

100 Years of Shannon: Chess, Computing and Botvinik

Iryna Andriyanova

▶ **To cite this version:**

Iryna Andriyanova. 100 Years of Shannon: Chess, Computing and Botvinik. Doctoral. United States. 2016. cel-01709767

HAL Id: cel-01709767
https://hal.science/cel-01709767

Submitted on 15 Feb 2018

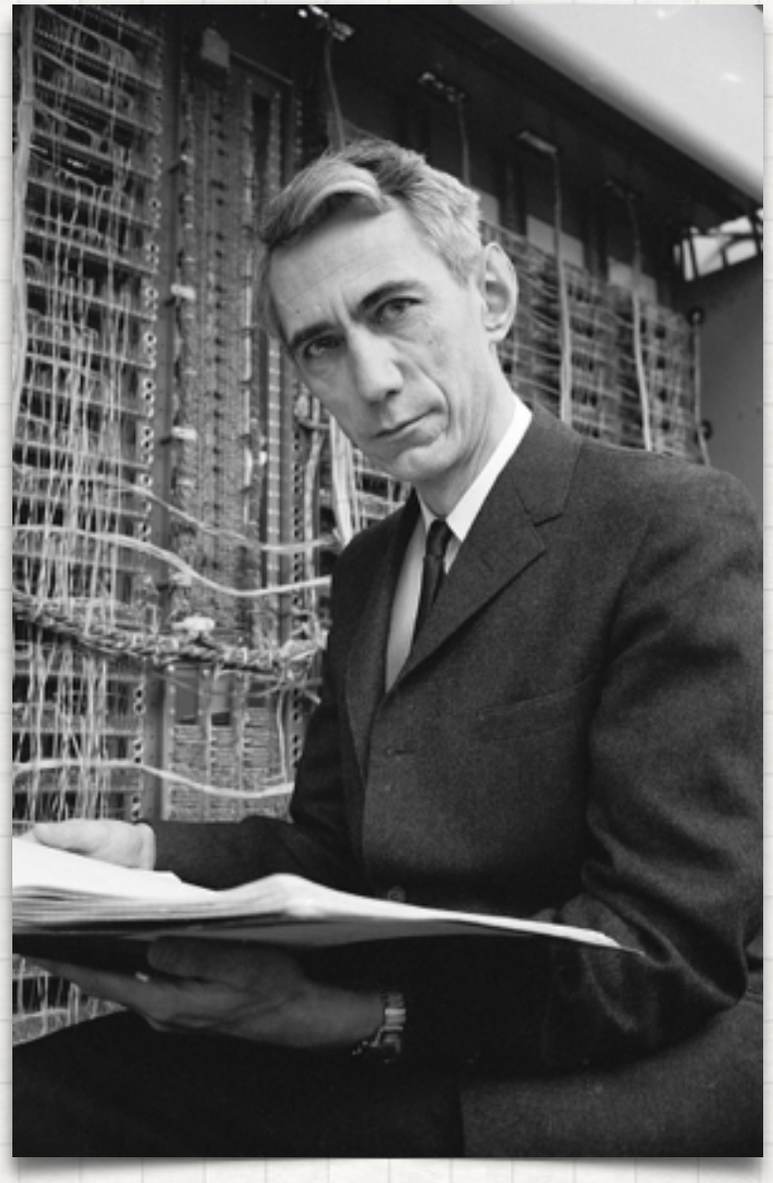
HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

CLAUDE E. SHANNON

1916–2001

- 1936 : Bachelor in EE and Mathematics from U.Michigan
- 1937 : Master in EE from MIT
- 1940 : PhD in EE from MIT
- 1940 : Research fellow at Princeton
- 1940-1956 : Researcher at Bell Labs
- 1948 : “A Mathematical Theory of Communication”, Bell System Technical Journal
- 1956-1978 : Professor at MIT



100 YEARS OF SHANNON

SHANNON, BOTVINNIK AND COMPUTER CHESS

Iryna Andriyanova

ETIS Lab, UMR 8051,
ENSEA/Université de Cergy-Pontoise/CNRS

54TH ALLERTON CONFERENCE

SHORT HISTORY OF

CHESS/COMPUTER CHESS



CHESS AND COMPUTER CHESS

TWO PARALLELS



W. Steinitz: simplicity and rationality

E. Lasker: more risky play

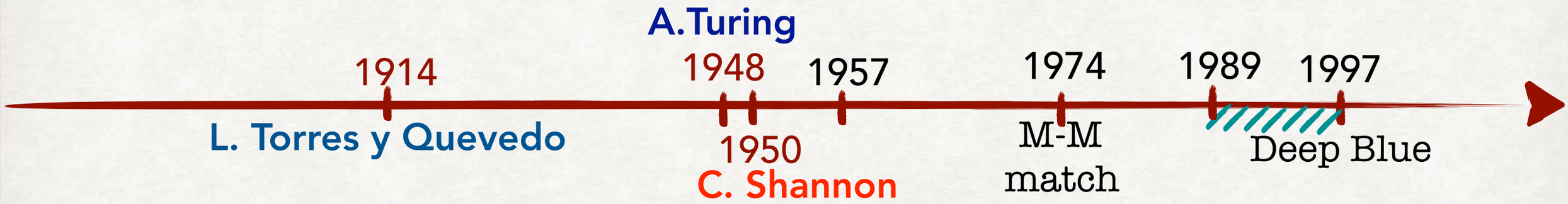
M. Botvinnik: complicated, very original positions

A. Karpov, G. Kasparov: Botvinnik's school

CHESS AND COMPUTER CHESS

TWO PARALLELS





A. Turing: TurboChamp algorithm

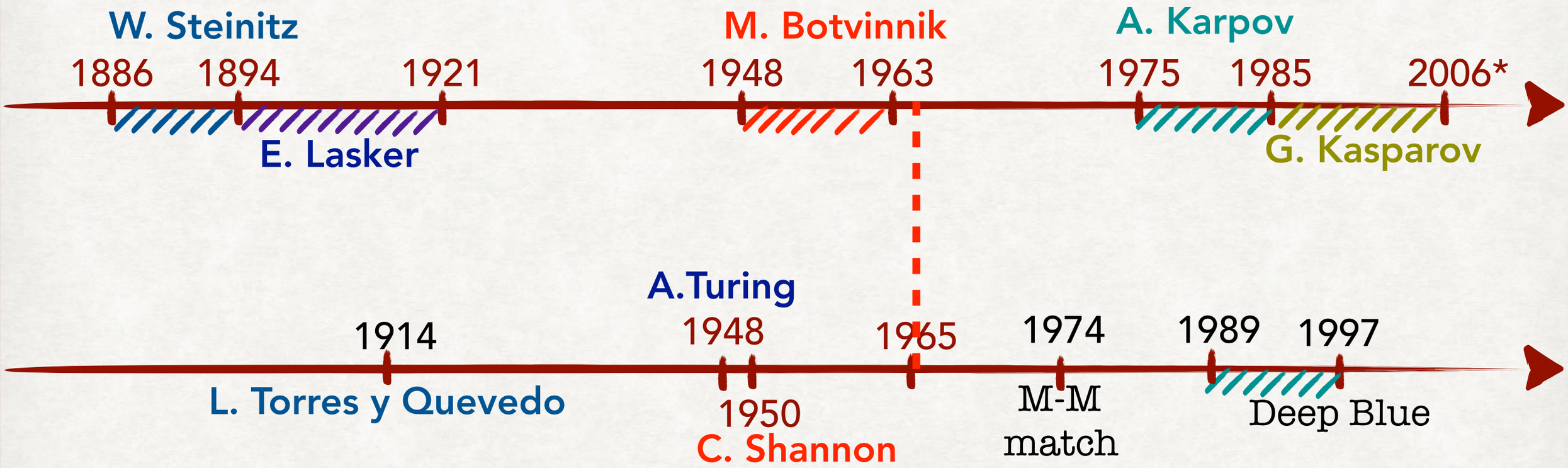
C. Shannon: "Programming a Computer for Playing Chess"

1957, MANIAC: first simple chess machine

1974, first chess machine world championship

CHESS AND COMPUTER CHESS

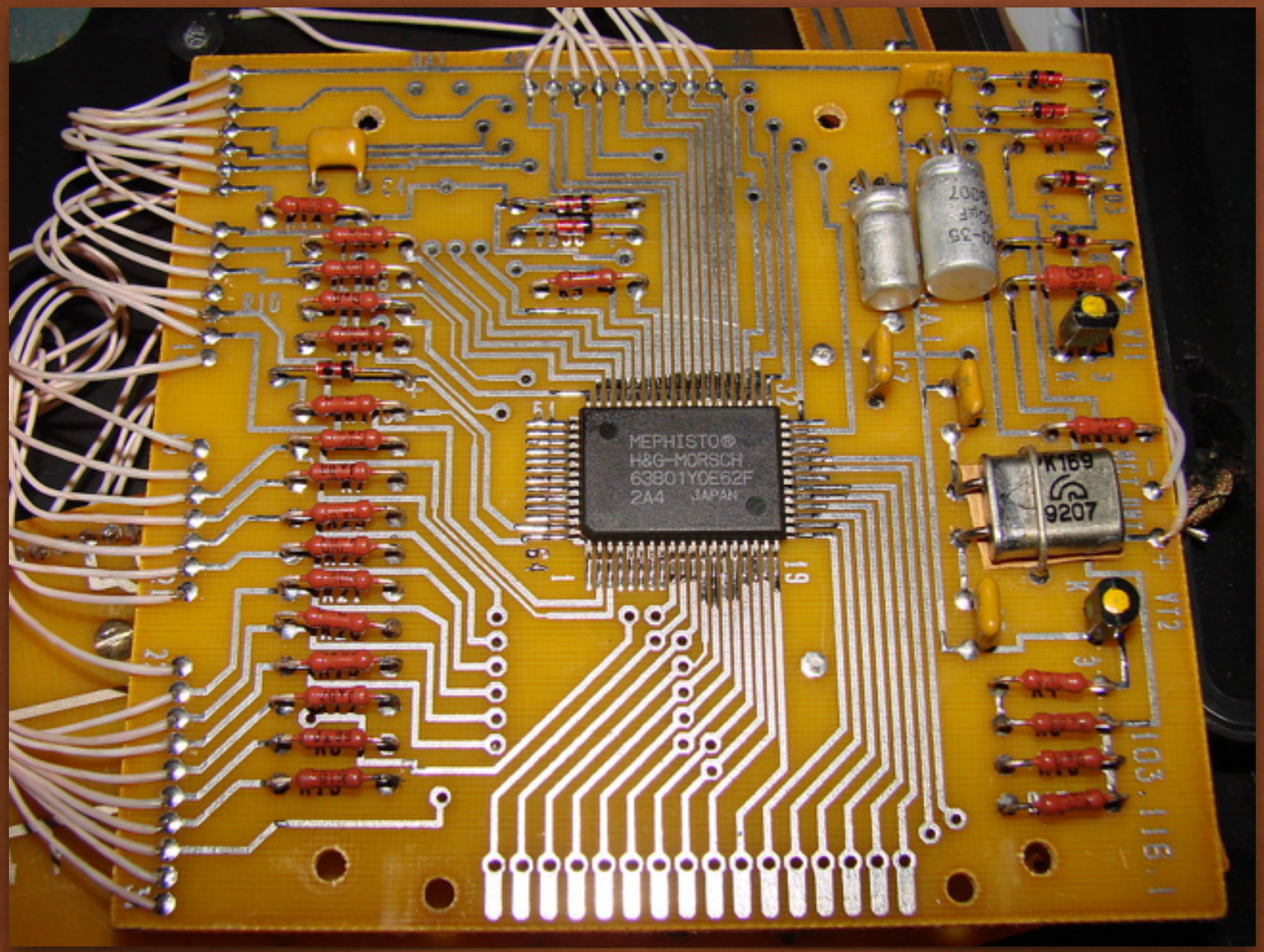
TWO PARALLELS



- Match between Shannon and Botvinnik in 1965
- “An Algorithm for Chess” by Botvinnik in 1968

INSIDE A

COMPUTER CHESS PROGRAM



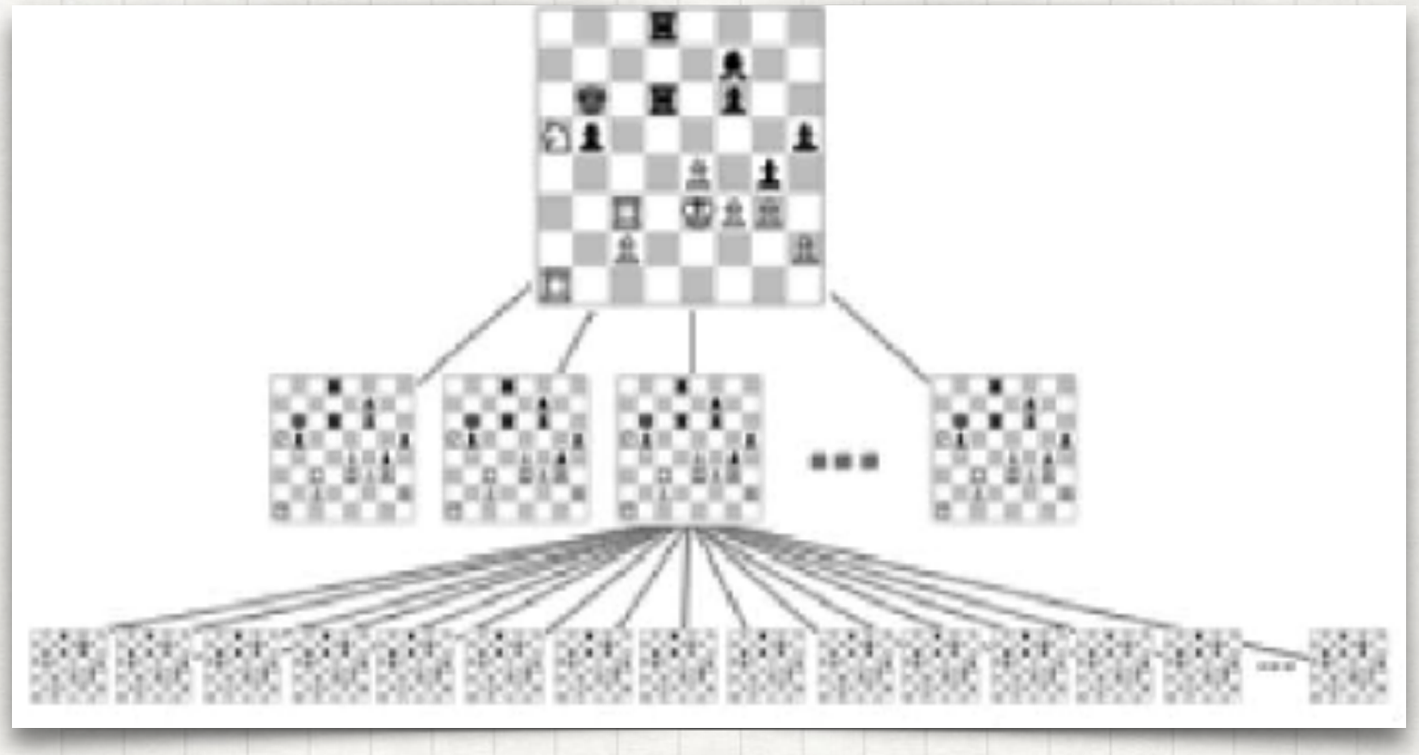
WHAT A MODERN CHESS PROGRAMM HAS INSIDE?

- Board representation
- Move generator
- Database of openings
- Tree searching techniques
- Positional evaluation
- Transposition tables

0,0	X	0,2	0,3	0,4	0,5	0,6	X
1,0	1,1	X	1,3	1,4	1,5	X	1,7
2,0	2,1	2,2	X	2,4	X	2,6	2,7
3,0	3,1	3,2	3,3	♗	3,5	3,6	3,7
4,0	4,1	4,2	X	4,4	X	4,6	4,7
5,0	5,1	X	5,3	5,4	5,5	X	5,7
6,0	X	6,2	6,3	6,4	6,5	6,6	X
X	7,1	7,2	7,3	7,4	7,5	7,6	7,7

WHAT A MODERN CHESS PROGRAMM HAS INSIDE?

- Board representation
- Move generator
- Database of openings
- Tree searching techniques
- Positional evaluation
- Transposition tables

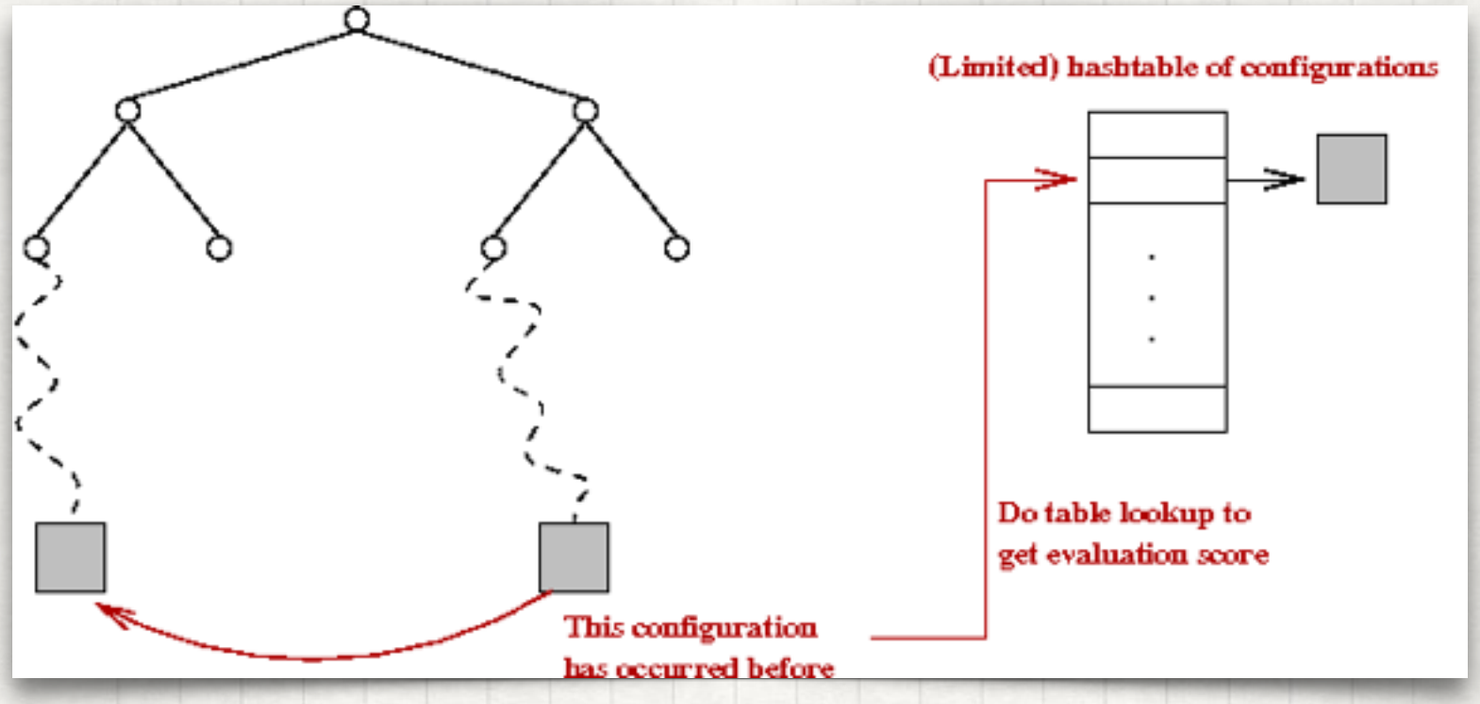


```
materialScore = kingWt   * (wK-bK)
              + queenWt  * (wQ-bQ)
              + rookWt   * (wR-bR)
              + knightWt* (wN-bN)
              + bishopWt* (wB-bB)
              + pawnWt * (wP-bP)

mobilityScore = mobilityWt * (wMobility-bMobility)
```

WHAT A MODERN CHESS PROGRAMM HAS INSIDE?

- Board representation
- Move generator
- Database of openings
- Tree searching techniques
- Positional evaluation
- Transposition tables



STRATEGIES BY C. SHANNON (1950)

FIRST FORMALISM OF COMPUTER CHESS

- **Sum-zero utility function**

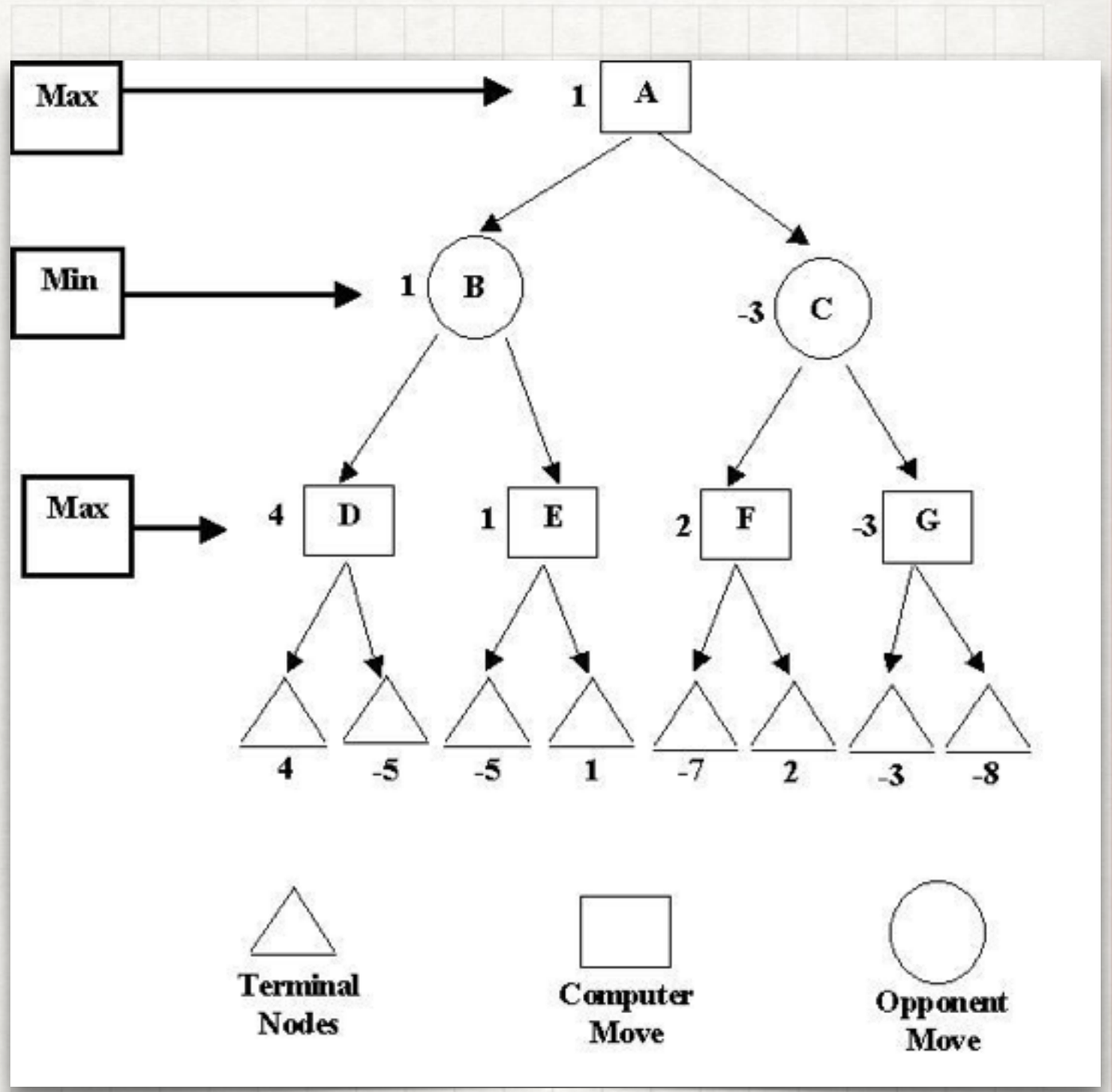
```
f(p) = 200(K-K')
       + 9(Q-Q')
       + 5(R-R')
       + 3(B-B' + N-N')
       + 1(P-P')
       - 0.5(D-D' + S-S' + I-I')
       + 0.1(M-M') + ...

KQRBNP = number of kings, queens,
rooks, bishops, knights and pawns

D,S,I = doubled, blocked and
isolated pawns

M = Mobility (the number of
legal moves)
```

- **Minimax evaluation** [vonNeuman'47]

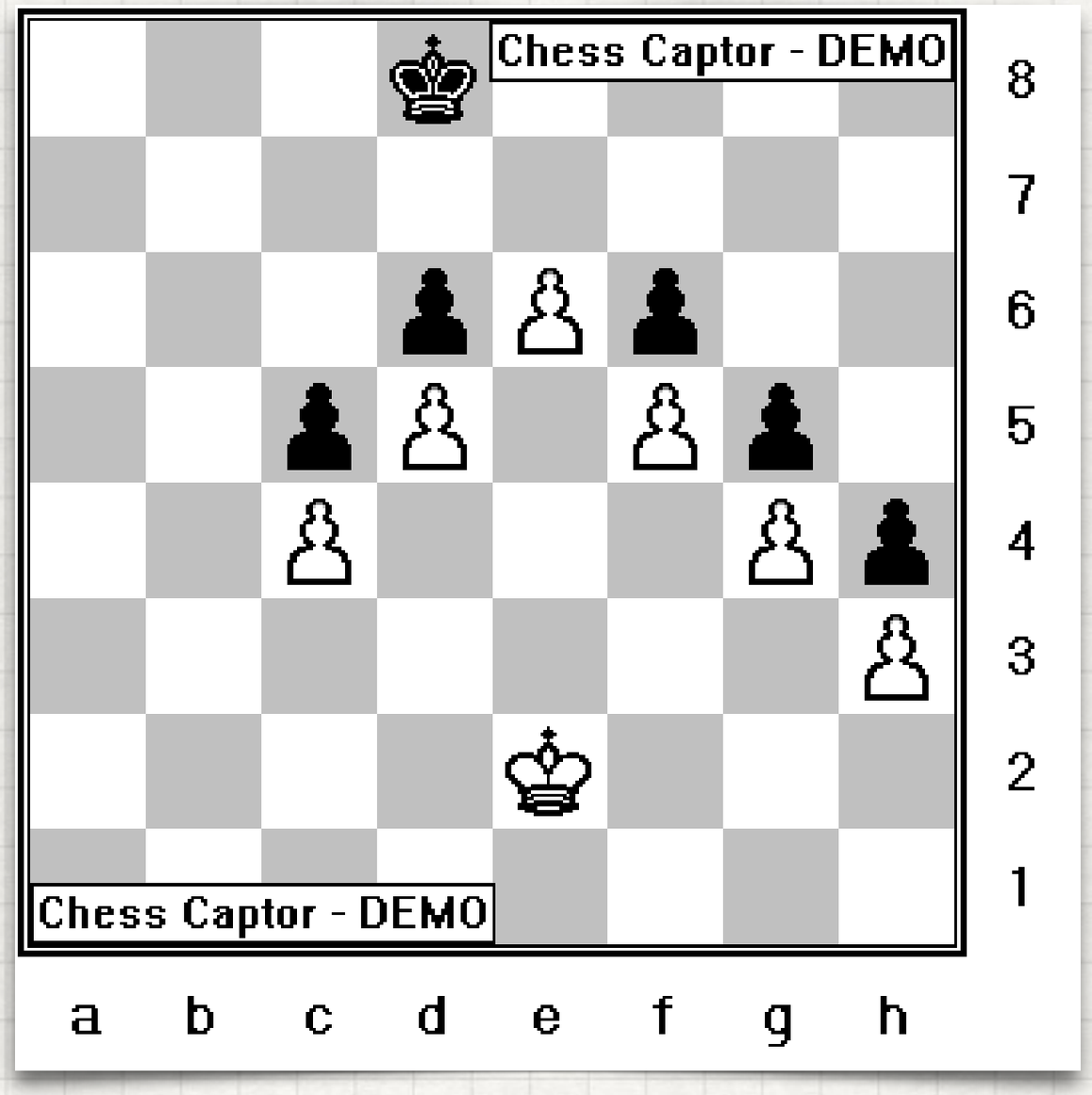


STRATEGIES BY C. SHANNON (1950)

TYPES A AND B

- **Type A strategy**
  search over all possible variations
  - *exponential explosion*
  - *horizon effect*

- **Type B strategy**
  consideration of carefully chosen better moves (human-like)
  - *"plausible move generators"*
  - *successive learning*
  - *the search depth is unlimited*



Position 1: White has a trivial win, but because it requires a 10 move maneuver a trivial type A strategy will not find it.

COMPUTER CHESS EVOLUTION



= COMPUTERS EVOLUTION

COMPUTER CHESS EVOLUTION

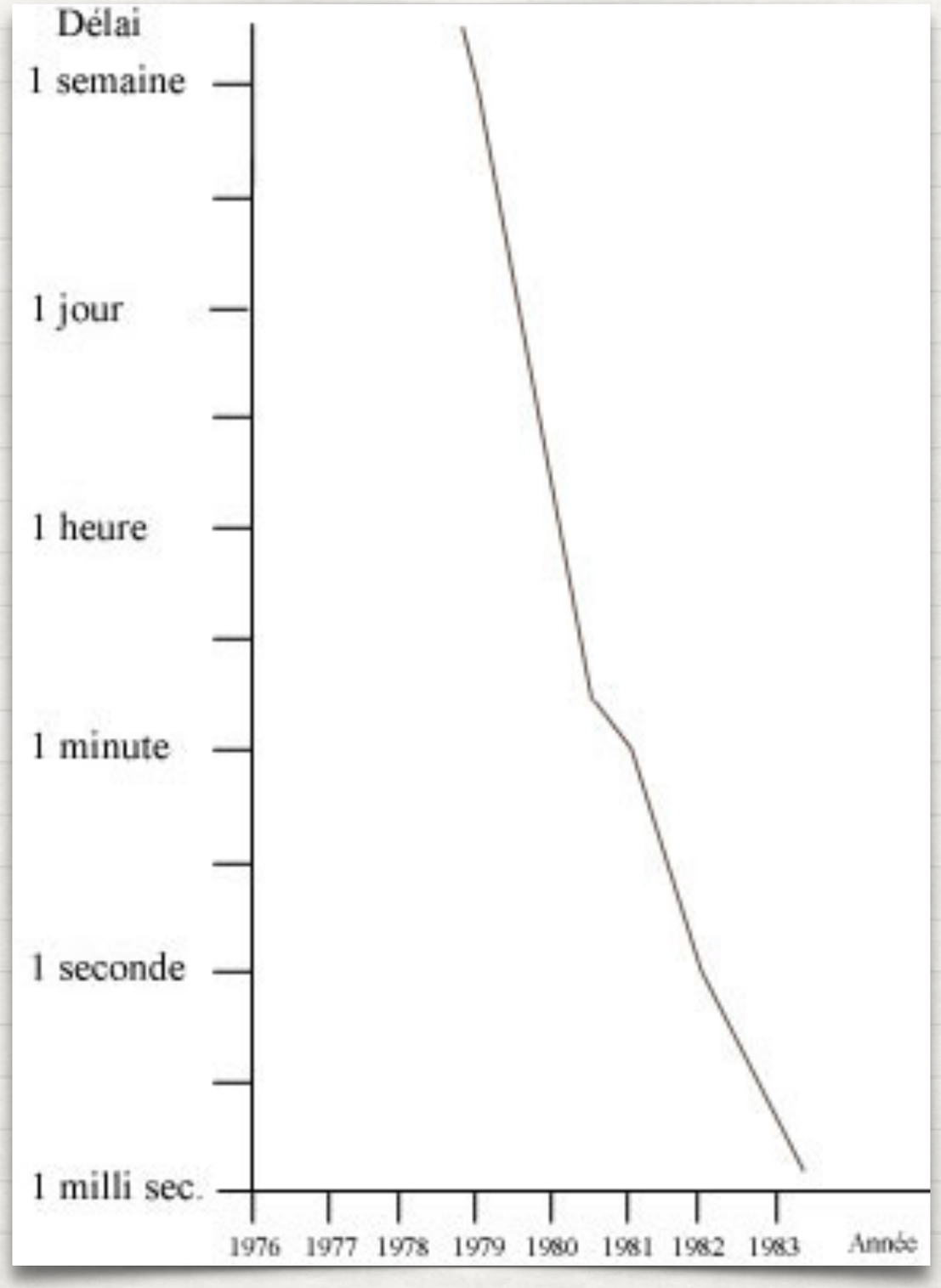
= COMPUTERS EVOLUTION

Toy example:



White give a checkmate in 3 moves:

1. Fxh7+ Rf8
2.Cg6+ fxg6
3.Tf4×



COMPUTER CHESS EVOLUTION

FROM 1970'S TILL NOWADAYS

- **Improved Type-A algorithms**
  - alpha-beta pruning
  - killer heuristic

  no need for very good plausible move generators

- **Improved computers**
  - parallelized processing
  - special chess hardware
  - memory volume
  - cheap disk storage

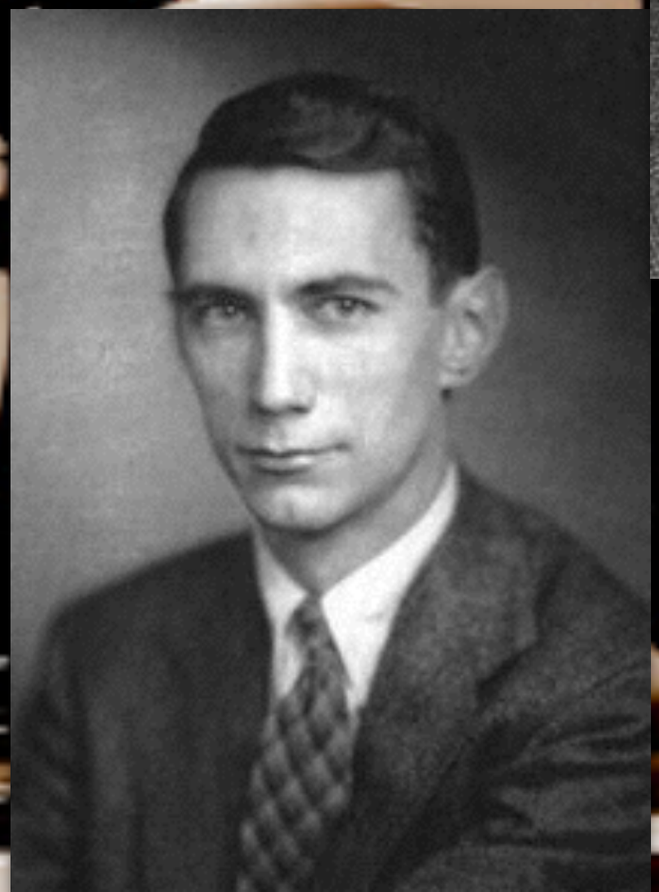


Kasparov against Deep Blue

Deep Blue chess machine : 30 parallel processors + 480 custom chess chips, 10 years of development

STRATEGY OF TYPE B



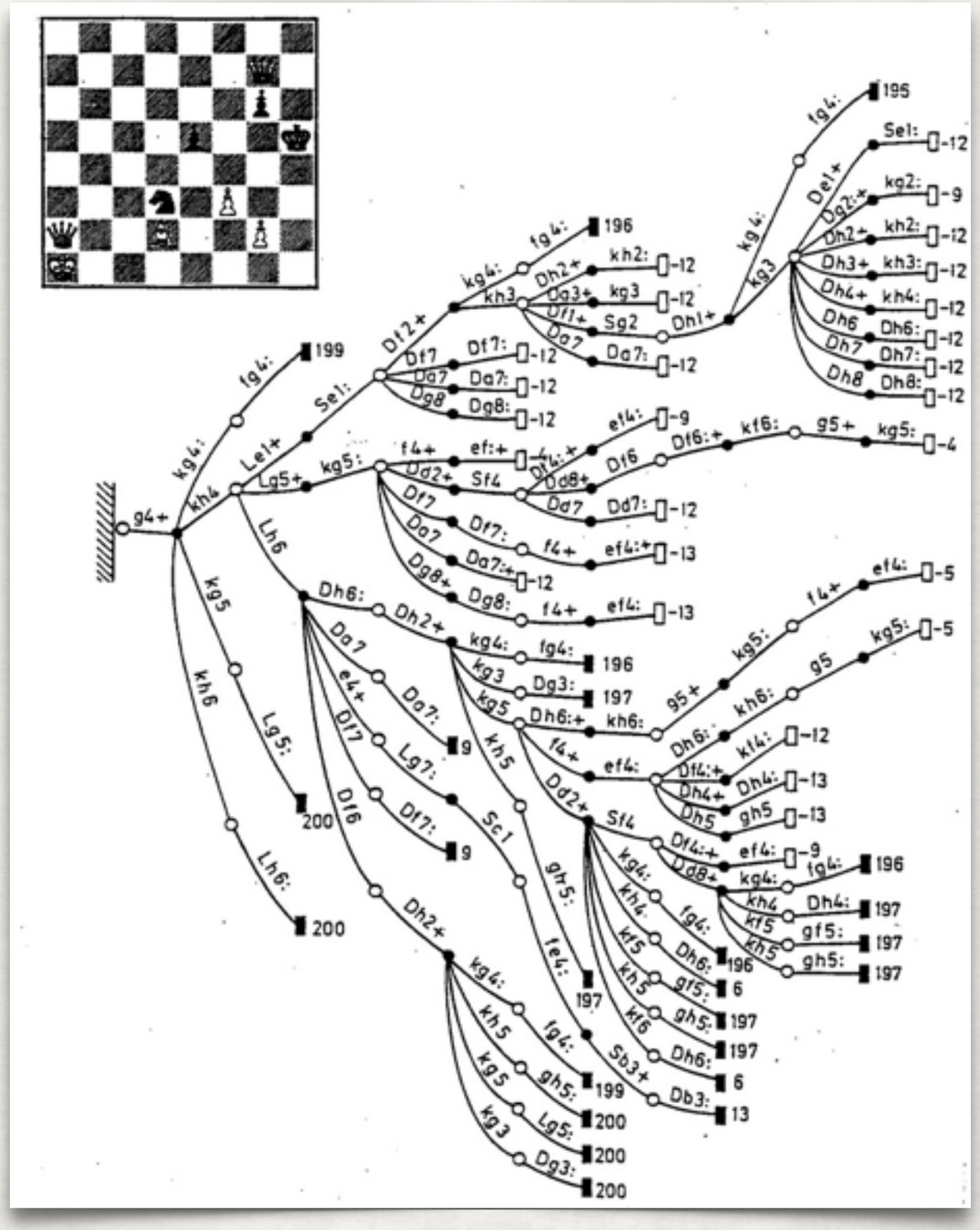


TYPE-B STRATEGY

BY C. SHANNON

(1)Examine forceful variations out as far as possible and evaluate only at reasonable positions, where some quasi-stability has been established.

(2) Select the variations to be explored by some process so that the machine does not waste its time in totally pointless variations.



Example from [Shannon'50]:

g(P) = 1 if any piece is attacked by a piece of lower value or by more pieces than defences or if any check exists on a square controlled by the opponent
g(P) = 0 otherwise

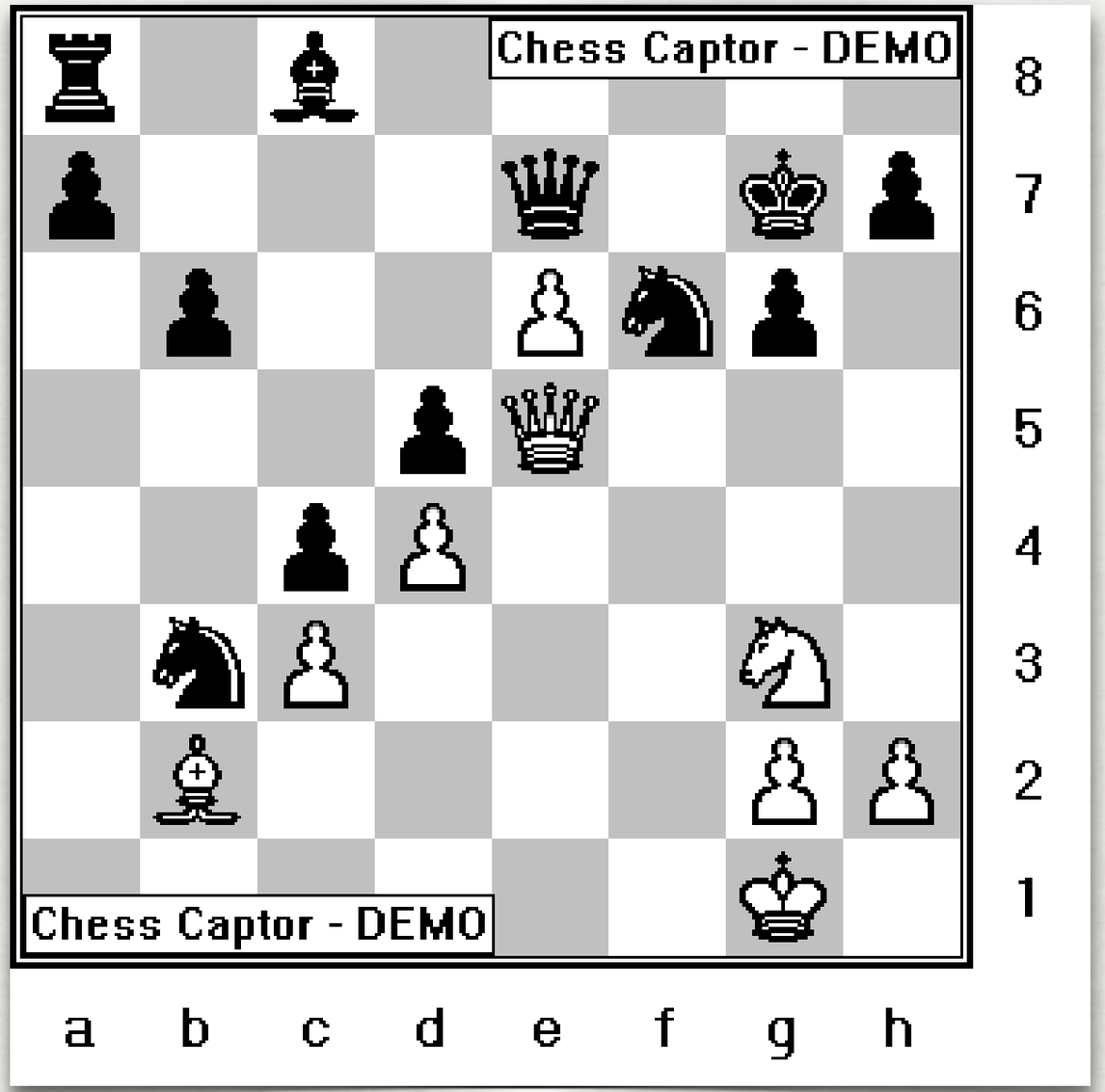
TYPE-B STRATEGY

BY M. BOTVINNIK

- "An Algorithm in Chess" 1968
- *Pioneer* project 1970s-1980s:

  - notion of attack maps
  - notion of trajectories and adversary counter-trajectories

  - endgame library with base positions, nodal positions and scores



Position 3: Botvinnik-Capablanca 1938. By playing 1.Ba3!! White starts a remarkable combination which gives it a victory 12 moves later. This position was used by Botvinnik as a test case for his chess algorithms.

CONCLUSION



I never considered Deep Blue intelligent in any way. It's just an excellent problem solver in this very specific domain.

— *M. Campbell, Deep Blue team member*

Brute-force alpha-beta may have done as much damage to the process and credibility of computer-chess research as it has given chess programs high ratings.

— *Donskoy, KAISSA developer*

MESSAGE TO TAKE HOME

- **B-type strategy**
  - by Shannon
  - by Botvinnik
  - machine learning

- **Contribution of computer chess to science**
  - alpha-beta pruning
  - transposition tables
  - artificial intelligence