National University of Singapore
School of Computing
CS3243 Introduction to AI

Tutorial 4: Adversarial Search and Constraint Satisfaction Problems

Issue: February 9, 2015 Due: March 6, 2015

Important Instructions:

- *Your solutions for this tutorial must be TYPE-WRITTEN.*
- *Make TWO copies of your solutions: one for you and one to be SUBMITTED TO THE TUTOR IN CLASS. Your submission in your respective tutorial class will be used to indicate your CLASS ATTENDANCE. Late submission will NOT be entertained.*
- *YOUR SOLUTION TO QUESTION* 3 *will be GRADED for this tutorial.*
- *You may discuss the content of the questions with your classmates. But everyone should work out and write up ALL the solutions by yourself.*

1. Consider Figure 5.1 in the AIMA 3rd edition textbook (reproduced in Figure 1). The Tic-Tac-Toe search space can actually be reduced by means of symmetry. This is done by eliminating those states which become identical with an earlier state after a symmetry operation (e.g. rotation). The following diagram shows a reduced state space for the first three levels with the player making the first move using "x" and the opponent making the next move with "o". Assume that the following heuristic evaluation function is used at each leaf node n:

$$Eval(n) = P(n) - O(n)$$

where $P(n)$ is the number of winning lines for the player while $O(n)$ is the number of winning lines for the opponent. A winning line for the player is a line (horizontal, vertical or diagonal) that either contains nothing or "x". For the opponent, it is either nothing or "o" in the winning line. Thus, for the leftmost leaf node in Figure 2, $Eval(n) = 6 - 5 = 1$.

   (a) Use the minimax algorithm to determine the first move of the player, searching 2-ply deep search space shown in Figure 2.

   (b) Assume that the "x" player will now make his second move after his opponent has placed an "o". Complete the following minimax tree in Figure 3 by filling the remaining blank boards at the leaf nodes. Compute the evaluation function for each of the filled leaf nodes and determine the second move of the "x" player (searching 2-ply deep).

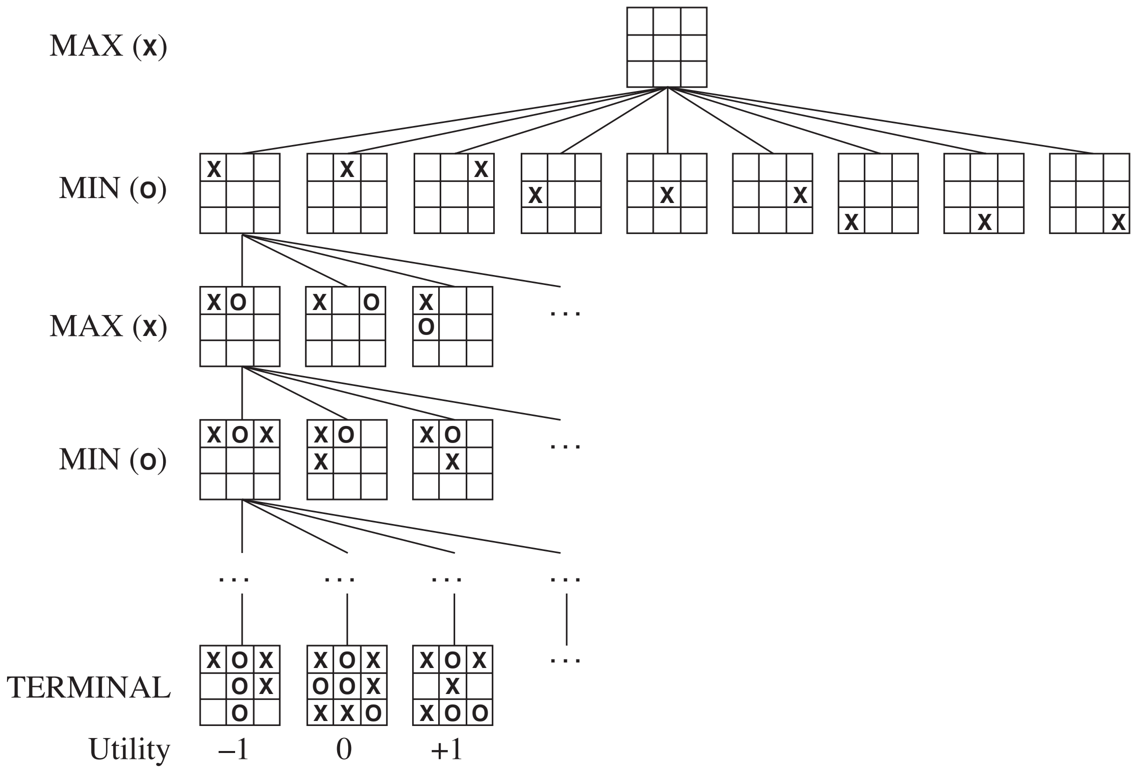


Figure 1: Search space for Tic-Tac-Toe.

(c) The minimax search tree in Figure 4 has heuristic evaluation function values with respect to the max player for all the leaf nodes, where square leaf nodes denote end of game with $+\infty$ representing that the max player wins the game and $-\infty$ representing that the min player is the winner. Do a minimax search and determine the next move of the max player from node A. Which is the target leaf node that the max player hopes to reach?

(d) Suppose we use alpha-beta pruning in the direction from left to right to prune the search tree in question 1c. Indicate which arcs are pruned by the procedure. Do you get the same answer in terms of the max player's next move and target leaf node?

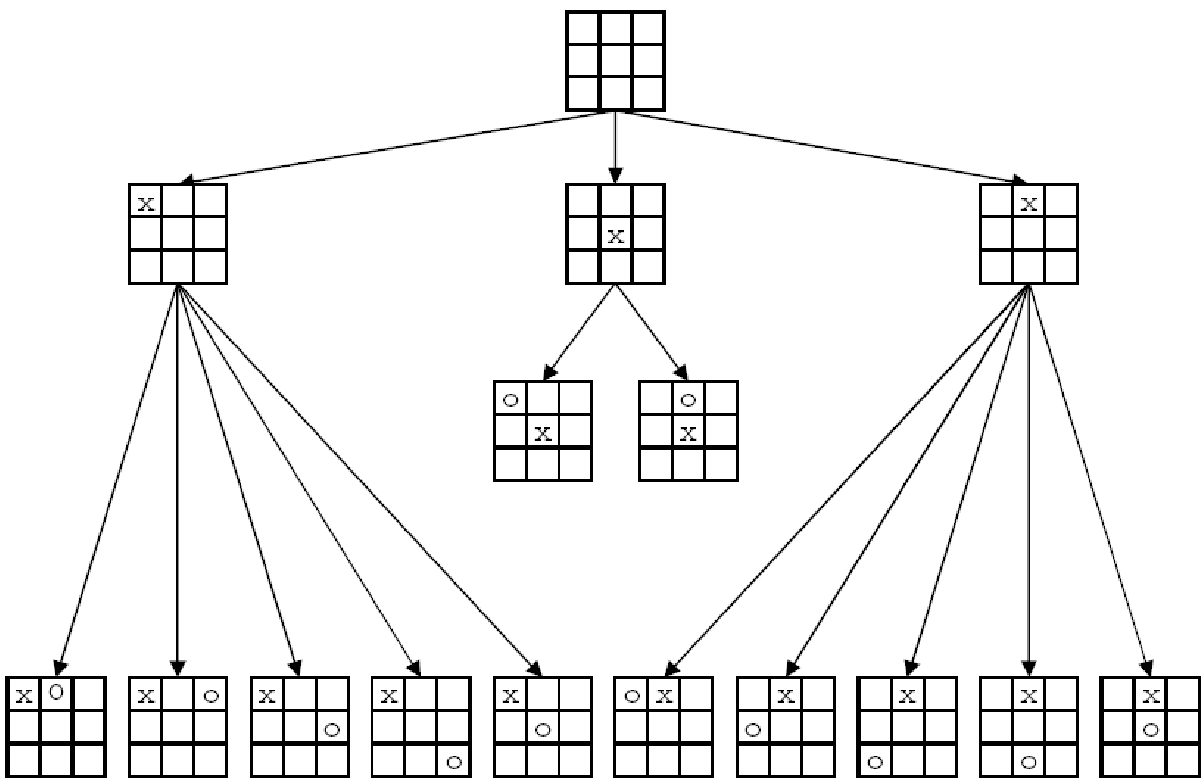


Figure 2: 2-ply deep search space

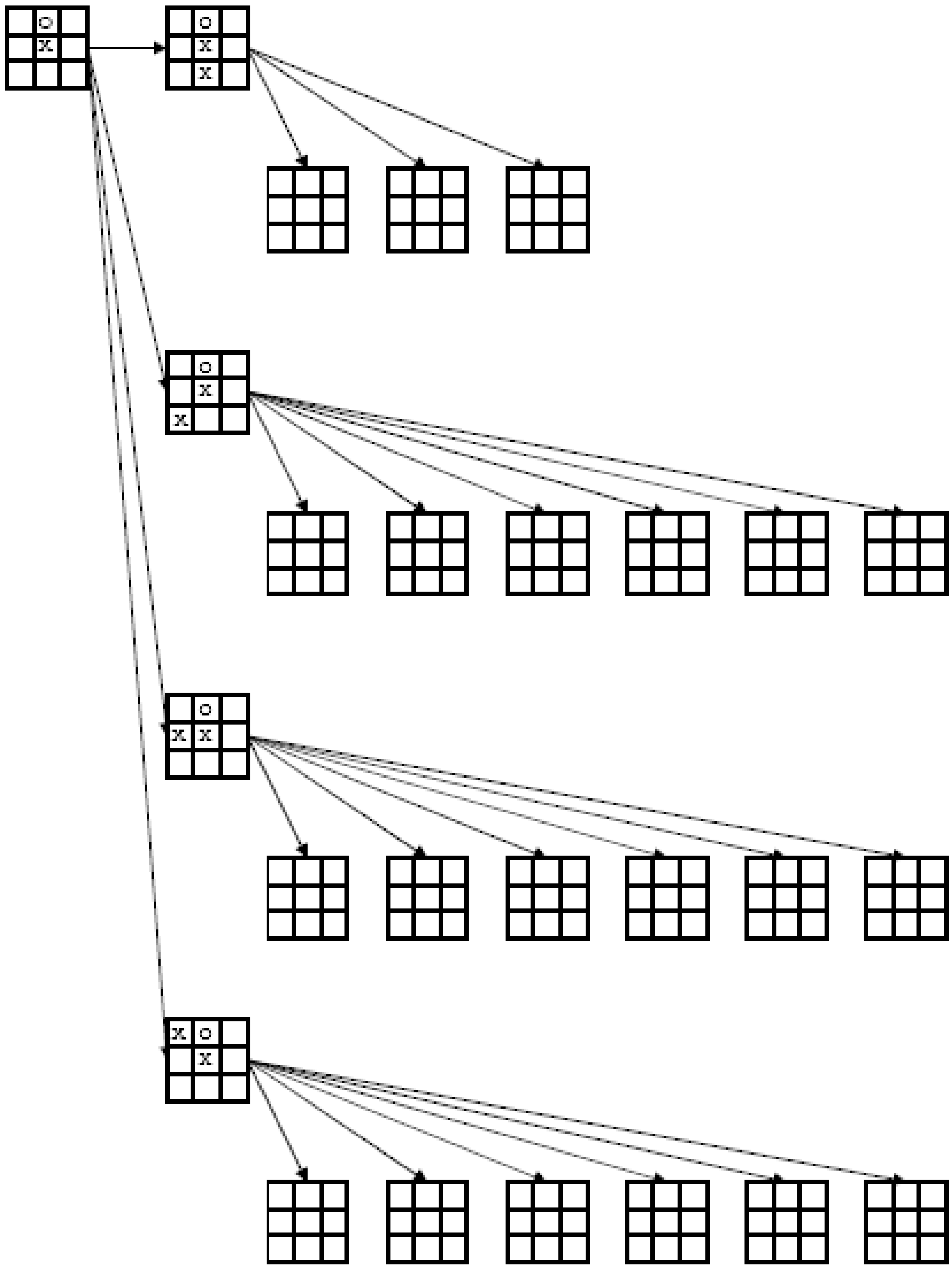


Figure 3: 3-ply deep search space

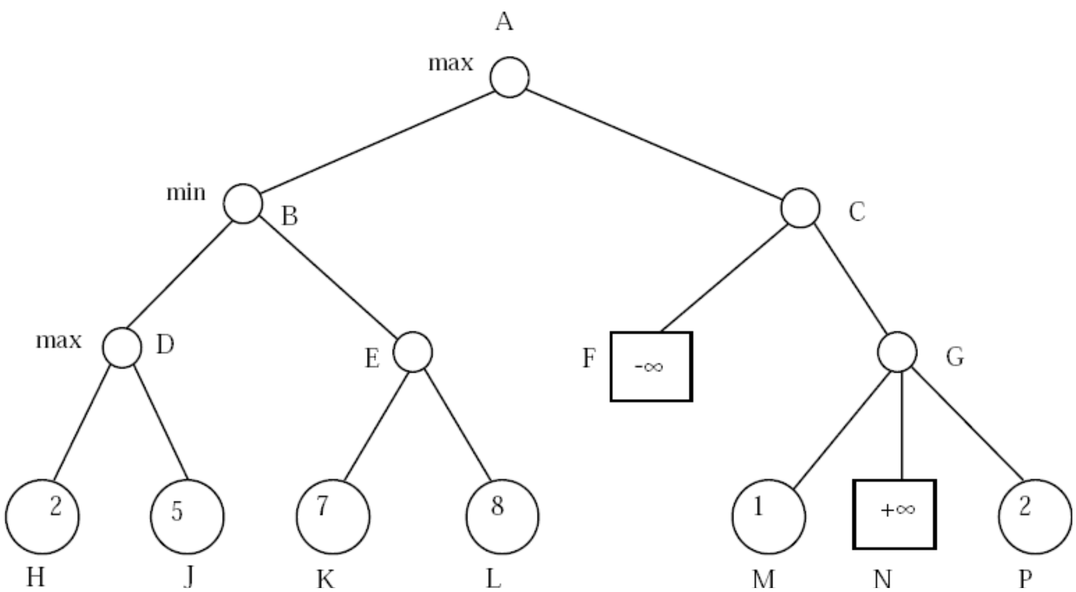


Figure 4: minimax search tree

2. (**Past-year exam question**) **Game of Nim Variant**. In this problem, we consider a **variant** of the Game of Nim that has the following rules:

   - There are two piles, each with two sticks.
   - Each player may **take only either one or two sticks** from an existing pile during his turn.
   - The player who picks the last stick **wins** the game.

   (a) Propose a representation for the state of this game.

   (b) Using your proposed representation in Part (a), draw the game tree for this game.

   (c) Let a state that results in a win for the first player (MAX) be of value 1, while a state that results in a win for the second player (MIN) be of value -1. Solve the game by assigning a value to each node in the game tree in Part B (Please annotate in your answer for Part B above). If your objective is to win this game, would you opt to move first or second?

   (d) Suppose instead of two piles of two sticks each, we have two piles of ***three*** sticks each. Now, would you opt to move first or second? Justify your answer.

3. Consider the following constraint satisfaction problem:

   Variables:
   $$A, B, C$$
   Domains:
   $$D_A = D_B = D_C = \{0, 1, 2, 3, 4\}$$
   Constraints:
   $$\begin{aligned} A &= B + 1 \\ B &= 2C \end{aligned}$$

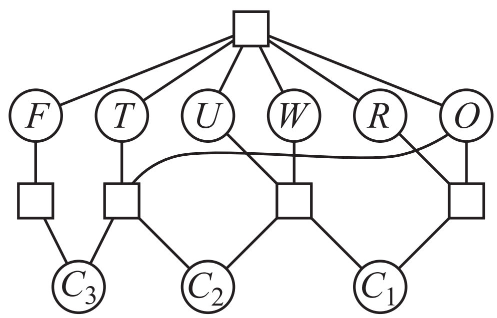
   Construct a constraint graph for this problem. Show a trace of the `AC-3` algorithm on this problem. Assume that initially, the arcs in queue are in the order $\{(A, B), (B, A), (B, C), (C, B)\}$.

4. Consider the 4-queens problem on a 4×4 chess board. Suppose the leftmost column is column 1, and the topmost row is row 1. Let Q_i denote the row number of the queen in column i, $i = 1, 2, 3, 4$. Assume that variables are assigned in the order Q_1, Q_2, Q_3, Q_4, and the domain values of Q_i are tried in the order $1, 2, 3, 4$. Show a trace of the backtracking algorithm with forward checking to solve the 4-queens problem.

5. Show a trace of the backtracking algorithm with forward checking to solve the cryptarithmetic problem shown in Figure 5. Use the most constrained variable heuristic, and assume that the domain values (digits) are tried in ascending order (i.e., 0, 1, 2, $\cdots$).

$$\begin{array}{cccc} & T & W & O \\ + & T & W & O \\ \hline F & O & U & R \end{array}$$

(a)



(b)

Figure 5: Cryptarithmetic puzzle.