Characterizing History Independent Data Structures

Jason D. Hartline, Edwin S. Hong, Alexander E. Mohr, William R. Pentney, and Emily C. Rocke

Abstract. We consider history independent data structures as proposed for study by Naor and Teague [3]. In a history independent data structure, nothing can be learned from the memory representation of the data structure except for what is available from the abstract data structure. We show that for the most part, strong history independent data structures have canonical representations. We provide a natural alternative definition of strong history independence that is less restrictive than [3] and characterize how it restricts allowable representations. We also give a general formula for creating dynamically resizing history independent data structures and give a related impossibility result.

Key Words. Data structures, History independence, Markov chains, Algorithms.

1. Introduction. On April 16, 2000, the New York Times published an article regarding the CIA’s role in the overthrow of the Iranian government in 1953. In addition to the article, the Times’ website posted a CIA file from 1954 that detailed the actions of various revolutionaries involved in the plot. The Times opted to black out many of the names mentioned in the document; some of the people referred to were still alive and residing in Iran, and could have been put at risk for retribution. The file was published as an Adobe PDF file that contained the original document in its entirety and an overlay covering up parts of the document. Shortly after releasing the document, some Internet users reverse engineered the overlay and made the original document available on the Web. In an environment where information is valuable, private, incriminating, etc., the Times’ blunder represents a particularly grievous instance of the problems caused by data structures that retain information about previous operations performed upon them. History independent data structures are designed not to reveal any information beyond that necessarily provided by the contents of the data structure.

The idea of maintaining a data structure so that no extraneous information is available was first explicitly studied by Micciancio [2]. This work studied “oblivious trees” where no information about past operations could be deduced from the pointer structure of the nodes in the search tree. In [4] Snyder studied bounds on the performance of search,
insert, and delete functions on uniquely represented, in terms of the pointer structure, tree-based data structures. More stringent history independence requirements were studied by Naor and Teague in [3]. In their model the entire memory representation of a history independent data structure, not just the pointer structure, must not divulge information about previous states of the data structure. Following [3] we consider two types of history independence: weak history independence, in which we assume that a data structure will only be observed once; and strong history independence, in which case the data structure may be observed multiple times. A data structure is history independent if nothing can be learned from the data structure’s memory representation during these observations except for the current abstract state of the data structure.

In Section 3 we give a simple definition of strong history independence. We show in Section 6 that this definition is equivalent to that of [3]. Under this definition, strong history independent implementations of many data structures must satisfy a natural canonicality criterion (Section 4). For example, a strongly history independent implementation of a hash table has the property that up to randomness in the initialization of the hash table, e.g., the choice of hash functions, the hash table’s representation in memory must be deterministically given by its contents. This answers an open question posed in [3] about the necessity of canonical representations.

In Section 5 we consider a natural relaxation of strong history independence, where non-canonical representations and randomness can be used. However, we show that even under this less restrictive definition, there are still very stringent limitations on using non-canonical representations.

Finally, in Section 7 we discuss the issue of creating dynamically resizing history independent data structures. In [3] Naor and Teague presented a weakly history independent dynamically resizable hash table. The scheme they employed can be generalized to give a technique for making any history independent data structure dynamically resizable using amortized constant time against an oblivious adversary. We give a general technique for dynamically resizing weak history independent data structures in amortized constant time against a non-oblivious adversary. We prove that no such technique exists for strongly history independent dynamically resizing data structures. This result provides insight into the open problem of whether there is a complexity separation between weak and strong history independence [3].

2. Preliminaries. The results presented in this paper apply to history independent data structures in general. To this end we must have a general understanding of data structures. An abstract data structure defines the set of operations for a data structure and its semantics. A data structure’s state is the current value or contents of the data structure as specified by its abstract data structure. A data structure’s representation in

---

6 We use two standard variants for analysis of data structures that use randomness. We consider worst-case performance on a sequence of operations created by an oblivious adversary, one that has no knowledge of the random coin flips in our data structure. Similarly, we consider worst-case performance on a sequence of operations constructed on the fly by a non-oblivious adversary, one that knows our random coin flips. A non-oblivious adversary, for instance, can cause a hash table with a randomized hash function to perform poorly by causing all inserted elements to hash to the same location.