Amortisierte Analyse von Union-Find

Sei eine Folge von m Operationen gegeben, unter denen n Make-Set sind.

Dem Wald, der unmittelbar nach der q-ten Operation vorliegt, ordnen wir ein Potential $\Phi_q := \sum_x \phi_q(x)$ zu, wobei die Summe über die Knoten des Waldes rangiert und die Knotenpotentiale $\phi_q(x)$ wie folgt gegeben sind.

- Ist $rank[x] = 0$, so $\phi(x) = 0$,
- Ist x Wurzel, so $\phi_q(x) = (\log^*(n) + 2) \cdot rank[x]$,

In allen anderen Fällen richtet sich das Potential eines Knotens nach dem Abstand seines Ranges zu dem des Elternknotens. Man beachte, dass dieser Abstand stets ≥ 1 ist und dass im Falle, den wir betrachten, auch $rank[x] \geq 1$.

Amortisierte Analyse von Union-Find

Wir teilen die Knoten in drei Stufen ein und definieren zu jedem Knoten ein Maß, welches sich nach dem Abstand und nach der Stufe richtet.

Stufe 0 Diese liegt vor, wenn $rank[x] + 1 \leq rank[p[x]] < 2 \cdot rank[x]$. Man setzt dann $maß(x) = rank[p[x]] - rank[x]$.

Stufe 1 Diese liegt vor, wenn $2 \cdot rank[x] \leq rank[p[x]] < 2^{rank[x]}$. Man setzt dann $maß(x) = \max\{k \mid 2^k \cdot rank[x] \leq rank[p[x]]\}$.

Stufe 2 Diese liegt vor, wenn $2^{rank[x]} \leq rank[p[x]]$. Man setzt dann $maß(x) = \max\{k \mid 2_k^{rank[x]} \leq rank[p[x]]\}$, wobei $2_0^j = j, 2_{i+1}^j = 2^{2_i^j}$.

In den Stufen 0 und 1 ist $1 \leq maß(x) < rank[x]$. In Stufe 2 hat man $1 \leq maß(x) \leq \log^*(n)$.

Nunmehr definiert man das Potential des Knotens x zu

$$\phi_q(x) = (2 + \log^*(n) - S) \cdot rank[x] - maß[x]$$

wobei S die Stufe von x ist.

Das Potential ist also nichtnegativ.

Im Laufe der Zeit kann die Stufe nicht abnehmen; nimmt das Mass ab, so muss im Gegenzug die Stufe ansteigen. In beiden Fällen nimmt das Potential echt ab.

Amortisierte Analyse von Union-Find

- Die amortisierten Kosten einer Make-Set Operation sind $O(1)$.
- Die amortisierten Kosten einer Link Operation sind $O(\log^*(n))$.
- Die amortisierten Kosten einer Find Operation sind $O(1)$.

Amortisierte Analyse von Union-Find

Die erste Aussage ist klar: keine Potentialänderung, tatsächliche Kosten $O(1)$.

Im zweiten Falle entstehen auch tatsächliche Kosten $O(1)$; das Gesamtpotential kann aber um maximal $2 + \log^*(n)$ ansteigen, nämlich dann, wenn sich der Rang einer Wurzel erhöht.

Im dritten Falle entstehen tatsächliche Kosten von $O(s)$ wobei s die Anzahl der Knoten auf dem Bestimmungspfad sind. Die konstanten amortisierten Kosten ergeben sich daraus, dass durch die Pfadkompression für mindestens $s - 5$ Knoten eine Potentialverringerung eintritt. Sei nämlich x ein innerer Knoten auf dem Bestimmungspfad, auf den weiter oben ein Knoten y der gleichen Stufe folgt und sei $f(x) = x + 1$, falls diese Stufe 0 ist, $f(x) = 2x$, falls diese Stufe 1 ist, $f(x) = 2^x$, falls diese Stufe 2 ist. Es gilt dann

$$
\begin{aligned}
&\mathit{rank}[p[x]] \geq f^{(\mathit{maß}(x))}(\mathit{rank}[x]) \\
&\mathit{rank}[p[y]] \geq f(\mathit{rank}[y]) \\
&\mathit{rank}[y] \geq \mathit{rank}[x]
\end{aligned}
$$

Also $\mathit{rank}[p[y]] \geq f^{(\mathit{maß}(x)+1)}(\mathit{rank}[x])$. Nach der Pfadkompression nimmt also entweder das Maß von x oder die Stufe um mindestens 1 zu.

Amortisierte Analyse von Union-Find

Durch Einbeziehung höherer Stufen mit entsprechender Definition erhält man eine Schranke von $O(m \cdot \alpha(n))$ an die Komplexität einer Folge von m Operationen auf n Daten. Hierbei ist $\alpha(n)$ die Inverse der Ackermannfunktion, also das kleinste k sodass $A(k,1) > n$. Tarjan hat gezeigt, dass diese Schranke für eine große Klasse von Algorithmen bestmöglich ist. Siehe Buch.