未決定文字列における欠如単語の検索の困難さ

Dominik Köppl and Jannik Olbrich

Abstract

単純な文字列の一般化の一つは未決定文字列である．未決定文字列の中に，各位置に一つの文字が格納されているだけではなく，複数の文字の選択肢が認められている．応用として，テキスト位置ですべての可能性を列挙することをモデル化する．実用的であるためには，パターン照合や未決定文字列の類似性を測定するなど，単純な文字列に対して知られている技術を未決定文字列に応用し，さまざまなクエリの種類に答えることに注目が集まっているが，遺伝子データの解析でよく利用されている欠如単語の検索にはまだ応用されていない．単純な文字列で，最短の欠如単語の検索を既に線形時間で答えられるが，未決定文字列上での計算は NP 困難であることを示す．

1 序論

不確実性はさまざまの応用において，完全な情報を必要とする古典的なアルゴリズムの適応を妨げる既知の問題である．本論は，すべての可能性を明示的に示すことで不確実性をモデル化する．このようなモデルは，遺伝子配列などの生物学的データを扱う際によく使われている．未決定文字列は，任意の位置に複数の代替文字を持つことができる文字列をモデル化する．未決定文字列の各要素は入力アルファベットの部分集合となり，これを**記号**と呼ぶ．記号は，DNA や RNA 配列における未決定の核酸を表現するために IUPAC 表記法 [22] を一般化したものである．図 1 は，未決定文字列の例を示している．

$$\widetilde{T} = \mathtt{A}\left\{\begin{matrix}\mathtt{A}\\ \mathtt{C}\end{matrix}\right\}\mathtt{C}\left\{\begin{matrix}\mathtt{G}\\ \mathtt{T}\end{matrix}\right\}, \qquad \mathcal{L}(\widetilde{T}) = \{\mathtt{AACG}, \mathtt{AACT}, \mathtt{ACCG}, \mathtt{ACCT}\}.$$

Figure 1: 未決定文字列 $\widetilde{T}$ の例．$\widetilde{T}$ の言語 $\mathcal{L}(\widetilde{T})$ は右に示されている．この言語は，未決定文字列によって表される文字列の集合である．1 つの文字 c のみを格納する記号は，$\{c\}$ または単に c として書かれる．

2 関連研究

関連研究は，未決定文字列と最小欠如単語、それぞれに特化した研究である．

2.1 未決定文字列

未決定文字列に関する研究の大部分は，パターン照合，構造的特性や規則性，必ずしも自己索引ではない索引からの未決定文字列の再構築，および 2 つの未決定文字列の比較に専念している．

パターン照合　未決定文字列に対しては，Boyer–Moore の適応版 [21]，ShiftAnd と Boyer–Moore–Sunday の組み合わせ [30]，および KMP を元した方法 [26] が提案されている．

構造的特性　未決定文字列の構造的特性については，カバーおよび/またはシードの計算 [4, 6]，Lyndon 分解の拡張 [15] が知られている．[23] は未決定文字列の新しい表現モデルを提案した．

再構築　データ構造からの未決定文字列の再構築も活発な研究分野である．再構築は，ボーダー配列，接尾辞配列，および LCP 配列から構築が可能である [25]．接頭辞配列に基づくグラフから [3]，または頂点がテキスト位置で辺が一致する文字を持つテキスト位置であるグラフからも再構築が行われている [20]．[11] は，配列が単純な文字列または未決定文字列の接頭辞配列であるかどうかの特徴を示した．最後に，[9] は，未決定文字列の接頭辞配列と無向グラフおよびボーダー配列との関係を研究した．

2.2 最小欠如単語

最小欠如単語（MAW: minimal absent word）は，潜在的な予防および治療的医療応用のためのバイオマーカーとして [29] によって導入された．MAW は，系統学 [10]，シーケンス比較 [13]，音楽コンテンツの情報検索 [12]，および円形バイナリ文字列の再構築 [28] において価値があることが示されている．MAW を計算するために，接尾辞配列 [7]，有向非巡回単語グラフ（DAWG: directed acyclic word graph） [18, 17]，または最大繰り返し [5] を使用するアルゴリズムが提案されている．並列処理 [8] や外部メモリ [19] で動作するアルゴリズムも提案されている．拡張として，連長圧縮文字列の MAW の計算 [2]，複数の文字列に共通する MAW の計算 [27]，または木上の MAW の計算 [16] がある．もう一つの拡張は，スライディングウィンドウ内での計算 [14, 24] であり，スライディングの動きに基づく解答集合の変化の数に関する境界が分析されている [1]．

3 予備知識

まず，文字列の基本概念を紹介し，その後，未決定文字列への一般化を形式化する．

文字列　アルファベットを Σ とする．Σ^* の要素は（単純な）**文字列**と呼ばれる．文字列 T が与えられたとき，T の i 番目の文字は $T[i]$ と表される（整数 $i \in [1..|T|]$ の場合），ここで $|T|$ は T の長さを表す．整数 i と j が $1 \leq i \leq j \leq |T|$ を満たすとき，位置 i から始まり位置 j で終わる T の部分文字列は $T[i..j]$ と表される．すなわち，$T[i..j] = T[i]T[i+1]\cdots T[j]$ である．T の部分文字列 P は，$P \neq T$ の場合に**真の部分文字列**と呼ばれる．

未決定文字列　単純な文字列の以下の拡張を研究する．そのために，アルファベット Σ の**文字**と，次の 3 種類の一般化のいずれかに属する文字列の**記号**を区別する．**未決定文字列**は，記号が Σ の空でない部分集合から引かれる文字列

$$\widetilde{T_j} = \left\{ \begin{matrix} 0 \\ 1 \end{matrix} \right\}^{a-1} \{ v_a \} \left\{ \begin{matrix} 0 \\ 1 \end{matrix} \right\}^{b-a-1} \{ v_b \} \left\{ \begin{matrix} 0 \\ 1 \end{matrix} \right\}^{c-b-1} \{ v_c \} \left\{ \begin{matrix} 0 \\ 1 \end{matrix} \right\}^{n-c}.$$

Figure 2: 定理 1 を証明するために定義した $\widetilde{T_j}$.

$\widetilde{S}[1..n]$ である．すなわち，$\emptyset \neq \widetilde{S}[i] \subset \Sigma$ である．$r = r(\widetilde{S}) = \max_i |\widetilde{S}[i]|$ は最大の記号のサイズを示す.

4 欠如単語の困難性

以下では，3-SAT の n 変数を持つ決定問題に対して，MAW の長さが最大 n であるかどうかを還元する．3-SAT の入力は，連言標準形（CNF: conjunctive normal form）の式 F である．すなわち，F は一連の節 C_i を連言（AND）で結合し，各節 C_i は 3 つのリテラルの選言である．n 個の変数 $x_1, \ldots, x_n$ が順序付けされていると仮定する．方針は，各節 C_i に対して C_i を満たさないすべての変数割り当てを指定することである．このような割り当ての集合は，未決定文字列 $\widetilde{S}$ にすべての変数を線形に書き込むことで表現できる．C_i で使用されていない変数は任意の値を取ることができる．構築により，これらの不足な割り当ての和集合をすべての可能な割り当てから引くことで，すべての充足する割り当てが得られる．長さ $n-1$ のすべての可能な部分文字列をカバーする未決定文字列を $\widetilde{S}$ に追加すると，MAW は少なくとも長さ n でなければならない．その後，$\widetilde{S}$ の最短の MAW が長さ n の部分文字列であり，その MAW が充足する割り当てを表すことを示す.

まず，単純な文字列に基づく MAW の形式的な定義から始める．

単純な文字列 T の**欠如単語** S とは，T の部分文字列ではない文字列を指する、ただし S の長さは 1 以上である．T の欠如単語 X は，X のすべての真の部分文字列が T に出現する場合，**最小欠如単語** (MAW) と呼ばれる．より一般的には，未決定文字列 $\widetilde{S}$ において，出現しない文字列を**欠如**していると言う．欠如文字列 X は，そのすべての真の部分文字列が $\widetilde{S}$ に出現する場合，$\widetilde{S}$ において**最小**である．以下では，特定の長さ以下の欠如単語を見つける決定問題が NP 困難であることを示す.

問題 1 (欠如単語問題)．未決定文字列 $\widetilde{S}$ と整数 k が与えられたとき，k-欠如単語問題は，長さが最大 k の欠如単語が $\widetilde{S}$ に存在するかどうかを決定することである．

n 個の変数と m 個の節を持つ 3-CNF $F = C_1 \wedge \cdots \wedge C_m$ が与えられたとする．アルファベット $\{0, 1, \$\}$ 上の未決定文字列 $\widetilde{S}$ を構築し，F が充足可能である場合に限り，長さが最大 n の MAW が $\widetilde{S}$ に存在することを示す．変数は $x_1, \ldots, x_n$ と仮定する．

節 $C_j = (\ell_a \vee \ell_b \vee \ell_c)$ のリテラルを ℓ_a, ℓ_b, ℓ_c とし，ℓ_i が変数 x_i のリテラルであるとする（$i \in a, b, c$）． 各 i に対して，ℓ_i が正の場合（$\ell_i = x_i$）は $v_i = 0$, それ以外の場合（$\ell_i = \neg x_i$）は $v_i = 1$ と設定し，v_i に基づいて図 2 の未決定文字列 $\widetilde{T_j}$ を構築する.

$$\widetilde{S} = \{0\}\{1\}\{0\}\left\{\begin{matrix}0\\1\end{matrix}\right\}\{\$\}\left\{\begin{matrix}0\\1\end{matrix}\right\}\{0\}\{1\}\{0\}\{\$\}\left\{\begin{matrix}0\\1\\\$\end{matrix}\right\}^{3}\{\$\}\left\{\begin{matrix}0\\1\end{matrix}\right\}^{3}.$$

Figure 3: 例 1 で利用される $\widetilde{S}$.

$\widetilde{S}$ を次のように定義する：

$$\widetilde{S} = \widetilde{T_1}\{\$\}\ldots\{\$\}\widetilde{T_m}\{\$\}\left\{\begin{matrix}0\\1\\\$\end{matrix}\right\}^{n-1}\{\$\}\left\{\begin{matrix}0\\1\end{matrix}\right\}^{n-1}.$$

各 $\widetilde{T_j}$ は n の記号から成り立つので，$\widetilde{S}$ は $\mathcal{O}(nm)$ の記号を持つ，かつ，$r(\widetilde{S}) = 3$ なので，$\widetilde{S}$ を $\mathcal{O}(nm)$ 領域で表現できる．$\widetilde{S}$ は，長さが最大 $n-1$ のアルファベット $\{0,1,\$\}$ のすべての文字列を部分文字列として含む．したがって，欠如単語は少なくとも長さ n でなければならない．任意の長さ n 以下の文字列が $ を含む場合，それは $\widetilde{S}$ の最後の 3 つの記号にも出現するため，長さが最大 n の欠如単語を見つけることは，長さ n のビット文字列を見つけることに帰着する．各ビット文字列 B は割り当てを表現する．つまり，$B[i] = 1 \Leftrightarrow x_i$ は真だとする．割り当てが節 C_i を満たさない場合，B は $\widetilde{T_i}$ に一致する．一方で，CNF に充足する割り当てが存在しない場合，アルファベット $\{0,1\}$ の長さ n のすべての部分文字列が $\widetilde{S}$ に存在しなければならない．したがって，$\widetilde{S}$ には長さが最大 n の欠如単語は存在しない．他方で，充足する割り当てが存在する場合，その割り当てを符号化するビット文字列が長さ n の欠如単語となる．

定理 1. 欠如単語 は $\sigma \geq 3$ および $r \geq 3$ の場合に NP 困難である．

例 1. 以下の 3-CNF に考える．

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

ただし $n = 4$ は変数 x_1, x_2, x_3, x_4 の個数を示す．CNF F から上記のように記述された操作手順で作った $\widetilde{S}$ が 図 3 に表示されている。文字列 1000 ($x_1 = 1, x_2 = x_3 = x_4 = 0$) は $\widetilde{S}$ に出現しない．しかし，その最長の真の接頭辞 100 と接尾辞 000 はそれぞれ $\widetilde{S}$ に出現する．したがって，1000 は MAW である．また，長さ 3 のすべての文字列が $\widetilde{S}$ に出現するため，1000 は最短の MAW でもある．

4.1 SAT 定式化

以下では，未決定文字列において指定された長さ x のユニークな部分文字列または欠如単語を計算するための SAT 定式化を提示する．SAT 定式化は，$x \in [1..n]$ に対する最適化目的を持つ MAX-SAT 定式化に変換できる．定式化のために，未決定文字列 $\widetilde{T}[1..n]$ の記号を文字のリストとして表現し，$\widetilde{T}[i]$ をリストとする．$\widetilde{T}[i][k]$ はリスト $\widetilde{T}[i]$ の k 番目の文字を表し，$|\widetilde{T}[i]|$ は $\widetilde{T}[i]$ に格納されている文

字の数を表す．ブール変数の真偽値を整数 1 と 0 として解釈し，和のような表現を使用できるようにする．解答を長さ x の文字列 $X[1..x] \in \Sigma^x$ としてモデル化する．長さ x は（決定問題として）与えられるか，最短長を得るための最適化対象となる．x を次のように n 個のブール変数 x_i でモデル化できる．

$$\sum_{i=1}^{n} x_i = 1. \qquad \text{(LENX)} \qquad \min i \in [1..n] : x_i = 1. \quad \text{(MINX)}$$

以下では，定義されていない変数は偽（設定されていない）と見なす．

X は，$\Sigma = 1, \ldots, \sigma$ から文字を選択する $x\sigma$ 個のブール変数を行列 $X'[1..x][1..\sigma]$ に配置することで表現できる．

$$\forall \ell \in [1..x] : \sum_{c=1}^{\sigma} X'[\ell, c] = 1. \qquad \text{(SETX)}$$

$\{\mathcal{O}(x),\, \mathcal{O}(\sigma)\}$

灰色の波括弧は，生成される節の数に対する漸近的な上限と，各節の最大サイズに対する漸近的な上限の 2 つを示している．ここでは，各 X の位置に対してサイズ σ の節を生成する．

X が欠如しているかどうかを確認するためには，すべてのテキスト位置 $t \in [1..n-x+1]$ において X の出現が始まらないことを確認する必要がある．言い換えれば，各 t に対して X の位置 $\ell \in [1..x]$ が存在し，$X[\ell]$ が $\widetilde{T}[t+\ell-1]$ に含まれないことを確認する．これを式として表現するために，これらの不一致をモデル化するブール変数 $M[k,t,\ell]$ の三次元グリッドを作成する．$X[\ell] \neq \widetilde{T}[t+\ell-1][k]$ の場合，$M[k,t,\ell]$ を真に設定する．ここで，$\widetilde{T}[t+\ell-1][k]$ はリスト $\widetilde{T}[t+\ell-1]$ の k 番目の文字を示す（存在する場合）．

$$\begin{aligned} \forall \ell \in [1..x], t \in [1..n-x+1], k \in [1..|\widetilde{T}[t+\ell-1]|] : \\ X[\ell] \neq \widetilde{T}[t+\ell-1][k] \implies M[k,t,\ell]. \end{aligned} \qquad \text{(M)}$$

$\{\mathcal{O}(xnr),\, \mathcal{O}(1)\}$

$r = r(\widetilde{T}) \leq \sigma$ はリスト $\widetilde{T}[i]$ が格納できる文字の最大数である．次に，$M[k,t,\ell]$ を $M'[t,\ell]$ に縮小し，$X[\ell]$ がリスト $\widetilde{T}[t+\ell-1]$ のすべての文字と一致しない場合にのみ $M'[t,\ell]$ を真に設定する．すなわち，

$$\begin{aligned} \forall \ell \in [1..x], t \in [1..n-x+1] : \\ \sum_{k=1}^{|\widetilde{T}[t+\ell-1]|} M[k,t,\ell] = |\widetilde{T}[t+\ell-1]| \implies M'[t,\ell]. \end{aligned} \qquad \text{(M')}$$

$\{\mathcal{O}(xn),\, \mathcal{O}(r)\}$

したがって，$M'[t,\ell]$ は $X[\ell]$ が $\widetilde{T}[t+\ell-1]$ に含まれない場合にのみ真である．最後に，X の位置 ℓ が存在し，$X[\ell]$ を $\widetilde{T}[t+\ell-1]$ のいずれの文字とも一致させることができないことを要求する．すなわち，

コード 1: MAW を計算する ASP の符号化． 最後の行（コマンド `show`）は出力用である．

```
1 1 { len(X) : X = 1..n } 1. %(LENX)
2 #minimize { X : len(X) }. %(MINX)
3 1 { x(L,C) : a(C) } 1 :- len(X), L=1..X. %(SETX)
4 m(K,T,L) :- t(K,T+L-1,C), x(L,D), C!=D, len(X), T=1..n-X+1. %(M)
5 m(T,L) :- r { m(K,T,L) : K=1..r }, len(X), L = 1..X, T=1..n-X+1.
    ↪ %(M')
6 :- { m(T,L) } 0, len(X), T=1..n-X+1. %(CONS)
7 #show x/2.
```

n	時間	x
10	0.02	2
100	169.62	4
150	566.88	4
200	1365.87	4
250	2615.42	5
260	2976.63	5
270	3301.57	5

Table 1: ランダムに生成された未決定文字列 $\widetilde{S}$ に対する最短 MAW の計算．$\sigma = 4$ および $r(\widetilde{S}) = 2$．時間は秒単位で，n は計算の入力として取られた $\widetilde{S}$ の接頭辞 $\widetilde{T}$ の長さである．

$$\forall t \in [1..n - x + 1] : \sum_{\ell=1}^{x} M'[t, \ell] \neq 0. \tag{CONS}$$

$\{\mathcal{O}(n), \mathcal{O}(x)\}$

もし Eq. (CONS) が成り立つならば，X は欠如単語でなければならない．合計で，$x\sigma + n$ の選択可能なブール変数がある．節の数は $\mathcal{O}(xnr)$ である．同じ上限が，すべての節のサイズの合計である CNF のサイズにも適用される．

リスト 1 は，上記の式のいずれかでコメントされた各行を含む解集合プログラミング（ASP: answer set programming）でのエンコーディングを示している．ランダムに生成された文字列に対する評価は 表 1 に示されている．評価は，Ubuntu 22.04 を搭載した Intel Xeon Gold 6330 CPU 上で，ASP の符号化を解釈するための clingo バージョン 5.7.1 を使用して実行された．

謝辞

本研究は JSPS 科研費 JP23H04378 の助成と山梨県若手研究者奨励事業費補助金 2291 の支援を受けたものである．

References

[1] Tooru Akagi, Yuki Kuhara, Takuya Mieno, Yuto Nakashima, Shunsuke Inenaga, Hideo Bannai, and Masayuki Takeda. Combinatorics of minimal absent words for a sliding window. *Theor. Comput. Sci.*, 927:109–119, 2022.

[2] Tooru Akagi, Kouta Okabe, Takuya Mieno, Yuto Nakashima, and Shunsuke Inenaga. Minimal absent words on run-length encoded strings. In *Proc. CPM*, volume 223 of *LIPIcs*, pages 27:1–27:17, 2022.

[3] Ali Alatabbi, M. Sohel Rahman, and William F. Smyth. Inferring an indeterminate string from a prefix graph. *J. Discrete Algorithms*, 32:6–13, 2015.

[4] Pavlos Antoniou, Maxime Crochemore, Costas S. Iliopoulos, Inuka Jayasekera, and Gad M. Landau. Conservative string covering of indeterminate strings. In *Proc. PSC*, pages 108–115, 2008.

[5] Aqil M. Azmi. On identifying minimal absent and unique words: An efficient scheme. *Cogn. Comput.*, 8(4):603–613, 2016.

[6] Md. Faizul Bari, Mohammad Sohel Rahman, and Rifat Shahriyar. Finding all covers of an indeterminate string in $o(n)$ time on average. In *Proc. PSC*, pages 263–271, 2009.

[7] Carl Barton, Alice Héliou, Laurent Mouchard, and Solon P. Pissis. Linear-time computation of minimal absent words using suffix array. *BMC Bioinform.*, 15:388, 2014.

[8] Carl Barton, Alice Héliou, Laurent Mouchard, and Solon P. Pissis. Parallelising the computation of minimal absent words. In *Proc. PPAM*, volume 9574 of *LNCS*, pages 243–253, 2015.

[9] Francine Blanchet-Sadri, Michelle Bodnar, and Benjamin De Winkle. New bounds and extended relations between prefix arrays, border arrays, undirected graphs, and indeterminate strings. *Theory Comput. Syst.*, 60(3):473–497, 2017.

[10] Supaporn Chairungsee and Maxime Crochemore. Using minimal absent words to build phylogeny. *Theor. Comput. Sci.*, 450:109–116, 2012.

[11] Manolis Christodoulakis, Patrick J. Ryan, William F. Smyth, and Shu Wang. Indeterminate strings, prefix arrays & undirected graphs. *Theor. Comput. Sci.*, 600:34–48, 2015.

[12] Tim Crawford, Golnaz Badkobeh, and David Lewis. Searching page-images of early music scanned with OMR: A scalable solution using minimal absent words. In *Proc. ISMIR*, pages 233–239, 2018.

[13] Maxime Crochemore, Gabriele Fici, Robert Mercas, and Solon P. Pissis. Linear-time sequence comparison using minimal absent words & applications. In *Proc. LATIN*, volume 9644 of *LNCS*, pages 334–346, 2016.

[14] Maxime Crochemore, Alice Héliou, Gregory Kucherov, Laurent Mouchard, Solon P. Pissis, and Yann Ramusat. Minimal absent words in a sliding window and applications to on-line pattern matching. In *Proc. FCT*, volume 10472 of *LNCS*, pages 164–176, 2017.

[15] Jacqueline W. Daykin and Bruce W. Watson. Indeterminate string factorizations and degenerate text transformations. *Math. Comput. Sci.*, 11(2):209–218, 2017.

[16] Gabriele Fici and Pawel Gawrychowski. Minimal absent words in rooted and unrooted trees. In *Proc. SPIRE*, volume 11811 of *LNCS*, pages 152–161, 2019.

[17] Yuta Fujishige, Takuya Takagi, and Diptarama Hendrian. Truncated DAWGs and their application to minimal absent word problem. In *Proc. SPIRE*, volume 11147 of *LNCS*, pages 139–152, 2018.

[18] Yuta Fujishige, Yuki Tsujimaru, Shunsuke Inenaga, Hideo Bannai, and Masayuki Takeda. Computing DAWGs and minimal absent words in linear time for integer alphabets. In *Proc. MFCS*, volume 58 of *LIPIcs*, pages 38:1–38:14, 2016.

[19] Alice Héliou, Solon P. Pissis, and Simon J. Puglisi. emMAW: computing minimal absent words in external memory. *Bioinform.*, 33(17):2746–2749, 2017.

[20] Joel Helling, Patrick J. Ryan, W. F. Smyth, and Michael Soltys. Constructing an indeterminate string from its associated graph. *Theor. Comput. Sci.*, 710:88–96, 2018.

[21] Jan Holub, William F. Smyth, and Shu Wang. Fast pattern-matching on indeterminate strings. *J. Discrete Algorithms*, 6(1):37–50, 2008.

[22] IUPAC-IUB Commission on Biochemical Nomenclature (CBN). Abbreviations and symbols for nucleic acids, polynucleotides and their constituents. *Journal of Molecular Biology*, 55(3):299–310, 1971.

[23] Felipe A. Louza, Neerja Mhaskar, and W. F. Smyth. A new approach to regular & indeterminate strings. *Theor. Comput. Sci.*, 854:105–115, 2021.

[24] Takuya Mieno, Yuki Kuhara, Tooru Akagi, Yuta Fujishige, Yuto Nakashima, Shunsuke Inenaga, Hideo Bannai, and Masayuki Takeda. Minimal unique substrings and minimal absent words in a sliding window. In *Proc. SOFSEM*, volume 12011 of *LNCS*, pages 148–160, 2020.

[25] Sumaiya Nazeen, M. Sohel Rahman, and Rezwana Reaz. Indeterminate string inference algorithms. *J. Discrete Algorithms*, 10:23–34, 2012.

[26] Mhaskar Neerja and William F. Smyth. Simple KMP pattern-matching on indeterminate strings. In *Proc. PSC*, pages 125–133, 2020.

[27] Kouta Okabe, Takuya Mieno, Yuto Nakashima, Shunsuke Inenaga, and Hideo Bannai. Linear-time computation of generalized minimal absent words for multiple strings. In *Proc. SPIRE*, volume 14240 of *LNCS*, pages 331–344, 2023.

[28] Takahiro Ota and Akiko Manada. A reconstruction of circular binary string using substrings and minimal absent words. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, 107(3):409–416, 2024.

[29] Armando J. Pinho, Paulo Jorge S. G. Ferreira, Sara P. Garcia, and João M. O. S. Rodrigues. On finding minimal absent words. *BMC Bioinform.*, 10, 2009.

[30] William F. Smyth and Shu Wang. An adaptive hybrid pattern-matching algorithm on indeterminate strings. *Int. J. Found. Comput. Sci.*, 20(6):985–1004, 2009.