New Bounds and Extended Relations Between Prefix Arrays, Border Arrays, Undirected Graphs, and Indeterminate Strings*

Francine Blanchet-Sadri[1], Michelle Bodnar[2], and Benjamin De Winkle[3]

1 **Department of Computer Science, University of North Carolina, Greensboro, USA**
blanchet@uncg.edu
2 **Department of Mathematics, University of California, San Diego, USA**
mbodnar@ucsd.edu
3 **Department of Mathematics, Tufts University, Medford, USA**
benjamin.de_winkle@tufts.edu

Abstract

We extend earlier works on the relation of prefix arrays of indeterminate strings to undirected graphs and border arrays. If integer array y is the prefix array for indeterminate string w, then we say w satisfies y. We use a graph theoretic approach to construct a string on a minimum alphabet size which satisfies a given prefix array. We relate the problem of finding a minimum alphabet size to finding edge clique covers of a particular graph, allowing us to bound the minimum alphabet size by $n + \sqrt{n}$ for indeterminate strings, where n is the size of the prefix array. When we restrict ourselves to prefix arrays for partial words, we bound the minimum alphabet size by $\lceil\sqrt{2n}\rceil$. Moreover, we show that this bound is tight up to a constant multiple by using Sidon sets. We also study the relationship between prefix arrays and border arrays. We show that the slowly-increasing property completely characterizes border arrays for indeterminate strings, whence there are exactly C_n distinct border arrays of size n for indeterminate strings (here C_n is the nth Catalan number). We also bound the number of prefix arrays for partial words of a given size using Stirling numbers of the second kind.

1998 ACM Subject Classification G.2.1 Combinatorics, G.2.2 Graph Theory

Keywords and phrases Indeterminate strings, Partial words, Prefix arrays, Border arrays, Undirected graphs

Digital Object Identifier 10.4230/LIPIcs.STACS.2014.162

1 Introduction

Strings are sequences of letters from a given alphabet. They have been extensively studied and several generalizations have been proposed in the literature which include *indeterminate strings* and *strings with don't cares* [10, 1, 3]. An indeterminate string allows positions to be subsets of cardinality greater than one of a given alphabet, while a string with don't cares allows positions to be the given alphabet. For example, $a\{a,b\}bb\{a,c\}$ is an indeterminate string of length 5 on the alphabet $\{a,b,c\}$ and $a\{a,b,c\}bb\{a,b,c\}$ is a string with don't cares

* This material is based upon work supported by the National Science Foundation under Grant No. DMS–1060775.

of same length on that alphabet. Strings with don't cares are also referred to as *partial words* and the don't care symbol is often represented by the $\diamond$ symbol, or *hole* symbol, which represents the alphabet. An alternative way of writing our example $a\{a,b,c\}bb\{a,b,c\}$ is $a\diamond bb\diamond$. Strings where each position is a singleton subset are referred to as *regular strings*.

The fundamental concept of *border array* has played an important role in pattern matching for over four decades [6, 15]. If non-empty strings u_1, u_2, v_1, v_2 exist such that $w = u_1v_1 = v_2u_2$ and u_1 matches u_2, denoted by $u_1 \approx u_2$, then string w has a *border* of length $|u_1| = |u_2|$. The border array β corresponding to a string w of length n is an integer array of same length such that for $j \in 0..n-1$, $\beta[j]$ is the length of the longest border of $w[0..j]$. For example, $a\{a,b\}bb\{a,c\}$ and $a\diamond bb\diamond$ give rise to the border arrays 01231 and 01234, respectively.

For a regular string w, any border of a border of w is also a border of w; thus w's border array gives all the borders of every prefix of w. This desirable property is lost however, when we consider indeterminate strings or partial words, due to the lack of the transitivity of $\approx$ (e.g., $a \approx \{a,b\}$ and $\{a,b\} \approx b$, but $a \not\approx b$ implying that $w = a\{a,b\}b$ has a border of length 2 having a border of length 1, but w has no border of length 1). Smyth and Wang [16] showed that for indeterminate strings, the concept of *prefix array* provides more information than the one of border array and specifies all the borders of every prefix. The prefix array y corresponding to a string w of length n is an integer array of same length such that for $j \in 0..n-1$, $y[j]$ is the length of the longest prefix of $w[j..n)$ that matches a prefix of w. For example, $a\{a,b\}bb\{a,c\}$ and $a\diamond bb\diamond$ give rise to the prefix arrays 53001 and 54001, respectively. Main and Lorentz [13] described the first algorithm for computing the prefix array of any given regular string as a routine in their well-known algorithm for finding all repetitions in a regular string, and Smyth and Wang [16] described an algorithm for efficiently computing the prefix array of any given indeterminate string.

The reverse problem of the one of designing an algorithm that computes the prefix array of any given string is the problem of designing an algorithm that tests if an integer array is the prefix array of some string and, if so, constructs such a string. Clément et al. [5] described an $O(n)$ time algorithm that tests if an integer array of size n is the prefix array of some regular string and, if so, constructs the lexicographically least string having it as a prefix array, the alphabet size of the string being bounded by $\log_2 n$. Recently, Christodoulakis et al. [4] described an algorithm for computing an indeterminate string corresponding to a given feasible prefix array. Such algorithmic characterizations of prefix arrays are not only interesting from a theoretical point of view, but also from a practical point of view, e.g., they help in the design of methods for randomly generating prefix arrays for software testing.

Christodoulakis et al. [4] established quite unexpected connections between indeterminate strings, prefix arrays, and undirected graphs. Among them, they proved the surprising result that every feasible array is the prefix array of some string. In this paper, we extend connections between indeterminate strings, prefix/border arrays, and undirected graphs, which yield combinatorial insights. In Section 2, we review some basics. In Section 3, we revisit the problem of constructing an indeterminate string on a minimal alphabet satisfying a given feasible prefix array y. We describe two methods: the first one relies on a graph $\mathcal{Q}_y$ built from y's associated prefix graph $\mathcal{P}_y$, while the second one examines induced subgraphs of $\mathcal{P}_y$. It turns out that the minimum alphabet size is the chromatic number of $\mathcal{Q}_y$ and is also the size of the smallest induced positive edge cover of $\mathcal{P}_y$. We bound the minimum alphabet size for an array of size n by $n + \sqrt{n}$ using results of Alon, Erdős et al., and Lovász on edge clique covers. In Section 4, we explore the relationship between prefix arrays and border arrays. In particular, we show that every slowly-increasing array is the border array for some indeterminate string, whence the number of border arrays of size n for indeterminate strings is the nth Catalan number. In Section 5, we restrict prefix arrays to partial words. We give a characterization of such prefix arrays y in terms of the prefix graph $\mathcal{P}_y$. Moreover, we give a method to construct a partial word on the smallest possible alphabet for a given prefix array. We bound the minimum alphabet size for an array of size n by $\left\lceil \sqrt{2n} \right\rceil$, this bound being tight (up to a constant multiple) using results of Erdős and Túran on Sidon sets. We also bound the number of prefix arrays of a given size valid for partial words using Stirling numbers of the second kind. Finally in Section 6, we conclude with some suggestions for future work.

2 Preliminaries

Throughout the paper, we use many graph theoretical concepts and constructions. We refer the reader to [11] for an introduction to these ideas.

A *string* w on alphabet A is a sequence of non-empty subsets of A, or may be empty. If A has cardinality μ, we also say that w is a string on μ letters. We call a 1-element subset of A a *regular* letter and larger subsets *indeterminate* letters. A string of all regular letters is called a *regular string* (also referred to as a *word*), and a string which contains at least one indeterminate letter is called an *indeterminate string*. A *hole*, denoted by $\diamond$, is an indeterminate letter which consists of the full alphabet, A. A *partial word* is a string which consists of only regular letters and holes. We denote the length of string w by $|w|$.

Two non-empty subsets of A, α and α', *match* if they have non-empty intersection. We denote this by $\alpha \approx \alpha'$. Similarly, two strings w and w' *match* if $|w| = |w'|$ and $w[i] \approx w'[i]$ for each $i \in 0..|w| - 1$. As before, this is denoted by $w \approx w'$.

An integer array y of size n such that $y[0] = n$ and for every $i \in 1..n-1$, $0 \leq y[i] \leq n - i$, is called *feasible*. The *prefix array* of a string w of length n is an array of integers y such that $y[j]$ is the length of the longest prefix of $w[j..n)$ that matches a prefix of w. Note that $y[0]$ is the size of y for any prefix array y. If y is the prefix array of some regular string, then y is called *regular*. If y is the prefix array of some partial word, then y is called *valid for partial words*. If y is the prefix array of a string w, then w *satisfies* y.

▶ **Lemma 2.1.** *[4] An integer array y of size n is the prefix array of a string w of length n if and only if for each position $i \in 0..n-1$, the following two conditions hold: (1) $w[0..y[i]) \approx w[i..i + y[i])$ and (2) if $i + y[i] < n$, then $w[y[i]] \not\approx w[i + y[i]]$.*

The most important graph construction that we use is that of the *prefix graph*, which is introduced in [4]. The prefix graph of a prefix array, y, of size n is denoted by $\mathcal{P}_y$ and is constructed as follows. Its vertex set, $V(\mathcal{P}_y)$, is $[0..n)$. Its edge set consists of two types of edges. Let $i \in 1..n-1$. For $j \in 0..y[i] - 1$, $\{j, i + j\}$ is a *positive edge*, while for $i + y[i] < n$, $\{y[i], i + y[i]\}$ is a *negative edge* (refer to Lemma 2.1).

Let $E^+(\mathcal{P}_y)$ be the set of positive edges of $\mathcal{P}_y$ and $E^-(\mathcal{P}_y)$ be the set of negative edges of $\mathcal{P}_y$ (note we may write just E^+ or E^-, respectively, when $\mathcal{P}_y$ is clear from context). We write $\mathcal{P}_y^+ = (V(\mathcal{P}_y), E^+(\mathcal{P}_y))$ (i.e., the graph with the same vertex set, but with only the positive edges) and $\mathcal{P}_y^- = (V(\mathcal{P}_y), E^-(\mathcal{P}_y))$ (same vertex set, only the negative edges). A string w *satisfies* negative edge $\{i, j\}$ if $w[i] \not\approx w[j]$ and w *violates* $\{i, j\}$ if $w[i] \approx w[j]$. Similarly, w *satisfies* positive edge $\{i, j\}$ if $w[i] \approx w[j]$ and w *violates* $\{i, j\}$ if $w[i] \not\approx w[j]$. The graph $\mathcal{P}_y$ encodes all the information of the prefix array y, that is to say, that string w satisfies y if and only if w satisfies all positive and negative edges of $\mathcal{P}_y$. Figure 1 shows an example.

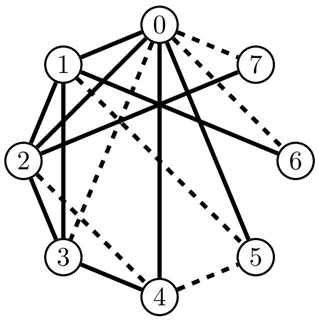

Figure 1 Prefix graph $\mathcal{P}_y$ for $y = 84201300$. Solid lines indicate positive edges and dashed lines indicate negative edges.

▶ **Lemma 2.2.** *Let y be a feasible prefix array and w be an indeterminate string satisfying y. If in $\mathcal{P}_y$, j_0 and j_k are joined by a negative edge and $j_0, j_1, \ldots, j_k$ is a path on only positive edges, then there exists some $i \in 0..k$ such that $w[j_i]$ is an indeterminate letter.*

3 Constructing Indeterminate Strings for Prefix Arrays

Returning to Figure 1, $\{a,c,e\}\{a,b\}\{a,b\}\{b,d\}\{c,d\}ebb$ satisfies the prefix array $y = 84201300$. It is constructed on an alphabet of five letters a, b, c, d, e. Is the alphabet size minimal? The answer is no since $\{a,b\}\{a,c\}\{b,c\}\{c,d\}\{a,d\}bcc$ also satisfies y but is constructed using only the four letters a, b, c, d. In this section, we describe two methods for constructing indeterminate strings on a minimum alphabet size that satisfy a given prefix array. For ease of notation, if y is a feasible prefix array, let $\mu(y)$ denote the minimum alphabet size for an indeterminate string that satisfies y.

Let us describe our first method. Let V^+ be the set of vertices of $\mathcal{P}_y$ which are incident to a positive edge. We construct a new graph $\mathcal{Q}$ from $\mathcal{P}_y$ as follows: $V(\mathcal{Q}) = E^+ \cup \{\{i,i\} \mid i \in V \setminus V^+\}$, and $\{\{i_1,j_1\},\{i_2,j_2\}\} \in E(\mathcal{Q})$ if and only if there exists some $\{r,s\} \in E^-$ such that $r \in \{i_1,j_1\}$ and $s \in \{i_2,j_2\}$. Since a prefix array y defines a unique $\mathcal{P}_y$, which in turn defines a unique $\mathcal{Q}$, we can call this graph $\mathcal{Q}_y$. We show how proper colorings of $\mathcal{Q}_y$ and indeterminate strings satisfying y are related.

Figure 2 gives an example. Since $\mathcal{Q}_y$ has chromatic number 2, associate a with vertices $\{0,2\}$ and $\{0,3\}$, and b with vertices $\{1,3\}$, $\{1,4\}$, and $\{5,5\}$. By assigning letters to each vertex in $\mathcal{P}_y$ corresponding to the letters associated with its incident positive edges, we obtain the indeterminate $\{a\}\{b\}\{a\}\{a,b\}\{b\}\{b\}$ which satisfies 602200.

▶ **Theorem 3.1.** *Let μ be the minimum alphabet size for a string satisfying a given feasible prefix array y. Then $\chi(\mathcal{Q}_y) = \mu$, where $\chi(\mathcal{Q}_y)$ denotes the chromatic number of $\mathcal{Q}_y$.*

Proof. Let $\mathcal{P} = \mathcal{P}_y$ and $\mathcal{Q} = \mathcal{Q}_y$. Suppose w is a string on μ letters which satisfies y, and associate a distinct color to each letter. For each edge $\{i,j\} \in E^+ \cup \{\{i,i\} \mid i \in V \setminus V^+\}$, color the vertex $\{i,j\}$ in $\mathcal{Q}$ with the color associated to the first element in $w[i] \cap w[j]$. The intersection is always non-empty because positive edges and loops represent matchings. Now suppose there is an edge connecting the vertices $\{i,j\}$ and $\{r,s\}$ in $\mathcal{Q}$. Then there is a negative edge in $\mathcal{P}$ connecting one endpoint of $\{i,j\}$ to an endpoint of $\{r,s\}$. Without loss of generality, assume $\{i,r\} \in E^-$. This implies $w[i] \cap w[r] = \emptyset$, so $\{i,j\}$ and $\{r,s\}$ must have different colors. Thus, we have obtained a proper coloring of $\mathcal{Q}$ with μ colors, so $\chi(\mathcal{Q}) \leq \mu$.

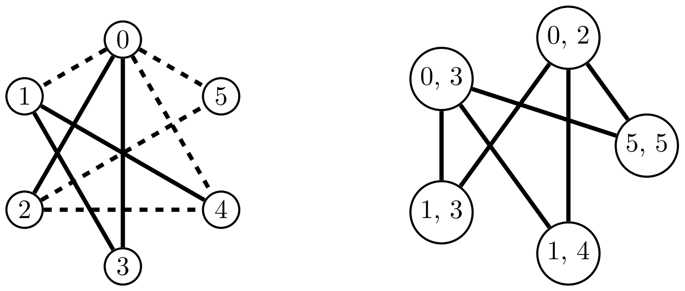

Figure 2 $\mathcal{P}_y$ (left) and $\mathcal{Q}_y$ (right) for the prefix array $y = 602200$, where solid lines indicate positive edges and dashed lines indicate negative edges.

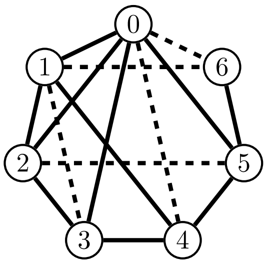

Figure 3 $\mathcal{P}_y$ for $y = 7612010$, where solid lines indicate positive edges and dashed lines indicate negative edges. One IPEC for $\mathcal{P}_y$ is given by the sets $V_0 = \{0, 1, 5\}$, $V_1 = \{0, 2, 3\}$, $V_2 = \{1, 2, 4\}$, and $V_3 = \{3, 4, 5, 6\}$. The construction in Theorem 3.2 gives the indeterminate string $w = \{a, b\}\{a, c\}\{b, c\}\{b, d\}\{c, d\}\{a, d\}d$, which satisfies y.

Now suppose we are given a proper coloring of $\mathcal{Q}$ using $\chi(\mathcal{Q})$ colors. We may think of each color as a letter and construct an indeterminate string w by assigning to $w[i]$ the color of each $\{k, j\} \in V(Q)$ such that $i = k$ or $i = j$. Let $\{i, j\}$ be any positive edge of $\mathcal{P}$. Then $w[i]$ and $w[j]$ both contain the color given to $\{i, j\}$, so it is satisfied. It remains to show that each negative edge is also satisfied. Suppose $\{i, j\} \in E^-$. Then $w[i] = \{c \mid c \text{ is the color on some } \{i, r\} \in E^+\}$, and $w[j] = \{c \mid c \text{ is the color on some } \{j, s\} \in E^+\}$. Moreover, if $\{i, r\}, \{j, s\} \in E^+$, then they are connected by an edge in $\mathcal{Q}_y$, so they have a different color. Hence $w[i] \cap w[j] = \emptyset$, so $\{i, j\}$ is satisfied. Therefore, w satisfies y. Moreover, w uses at most $\chi(\mathcal{Q})$ letters, which implies $\mu \leq \chi(\mathcal{Q})$. ◀

Let us describe our second method. Suppose indeterminate string w on alphabet $A = \{a_0, a_1, \ldots, a_{\mu-1}\}$ satisfies prefix array y, and define $V_i = \{j \mid a_i \in w[j]\}$. Notice that the subgraph of $\mathcal{P}_y$ induced by V_i contains no negative edges. Moreover, each positive edge is in the subgraph induced by some V_i. This observation motivates the following definitions.

A subgraph of $\mathcal{P}_y$ is *negative-free* if it does not contain any negative edges. We use the notation $\mathcal{P}_y[V_i]$ to denote the subgraph of $\mathcal{P}_y$ induced by V_i. A set $\{V_0, V_1, \ldots, V_k\}$, where $V_i \subset V(\mathcal{P}_y)$, is an *induced positive edge cover* (IPEC) of $\mathcal{P}_y$ if $\mathcal{P}_y[V_i]$ is negative-free for all $i \in 0..k$, each positive edge of $\mathcal{P}_y$ is in some $\mathcal{P}_y[V_i]$, and each vertex of $\mathcal{P}_y$ is in some $\mathcal{P}_y[V_i]$. Figure 3 gives an example of an IPEC.

▶ **Theorem 3.2.** *Let y be a feasible prefix array. The minimum alphabet size for an indeterminate string satisfying y is exactly the size of the smallest IPEC of $\mathcal{P}_y$.*

Proof. Let μ be the minimum alphabet size for an indeterminate string satisfying y, and let σ be the size of the smallest IPEC of $\mathcal{P}_y$. Suppose w is an indeterminate string that satisfies y

on the alphabet $\{a_0, a_1, \ldots, a_{\mu-1}\}$. Let V_i be as defined above. We claim $\{V_0, V_1, \ldots, V_{\mu-1}\}$ is an IPEC of $\mathcal{P}_y$. It is clear that each vertex is in some V_i, because each position of w is non-empty. Suppose $\{i, j\}$ is a negative edge of $\mathcal{P}_y$. Since w satisfies y, it must satisfy $\{i, j\}$, so $w[i] \cap w[j] = \emptyset$. Hence there is no V_k which contains both i and j, which implies $\{i, j\}$ is not in $\mathcal{P}_y[V_k]$ for any k. This holds for any negative edge, so each $\mathcal{P}_y[V_k]$ is negative-free. Now suppose $\{i, j\}$ is a positive edge of $\mathcal{P}_y$. As before, w must satisfy $\{i, j\}$, which implies there exists some $a_k \in w[i] \cap w[j]$. Then $i, j \in V_k$, so $\{i, j\}$ is in the subgraph induced by V_k. This proves our claim and shows $\sigma \leq \mu$.

Now suppose $\mathcal{C} = \{V_0, V_1, \ldots, V_{\sigma-1}\}$ is an IPEC of $\mathcal{P}_y$. Let $\{a_0, a_1, \ldots, a_{\sigma-1}\}$ be a collection of distinct letters and construct an indeterminate string w by setting $w[i] = \{a_j \mid i \in V_j\}$. Since each i is in some V_j, $w[i]$ is non-empty for all i. We claim w satisfies y. Suppose i and j are connected by a negative edge in $\mathcal{P}_y$. Then there is no $V_k \in \mathcal{C}$ which contains both i and j, so by construction, $w[i] \cap w[j] = \emptyset$. Hence all negative edges of $\mathcal{P}_y$ are satisfied. Now suppose i and j are connected by a positive edge. This edge is in $\mathcal{P}_y[V_k]$ for some $V_k \in \mathcal{C}$, which implies $a_k \in w[i] \cap w[j]$, satisfying the positive edge. Thus all edges of $\mathcal{P}_y$ are satisfied, which proves our claim. Note w uses σ letters, so $\mu \leq \sigma$. Therefore, $\mu = \sigma$. ◀

We can use this construction to bound $\mu(y)$, but first we introduce a few concepts. Given a graph G, an *edge clique cover* of G is a set of cliques in G such that each edge of G is in at least one of these cliques. The *edge clique cover number* of G is the size of the smallest edge clique cover of G, which we denote by $\theta(G)$. We denote the complement of G by $\overline{G}$, i.e., the graph defined by $V(\overline{G}) = V(G)$ and two vertices of $\overline{G}$ are joined by an edge if and only if they are not joined by an edge in G.

▶ **Lemma 3.3.** *Let y be a feasible prefix array of size n such that $y \neq n00\cdots0$ and $y \neq n(n-2)(n-3)\cdots0$. Then $\mu(y) \leq \theta\left(\overline{\mathcal{P}_y^-}\right)$.*

Edge clique covers have been well studied, and we can use results on them to bound $\mu(y)$. Specifically, we use the following results.

▶ **Theorem 3.4** ([8, Theorem 2]). *Let G be a graph on n vertices. Then $\theta(G) \leq \lfloor \frac{n^2}{4} \rfloor$.*

▶ **Theorem 3.5** ([12, Theorem 5]). *Let G be a graph on n vertices with $m \geq \lfloor \frac{n^2}{4} \rfloor$ edges. Further, set $k = \binom{n}{2} - m$ (i.e., k is the number of edges in $\overline{G}$) and let t be the greatest integer such that $t^2 - t \leq k$. Then $\theta(G) \leq k + t$.*

▶ **Theorem 3.6** ([2, Theorem 1.4]). *Let G be a graph on n vertices with maximum degree d. Then $\theta(\overline{G}) \leq 2e^2(d+1)^2 \ln n$, where e is the base of the natural logarithm.*

Now we can state a bound on $\mu(y)$.

▶ **Theorem 3.7.** *Let y be a feasible prefix array of size n, and let r be the number of negative edges in $\mathcal{P}_y$. Then $\mu(y) \leq \min\{r + \sqrt{r} + 1, 2e^2(d+1)^2 \ln n\}$, where e is the base of the natural logarithm and d is the maximum degree of a vertex in $\mathcal{P}_y^-$.*

Proof. We use Lemma 3.3, so we first check the cases $y = n00\cdots0$ and $y = n(n-2)(n-3)\cdots0$. Suppose $y = n00\cdots0$. Notice that y is satisfied by $abb\cdots b$. Similarly, if $y = n(n-2)(n-3)\cdots0$, then y is satisfied by $aa\cdots ab$. In both of these cases, any string satisfying y must have at least two letters. Moreover, in both of these cases $r = n-1$, hence $\mu(y) = 2 \leq \min\{r + \sqrt{r} + 1, 2e^2(r+1)^2 \ln n\}$ and the result holds.

Thus, by Lemma 3.3, we may assume that $\mu(y) \leq \theta\left(\overline{\mathcal{P}_y^-}\right)$. Let $\theta = \theta\left(\overline{\mathcal{P}_y^-}\right)$. It follows from Theorem 3.6 that $\theta \leq 2e^2(d+1)^2 \ln n$. To get the $r + \sqrt{r} + 1$ bound, we apply

Theorem 3.5, but this requires that $\overline{\mathcal{P}_y^-}$ has at least $\lfloor \frac{n^2}{4} \rfloor$ edges. Note that $\overline{\mathcal{P}_y^-}$ has $\binom{n}{2} - r$ edges. Since $r < n$, the number of edges in $\overline{\mathcal{P}_y^-}$ is at least $\binom{n}{2} - (n-1) = \frac{n^2-3n+2}{2}$.

Define the function $f(n) = \frac{n^2-3n+2}{2} - \frac{n^2}{4} = \frac{n^2-6n+4}{4}$. Whenever f is positive, $\overline{\mathcal{P}_y^-}$ has at least $\frac{n^2}{4}$ edges. Note that $f(6) = 1 > 0$, and $f'(n) = \frac{n-3}{2}$ is positive for any $n > 3$. Hence f is positive for all $n \geq 6$. Note that the complement of $\overline{\mathcal{P}_y^-}$ is $\mathcal{P}_y^-$, which has r edges. Hence, by Theorem 3.5, if $n \geq 6$ and t is the greatest integer such that $t^2 - t \leq r$, then $\theta \leq r + t$. Notice that $t < \sqrt{r} + 1$, so $\theta < r + \sqrt{r} + 1$. This just leaves the cases where $n < 6$. Note that in these cases, $r + \sqrt{r} + 1 < 2e^2(d+1)^2 \ln n$, because $r - 1 < n$. Moreover, we can easily check that for each combination of n and r, either $\overline{\mathcal{P}_y^-}$ has at least $\frac{n^2}{4}$ edges, or $\frac{n^2}{4} < r + \sqrt{r} + 1$. In the former case, we can apply Theorem 3.5 to give $\theta < r + \sqrt{r} + 1$. In the latter case, Theorem 3.4 gives us $\theta \leq \frac{n^2}{4}$, implying $\theta < r + \sqrt{r} + 1$. ◀

Since $r \leq n - 1$, we get the following corollary.

▶ **Corollary 3.8.** *Let y be a feasible prefix array of size n. Then $\mu(y) \leq n + \sqrt{n}$.*

4 Connecting Prefix Arrays and Border Arrays

An indeterminate string, w, of length n has a *border* of length $\ell \in 0..n-1$ if $w[0..\ell) \approx w[n-\ell..n)$. The *border array*, $\beta[0..n)$, of an indeterminate string w is an integer array such that $\beta[0] = 0$ and for $i > 0$, $\beta[i]$ is the length of the longest border of $w[0..i]$. For example, $a\{a,b\}\{a,b\}bac$ has border array $\beta = 012330$. A border array β is *feasible* if there exists an indeterminate string such that β is its border array. A prefix array y *satisfies* a border array β if all strings with prefix array y also have border array β.

▶ **Theorem 4.1.** *Let β be a border array of size n. Then a prefix array y satisfies β if and only if the following two conditions hold: (1) $\beta[j] \leq y[j - \beta[j] + 1]$ for all $j \in 0..n-1$, and (2) $y[i] \leq j - i$ for all $i \leq j - \beta[j]$.*

Proof. Let y be a prefix array that satisfies β and w be any string with prefix array y. Since $w[0..j]$ has a maximal border of length $\beta[j]$, we have $w[0..\beta[j]) \approx w[j - \beta[j] + 1..j]$. Since $y[j - \beta[j] + 1]$ gives the length of the longest prefix of w that matches a prefix of $w[j - \beta[j] + 1..|w|)$, we have $y[j - \beta[j] + 1] \geq \beta[j]$ which gives (1). Now let $i \leq j - \beta[j]$ and $y[i] = j - i + r$ for some r. Then $w[0..j-i+r) \approx w[i..j+r)$. If $r > 0$ then $w[0..j-i] \approx w[i..j]$, so $w[0..j]$ has a border of length at least $j - i + 1$. However, since $\beta[j] \leq j - i$, $w[0..j]$ has a maximal border of length $j - i$, so $r \leq 0$. Therefore $y[i] \leq j - i$, so (2) must hold.

For the reverse implication, let y be a prefix array satisfying (1) and (2), and w be a string with prefix array y. We show that w must have border array β. Let $j \in 0..n-1$ be arbitrary. By (1) the factor of w of length $\beta[j]$ starting at position $j - \beta[j] + 1$ matches $w[0..\beta[j])$, so $w[0..j]$ has a border of length $\beta[j]$. To see that this border is maximal, suppose there exists a border of $w[0..j]$ with length $\beta[j] + r$ for some $r \geq 1$. Then $y[j - \beta[j] - r + 1] \geq \beta[j] + r$ which contradicts (2). Thus, y satisfies β. ◀

This characterization of prefix arrays which satisfy a given border array leads to a natural question: Given a border array, what degree of freedom do we have in creating a prefix array which satisfies it? The following corollary answers this question, but requires one property of border arrays referred to as the *slowly-increasing property*: for any border array $\beta = \beta[0..n)$ feasible by some indeterminate string w, $\beta[j+1] \leq \beta[j] + 1$ for all $j \in 0..n-2$.

▶ **Corollary 4.2.** *Let y be a prefix array that satisfies border array β. Then β completely determines $y[i]$, where $i > 0$ if and only if $\beta[i] = 0$ or there exists some j such that $i = j - \beta[j] + 1$. Moreover, if $i = j - \beta[j] + 1$ for some j, then $y[i] = \beta[j]$ for the largest j with this property.*

The slowly-increasing property characterizes border arrays of indeterminate strings.

▶ **Theorem 4.3.** *Every slowly-increasing array is the border array for some indeterminate string.*

Proof. Since every feasible prefix array is satisfied by some indeterminate string, it suffices to show that the set of prefix arrays which satisfy any slowly-increasing array is non-empty and feasible. We use the conditions given in Theorem 4.1. Recall that if a prefix array y is feasible, $y[i] \leq n - i$ for all $i \in 0..n-1$. Let β be a slowly-increasing array. For $j \in 0..n-1$ we have that $\beta[j] \leq n - (j - \beta[j] + 1)$, so satisfying (1) never forces y to be infeasible.

Now we check that (1) and (2) never force an empty set of possible assignments to a position of a prefix array. Suppose to the contrary that there exist such positions j_1 and j_2. Condition (1) gives $\beta[j_1] \leq y[j_1 - \beta[j_1] + 1]$ and Condition (2) gives $y[j_1 - \beta[j_1] + 1] \leq j_2 - j_1 + \beta[j_1] - 1$ for $j_1 - \beta[j_1] + 1 \leq j_2 - \beta[j_2]$. Since we assume no y can satisfy both of these, $j_2 - (j_1 - \beta[j_1] + 1) < \beta[j_1]$, or equivalently $j_2 \leq j_1$. This means that for $i = j_1 - \beta[j_1] + 1$, there is no possible assignment for $y[i]$ and thus no prefix array satisfying β. Condition (2) requires that $i \leq j_2 - \beta[j_2]$, so in this case we have $j_1 - \beta[j_1] + 1 \leq j_2 - \beta[j_2]$. However, rearranging this gives $\beta[j_2] + j_1 - j_2 + 1 \leq \beta[j_1]$. Since $j_2 \leq j_1$, this violates the slowly-increasing property of β, a contradiction. Thus, no such j_1 and j_2 can exist and we conclude that there exists a non-empty set of assignments to $y[i]$ for each i, so β is feasible. ◀

The following theorem counts the total number of slowly-increasing arrays of a given size.

▶ **Theorem 4.4.** *For all $n \geq 1$, the number of slowly-increasing arrays of size n is $C_n = \frac{1}{n+1}\binom{2n}{n}$, the nth Catalan number.*

Proof. This follows easily using basic enumerative combinatorics (see, e.g., [18, 17]). ◀

This gives us the following corollary.

▶ **Corollary 4.5.** *The number of distinct border arrays of size n for indeterminate strings is exactly the n^{th} Catalan number, $C_n = \frac{1}{n+1}\binom{2n}{n}$.*

5 Restricting Prefix Arrays to Partial Words

Since a hole matches any other letter, the following lemma follows directly from Lemma 2.1.

▶ **Lemma 5.1.** *Let y be the prefix array for some partial word w. Then $w[i]$ can be a hole if and only if there does not exist $j \in 0..n-1$ such that either (1) $y[j] = i$ and $i + j < n$ or (2) $j + y[j] = i$ if and only if i is not incident to any negative edges in $\mathcal{P}_y$.*

The next theorem gives a characterization of prefix arrays which can be satisfied by a partial word. It is similar to a characterization of regular prefix arrays given by [4, Lemma 10].

▶ **Theorem 5.2.** *Let y be a feasible prefix array and let V^- be the set of vertices in $\mathcal{P}_y$ which are incident to a negative edge. The following are equivalent: (1) the array y is the prefix array for some partial word, (2) every cycle in $\mathcal{P}_y$ which contains exactly one negative edge has at least one vertex which is not in V^-, and (3) every negative edge of $\mathcal{P}_y$ connects vertices in two different connected components of $\mathcal{P}_y^+[V^-]$.*

Proof. First we prove (1) implies (2) by contraposition. Assume $\{i, j\}$ is a negative edge in $\mathcal{P}_y$ which connects vertices i and j, where i and j lie in the same connected component of $\mathcal{P}_y^+[V^-]$. Then there exists a path, p, in $\mathcal{P}_y^+[V^-]$ from i to j. Further suppose w is an indeterminate string which satisfies y. Since each edge in path p is a positive edge, Lemma 2.2 implies there exists some k such that k is in this path and $w[k]$ is an indeterminate letter. However, $k \in V^-$, so by Lemma 5.1, $w[k]$ cannot be a hole. Since holes are the only indeterminate letters allowed in a partial word, w is not a partial word.

Next we prove (2) implies (3), again by contraposition. Suppose $\{i, j\}$ is a negative edge such that i and j are in the same connected component of $\mathcal{P}_y^+[V^-]$. Then there exists a path from j to i which lies in $\mathcal{P}_y^+[V^-]$. Note that all the edges in this path are positive, and all the vertices are in V^-. Then concatenating $\{i, j\}$ to this path creates a cycle in $\mathcal{P}_y$ which contains exactly one negative edge, but whose vertices all are in V^-.

Finally, we prove (3) implies (1). Assume every negative edge of $\mathcal{P}_y$ connects vertices in two different connected components of $\mathcal{P}_y^+[V^-]$. Let $C_0, C_1, \ldots, C_{\ell-1}$ be the connected components of $\mathcal{P}_y^+[V^-]$, and construct a partial word w on the alphabet $\{a_0, a_1, \ldots, a_{\ell-1}\}$ by setting $w[i] = a_j$ if $i \in C_j$ and $w[i] = \diamond$ if $i \notin V^-$. Note that this construction does indeed assign one letter (or hole) to each position of w, and we claim that w satisfies y.

Suppose $\{i, j\}$ is a negative edge in $\mathcal{P}_y$. By assumption, i and j are in different connected components of $\mathcal{P}_y^+[V^-]$, so $w[i] \not\approx w[j]$. Hence this edge is satisfied. Now suppose $\{i, j\}$ is a positive edge in $\mathcal{P}_y$. If $i \notin V^-$, then $w[i] = \diamond$, which implies $w[i] \approx w[j]$ and w satisfies $\{i, j\}$. A symmetric argument holds for $j \notin V^-$. Now assume $i, j \in V^-$. This implies i and j are in the same connected component of $\mathcal{P}_y^+[V^-]$, so $w[i] = w[j]$, satisfying $\{i, j\}$. Therefore w satisfies all edges of $\mathcal{P}_y$, proving that y is the prefix array for the partial word w. ◀

Based upon the construction given in the above proof, we define $V^-(\mathcal{P}_y)$ (or just V^- if $\mathcal{P}_y$ is clear from context) to be the set of vertices in $\mathcal{P}_y$ which are incident to a negative edge. Further, construct the graph $\mathcal{C}_y$ as follows: make one vertex in $\mathcal{C}_y$ for each connected component in $\mathcal{P}_y^+[V^-]$ and join two vertices in $\mathcal{C}_y$ if and only if there exists a negative edge in $\mathcal{P}_y$ between their corresponding components. Finally, if y is the prefix array for some partial word, $\mu_\diamond(y)$ will denote the minimum alphabet size for a partial word satisfying y.

▶ **Theorem 5.3.** *Let y be the prefix array for some partial word. Then $\mu_\diamond(y) = \chi(\mathcal{C}_y)$.*

Proof. Let $C_1, C_2, \ldots, C_\ell$ be the connected components of $\mathcal{P}_y^+[V^-]$ and assume we have a valid coloring of $\mathcal{C}_y$. We treat these colors as letters and construct w using a similar method to the one given in the proof of Theorem 5.2. We let $w[i]$ be the color of C_k if $i \in C_k$ and let $w[i] = \diamond$ otherwise. The proof that w satisfies y follows exactly the last part of the proof of Theorem 5.2. Hence $\mu_\diamond(y) \leq \chi(\mathcal{C}_y)$.

Let w be a partial word satisfying y on an alphabet, A, of minimum size. Suppose i and i' are in the same connected component of $\mathcal{P}_y^+[V^-]$. Then there exists a path, $i = j_1, j_2, \ldots, j_k = i'$, of positive edges from i to i' such that each vertex in this path is in V^-. By Lemma 5.1, $w[j_\ell]$ must be a regular letter for each $\ell \in 1..k$. Then, since j_ℓ and $j_{\ell+1}$ are joined by a positive edge for each $\ell \in 1..k-1$, it follows that $w[j_1] = w[j_2] = \cdots = w[j_k]$. Hence $w[i] = w[i']$. This implies that all positions of w associated with vertices in a given connected component of $\mathcal{P}_y^+[V^-]$ must have the same letter.

Now we give a coloring of $\mathcal{C}_y$ using the letters in A. Color a vertex, v, of $\mathcal{C}_y$ with $a \in A$ if there exists some i in the connected component of $\mathcal{P}_y^+[V^-]$ represented by v such that $w[i] = a$. It follows from the above discussion that this coloring operation is well-defined. We claim that this is a valid coloring. Suppose u and v are vertices of $\mathcal{C}_y$ joined by an edge. This edge corresponds to some negative edge $\{i, j\}$ in $\mathcal{P}_y$, where i is in the connected component of $\mathcal{P}_y^+[V^-]$ represented by u and j is in the connected component of $\mathcal{P}_y^+[V^-]$ represented by v. Since i and j are connected by a negative edge, it must be that $w[i] \not\approx w[j]$ which implies u and v have different colors. Hence this is a valid coloring and $\chi(\mathcal{C}_y) \leq \mu_\diamond(y)$. ◀

It is well known that if a graph G has e edges, then $\frac{\chi(G)(\chi(G)-1)}{2} \leq e$. We can use this fact to bound $\mu_\diamond(y)$.

▶ **Corollary 5.4.** *Let y be the prefix array for some partial word such that $|y| = n$. Then $\mu_\diamond(y) \leq \left\lceil \sqrt{2n} \right\rceil$.*

We can show that this bound is tight within a constant multiple using Sidon sets. In number theory, a set $S = \{s_0, s_1, \ldots, s_{m-1}\}$ of natural numbers is a finite Sidon set, named after the Hungarian mathematician Simon Sidon, if the pairwise sums $s_i + s_j$, $i \leq j$, are all different. It is easy to show that Sidon sets have the property that the pairwise differences $|s_i - s_j|$, $i < j$, are also all different.

▶ **Proposition** 5.5. There exists a prefix array, y, of size n such that $\mu_\diamond(y) = (1 - o(1))\sqrt{n}$.

Proof. Erdős and Turán [7, 9] showed that there exists a Sidon set with $(1 - o(1))\sqrt{n}$ elements such that each element is less than n. Let $S = \{s_0, s_1, \ldots, s_{m-1}\}$ be such a set, where $m = (1 - o(1))\sqrt{n}$. Define $y = y[0..n)$ by

$$y[i] = \begin{cases} s_j & \text{if } i = s_k - s_j, \text{ for some } s_j, s_k \in S, s_j < s_k, \\ n - i & \text{otherwise.} \end{cases}$$

We show that $\mathcal{P}_y^-$ contains an m-clique. Consider $s_r, s_t \in S$, where $s_r < s_t$. This implies $y[s_t - s_r] = s_r$, and since $s_t - s_r + s_r = s_t < n$, it follows that $\{y[s_t - s_r], s_t - s_r + y[s_t - s_r]\} = \{s_r, s_t\} \in E^-$, where $\{s_r, s_t\}$ indicates an edge between the vertices indexed by s_r and s_t. Moreover, this holds for any pair of elements in S, so the vertices indexed by S form an m-clique in $\mathcal{P}_y^-$. This implies $\mu_\diamond(y) \geq m$.

Now construct a partial word w on the alphabet $A = \{a_0, a_1, \ldots, a_{m-1}\}$ by $w[i] = a_j$ if $i = s_j \in S$, while $w[i] = \diamond$ otherwise. We claim that w satisfies y. Since $i + y[i] = n$ whenever $y[i] \notin S$, the only negative edges of $\mathcal{P}_y$ are of the form $\{s_r, s_t\}$ where $s_r, s_t \in S$. Note that $w[s_r] = a_r \not\approx a_t = w[s_t]$, so w satisfies all of these negative edges. Moreover, since S is an m-clique in $\mathcal{P}_y^-$, any positive edge must be incident to some vertex i where $i \notin S$. Then $w[i] = \diamond$, which matches anything, so this positive edge must be satisfied. Therefore w satisfies y on m letters, implying $\mu_\diamond(y) \leq m$. ◀

Referring to the proof of Proposition 5.5, the Sidon set $\{0, 1, 4, 6\}$ determines the prefix array $y = 7041010$. Note that $\mathcal{P}_y^-$ contains the 4-clique $\{0, 1, 4, 6\}$.

Finally, two partial words w and w' of length n are *p-equivalent* if for all i and j such that $0 \leq i \leq j < n$ we have $w[i] \approx w[j]$ if and only if $w'[i] \approx w'[j]$. In other words, w' is just a relabeling of w. If two partial words are not p-equivalent, we say they are *p-distinct*. We use this notion to bound $\text{pref}_\diamond(n)$, the number of prefix arrays of size n valid for partial words.

▶ **Proposition** 5.6. For sufficiently large n,

$$\text{pref}_\diamond(n) \leq \left\lceil \sqrt{2n} \right\rceil \left\{ {n+1 \atop \lceil \sqrt{2n} \rceil + 1} \right\},$$

where $\left\{ {n+1 \atop \lceil \sqrt{2n} \rceil + 1} \right\}$ denotes a Stirling number of the second kind.

Proof. By Corollary 5.4, any prefix array valid for partial words can be satisfied by a partial word with at most $\left\lceil\sqrt{2n}\right\rceil$ letters. Partial words which are p-equivalent have the same prefix array, so different prefix arrays are necessarily p-distinct. We may bound the number of prefix arrays valid for partial words by the number of p-distinct partial words on at most $\left\lceil\sqrt{2n}\right\rceil$ letters. Using ideas from [14], the number of p-distinct partial words of length n with h holes and μ letters is $\binom{n}{h}\left\{{n-h \atop \mu}\right\}$. Summing over all possible numbers of holes we have $\sum_{h=0}^{n-\mu}\binom{n}{h}\left\{{n-h \atop \mu}\right\} = \sum_{j=\mu}^{n}\binom{n}{j}\left\{{j \atop \mu}\right\} = \left\{{n+1 \atop \mu+1}\right\}$.

Consider p-distinct partial words using less than $\left\lceil\sqrt{2n}\right\rceil$ letters. The following facts about Stirling numbers are useful. First, for sufficiently large μ and n we have $\left\{{n \atop \mu}\right\} < \frac{\mu^n}{\mu!}$. Second, for fixed n, $\left\{{n \atop \mu}\right\}$ is a unimodal sequence with mode asympototically approaching $\frac{n}{\log n}$. Thus for sufficiently large n, we have $\left\{{n+1 \atop \lceil\sqrt{2n}\rceil+1}\right\} \geq \left\{{n+1 \atop j}\right\}$ for all $j < \left\lceil\sqrt{2n}\right\rceil$. Summing over each possible number of letters and using the unimodality of Stirling numbers,

$$\mathrm{pref}_\diamond(n) \leq \sum_{\mu=1}^{\lceil\sqrt{2n}\rceil}\left\{{n+1 \atop \mu+1}\right\} \leq \left\lceil\sqrt{2n}\right\rceil\left\{{n+1 \atop \lceil\sqrt{2n}\rceil+1}\right\}.$$

◀

6 Conclusion and Future Work

In Section 3, we demonstrated two methods for constructing an indeterminate string which satisfies a given prefix array using the smallest alphabet possible. Moreover, we showed that the minimum alphabet size for an indeterminate string satisfying a prefix array y of size n is at most $n + \sqrt{n}$. Indeed, we showed that the minimum alphabet size is at most $r + \sqrt{r} + 1$, where r is the number of negative edges in $\mathcal{P}_y$. One suggestion for future work is to improve this bound or show it is tight. Since there are many results bounding chromatic numbers, we believe the method involving $\mathcal{Q}_y$ may be useful in lowering this bound. Examining many prefix arrays has led us to the following conjecture.

▶ **Conjecture 6.1.** Let y be a feasible prefix array of size n. Then $\mu(y) < n$.

Another suggestion for future work, as mentioned in [4], is to develop an efficient algorithm to compute a string on an alphabet of minimum size for a given prefix array.

In section 5, we restricted ourselves to considering prefix arrays for partial words. We gave a characterization of prefix arrays y valid for partial words in terms of the prefix graph $\mathcal{P}_y$. Moreover, we gave a method to construct a partial word on the smallest possible alphabet for a given prefix array. We showed that the minimum alphabet size for a partial word satisfying a given prefix array of size n is at most $\left\lceil\sqrt{2n}\right\rceil$, and we showed that this bound is tight (up to a constant multiple) using Sidon sets. We also bounded $\mathrm{pref}_\diamond(n)$, the number of prefix arrays of size n valid for partial words, for large enough n, in terms of Stirling numbers of the second kind. Based on experimental data, it seems that our bound is not tight, and we believe that there is actually an exponential upper bound for $\mathrm{pref}_\diamond(n)$.

▶ **Conjecture 6.2.** For all n, $\mathrm{pref}_\diamond(n) \leq 4^{n-1}$.

One final suggestion for future work is to develop an algorithm which enumerates all prefix arrays of a given size valid for partial words. In the case of regular strings, all prefix arrays of size n can be enumerated in constant time with respect to the output size [14].

In addition, we established a World Wide Web server interface at

`http://www.uncg.edu/cmp/research/arrays` for automated use of a program that produces an indeterminate string with the minimum number of letters for a given prefix array.

Acknowledgements. We thank Professor W. F. Smyth for sending us his paper on indeterminate strings, prefix arrays, and undirected graphs. We also thank Brian Bowers and Nathan Fox for the slowly-increasing property, the number of slowly-increasing arrays of size n equalling the nth Catalan number, and the fact that the number of p-distinct partial words of length n with h holes and μ letters is $\binom{n}{h}\left\{{n-h \atop \mu}\right\}$, where $\left\{{n \atop \mu}\right\}$ denotes a Stirling number of the second kind.

References

1 K. Abrahamson. Generalized string matching. *SIAM Journal on Computing*, 16:1039–1051, 1987.
2 N. Alon. Covering graphs by the minimum number of equivalence relations. *Combinatorica*, 6:201–206, 1986.
3 F. Blanchet-Sadri. *Algorithmic Combinatorics on Partial Words.* Chapman & Hall/CRC Press, Boca Raton, FL, 2008.
4 M. Christodoulakis, P. J. Ryan, W. F. Smyth, and S. Wang. Indeterminate Strings, Prefix Arrays & Undirected Graphs. preprint, 2013.
5 J. Clément, M. Crochemore, and G. Rindone. Reverse engineering prefix tables. In S. Albers and J.-Y. Marion, editors, *STACS 2009, 26th International Symposium on Theoretical Aspects of Computer Science, Freiburg, Germany*, volume 3 of *LIPIcs*, pages 289–300. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Germany, 2009.
6 M. Crochemore, C. Hancart, and T. Lecroq. *Algorithms on Strings.* Cambridge University Press, 2007.
7 P. Erdős. On a problem of Sidon in additive number theory and on some related problems. Addendum. *Journal of the London Mathematical Society, Second Series*, 19:208, 1944.
8 P. Erdős, A. W. Goodman, and L. Pósa. The representation of a graph by set intersections. *Canadian Journal of Mathematics*, 18:106–112, 1966.
9 P. Erdős and P. Turán. On a problem of Sidon in additive number theory, and on some related problems. *Journal of the London Mathematical Society, Second Series*, 16:212–215, 1941.
10 M. Fischer and M. Paterson. String matching and other products. In R. Karp, editor, *7th SIAM-AMS Complexity of Computation*, pages 113–125, 1974.
11 J. L. Gross and J. Yellen. *Handbook of Graph Theory.* CRC Press, 2004.
12 L. Lovász. On covering of graphs. In *Theory of Graphs (Proceedings of the Colloquium, Tihany, 1966)*, pages 231–236. Academic Press, New York, 1968.
13 M. G. Main and R. J. Lorentz. An O(nlog n) algorithm for finding all repetitions in a string. *Journal of Algorithms*, 5:422–432, 1984.
14 D. Moore, W. F. Smyth, and D. Miller. Counting distinct strings. *Algorithmica*, 23:1–13, 1999.
15 W. F. Smyth. *Computing Patterns in Strings.* Pearson Addison-Wesley, 2003.
16 W. F. Smyth and S. Wang. New perspectives on the prefix array. In *15th Symposium on String Processing and Information Retrieval*, volume 5280 of *Lecture Notes in Computer Science*, pages 133–143. Springer-Verlag, Berlin, Heidelberg, 2008.
17 R. P. Stanley. *Enumerative Combinatorics*, volume 2. Cambridge University Press, 2001.
18 R. P. Stanley. *Enumerative Combinatorics*, volume 1. Cambridge Studies in Advanced Mathematics, 2011.