Games and Search

“Unpredictable” opponent ⇒ solution is a contingency plan

Time limits ⇒ unlikely to find goal, must approximate

Plan of attack:

- algorithm for perfect play (Von Neumann, 1944)
- finite horizon, approximate evaluation (Zuse, 1945; Shannon, 1950; Samuel, 1952–57)
- pruning to reduce costs (McCarthy, 1956)
Game Trees and Minimax

Perfect play for deterministic, perfect-information games

Idea: choose move to position with highest minimax value
   = best achievable payoff against best play

E.g., 2-ply game:

```
function Minimax-Decision(game) returns an operator
    for each op in Operators[game] do
        Value[op] ← Minimax-Value(Apply(op, game), game)
    end
    return the op with the highest Value[op]

function Minimax-Value(state, game) returns a utility value
    if Terminal-Test(game)(state) then
        return Utility[game](state)
    else if MAX is to move in state then
        return the highest Minimax-Value of Successors(state)
    else
        return the lowest Minimax-Value of Successors(state)
```
Evaluation Functions

For chess, typically \( \text{linear weighted sum of features} \)

\[
\text{Eval}(s) = w_1f_1(s) + w_2f_2(s) + \ldots + w_nf_n(s)
\]

e.g., \( w_1 = 9 \) with

\( f_1(s) = \text{(number of white queens)} - \text{(number of black queens)} \)

etc.
Close Enough is Good Enough

MAX

MIN

Behaviour is preserved under any *monotonic* transformation of *Eval*

Only the order matters:
- payoff in deterministic games acts as an *ordinal utility function*
Cutting of Search

\textsc{MinimaxCutoff} is identical to \textsc{MinimaxValue} except
1. \textsc{Terminal?} is replaced by \textsc{Cutoff}?
2. \textsc{Utility} is replaced by \textsc{Eval}

Does it work in practice?

\[
b^m = 10^6, \quad b = 35 \quad \Rightarrow \quad m = 4
\]

4-ply lookahead is a hopeless chess player!

4-ply $\approx$ human novice
8-ply $\approx$ typical PC, human master
12-ply $\approx$ Deep Blue, Kasparov