Die Seidel-Sharir Analyse für den Union-Find Algorithmus

Wolfgang Mulzer

1 Elimination der UNION-Operationen

Sei F ein Wald. Ein *Elternpfad* P in F ist eine Folge v_1, v_2, …, v_k von Knoten, so dass v_{i+1} der Elternknoten von v_i ist, für $i = 1, \ldots, k-1$. Ein Elternpfad heißt *Wurzelpfad*, wenn v_k eine Wurzel in F ist.

Eine *verallgemeinerte Kompression* in F ist gegeben durch einen Elternpfad $P = v_1, v_2, \ldots, v_k$ in F. Wir ersetzen die Elternknoten aller Knoten v_1, v_2, …, v_{k-1} durch den Elternknoten von v_k (durch $\perp$, wenn v_k ein Wurzelknoten ist). Die *Kosten* von P sind 0, wenn P Wurzelpfad ist, und $k-1$ sonst.

Eine *Folge* von verallgemeinerten Kompressionen $C = (P_1, P_2, \ldots, P_m)$ für einen Wald F ist wie folgt definiert. Sei $F_0 := F$. Für $i = 1, \ldots m$ bezeichnen wir mit F_i den Wald, den man erhält, wenn man die verallgemeinerte Kompression P_i in F_{i-1} ausführt. Dazu muss P_i jeweils ein Elternpfad in F_{i-1} sein. Die *Kosten* der Folge C, $\mathrm{cost}(C)$, sind definiert als die Summe der Kosten der P_i. Die Länge von C, $\ell(C)$, ist die Anzahl der *Nichtwurzelpfade* in C.

Lemma 1. *Gegeben sei eine Folge X von **Union-Find** Operationen, die m **Find**s enthält. Wir beginnen mit der Partition $\{\{1\}, \{2\}, \ldots, \{n\}\}$ und verwenden Union-By-Rank mit Pfadkompression. Dann existieren ein Wald F und eine Folge C von verallgemeinerten Kompressionen für F, so dass folgendes gilt: (i) $\ell(C) \leq m$; (ii) die Gesamtlaufzeit von X ist $O(m + n + \mathit{cost}(C))$; und (iii) jeder Knoten im Wald F hat einen Rang $\mathit{rg}(v) \in \{0, \ldots, \log n\}$, so dass es höchstens $n/2^k$ Knoten v mit $\mathit{rg}(v) \geq k$ gibt und der Rang eines jeden Knoten echt kleiner ist als der seines Elternknotens.*

Beweis. Führe zunächst nur die **Union**-Operationen in X mit Hilfe der Union-by-Rank Heuristik durch. Sei F der resultierende Wald. Aus den Eigenschaften der Union-By-Rank Heuristik folgt, dass F Eigenschaft (iii) erfüllt.

Nun konstruieren wir C. Dazu betrachten wir jede Operation $\mathbf{Find}(s)$ in X. Sei s' das Ergebnis von $\mathbf{Find}(s)$. Wenn $s \neq s'$, so fügen wir zu C den Pfad von s nach s' im aktuellen Wald hinzu, wobei aber wir s' allerdings weglassen. Offenbar ist dann $\ell(C) \leq m$. C ist auch eine gültige Folge von Kompressionen für F. Nun zur Laufzeit: X enthält höchstens n **Union**s, von denen jedes konstante Zeit braucht. Für die m **Find**s berechnen wir jeweils konstante Laufzeit plus die Zeit zum Ablaufen des Pfades. Letzteres ist durch $\mathrm{cost}(C)$ abgedeckt (kein Pfad in C ist Wurzelpfad). Somit folgt (ii). □

2 Verstehen der verallgemeinerten Pfadkompression

Sei $F = (V, E)$ ein Wald. Sei $V = V_b \cup V_t$ eine Zerlegung von V in disjunkte Teilmengen, so dass V_t nach oben abgeschlossen ist. Das heißt, wenn $v \in V_t$ ist, so ist auch der Elternknoten von v in V_t. Wir bekommen eine Zerlegung von F in $F_t = (V_t, E_t)$ und $F_b = (V_b, E_b)$, wobei $E_t \subseteq E$ die Kanten mit beiden Endknoten in V_t sind und $E_b \subseteq E$ die Kanten mit beiden Endknoten in V_b. F_t ist der *obere Wald*, F_b der *untere Wald*. Jetzt kommt das entscheidende Lemma

Lemma 2. *Sei F ein Wald und C eine Folge von Kompressionen für F. Sei $F_t = (V_t, E_t)$, $F_b = (V_b, E_b)$ eine Zerlegung von F. Dann existieren Kompressionsfolgen C_t für F_t, C_b für F_b, so dass gilt: (i) $\ell(C_t) + \ell(C_b) \leq \ell(C)$; und (ii) $cost(C) \leq cost(C_t) + cost(C_b) + |V_b| + \ell(C_t)$.*

Beweis. Wir gehen die Kompressionen in C einzeln durch. Für jede Kompression $P \in C$ bilden wir den Schnitt $P_b = C \cap V_b$, und wir fügen P_b zu C_b hinzu, falls er nicht leer ist. Ebenso bilden wir $P_t = C \cap V_t$, und wir fügen P_t zu C_t hinzu, falls $P_t \neq \emptyset$. Wenn sowohl P_t als auch P_b nicht leer ist, so ist P_b ein Wurzelpfad in F_b und zählt nicht zu $\ell(C_b)$. Daher gilt (i).

Nun beweisen wir (ii). Die Kosten $\text{cost}(C)$ zählen die folgenden Ereignisse, die bei einer Kompression passieren können: (a) ein Knoten aus V_t bekommt einen neuen Elternknoten in V_t; (b) ein Knoten aus V_b bekommt einen neuen Elternknoten in V_b; (c) ein Knoten in V_b bekommt einen neuen Elternknoten in V_t. Die Ereignisse (a) werden auf der rechten Seite von $\text{cost}(C_t)$ gezählt, die Ereignisse (b) von $\text{cost}(C_b)$. Das erste Mal, dass Ereignis (c) einem Knoten aus $v \in V_b$ zustößt, wir von dem Term $|V_b|$ abgedeckt. Jedes weitere Mal wird von dem Term $\ell(C_t)$ abgedeckt, da dies dem Knoten v nur widerfahren kann, wenn ein Pfad $P \in C$ in zwei nichtleere Teile geteilt wurde und v der letzte Knoten in P_b ist (Es wird kein Pfad P mehrfach dazu herangezogen, da es pro Pfad nur nur einen solchen Knoten gibt). □

3 Bootstrapping

Beobachtung 3. *Sei $F = (V, E)$ ein Wald, in dem jeder Knoten höchstens Rang r hat. Dann gilt für jede Kompressionsfolge C für F, dass $cost(C) \leq r \cdot |V|$ ist.*

Beweis. jedesmal wenn ein Knoten v bei einer Kompression einen neuen Elternknoten erhält, hat dieser einen höheren Rang als der alte Elternknoten von v. Daher kann dies nur r mal pro Knoten passieren. □

Setze $s := \log r$. Sei $V_{\leq s}$ die Menge aller Knoten in F mit Rang höchstens s und $V_{>s}$ die Menge der Knoten in F mit Rang größer s. Dann erzeugt $V_{\leq s}$, $V_{>s}$ eine Zerlegung von F in die Wälder $F_{\leq s}$ und $F_{>s}$, und Lemma 2(ii) gibt:

$$\text{cost}(C) \leq \text{cost}(C_{\leq s}) + \text{cost}(C_{>s}) + |V_{\leq s}| + \ell(C_{>s}).$$

Nun wollen wir diese Gleichung interpretieren. Zunächst gilt wegen Lemma 1(iii) für die Anzahl der Knoten im oberen Wald $|V_{>s}| \leq n/2^s \leq n/r$. Also folgt nach Beobachtung 3, dass $\text{cost}(C_{>s}) \leq r \cdot |V_{>s}| \leq n$. Außerdem ist $|V_{\leq s}| \leq n$ (da es nur n Knoten gibt) und $\ell(C_{>s}) \leq \ell(C) - \ell(C_{\leq s})$ wegen Lemma 2(i). Somit

$$\text{cost}(C) - \ell(C) \leq \text{cost}(C_{\leq s}) - \ell(C_{\leq s}) + 2n.$$

Nun ist aber $F_{\leq s}$ ein Wald mit Rang höchstens $s = \log r$. Daher können wir nun das gleiche Spiel mit $F_{\leq s}$ und $s' = \log s = \log \log r$ wiederholen, und wir erhalten

$$\text{cost}(C_{\leq s}) - \ell(C_{\leq s}) \leq \text{cost}(C_{\leq s'}) - \ell(C_{\leq s'}) + 2n,$$

also

$$\text{cost}(C) - \ell(C) \leq \text{cost}(C_{\leq s'}) - \ell(C_{\leq s'}) + 4n.$$

Das geht nun weiter bis wir bei einem Wald angekommen sind, in dem jeder Rang echt kleiner als 2 ist. Dies ist nach $\log^* r$ Schritten der Fall. Bei jedem Schritt erhöht sich der Faktor vor dem n um 2, somit

$$\text{cost}(C) - \ell(C) \leq 2n \log^* r.$$

Dies ist aber noch lange nicht alles. Mit der neuen Schranke können wir jetzt wieder von vorne anfangen, und erhalten noch eine bessere Schranke. Dazu definieren wir rekursiv eine Funktion $\log^{*(j)} r$ wie folgt: $\log^{*(0)} r = \log r$, für alle r. Angenommen, wir haben $\log^{*(j)}$ definiert. Dann ist $\log^{*(j+1)} r = 0$, falls $r < 2$, und $\log^{*(j+1)} r = 1 + \log^{*(j+1)}(\log^{*(j)} r)$, sonst. In Worten: $\log^{*(j+1)} r$ besagt, wie oft man die Funktion $\log^{*(j)}$ auf r anwenden muss, bis man eine Zahl kleiner als 2 erhält.

Lemma 4. *Angenommen, für jeden Wald mit Rang höchstens r und für jede Kompressionsfolge C gilt*

$$\mathit{cost}(C) \leq j\ell(C) + (j+1)n \log^{*(j)} r.$$

Dann gilt auch

$$\mathit{cost}(C) \leq (j+1)\ell(C) + (j+2)n \log^{*(j+1)} r.$$

Beweis. Sei F ein Wald mit Rang höchstens r und C eine Kompressionsfolge. Setze $s = \log \log^{*(j)} r$, und sei $F_{\leq s}$, $F_{>s}$ die Zerlegung von F in Knoten mit Rang höchstens s und Knoten mit Rang größer s. Lemma 2(ii) gibt

$$\text{cost}(C) \leq \text{cost}(C_{\leq s}) + \text{cost}(C_{>s}) + |V_{\leq s}| + \ell(C_{>s}).$$

Nach Annahme ist

$$\text{cost}(C_{>s}) \leq j\ell(C_{>s}) + (j+1)|V_{>s}| \log^{*(j)} r \leq j\ell(C_{>s}) + (j+1)n,$$

denn $|V_{>s}| \leq n/2^s \leq n/\log^{*(j)} r$. Weiter ist $|V_{\leq s}| \leq n$. Also folgt

$$\text{cost}(C) \leq \text{cost}(C_{\leq s}) + (j+2)n + (j+1)\ell(C_{>s}).$$

Nun benutzen wir noch Lemma 2(i) und erhalten

$$\text{cost}(C) - (j+1)\ell(C) \leq \text{cost}(C_{\leq s}) - (j+1)\ell(C_{\leq s}) + (j+2)n.$$

Nun wiederholen wir das ganze mit $F_{\leq s}$, bis wir schließlich wieder bei Rang kleiner als zwei angelangt sind. Da wir jedesmal die Funktion $\log \log^{*(j)}$ auf den Rang anwenden, ist dies nach höchstens $\log^{*(j+1)} r$ Schritten der Fall. Jedesmal kommt ein $(j+2)n$ Term hinzu, woraus das Lemma folgt. □

Aus Lemma 3 können wir folgern, dass für jeden Wald F, jede Kompressionsfolge C und jedes $j \geq 0$ gilt:

$$\text{cost}(C) \leq (j+1)m + (j+1)n \log^{*(j)} n.$$

Hierbei haben wir $\ell(C) \leq m$, $r \leq n$ und $j \leq (j+1)$ abgeschätzt, um die Ungleichung zu vereinfachen. Wie sollen wir nun j wählen? Am besten sollen beide Summanden balanciert werden, also sollte gelten

$$(j+1)m \approx (j+1)n \log^{*(j)} n \Leftrightarrow \log^{*(j)} n \approx m/n.$$

Hierbei haben wir angenommen, dass $m \geq n$ ist (was man leicht erreichen kann, ohne die asymptotische Laufzeit zu erhöhen, z.B., indem man am Anfang auf jedem Element ein **Find** ausführt). Wir definieren nun:

$$\alpha(m,n) := \min\{j \geq 0 \mid \log^{*(j)} n \leq m/n + 1\}.$$

Die Funktion $\alpha(n,m)$ heißt die *inverse Ackermannfunktion* von m und n. Wenn wir $j = \alpha(m,n)$ wählen, ist also

$$\text{cost}(C) \leq (2m+n)(\alpha(m,n)+1).$$

Wenn wir alles zusammensetzen, erhalten wir:

Satz 5. *Gegeben sei eine Folge X von **Union-Find** Operationen, die $m \geq n$ **Find**s enthält. Wir beginnen mit der Partition $\{\{1\},\{2\},\ldots,\{n\}\}$ und verwenden Union-By-Rank mit Pfadkompression. Dann ist die Laufzeit von X höchstens $O(n + m\alpha(m,n))$.*

Wir sagen: die *amortisierte* Laufzeit für eine **Union**-Operation ist $O(1)$ und die amortisierte Laufzeit für eine **Find**-Operation ist $O(\alpha(m,n))$.