

Chess and AI

A love story

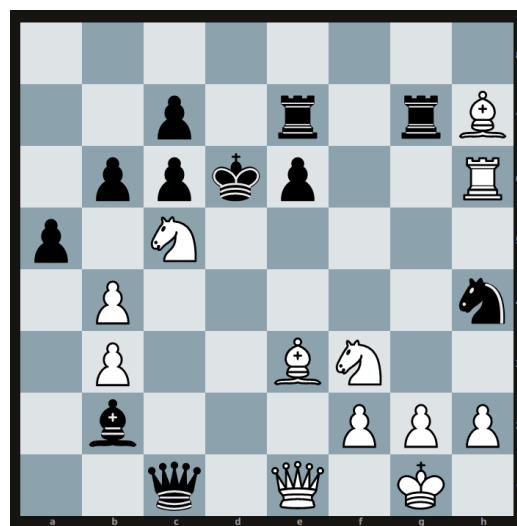
Giovanni Squillero
squillero@polito.it



2

2

White to move



Chess & AI

squillero@polito.it

5

5

1

Solution

Engine: Stockfish 15
Stopped 4 CPUs Depth=245/8 19970 kN/s 0:00:01
+- #4 1. Qxf4+ Bxf4 2. Bxe6+ Qd5 3. Qe4+ Bxe4 4. Bxe4#
White mates: +- (#4) Depth: 245/8 00:00:00 659kN

Engine: Lc0 v0.28.0
Lc0 v0.28.0 2 CPUs 20 kN/s
+- #4 Depth=8/22 1. Qxf4+ Bxf4 2. Bxe6+ Qd5 3. Qe4+ Bxe4 4. Bxe4#
White mates: +- (#4) Depth: 8/22 00:00:02 37kN

#4 Stockfish 14+ NNUE Depth 245/245, 10781k nodes/s ✓
1. Qxf4+ Bxf4 2. Bxe6+ Qd5 3. Qe4+ Bxe4 4. Bxe4# ▾

squillero@polito.it Chess & AI 6

6

White to move (2)

squillero@polito.it Chess & AI 7

7

White to move (2)

Lc0 v0.28.0

Engine: Lc0 v0.28.0
Lc0 v0.28.0 Depth=14/30 2 CPUs 2966
1. $\text{d}1\text{a}4+$ $\text{d}8\text{c}6$ 2. $\text{d}1\text{e}4+$ $\text{d}8\text{b}5$ 3. $\text{d}1\text{x}c1$ $\text{d}7\text{x}f3+$ 4. $\text{d}1\text{x}f3$ $\text{b}6\text{c}5$ 5. $\text{d}1\text{x}b2$ $\text{c}6$ 6. $\text{d}1\text{e}5$ $\text{d}7\text{x}g2+$ 7. $\text{d}1\text{x}g2$ $\text{d}6$
8. $\text{d}1\text{b}8$ $\text{d}7$ 9. $\text{b}5+$
White is clearly winning: +- (30.98) Depth: 14/30 00:04:11 744kN

Stockfish 15

Engine: Stockfish 15
Stockfish 15 Depth=50/20 4 CPUs 4969 kN/s 1.0%
1. $\text{d}1\text{f}4+$ $\text{d}8\text{c}6$ 2. $\text{d}1\text{e}4+$ $\text{d}8\text{b}5$ 3. $\text{d}1\text{d}3+$ $\text{d}6\text{c}6$ 4. $\text{d}1\text{d}4+$ $\text{d}8\text{x}d4$ 5. $\text{b}5+$ $\text{d}5\text{d}5$ 6. $\text{d}1\text{e}4+$ $\text{d}8\text{x}c5$ 7. $\text{d}1\text{x}c1+$
 $\text{d}8\text{b}4$ 8. $\text{d}1\text{c}4+$ $\text{d}8\text{a}3$ 9. $\text{d}1\text{c}1+$ $\text{d}8\text{a}2$ 10. $\text{d}1\text{a}4\#$
White mates: +- (#10) Depth: 50/20 00:00:02 7151kN, tb=3

8

8

1770–1854: The Turk

- First machine able to play chess at a good level
- Run from 1770 to 1854 (when it was destroyed in a fire)



squillero@polito.it

Chess & AI

9

9

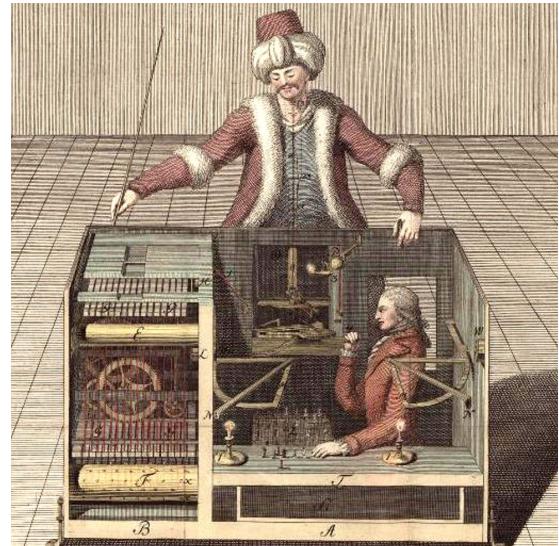
3

1857: The Turk

- First machine able to play chess at a good level
- Run from 1770 to 1854 (when it was destroyed in a fire)
- Discovered to be a **hoax** in 1857

squillero@polito.it

Chess & AI



10

More Fake

- 1868: Ajeeb (عجيب)
 - “Particularly intriguing” piece of faux mechanical technology
 - Played against Harry Houdini, Theodore Roosevelt, and O. Henry (William Sydney Porter) — also played checkers
- 1876: Mephisto
 - Remotely controlled

squillero@polito.it

Chess & AI



11

1912: El Ajedrecista

- Automaton built in 1912 by Leonardo Torres y Quevedo
 - True autonomous machine capable of playing chess
 - Only endgames KRvK, moving the white king and rook to checkmate the black king moved by a human opponent
- It checkmates the opponent every time
 - ... but does not checkmate in the minimum number of moves, nor within 50 moves



squillero@polito.it

Chess & AI

12

1948: Turochamp

- Originally developed by Alan Turing and David Champernowne
 - Turing failed to hack a Ferranti Mark 1 version in 1951
 - Turing run the program “by hand” against Alick Glennie in 1952
 - Depth: 2 + some heuristics
- Working version in 1951 for Ferranti Mark 1
 - Able to solve mate-in-2 problems
- Recreation of Turochamp constructed in 2012
 - Played against Kasparov at the Alan Turing Centenary Conference



squillero@polito.it

Chess & AI

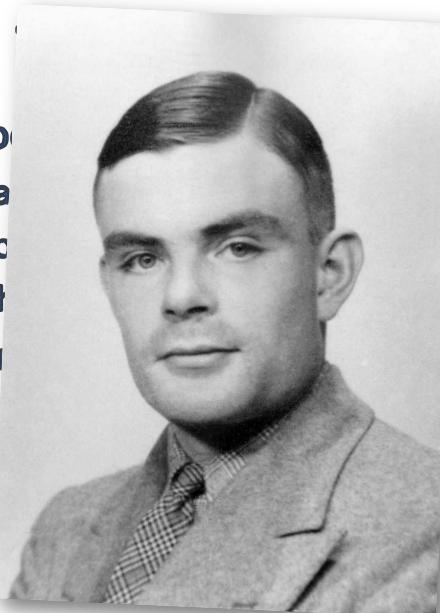
13

13

- Originally developed chess program
 - Turing failed to have it play against himself
 - Turing ran the program on his own machine
 - Depth: 2 + some branching factor
- Working version implemented
 - Able to solve mat endgame positions
- Recreation of Turing's chess program
 - Played against Kasparov at the Alan Turing Centenary Conference

squillero@polito.it

Chess & AI



12

David Champernowne
in 1951
Glennie in 1952



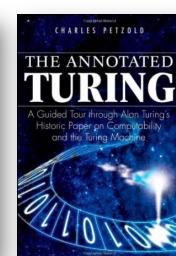
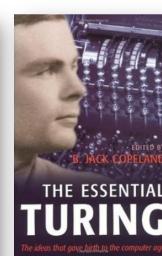
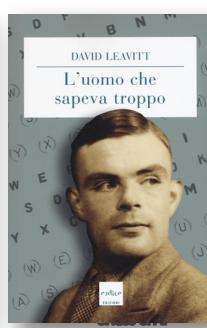
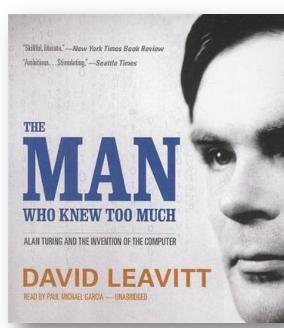
14

14

Suggested Readings

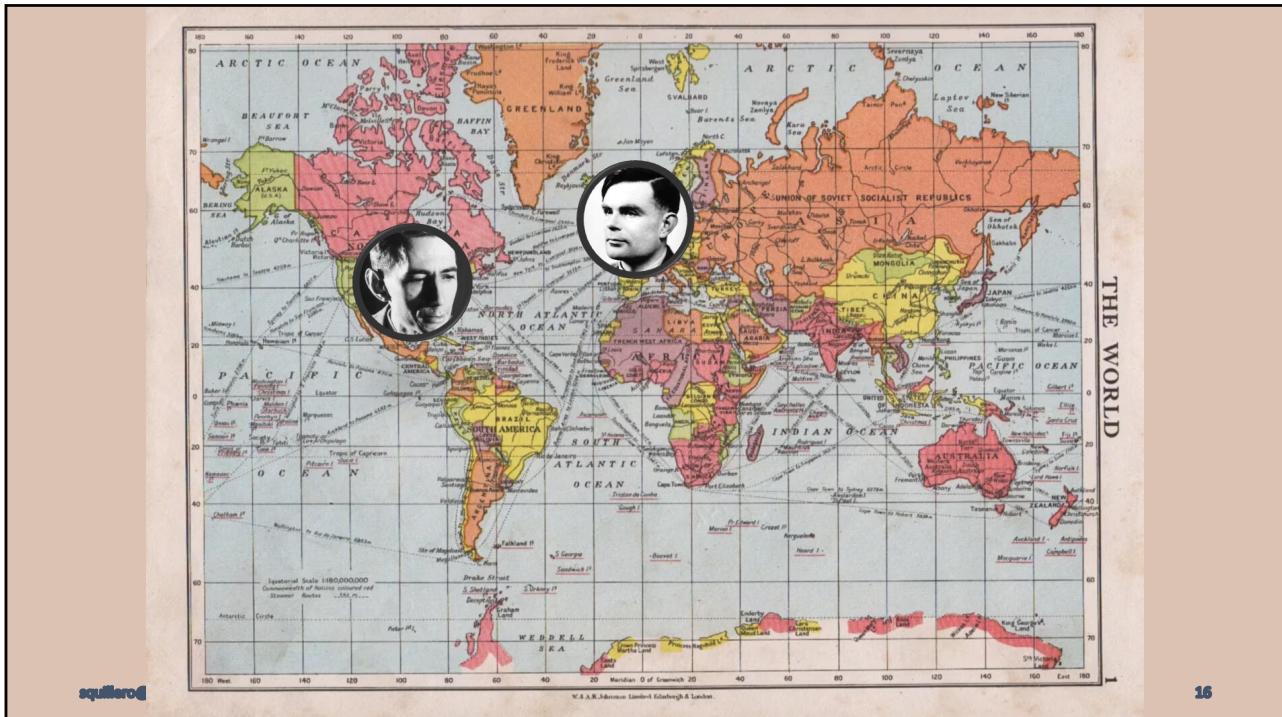
- David Leavitt
 - *The Man Who Knew Too Much: Alan Turing and the Invention of the Computer*, 2006
 - *L'uomo che sapeva troppo*, 2016 (Carolina Sargian, traduttore)

squillero@polito.it



15

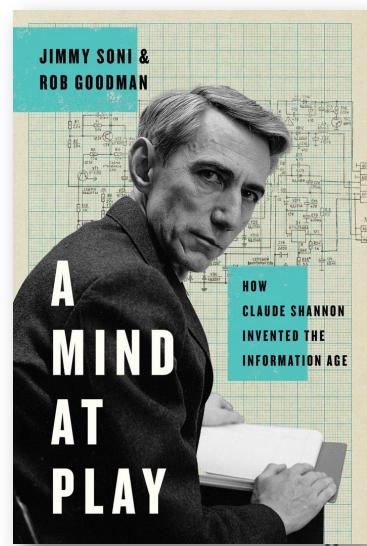
15



16

Suggested Readings

Jimmy Soni & Rob Goodman:
***A Mind at Play: How Claude Shannon
 Invented the Information Age***
 Simon & Schuster, 2017



squillero@polito.it

Chess & AI

17

1949: Caissac

- Named after Caïssa, the fictional “patron goddess of chess”
- 150 relay switches to perform calculation
- 10÷15 s/move
- Only endgame, max 6 pieces



18

1949: Caissac

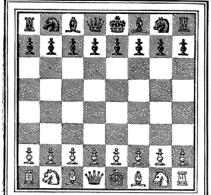
- Named after Caïssa, the fictional “patron goddess of chess”
- 150 relay switches to perform calculation
- 10÷15 s/move
- Only endgame, max 6 pieces



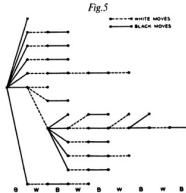
19

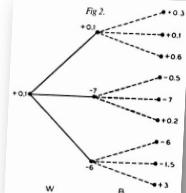
1950: Theory

- Claude Shannon: “Programming a Computer for Playing Chess”, *Philosophical Magazine*
- Minimax
- Static evaluation:
 - Includes: material; pawn formation; position; commitments; mobility



 $\begin{matrix} \text{Max } M_{ijkl}, & \text{Min } M_{ijk}, & f(M_{ijkl}), & M_{ij}, & M_i, & P \\ M_i, & M_{ij}, & M_{ijk}, & M_{ijkl}, & \dots, & (1) \end{matrix}$





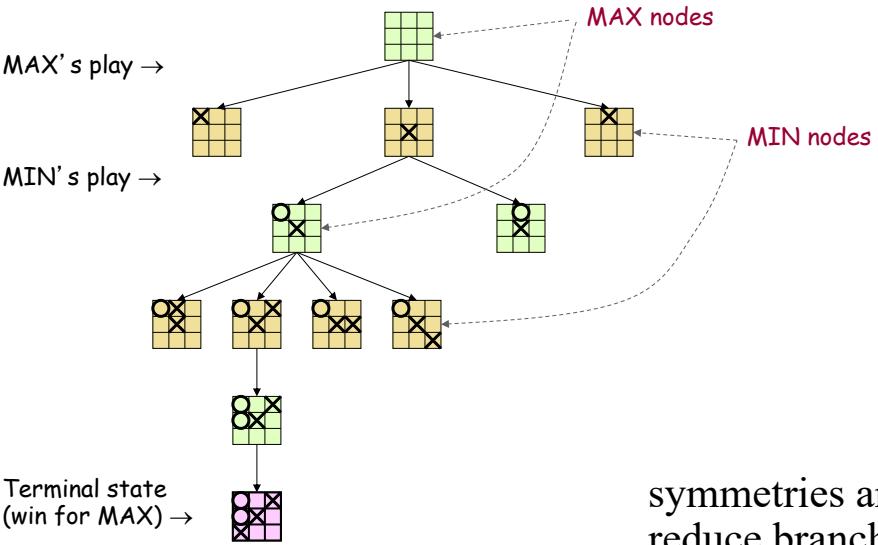
BLACK							
70	71	72	73	74	75	76	77
60	61	62	63	64	65	66	67
50	51	52	53	54	55	56	57
40	41	42	43	44	45	46	47
30	31	32	33	34	35	36	37
20	21	22	23	24	25	26	27
10	11	12	13	14	15	16	17
00	01	02	03	04	05	06	07

W
B

squillero@polito.it

20

Game Tree (Tic-Tac-Toe)



MAX's play →
 MIN's play →
 Terminal state (win for MAX) →

symmetries are used to reduce branching factor

21

1950: Theory

- Type A: **Brute-force** search, every variation to a given depth
- Type B: **Selective** search, “important” branches only

party. This is conservative for our calculation since the machine should calculate out to checkmate, not resignation. However, even at this figure there will be 10^{120} variations to be calculated from the initial position. A machine operating at the rate of one variation per micro-microsecond would require over 10^{90} years to calculate its first move!

squillero@polito.it

Chess & AI

22

22

Solving Chess

- Shannon estimated 10^{120} possible variations
- Atoms in the universe: 10^{80}
- Seconds in 1 year: $\pi 10^7$
- Nanoseconds in 1 second: 10^9
- Years since the big bang: 10^{10}
- If all atoms in the universe had been doing static evaluation at nanosecond speed since the big bang: 10^{106}



squillero@polito.it

Chess & AI

29

29



30

Solving Checkers

- J. Schaeffer, N. Burch, Y. Björnsson, A. Kishimoto, M. Müller, R. Lake, P. Lu, S. Sutphen, "Checkers Is Solved", 2007
- Only 500,995,484,682,338,672,639 positions

The game of checkers has roughly 500 billion billion possible positions (5×10^{20}). The task of solving the game, determining the final result in a game with no mistakes made by either player, is daunting. Since 1989, almost continuously, dozens of computers have been working on solving checkers, applying state-of-the-art artificial intelligence techniques to the proving process. This paper announces that checkers is now solved: Perfect play by both sides leads to a draw. This is the most challenging popular game to be solved to date, roughly one million times as complex as Connect Four. Artificial intelligence technology has been used to generate strong heuristic-based game-playing programs, such as Deep Blue for chess. Solving a game takes this to the next level by replacing the heuristics with perfection.

Pieces	Number of positions
1	120
2	6,972
3	261,224
4	7,092,774
5	148,688,232
6	2,921,648,48
7	34,779,531,480
8	406,309,208,401
9	4,048,627,642,976
10	34,778,882,769,216
Total 1–10	39,271,258,813,439
11	259,669,578,902,016
12	1,695,618,000,654,976
13	97,974,638,839,356
14	49,134,211,067,979,776
15	238,511,510,918,189,056
16	852,888,183,557,922,816
17	2,905,162,728,973,680,640
18	8,568,043,414,939,516,928
19	21,661,954,506,100,113,408
20	46,352,957,062,510,379,000
21	82,405,638,813,062,389,29
22	110,435,747,234,017,856,512
23	129,406,908,049,181,900,800
24	90,072,726,844,888,186,880
Total 1–24	500,995,484,682,338,672,639

c

31

1959–1962: The Kotok-McCarthy Program

- A.k.a., “A Chess Playing Program for the IBM 7090 Computer”
- Shannon Type-B (selective search)
- Alpha-beta pruning



squillero@polito.it

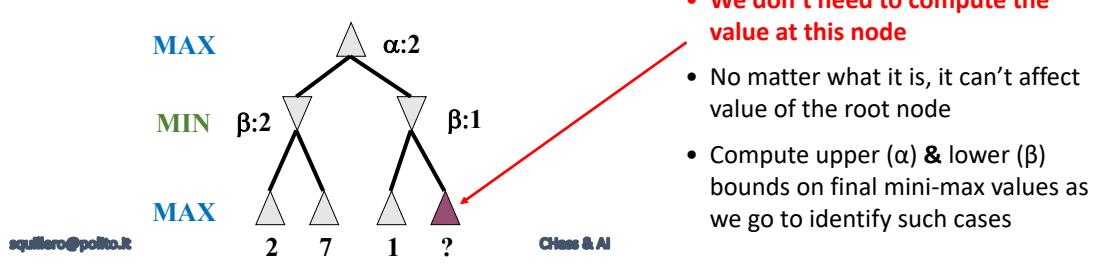
Chess & AI

32

32

Alpha-beta Pruning

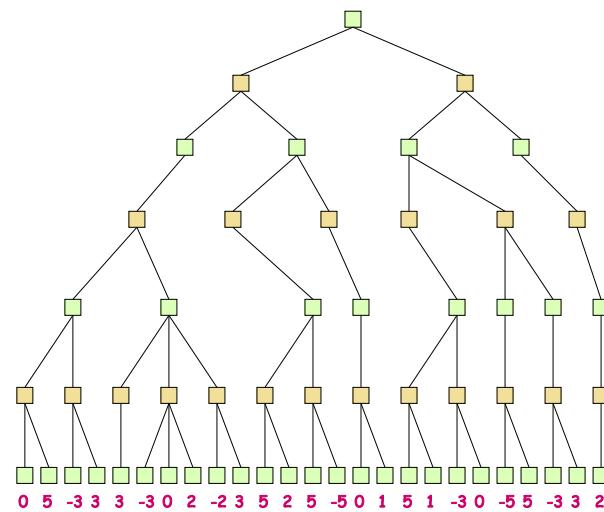
- “If you have an idea that is surely bad, don’t take the time to see how truly awful it is” (Pat Winston, MIT)
- Reinvented a number of times by Richards, Hart, Levin, Edward, Newell, Simon, McCarty, Kotok, Samuel...
- Refined by Knuth and Moore



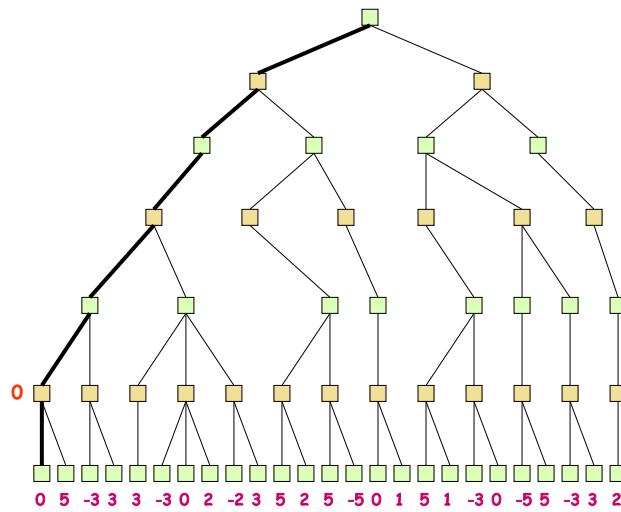
33

33

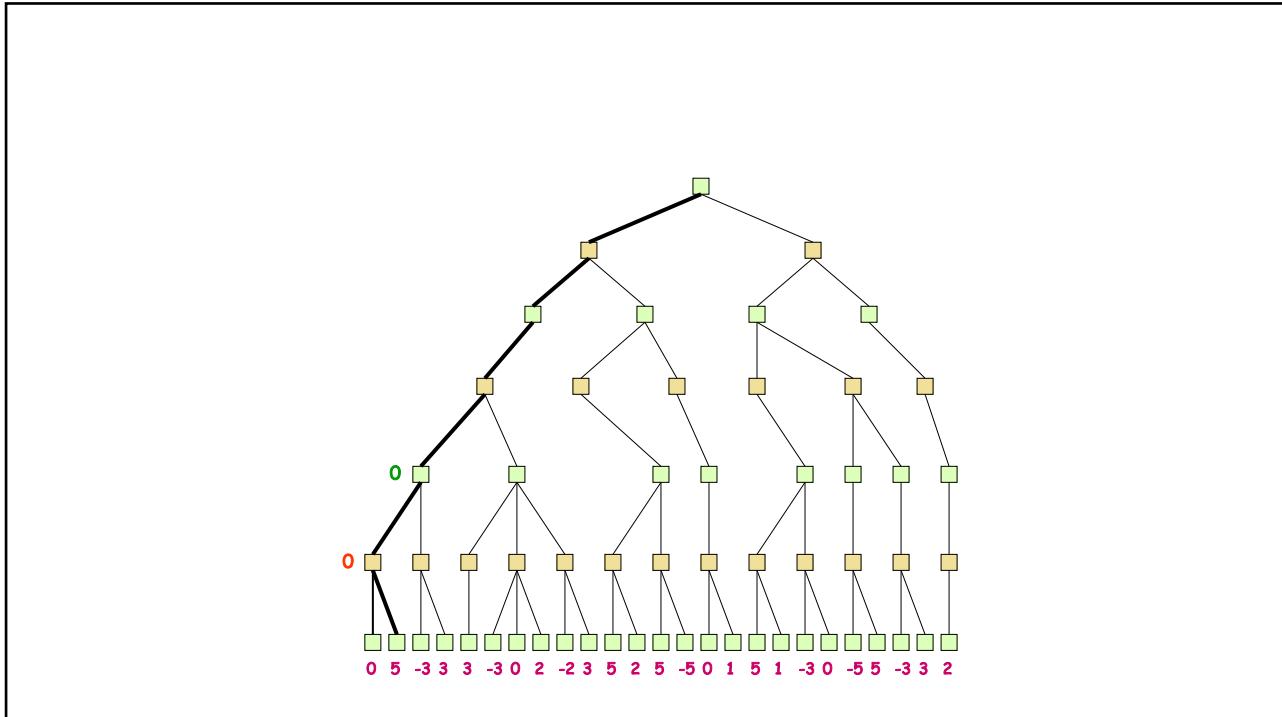
Alpha-beta Pruning



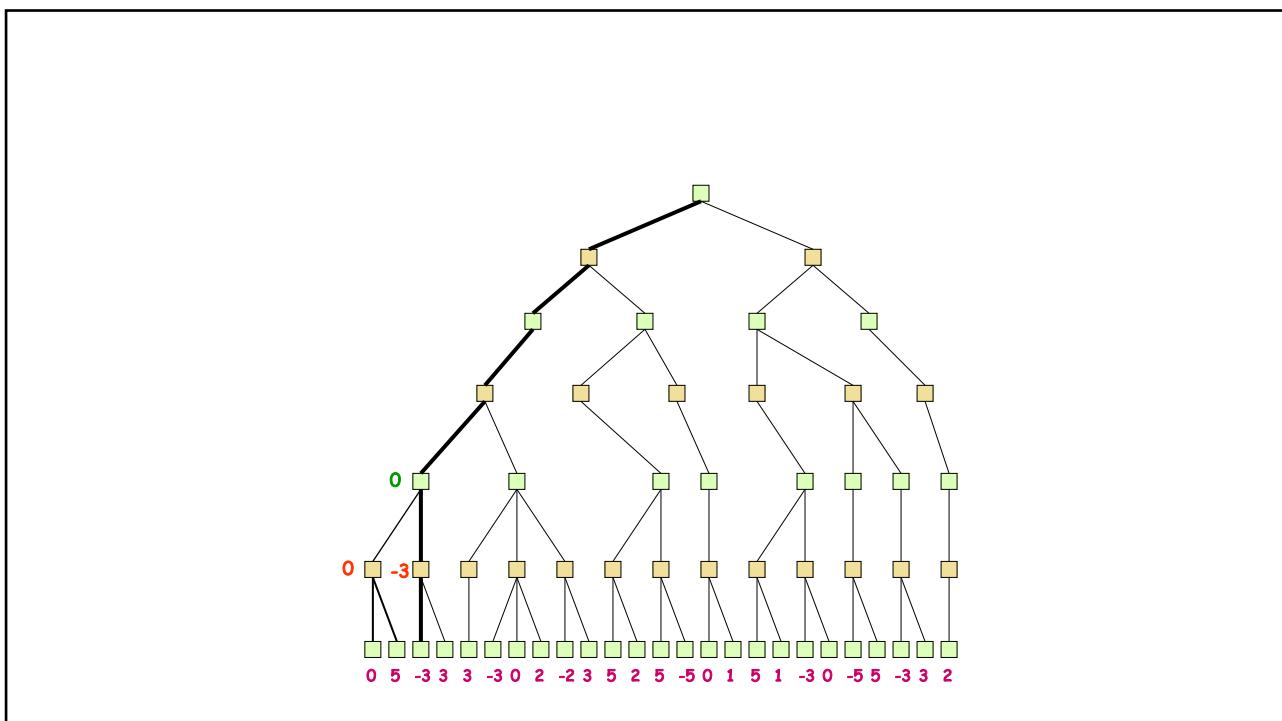
34



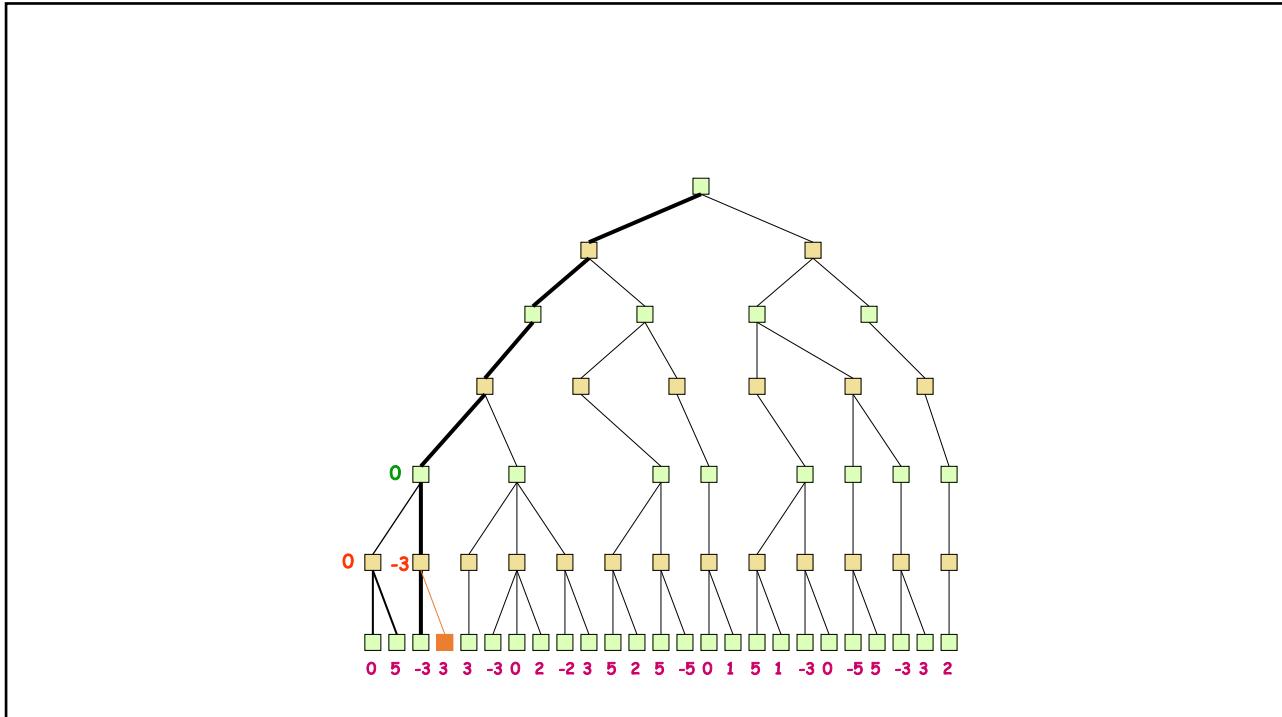
35



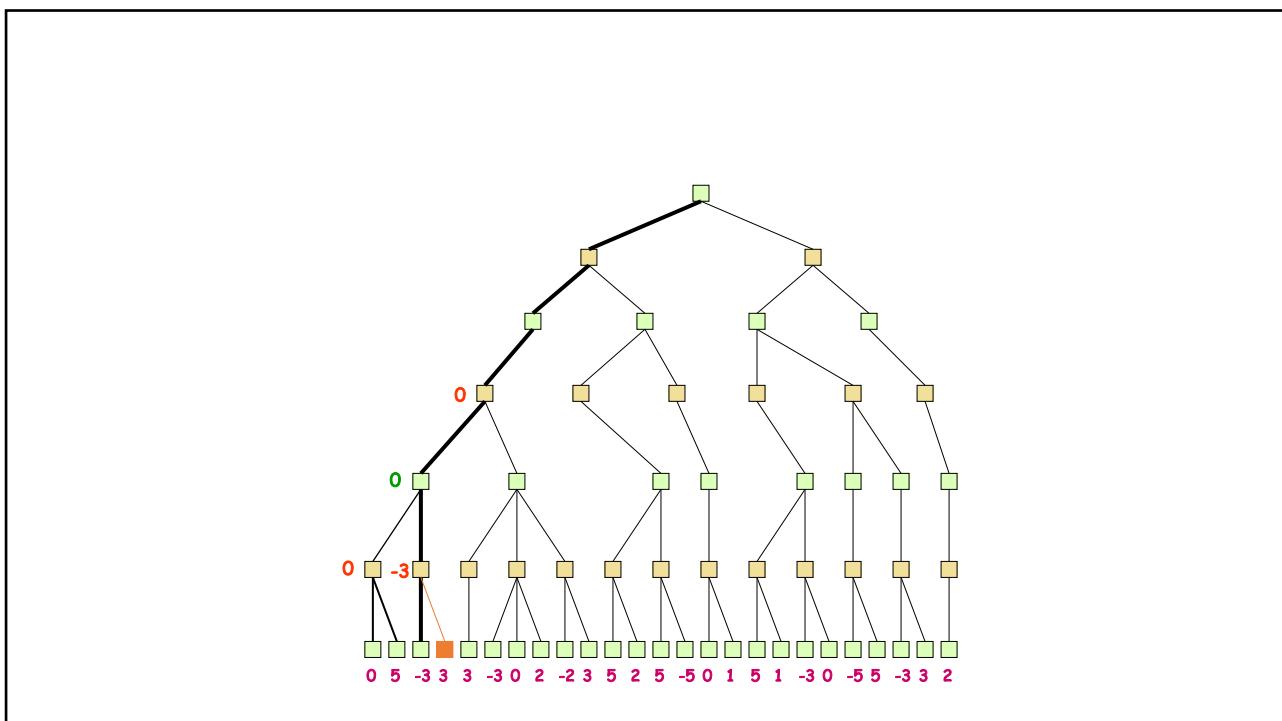
36



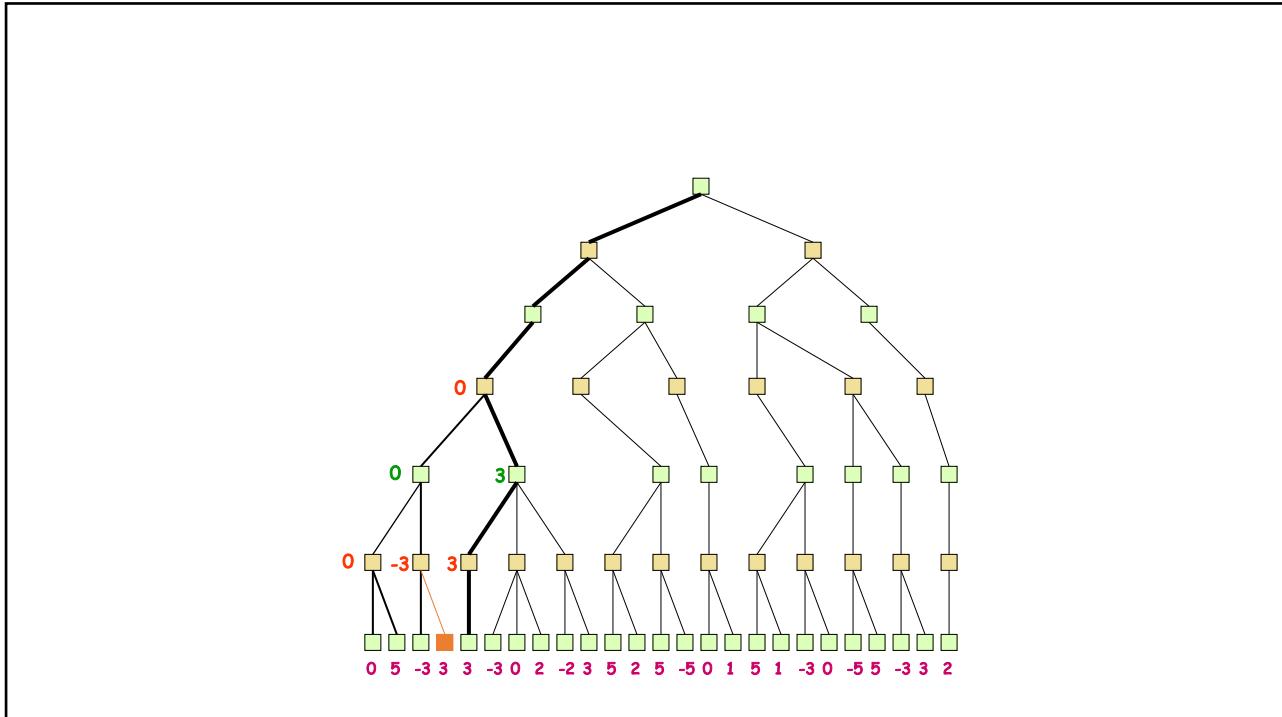
37



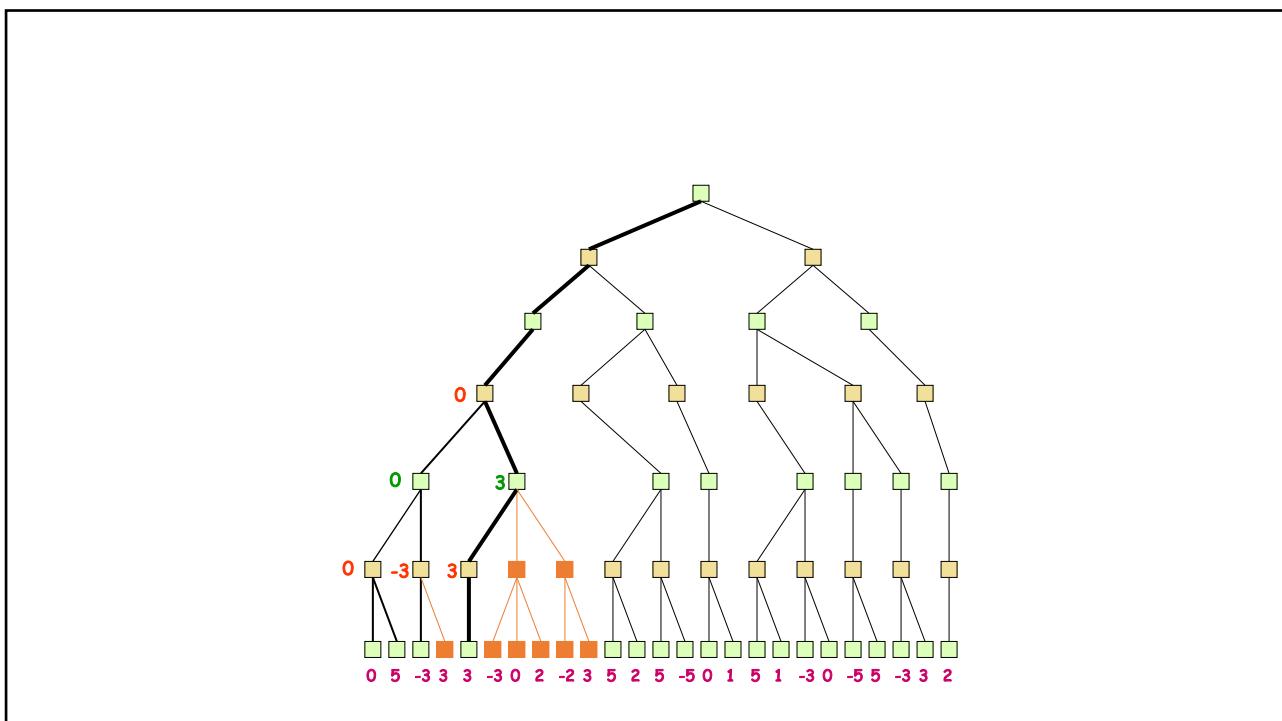
38



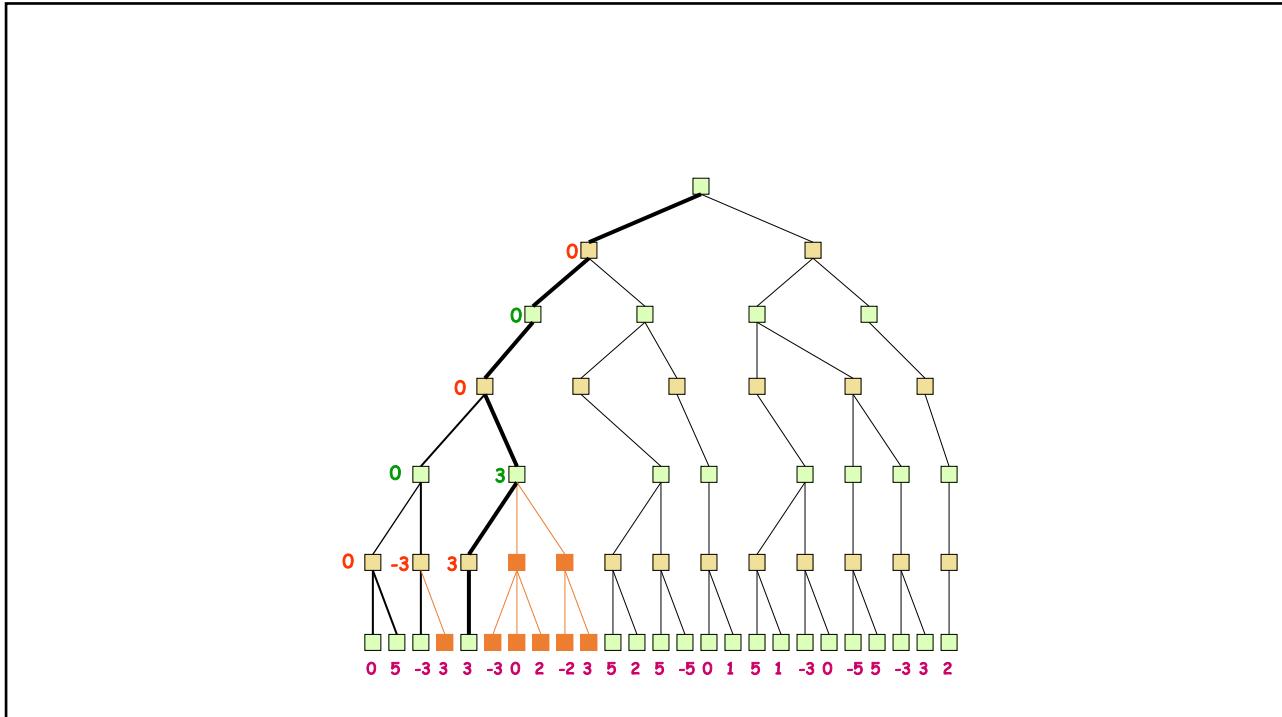
39



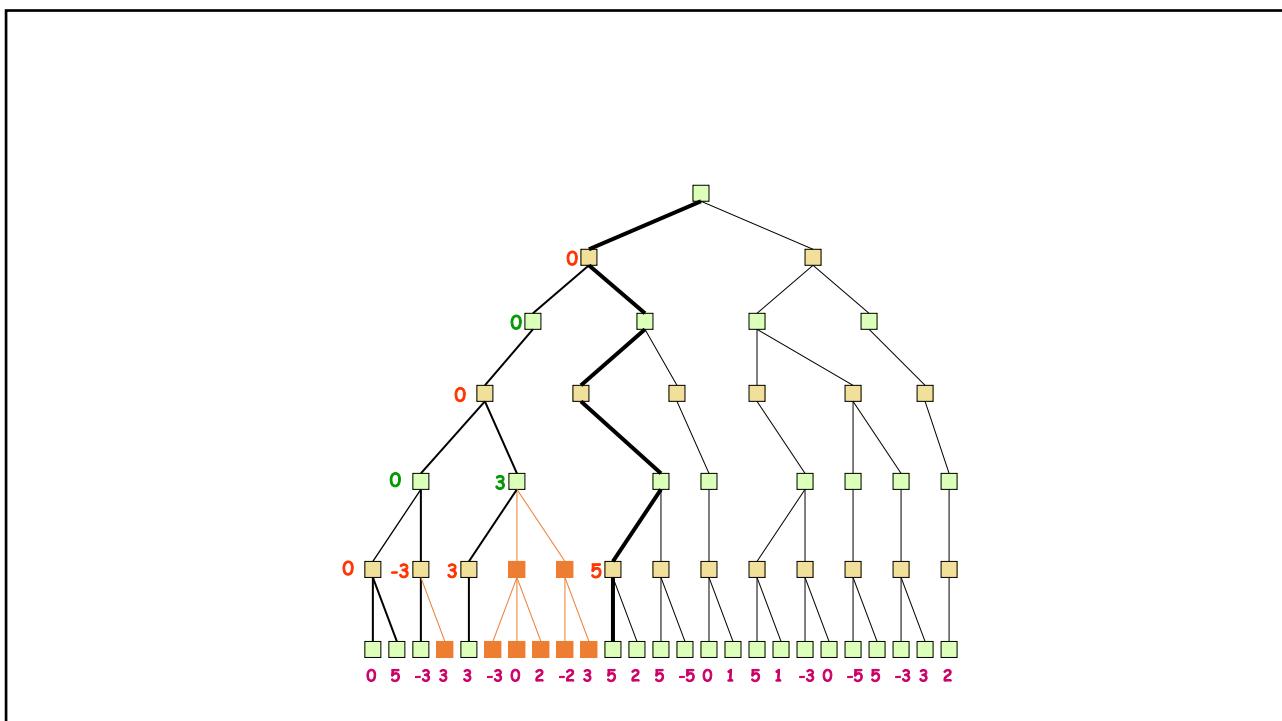
40



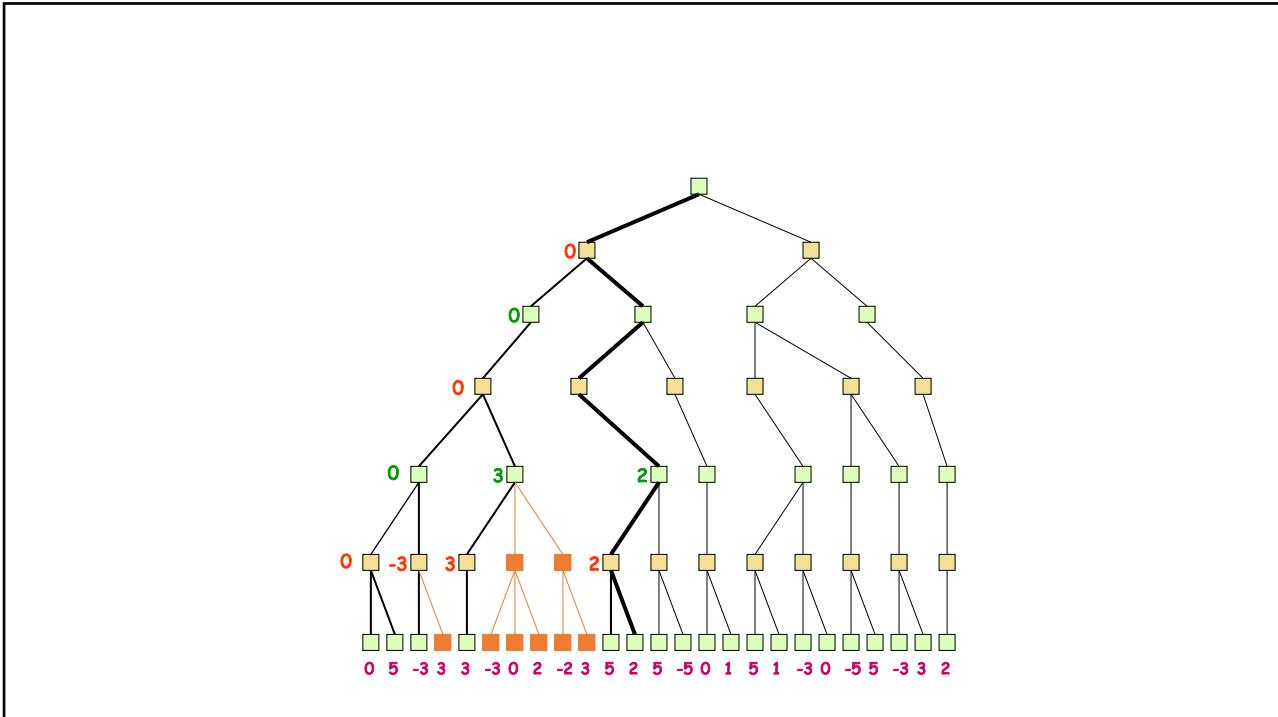
41



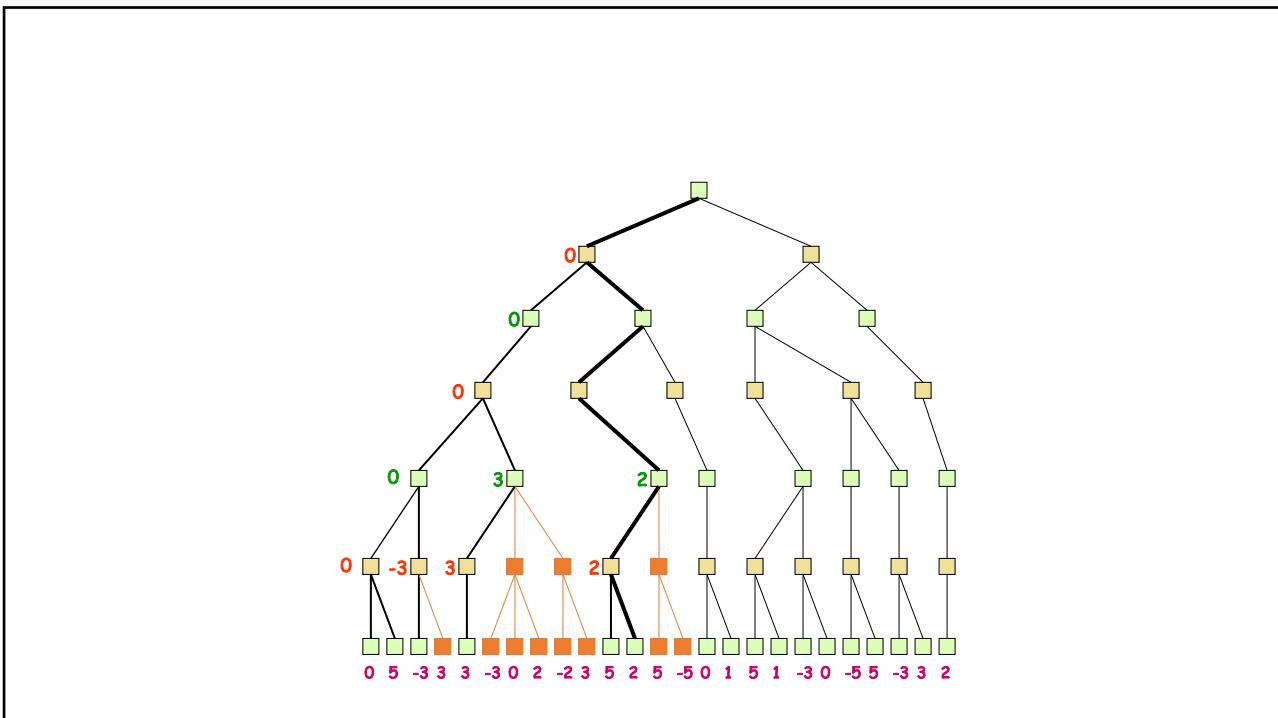
42



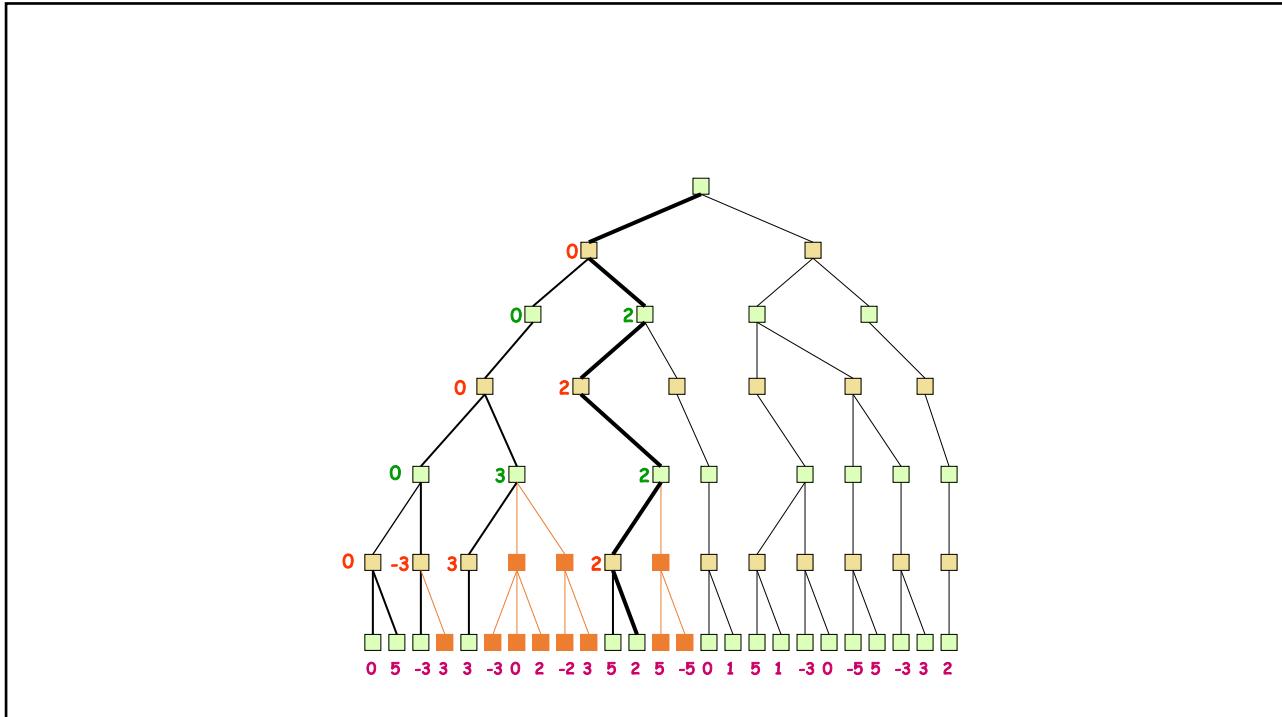
43



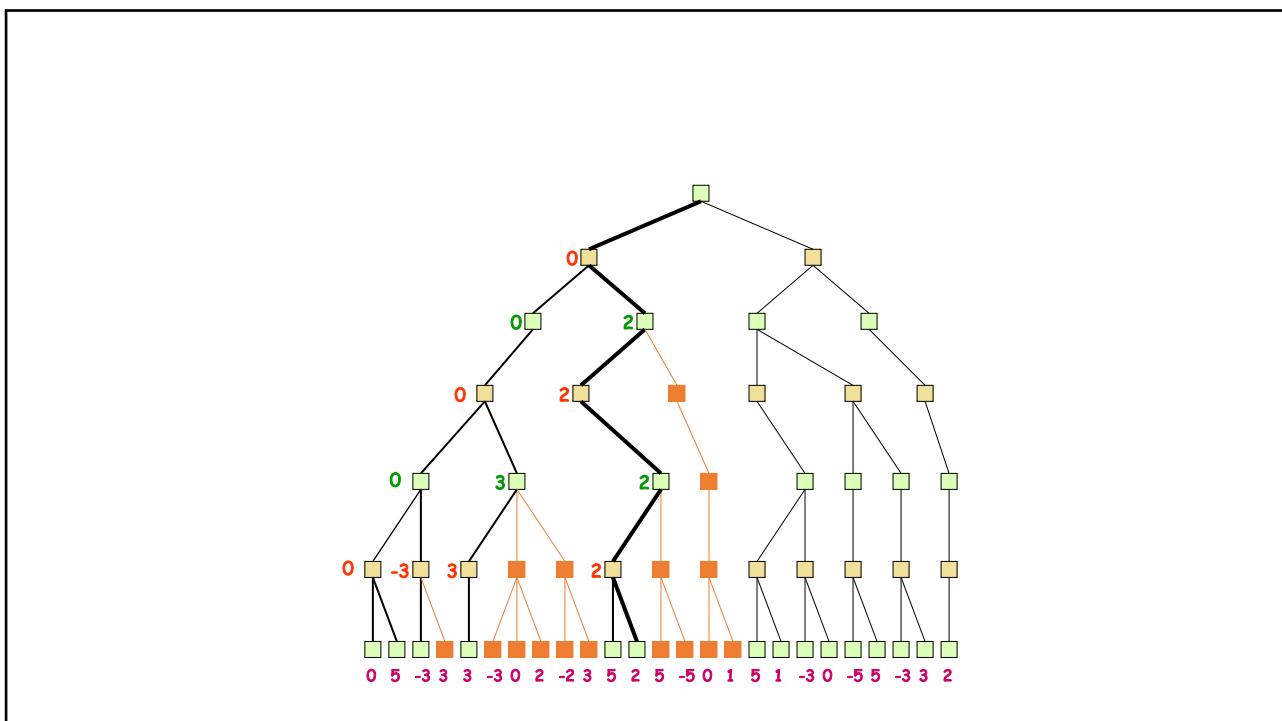
44



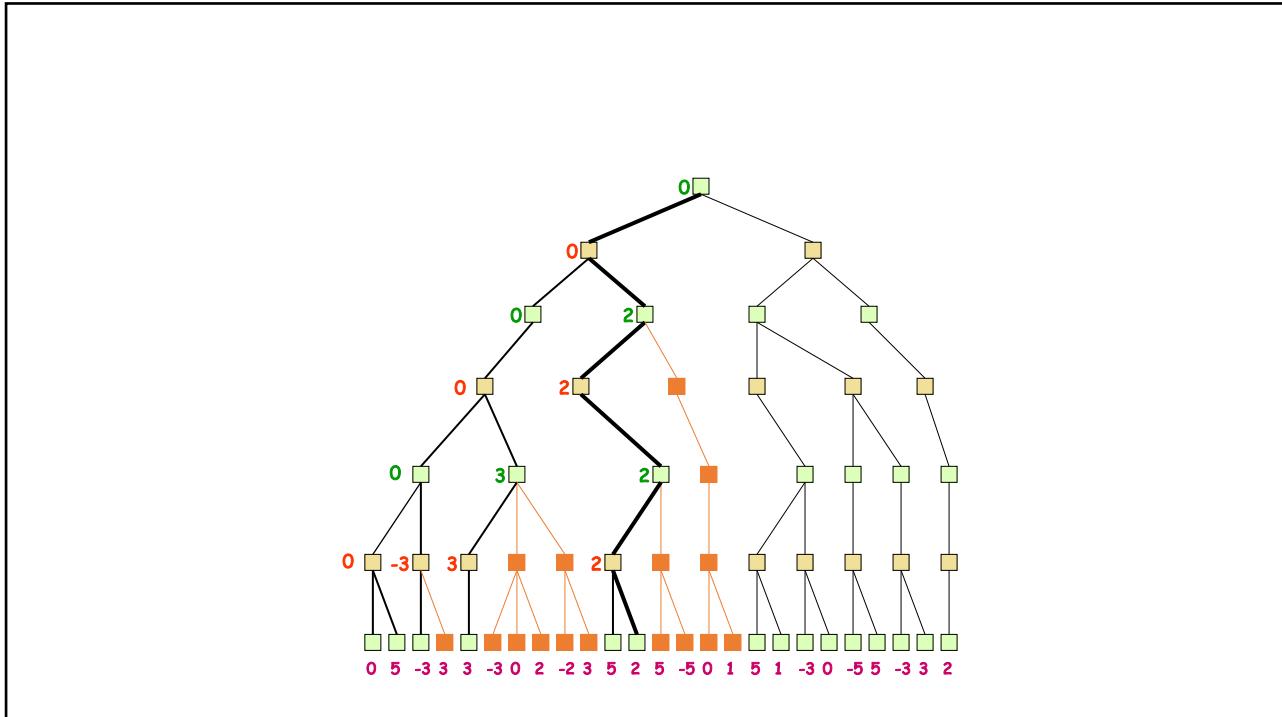
45



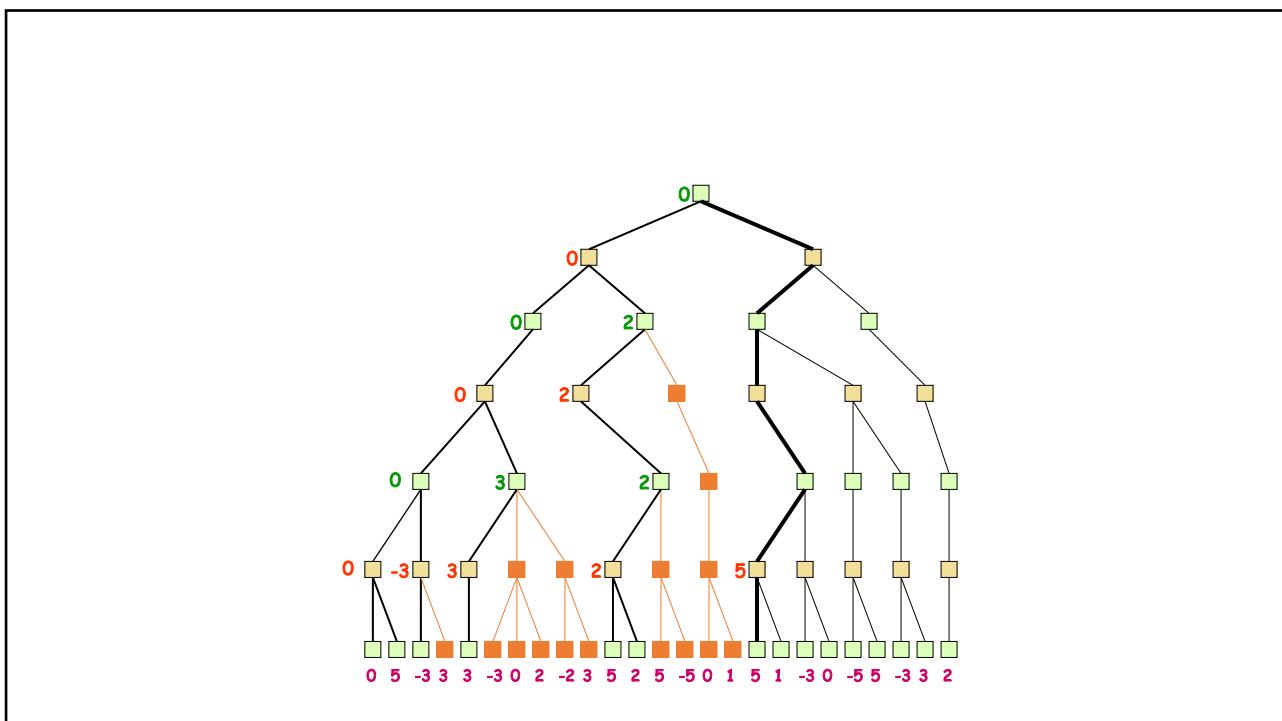
46



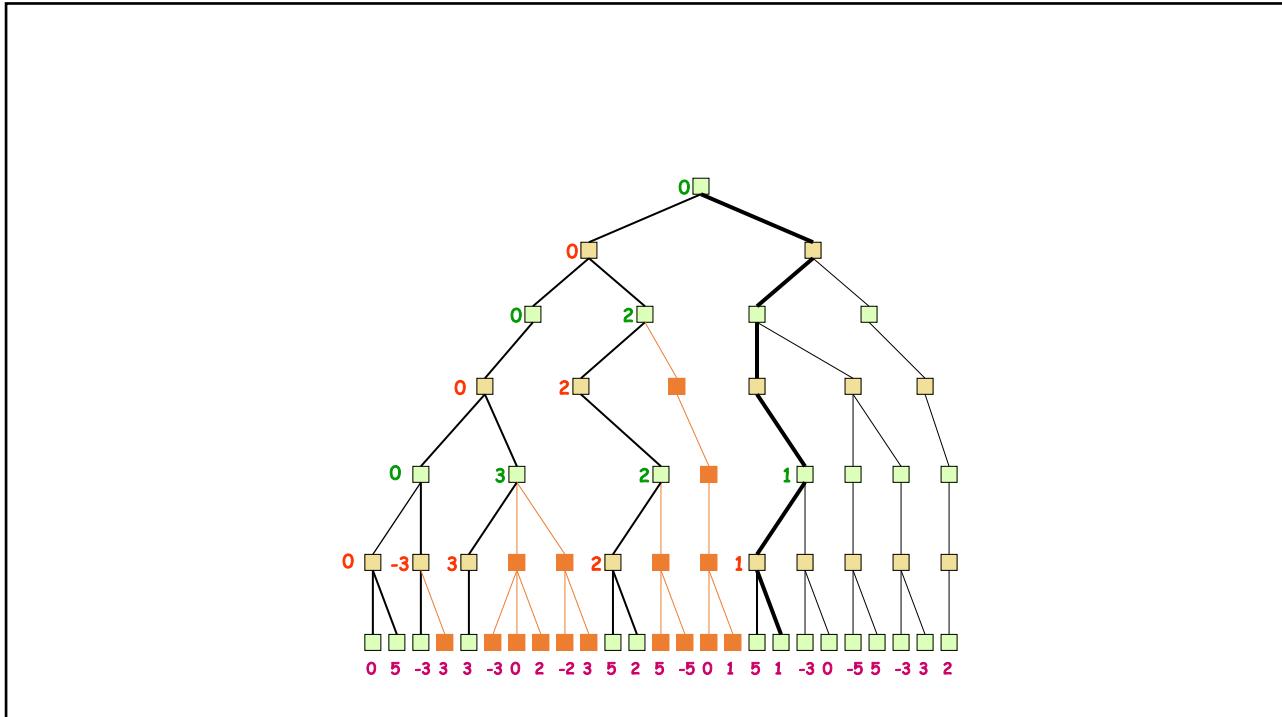
47



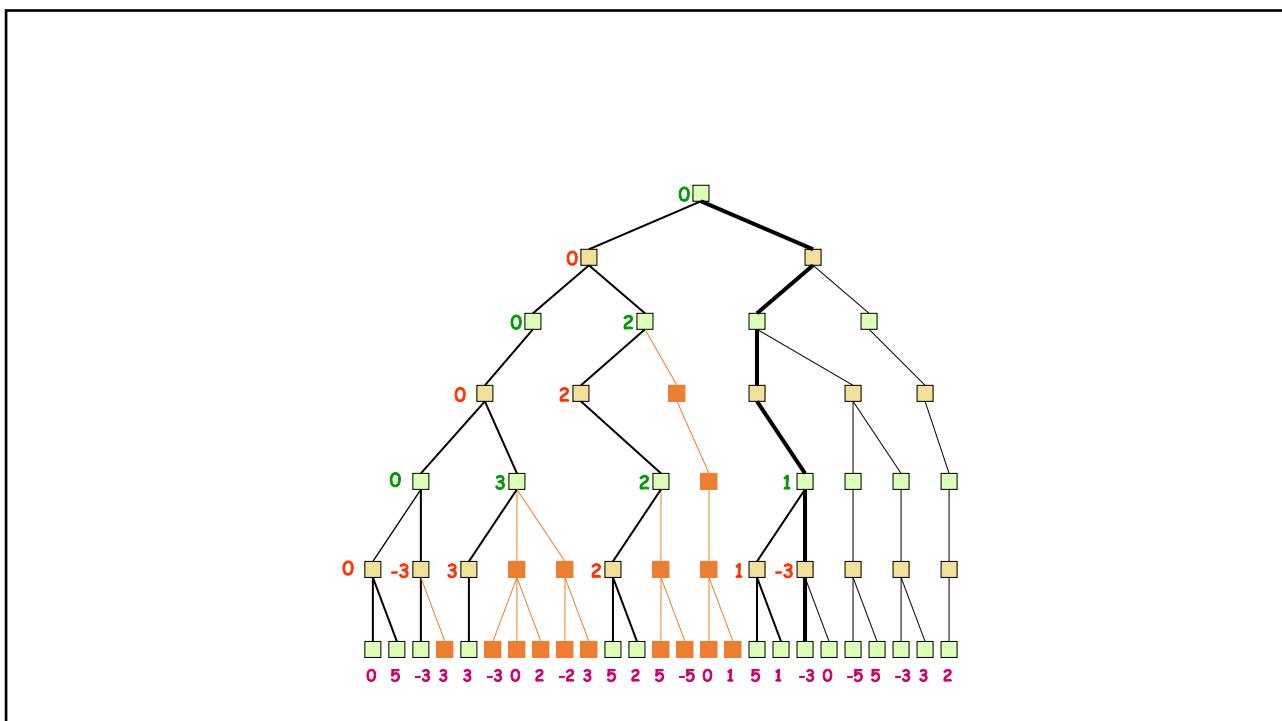
48



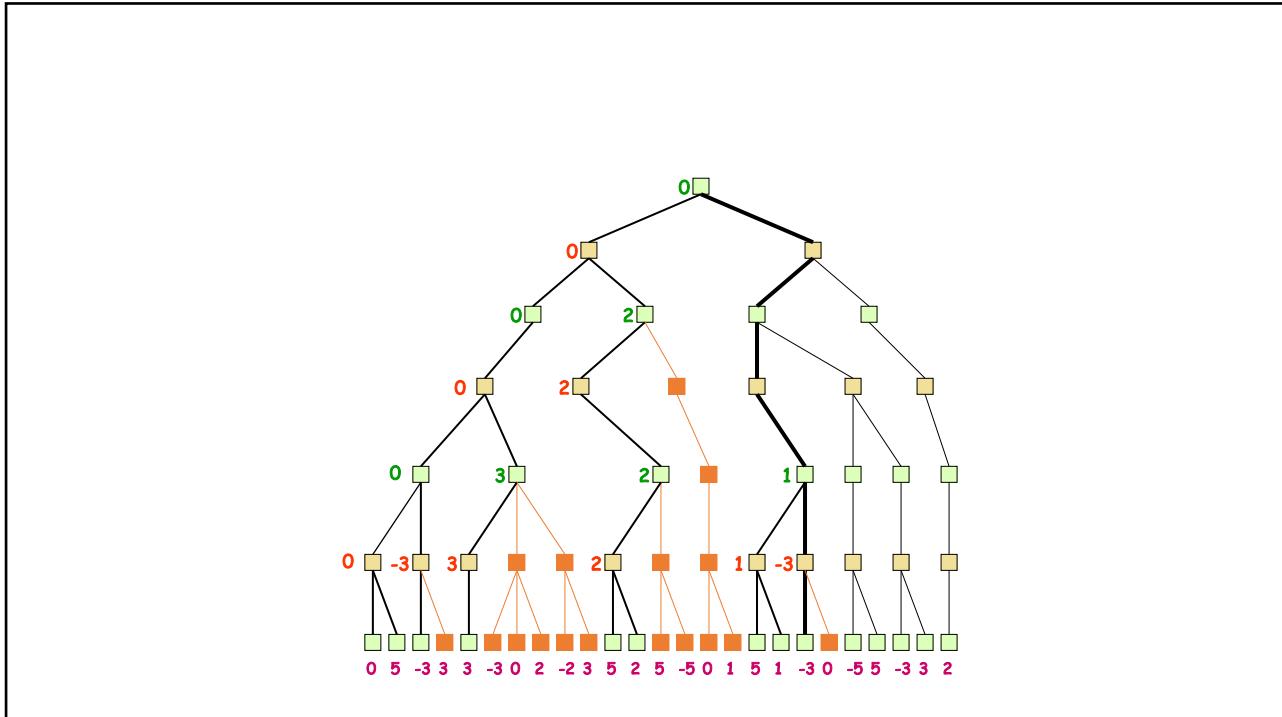
49



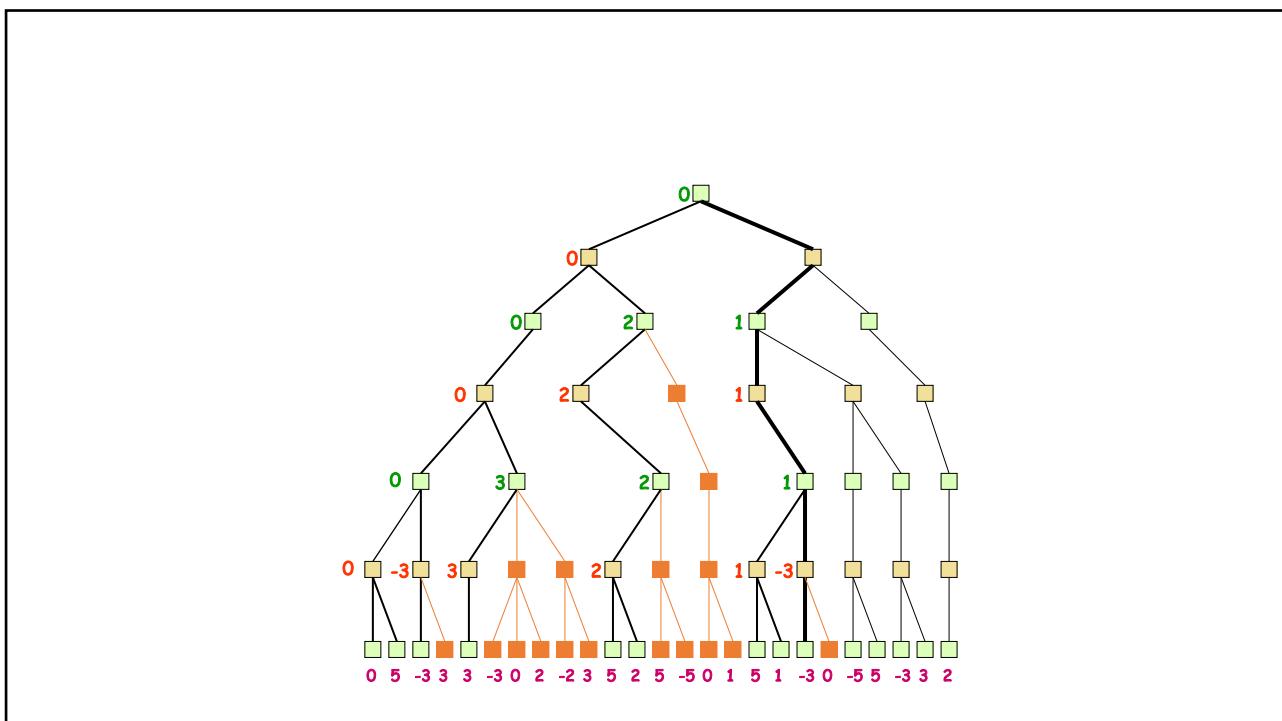
50



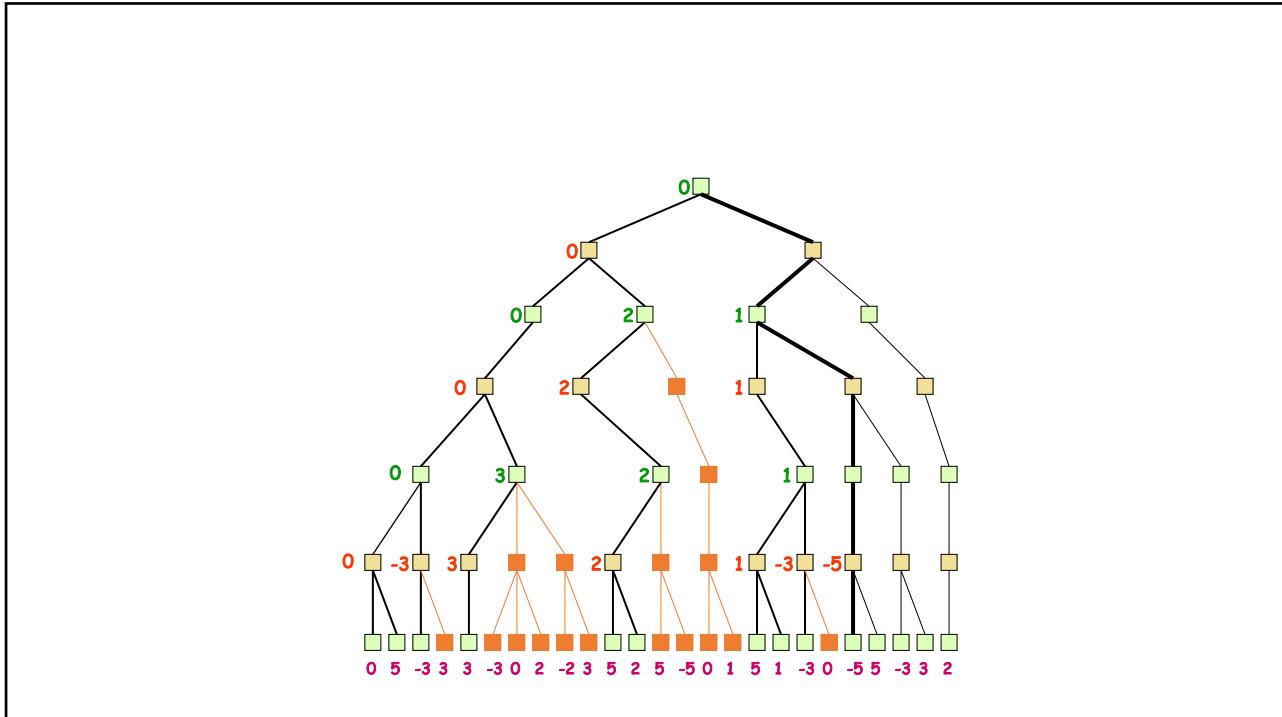
51



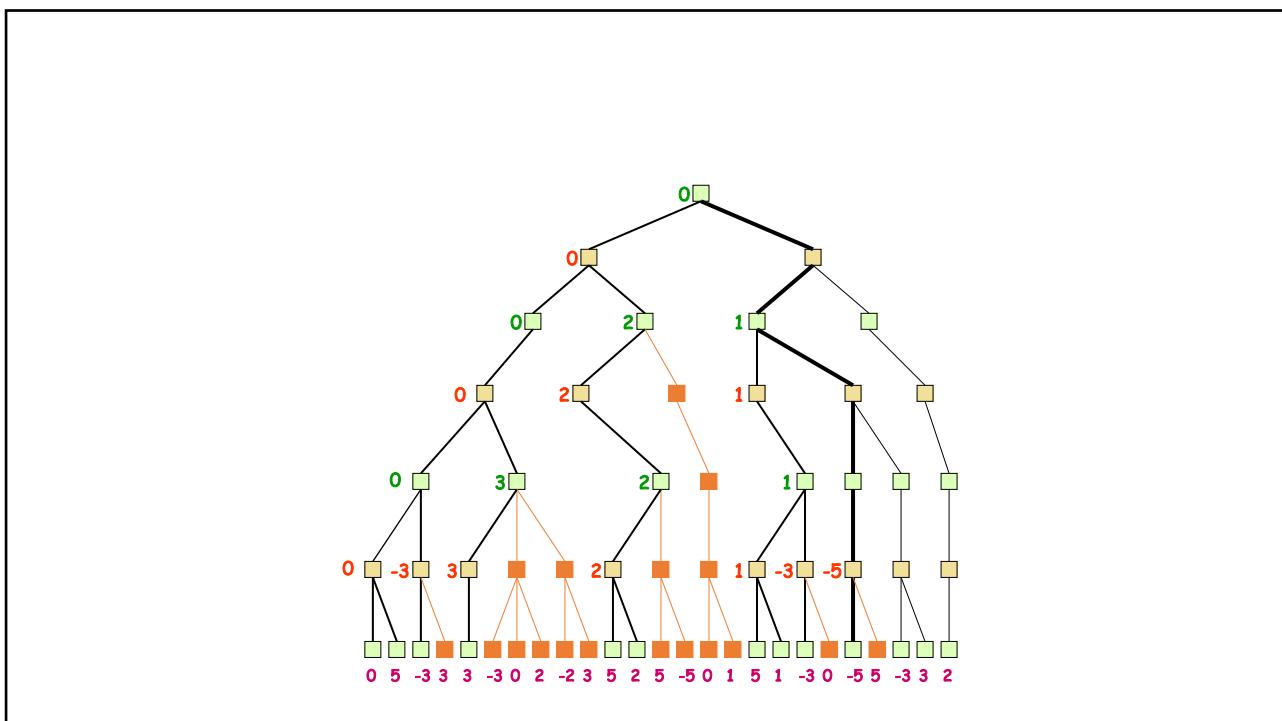
52



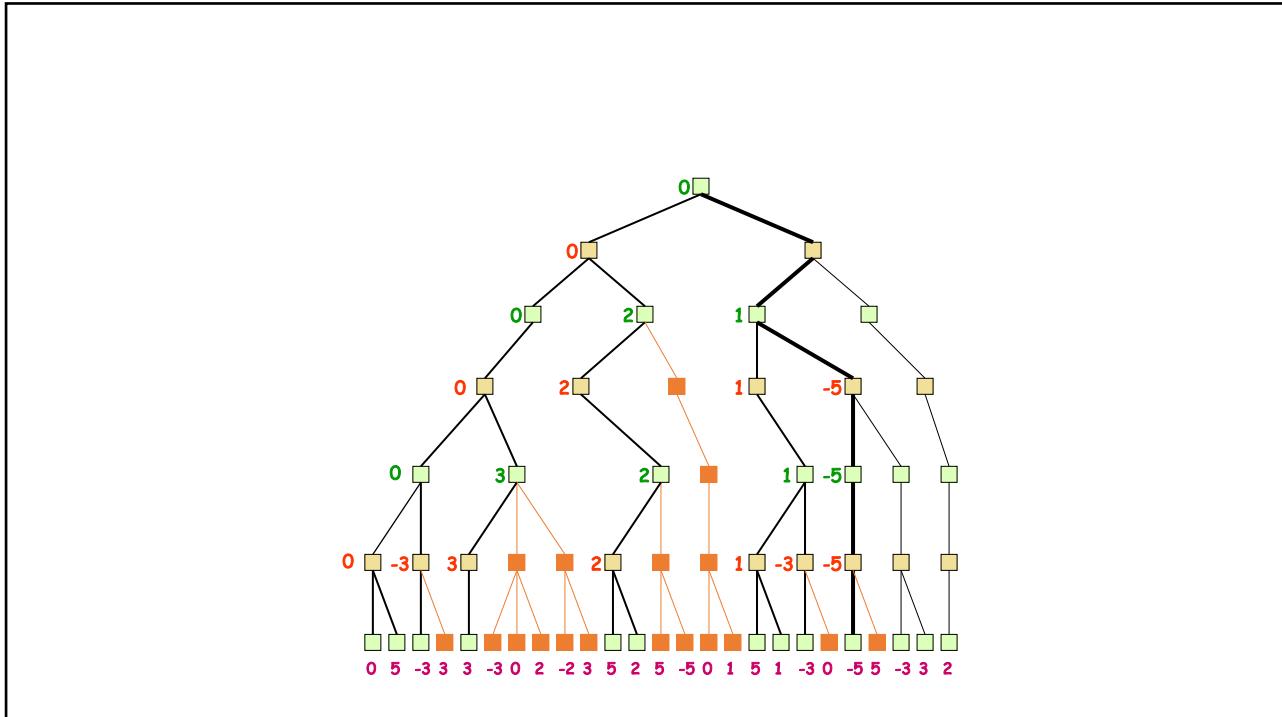
53



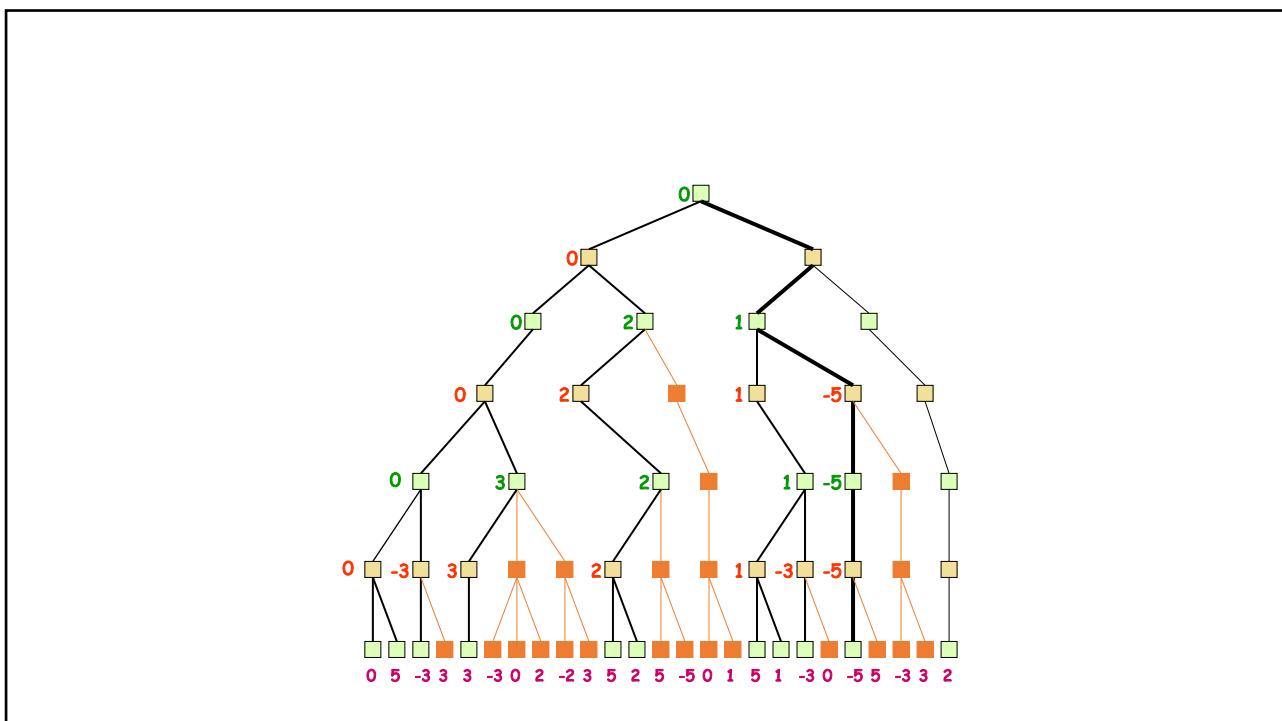
54



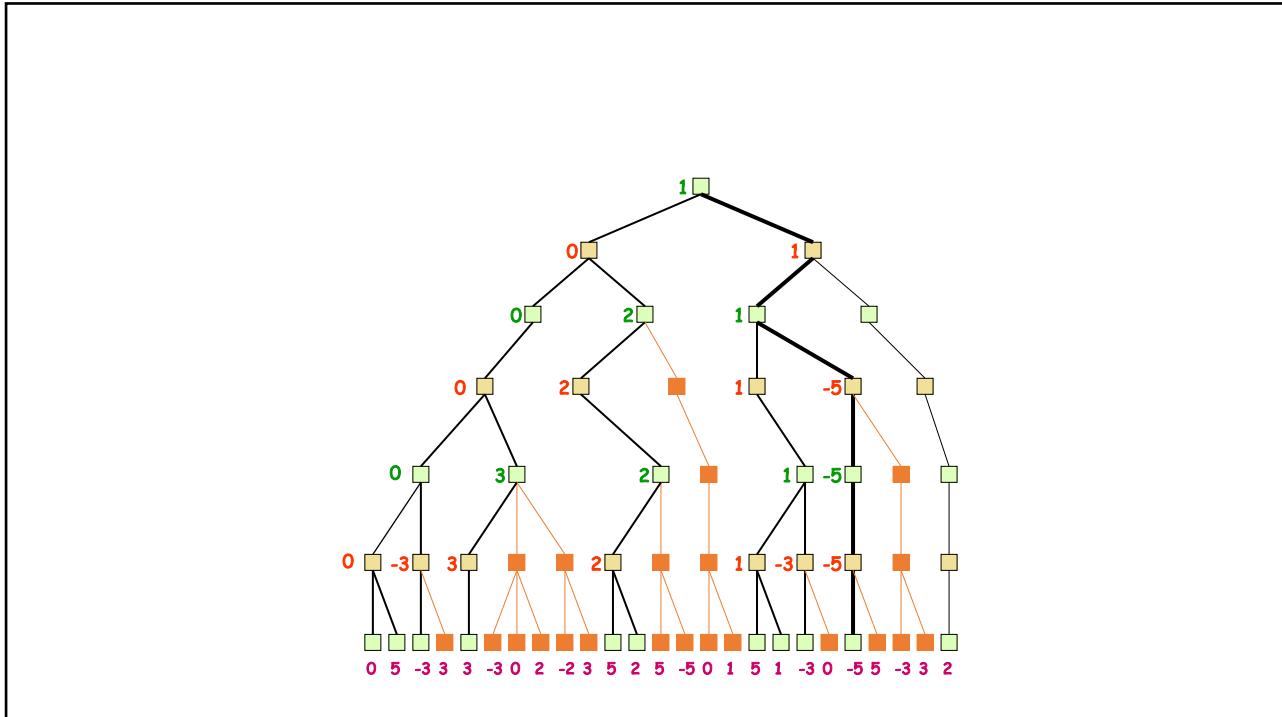
55



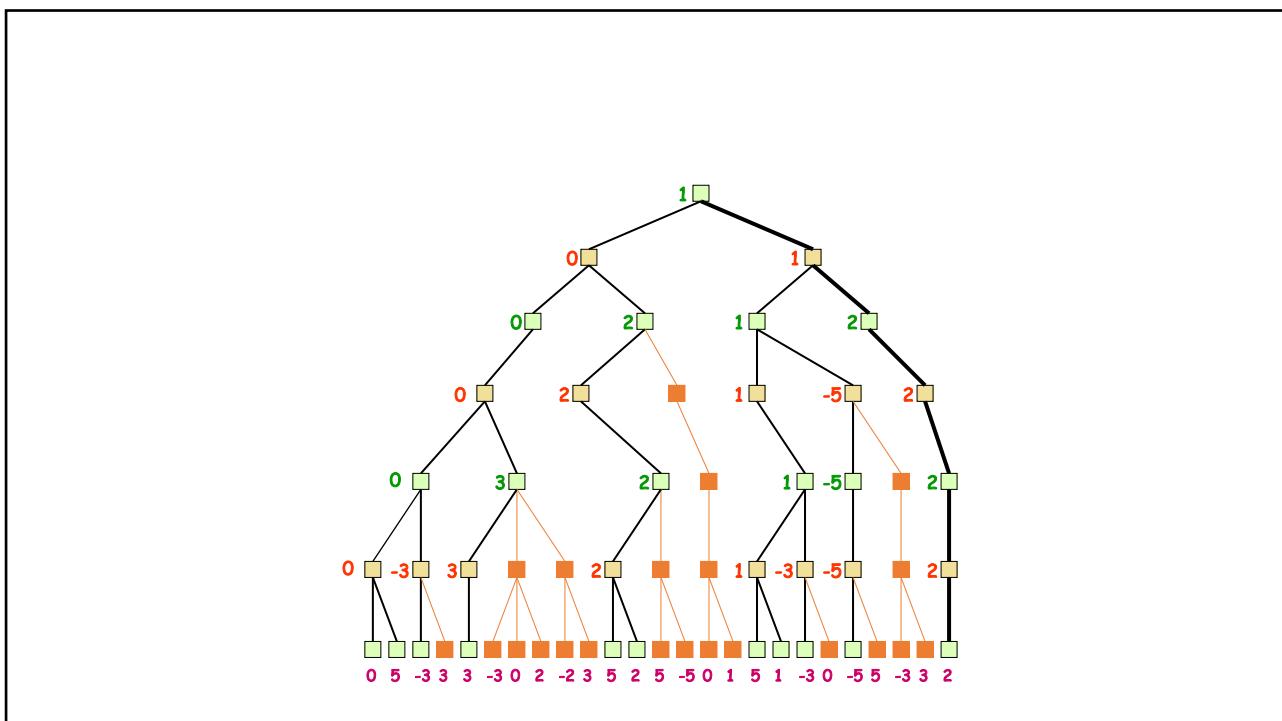
56



57

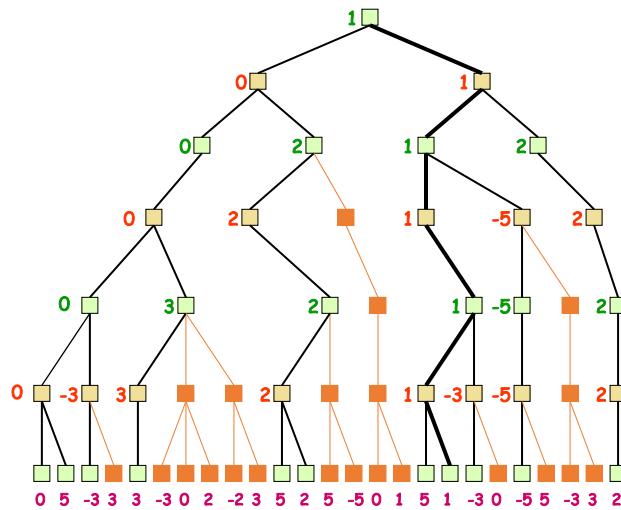


58



59

With alpha-beta we avoided computing a static evaluation metric for 14 of the 25 leaf nodes



60

The Levy—McCarthy Bet

- In 1969 chess master David Levy bet \$1000 with John McCarthy that he could defeat any chess playing program in the next 10 years
- Levy won the bet — in 1979 he defeated the best chess programs of the day



squillero@polito.it

Chess & AI

61

The Levy—McCarthy Bet (again)

- In 1969 Levy extended his bet for the next 10 years
- He eventually lost the bet to Deep Thought

Feng-hsiung Hsu
許峰雄
C.B. (Crazy Bird)
squillero@polito.it



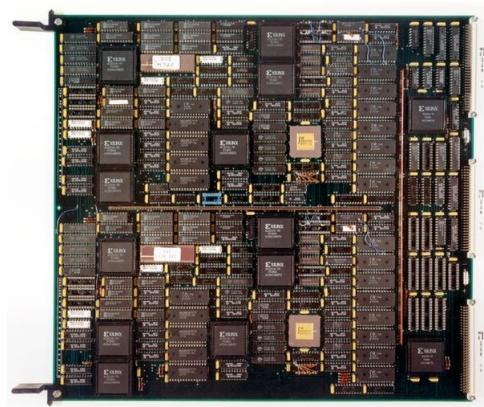
62

Deep Thought

- Feng-hsiung Hsu's special-purpose chess chip
 - Evaluate 500M position/s
 - Horizon: 10-11 moves + “extensions”
 - Once found checkmate in 35 moves!
- First machine to beat a grandmaster in a tournament

squillero@polito.it

Chess & AI



63

Deep Thought (1989)

- Lost a 2-game match to then chess world champion Garry Kasparov



squillero@polito.it

Chess & AI

64

64

Suggested Readings

Douglas Adams

*Guida galattica per gli autostoppisti
(Il ciclo completo e un racconto inedito)*

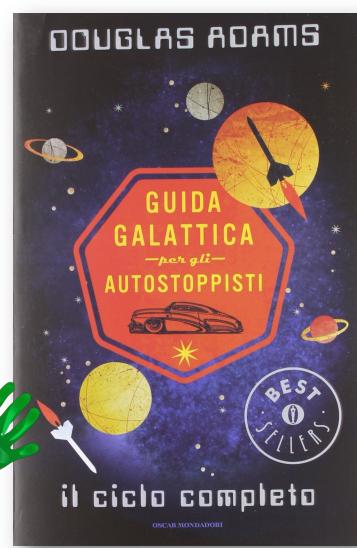
Mondadori (1999)



squillero@polito.it



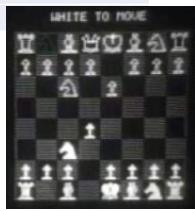
Chess & AI



65

List of Early Computer/Programs

Name	Author(s)	Year	Features
The Bernstein Chess Program	Alex Bernstein et al.	1958	Shannon Type B
NSS Chess Program	Allen Newell, Cliff Shaw, Herbert Simon	1958	Shannon Type B, Alpha-Beta approx.
Kotok-McCarthy-Program	Alan Kotok, John McCarthy et al.	1962	Shannon Type B, Alpha-Beta
ITEP Chess Program	Georgy Adelson-Velsky et al.	1963	Shannon Type A, Alpha-Beta
The Greenblatt Chess Program	Richard Greenblatt et al.	1967	Shannon Type B, Transposition Table, Opening Book
The Technology Chess Program	James Gillogly et al.	1970	Shannon Type A, Pondering
Kaissa	Mikhail Donskoy et al.	1970	Shannon Type A, Bitboards, Null Move Pruning
Chess 4.0	David Slate, Larry Atkin	1973	Shannon Type A, Bitboards
Belle	Ken Thompson, Joe Condon	1978	Shannon Type A, Chess Hardware
Cray Blitz	Robert Hyatt, Albert Gower, Harry Nelson	1983	Shannon Type A, Dynamic Tree Splitting
HiTech	Hans Berliner et al.	1985	Alpha-Beta or B*, Chess Hardware
Deep Thought	Feng-hsiung Hsu et al.	1986	Shannon Type A, Chess Hardware



66

Deep Blue (February 1996)



squillero@polito.it

Chess & AI

67

67

Deep Blue (February 1996)

- Six-game chess match, Kasparov won by 4-2
- **February 10th 1996:** The first time that a chess-playing computer defeated a **reigning** world champion in standard chess tournament settings (40 moves in 2h)
- The real **turning point** (a milestone in the history of artificial intelligence)

squillero@polito.it

Chess & AI

68

68

Deep Blue (February 1996)

Game #	White	Black	Result	Method of conclusion
1	Deep Blue	Kasparov	1–0	Resignation
2	Kasparov	Deep Blue	1–0	Resignation
3	Deep Blue	Kasparov	½–½	Draw by mutual agreement
4	Kasparov	Deep Blue	½–½	Draw by mutual agreement
5	Deep Blue	Kasparov	0–1	Resignation
6	Kasparov	Deep Blue	1–0	Resignation
Result: Kasparov–Deep Blue: 4–2				

squillero@polito.it

Chess & AI

69

69



70

Deep Blue (1997)

Game #	White	Black	Result	Method of conclusion
1	Kasparov	Deep Blue	1–0	Resignation
2	Deep Blue	Kasparov	1–0	Resignation
3	Kasparov	Deep Blue	½–½	Draw by mutual agreement
4	Deep Blue	Kasparov	½–½	Draw by mutual agreement
5	Kasparov	Deep Blue	½–½	Draw by mutual agreement
6	Deep Blue	Kasparov	1–0	Resignation

Result: Deep Blue–Kasparov: 3½–2½

Deep Blue



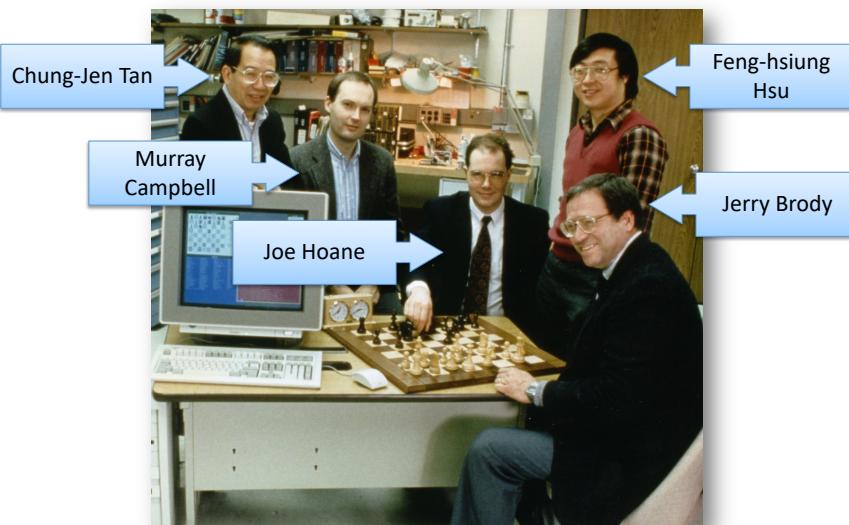
squillero@polito.it

Chess & AI

72

72

Deep Blue Team



squillero@polito.it

Chess & AI

73

73



74

Suggested Readings

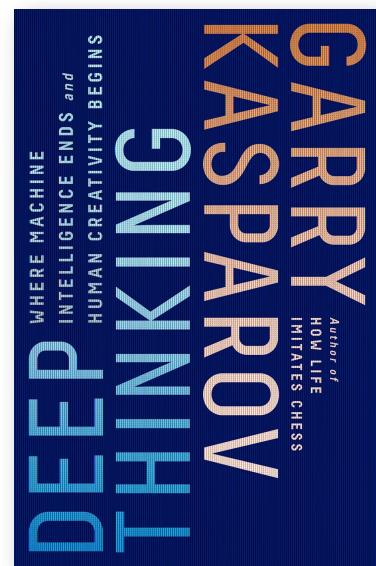
Gary Kasparov

*Deep Thinking: Where Machine
Intelligence Ends and Human
Creativity Begins*

PublicAffairs (2017)

squillero@polito.it

Chess & AI



75

Deep Blue

- Minimax with alpha-beta pruning
- Custom hardware
 - IBM RS/6000 SP Supercomputer (30 PowerPC 604e)
 - 480 custom 600 μ m CMOS VLSI chess chips and “some” FPGAs
- Static evaluation (200M position/s)
 - Fast vs. Slow evaluations
 - 8,000 blocks with adapted weights
- Opening book: 4,000 positions (+ 700,000 GM games)
- Endgame database

squillero@polito.it

Chess & AI

76

76



77

Stockfish

Domain knowledge, extensions, heuristics in TCEC world champion Stockfish:

Board Representation: Bitboards with Little-Endian Rank-File Mapping (LERF), Magic Bitboards, BMI2 – PEXT Bitboards, Piece-Lists, **Search:** Iterative Deepening, Aspiration Windows, Parallel Search using Threads, YBWC, Lazy SMP, Principal Variation Search. **Transposition Table:** Shared Hash Table, Depth-preferred Replacement Strategy, No PV-Node probing, Prefetch **Move Ordering:** Countermove Heuristic, Counter Moves History, History Heuristic, Internal Iterative Deepening, Killer Heuristic, MVV/LVA, SEE, **Selectivity:** Check Extensions if SEE ≥ 0 , Restricted Singular Extensions, Futility Pruning, Move Count Based Pruning, Null Move Pruning, Dynamic Depth Reduction based on depth and value, Static Null Move Pruning, Verification search at high depths, ProbCut, SEE Pruning, Late Move Reductions, Razoring, Quiescence Search, **Evaluation:** Tapered Eval, Score Grain, Point Values Midgame: 198, 817, 836, 1270, 2521, Endgame: 258, 846, 857, 1278, 2558, Bishop Pair, Imbalance Tables, Material Hash Table, Piece-Square Tables, Trapped Pieces, Rooks on (Semi) Open Files, Outposts, Pawn Hash Table, Backward Pawn, Doubled Pawn, Isolated Pawn, Phalanx, Passed Pawn, Attacking King Zone, Pawn Shelter, Pawn Storm, Square Control, Evaluation Patterns, **Endgame Tablebases:** Syzygy TableBases

squillero@polito.it

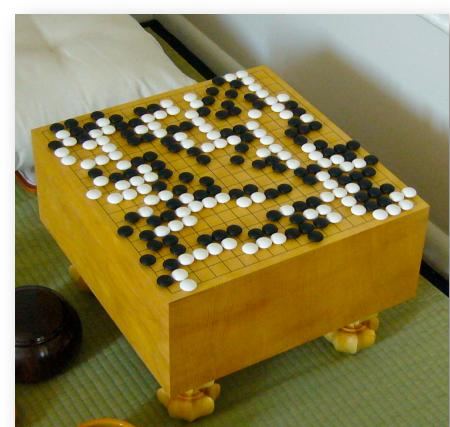
Chess & AI

78

78

Go (圍棋)

- Invented more than **2,500** years ago (played during Zhou dynasty)
- Played by alternately placing black and white stones on the vacant intersections of a **19×19** grid
- The objective is to control a larger territory than the opponent
- **≈200** possible alternatives for each move, **≈ 10^{170}** possible positions
 - ... prior to 2015, the best Go programs only managed to reach amateur dan level



79

squillero@polito.it

Chess & AI

79

Suggested Readings

Yumi Hotta, Takeshi Obata

Hikaru no Go

Weekly Shōnen Jump (1999–2003)
(23 tankōbon volumes)



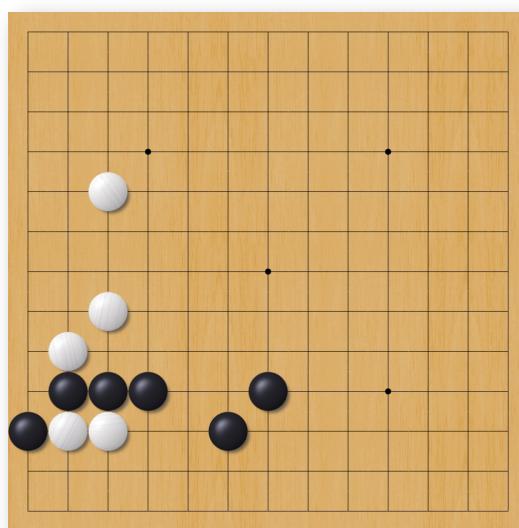
squillero@polito.it

Chess & AI

80

80

White to move



squillero@polito.it

Chess & AI

81

81

Arthur Samuel (1959)



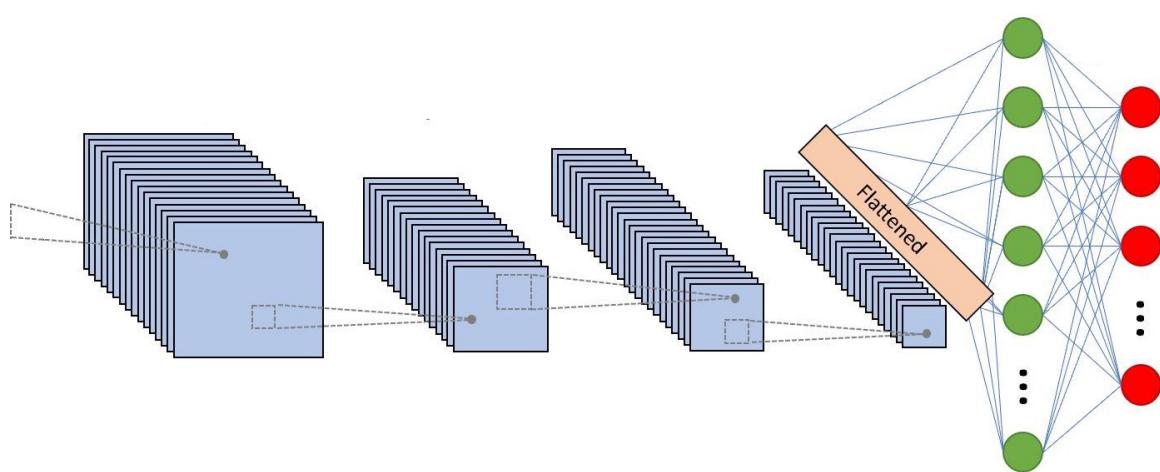
squillero@polito.it

Chess & AI

82

82

Neural Network (1943–)

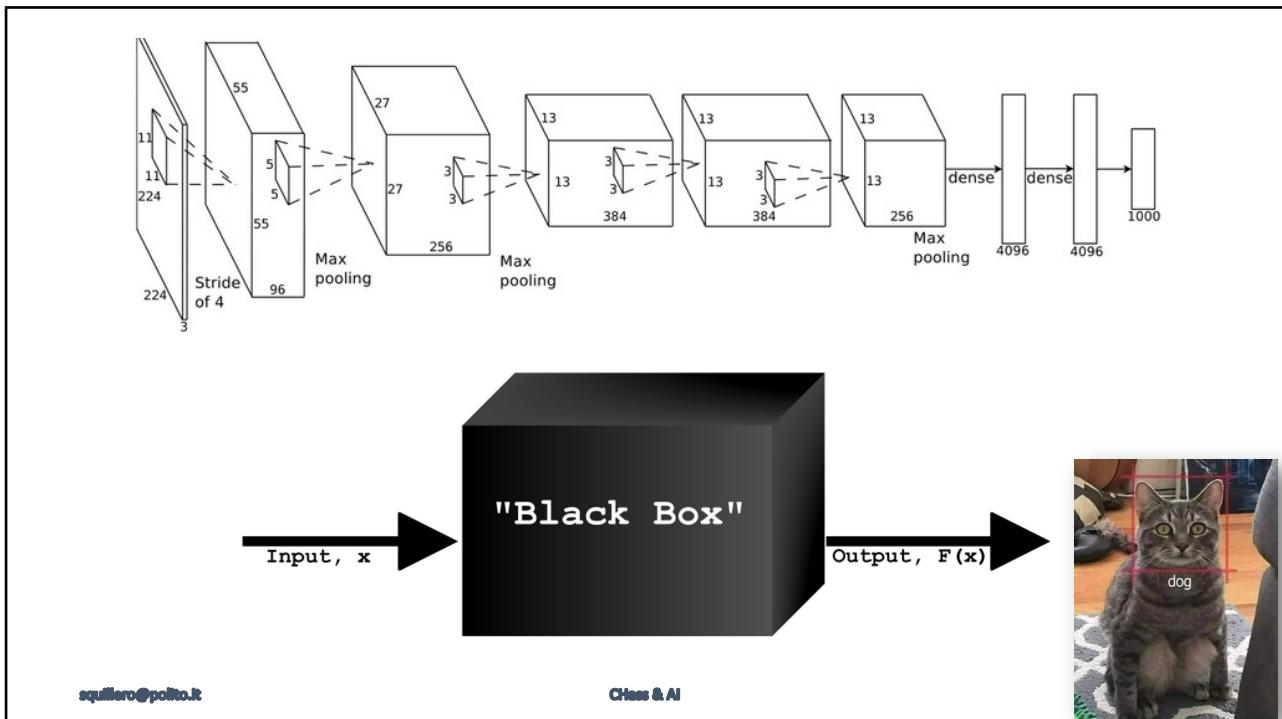


squillero@polito.it

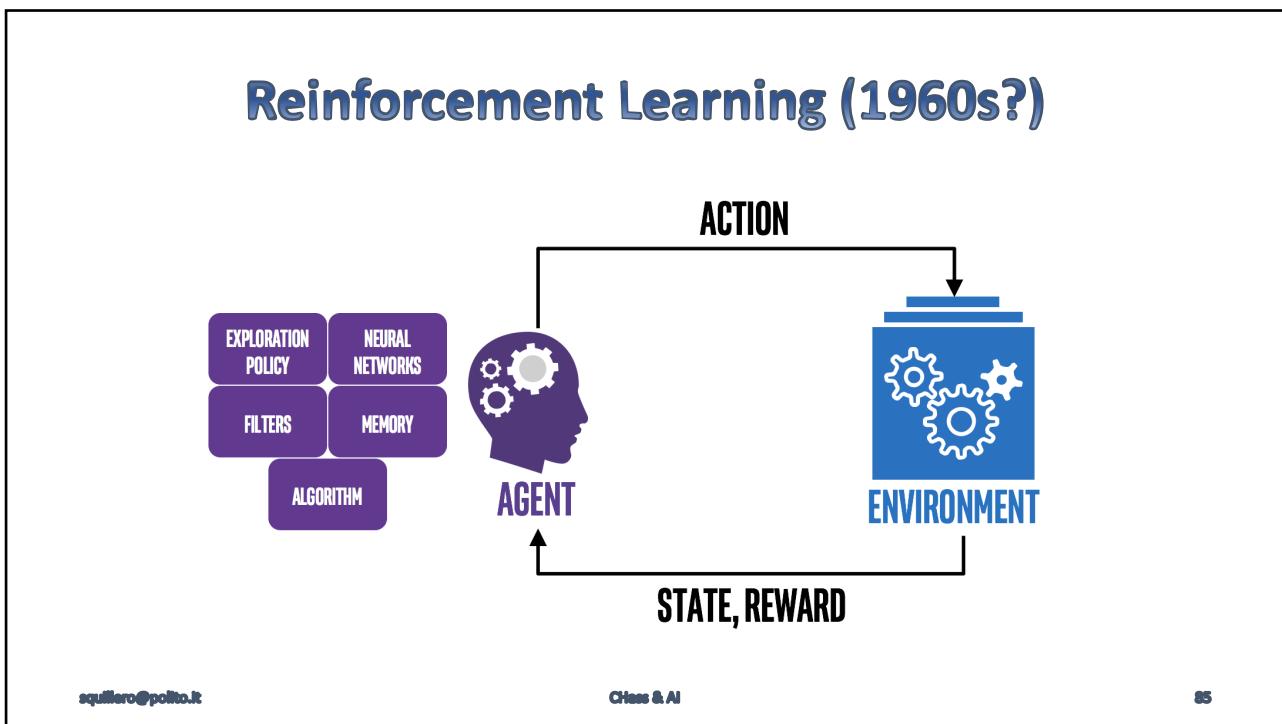
Chess & AI

83

83

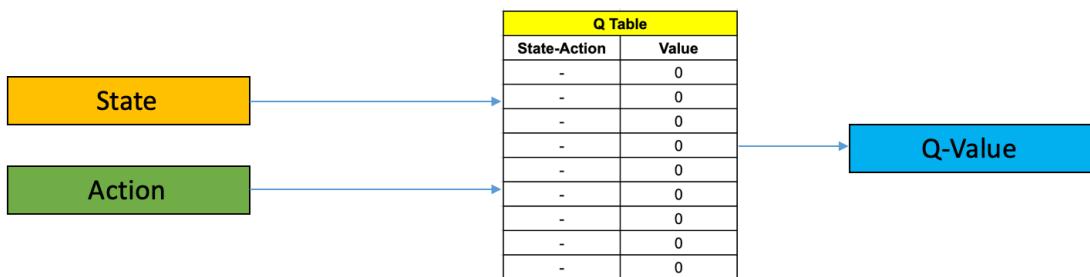


84



85

Q Learning



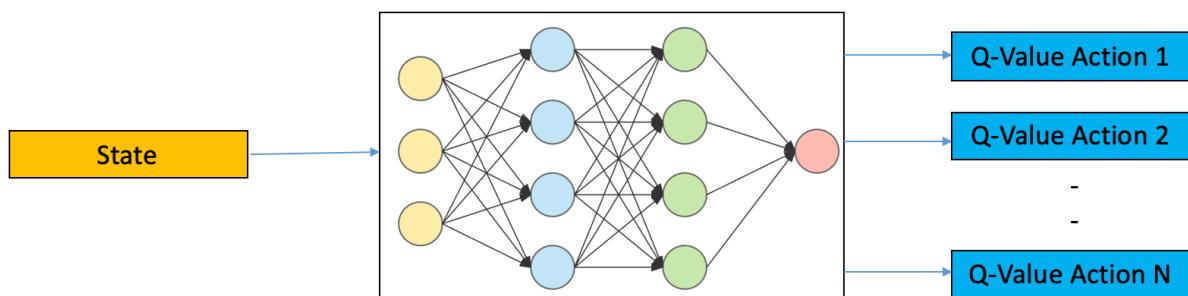
squillero@polito.it

Chess & AI

86

86

Deep Q Learning



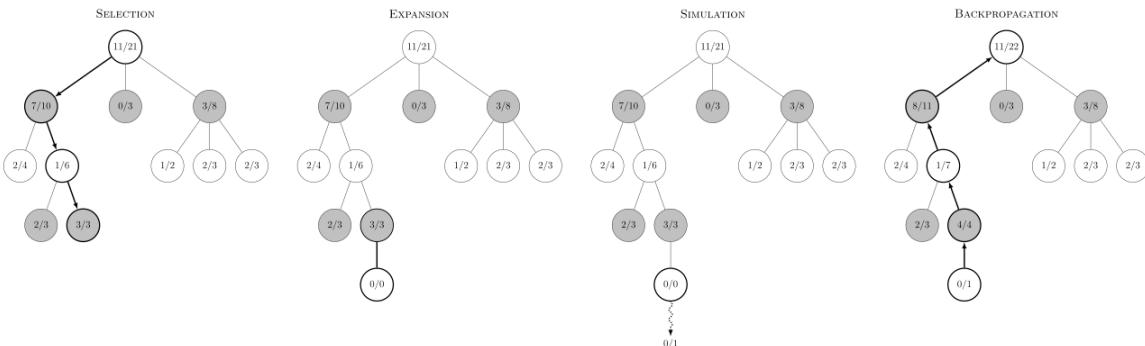
squillero@polito.it

Chess & AI

87

87

Montecarlo Tree Search (2006)



squillero@polito.it

Chess & AI

88

88

AlphaGo (2016)



squillero@polito.it

Chess & AI

89

89

40

AlphaGo (2016)

- Two “deep” neural networks
- Reinforcement Learning
- Montecarlo Tree Search

The diagram illustrates the AlphaGo architecture. It starts with "Human expert positions" which are used for "Classification" to train the "Rollout policy" (p_π) and the "SL policy network" (p_σ). These policies are then used in a "Policy gradient" loop to refine the "RL policy network" (p_ρ). Simultaneously, "Self-play positions" are generated through "Self-play" (using the RL policy) and "Regression" (using the Value network v_θ). These self-play positions are used to train the "Policy network" ($p_{\phi(p)}(a|s)$) and the "Value network" ($v_\theta(s')$).

Legend for diagrams a-f:

- Percentage of simulations
- ◆ Tree evaluation from value net
- Tree evaluation from rollouts
- ▲ Policy network
- Principal variation

90

2nd Game: Move 37

The Go board shows the state of the game after Move 37. Black stones are at (B, 16), (C, 16), (D, 16), (J, 17), (M, 16), (N, 16), (O, 15), (P, 15), (Q, 15), (R, 15). White stones are at (F, 13), (D, 10), (P, 11), (P, 14), (Q, 14), (R, 14). Shaded stones represent captures or influence.

91

Suggested “Readings”



92

AlphaGo Family

- AlphaGo Fan (2015) Won vs. Fan Hui
- AlphaGo Lee (2016) 4:1 vs. Lee Sedol
- AlphaGo Master (2017) 60:0 vs. professional players
- AlphaGo Zero (2017) 100:0 vs. AlphaGo Lee
 89:11 vs. AlphaGo Master

squillero@polito.it

Chess & AI

93

93

AlphaZero

- AlphaGo Fan (2015) Won vs. Fan Hui
- AlphaGo Lee (2016) 4:1 vs. Lee Sedol
- AlphaGo Master (2017) 60:0 vs. professional players
- AlphaGo Zero (2017) 100:0 vs. AlphaGo Lee
89:11 vs. AlphaGo Master
- **AlphaZero (2017)** 60:40 vs. AlphaGo Zero

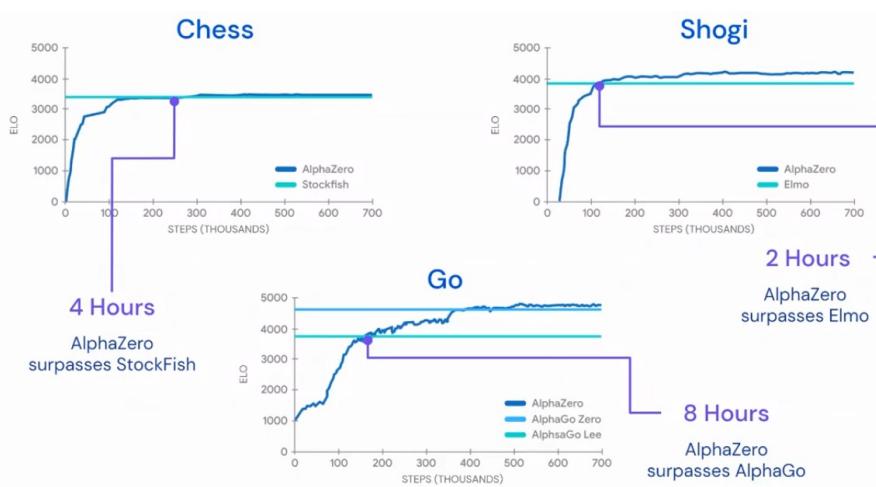
squillero@polito.it

Chess & AI

94

94

Learn from Self Play



squillero@polito.it

Chess & AI

95

95

Amount of Search Required

- State-of-the-art Chess Engine
 - **10,000,000** positions
- AlphaZero
 - **10,000** positions
- Human GM
 - **100** positions

squillero@polito.it

Chess & AI

96

96

AlphaZero's Zugzwang Game



squillero@polito.it

Chess & AI

97

97

MuZero

- AlphaGo Fan (2015) Won vs. Fan Hui
- AlphaGo Lee (2016) 4:1 vs. Lee Sedol
- AlphaGo Master (2017) 60:0 vs. professional players
- AlphaGo Zero (2017) 100:0 vs. AlphaGo Lee
89:11 vs. AlphaGo Master
- AlphaZero (2017) 60:40 vs. AlphaGo Zero
- MuZero (2019)

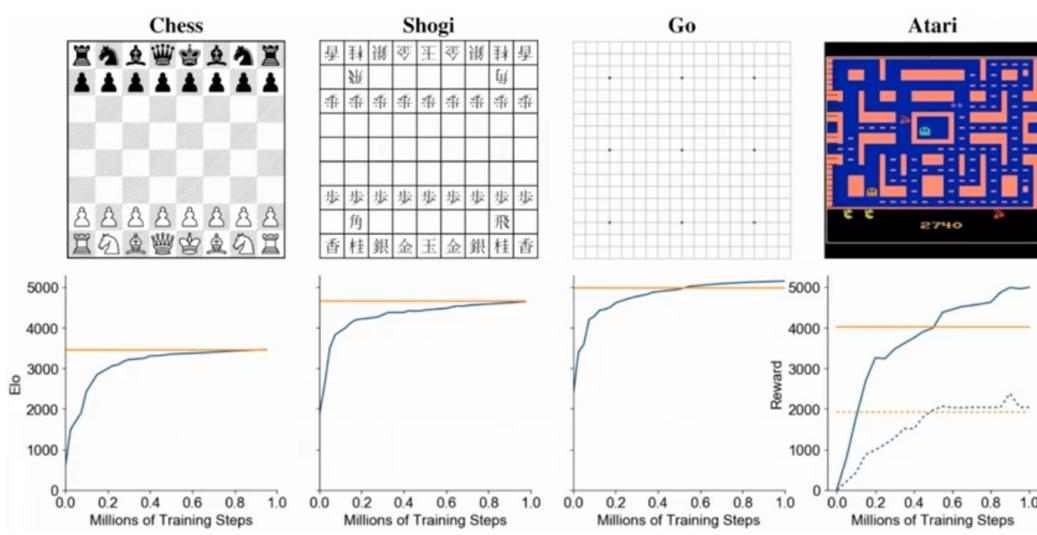
squillero@polito.it

Chess & AI

98

98

MuZero Performances



squillero@polito.it

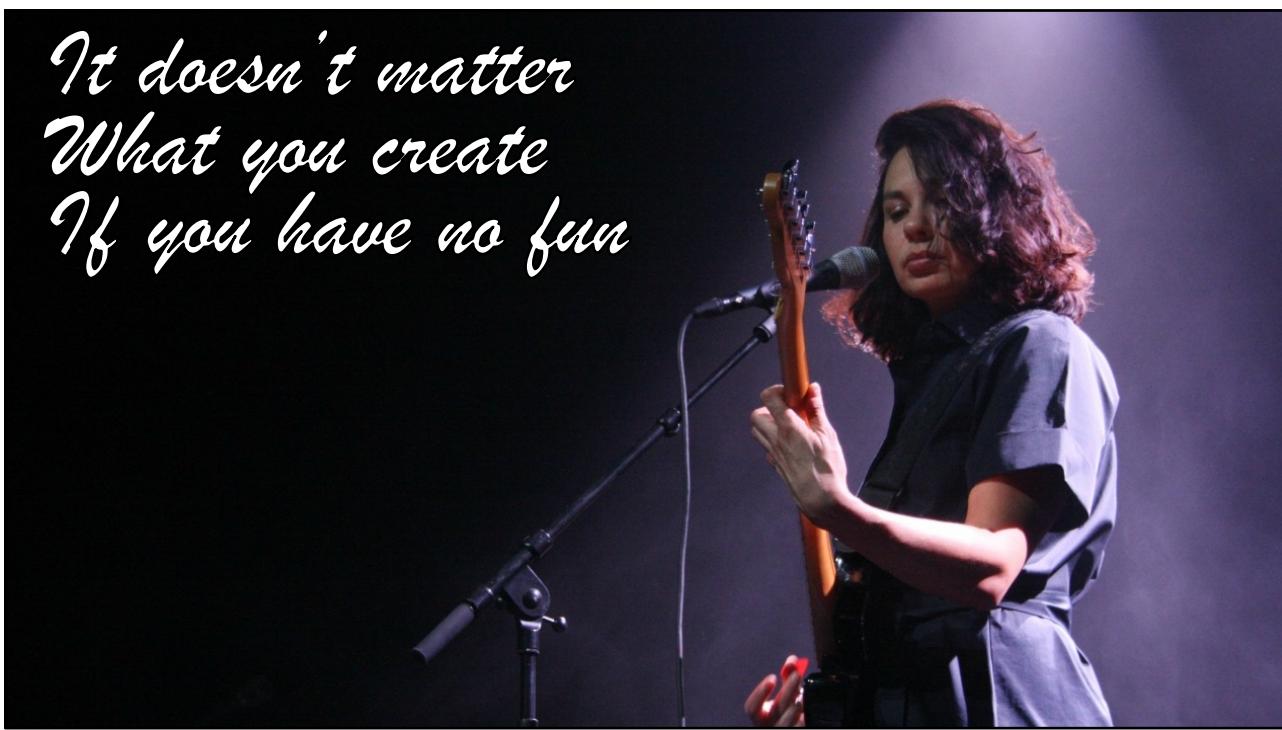
Chess & AI

99

99



100



103