

未決定文字列における一意単語の検索の困難さ

*Dominik Köppl*¹ and *Jannik Olbrich*²

¹: 山梨大学, ²: University of Ulm



山梨県若手研究者奨励事業費補助金 2291 による助成研究

$$\tilde{T} = A \begin{Bmatrix} A \\ C \end{Bmatrix} C \begin{Bmatrix} G \\ A \end{Bmatrix}, \text{一意単語: } AA, CC, CA, G$$

一意単語

定義 (一意単語 = unique word)

- T : 文字列, Σ : アルファベット
- T の一意単語 X : 文字列 T 中に部分文字列として丁度 1 回出現する文字列

例

- $T = AACAC, \Sigma = \{A, C\}$
- MUSs: AA, AC

定義 (最小一意単語 = MUS (minimum unique substring))

- X : 一意単語
- X は最小一意単語 : $\Leftrightarrow X$ のすべての真の部分文字列が T 中に少なくとも 2 回出現

未決定文字列

定義

- 任意の位置に複数の代替文字を持つ
- 位置を記号と呼ぶ
- \tilde{T} : 未決定文字列
- $\tilde{T}[i] \subset \Sigma$

例

IUPAC 表記法は DNA や RNA 配列における未決定の核酸を表現、例：

- R は A または G
- N は任意の核酸

$$\tilde{T} = A \begin{Bmatrix} A \\ C \end{Bmatrix} C \begin{Bmatrix} G \\ A \end{Bmatrix}$$

$$\mathcal{L}(\tilde{T}) = \{AACG, AAC A, ACCG, ACCA\}$$

- \tilde{T} : 未決定文字列の例
- $\mathcal{L}(\tilde{T})$: \tilde{T} の言語
- $\mathcal{L}(\tilde{T})$ は \tilde{T} で表現されている文字列の集合

未決定文字列の MUS

定義 (一意単語)

- \tilde{T} : 未決定文字列
- \tilde{T} の一意単語 X :
 - 少なくとも一つの $\mathcal{L}(\tilde{T})$ の文字列中に出現
 - すべての $\mathcal{L}(\tilde{T})$ の文字列中に部分文字列として最大 1 回出現する文字列

$$\tilde{T} = A \begin{Bmatrix} A \\ C \end{Bmatrix} C \begin{Bmatrix} G \\ A \end{Bmatrix}$$

$$\mathcal{L}(\tilde{T}) = \{AACG, AAC A, ACCG, ACC A\}$$

MUS: AA, CC, AC, CA, G

定義 (MUS)

- X : 一意単語
- X は最小一意単語: $\Leftrightarrow X$ のすべての真の部分文字列が少なくとも一つの $\mathcal{L}(\tilde{T})$ の文字列中に少なくとも 2 回出現

計算量

- 単純な文字列： $O(n)$ 時間 Pei+'13
- 連超圧縮された文字列： $O(m \lg m)$ 時間 Mieno+'16
- 未決定文字列： NP 完全 (一つの MUS の検索だけ)

ただし

- m は入力文字列の連超圧縮サイズ
- 注意： $|\tilde{T}| = n$ としても、 $|\mathcal{L}(\tilde{S})| = \Omega(2^n)$ になる可能性がある

例：

$$\tilde{T} = \left\{ \begin{array}{c} a \\ b \end{array} \right\} \left\{ \begin{array}{c} a \\ b \end{array} \right\} \left\{ \begin{array}{c} a \\ b \end{array} \right\} \left\{ \begin{array}{c} a \\ b \end{array} \right\} \dots$$

問題 (k -一意単語問題)

- 入力：未決定文字列 \tilde{S} と整数 k
- 出力：長さが k の一意単語が \tilde{S} に存在するかどうか
- k -一意単語問題は判定問題
- 判定問題の答えは Yes・No

例

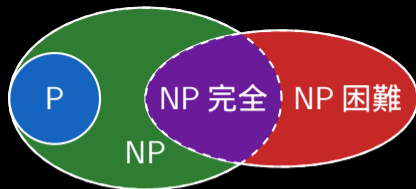
$$\tilde{T} = A \left\{ \begin{array}{c} A \\ C \end{array} \right\} C \left\{ \begin{array}{c} G \\ A \end{array} \right\}$$

$$\mathcal{L}(\tilde{T}) = \{AACG, AACA, ACCG, ACCA\}$$

- $k = 1$: Yes: G
- $k = 2$: Yes: CA
- $k \geq 3$: No

問題 (k -一意単語問題)

- ▶ 入力：未決定文字列 \tilde{S} と整数 k
- ▶ 出力：長さが k の一意単語が \tilde{S} に存在するかどうか
- ▶ k -一意単語問題は判定問題
- ▶ 判定問題の答えは Yes・No



方針

- ▶ 一意単語問題 \in NP を証明
 - ▶ 一意単語問題 \in NP 困難 を証明
- \Rightarrow 一意単語問題 \in NP 完全

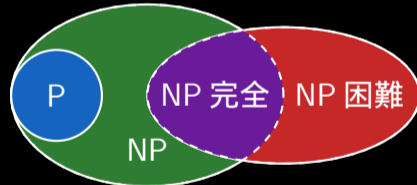
NP 問題

定義 (クラス NP)

- 問題 M は以下の条件を満たすと、NP の要素となる
- 入力に対する正しい出力が Yes であるとき、それを多項式時間で検証するための証拠が存在

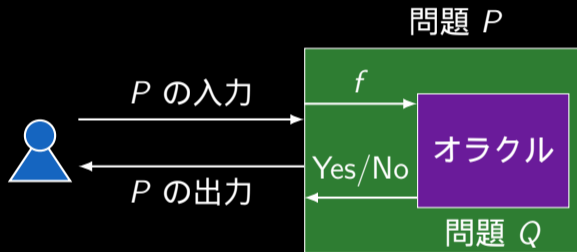
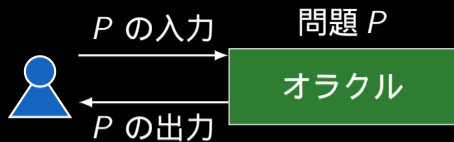
問題 (証明検証)

- X : 文字列, \tilde{S} : 未決定文字列
 - X は \tilde{S} の MUS の一つかどうか
 - X が \tilde{S} に何回出現するか、多項式時間で判断 Abrahamson'87
 - X のすべての部分文字列の個数は $O(n^2)$
 - 証明検証問題を多項式時間で解ける
- ⇒ 一意単語問題は NP 問題



定義 (Karp の帰着方法)

- 入力：判定問題 P と Q 、多項式時間アルゴリズム f
- $f(P \text{ の入力}) = Q \text{ の入力}$
- P の入力 S が P の Yes 入力 $\Leftrightarrow f(S)$ は Q の Yes 入力
- その際、 P が Q に多項式時間多対一帰着可能であると言う



補題

- P は NP 困難
- P が Q に多項式時間多対一帰着可能
- $\Rightarrow Q$ も NP 困難

- ▼ 一意単語問題は NP 困難だと証明したい $\Rightarrow Q =$ 一意単語問題
- ▼ $P = 3\text{-SAT}$ とする

定義 (3-SAT 判定問題)

入力:

- ▼ 連言標準形 (CNF: conjunctive normal form) の式 F
- ▼ F は一連の節 C_i を連言 (AND) で結合
- ▼ 各節 C_i は3つのリテラルの選言である
- ▼ n 個の変数 x_1, \dots, x_n が順序付け

出力: 論理式全体の値を真にするような真偽値 x_1, \dots, x_n があるかどうか

3-SAT 判定問題は NP 困難 Karp'72

例:

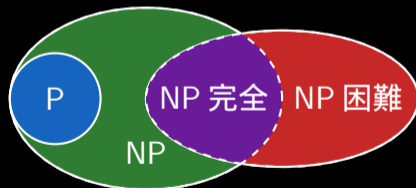
- ▼ 変数 x_1, x_2, x_3, x_4
- ▼ $C_1 = x_1 \vee \neg x_2 \vee x_3$
- ▼ $C_2 = \neg x_1 \vee x_2 \vee \neg x_4$
- ▼ $C_3 = x_2 \vee x_3 \vee \neg x_4$
- ▼ $F = C_1 \wedge C_2 \wedge C_3$

出力: Yes:

- ▼ $x_1 = x_2 =$ 真
- $\Rightarrow F =$ 真

方針

- 線形量を取る未決定文字列 \tilde{S} で F の反対を表現
 - つまり、 C_i を満たさない真偽を \tilde{S} の長さ n の部分文字列で表現
 - 長さ n のすべての可能な部分文字列を \tilde{S} で列挙
- ⇒ F が充足する割り当てを持つ $\Leftrightarrow \tilde{S}$ 中に長さは n を持つ MUS がある



\tilde{T}_j の定義

任意節 $C_j = (l_a \vee l_b \vee l_c)$ に対して：

- l_i が変数 x_i のリテラル
- $v_i = \begin{cases} 0 & \text{ただし } l_i = x_i \\ 1 & \text{ただし } l_i \neq x_i \end{cases}$
- 以下の未決定文字列 \tilde{T}_j を定義

$$\tilde{T}_j = \left\{ \begin{array}{c} 0 \\ 1 \end{array} \right\}^{a-1} \{v_a\} \left\{ \begin{array}{c} 0 \\ 1 \end{array} \right\}^{b-a-1} \{v_b\} \left\{ \begin{array}{c} 0 \\ 1 \end{array} \right\}^{c-b-1} \{v_c\} \left\{ \begin{array}{c} 0 \\ 1 \end{array} \right\}^{n-c}.$$

例

- 変数: x_1, x_2, x_3, x_4
- $C_j = (x_1 \vee \neg x_2 \vee x_3)$
- $T_j = \{0\} \{1\} \{0\} \left\{ \begin{array}{c} 0 \\ 1 \end{array} \right\}$

\tilde{S} の定義

$$\tilde{S} = \tilde{T}_1 \{\$ \} \dots \{\$ \} \tilde{T}_m \{\$ \} \begin{Bmatrix} 0 \\ 1 \end{Bmatrix}^n \{\$ \} \begin{Bmatrix} 0 \\ 1 \end{Bmatrix}^{n-1} \{\$ \} \begin{Bmatrix} 0 \\ 1 \end{Bmatrix}^{n-1}.$$

- \tilde{T}_j は n の記号を持つ
- ⇒ \tilde{S} は $\mathcal{O}(nm)$ の記号を持つ
- $|\tilde{S}[i]| \leq 3 \Rightarrow \tilde{S}$ は F を $\mathcal{O}(nm)$ 領域で表現

例

$$F = (x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee \neg x_3 \vee x_4),$$

$$\tilde{S} = \{0\} \{1\} \{0\} \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} \{\$ \} \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} \{0\} \{1\} \{0\} \{\$ \} \begin{Bmatrix} 0 \\ 1 \end{Bmatrix}^4 \{\$ \} \begin{Bmatrix} 0 \\ 1 \end{Bmatrix}^3 \{\$ \} \begin{Bmatrix} 0 \\ 1 \end{Bmatrix}^3.$$

例の MUS

$$F = (x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee \neg x_3 \vee x_4),$$

$$\tilde{S} = \{0\}\{1\}\{0\}\begin{Bmatrix} 0 \\ 1 \end{Bmatrix}\{\$}\begin{Bmatrix} 0 \\ 1 \end{Bmatrix}\{0\}\{1\}\{0\}\{\$}\begin{Bmatrix} 0 \\ 1 \end{Bmatrix}^4\{\$}\begin{Bmatrix} 0 \\ 1 \end{Bmatrix}^3\{\$}\begin{Bmatrix} 0 \\ 1 \end{Bmatrix}^3.$$

1000 は MUS (特に、最短の MUS):

- 1000 は $(x_1 = 1, x_2 = x_3 = x_4 = 0)$ を表現
- すべての長さ 4 以下のバイナリー文字列は出現
- 長さ 3 のすべてのバイナリー部分文字列が \tilde{S} に 2 回出現
- 長さ 4 の部分文字列は $\$$ を含むと、 \tilde{S} に 2 回出現

帰着定理

$$\tilde{S} = \tilde{T}_1 \{\$ \} \dots \{\$ \} \tilde{T}_m \{\$ \} \begin{Bmatrix} 0 \\ 1 \end{Bmatrix}^n \{\$ \} \begin{Bmatrix} 0 \\ 1 \end{Bmatrix}^{n-1} \{\$ \} \begin{Bmatrix} 0 \\ 1 \end{Bmatrix}^{n-1}, \Sigma = \{0, 1, \$\}$$

定理

F が充足する割り当てが持つ $\Leftrightarrow \tilde{S}$ 中に長さは n を持つ MUS がある

- 長さが最大 $n-1$ のすべてのバイナリー部分文字列が \tilde{S} 中に 2 回出現
 \Rightarrow 一意単語は少なくとも長さ n を持つ
- 任意長さ n を持つ $\$$ を含む文字列が最後の 4 つの記号にも出現
 \Rightarrow 長さ n の MUS は $\$$ を含むことができない

帰着定理の証明: 必要条件 (\Rightarrow)

$$\tilde{S} = \tilde{T}_1 \{\$ \} \dots \{\$ \} \tilde{T}_m \{\$ \} \left\{ \begin{array}{c} 0 \\ 1 \end{array} \right\}^n \{\$ \} \left\{ \begin{array}{c} 0 \\ 1 \end{array} \right\}^{n-1} \{\$ \} \left\{ \begin{array}{c} 0 \\ 1 \end{array} \right\}^{n-1}, \Sigma = \{0, 1, \$\}$$

定理

F が充足する割り当てを持つ $\Leftrightarrow \tilde{S}$ 中に長さは n を持つ MUS がある

証明. 仮定: CNF を充足する割り当てがある.

- 割り当てをビット配列 $B[1..n]$ で表現する
- i 番目のビット $B[i]$ は x_i の真偽値を表現 (偽 = 0、真 = 1)
- B は部分文字列として出現
- B は \tilde{T}_i に不一致すると、 B の割り当てが節が C_i を満たす
 $\Rightarrow B$ は \tilde{S} の MUS

帰着定理の証明: 十分条件 (\Leftarrow)

$$\tilde{S} = \tilde{T}_1 \{ \$ \} \dots \{ \$ \} \tilde{T}_m \{ \$ \} \begin{Bmatrix} 0 \\ 1 \end{Bmatrix}^n \{ \$ \} \begin{Bmatrix} 0 \\ 1 \end{Bmatrix}^{n-1} \{ \$ \} \begin{Bmatrix} 0 \\ 1 \end{Bmatrix}^{n-1}, \Sigma = \{0, 1, \$\}$$

定理

F が充足する割り当てを持つ $\Leftrightarrow \tilde{S}$ 中に長さは n を持つ MUS がある

証明.

- ▮ 仮定: CNF を充足できない
- ▮ すべての長さ n のビット配列は \tilde{S} 中に 2 回出現
- ▮ \tilde{S} に長さ n を持つ一意単語はない

□

今後の展望

	単純な文字列	未決定文字列	GD-文字列	ED-文字列	2ED-文字列
MUS	線形	NP 完全			
MAW	線形	NP 完全			
anti-power	N	?	NP 完全		
LPF	線形	N	N	NP 完全	
不一致	線形	N	N	?	NP 完全

- 単純な文字列: $S[i] \in \Sigma$
- 未決定文字列: $\tilde{S}[i] \subset \Sigma$
- GD-文字列: $\forall i \exists k : \tilde{S}[i] \subset \Sigma^k$
- ED-文字列: $\tilde{S}[i] \subset \Sigma^*$
- 2ED-文字列: $\tilde{S}[i] \subset \text{ED-文字列}$

: 200 番目のアルゴリズム研究会の課題

文字列の一般化の階段

■ 単純な文字列: $T = \text{AACG}$

■ 未決定文字列: $\tilde{T} = A \begin{Bmatrix} A \\ C \end{Bmatrix} C \begin{Bmatrix} G \\ A \end{Bmatrix}$

■ generalized degenerate (GD)-文字列:

$$\tilde{T} = A \begin{Bmatrix} \text{AAC} \\ \text{CAT} \end{Bmatrix} C \begin{Bmatrix} G \\ T \end{Bmatrix}$$

■ elastic-degenerate (ED)-文字列:

$$\tilde{T} = A \begin{Bmatrix} \epsilon \\ \text{CAT} \end{Bmatrix} C \begin{Bmatrix} \text{GTCG} \\ T \end{Bmatrix}$$

■ 2ED-文字列: $\tilde{T} = \left\{ \begin{array}{l} T \begin{Bmatrix} \text{ACT} \\ C \end{Bmatrix} A \\ A \begin{Bmatrix} T \\ \text{AAG} \end{Bmatrix} \end{array} \right\} \begin{Bmatrix} \epsilon \\ \text{CAT} \end{Bmatrix} C \begin{Bmatrix} \text{GTCG} \\ T \end{Bmatrix}$