1 handout: slides
730W blog entries were due

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

Constraint Satisfaction Problems

Types of Search Problems

- Shortest-path (vacuum, tile puzzle, M&C)
  - given operators and their costs
  - want least-cost path to a goal
  - goal depth/cost unknown
- Constraint satisfaction (map coloring, n-queens)
  - any goal is fine
  - fixed depth
  - explicit constraints on partial solutions

Variable choice: choose most constrained variable (smallest domain)

- want to keep tree small, failing quickly

Value choice: try least constraining value first (fewest removals)

- might as well succeed sooner if possible

Example Results

	BT	FC	FC+MCV
USA	> 1M	2K	60
n-Queens	> 40M	> 40M	820K
Zebra	3.9M	35K	500
Random 1	420K	26K	2K
Random 2	940K	77K	15K

Maintaining Arc Consistency

Ensure every value for x has a legal value in all neighbors y. If one doesn't, remove it and ensure consistency of all y.

Maintaining Arc Consistency

Ensure every value for x has a legal value in all neighbors y. If one doesn't, remove it and ensure consistency of all y.

while Q is not empty
    $(x, y) \leftarrow$ pop Q
    if **revised**(x, y) then
        if x's domain is now empty, return failure
        for every other neighbor z of x
            push (z, x) on Q

revise(x, y)
revised$\leftarrow$ false
foreach v in x's domain
    if no value in domain of y is compatible with v
        remove v from x's domain
        revised$\leftarrow$ true
return *revised*

Other Algorithms for CSPs

- (Conflict-directed) Backjumping
- Dynamic backtracking
- Randomized restarting

Course projects!

Break

- what is a course project?
- asst 1 going out on Wed

Combinatorial Optimization

Types of Search Problems

- Shortest-path (M&C, vacuum, tile puzzle)
  - want least-cost path to a goal
  - goal depth unknown
  - given operators and their costs
- Constraint satisfaction (map coloring, n-queens)
  - any goal is fine
  - maximum depth $=$ number of variables
  - given explicit constraints on variables
- Combinatorial optimization (TSP, max-CSP)
  - want least-cost goal
  - maximum depth $=$ number of variables
  - every leaf is a solution

Hill-Climbing

Sol ← some random solution (probably poor quality).

Do *limit* times

  New ← random **neighbor** of *Sol*.

  If *New* better than *Sol*,

    then *Sol* ← *New*.

Hill-Climbing

Sol $\leftarrow$ some random solution (probably poor quality).
Do *limit* times
    New $\leftarrow$ random **neighbor** of *Sol*.
    If *New* better than *Sol*,
        then *Sol* $\leftarrow$ *New*.

Elaborations: best neighbor (aka gradient-descent)
restarts
simulated annealing
population (GAs, 'go with the winners')

Adversarial Search

Another Twist on Search

- Shortest-path (M&C, vacuum, tile puzzle)
  - want least-cost path to goal at unkown depth
- Constraint satisfaction (map coloring, n-queens)
  - any goal that satisfies constraints (fixed depth)
- Combinatorial optimization (TSP, max-CSP)
  - want least-cost goal (fixed depth)
- Decisions with an adversary (chess, tic-tac-toe)
  - adversary might prevent path to best goal
  - want best assured outcome

Adversarial Search: Minimax

Each *ply* corresponds to half a *move*.

Terminal states are labeled with value.

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

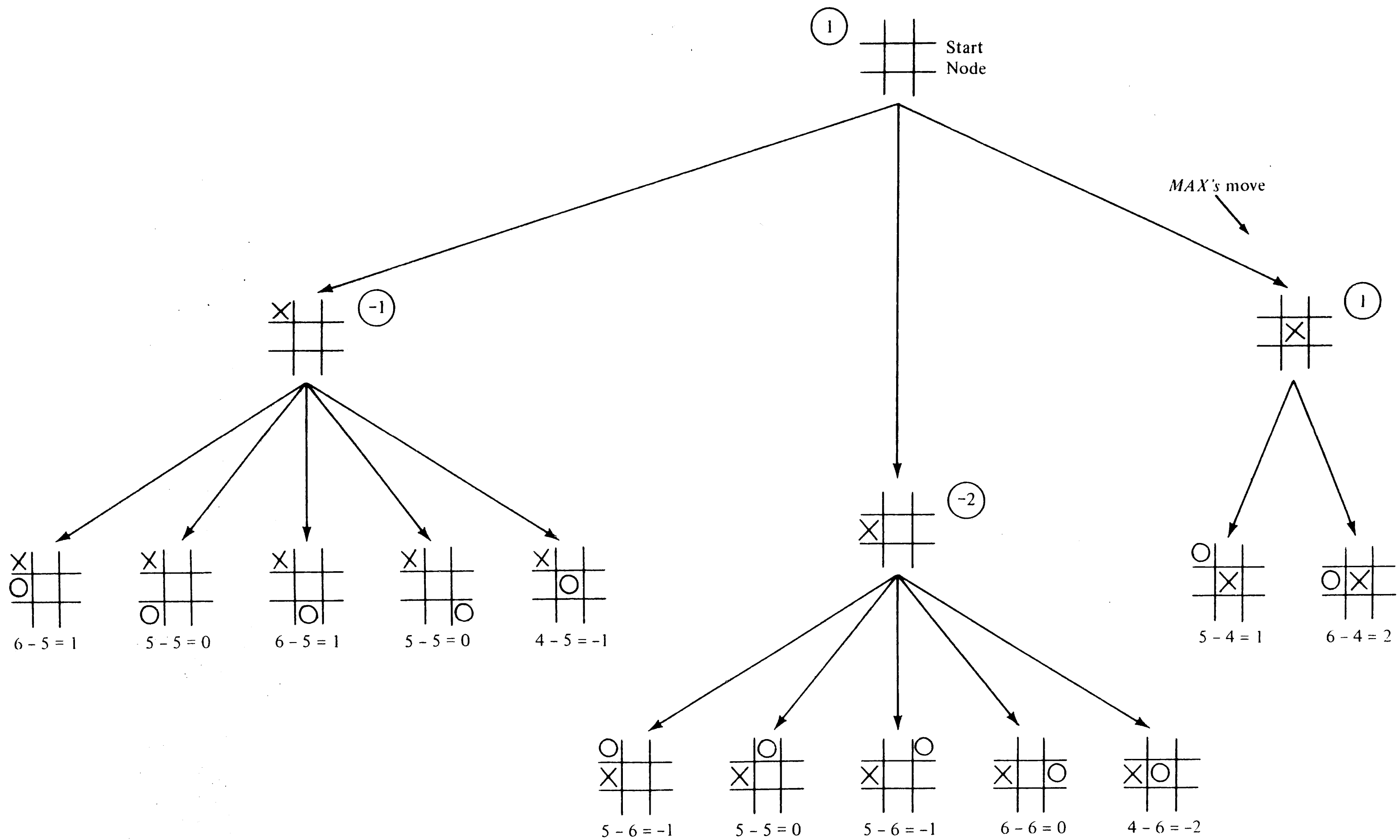

Fig. 3.8 Minimax applied to tic-tac-toe (stage 1).

Tic-tac-toe: second move

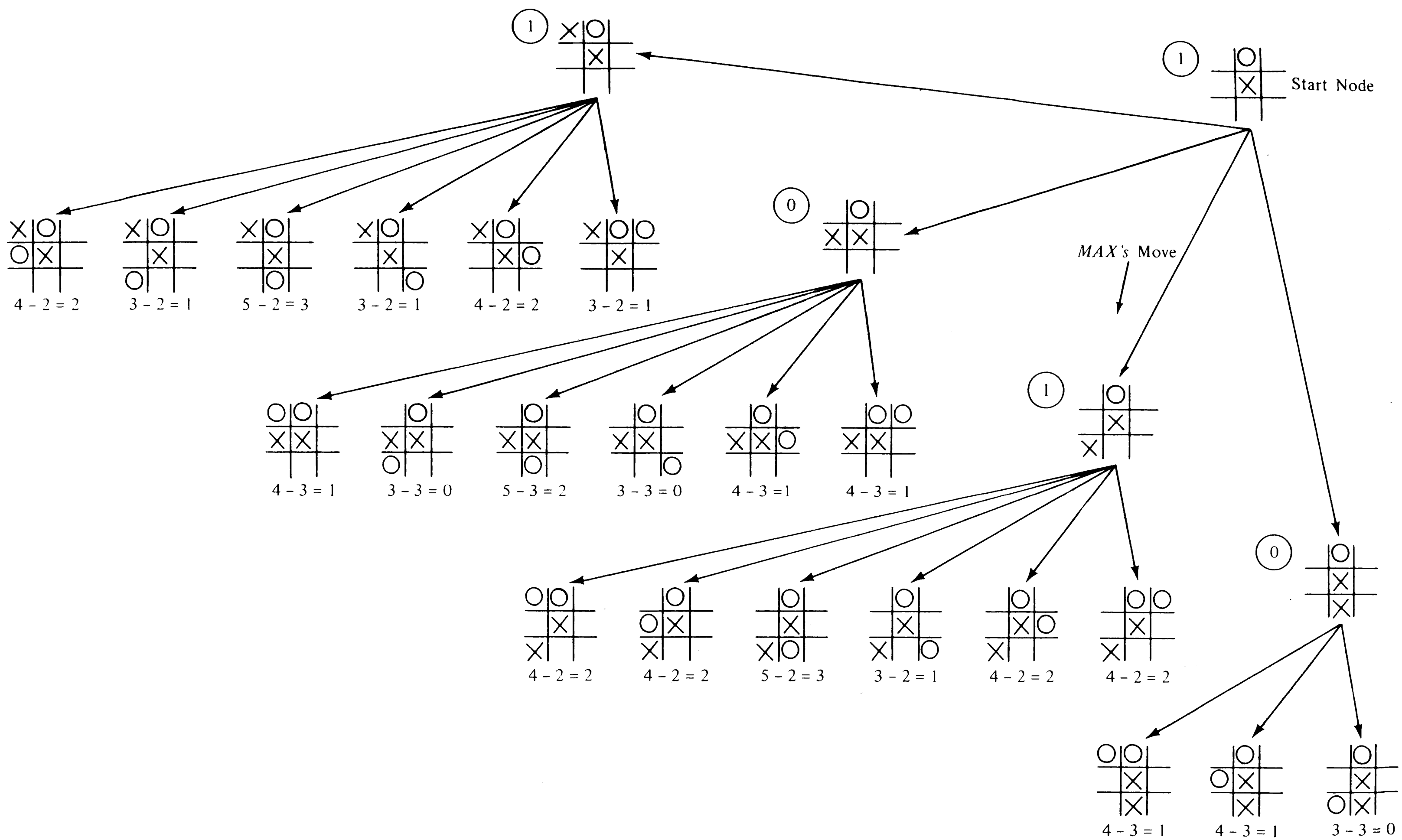

Fig. 3.9 Minimax applied to tic-tac-toe (stage 2).

Tic-tac-toe: third move

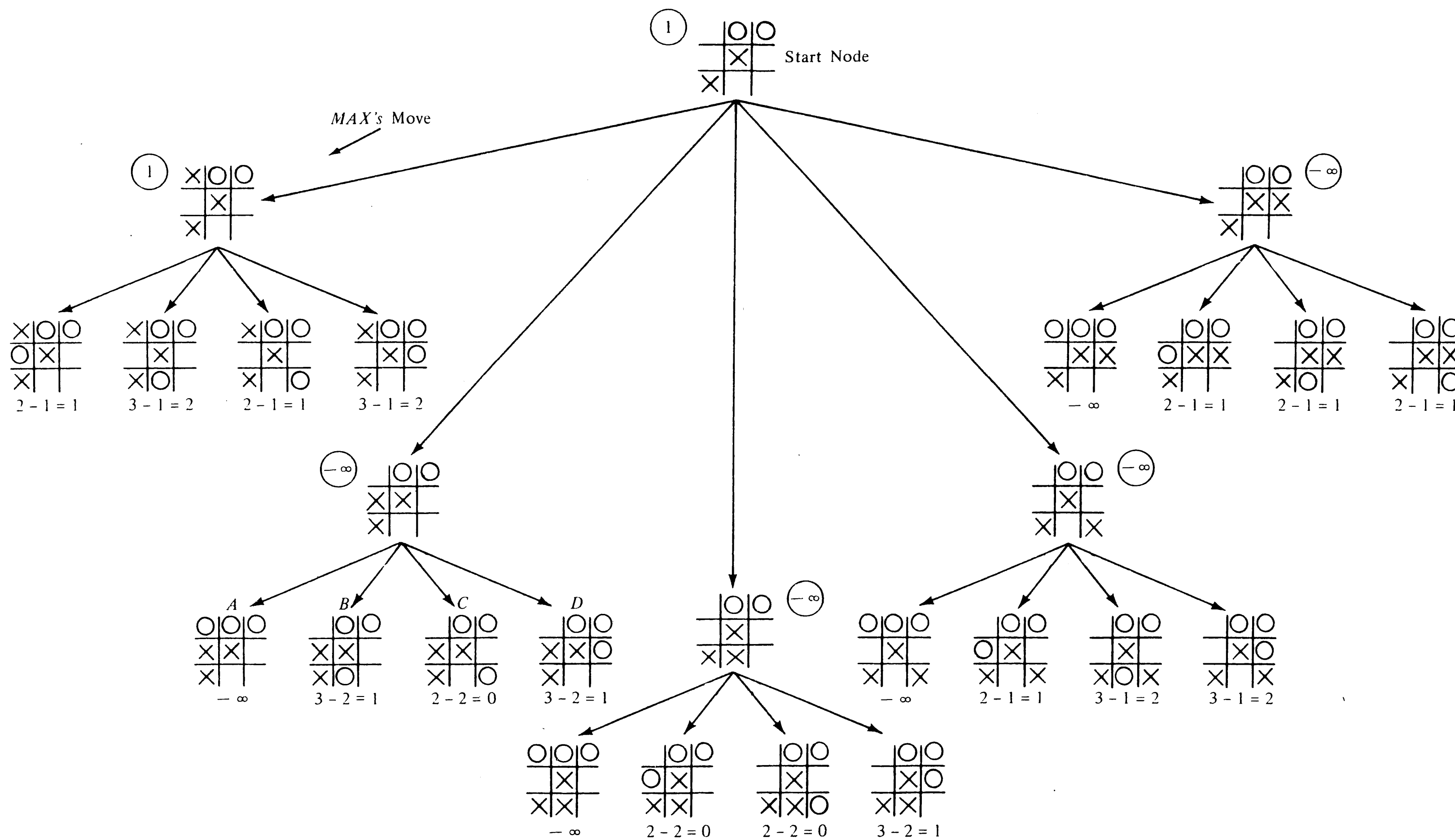

Fig. 3.10 Minimax applied to tic-tac-toe (stage 3).

Improving the Search

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

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!