Inverse Ackermann in Linear Time

Abstract

We build a heirarchy of functions in Gallina that computes the result of the inverse Ackermann function in linear time. Some of the functions in the heirarchy are themselves novel results.

Keywords: Inverse Ackermann, Automata, Union-Find, Division

1. Overview

The time complexity of the union-find data structure has traditionally been hard to estimate, especially when it is implemented with the heuristic rules of *path compression* and *weighted union*. Tarjan showed that for a sequence of m FINDs intermixed with $n-1$ UNIONs such that $m \geq n$, the time required $t(m, n)$ is bounded as: $k_1 m\alpha(m, n) \leq t(m, n) \leq k_2 m\alpha(m, n)$. Here k_1 and k_2 are positive constants and $\alpha(m, n)$ is the inverse of the Ackermann function.

The Ackermann function, commonly denoted $A(m, n)$, has a few variants, but a well-known version is: blah

The inverse of this function, commonly denoted $\alpha(m, n)$, also has a few variants. Happily, however, the various definitions of α have very similar asymptotic behaviour, in that they differ by small constants only. In this paper, we mechanize a particularly elegant version of α in Gallina, and argue for blah time bound. We then manoeuvre towards (this other definition) of α.

2. The Inverse Ackermann Heirarchy

blah

[1]