

## Computer chess

- chess-playing programs essentially explore huge numbers of potential future moves by both players and apply a relatively simple evaluation function to the positions that result
  - [directed reading](http://en.wikipedia.org/wiki/Computer_chess) <http://en.wikipedia.org/wiki/Computer\_chess>
  - [end games](http://en.wikipedia.org/wiki/Endgame_tablebase) <http://en.wikipedia.org/wiki/Endgame\_tablebase>
- in the early days of computer chess, 1950s it was thought that by studying how a machine could play chess we would understand how the human brain works
  - alas this proved to be untrue

## Computer chess

- Claude Shannon wrote the first paper on computer chess in 1950
- in his paper he mentions two techniques which could be used to solve this problem
  - type A strategy - brute force looking ahead say 6 moves and applying the minimax algorithm (AlphaBeta was not discovered until a decade later)
  - type B strategy - which examines only the *interesting* moves at each ply
- he concluded that type B was the better technique
  - time has shown the opposite!
  - easier to compute all moves (type A), than write a program to work out the *interesting* moves - and faster..

## Computer chess

- Shannon estimated that with approximately thirty moves possible in a typical real-life position, he expected that searching the approximately  $30^6$  (over 700,000,000) positions involved in looking three moves ahead for both side (six plies) would take about sixteen minutes, even in the "very optimistic" case that the program evaluated a million positions every second
  - it took about forty years to achieve this speed

## Modern chess programs

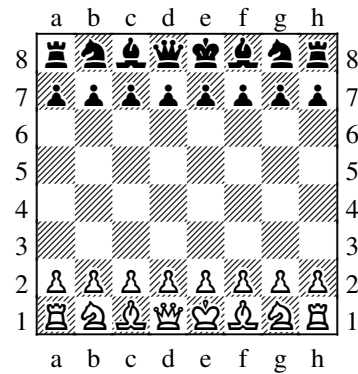
- type B chess programs were abandoned in 1973 in favour of type A style programs
- modern chess programs evaluate a full ply width of moves up until a certain level
  - then they selectively reduce the width
  - try and eliminate null moves (circular board positions etc)

## End game tablebases

- one of the severe weaknesses of computer chess was in the end game
- much work was done by Ken Thompson and others (1977 onwards) to explore check mate and the move before this
  - and the move before this etc..
- they found some surprising facts
  - technically king and rook can draw against king and queen
  - before it was generally accepted that king and queen could win
- read the wikipedia entry for more details

## Chess evaluation function within a game tree

■



## Chess evaluation function within a game tree

- in these notes the board will be labelled a-h along the x-axis and 1-8 along the y-axis

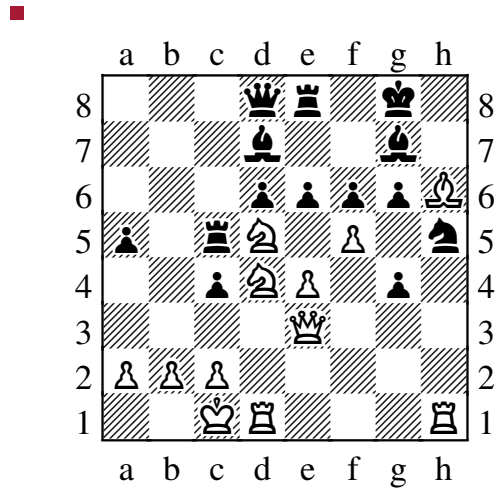
## Evaluation function

- in Othello we have seen that a simple material evaluation function delivers a reasonably strong computer opponent
- in Chess, just using material in an evaluation function requires an enormous amount of computing speed to deliver good results
  - thus even the most powerful of chess computers use a more sophisticated evaluation function than just material advantage

## Chess evaluation function

- calculates material
  - ∞ king
  - 900 Queen
  - 500 Rook
  - 300 Knight
  - 300 Bishop
  - 100 Pawn
- maybe Knight are worth more than Bishops...
  - maybe a Rook is worth two Knights...
  - possibly these values should change over the stages of the game

## Chess evaluation function



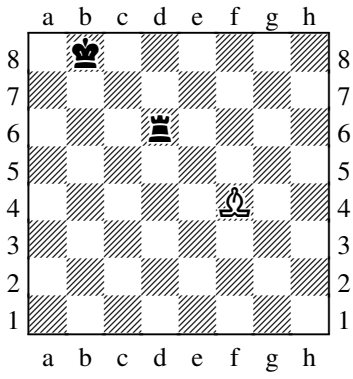
## Positional evaluation

- mobility
- development and castling
- centre control
- king attack

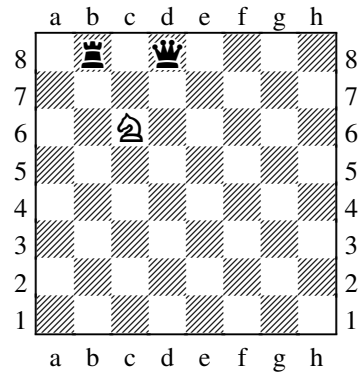
## Positional evaluation

- king safety
- pawn structure
- piece attack and piece defense

### Chess Pin



### Chess Fork



### Pawn Structure

