

# Paradise

Geführte Suche im Schach

Von

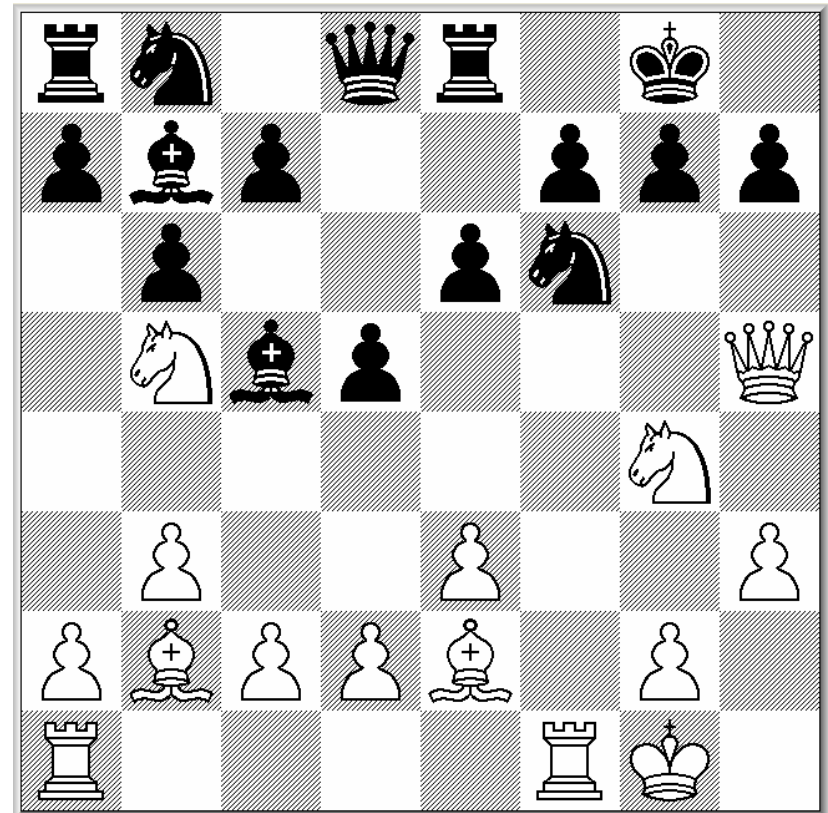
Michael Achenbach

# Motivation

- ◆ Problem: Spielbaum beim Schach ist zu mächtig
- ◆ Verzweigungs-Faktor ist bei durchschnittlich 38 Zugmöglichkeiten sehr hoch
- ◆ Die meisten Spiele haben mehr als 40 Züge
- ◆ Lösung: Verwendung von Expertenwissen, um den Suchraum zu begrenzen

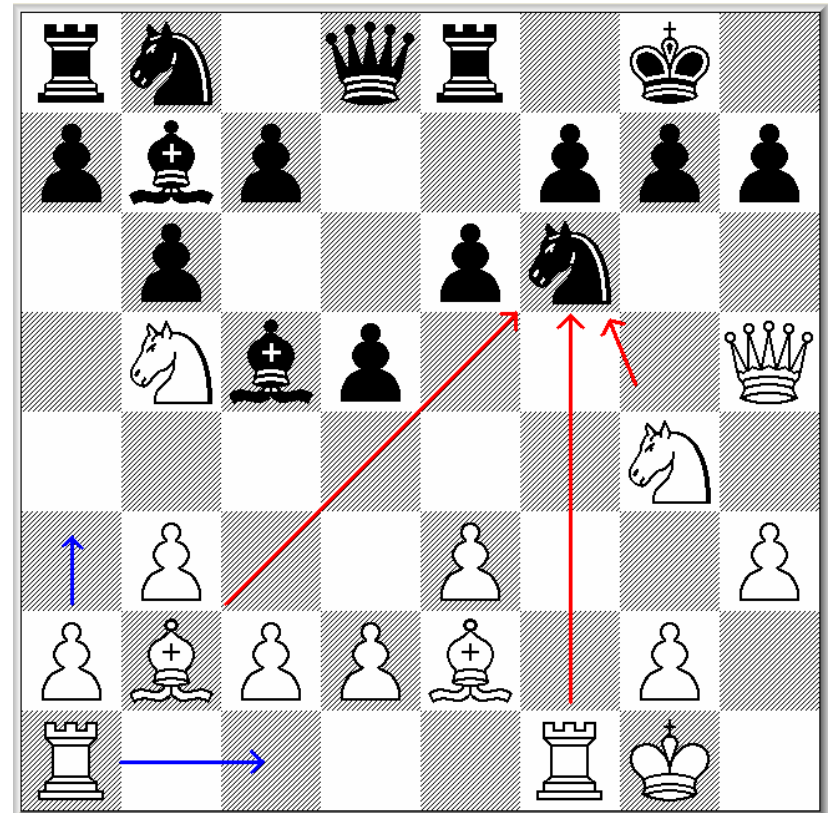
# Motivation

- ◆ Selektion guter Bedrohungen
- ◆ Schnelles Erkennen von schwachen Zügen
- ◆ Hier: Weis am Zug



# Motivation

- ◆ **Bedrohungen**
- ◆ **Irrelevante und schlechte Züge**
- ◆ Aufstellung von Plänen, die die Suche führen sollen
- ◆ „Lieber konsequent einem nicht optimalen Plan folgen, als planlos zu spielen“



# Paradise: Das Programm

- ◆ „**P**attern **R**ecognition **A**ppplied to **D**irecting **S**Earch“
- ◆ Erstellung von Konzepten mit einem **Produktionssystem**
- ◆ Zusammenführung dieser Konzepte zu Plänen und Zielen
- ◆ Suche nach dem besten Zug durch Orientierung an den Plänen
- ◆ Wissensbasis unterteilt in Wissensquellen

# Produktionssysteme

- ◆ Aufbau des Systems: Speicher und Regeln
- ◆ Inhalt im Speicher:  
( *Konstante1 Konstante2* )
- ◆ Allgemeine Form für eine Regel:  
*wenn <Bedingungen> dann <Aktionen>*
- ◆ Ausführungszyklus von OPS5:  
*Musterung > Regelauswahl > Aktion >*

# Produktionsregeln in Paradise

((Figur1)

(Keine Fluchtmöglichkeit für Figur1)

(Keine andere Figur deckt Figur1))

➤ Aktion Attackiere

((Gegner auf Position von Figur1)

(Drohe mit dem Gewinn von Figur1)

(Anzahl der nichtzwingenden Züge des Gegners))

# Produktionsregeln in Paradise

(dmp1)

(Never (Exists (sq)(Pattern Mobil dmp1 sq)))

(Never (Exists (p1)(Pattern Enpris p1 dmp1)))

(Action Attack

((Other-color dmp1)(location dmp1))

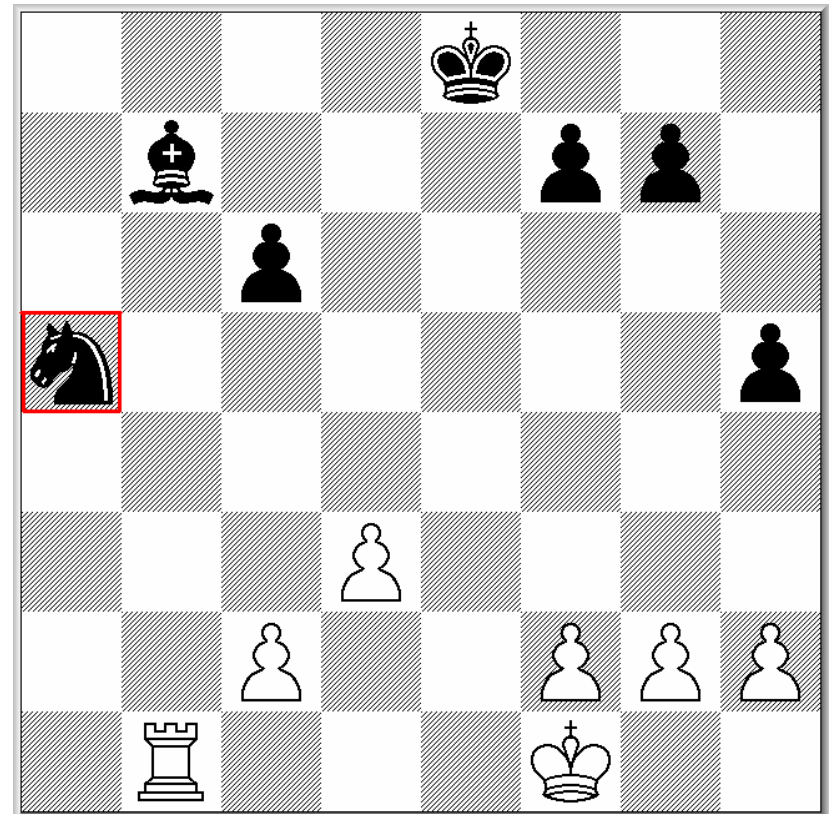
(Threat (Win dmp1))

(Likely 0))



# Beispiel

- ◆ Springer ist ungedeckt
- ◆ Springer hat kein Feld zum Ausweichen
- ◆ Folge: Mögliche Attacke des Feldes durch den Turm

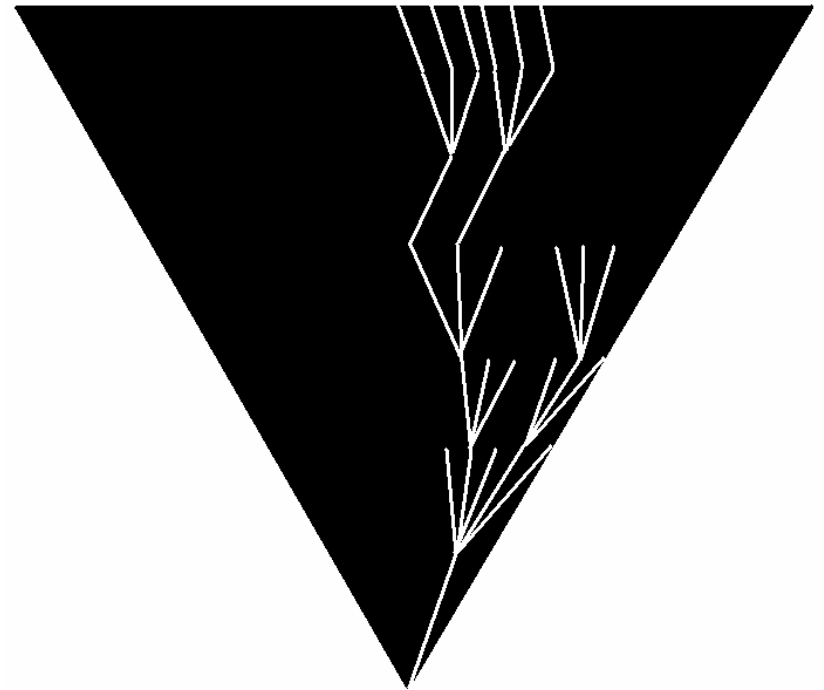


# Verwendung der Wissensbasis

- ◆ Grundlegende Stellungsbeschreibung mit z.B. ENPRIS, MOBIL etc.
- ◆ Statische Analyse
- ◆ Ausführung eines Plans während der Suche
- ◆ Gegnerische Züge
- ◆ Quiescence Suche

# Pläne

- ◆ Teilbaum eines Suchbaums
- ◆ Zu allgemein: Branching-Faktor wird zu groß
- ◆ Zu speziell: Einschränkung der Alternativen von gegnerischen Zügen
- ◆ Attributliste verschiedener Wissensquellen bzgl. eines Plans



# Beispiel - Plan

(weißer Springer F3)

((schwarzer Springer NIL)

(SAFEMOVE weißer Turm E7))

((ANYBUT schwarzer Springer)

(SAVECAPTURE

weißer Turm schwarzen Springer))

# Beispiel - Attribute

(THREAT (PLUS  
 (EXCHVAL weißer Springer F3)  
 (FORK weißer Turm,  
 schwarze Dame, schwarzer Läufer))  
(LIKELY 1)

# Beispiel - Attribute

(THREAT (PLUS

(EXCHVAL weißer Springer F3)

(EXCH weißer Turm schwarze Dame))

(LIKELY 1)

# Ablauf der Suche

Eingabe: Einer oder mehrere Pläne

- ◆ Sofortiger Gewinn?
- ◆ Sieht Plan einen Zug vor → neu suchen
- ◆ Offensichtlicher Gewinnplan?
- ◆ Durchsuche aktuellen Plan
- ◆ Komplette statische Analyse (teuer)

Ausgabe: Wertigkeiten der Suchen / weitere Informationen

# Kürzungspunkte der Suche

- ◆ Zugwiederholung
- ◆ Planwiederholung
- ◆ Niedrige Gewinnwahrscheinlichkeit
- ◆ Alpha-Beta Schnitt
- ◆ Besserer Plan höheren Levels
- ◆ Ursachenforschung (causality facility)



# Statistik der Ressourcen

	Mittel	Hoch	Niedrig
Zeit für das Gesamtproblem	332.9	1958	19
Knoten	38.06	215	3
% Primitive Berechnen	52.85	74.3	34.1
Zeit einer statischen Analyse	12.22	26.5	2.2
Anzahl statischer Analysen	3.73	35	0
% Zeit aller stat. Analysen	11.61	33.7	0.0
% Zeit Bearbeitung der WB	91.6	97.7	81.5

# Paradise in der Praxis

- ◆ Baumgröße von Chess 4.4 bei 28.496,8
- ◆ Spezialisierung auf Mittel-End-Positionen
- ◆ 89 von 92 Schachproblemen gelöst
- ◆ In der Entwicklung 21 Probleme
- ◆ Leichte Korrekturen in der Wissensbasis
- ◆ Zufügen neuer Produktionsregeln
- ◆ Problem: Zu viele neue Regeln kosten mehr Zeit

# Quellen

Wilkins, D. E. - Using Patterns and Plans in Chess, 1980

Wilkins, D. E. - Using Knowledge to Control Tree Search Searching

Bibel, W. – Wissensrepräsentation und Inferenz, 1993

Stuart J. R., Peter N. - Artificial Intelligence: A Modern Approach, 1995

