

Ausarbeitung

Algorithmus zum (hübschen) Zeichnen von binären Bäumen

Kurs „Wissenschaftliches Arbeiten in der Informatik“

Niklas Hartinger

THM*

29.04.2019

Zusammenfassung

Diese Ausarbeitung befasst sich mit der Aufgabe des ästhetischen Zeichnens eines Binärbaums. Dafür wurde der Reingold-Tilford-Algorithmus verwendet.

*FB MNI Mathematik, Naturwissenschaften, Informatik

Inhaltsverzeichnis

1 Einführung und Aufbau	2
2 Ästhetischer binärer Baum	2
2.1 Gegenbeispiel unästhetischer Binärbaum	3
3 Reingold-Tilford-Algorithmus	4
3.1 Linke und rechte Kontur	4
3.2 Die grundlegende Idee	5
3.3 Optimierung	5
3.4 Diskussion	6
4 Anwendung	7
5 Fazit	12

1 Einführung und Aufbau

Eine ästhetisch gezeichnete Datenstruktur hilft dem menschlichen Betrachter bei Verständnis und Übersicht der gezeichneten Struktur. In diesem Artikel geht es um das „schöne“ Zeichnen eines binären Baumes. Dazu wird zuerst erklärt, wie ein ästhetischer binärer Baum definiert ist. Setzt ein Algorithmus diese Zeichenregeln nicht um, führt das zu binären Bäumen, die für den Betrachter unübersichtlich und ungeordnet wirken können. Dies wird in diesem Artikel durch ein Beispiel von einem Algorithmus von Donald Knuth verdeutlicht. Der Reingold-Tilford-Algorithmus setzt die ästhetischen Regeln deutlich besser um. Dessen Eigenheiten und Probleme werden erklärt. Die Funktionsweise des Reingold-Tilford-Algorithmus wird anhand eines ausführlichen Beispiels erläutert. Der Artikel endet mit einem Fazit.

2 Ästhetischer binärer Baum

Reingold und Tilford [1] formulieren folgende Kriterien für die „Ästhetik“ einer Baum-Visualisierung:

- Vaterknoten sind zentriert über den Kindknoten angeordnet
- Knoten und Kanten dürfen sich nicht überschneiden
- linke Kindknoten müssen links vom Vaterknoten und rechte Kindknoten rechts vom Vaterknoten angeordnet werden
- strukturell gleiche Teilbäume müssen immer gleich gezeichnet werden

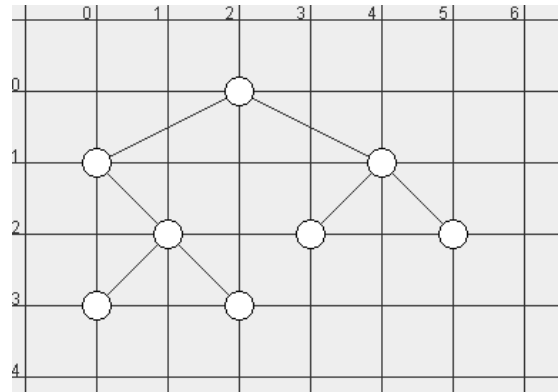


Abbildung 1: Ästhetisch gezeichneter Binärbaum¹

Die Einhaltung dieser Regeln führt wie in Abbildung 1 zu einem platzsparenden, übersichtlichen Binärbaum und der Betrachter erkennt leicht strukturell gleiche Teilbäume.

2.1 Gegenbeispiel unästhetischer Binärbaum

Donald Knuth veröffentlichte 1971 einen Algorithmus, für das Zeichnen eines Binärbaums [2]. Dieser Algorithmus arbeitet nach dem einfachen Prinzip, dass die y-Koordinate jedes Knotens dessen Tiefe im Baum repräsentiert und die x-Koordinate in einem In-Order-Verfahren von Null aufsteigend den Knoten zugeordnet wird.

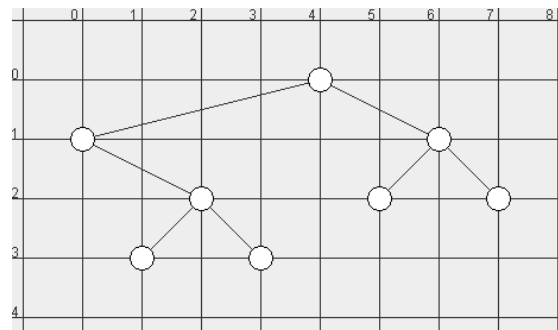


Abbildung 2: Unästhetisch gezeichneter Binärbaum

Damit ergibt sich ein Baum, der nicht alle Ästhetikregeln erfüllt. Das Ignorieren der Regeln führt zu einem verzerrt wirkenden binären Baum. Im Vergleich zu demselben Baum aus Abbildung 1 benötigt der Baum aus Abbildung 2 auch deutlich mehr Platz. Ein Algorithmus, der die Ästhetikregeln besser umsetzt, ist der Reingold-Tilford-Algorithmus.

¹Alle Abbildungen wurden mit einem eigenen Java-Programm erstellt. Code einsehbar in <https://github.com/nhrt/Reingold-Tilford>

3 Reingold-Tilford-Algorithmus

Der Reingold-Tilford Algorithmus wurde im März 1981 von Edward M. Reingold und John S. Tilford veröffentlicht [1]. Er beschreibt ein Divide-and-Conquer-Vorgehen um binäre Bäume ästhetisch zu zeichnen. Eine konkrete Anwendung war bereits in Abbildung 1 zu sehen.

3.1 Linke und rechte Kontur

Im Fall eines Baumes besteht die Kontur aus den Knoten, die sich auf der jeweiligen Baumtiefe am weitesten links für die Linkskontur bzw. rechts für die Rechtskontur befinden. In Abbildung 3 ist ein Baum zu sehen, bei dem die Konturen der Teilbäume hervorgehoben sind. Die Links- und Rechtskontur ist zum Zeichnen mit dem Reingold-

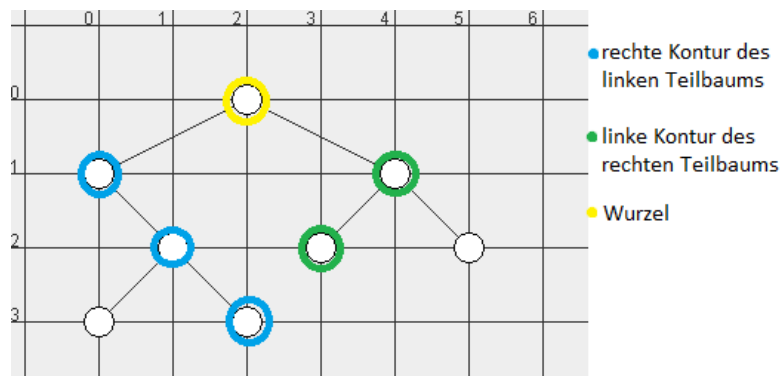


Abbildung 3: Binärbaum mit Konturen

Tilford-Algorithmus wichtig, weil dieser mit den Informationen der Konturen eine Überschneidung von Teilbäumen verhindert [1].

3.2 Die grundlegende Idee

Der Algorithmus unterscheidet zwischen Blattknoten und inneren Knoten. Von Blattknoten ausgehend kann es keine Überschneidungen mehr geben, weil von diesem keine Teilbäume mehr ausgehen. Daher ist die Berechnung der Koordinaten trivial. Betrachtet man einen Knoten, dessen Kindknoten bereits mit dem Reingold-Tilford-Algorithmus berechnet wurden, kommt es bei dessen Teilbäumen zu keinen Verletzungen der Ästhetikregeln. Nun müssen die Teilbäume so nah wie möglich aneinander platziert werden, ohne Überschneidungen zu erzeugen. Das wird dadurch gewährleistet, dass man die linke Kontur des rechten Teilbaums und die rechte Kontur des linken Teilbaums betrachtet. Kommt es zwischen diesen Knoten und Kanten zu keinen Regelverletzungen, so ist dies auch bei der restlichen Struktur der Fall. Anschließend wird der Vaterknoten zentriert über den Kindknoten platziert. Dieses Vorgehen wird in einem Post-Order-Verfahren für jeden Knoten der Baumstruktur durchgeführt.

3.3 Optimierung

Dadurch, dass Konturen nicht zusammenhängend sein müssen, wie in Abbildung 4 zu sehen ist, hat das zur Folge, dass der Algorithmus für jede Konturberechnung den gesamten Baum durchlaufen müsste. Um die Laufzeit deutlich zu verbessern, nutzten Reingold und Tilford temporäre Kanten, um zusammenhängende Konturen zu erzeugen. Im Originalartikel werden diese temporären Kanten „Threads“ genannt. Diese Kante geht von dem untersten Knoten der Kontur des kürzeren Teilbaums, zu dem nächst tieferen Knoten der Kontur des längeren Teilbaums. Da die unterste Kante des kürzeren Teilbaums immer ein Blattknoten ist, können dessen nicht verwendete Kind-Zeiger zum Speichern des Threads verwendet werden. Dieser Knoten muss als Thread gekennzeichnet werden, um die temporären Kanten wieder entfernen zu können. So sieht man in Abbildung 5, dass die rechte Kontur zusammenhängend ist. Anschließend ist ein einfaches Traversieren der Kontur möglich. Dieses Verfahren muss immer angewendet werden, wenn zwei Teilbäume aneinander gezeichnet werden.

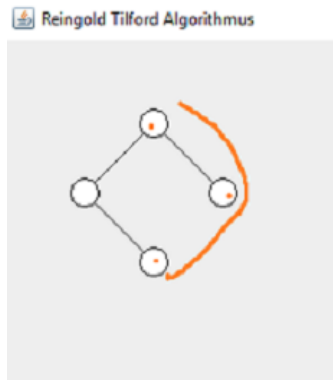


Abbildung 4: Binärbaum mit nicht zusammenhängender rechter Kontur

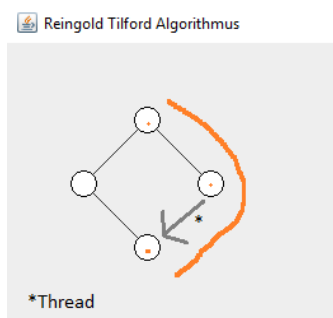


Abbildung 5: Kontur mit Thread

3.4 Diskussion

Der Reingold-Tilford-Algorithmus erzeugt nicht immer eine Zeichnung mit der minimalen Breite. Dies ist daraus geschuldet, dass Teilbäume nach den Ästhetikregeln bei gleicher Struktur gleich gezeichnet werden müssen. Bei dem Vergleich von Abbildung 6 und Abbildung 7 wird deutlich, dass unter Einbeziehung von Kontextinformationen bei der Verschiebungsberechnung, ein schmalere Baum bei bestimmten Baumstrukturen möglich wäre.

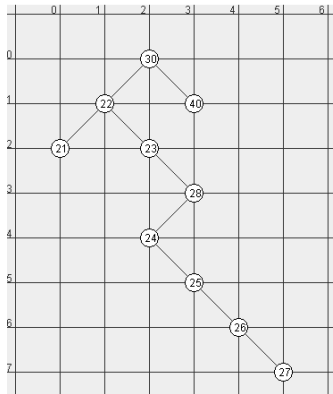


Abbildung 6: Baum gezeichnet mit dem R.-T.-Algorithmus

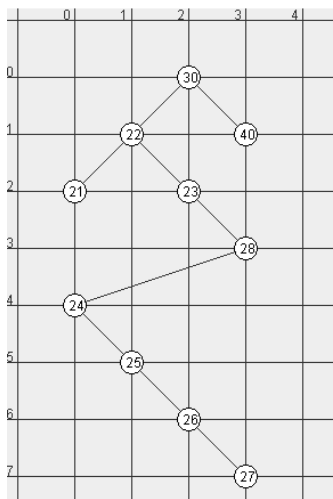


Abbildung 7: Baum mit optimaler Breite

4 Anwendung

Die noch nicht gezeichnete Baumstruktur aus Abbildung 8 soll mit dem Reingold-Tilford-Algorithmus gezeichnet werden.

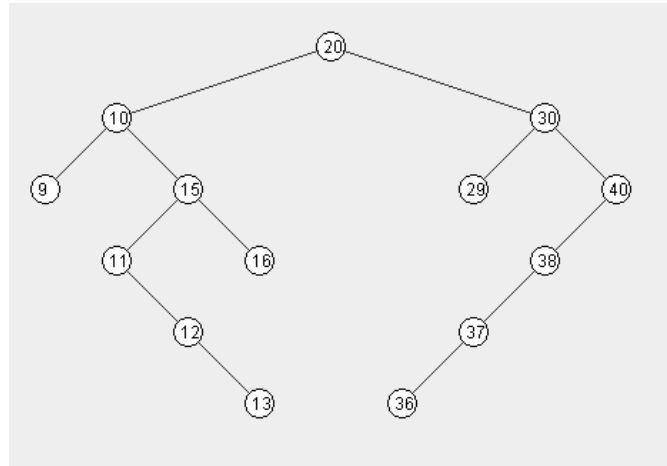


Abbildung 8: Anwendungsbeispiel

Die y-Koordinate entspricht wie schon beim Knuth-Algorithmus der Tiefe des jeweiligen Knotens. Es wird aus diesem Grund vor allem auf die Verschiebung und daraus resultierende x-Koordinate eingegangen. Die minimale Verschiebung von Ebene zu Ebene ist 1 [1]. Somit ergibt sich ein minimaler Abstand von Knoten auf einer Ebene von 2. Nur wenn dieser Abstand unterschritten wird, muss die Verschiebung dementsprechend erhöht werden. Da der Algorithmus in Post-Order-Reihenfolge durchlaufen wird, ist der erste Knoten, den der Algorithmus auswertet, der Knoten mit dem Wert 9. Dieser Knoten ist ein Blattknoten. Der Algorithmus speichert nun die Information, dass Knoten 9 im linken Teilbaum von Knoten 10 in der rechten und linken Kontur ist. Darüberhinaus kommt es im linken Teilbaum von Knoten 10 zu keinen Überschneidungen. Anschließend werden die Knoten 11, 12, 13 und 16 analog ausgewertet. Bis zu diesem Zeitpunkt ist eine Überschneidung noch nicht möglich, weil jeder Knoten nur maximal ein Kind hatte. Interessant wird es nun bei Knoten 15.

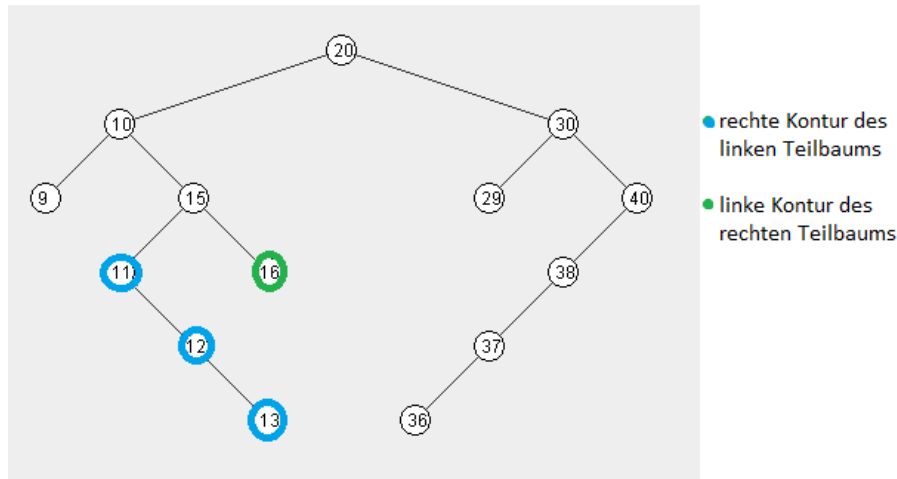


Abbildung 9: Anwendungsbeispiel

Da von diesem Knoten zwei Teilbäume ausgehen kann es hier das erste Mal zu einer Überschneidung kommen. Durch die vorherigen Auswertungen kennt Knoten 15 die rechte Kontur seines linken Teilbaums und die linke Kontur seines rechten Teilbaums. Dies ist in [Abbildung 9](#) hervorgehoben. Mit diesen Informationen muss Ebene für Ebene geprüft werden ob ein Mindestabstand zwischen den Konturknoten gegeben ist. Der erste Knoten in der rechten Kontur des linken Teilbaums ist Knoten 11 und der erste Knoten in der linken Kontur des rechten Teilbaums ist Knoten 16. Knoten 11 und 16 werden jeweils um die Standardverschiebung verschoben. Der Abstand ist somit 2. Dieser Abstand unterschreitet den Minimumabstand nicht und die Verschiebung muss somit nicht erhöht werden. Es gibt keine weiteren Knoten in der linken Kontur des rechten Teilbaums, somit kann es auch zu keinen weiteren Überschneidungen kommen. Da der linke und der rechte Teilbaum eine unterschiedliche Tiefe haben, müssen wie schon bei Abschnitt „Optimierung“ beschrieben, die Konturen des neuen Baums bestimmt werden. Da der rechte Teilbaum kürzer ist, muss eine Verbindung zwischen Knoten 16 und Knoten 12 hergestellt werden. Die Kante wird in Knoten 16 gespeichert. Nun ergibt sich ein neuer Teilbaum wie in [Abbildung 10](#) zu sehen ist.

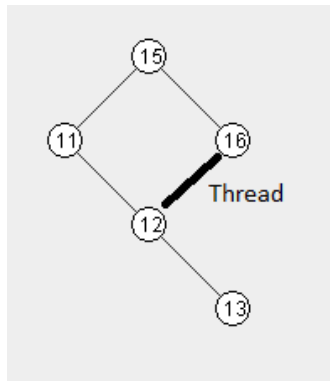


Abbildung 10: Anwendungsbeispiel

Führt man diese Prozedur für die beiden Teilbäume von Knoten 20 durch, ergeben sich zwei Teilbäume, in denen keine Überschneidungen vorliegen. Bis zu diesem Zeitpunkt konnte noch kein Nutzen aus den temporären Kanten gezogen werden. Die neue Struktur des Baumes ist in Abbildung 11 zu sehen. Um die rechte Kontur des linken Teilbaums und die linke Kontur des rechten Teilbaums zu traversieren sind die Threads erforderlich.

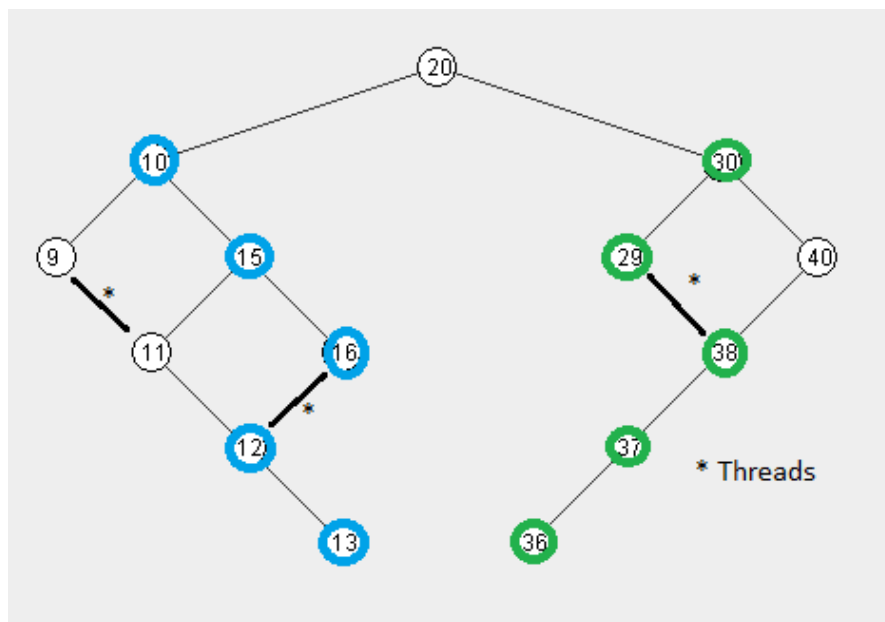


Abbildung 11: Anwendungsbeispiel

Versucht man nun die beiden Teilbäume zu platzieren, fällt auf, dass der Mindestabstand zwischen Knoten 15 und 29 unterschritten wird. Also muss die Verschiebung der Teilbäume um 1 erhöht werden. Vergleicht man die Konturen nun weiter, wird der Mindestabstand bei Knoten 13 und 36 wiederum unterschritten. Somit ergibt sich eine Verschiebung der Teilbäume um 3 und es ist sichergestellt, dass der gesamte Baum keine Überschneidungen aufweist. Um anschließend absolute Koordinaten zu erhalten, wird der Baum in Pre-Order-Reihenfolge durchlaufen und die x-Koordinaten aller Knoten berechnet. Bei diesem Durchlauf werden alle Threads entfernt. Die Threads waren zur Abstandsberechnung notwendig, würden aber zu falschen Kanten führen. Verschiebt man den Baum nun in den Bereich mit positiven x-Koordinaten ergibt sich die Baumzeichnung aus Abbildung 12.

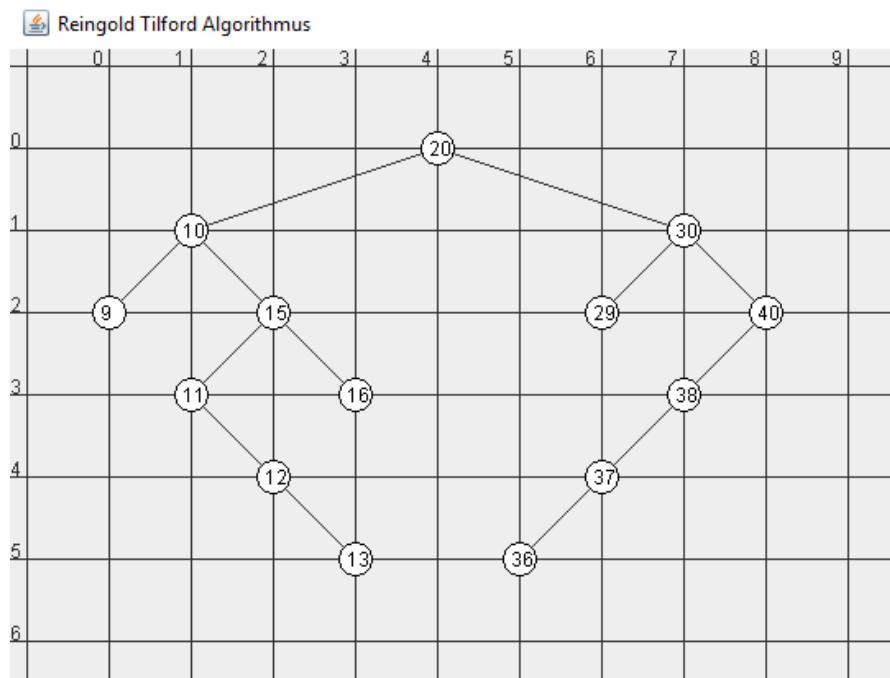


Abbildung 12: Anwendungsbeispiel

Zu beobachten ist, dass alle Ästhetikregeln eingehalten sind.

5 Fazit

Diese Ausarbeitung diskutierte das Vorgehen zum „hübschen“ Zeichnen eines binären Baumes. Dabei ging es hauptsächlich um die Einhaltung der Ästhetikregeln. Für diese Aufgabe hat sich der Reingold-Tilford-Algorithmus sehr gut geeignet. Er war in der Lage einen Binärbaum nach allen vorgegebenen Regeln zu Zeichnen. Die eigentliche Idee des Algorithmus ist leicht zu verstehen. Jedoch führt die Optimierung dazu, dass der Algorithmus schwierig zu implementieren ist. Darüber hinaus gibt es Baumstrukturen, die sich mit dem Reingold-Tilford-Algorithmus nicht optimal zeichnen lassen. Ein Algorithmus, der die Gesamtstruktur des Baumes beachtet und nicht nur die Konturen der Teilbäume, könnte zu einem noch besseren Zeichenergebnis führen.

Literatur

- [1] E. M. Reingold und J. S. Tilford, *Tidier Drawings of Trees*, IEEE Transactions of Software Engineering, Vol. SE-7, Nummer 2, <https://reingold.co/tidier-drawings.pdf>, 1981.
- [2] D. E. Knuth, *Optimum binary search trees*, Acta Informatica 1, 14-25, Springer-Verlag, 1971