Spielbäume

oder

Tic-Tac-Toe like a Pro

Proseminar Theoretische Informatik

Joschka Heinrich[*], TU Dresden

3. Februar 2018

Zusammenfassung

Was sind Spielbäume und wozu können sie verwendet werden? In dieser Ausarbeitung wird das Konzept des Spielbaumes formal eingeführt und anhand des Spiels *Tic-Tac-Toe* beispielhaft erläutert. Mit der Erklärung des Alpha-Beta-Prunings als Optimierung des Minmax-Verfahrens und anderer Heuristiken wird eine Anwendung von Spielbäumen illustriert.

I. Einführung

Motivation. Überblick Aufbau Text.

II. Spielbäume

Um im Folgenden mit Spielbäumen arbeiten zu können, führen wir das Konzept zunächst formal ein und definieren dazu unter anderem den Begriff des *Spiels*, der *Konfiguration* und des *Spielbaums* selbst.

i. Definition

Alle folgenden Betrachtungen nehmen wir aus der Perspektive eines gewinnorientierten Spielers A vor, der sich einer Anzahl Gegner gegenüber sieht. Es ist also das Ziel, Züge für A so zu finden, dass dessen Gewinn maximiert, bzw. der Gewinn der Gegner minimiert wird, wobei wir davon ausgehen, dass alle Gegner optimale Entscheidungen treffen. Wir vereinfachen die

[*]joschka.heinrich@tu-dresden.de

Betrachtung von Spielen, indem wir uns auf solche ohne Zufallskomponente, d.h. reine Strategiespiele mit vollkommener Information und Spiele mit zwei Kontrahenten—Spieler A und Spieler B—beschränken. Mit „der Gegner" ist also im Folgenden stets Spieler B gemeint.

Ein **Spiel** $S = (R, a, E)$ ist nun durch Regeln, in Form einer endlichen Menge von legalen Spielzügen R, eine Anfangskonfiguration $a \in K$ und eine Reihe möglicher Endkonfigurationen $E \subset K$ gegeben, mit K, der Menge aller *zulässigen* Konfigurationen. Eine **Konfiguration** $k \in K$ repräsentiert dabei einen möglichen Zustand des Spieles, bestehend aus einer Beschreibung wiederum der Zustände aller relevanten Spielelemente (bspw. die Position der Zeichen auf dem Tic-Tac-Toe-Feld) inklussive des Spielers, der als nächster an der Reihe ist (bei uns entweder A oder B).

In Abgrenzung zur Menge der *legalen* Spielzüge können wir uns beliebige andere Spielzüge vorstellen, die zwar möglich, allerdings in dem betrachteten Spiel nicht erlaubt sind. Analog dazu sind über K hinaus weitere Konfigurationen denkbar, die allerdings nicht zulässig sind, d.h. in einem regelkonformen Spiel niemals auftreten können. Durch Anwenden eines legalen Spielzuges auf eine Konfiguration gelangen wir zu einer neuen Konfiguration. Ein legaler Spielzug kann also als eine Funktionen $r : K \to K$ verstanden werden. Seien $u, v \in K$ Konfigurationen und $r \in R$ ein legaler Spielzug, mit $v = r(u)$, dann heißt v **Folgekonfiguration** von u (bezüglich r) und u **Elternkonfiguration** von v (bezüglich r).

Wenn das Anwenden eines Spielzuges zu einer neuen Konfiguration führt, also $u \neq v$ gilt, dann heißt r *anwendbar* auf u.[1] Gibt es mehrere auf eine Konfiguration $k \in K$ anwendbare Spielzüge, erhalten wir eine Menge von Folgekonfigurationen $F(k) \subset K$ mit $F(k) = \{r(k) \mid r \in R,\ r \text{ anwendbar auf } k\}$. Auf eine Endkonfiguration $k_e \in E$ sind keine Spielzüge anwendbar, da das Spiel mit Erreichen einer dieser Konfigurationen als beendet gilt: $F(k_e) = \varnothing$.

Die Menge K können wir nun induktiv über die Folgekonfigurationen definieren:

- a ist Element von K.
- Wenn $k \in K$, dann auch alle $k' \in F(k)$.

Alle zulässigen Konfigurationen lassen sich also aus der Anfangskonfiguration und den legalen Spielzügen ableiten. Zu jeder Konfiguration $k \in K \setminus E$ gehört eindeutig eine Menge von Folgekonfigurationen $F(k)$ und zu jeder Konfiguration $k \in K \setminus a$ eine Elternkonfiguration k'. Diese Beziehungen können durch einen Graphen anschaulich dargestellt werden.

Ein **Spielgraph** ist ein gerichteter Graph $G(V, E)$ mit:

- Knoten $V = K$ und
- Kanten $E = \bigcup\limits_{u \in K} \{(u, v) \mid v \in F(u)\}$

Es ergibt sich insbesondere mit dem Graphen G ein Baum, mit a als Wurzel und E als Blätter. So kann über den Graphen entlang legaler Spielzüge traversiert werden und bspw. ein kompletter Spielverlauf mit n Zügen als Pfad $(k_0 = a, k_1, \ldots, k_n \in E)$ subsequenter Folgekonfigurationen dargestellt werden.

Als zusätzliche Vereinfachung schließen wir aus, dass eine über R aus a generierte Konfiguration durch erneutes Anwenden legaler Spielzüge wieder erreicht werden kann, dass wir also im weite-

[1] Das bedeutet nicht notwendigerweise, dass sich die Konfiguration der Spielelemente verändert. Zwei Spielkonfigurationen können sich auch darin unterscheiden, welcher Spieler an der Reihe ist.

ren Spielverlauf zu einer Situation gelangen, die bereits aufgetreten ist.[2] Daraus folg unmittelbar, dass G zyklenfrei ist.

ii. Am Beispiel Tic-Tac-Toe

Um obige Definitionen zu veranschaulichen, werden wir sie nun auf das Spiel Tic-Tac-Toe anwenden. Das Tic-Tac-Toe-Spielfeld besteht aus einem 3×3-Raster mit neun Feldern in die zwei Spieler abwechelnd ihre Zeichen setzen. Dies sind üblicherweise „o“ bzw. „x“, daher werden wir die Spieler im Folgenden nicht mit A und B, sondern als „Spieler O“ und „Spieler X“ bezeichnen.

Ziel jedes Spielers ist es, drei seiner Zeichen nebeneinander zu setzen, d.h. in einer Reihe, Spalte oder Diagonale, und gleichzeitig zu verhindern, dass der Gegner seine Zeichen in dieser Weise setzen kann. Das Spiel endet entweder, wenn einer der beiden Spieler gewinnt, sobald er dieses Ziel erreicht, oder wenn kein Zug mehr möglich ist, da alle neun Felder belegt sind, dann geht das Spiel unentschieden aus.

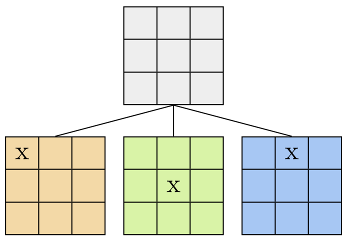

Abbildung 1: Ein beispielhafter Spielbaum (sogar mit Farben!)

Anwendung der Definition.

III. Zugplanung

Kapitel in einem Satz.

i. Minmax-Verfahren

ii. Alpha-Beta-Pruning

IV. Zusammenfassung

Schlussfolgerung. Ausblick. Anwendbarkeit, Zufallsspiele

[2] Diese Einschränkung schließt viele Spiele wie bspw. Schach von der folgenden Betrachtung aus, da dort durch legale Spielzüge Konfigurationen reproduziert werden können. Viele der Aussagen lassen sich dennoch auch auf diese Art Spiele übertragen.

Literatur

[Klüppelholz, 2016] Sascha Klüppelholz. Entwurfs- und Analysemethoden für Algorithmen – Skript zur Vorlesung SS 2016