグラフ G_S における点彩色の解と対応がつく．

例 1. $S = \{\langle 1, 4, 2\rangle, \langle 4, 7, 2\rangle, \langle 7, 8, 1\rangle\}$ として，$Runs(w) = S$ を満たす文字列 $w(|w| = 9)$ を考える．S を元にグラフを構成すると，

$$\begin{aligned} R &= \{(1, 3), (2, 4), (4, 6), (5, 7), (7, 8)\}, \\ \bar{R} &= \{(3, 5), (6, 8), (8, 9)\}. \end{aligned}$$

これを縮約したグラフ $G_S = (V_S, E_S)$ は，

$$\begin{aligned} V_1 = \{1, 3\}, \quad V_2 = \{2, 4, 6\}, \\ V_3 = \{5, 7, 8\}, \qquad V_4 = \{9\}. \end{aligned}$$

として，

$$\begin{aligned} V_S &= \{V_1, V_2, V_3, V_4\}, \\ E_S &= \{(V_1, V_3), (V_2, V_3), (V_3, V_4)\} \end{aligned}$$

となる．G_S を図示すると図 1 のようになる．

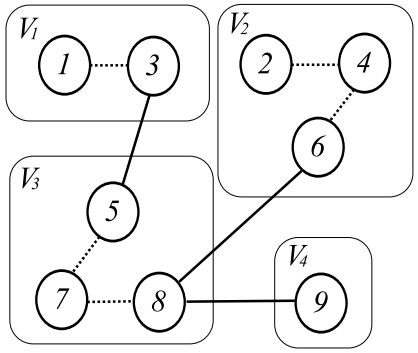

図 1: $S = \{\langle 1, 4, 2\rangle, \langle 4, 7, 2\rangle, \langle 7, 8, 1\rangle\}$ を表すグラフ G_S.

ただし，連の表す情報すべてをこのグラフに表現することはできないので，逆は成り立たない．たとえば，条件 3 で述べているように，$w[i : i+p-1]$ が素な文字列であることはグラフで表現できない．また，周期 2 以上の連が，特定の部分に存在しないという情報も表現が難しい．しかしながら，多くの必要条件を書き表すことができるので，解の候補とな

る文字列を絞り込むことができる．これを踏まえて，次章より連の情報からの文字列推測問題について考察していく．

4 無限アルファベット上の文字列推測

この章では，アルファベットサイズを限定しない問題について考察する．

問題 3. (繰り返し構造からの文字列推測問題)
文字列に含まれる連を表す 3 つ組 (始点，終点，周期) の集合 $S \subseteq \mathcal{N}^3$ と整数 n が与えられたとき，$Runs(w) = S$ を満たす文字列 $w \in \Sigma^n$ が存在するかどうかを判定せよ．存在するなら，そのような w をひとつ出力せよ．

前章で述べたように，文字列 w が与えられた連情報 S を満たすとき，w に加わる制約はグラフで表現することができる．本章では，アルファベットサイズの制約がないことから，同値関係 R に基づいて，同じ文字が割り当たるものをまとめるだけで十分である．同値関係 R を満たす文字列を T_S と表記する．

例 2. 集合 $S = \{\langle 3,4,1\rangle, \langle 4,8,2\rangle, \langle 1,6,3\rangle\}$ に関する同値関係を求めると，$R = \{(1,4),(6,7),(1,4),(2,5),(3,6),(4,7),(5,8)\}$ となる．文字列 $T_S =$ abaababa は同値関係 R を満たす．

定理 4. 集合 $S \subseteq \mathcal{N}^3$ が与えられたとき，S がある文字列の連集合であるかどうかは $O(n^2)$ 時間で判定できる．

証明. 長さ n の文字列に含まれる連の個数は $O(n)$ であるから，$|S| = O(|w|)$ となる．ゆえに，同値関係 R の大きさは $O(n^2)$ であり，すなわち，解の候補文字列 T_S も同様の時間で求められる．この候補は連となるためのすべての必要条件を満たしているわけではないので，残りの条件も満たすかどうか確かめなければならない．そのため，T_S に含まれる連集合を求め，$Runs(T_S) = S$ かどうかを判定する．定理 1 より，$Runs(T_S)$ は $O(n)$ 時間で求まり，集合の等価性判定は $O(n \log n)$ 時間で行える．以上より, S がある文字列の連集合であるかどうかは，$O(n^2)$ 時間で判定できる． □

5 バイナリ文字列上の文字列推測

繰り返し構造からの文字列推測問題において，推測する文字列のアルファベットサイズに関する依存性を議論する．

問題 4. 連を表す 3 つ組 (始点, 終点, 周期) の集合 $S \subseteq \mathcal{N}^3$ と整数 n, k が与えられたとき，$Runs(w) = S$ かつ $|w| = n$ を満たす文字列 $w \in \Sigma_k^*$ が存在するかどうかを判定せよ．存在するなら，そのような w をひとつ出力せよ．

まず本章では，問題がバイナリ文字列の解 ($k = 2$) を持つとき，高速に文字列推測が可能であることを示す．

定理 5. $k \leq 2$ について，問題 4 は $O(n \log n)$ 時間で解くことができる．

証明. まず，集合 S から周期が 1 であるものを取り出す．周期 1 の連に入っている区間には同じ文字を割り当て，それ以外の場合には異なる文字を割り当てればよい．すなわち，$R_1 = \{k \mid i \leq k < j, \langle i, j, 1\rangle \in S\}$ とすると，$k \in R_1$ なら，$w[k] = w[k+1]$ であり，$k \notin R_1$ なら，$w[k] \neq w[k+1]$ が成り立つ．ここで，バイナリであるときには，異なる文字が一意に定まるので，解の候補文字列 w は $O(n)$ 時間で見つけられる．候補 w が解であるかどうかを確かめるには，定理 4 の場合と同様にして，$O(n \log n)$ 時間で行える．ゆえに，全体で $O(n \log n)$ 時間で解ける． □

6 アルファベットサイズ制約問題の困難さ

この章では，連の逆問題がアルファベットサイズ 4 以上のとき，NP 完全となることを示す．アイデア

としては，NP 完全問題として知られている，点彩色問題から連の逆問題へ多項式時間変換することで証明する．

6.1 インスタンス変換

連の情報を満たすような文字列が，グラフの彩色に対応するよう設計する．すなわち，連情報による局所的周期性が与えられたとき，対応する文字列の代入に発生する制約が，グラフの辺集合による彩色に対する制約を模倣するよう設計する．

k-点彩色問題のインスタンスを $\langle G, k\rangle$ と表す．これを連の逆問題のインスタンス $\langle S, k+1, n\rangle$ に変換する．

まず，グラフ $G = (V, E)$ を表す文字列 $g \in (\Sigma_{|V|} \cup \{\$\})^*$ を構築する方法を示す．第一に，点集合 V を文字列 v に変換する．$V = \{v_1, v_2, \ldots, v_n\}$ とすると，k 番目の点 v_k を次の文字列 u_k で表す．

$$u_k = \$k^{(k+1)}\$k^{(k+1)}\$$$

また，v はグラフの点集合 V を表現する文字列である．

$$v = u_1 u_2 \ldots u_k$$

例として，図 2 に 3 つの点をもつ点集合を表す文字列を示す．

次に，辺集合を文字列で表す方法を示す．$1 \le k \le n$ なるすべての k について，文字列 ℓ_k, r_k を次のように定義する．

$$\begin{aligned} \ell_k &= v_1 v_2 \ldots v_{k-1}\$k^{k+1}, \\ r_k &= k^{k+1}\$v_{k+1} \ldots v_n. \end{aligned}$$

すなわち，ℓ_k は v の接頭辞であり，r_k は v の接尾辞となる．これを用いて，辺 (i, j) を表す文字列を $\ell_i r_j$ と表す．

辺および点を表す文字列を用いて，$vv\ell_i r_j vv$ という文字列を作る．このとき，文字列には図 3 のような連が発生する．

このガジェットを用いてグラフ $G = (V, E)$ から文字列 g への変換関数 enc を次のように定義する．

$$g = enc(G) = \Pi_{k=1}^{n}(a_k \ell_{i_k} r_{j_k}) a_{n+1}$$

ただし，k が奇数のとき，$a_k = v^2$ であり，k が偶数のとき $a_k = v^3$ である．すなわち，文字列 g は辺を表す文字列と vv および vvv の組み合わせで構築される．このとき，$\ell_{i_k} r_{j_k}$ によって，v の代入に制約をかけることができる．このガジェットを用いて作成した文字列 g に含まれる連集合 $S = Runs(g)$ を求め, これを連の逆問題のインスタンス $\langle S, k+1, n\rangle$ として変換が完了する．

6.2 帰着の正当性

次に正しく，多項式時間帰着できているどうかを以下の 2 つの主張により示す．

主張 1. あらゆるグラフ $G = (V, E)$ について，インスタンス変換は多項式時間で行える．

証明. はじめに，グラフ G から文字列への変換に関して，変換後の文字列長 $|enc(G)|$ について評価する．定義より，$|v_k| = |\$k^{(k+1)}\$k^{(k+1)}\$| = 2k + 5$ であるので，点集合を表す文字列 v の長さは，$|v| = \sum_{k=1}^{n} |v_k| = \sum_{k=1}^{n}(2k + 5) = O(n^2)$ となる．また，任意の i, j(ただし $1 \le i < j \le n$) について $|\ell_i r_j| < |v|$ となることと，任意のグラフ $G = (V, E)$ について，$|E| = O(|V|^2)$ となることに注意すると，グラフ G を表す文字列 g の長さは，

$$\begin{aligned} |g| &= \sum_{k=1}^{m}(|a_k \ell_{i_k} r_{j_k}|) + |a_{n+1}| \\ &< m(3|v| + |v|) + 3|v| \\ &= 4m|v| + 3|v| \\ &= O(m|v|) \\ &= O(n^4) \end{aligned}$$

となり，変換後の文字列長は与えられたグラフの点の数に対して，多項式サイズで収まる，次に，文字列 g に含まれる連を取り出すためには，定理 1 より，

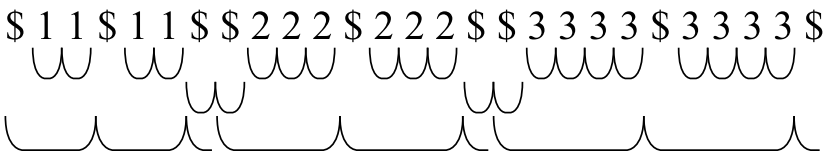

図 2: 3 つの点をもつ点集合を表す文字列.

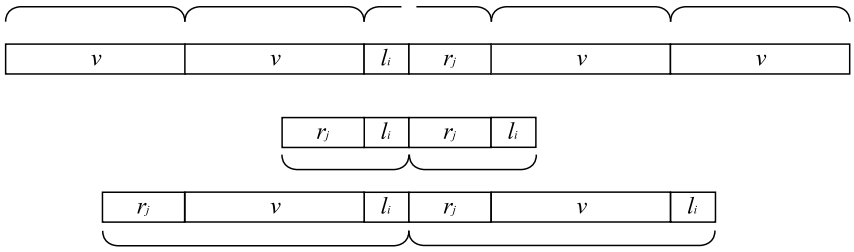

図 3: 辺 (i, j) を表す文字列.

$O(g) = O(|V|^4)$ 時間で求められる．以上よりインスタンス変換は多項式時間で処理できる． □

主張 2. 変換後のインスタンス $\langle S, k{+}1, n\rangle$ に解があるとき，かつそのときに限り，グラフの k 点彩色問題 $\langle G, k\rangle$ に解が存在する．

証明. (sketch)
与えられた連集合 S に関する代入制約を表現するグラフ G_S を構築する．このグラフ G_S から，$1 \in V_k$ となる点 V_k および，V_k につながるすべての辺を取り除くと，彩色問題の入力 G と等しくなる．すなわち，G が k 点彩色可能であるとき，連の逆問題のインスタンス $\langle S, k{+}1, n\rangle$ に解が存在する． □

上記の主張 1, 2 より以下の定理が導かれる．

定理 6. グラフの k 点彩色問題は，アルファベット Σ_{k+1} 上の連の逆問題に多項式時間で帰着できる．

定理 6 と，グラフの点彩色問題の NP 完全性から，以下の系が導ける．

系 1. 連の集合 S と文字列長 n が与えられたとき，$Runs(w) = S$, $|w| = n$ を満たすような文字列の中で，アルファベットサイズが最小となるものを見つけ出す問題は NP 困難である．

系 2. $k \geq 4$ について，問題 4 は NP 完全である．

7 まとめと今後の課題

本論文では，与えられた連集合を満たす文字列を推測する問題を定式化し，時間計算量やアルファベットサイズ依存性を解析した．結果として，アルファベットサイズを制約しない場合には $O(n^2)$ 時間で判定できることを示した，その一方で，最小アルファベットサイズを求める問題は NP 困難であることを示した．またアルファベットサイズ k を制約した問題を考え，$k \leq 2$ のとき，$O(n \log n)$ 時間で判定可能で，$k \geq 4$ のとき，NP 完全であることを示した．今後の課題は，$k = 3$ の問題を多項式時間で解けるかを解明することである．

謝辞

本研究を進めるにあたり，定式化や証明において，非常に有益なコメントを頂いた石野 明先生 (現 Google Japan Inc.) に感謝いたします．

参考文献

[1] H. Bannai, S. Inenaga, A. Shinohara, and M. Takeda. Inferring strings from graphs ans arrays. In *MFCS2003*, volume 2747 of *LNCS*, pages 208–217. Springer, 2003.

[2] J. Clément, M. Crochemore, and G. Rindone. Reverse engineering prefix tables. In *STACS*, pages 289–300, 2009.

[3] M. Crochemore, L. Ilie, and L. Tinta. Towards a solution to the "runs" conjecture. In *Proc. CPM 2008*, volume 5029 of *LNCS*, pages 290–302, 2008.

[4] J. Duval, T. Lecroq, and A. Lefevre. Border array on bounded alphabet. In *Proc. The Prague Stringology Conference '02 (PSC'02)*, pages 28–35, 2002.

[5] F. Franek, S. Gao, W. Lu, P. J. Ryan, W. F. Smyth, Y. Sun, and L. Yang. Verifying a border array in linear time. *J. Comb. Math. Comb. Comput*, 42:223–236, 2000.

[6] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified np-complete problems. In *STOC '74: Proceedings of the sixth annual ACM symposium on Theory of computing*, pages 47–63, New York, NY, USA, 1974. ACM.

[7] R. Kolpakov and G. Kucherov. Finding maximal repetitions in a word in linear time. In *Proc. 40th Annual Symposium on Foundations of Computer Science (FOCS'99)*, pages 596–604, 1999.

[8] W. Matsubara, K. Kusano, H. Bannai, and A. Shinohara. A series of run-rich strings. In *Proc. 3rd International Conference on Language and Automata Theory and Applications (LATA'09)*, pages 578–587. Springer, 2009.

[9] J. Simpson. Modified Padovan words and the maximum number of runs in a word. *Australasian Journal of Combinatorics (to appear)*, 2009.