Informatik II

Übungsstunde 9

simon.mayer@inf.ethz.ch
Distributed Systems Group, ETH Zürich

Ablauf

- Besprechung der Vorlesung
- Uebungsbezogene Themen:
  Spielbaumauswertung
- Zeit zum Programmieren...
  und fuer noch mehr Fragen

Beispiel: Das Knobelspiel „Schere, Stein, Papier"

- Zwei Spieler formen gleichzeitig mit einer Hand ein Symbol („Schere", „Stein" oder „Papier")
  - Spiel besteht nur aus einem einzigen (Doppel)-Zug

- Regel:
  - Schere schlägt Papier (schneidet es)
  - Stein schlägt Schere (macht sie stumpf)
  - Papier schlägt Stein (wickelt ihn ein)

- Auszahlungsmatrix:

	Schere	Stein	Papier
Schere	0	1	-1
Stein	-1	0	1
Papier	1	-1	0

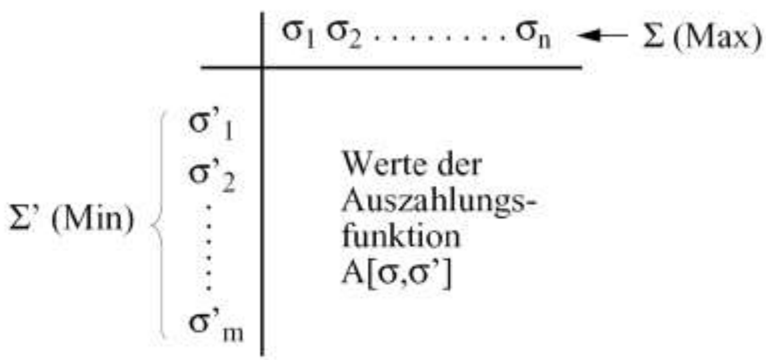
- Jeder Spieler kann zwischen 3 Strategien Schere, Stein, Papier (für seinen einzigen Zug!) wählen
- Gibt es eine „beste" Strategie?

Maximierung des Minimalgewinns

- Annahme: Gegner spielt optimal
  - Nutzt jeden Fehler schonungslos aus
  - Lässt sich nicht bluffen
  - Weiss bestens Bescheid

Ist in der Praxis vielleicht eine übertrieben pessimistische Annahme!

- Was ist dann die beste eigene Strategie?
  - → Möglichst vorsichtig und risikolos spielen!



Optimale Strategien

- Satz (o. Bew.): *Für endliche rein strategische 2-Personen Nullsummenspiele mit vollständiger Information gilt $G = G'$*
- Die zugehörigen Strategien σ^* bzw. σ'^* heissen optimale Strategien (für Max bzw. Min)
- Das Paar (σ^*,σ'^*) bildet einen Gleichgewichtspunkt:
  - Es gilt: $A[\sigma^*,\sigma'^*] \geq A[\sigma,\sigma'^*]$ für alle σ von Max
  - D.h.: wenn Min seine optimale Strategie σ'^* anwendet, kann Max nichts sinnvolleres tun, als ebenfalls seine optimale Strategie σ^* anzuwenden – entsprechendes gilt auch umgekehrt

Eine Situation, bei der sich kein Spieler verbessern kann, indem (nur) er von der Strategiekombination abweicht, heisst auch Nash-Gleichgewicht (Theorie 1951 von John Nash, dafür 1994 Nobelpreis für Wirtschaftswissenschaften)

Gewinnstrategie und Gewinnstellung

- Die Blätter des Spielbaums seien mit >0 („Max gewinnt") bzw. <0 („Min gewinnt") markiert; mit 0 bei unentschieden
- Eine Strategie heisst Gewinnstrategie (für Max), wenn damit auf jeden Fall ein Wert >0 erreichbar ist:

$$A[\sigma^*, \sigma'^*] = \begin{cases} > 0 & \rightarrow \sigma^* \text{ ist Gewinnstrategie für Max} \\ < 0 & \rightarrow \sigma'^* \text{ist Gewinnstrategie für Max} \\ = 0 & \rightarrow \text{bei optimalem Spiel kann unent-} \\ & \quad \text{schieden erzwungen werden} \end{cases}$$

- Gewinnstellung (oder „sichere Position"):
  - Der Gegner kann nur noch gewinnen, wenn man selbst einen Fehler macht (in eine „unsichere" Position zieht)
  - Aus einer eigenen Gewinnstellung kann man nach einem beliebigen Gegenzug wieder eine Gewinnstellung erreichen

Minimax-Algorithmus

- Eine optimale Strategie kann effizienter als nach der Definition (Enumeration aller Strategien) gefunden werden
- Sei γ die Auszahlungsfunktion für Blätter; dann definiere den Minimaxwert v(k) eines Knotens k so:

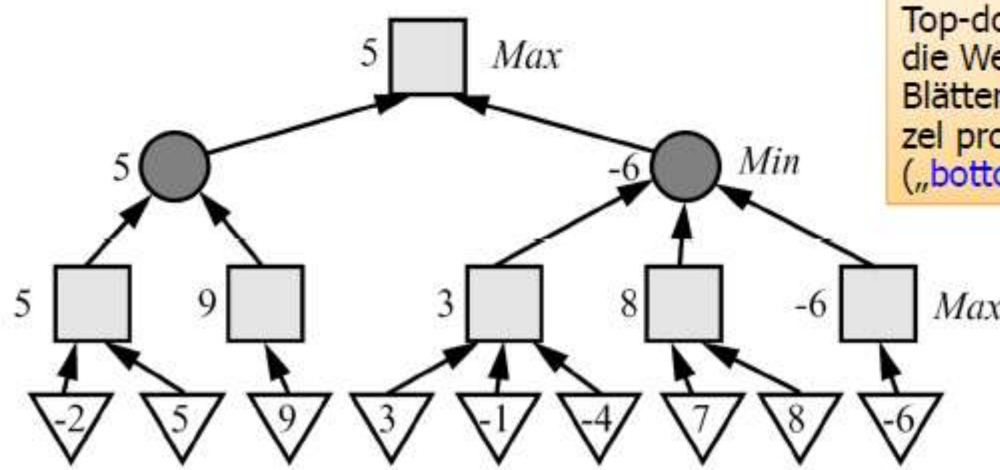
$$v(k) = \begin{cases} \gamma(\mathrm{k}) & \text{, falls k ein Blatt ist} \\ \max\{\mathrm{v(n)} \mid \text{n ist direkter Nachfolger von k}\} & \text{, falls k innerer Max–Knoten ist} \\ \min\{\mathrm{v(n)} \mid \text{n ist direkter Nachfolger von k}\} & \text{, falls k innerer Min–Knoten ist} \end{cases}$$

- Die rekursive Definition lässt sich direkt in einen entsprechenden rekursiven Algorithmus umsetzen

Minimax-Algorithmus

Es wird von zwei Seiten am Werte ... hin und her gezerrt, nämlich durch S1, der ihn möglichst groß, und durch S2, der ihn möglichst klein machen will.
John von Neumann, 1928

Bottom-up-Minimax-Algorithmus



Alternativ zum rekursiven Top-down-Ansatz können die Werte auch von den Blättern in Richtung Wurzel propagiert werden („bottom up")

- Interpretation:
  - Die Wurzel erhält den Wert, den Max mindestens erreichen kann
  - Max wählt Alternative zum grössten direkten Nachfolger
  - Min wählt Alternative zum kleinsten direkten Nachfolger

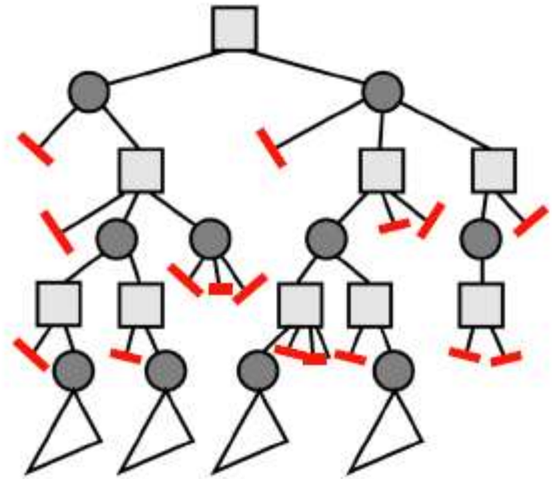
Optimale Strategie

Spielbaumauswertung

- Depth-First
- Breadth-First
- Best-First
  - Mit Schaetzwert...
- Partielle Auswertung
  - Auswerten nur bis zu gewisser Tiefe, dort Schaetzung des Wertes

Schlanke Spielbäume

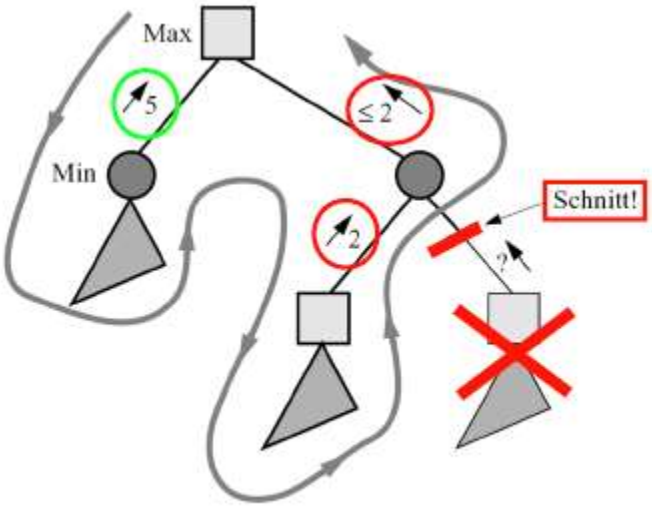
- Man versucht möglichst, Spielbäume schlank zu halten
  - Erlaubt eine grössere Tiefe bzgl. der interessanten Knoten
  - Erfordert aber eine Bewertung aller „Geschwister" oder Nachkommen, bevor man tiefer absteigt, um nur die interessantesten weiterzuverfolgen (→ Bestensuche)



- Menschen spielen offenbar sehr stark selektiv
- Vorsicht: man kann sich bei der Bewertung täuschen und damit interessante Züge „irrtümlich" abschneiden!

Baumschnitte

- Man kann oft einige Knoten / Unterbäume abschneiden, die den Minimaxwert garantiert nicht beeinflussen
- Ziel: Möglichst früh solche verzichtbaren Unterbäume erkennen und möglichst viele davon abschneiden



Begründung für den Schnitt: Max wird von rechts einen Wert ≤ 2 erhalten und daher sowieso den linken Zug (5) wählen; den Wert des abgeschnittenen Unterbaums braucht man also nicht zu kennen!

Wie erkennt man Schnittmöglichkeiten systematisch?

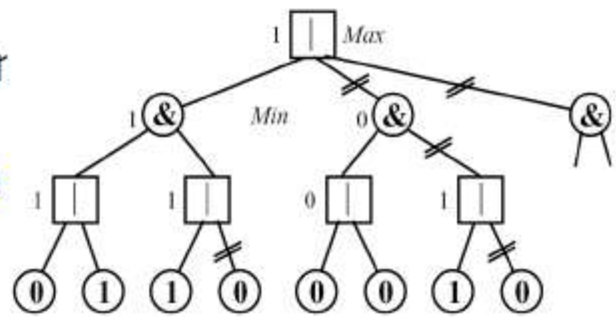
Wie maximiert man die Zahl der Schnitte?

Der Alpha-Beta-(„α-β“)-Algorithmus

- Reduziert den Spielbaum systematisch durch Schnitte, aber liefert den gleichen Minimaxwert der Wurzel wie der eigentliche Minimax-Algorithmus

- Tiefensuche, wobei Knoten nur dann expandiert werden, wenn sie (nach bisheriger Information) den Minimaxwert beeinflussen können

- Man vergleiche dies mit der Shortcut-Auswertung boolescher Operatorbäume



Definition α-β-Schranken

- **α-Schranke**: Wert, den Max (bei optimalem eigenen Spiel) bei den bisher untersuchten Zügen auf jeden Fall erzielen kann
  - Ist eine monoton wachsende untere Schranke für den Gewinn
  - Ist relevant für Min-Knoten: Evaluierung der weiteren Nachfolger kann abgebrochen werden, sobald der berechnete Rückgabewert unter α fällt („α-Schnitt")
- **β-Schranke**
  - Analog (bzgl. Min)
- **α-β-Schranken** bilden zusammen ein Suchfenster (α,β), das mit $(-\infty,+\infty)$ initialisiert wird und im Laufe der Baumauswertung schrumpft
  - Erlaubt damit sukzessive mehr Schnittmöglichkeiten!

Ablauf

- Besprechung der Vorlesung
- Uebungsbezogene Themen:
  Spielbaumauswertung
- Zeit zum Programmieren...
  und fuer noch mehr Fragen

Übung 9

1. Das allgemeine Rucksackproblem (KSP)
   - K Gegenstaende x1, ..., xK; Jeweils bekannter Wert und Gewicht
   - Auswahl von Gegenstaenden, sodass Gesamtgewicht nicht ueberschritten wird
   - Optimierungsproblem: Maximieren des Wertes der ausgewaehlten Gegenstaende

   a) Theorie
   b) Brute Force Ansatz
   c) Backtracking Ansatz
   d) Vergleich von Brute Force und Backtracking

Übung 9

2. Spieltheorie/Spielbaumauswertung

   a) Bisschen Theorie

   b) Minimax-Algorithmus

   c) Optimale Strategie fuer MAX-Spieler

   d) Alpha/Beta-Algorithmus

Übung 9

3. Reversi

a) Reversi-Spieler mit MIN/MAX-Algorithmus
   Konfigurierbare Suchtiefe!

b) Anpassen des Spielers an Zeitbegrenzung
   Suchtiefe immer weiter erhoehen
   Abbruch bei Ablauf der Zeit (Timeout-Exception...)

c) Literaturrecherche fuer bessere Bewertungsfunktion

Ablauf

- Besprechung der Vorlesung
- Uebungsbezogene Themen:
  Spielbaumauswertung
- Zeit zum Programmieren...
  und fuer noch mehr Fragen

Projekt: Nicht so ganz klar…

Informatik II

Übungsstunde 9

simon.mayer@inf.ethz.ch
Distributed Systems Group, ETH Zürich