Bachelorarbeit

Andreas Windorfer

21. Juni 2020

Inhaltsverzeichnis

1 Tango Baum

Der Tango Baum ist ein aus BSTs, den **auxiliary trees**, bestehender BST. Auf die Anforderungen an die auxiliary trees wird in Abschnitt 1.2 eingegangen und mit dem Rot-Schwarz-Baum wird eine mögliche Variante noch detailliert vorgestellt. Der Tango Baum wurde in [1] beschrieben, inklusive eines Beweises über seine $lg\, lg\, n$-competitiveness. Ebenfalls in [1] enthalten ist eine als **Interleave Lower Bound** bezeichnete Variation der ersten unteren Schranke von Wilber. Da diese für das Verständnis des Tango Baumes wesentlich ist, wird mit ihr gestartet, bevor es zur Beschreibung der Struktur selbst kommt.

1.1 Interleave Lower Bound

Sei X eine Zugriffsfolge und sei $K = \{k \in \mathbb{N} | k \textit{ ist in } X \textit{ enthalten}\}$. Auch hier wird ein lower bound tree verwendet, dieser ist jedoch etwas anders definiert als in Abschnitt **??**. Hier ist der lower bound tree Y zu einer Zugriffsfolge X, der vollständig balancierte BST mit Schlüsselmenge K, bei dem die unterste Ebene von links startend nach rechts besetzt wird. Auch in der untersten Ebene gibt es also keine Möglichkeit einen Knoten links von einem bereits vorhandenen Knoten einzufügen. Anders als in Abschnitt **??** gibt es hier somit zu jeder Zugriffssequenz nur genau einen lower bound tree. Abbildung 1 zeigt den lower bound tree zur Zugriffsfolge $1, 2, .., 15$. Zu jedem Knoten v in Y werden zwei Mengen definiert. Die **linke Region** von v enthält den Schlüssel von v, sowie die im linken Teilbaum von v enthaltenen Schlüssel. Die **rechte Region** von v enthält die im rechten Teilbaum von v enthaltenen Schlüssel. Sei l der kleinste Schlüssel im Teilbaum mit Wurzel v und r der größte. Sei $X = \{x_1, x_2, ., x_m\}$ die Zugriffssequenz und $X_l^r = x_{1'}, x_{'2}, .., x_{jm'}$ wie in Abschnitt **??** definiert. $i \in \{2, 3, .., m'\}$ ist ein „Interleave durch v“ wenn $x_{(i-1)}$ in der linken Region von v liegt und x_i in der rechten Region von v, oder umgekehrt. Sei *inScore(v)* die Funktion die zu dem Knoten v die Anzahl der Interleaves durch v zurückgibt. Sei U die Menge der Knoten von Y. Die Funktion $IB(X)$ ist definiert durch:

$$IB\,(X) = \sum_{u \in U} inScore\,(u)$$

Sei T_0 der BST mit Schlüsselmenge K auf X ausgeführt wird. Für $i \in \{1, 2, .., m\}$ sei T_i der BST, der entsteht nachdem $access(x_i)$ auf T_{i-1} ausgeführt wurde. Zu $u \in U$ und $j \in \{0, 1, .., m\}$ gibt es einen **transition point** v in T_j. v ist ein Knoten mit folgenden Eigenschaften:

1. Der Pfad von der Wurzel von T_j zu v enthält einen Knoten dessen Schlüssel in der linken Region von u enthalten ist.

2. Der Pfad von der Wurzel von T_j zu v enthält einen Knoten dessen Schlüssel in der rechten Region von u enthalten ist.

3. In T_i ist kein Knoten mit Eigenschaft 1 und 2 enthalten, der eine kleinere Tiefe als v hat.

Im Beweis dieses Abschnittes wird gezeigt das $OPT(X) \geq IB(X)$ gilt. Dafür werden jedoch noch drei Lemmas zu den Eigenschaften von Y benötigt.

Lemma 1.1. *Zu jedem Knoten $u \in U$ und $j \in \{0, 1, .., m\}$ gibt es genau einen transition point in T_j.*

Beweis. Sei l der gemeinsame Vorfahre aller Knoten mit einem Schlüssel aus der linken Region von u in T_j. Sei r der gemeinsame Vorfahre aller Knoten mit einem Schlüssel aus der rechten Region von u in T_j. $key(l)$ bzw. $key(r)$ muss selbst in der linken bzw. rechten Region von u enthalten sein, vergleiche ??.

□

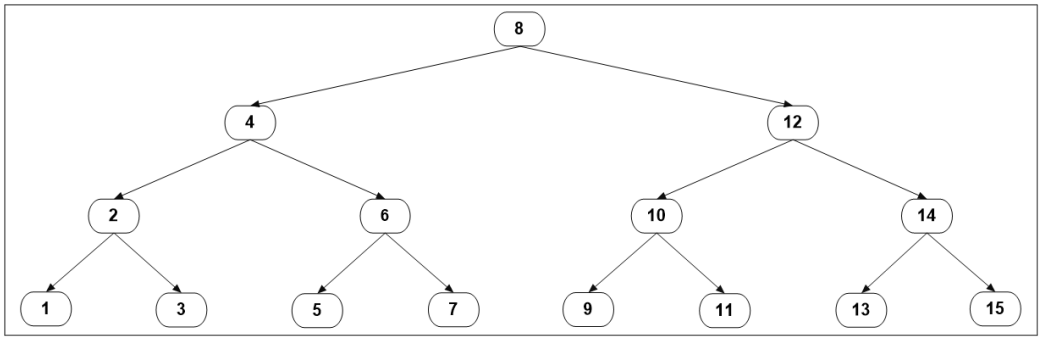

Abbildung 1: Der lower bound tree zur Zugriffsfolge $1, 2, .., 15$

1.2 Aufbau des Tango Baum

1.3 Die *access* Operation beim Tango Baum

1.4 Laufzeitanalyse

Literatur

[1] Erik D. Demaine, Dion. Harmon, John. Iacono, and Mihai. Patrascu. Dynamic optimality—almost. *SIAM Journal on Computing*, 37(1):240–251, 2007.