

# Künstliche Intelligenz: Bots

- Künstliche Intelligenz
- Bots: Begriffsklärung
- Klassische Techniken am Beispiel von Schach
- Unwissende KI: Moderne Echtzeitspiele



# Künstliche Intelligenz

- schwammiger Begriff
- starke vs schwache KI
- Agentensysteme, Expertensysteme:  
Lösen komplexe Aufgaben
- vollständige Information vs  
Unwissenheit
- mathematische Betrachtung bereits  
durch von Neumann, Shannon etc.

# Künstliche Intelligenz

- Was braucht eine KI mindestens:
- Repräsentation der aktuellen Situation
- schnell mögliche Züge finden
- Züge bewerten, sprich benennen und auswählen nach Kriterien
- muss ausführbar sein und überwacht werden können (UI)

# Bots: Begriffsklärung

„Unter einem Bot (vom Begriff robot abgeleitet) versteht man ein Computerprogramm, das weitgehend autonom ständig gleichen, sich wiederholenden Aufgaben nachgeht. Es handelt sich dabei meist um ein eher simples, aber effektives Programm (Skript), das auf einem Computer ohne sonderliche Interaktion mit einem menschlichen Benutzer läuft.“

# Bots: Begriffsklärung

- Bots in PC-Spielen: intelligent wirkendes Verhalten, imitiert menschliches Verhalten 'gut'
- löst komplexe Aufgaben durch iterative Lösung von Teilaufgaben
- es gibt viele intuitive Unterscheidungen solcher Bots nach Einsatzgebiet, Effektivität, Menschlichkeit, Stil ...

# Bots: Begriffsklärung

- Beispiel Aufbau-Strategie-Spiel
- Bot als KI-Gegner will Spiel gewinnen
- arbeitet regelbasiert, baut danach Rohstoffe ab, Basis auf, Einheiten, erforscht Gegend und bewegt Truppen in bestimmte Richtung
- lokale geskriptete Kämpfe
- selten Bayesian Network (dynamic belief nw) zum Erlernen des Gegners

# Bots: Begriffsklärung

- allerdings komplett regelbasierte Strategie (triggered, event-based)
- achtet nicht auf Strategie der Gegner, Ungleichgewicht, Stellung ...
- mitunter sehr komplexe Regeln, die auch Reaktionen ermöglichen, allerdings immer noch event-gesteuert
- Zwischenziele ergeben sich intuitiv

# Bots: Begriffsklärung

- Kandidaten dieses Beispiels:
- die Reihen: Warcraft, Command&Conquer, Dune, ...
- Merkmale: statische Wege, nachvollziehbare Regeln, Kampf nur lokal, gelangweilte und selbstmordgefährdete Einheiten etc, häufig virtual waypoints erkennbar (strategische Sammelpunkte)



# Bots: Begriffsklärung

- Beweggründe Bots zu nutzen
- Entwickler: Bots sollen den Spielern mehr Spielspaß gönnen, Herausforderung darstellen und menschliche Spieler simulieren
- heutzutage: nahezu jedes Spiel für mehr als 5€ enthält KI-Elemente, von einfachen creeps/mobs über Bosse/Raidencounter zu KI-Spielern

# Bots: Begriffsklärung

- bei Entwicklung der KI muss also ihre spätere 'Nutzung' erraten werden und die Stärke der Elemente vollständig bekannt sein, möglichst noch mit mehreren Stufen
- aber: kurze Entwicklungszeiten von Spielen, kurze Lebensdauer der meisten, komplexe Materie im Hintergrund und das Problem von 'intelligent, aber menschlich'

# Bots: Begriffsklärung

- Motivation ein bestehendes Spiel mit Bots zu erweitern:
- Automatisierung sich häufig wiederholender Aufgaben, Simulation von Mitspielern, Ausnutzen von Lücken des Programms, Erweiterung des menschenmöglichen, Erlangung von zusätzlichen Informationen

# Spiele vollständiger Information

- Brett-repräsentation im Schach
- erste Idee: 64 B Array (1B je Feld)
- Mainframes: Bitboards
- später 32B Felder (1B je Figur)
- viele Abwandlung, heutzutage ohne Bedeutung
- benötigt meist noch einige B extra für Rochade, en passant etc.

# Spiele vollständiger Information

- wie entscheidet man nun, was man ziehen will?
- Stellungsanalyse und go?
- alle Züge iterieren?
- einige und dann Minimax?

# Spiele vollstandiger Information

- Stellungsanalyse: siehe spater stark bewertende Ansatze, selbst eine perfekte Analyse gibt keine Auskunft daruber wie fortzusetzen ist
- alle Zuge: es gibt mehr mogliche Positionen als Atome im Universum
- Minimax: Komplexitat  $O(b^n)$  mit  $b$ ..branching factor,  $n$ ..Halbzuge
- $b \sim 35, n > 8 \Rightarrow$  mehr als 2 Billionen

# **Spiele vollständiger Information**

- es muss also geschickt gesucht werden
- 3 unterschiedliche Ansätze
- Forward Pruning
- Incremental Generation
- Complete Generation

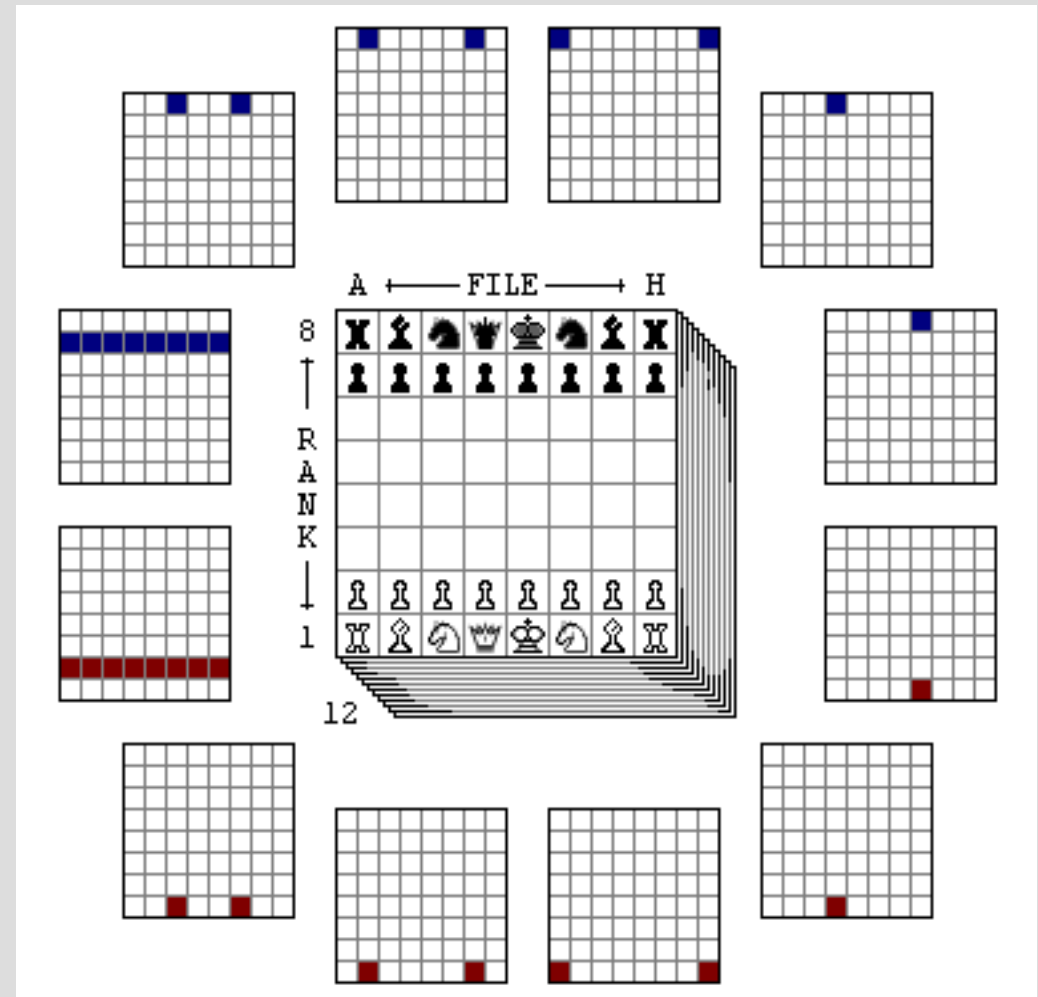
# Spiele vollständiger Information

- Forward Pruning (Shannon '49)
- finde nur Kandidatenzüge und rechne diese tief durch
- Problem: je 5 Kandidaten in 40 Zug Partie, 95% Wahrscheinlichkeit für besten Zug in diesen 5 => 87% für Ungenauigkeit unbekanntes Ausmaßes
- M. Botvinnik bis Anfang 80er, danach ausgestorben



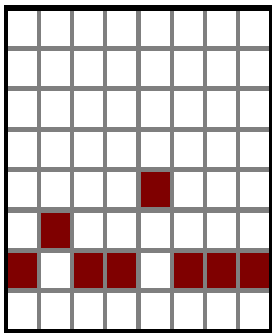
# Spiele vollständiger Information

- complete generation
- generiere alle Züge mittels Bitboards

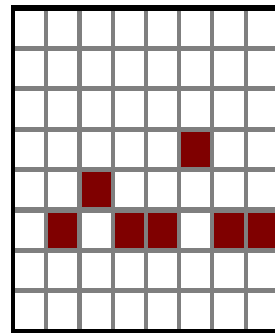


# Spiele vollständiger Information

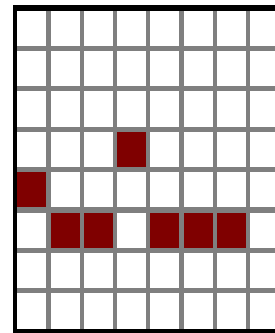
- Beispiel Bitboards
- PAWN\_ATTACKS=  
 $((WP \ll 7) \& \sim RANK\_A) \&$   
 $((WP \ll 9) \& \sim RANK\_H)$



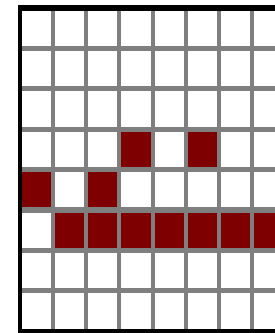
WP



$((WP \ll 7)$   
 $\& \sim RANK\_A$



$((WP \ll 9)$   
 $\& \sim RANK\_H$



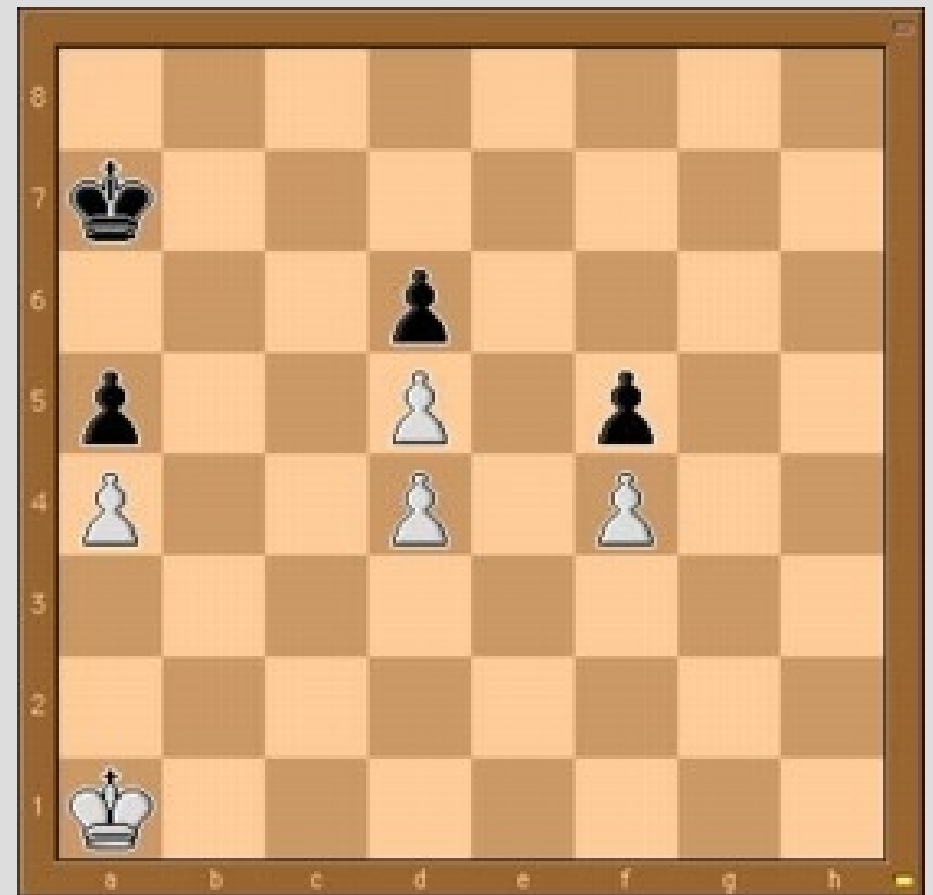
PAWN\_ATTACKS

# Spiele vollständiger Information

- damit werden also schnell alle Züge gefunden und noch zusätzliche Informationen, wie Schach, Matt, Patt, Bedrohung, Kontrolle, Beweglichkeit
- Positionen sichern in Hashtables (mittels Zobrist-Hashing für Schach)
- insbesondere Transposition tables um Aufwand zu sparen für Positionen die auf unterschiedlichen Wegen erreicht werden können

# Spiele vollständiger Information

- $n$  Züge  $\Rightarrow \max (n!)^2$   
Zugumstellung
- SigmaChess:
- 10B pro Stellung
- 80k-360M TT
- nutzt 2TT
- löst Laskerpos in wenigen Sekunden, ohne TT ..Stunden



# Systeme vollständiger Information

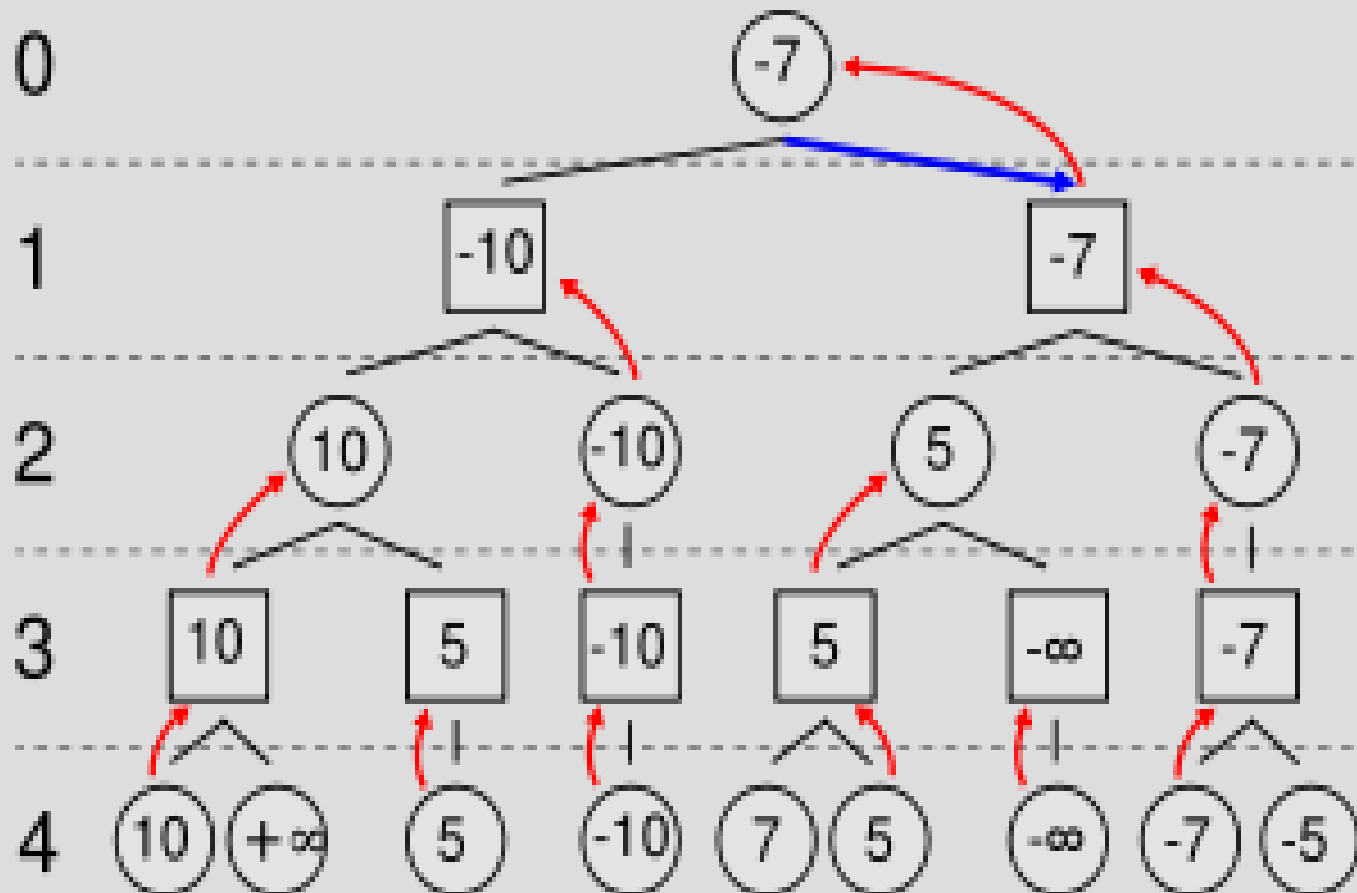
- Anwendung der Transposition Table:
- Zugumstellungen, eben besprochen
- Umsortieren der Liste -> Bessere Auslastung der Liste
- Vorsortieren der Suchliste, ermöglicht bsp. Iterative Deepening deutlich tiefere Suche in interessanten Bereichen!

# Systeme vollständiger Information

- Gedanken zum Vorsortieren
- gute Vorsortierung ist Grundlage aller Optimierung im Suchbaum
- Komplexität Exponent  $\frac{1}{2}$  im Vergleich zum unsortierten Ansatz (allerdings kann die Sortierung nicht perfekt sein)
- erste Vorsortierungsidee: Tausch, Bauernumwandlung, Schachs, 'Killerzüge'/Patzner, Rest

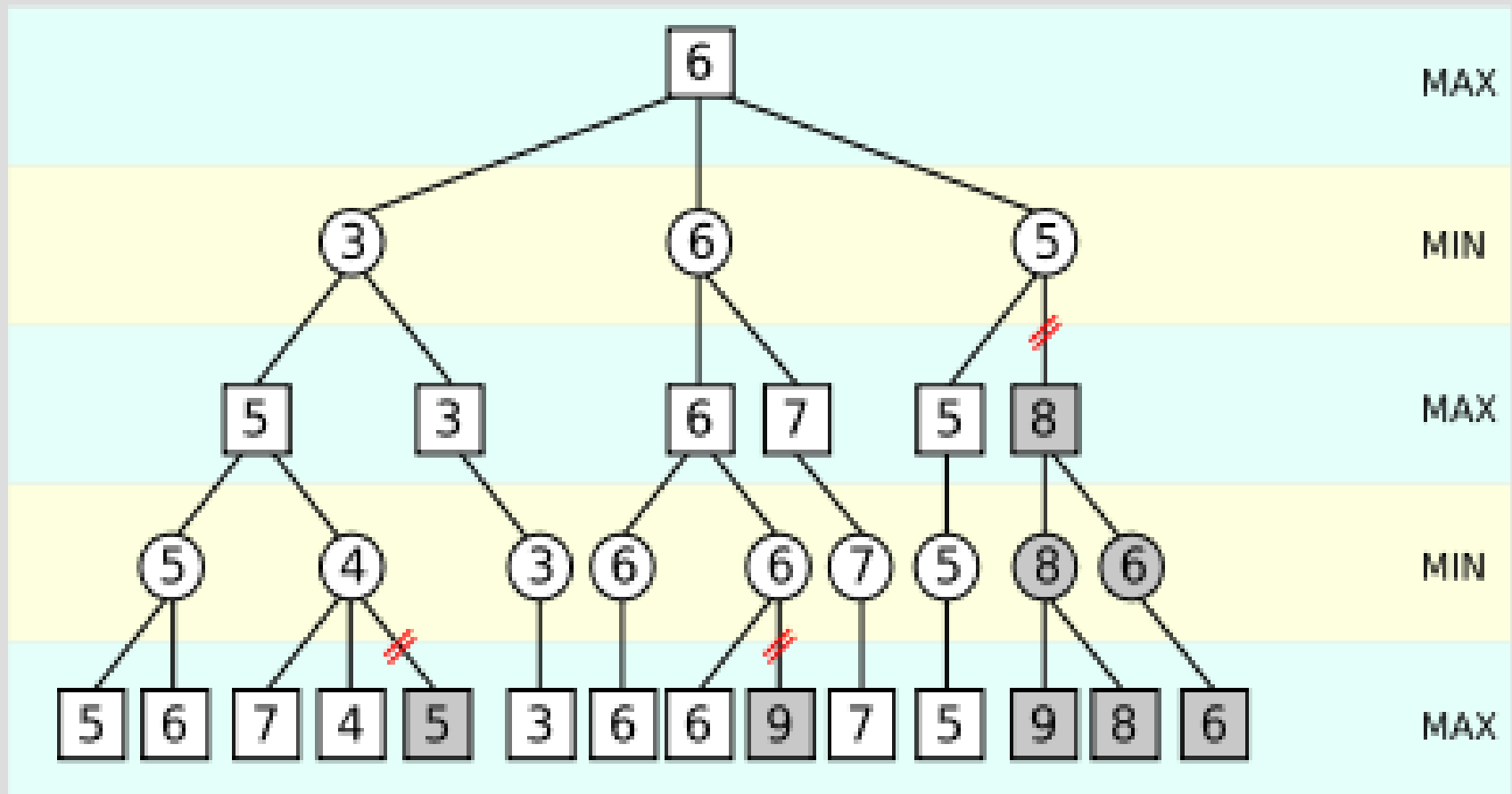
# Systeme vollständiger Information

- Suchtechniken: Minimax



# Spiele vollständiger Information

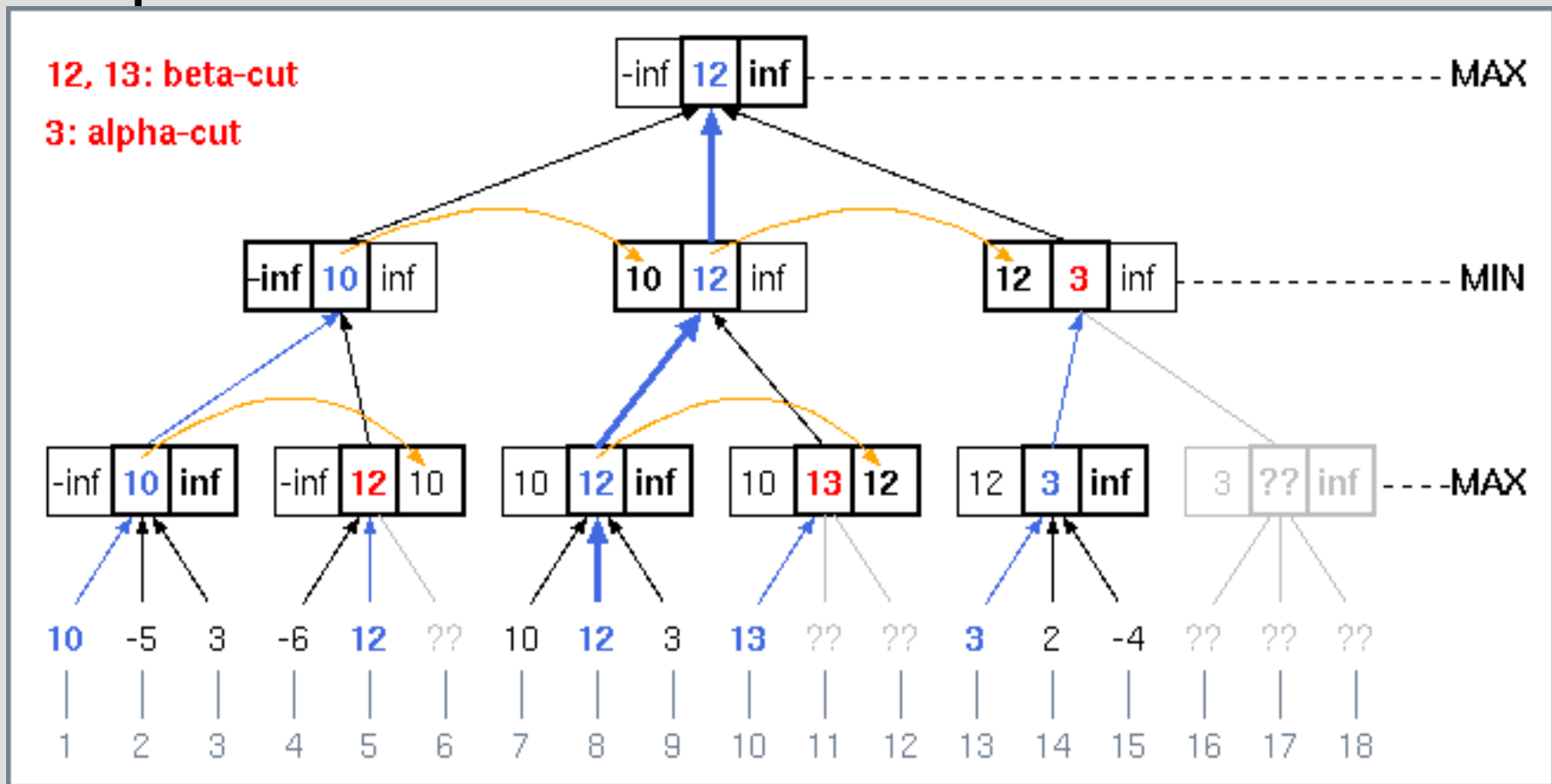
- Suchtechniken: AlphaBeta über Minimax





# Spiele vollständiger Information

- Alpha-Beta-Cutoffs



# Spiele vollständiger Information

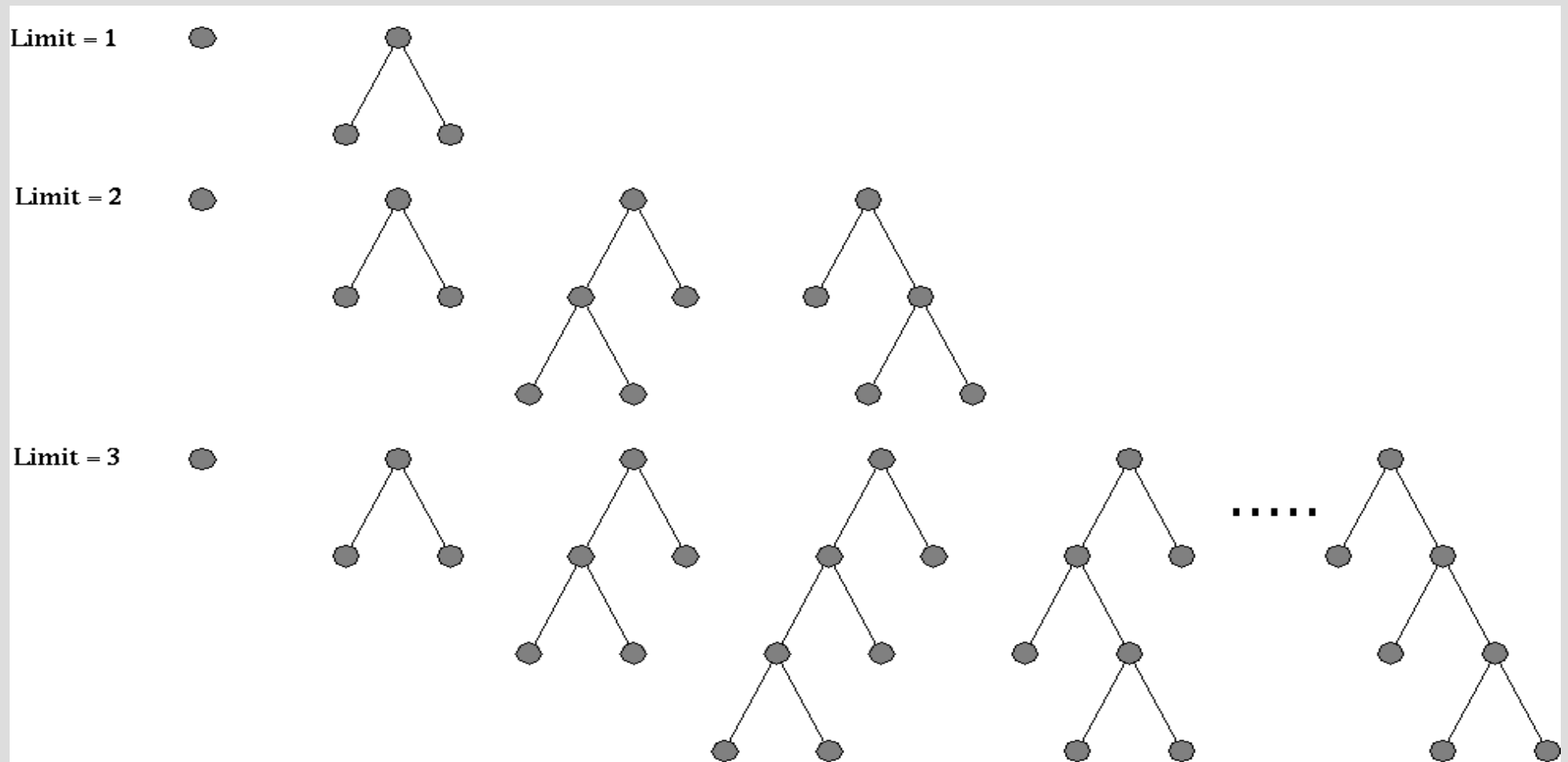
- alphabeta optimieren, das heißt frühe und häufige Cutoffs zu provozieren
- mittels Transposition list vorsortieren
- intuitive Zwischensortierung
- Killer-Heuristic
- History-table
- „mathematische Drosophila Melanogaster“

# Spiele vollständiger information

- Iterative Deepening („iter. Tiefensuche“)
- Suche in Graphen: Speicher-effizienz  $O(b*d)$ , Suche ist vollständig
- informell: nutze Tiefe  $d-1$  zum sortieren, baue Tiefe  $d$  danach auf
- math.: Tiefe iterierende Tiefensuche
- „Der praktische Unterschied zwischen Theorie und Praxis ist größer als der theoretische.“

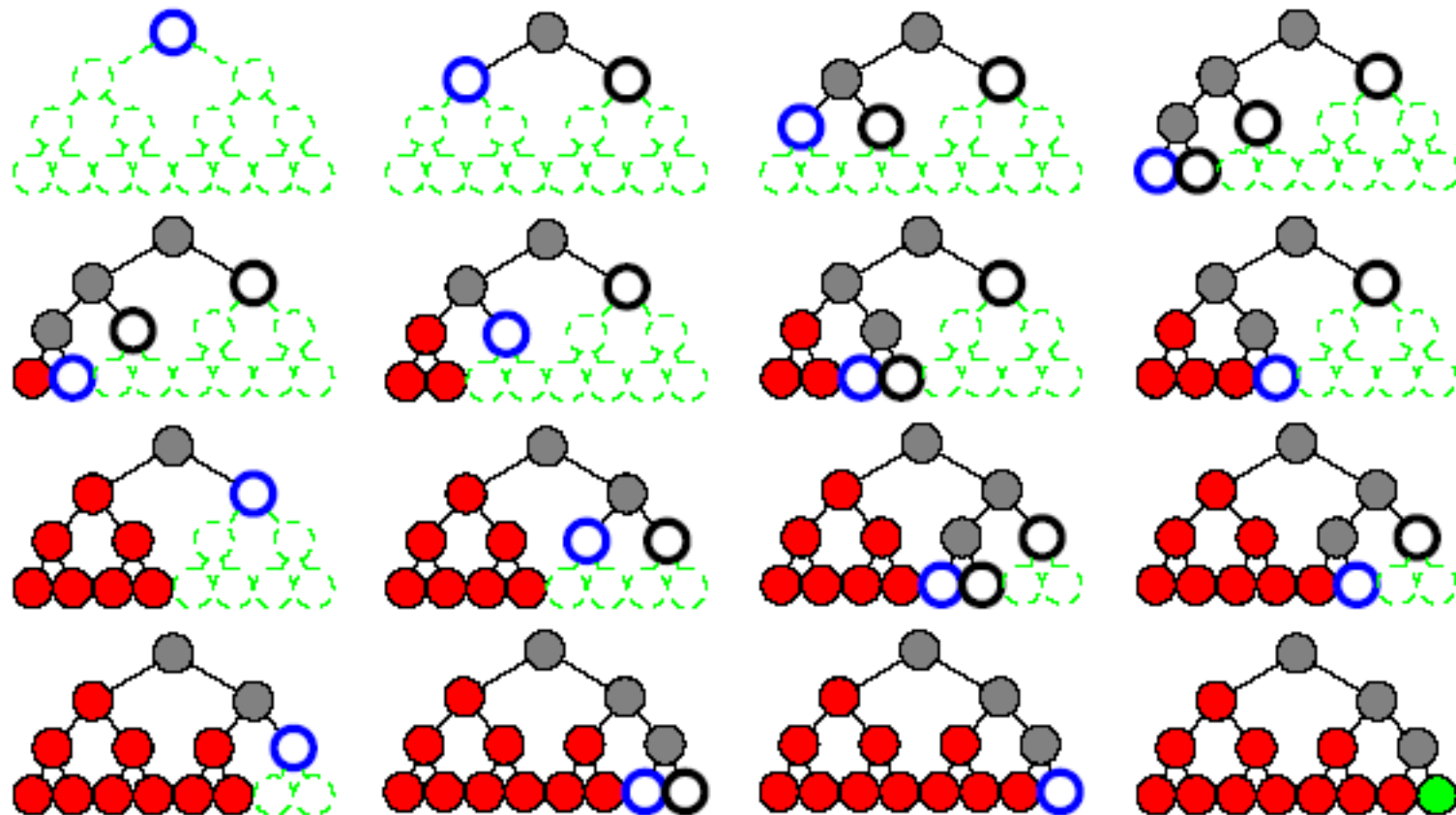
# Spiele vollständiger Information

- iterative Tiefensuche:



Depth-limited search: Depth-first search with depth limit

Iterative deepening search: Depth-limit search with ever increasing limits



Complete	Yes
Time	$(d + 1)b^0 + db^1 + (d - 1)b^2 + \dots + b^d \in O(b^{d+1})$
Space	$O(bd)$
Optimal	Yes (if step cost = 1)

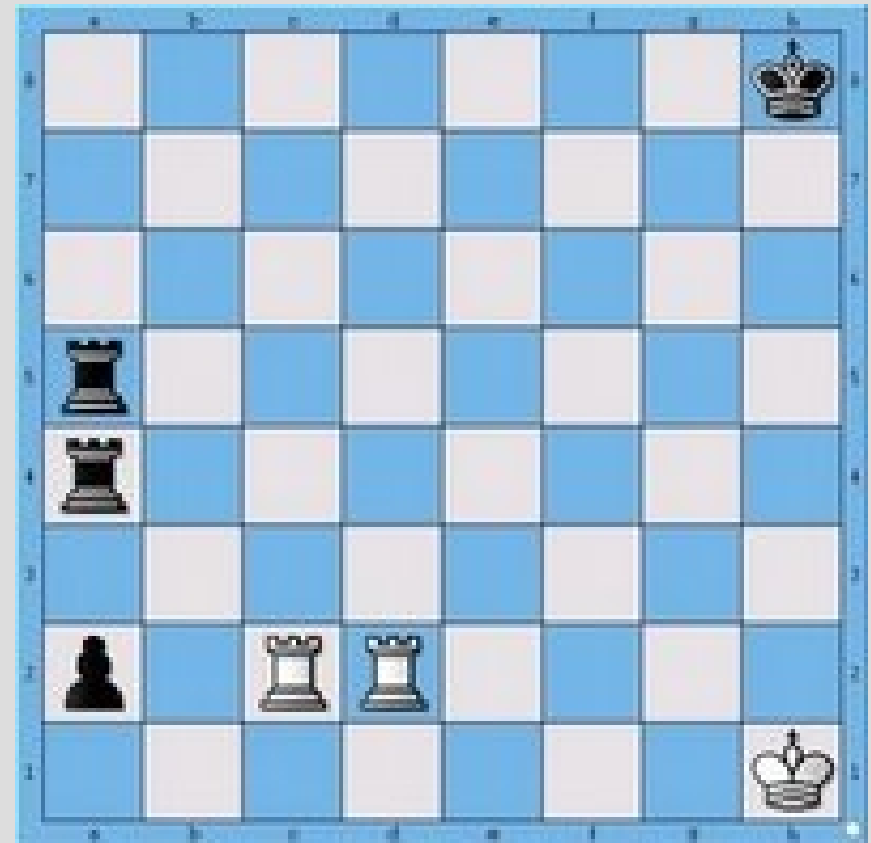
(Depth-First) Iterative-Deepening Search often used in practice for search spaces of large, infinite, or unknown depth.

# Spiele vollständiger Information

- vollständige Breitensuche zu groß
- Ziele: Bäume entarten, viele cuts
- begrenzte Tiefe bringt viele Probleme
- finalstate im Baum (Fehlberechnung)
- im letzten Zug heiße Situation
- horizon effect

# Spiele vollständiger Information

- einfaches Beispiel zum Verdeutlichen des Horizon effects
- | markiert Horizont
- 1.Ta2| Ta2 2.Ta2 Ta2
- 1.Ta2 Ta2| 2.Ta2 Ta2
- 1.Ta2 Ta2 2.Ta2| Ta2
- 1.Ta2 Ta2 2.Ta2 Ta2|



# Spiele vollständiger Information

- realeres Beispiel  
horizon effect
- 1. ... - Lh2
- 2. Kh2 – Sbc5
- 3 Halbzüge tiefe  
Suche patzt Lh2
- 4 Halbzüge tiefe  
Suche sieht das volle  
Ausmaß der Gabel





# Spiele vollständiger Information

- Optimierung der Suche durch Baumentartung
- Nullzug-einführung führt zu simulierter Entartung
- Beispiel: Situation ist so gut, dass bei Nullzug schon cutoff in der Antwort entsteht => man braucht den Ast nicht entwickeln
- Ersparnis im Schach: Faktor  $1/4-1/3$  !

# Spiele vollständiger Information

- Quiescence search: Suche nach kälteren, ruhigeren Situationen
- kaum eine Situation ist ganz ruhig, finde also Kriterien für 2 Wege: tiefe Suche mit einfacherer Bewertung vs teure strategische Bewertung einer ruhigen Situation
- intuitiv: untersuche heiße Züge, bei schlechter Bewertung => ruhige Pos.

# Spiele vollständiger Information

- weitere Entartungen:
- Singular Extensions (Deep Blue)
- finde heiße Züge und expandiere diese sehr tief, speziell Schlag und Schach-situationen über mehre 'stille' Zwischenzüge
- benötigt mehrere sehr gut durchdachte Bewertungen

# Spiele vollständiger Information

- weitere Ideen:
- aspirated search
- MTD(f)
- Negascout
- Zukunftsideen:
- strategic history table, context-sensitive opening und immernoch viele weitere

# Spiele vollständiger Information

- Bewertungsfunktionen
- Material (Un-)Gleichgewicht
- Beweglichkeit der Figuren
- Einflussbereich, Raum
- Entwicklung, Grundaufbau
- Bauernstruktur, doppelte., Freibauern
- Königsstellung
- ...

# Spiele vollstandiger Information

- perfekte Bewertung gibt's nicht, Suche bleibt, nicht mal teure Bewertung lohnt sich, da man einfach noch ein paar alphas und betas weiter suchen konnen
- mitunter: generisches Algorithmen

The screenshot displays a chess engine interface with the following elements:

- Engine Name:** Fritz 9
- Game ID:** 76...♔d1 (1/12)
- Control Buttons:** Stop (red), +, -
- Game Status:** ● +- (3.03)
- Search Progress:** depth=17/17
- Search Speed:** 353 kN/s
- Game History:** 76...♔d1 77.h5 ♔xh5 78.♔xh5 ♔c8 79.♔e2 ♔b8 80.a6 ♔a8 81.♔e6 ♔a7 82.♔c4 ♔a8 83.♔d7 ♔a7
- Statistics:** +- (3.03) Depth: 16/31 00:00:12 4314kN, tb=591

# Spiele mit Unsicherheiten

- benötigt mathematische Handhabung der unbekanntenen Informationen, Arbeit darauf erfolgt mit altbekanntem Suchen und Bewerten, ABER
- Echtzeitspiele wie Strategiespieler, Rollenspiele, Egoshooter
- mögliche Züge in immenser Zahl, Traversierung in Echtzeit nicht möglich!