

Bachelorarbeit

Andreas Windorfer

2. April 2020

Zusammenfassung

Inhaltsverzeichnis

1	Einleitung	4
2	Fazit	4
3	Dynamische binäre Suchbäume	4
4	Balancierte Suchbäume	4
4.1	Rot-Schwarz-Baum	4
4.1.1	Einfügen in den Rot-Schwarz-Baum	5
4.1.2	Löschen aus dem Rot-Schwarz-Baum	8
4.1.3	Suchen im Rot-Schwarz-Baum	8

1 Einleitung

2 Fazit

3 Dynamische binäre Suchbäume

innerer Knoten

Teilbaum

Höhe Nur eine Wurzel vorhanden, dann Höhe 0

innerer Knoten, auch Wurzel

Rotationen

Einfügen

Suchen

Löschen

Bruder

4 Balancierte Suchbäume

4.1 Rot-Schwarz-Baum

Der Rot-Schwarz-Baum gehört zur Gruppe der balancierten binären Suchbäume. Zusätzlich zu den Zeigern *links*, *rechts* und *vater* benötigt jeder Knoten ein zusätzliches Bit Speicherplatz, um die Farbinformation zu speichern. Der Name der Datenstruktur kommt daher, dass die beiden durch das zusätzliche Bit unterschiedenen Zustände als *Rot* und *Schwarz* bezeichnet werden. Die Farbe ist also eine Eigenschaft der Knoten. Zeiger auf Kinder oder Väter die aktuell im Baum nicht vorhanden sind zeigen auf einen schwarzen Sonderknoten. Dieser Sonderknoten enthält in dieser Arbeit den Schlüsselwert *null*.

Folgende zusätzliche Eigenschaften müssen bei einem Rot-Schwarz-Baum erfüllt sein.

1. Jeder Knoten ist entweder rot oder schwarz.
2. Die Wurzel ist schwarz.
3. Jedes Blatt ist schwarz.
4. Beide Kinder eines roten Knotens sind schwarz.

5. Für jeden Knoten enthalten alle Pfade, die an diesem Knoten starten und in einem Blatt enden, die gleiche Anzahl an schwarzen Knoten.

Der Zweck dieser Einschränkungen ist es die Höhe des Rot-Schwarz-Baumes zu begrenzen, um die üblichen Operationen effizient ausführen zu können. Als Schwarz-Höhe $bh(x)$ eines Knotens x wird die Anzahl der schwarzen Knoten in einem Pfad, der am dem Knotens startet und bei einem Blatt endet, bezeichnet. Die eigene Farbe des betrachteten Knotens bleibt dabei außen vor. Aufgrund der Eigenschaft fünf ist die Schwarz-Höhe eines Knotens wohldefiniert. Die Höhe h ist beim Rot-Schwarz-Baum wie bei anderen binären Suchbäumen definiert und berücksichtigt auch die roten Knoten.

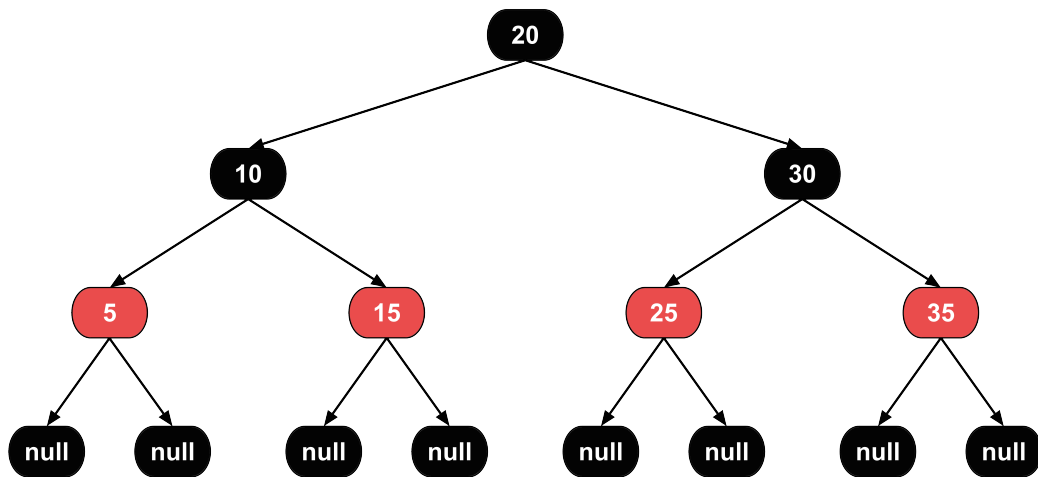


Abbildung 1: Rot-Schwarz-Baum ohne Verletzung von Eigenschaften.

4.1.1 Einfügen in den Rot-Schwarz-Baum

Ein neuer Schlüssel wird zunächst wie in Ref: (normales Einfügen) eingefügt. Zusätzlich werden dann noch die Zeiger auf die Kinder auf den schwarzen Sonderzeiger gesetzt. Der neue Knoten wird rot gefärbt. Durch den neu eingefügten Knoten können Korrekturen notwendig werden, um die Rot-Schwarz-Baum Eigenschaften zu erhalten. Zunächst betrachten wir welche der fünf Eigenschaften betroffen sein können. Es ist immer noch jeder Knoten entweder rot oder schwarz. Aufgrund des Sonderknotens sind die Blätter immer noch schwarz. Da der neue Knoten rot ist, ändern sich die Schwarz-Höhen anderer Knoten nicht. Beide Pfade die von neu eingefügten Knoten starten können haben Schwarz-Höhe eins. Eigenschaft fünf bleibt also erhalten. Eigenschaft vier ist verletzt, wenn der Vater des neuen Knotens ebenfalls rot

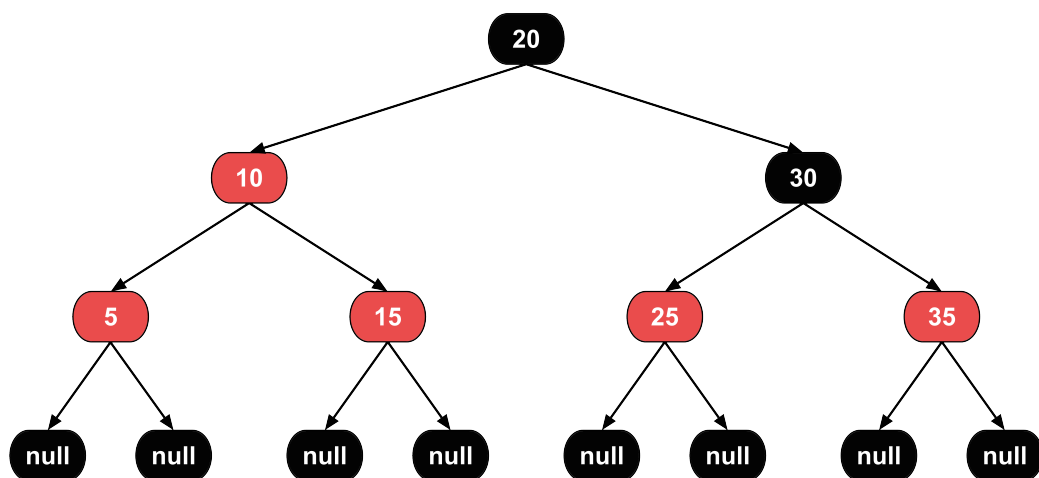


Abbildung 2: Rot-Schwarz-Baum bei dem Eigenschaft 4 verletzt ist.

ist. Eine rote Wurzel erhält man nur wenn in den leeren Baum eingefügt wurde. Es kann also nur eine der beiden Eigenschaften zwei oder vier verletzt sein, aber nicht beide.

Zur Korrektur wird mit einer zusätzlichen Routine gearbeitet. Diese Routine arbeitet sich von der Einfügestelle solange nach oben in einer Schleife durch, bis alle Eigenschaften wieder erfüllt sind. Die Schleifenbedingung ist, dass eine Verletzung vorliegt. Dazu muss einfach geprüft werden ob der betrachtete rote Knoten die Wurzel ist, oder einen roten Vorgänger hat. Beim ersten Durchlauf wird der neu eingefügte Knoten übergeben. Innerhalb der Schleife werden sechs Fälle unterschieden. Im folgenden wird auf vier Fälle detailliert eingegangen. Die restlichen zwei verhalten sich symmetrisch zu einem solchen. Jeder der Fälle verantwortet, dass zum Start der nächsten Iteration wieder nur eine der beiden Eigenschaften zwei oder vier verletzt ist. Eigenschaft eins bleibt in der Beschreibung außen vor, da es während der gesamten Laufzeit der Routine nur Knoten gibt, die rot oder schwarz sind.

Fall 1: Die Wurzel ist rot: Dieser Fall wird behandelt in dem man die Wurzel schwarz färbt. Man muss noch zeigen, dass es durch das Umfärben zu keiner weiteren Verletzung gekommen ist. Bei den Eigenschaften zwei und drei sieht man das direkt. Die Vierte kann nicht verletzt worden sein, da kein roter Knoten hinzugekommen ist. Die Fünfte wurde nicht beeinträchtigt, da die Wurzel bei keiner Schwarz-Höhe eines Knotens beachtet werden muss. Es wird also keine Eigenschaft mehr verletzt und die Schleife wird keine weitere Iteration durchführen.

Die Fälle 2 - 6 behandeln nun die Situation zweier aufeinanderfolgender roter Knoten. Der untere rote Knoten wird als x bezeichnet, der obere als y . Da in

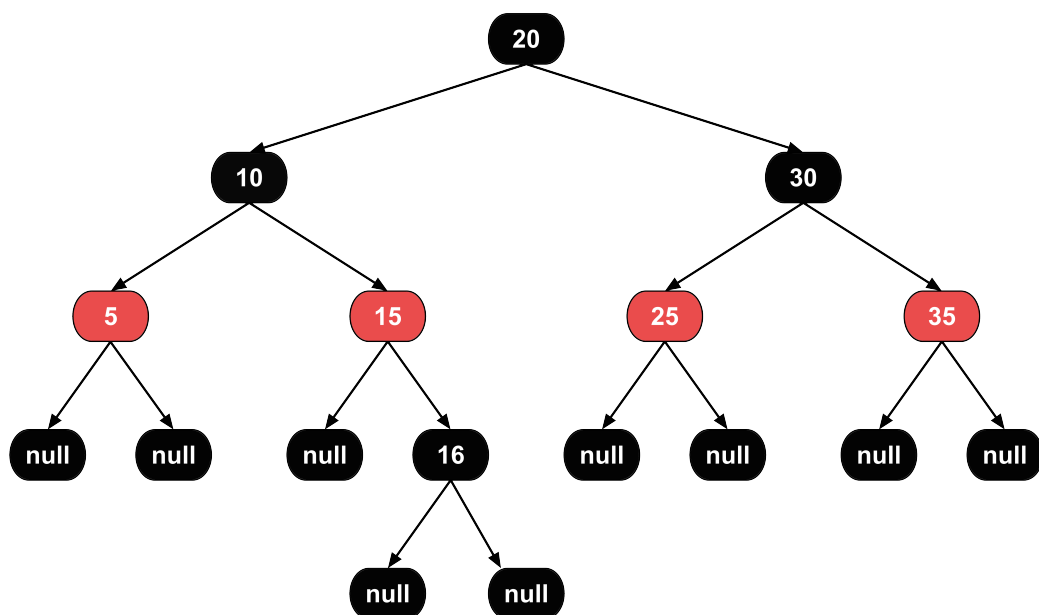


Abbildung 3: Rot-Schwarz-Baum bei dem Eigenschaft fünf verletzt ist.

einen Baum eingefügt wurde, der alle fünf Eigenschaften erfüllt muss y einen Bruder haben. Denn da y rot ist, kann es nicht die Wurzel sein, also muss auch y einen Vater z haben. Von z aus starten über y mindestens zwei Pfade zu schwarzen Blättern. Hätte z nicht auch einen rechten Kind Knoten, wäre bereits vor der Iteration Eigenschaft fünf verletzt gewesen.

Fall 2: Der Bruder von y ist rot: Diesen Fall veranschaulicht Abbildung ???. Da y rot ist, muss z nach Eigenschaft vier schwarz sein. Nun wird z rot gefärbt und beide Kinder von z , also y und dessen Bruder, schwarz. Somit hat x nun einen schwarzen Vorgänger und die Verletzung der Eigenschaft vier wurde an dieser Stelle behoben. Wie sieht es aber mit den Verletzungen insgesamt aus ?

Der rot umgefärbte Knoten z hat zwei Kinder, somit wurde kein Blatt rot gefärbt. Die Schwarz-Höhen der Pfade mit z als inneren Knoten ändern sich nicht, da sie auch genau eines der schwarz umgefärbten Kinder von z enthalten müssen. Die Schwarz-Höhen der Pfade mit z als Startknoten werden um eins verringert. Dies gilt jedoch für jeden von ihnen, so dass Eigenschaft fünf weiterhin nicht verletzt wird. Wenn der rot gefärbte Knoten z nicht die Wurzel ist, könnte er einen roten Vorgänger haben und Eigenschaft vier ist weiterhin Verletzt, das Problem liegt nun aber zwei Baumebenen höher. Wenn z die Wurzel ist kann Eigenschaft vier nicht mehr verletzt sein, allerdings hat man nun eine rote Wurzel. Man ist also in der Situation die nächste Iteration durchführen zu können.

Fall 3: y ist ein linker Nachfolger und sein Bruder ist schwarz. z ist linker Nachfolger:

4.1.2 Löschen aus dem Rot-Schwarz-Baum

4.1.3 Suchen im Rot-Schwarz-Baum

Lemma: Maximale Höhe des Rot-Schwarz-Baum:

Für einen Rot-Schwarz-Baum mit Höhe h und n inneren Knoten gilt $h \leq 2\lg(n+1)$.

Beweis:

Zunächst wird gezeigt, dass ein Teilbaum t_1 mit Wurzel x_1 und Schwarzhöhe $bh(x_1)$ zumindest über $2^{bh(x_1)} - 1$ innere Knoten verfügt, also $n \geq 2^{bh(x_1)} - 1$. Dies wird durch Induktion über die Höhe h_{x_1} gezeigt. Für $h_{x_1} = 0$ besteht t_1 nur aus einem Blatt und enthält keine inneren Knoten. Natürlich gilt in diesem Fall auch $bh(t_1) = 0$.

Induktionsanfang mit $h_{x_1} = 0$:

$$2^0 - 1 = 0$$

Induktionsschritt:

Nun wird ein Teilbaum T_2 mit Wurzel x_2 mit Höhe $h+1$ betrachtet. Jedes seiner beiden Kinder hat entweder Schwarzhöhe $bh(x_2)$, wenn es rot ist, oder $bh(x_2) - 1$, wenn es schwarz ist. Die Höhe beider Kinder ist niedriger als die eigene Höhe von x_2 . Somit kann bei beiden Kindern jeweils die Induktionsannahme für die Mindestanzahl der inneren Knoten eingesetzt werden.

$$2^{bh(x_1)-1} - 1 + 2^{bh(x_1)-1} - 1 = 2^{bh(x_1)} - 2$$

Addiert man einen inneren Knoten aufgrund der Wurzel x_2 hinzu, erhält man die Behauptung.

$$2^{bh(x_1)} - 2 + 1 = 2^{bh(x_1)} - 1.$$

Es gilt also $n \geq 2^{bh(x)} - 1$.

Auf einem Pfad in einem Rot-Schwarz-Baum t_3 von der Wurzel bis zu einem Blatt sind der erste und der letzte Knoten, sowie mindestens jeder zweite der weiteren Knoten schwarz. Es gilt also $bh(t_3) \geq \frac{h(t_3)}{2}$. Damit kann man in der Ungleichung $bh(t_3)$ durch $\frac{h(t_3)}{2}$ ersetzen, woraus dann das Lemma folgt.

$$n \geq 2^{\frac{h(t_3)}{2}} - 1 \Rightarrow n+1 \geq 2^{\frac{h(t_3)}{2}} \Rightarrow \lg(n+1) \geq \frac{h(t_3)}{2} \Rightarrow 2\lg(n+1) \geq h(t_3)$$

Literatur

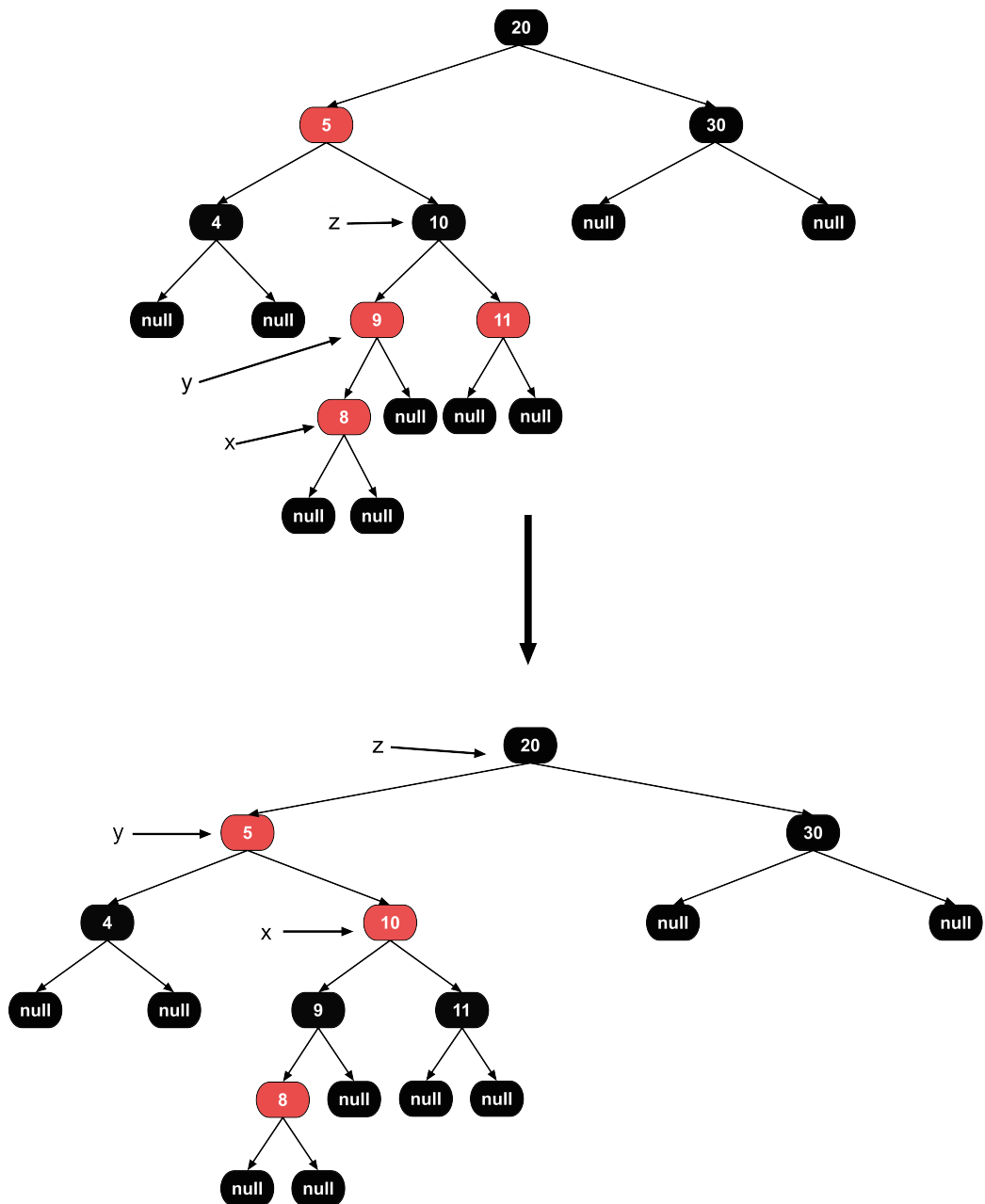


Abbildung 4: Einfügen Korrektur. Dargestellt ist eine Iteration mit Fall 2

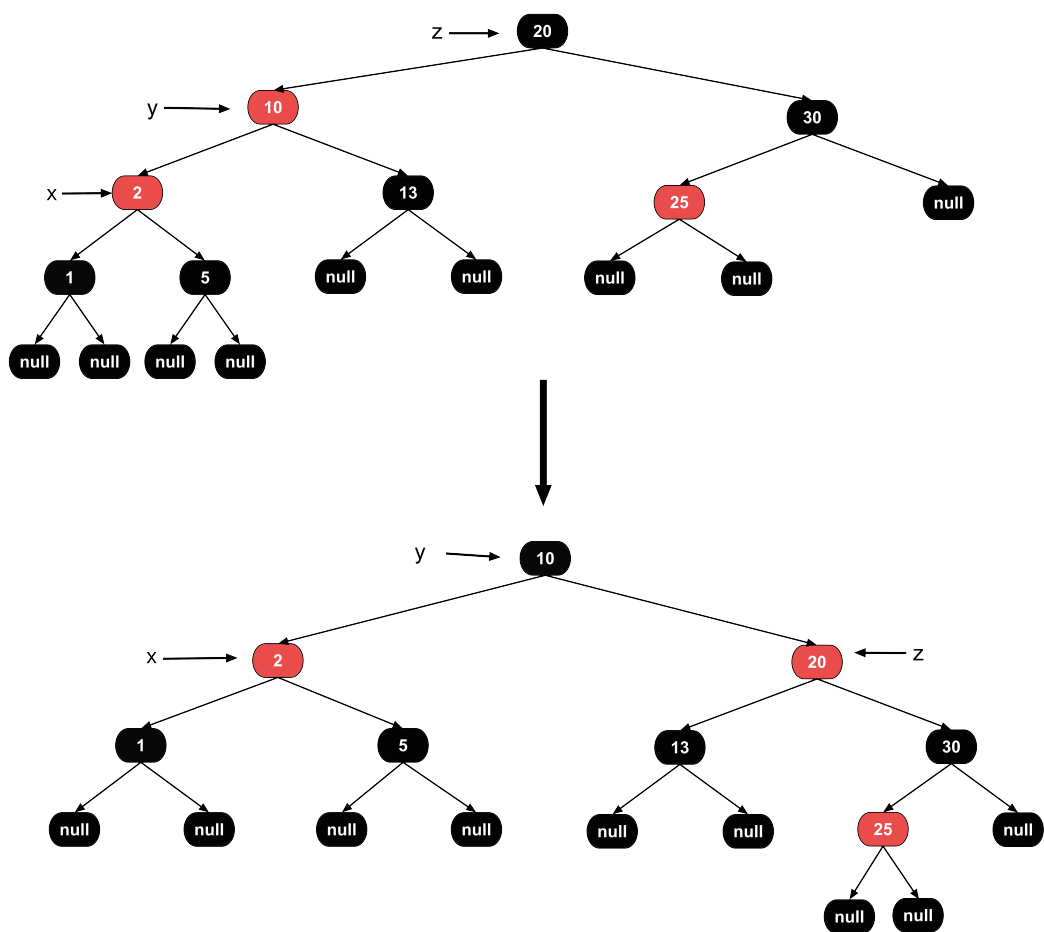


Abbildung 5: Einfügen Korrektur. Dargestellt ist Fall 3