1 handout: slides

You think you know when you can learn, are more sure when you can write, even more when you can teach, but certain when you can program.

EOLQs

Adversarial Search

Planning Problems

Observability: complete, partial, hidden
State: discrete, continuous
Actions: deterministic, stochastic, discrete, continuous
Nature: static, deterministic, stochastic
Interaction: one decision, sequential
Time: static/off-line, on-line, discrete, continuous
Percepts: discrete, continuous, uncertain
Others: solo, cooperative, competitive

Multi-agent is Different

- Shortest-path (M&C, vacuum, tile puzzle)
  - want least-cost path to goal at unkown depth
- Decisions with an adversary (chess, tic-tac-toe)
  - adversary might prevent path to best goal
  - want best assured outcome assuming rational opponent
  - irrational opponent can only be worse

Adversarial Search: Minimax

Each *ply* corresponds to half a *move*.
Terminal states are labeled with value.

incorrect version by Zermelo (1912)
full treatment by von Neumann and Morgenstern (1944)

Can also bound depth and use a *static evaluation function* on non-terminal states.

Evaluation for Tic-tac-toe

A *3-length* is a complete row, column, or diagonal.

$$\begin{aligned}\text{value of position} &= \infty \text{ if win for me,}\\ \text{or} &= -\infty \text{ if a win for you,}\\ \text{otherwise} &= \#\text{ 3-lengths open for me } -\\ &\qquad \#\text{ 3-lengths open for you}\end{aligned}$$

Tic-tac-toe: two-ply search

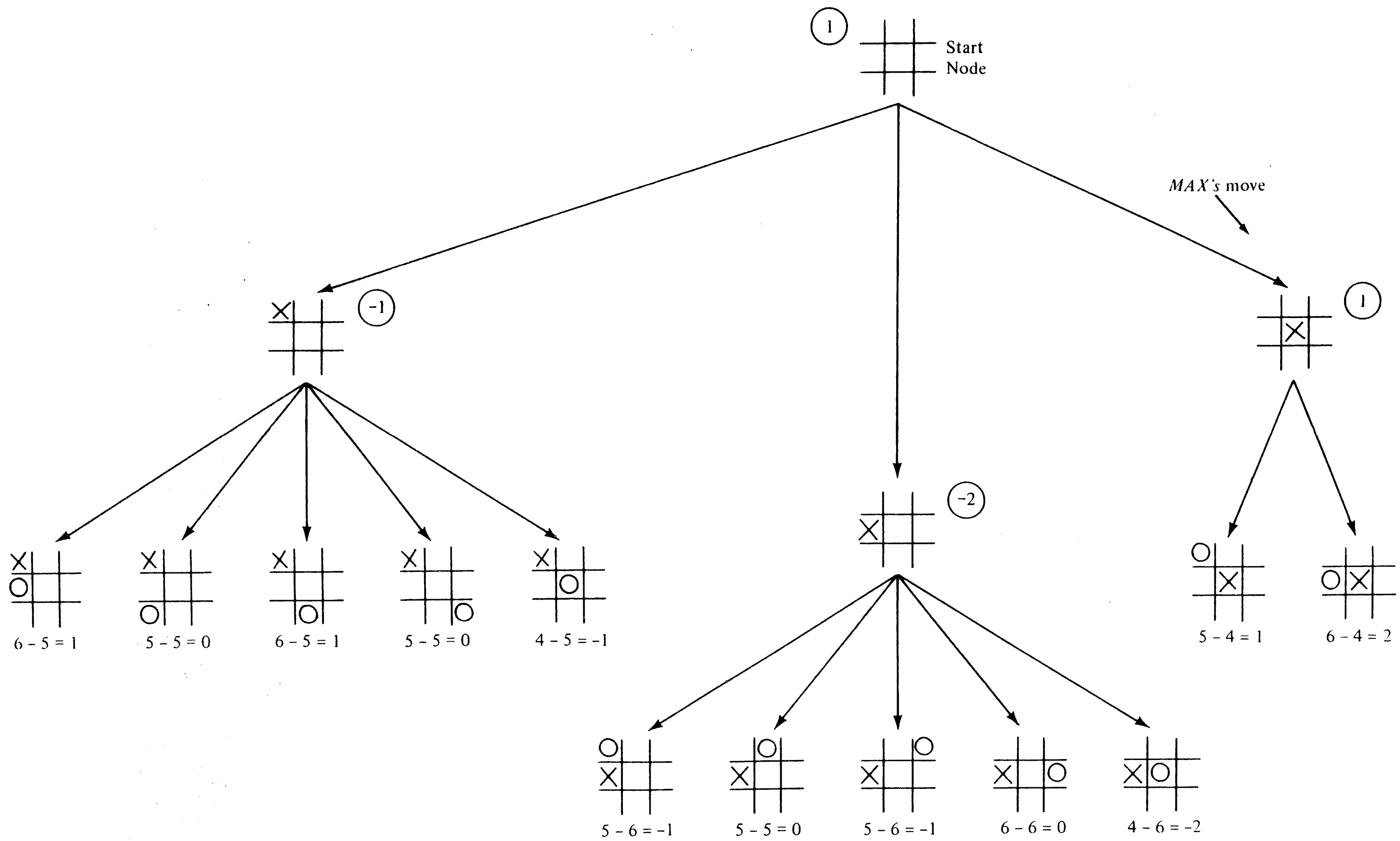

Fig. 3.8 Minimax applied to tic-tac-toe (stage 1).

Tic-tac-toe: second move

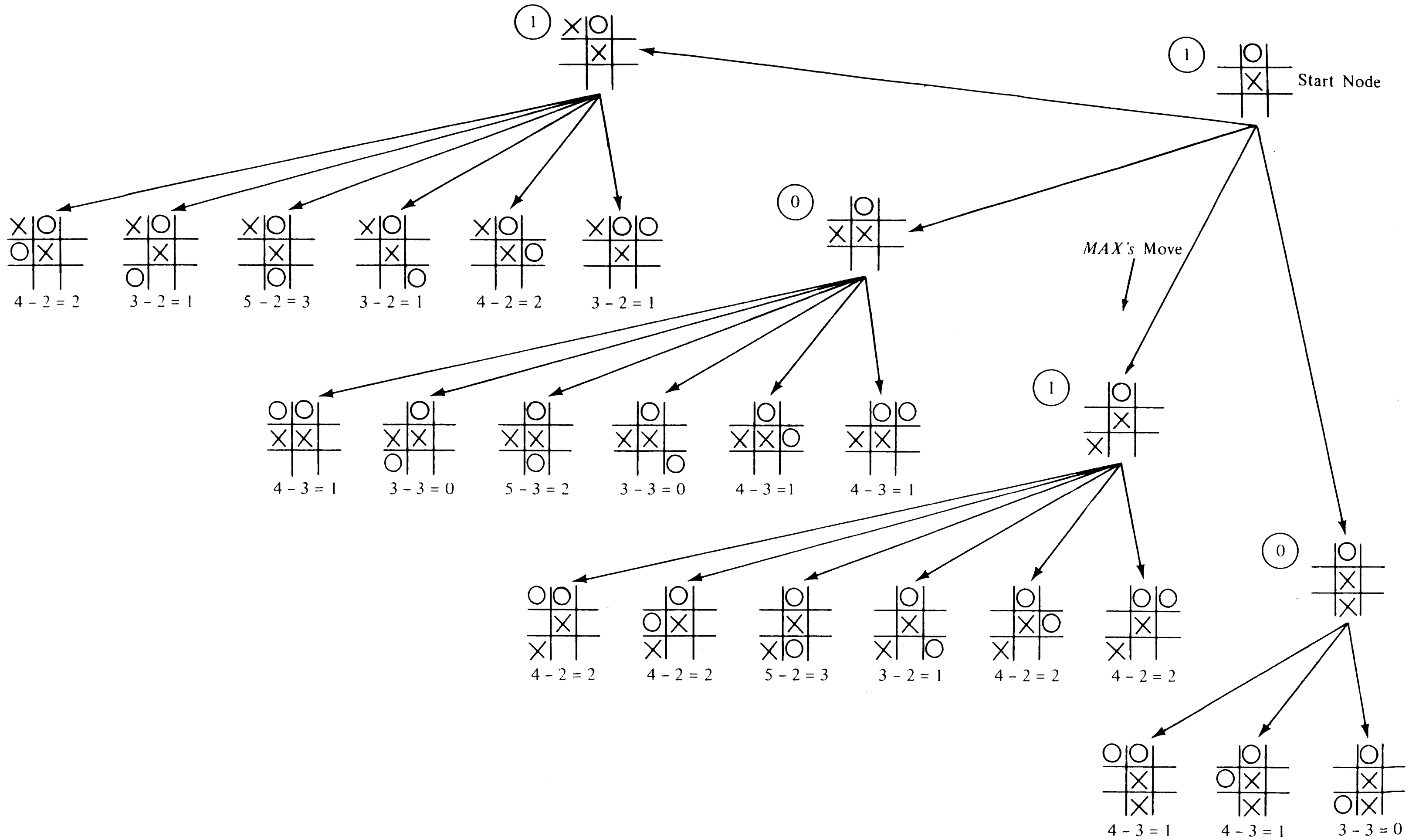

Fig. 3.9 Minimax applied to tic-tac-toe (stage 2).

Tic-tac-toe: third move

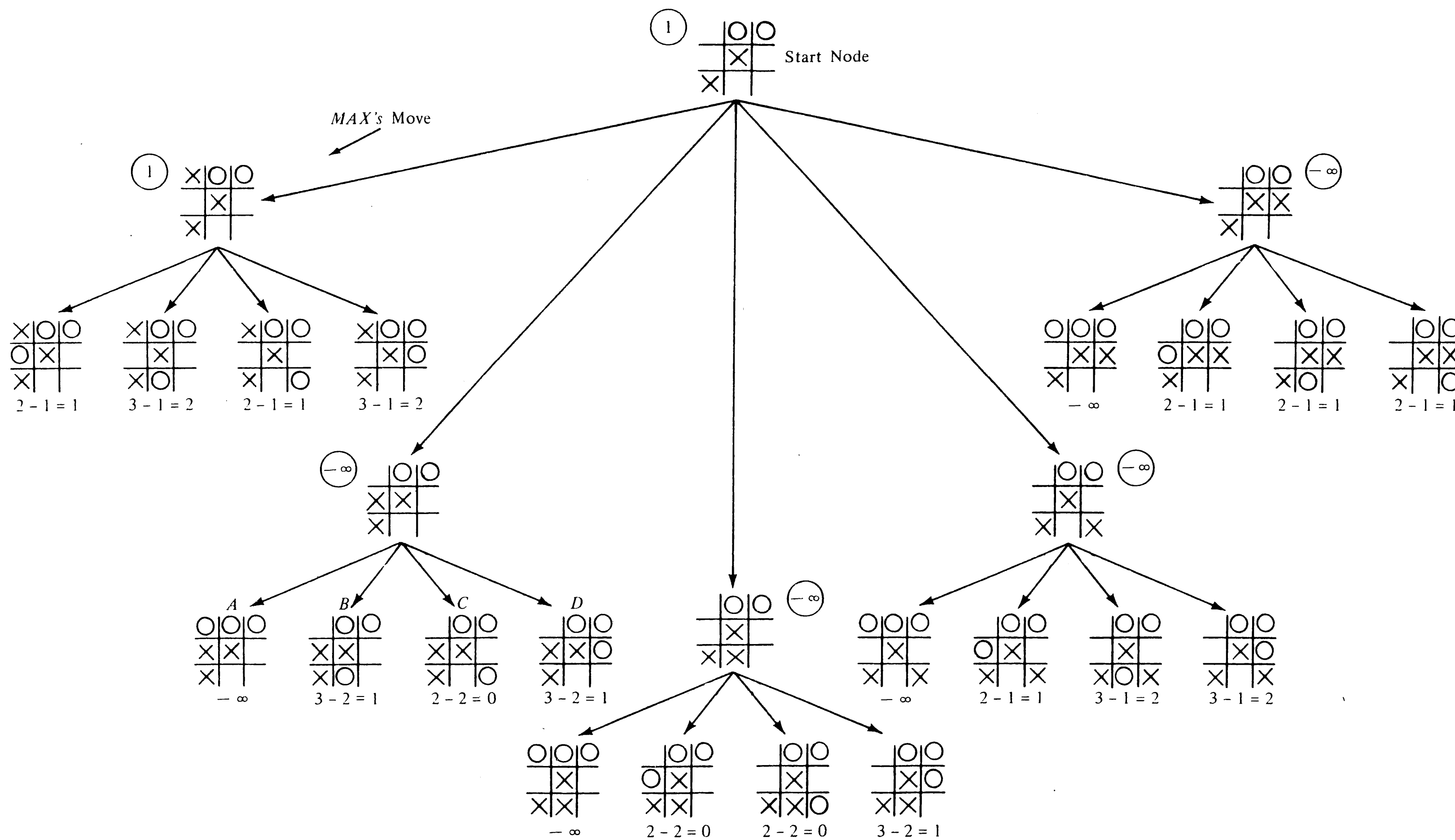

Fig. 3.10 Minimax applied to tic-tac-toe (stage 3).

Improving the Search

- partial expansion, SEF
- symmetry ('transposition tables')
- search more ply as we have time (De Groot figure)
- avoid unnecessary evaluations

Break

- asst 3
- asst 4
- projects! proposal draft due in class Mar 11

Which Values are Necessary?

α-β Pruning

α best outcome Max can force at previous decision on this path (init to $-\infty$)
β best outcome Min can force at previous decision on this path (init to ∞)

α and β values are copied down the tree (but not up).
Minmax values are passed up the tree, as usual.

John McCarthy (1956 but never published)
simple version used by Newell, Shaw, and Simon (1958)
published by Hart and Edwards (1961)
proved correct and analyzed by Knuth and Moore (1975)
proved optimal by Pearl (1982)

α-β Pseudo-code

Max-value (state, α, β):
  when depth-cutoff (state), return SEF(state)
  for each child of state
    $\alpha \leftarrow$ max(α, Min-value (child, α, β))
    when $\alpha \geq \beta$, return α
  return α

Min-value (state, α, β):
  when depth-cutoff (state), return SEF(state)
  for each child of state
    $\beta \leftarrow$ min(β, Max-value (child, α, β))
    when $\beta \leq \alpha$, return β
  return β

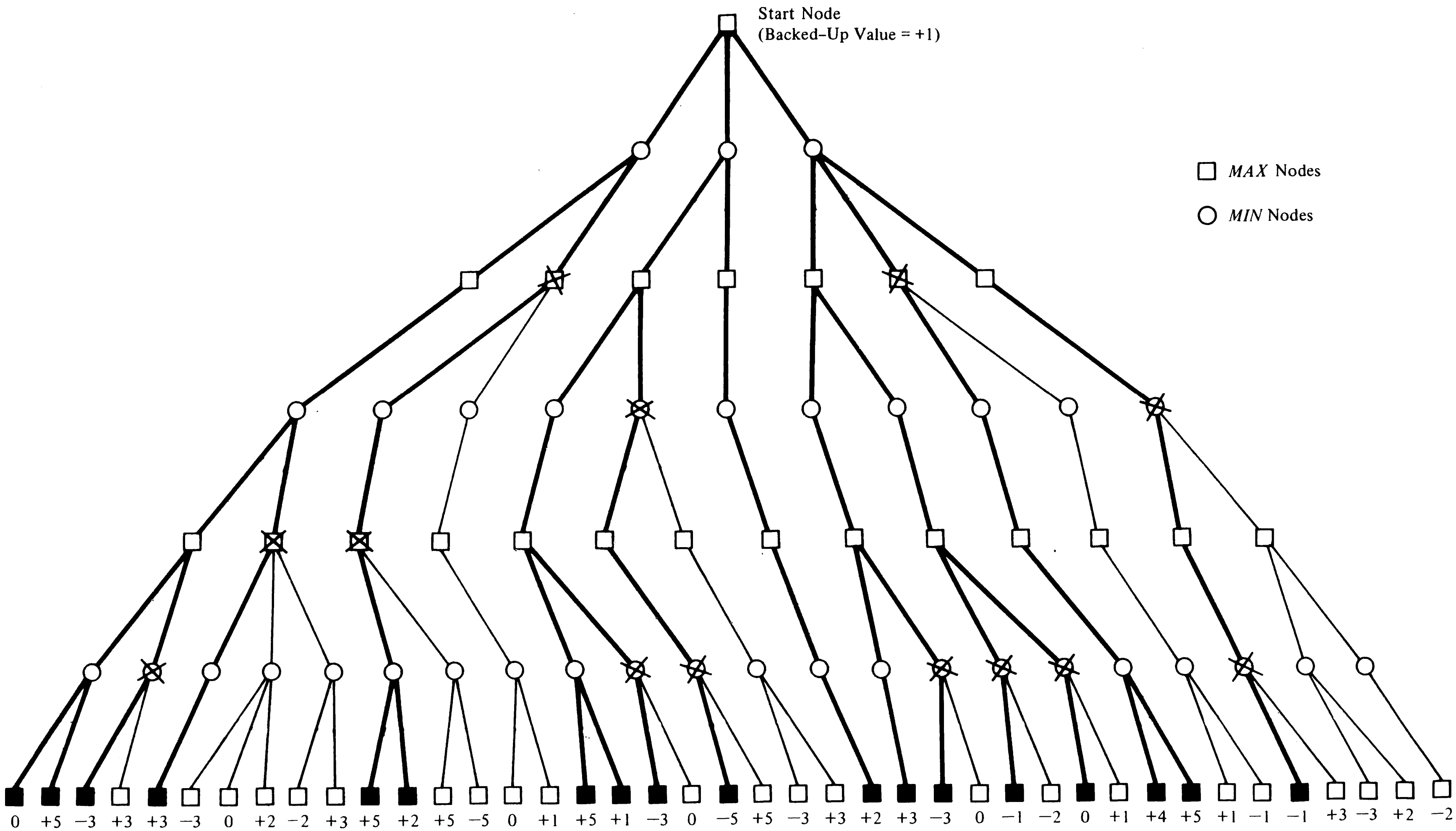

Fig. 3.12 An example illustrating the alpha-beta search procedure.

Why α-β?

Time complexity of α-β is about $O(b^{d/2})$

Progress on Games

Computers best: chess, checkers, backgammon, Scrabble, Jeopardy, Go

Computers competitive: bridge, crosswords, poker, StarCraft

Computers amateur: soccer?

EOLQs

Please write down the most pressing question you have about the course material covered so far and put it in the box on your way out.

Thanks!