Knowledge Engineering

TECHNISCHE UNIVERSITÄT DARMSTADT

Solved Games (I) – Now and in the future

H.J. van den Herik (2002)

von Florian Scheuring

Agenda

TECHNISCHE UNIVERSITÄT DARMSTADT

I. Einleitung
II. Grundlagen
III. Eingesetzte Lösungsmethoden
IV. Vorstellung bereits gelöster Spiele
V. Betrachtung ausgewählter Spiele
VI. Zusammenfassung der Resultate
VII. Zukunftsausblick
VIII. Übersicht gelöster Spiele
IX. Auswirkungen auf den Menschen

Einleitung

TECHNISCHE UNIVERSITÄT DARMSTADT

Überblick der verschieden Ansätze zum Lösen von Spielen

- *brute-force method*
- *Knowledge-based method*

Ziel ist das Feststellen des „*game-theoretic value*“ eines Spiels

- Ist das Spiel ein first-player win / draw / lose?
- Ist ein beliebiger Spielzustand mittels middle- und endgame-DB bestimmbar?

Beobachten von „intelligenten Computern“

- Lösung eines Spiels/Spielzuges muss nicht für den Menschen verständlich sein

Grundlagen - Spielauswahl

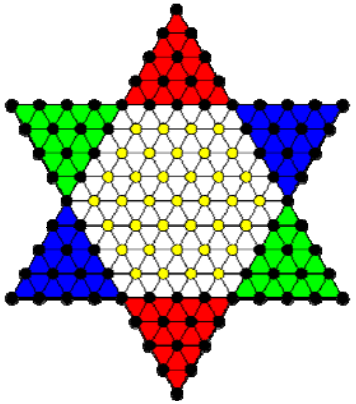
TECHNISCHE UNIVERSITÄT DARMSTADT

Auswahl der zu lösenden Spiele

- Zwei-Personen „*zero-sum*“ Spiele
- Perfekte Informationen

Beschränkungen der ausgewählten Spiele

- Kein Spiel auf Zeit (*Halma*)
- Kein Zufall (*Backgammon*)
- Keine Ungewissheit (*Bridge*)
- Keine unvollständige Informationen (*Stratego*)
- Keine Verhandlungen (*Monopoly*)
- Nicht Mathematisch lösbar (*Nim, gelöst von John Nash*)



Zu Beachten!

TECHNISCHE UNIVERSITÄT DARMSTADT

Nur Spiele, bei deren Lösung der Computer eine Rolle spielt und nicht trivial sind!

Grundlagen - Klassifizierung

TECHNISCHE UNIVERSITÄT DARMSTADT

ultra-weakly solved

- Bestimmung des *game-theoretic values* des initialen Spielzustandes, unabhängig von der jeweiligen perfekten Spielweise beider Spieler

weakly solved

- Angabe eines Algorithmus zur Bestimmung der optimalen Spielstrategie von dem initialen Spielzustand, unabhängig von der gegnerischen Spielweise

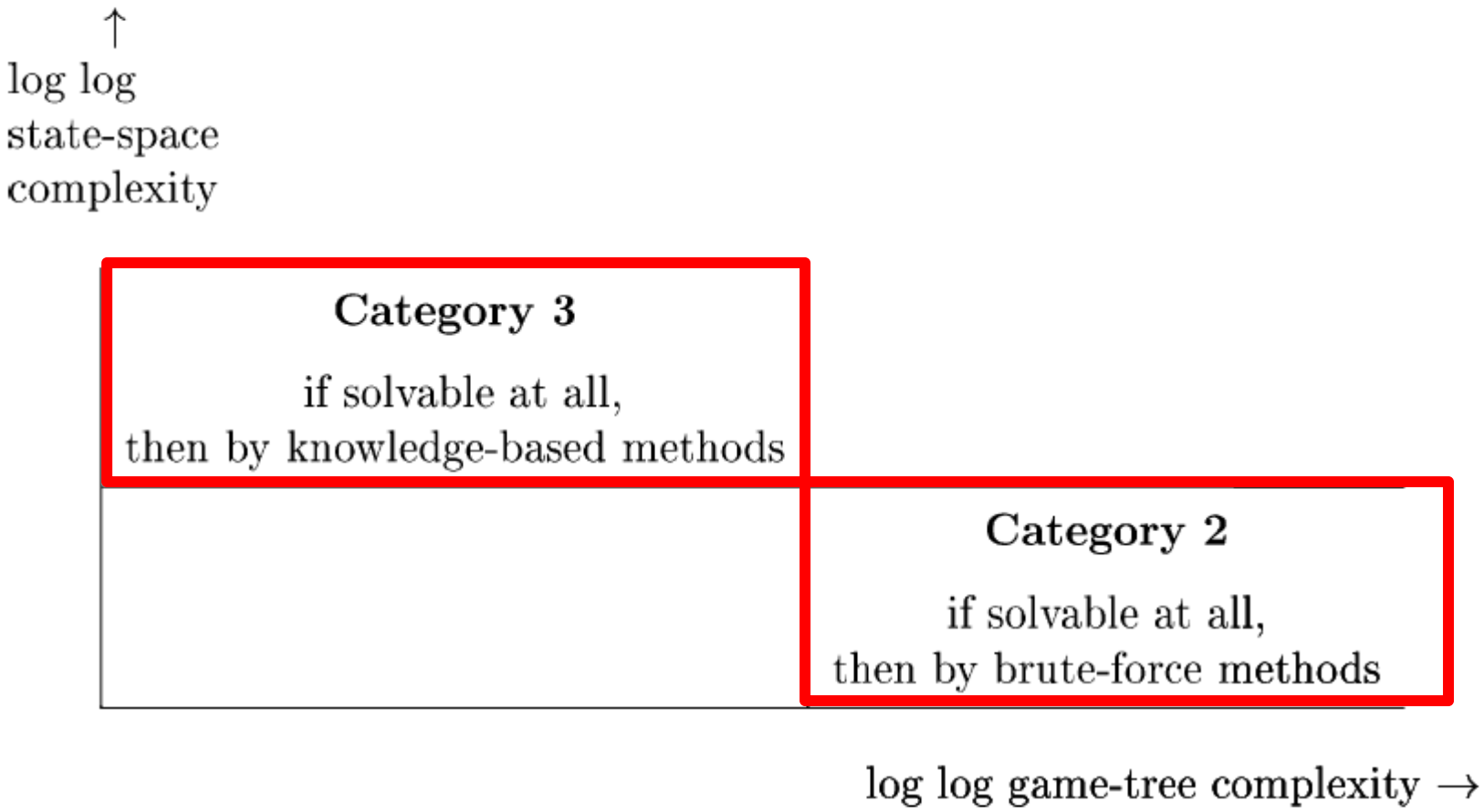
strongly solved

- Angabe eines Algorithmus zur Bestimmung der optimalen Spielstrategie von jedem beliebigen legalen Spielzustand, unabhängig von der vorangegangen Spielweise

Terminologie nach Allis [1]

V. Allis, *Searching for Solutions in Games and Artificial Intelligence.* PhD thesis, Department of Computer Science, University of Limburg, 1994

Grundlagen - Einordnung der Lösungskomplexität



Grundlagen - Komplexitätsfaktoren

TECHNISCHE UNIVERSITÄT DARMSTADT

state-space complexity

- Anzahl legaler Spielpositionen von der anfänglichen Spielposition
- *state-space* stellt eine natürliche Hürde für die Lösbarkeit, bzw. die Berechnungsdauer aller Lösungen, dar

game-tree complexity

- Anzahl der Blattknoten des *solution search tree* der anfängl. Spielsituation
- Wissen um die Spielregeln und –abläufe ist für die Suche wichtig
- Die echte Herausforderung liegt in der Suche nach einer intelligenten Suchmethode

Für eine berechenbare Lösung ist eine geringe *state-space compexity* vorteilhafter, als eine *geringe game-tree complexity*!

Komplexitätsübersicht

TECHNISCHE UNIVERSITÄT DARMSTADT

Table 6
State-space complexities and game-tree complexities of various games

Id.	Game	State-space compl.	Game-tree compl.	Reference
1	Awari	10^{12}	10^{32}	[3,7]
2	Checkers	10^{21}	10^{31}	[7,94]
3	Chess	10^{46}	10^{123}	[7,29]
4	Chinese Chess	10^{48}	10^{150}	[7,113]
5	Connect-Four	10^{14}	10^{21}	[2,7]
6	Dakon-6	10^{15}	10^{33}	[62]
7	Domineering (8×8)	10^{15}	10^{27}	[20]
9	Go (19×19)	10^{172}	10^{360}	[7]
10	Go-Moku (15×15)	10^{105}	10^{70}	[7]
11	Hex (11×11)	10^{57}	10^{98}	[90]
12	Kalah(6,4)	10^{13}	10^{18}	[62]
13	Nine Men's Morris	10^{10}	10^{50}	[7,44]
14	Othello	10^{28}	10^{58}	[7]
16	Qubic	10^{30}	10^{34}	[7]
17	Renju (15×15)	10^{105}	10^{70}	[7]
18	Shogi	10^{71}	10^{226}	[76]

Eingesetzte Lösungsmethoden

TECHNISCHE UNIVERSITÄT DARMSTADT

Brute-force methods

- *Retrograde analysis*
- *Enhanced transposition-table methods*

Knowledge-based methods (Heuristiken)

- *Threat-space search*
- *Proof-number search*
- *Pattern search*

Brute-force methods

TECHNISCHE UNIVERSITÄT DARMSTADT

Retrograde analysis

- Für *endgame* / *middlegame* wird die Anzahl der Spielzüge zur besten Position (*win*, *draw*) gespeichert
- DB enthält manchmal nur *win*, *draw* oder *lose* (Damen-DB)
- Datenbank baut sich von der finalen Position (*endgame*) rückwärts auf um so das perfekte Spiel zu garantieren
- Wahl des Spielzuges durch kürzeste Spielzüge (starker Spieler) oder längste Spielzüge (schwacher Spieler) im Suchbaum
- Führte Begriffe wie *max-to-mate, max-to-conversion* ein

Enhanced transposition-table methods

- Begrenzung der abgesuchten Teilbäume durch ersetzen bereits bekannte Spielsituationen
- Ersetzung von (Teil)Suchbäumen mindestens gleicher Höhe

Knowledge-based methods I

TECHNISCHE UNIVERSITÄT DARMSTADT

Threat-space search

- Analyse der Spielsituation durch entsprechenden *game-tree*
- Suche nach *threats* oder einer *threat-sequence*, auf die der Gegner nur eingeschränkt reagieren kann oder ein Sieg erzwungen werden kann

Proof-number search

- Erfunden von Victor Allis
- *Best first search* mit binärem Ziel (Spieler am Zug gewinnt), Baum wird zu Und-Oder-Baum transformiert
- Kostenfunktion, die nach der Anzahl der zu expandierenden nachfolgenden Knoten entscheidet, welcher Knoten als nächstes betrachtet wird um das Ziel zu beweisen.
- Ziel ist die minimale Anzahl der zu expandierenden nachfolgenden Knoten

Knowledge-based methods II

TECHNISCHE UNIVERSITÄT DARMSTADT

Pattern search

- Genutzt bei Spielen, bei denen sich die Position der Spielsteine nicht mehr verändern, Beispiel Hex
- *threat-pattern* bekannt, meist wird eine Datenbank für die Speicherung genutzt
- Sucht nach *threat-pattern* (Anordnung leerer Felder) die schlussendlich zum Sieg führen können (bereits bekannte Pattern)

Generell werden die vorgestellten Suchmethoden miteinander kombiniert um den Sucherfolg zu vergrößern oder um Rechenzeit einzusparen.

Konvergierende Spiele

TECHNISCHE UNIVERSITÄT DARMSTADT

Unterteilung der Spiele nach Kriterium der Spielstein Zu- oder Abnahme

Spielsteine nehmen mit Spielverlauf ab

Erlaubt die Konstruktion von *Middle*- und *Endgame*-Datenbank

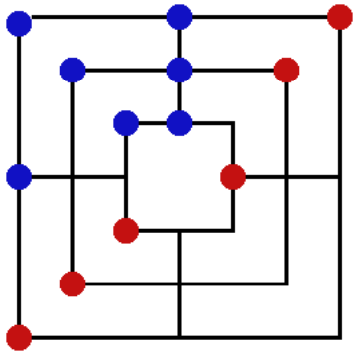
- Hilfsmittel für die Lösung eines Spiels durch *retrograde analysis*

Betrachtete Spiele:

- Mühle
- Schach
- Dame
- Mancala Games

Mühle

TECHNISCHE UNIVERSITÄT DARMSTADT



3000 Jahre alt

Erst divergierend – dann konvergierend

1995 von Grasser gelöst mit folgendem Ansatz:

- Unterteilung des Spiels in 3 Phasen: opening, middle- und endgame
- Erstellte Datenbank mit Datenbankorientierter Suche
  - Alle 28 w-b middle- und endgame DBs mit max 9 w/b, min 3 w/b Steine; w>=b
  - 7,673,759,269 mögliche Positionsanordnungen, inkl. aller unerreichbarer und symmetrischer Positionen
- Eine 18-Ply-Suche von der Startposition aller 9-9, 9-8 und 8-8 Datenbanken ergab den *game theoretic value* draw

Schach

TECHNISCHE UNIVERSITÄT DARMSTADT

Selbstständiges Thema => 2 separate Vorträge

Schwerpunkt der Forschung ist die Erstellung einer Endspieldatenbank

- Bekannte Beispiele sind die KBBKN-Datenbanken (5 Figuren)
- 2001 nahm man 6-Figuren-Endgame-DBs noch nicht ernst, aktuell sind die wichtigsten mit 6 und vereinzelt 7 Figuren bekannt

Forschung hat Einfluss auf die reale Welt!

- Nachfeststellen das einzelne Spielzüge nicht in n <= 50 Spielzügen beendet werden kann, würde eine n+1 Regel eingeführt.
- Durch Benachteiligung des verteidigend Spielers wurde diese Regel jedoch wieder abgeschafft
- Stand 2001 war ein Spiel mit 258 Zügen bekannt, dessen Spielzüge selbst Großmeister nur eingeschränkt nachvollziehen konnten.

Divergierende Spiele

TECHNISCHE UNIVERSITÄT DARMSTADT

Die Anzahl der Spielsteine nimmt im Spielverlauf zu

Betrachtete Spiele

- Vier Gewinnt
- Qubic
- Go-Monku
- k-in-a-row

} Spiele mit Verbindungsreihen

- Hex
- Othello
- Shogi
- Go

K-in-a-row Spiele

TECHNISCHE UNIVERSITÄT DARMSTADT

Auch bekannt als mnk-Spiele mit Spielfeldbreite (m) und –länge (n) sowie der Anzahl zum Sieg notwendigen, k in einer Reihe liegenden Steine

Beispiel für m=7, n=6, k=4 (Vier gewinnt)

- brute force DFS mit α-β-Suche
- *Transposition-tables* werden genutzt um den *game-state* für jede Position bestimmen zu können
- *killer-move heuristics* werden zur Verbesserung der α-β-Suche genutzt
- Zur *weak*-Lösung brauchte ein PC mit ca. 20 MHz und 32 MB RAM nur 300 Stunden!

Table 3
Game values of *mnk*-games

mnk-games ($k = 1, 2$)	W
333-game (Tic-Tac-Toe)	D
*mn*3-games ($m \geqslant 4, n \geqslant 3$)	W
*m*44-games ($m \leqslant 8$)	D
*mn*4-games ($m \leqslant 5, n \leqslant 5$)	D
*mn*4-games ($m \geqslant 6, n \geqslant 5$)	W
*mn*5-games ($m \leqslant 6, n \leqslant 6$)	D
15,15,5-game (Go-Moku)	W
mnk-games ($k \geqslant 8$)	D

Divergierende Spiele ohne vollständige Lösung

TECHNISCHE UNIVERSITÄT DARMSTADT

Nicht alle Spiele sind zum jetzigen Zeitpunkt lösbar!

- Shogi
- Go

Einschränkung der Lösbarkeit des Standartspiels

- Hex
  - 4x4 und 6x6 sind Siege für den *first-player*
  - für 7x7 gibt es Siegstrategie, veröffentlicht von Yang 1990
  - QUEENBEE löste das Spiel 1999 *strong* mit *pattern search*
  - Spielstärkstes Computerprogramm ist Hexy mit *virtual connection search*
- Othello
  - 6x6 ist *weak* gelöst durch Feinstein; Niederlage für den *first-player*
  - 8x8 konnte bisher nicht gelöst werden
  - Spielstärkstes Computerprogramm ist Logistello

Zusammenfassung der Resultate

TECHNISCHE UNIVERSITÄT DARMSTADT

↑
log log
state-space
complexity

Category 3 Go-Moku und *k-in-a-row* Spiele	**Category 4** Go, Shogi, Schach
Category 1 Vier gewinnt und Qubic	**Category 2** Mühle, Mancala Spiele

log log game-tree complexity →

First-player initiative

TECHNISCHE UNIVERSITÄT DARMSTADT

Auswirkung des Rechts der Spieleröffnung auf *game-theoretic value*

- *first-player* muss die ausreichend Möglichkeiten zum Sieg haben
- *first-player* macht einen Zug, der Gegenspieler reagiert darauf
- *first-player* sucht nach einer Bedrohung, um den Gegenspieler zur Reaktion zu zwingen
- Ziel ist mind. eine Doppelbedrohung (threat-sequence), welche zum Sieg führt

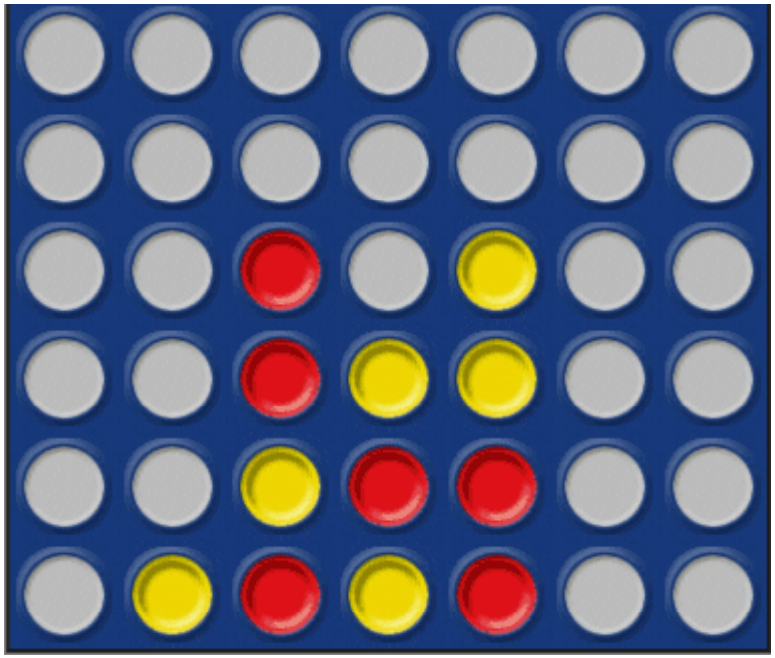
Zusammenhang von *knowl.-based search*, Spielinitiative und Spiellösung

- Spiele mit klarem first-player-Vorteil sind mittels threat-space-search leichter zu lösen (Go-Moku)
- Eigener Forschungsschwerpunkt
  - J.W.H.M. Uiterwijk, H.J. van den Herik, The advantage of the initiative, Information Sciences 122 (1) (2000) 43–58
  - D. Singmaster, Almost all games are first person games, Eureka 41 (1981) 33–37.

Beispiele für *first player win*

TECHNISCHE UNIVERSITÄT DARMSTADT

Vier gewinnt!



L L D W D L L

Zukunftsausblick des Papers

TECHNISCHE UNIVERSITÄT DARMSTADT

- Stand `00: Lösung von 10^11
- Voraussage `02: Lösung von 10^13
- Lösung von Dame bis 2010
- Computer sollten Scrabble nahezu perfekt spielen können

Erwartungen der Leistungsfähigkeit bei weitem übertroffen, Spiele mit *state-space complexity* von 10^21 bereits 2004 gelöst.

Voraussage von 1990

Table 1
Predicted program strengths for the Computer Olympiad games in the year 2000

Solved or cracked	Over champion	World champion	Grand master	Amateur
Connect-Four	Checkers (8 × 8)	Chess	Go (9 × 9)	Go (19 × 19)
Qubic	Renju	Draughts (10 × 10)	Chinese chess	
Nine Men's Morris	Othello		Bridge	
Go-Moku	Scrabble			
Awari	Backgammon			

Voraussage von 2001

Table 7
Predicted program strengths for the Computer Olympiad games in the year 2010

Solved or cracked	Over champion	World champion	Grand master	Amateur
Awari	Chess	Go (9 × 9)	Bridge	Go (19 × 19)
Othello	Draughts (10 × 10)	Chinese Chess	Shogi	
Checkers (8 × 8)	Scrabble	Hex		
	Backgammon	Amazons		
	Lines of Action			

Übersicht aller gelösten Spiele

TECHNISCHE UNIVERSITÄT DARMSTADT

Name	weak / strong	Jahr	first player win / draw / lose
Mühle	weak	1993 v. Gasser	Perf. verliert nie – draw
Awari	strong	2002 v. Bal	J. Spieler kann draw erzwingen
Kalah	weak	2000 v. Irving	Tendenz win, 6x6 ungelöst
Dame	?	?	=> Nächster Vortrag!
Vier gewinnt	strong	1988 v. Allis	Siehe Beispiel
Qubic	weak	1980 v. Patashnik	win
Go-Moku	strong	2000 v. Allis	win mit Gewinnstrategie
Renji	weak	2000 v. Wágner	win
Hex	weak	1947 v. Nash	win
Tic-Tac-Toe	strong	-	draw*
Othello	weak	1997 v. Feinstein	4x4,6x6 lose; 8x8 wahrsch. draw*

Auswirkungen auf den Menschen

TECHNISCHE UNIVERSITÄT DARMSTADT

Beeinflussung des menschlichen Spielers durch die Erkenntnisse aus den gelösten Spielen sehr unterschiedlich

- Gelöste Spiele teilweise nicht für den Menschen verständlich
- *Threats* werden vom Menschen meist direkt erkannt
- Regeln können generell nicht für jedes Spiel erstellt werden
- Es gibt durch die Anzahl der verschiedenen Spiele keine ad-hoc-Lösung
- Ergebnisse führt bestenfalls zur Korrektur der von Experten verfassten Spielstrategien
- Spieler lernen jedoch neue Spielstrategien von den Computerprogrammen wie TD-GAMMON und dem Scrabbleprogramm MAVEN

Noch Fragen?

TECHNISCHE UNIVERSITÄT DARMSTADT

Vielen Dank für Eure Aufmerksamkeit!

Referenzen

[1] V. Allis, *Searching for Solutions in Games and Artificial Intelligence.* PhD thesis, Department of Computer Science, University of Limburg, 1994

[2] H. Jaap van den Herik, Jos W.H.M. Uiterwijk, Jack van Rijswijck, *Games solved: Now and in the future*, Artificial Intelligence 134 (2002) 277–311

Bildreferenzen

TECHNISCHE UNIVERSITÄT DARMSTADT

Übersichtsfolie: http://www.holzspielzeug-paradies.de/shop/erwachsene/41h1LrHVvuL__SL500_AA240_.jpg
Halma: http://upload.wikimedia.org/wikipedia/commons/thumb/e/e8/Sternhalma.svg/517px-Sternhalma.svg.png
Stratego: http://www.coolest-toys.com/wp-content/uploads/2008/11/800px-stratego_board.jpg
Mühle: http://upload.wikimedia.org/wikipedia/commons/f/f0/Nine_Mens_Morris_maximum.PNG
Mancala: http://natalie17.files.wordpress.com/2009/03/mancala.jpg
Vier-gewinnt: http://l-ectron-x.javalinkbase.de/pictures/connect4_Example.png
Fragezeigen: http://www.farmersfriend.de/Fuer_Sie/images/Haeufig_gestellte_Fragen.jpg

Sonstige Grafiken wurden aus dem Paper H. Jaap van den Herik, Jos W.H.M. Uiterwijk, Jack van Rijswijck, *Games solved: Now and in the future*, Artificial Intelligence 134 (2002) 277–311 entnommen