Adaptive Data Structures for Permutations and Binary Relations*

Francisco Claude[1,2,3] and J. Ian Munro[3]

[1] Akori S.A.
Santiago, Chile
[2] Escuela de Informática y Telecomunicaciones
Universidad Diego Portales, Chile
[3] David R. Cheriton School of Computer Science
University of Waterloo, Canada

Abstract. We present new data structures for representing binary relations in an adaptive way, that is, for certain classes of inputs we achieve space below the general information theoretic lower bound, while achieving reasonable space complexities in the worst case. Our approach is derived from a geometric data structure [Arroyuelo et al., TCS 2011]. When used for representing permutations, it converges to a previously known adaptive representation [Barbay and Navarro, STACS 2009]. However, this new way of approaching the problem shows that we can support range searching in the adaptive representation. We extend this approach to representing binary relations, where no other adaptive representations using this chain decomposition have been proposed.

1 Introduction

Binary relations and permutations arise in many applications in computer science. Examples include text indexing [12] and graph representations [8], among others. These fundamental objects have been heavily studied [11,4,5,6], and very efficient data structures supporting a wide range of operations have emerged. However, most of them remain bounded by the information theoretic lower bound in their space consumption, even in the cases where the objects have exploitable properties; for example Web Graphs [7]. Some exceptions are all the developments for compressed suffix arrays [12] and the work by Barbay and Navarro [6] on more general permutations.

In this paper, we present space-efficient data structures that have adaptive space and time complexities. Our approach comes from a geometric perspective, and for permutations, converges to the representation by Barbay and Navarro [6]. However, our new approach brings a new perspective, showing how to support range searching operations. We show that the work of [6] serves to represent binary relations with Theorem 1, and also prove an alternative tradeoff based on our own formulation of their structure.

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-3-319-02432-5_33

* First author funded in part by Google U.S./Canada PhD Fellowship. Second author funded in part by NSERC and the Canada Research Chairs Programme.

O. Kurland, M. Lewenstein, and E. Porat (Eds.): SPIRE 2013, LNCS 8214, pp. 64–71, 2013.

The paper is organized as follows. In Section 2 we present related work on representing permutations and binary relations, we also include some background on data structure for range searching proposed by Arroyuelo et al. [1]. In Section 3 we present our adaptive representation for permutations, and show how this representation converges to the one by Barbay and Navarro. Next, in Section 4, we extend the representation for permutations to cover binary relations in general, presenting two different approaches. Then, in Section 4.2, we present one simple application of our structure. Finally, in Section 5, we present our conclusions.

2 Related Work

We present the related work in the next two subsections. The first one presents previous results on representing permutations and binary relations; the second covers recent results on adaptive range searching.

2.1 Permutation and Binary Relations

The most common queries for a permutation Π over $[n]$[1] are: (1) $\pi(i)$: obtain the value of $\Pi[i]$; (2) $\pi^{-1}(j)$: find i such that $j = \Pi[i]$; (3) $\pi^k(i)$: apply π k times, similarly we define $\pi^{-k}(j)$; and (4) $\mathcal{R}_\Pi(i_1, i_2, j_1, j_2)$: find elements i such that $i_1 \leq i \leq i_2$ and $j_1 \leq \pi(i) \leq j_2$.

One efficient representation for arbitrary permutations is that of Munro et al. [11]. This representation achieves $(1+\epsilon)n \lg n(1+o(1))$ bits. It supports π in $O(1)$ time and π^{-1} in $O(\frac{1}{\epsilon})$ time. They also showed that $\pi^{\pm k}$ can be supported in the same time as the time required to perform both π and π^{-1} by using only $O(n)$ extra bits. This extension applies to any representation, and thus, to our results too.

The $\mathcal{R}$ operation is less commonly required, but also of interest. For instance, consider a position-restricted search using a suffix array. The suffix array is a permutation and searching for a pattern P between positions p_1 and p_2 is just the result of doing a range query over the range of suffixes starting with P (i.e., $[i_1, i_2]$) and those pointing to positions in $[p_1, p_2]$. Mäkinen and Navarro showed how to use wavelet trees to solve this operation and all others in $O(\lg n)$ time within $n \lg n(1+o(1))$ bits of space [10].

Prior to this paper the only adaptive representation for permutations was that proposed by Barbay and Navarro [6]. They show many possible decompositions into monotonic sequences and subsequences, and give their space/time complexities in term of the entropy of such sequences. As we will see later, we converge to the same structure at the end of Section 3.

A natural representation for a permutation is a binary matrix of $n \times n$ where we mark the coordinates (i, j) with a 1 iff $\pi(i) = j$. We will use this conceptual representation in our construction. For a binary relation $\mathcal{B}$ over two sets $[n_1]$

[1] We use $[n]$ to represent $\{1, 2, \ldots, n\}$.