Tradeoff in Union-Find with Path Compression

Amir M. Ben-Amram

March 12, 2008

Abstract

This note shows how to trade union time for find time in the path-compression framework.

The Algorithm. We use ordinary find, while union is a modification of the ordinary algorithm, as follows.

- Every tree root has a size field as well as a rank.

- In a union, if the smaller tree does not exceed $\alpha_k(\lfloor \log n \rfloor)$ nodes, it is shattered: all its nodes become immediate children of the other tree’s root, and their rank is set to 0 (it could only be 0 or 1 before the operation).

  If the smaller tree exceeds $\alpha_k(\lfloor \log n \rfloor)$, ordinary union by rank is applied, so the rank may grow (in this case the size field can be abandoned and not updated any more).

Theorem 1. Let $k \geq 0$. Union-find (based on Galler-Fischer trees and path compression) can be implemented such that union has worst-case $O(\alpha_k(\lfloor \log n \rfloor))$, and a sequence of $m \geq n$ finds has amortized cost $O(k)$ per find.

Proof. We split the nodes into two types: rank 0 nodes and the rest. In a find, at most one node (the first) may have rank 0. Therefore, such nodes contribute $O(1)$ to the cost of each find. Let $X$ be the part of the reference forest that consists of nodes that have rank greater than zero, excluding nodes of rank 1 that are roots (clearly, these do not contribute to the cost of finds).

In other words, nodes in $X$ have participated in unions in which they were hung on another root, and not shattered.

We claim that $|X| \leq n/\alpha_k(n)$. This is easy: each node in $X$ has at least $\alpha_k(n) - 1$ leaves of rank 0 hanging from it. Thus, using Sharir and Seidel’s theorem, the cost of $m$ compressions in this forest is bounded by

$$mk + 2|X|\alpha_k(r)$$
where \( r \) is the maximal rank, clearly bounded by \( \log n \). Thus, using our bound on \(|X|\),

\[
mk + 2|X|\alpha_k(r) \leq mk + 2n
\]

which, assuming \( m \geq n \), bounds the amortized cost of a find by \( k + 2 \) (to which 1 has to be added for the rank-0 node). \( \square \)