June 2008

MADALGO seminar by Seth Pettie, University of Michigan

Analyzing Splay Trees Using Davenport-Schinzel Sequences

The splay tree is a dynamic binary search tree that is conjectured to be universally efficient in the following way. On any sequence of accesses the splay tree is conjectured to take time within a constant factor of the optimal (offline) dynamic binary search tree. Splay trees are traditionally analyzed using potential functions or some other counting argument.

In this talk I will present a new way to analyze splay trees (and dynamic data structures in general). The three-part strategy is to (1) transcribe the operations of the data structure as some combinatorial object, (2) show the object has some forbidden substructure, and (3) to prove upper bounds on the size of such a combinatorial object. As an example of this strategy, we show that splay trees execute a sequence of N deque operations (push, pop, inject, and eject) in O(Na^* (N)) time, where a^* is the iterated-inverse-Ackermann function. This bound is within a tiny a^*(N) factor of that conjectured by Tarjan in 1985.