AI - HEURISTIC SEARCH

- **Heuristic search** refers to a **search** strategy that attempts to optimize a problem by iteratively improving the solution based on a given **heuristic** function or a cost measure.

- A Heuristic is a technique to solve a problem faster than classic methods, or to find an approximate solution when classic methods cannot.

- This is a kind of a shortcut as we often trade one of optimality, completeness, accuracy, or precision for speed.

Fig 4.1 First three levels of the tic-tac-toe state space reduced by symmetry

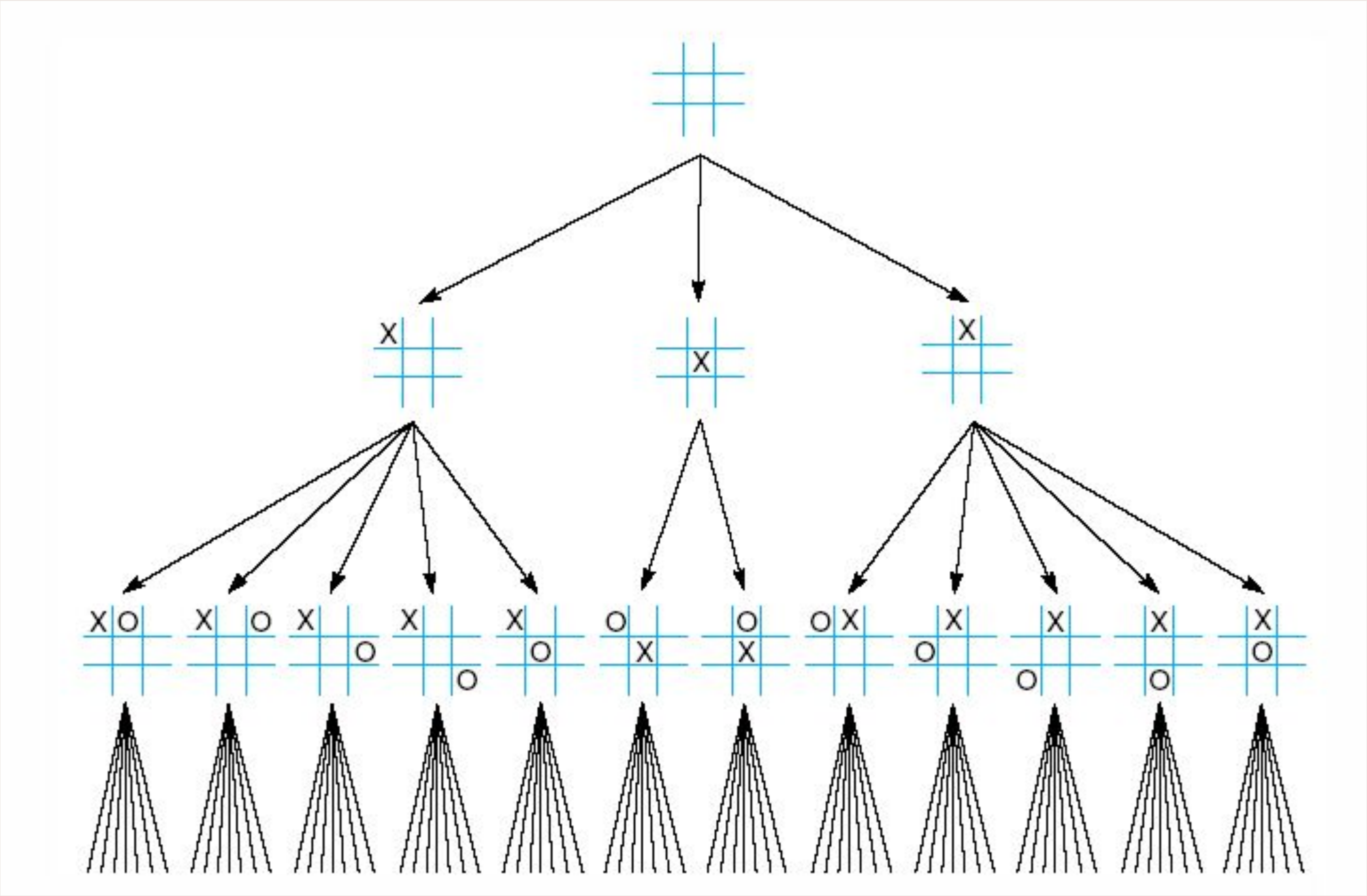


Fig 4.2 The "most wins" heuristic applied to the first children in tic-tac-toe.

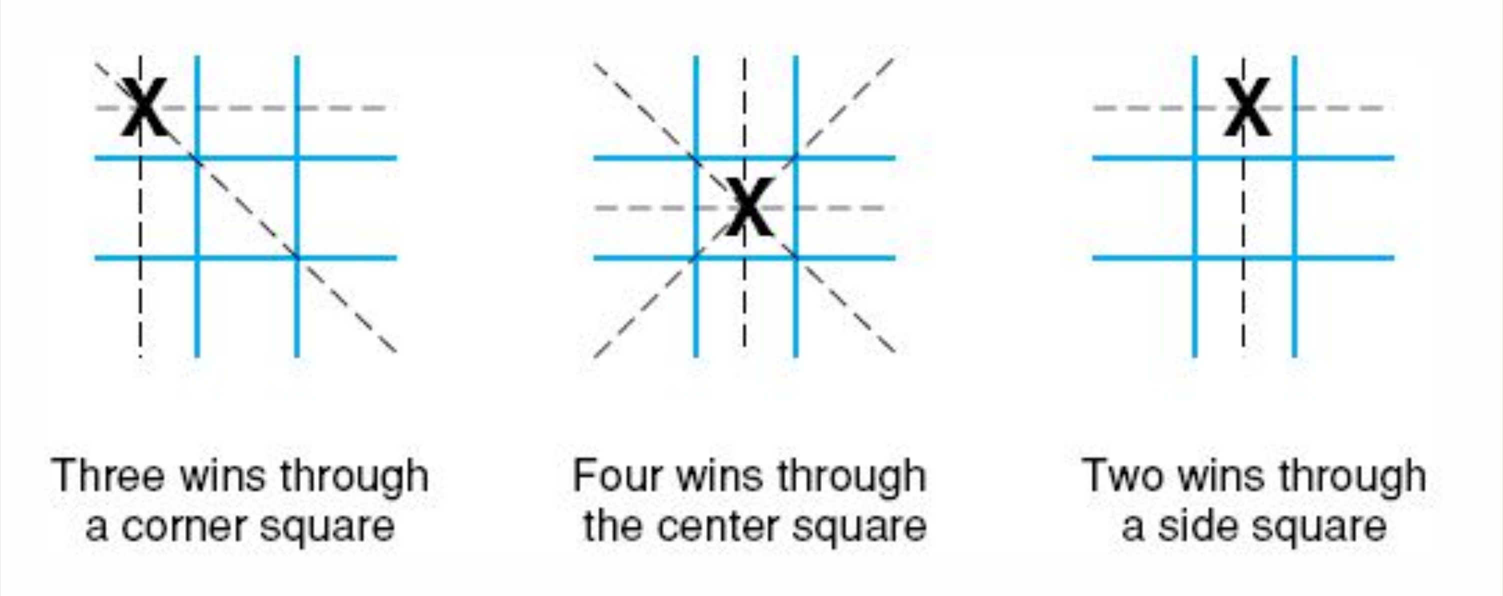


Fig 4.3 Heuristically reduced state space for tic-tac-toe.

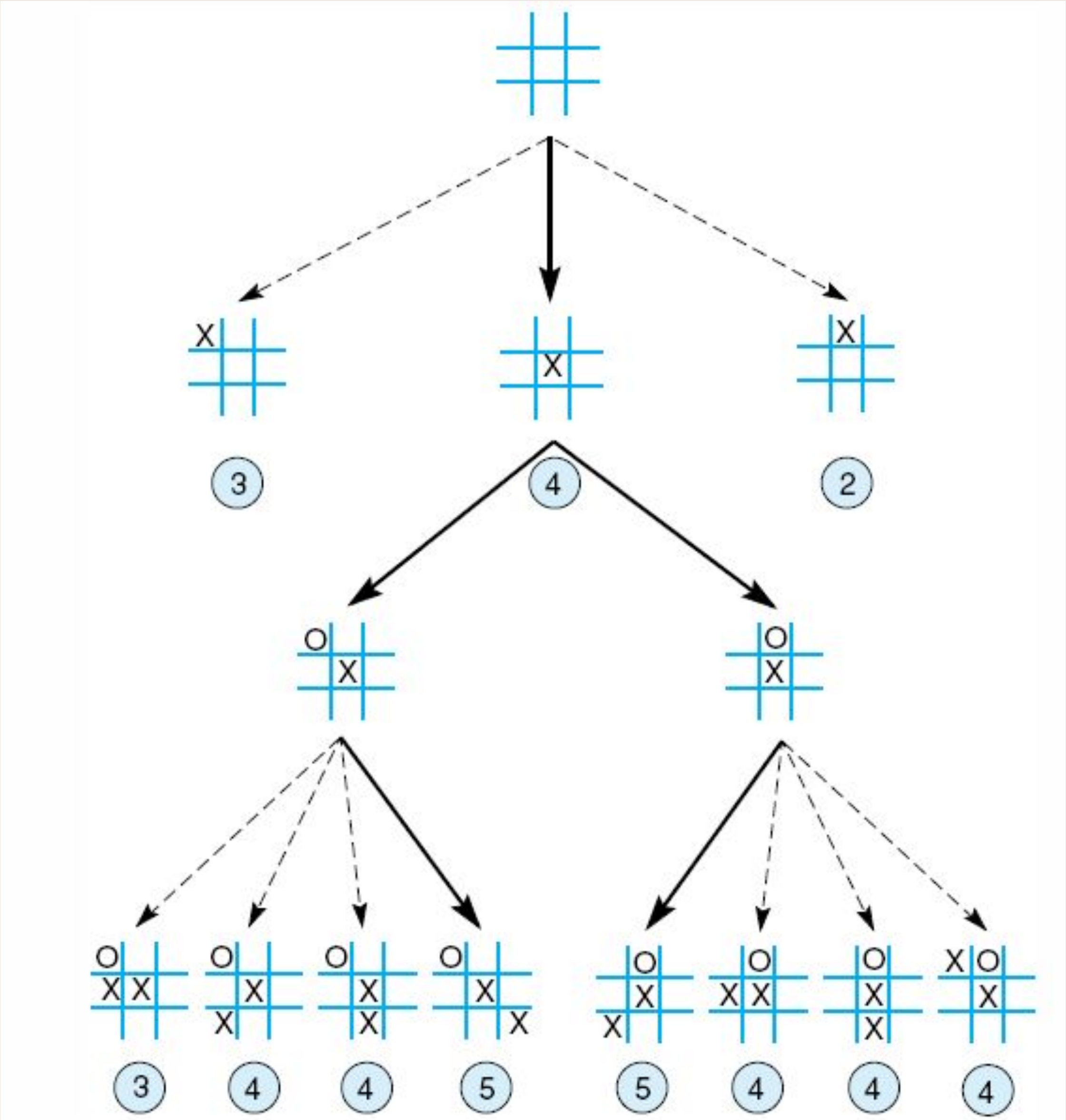


Fig 4.12 The start state, first moves, and goal state for an example-8 puzzle.

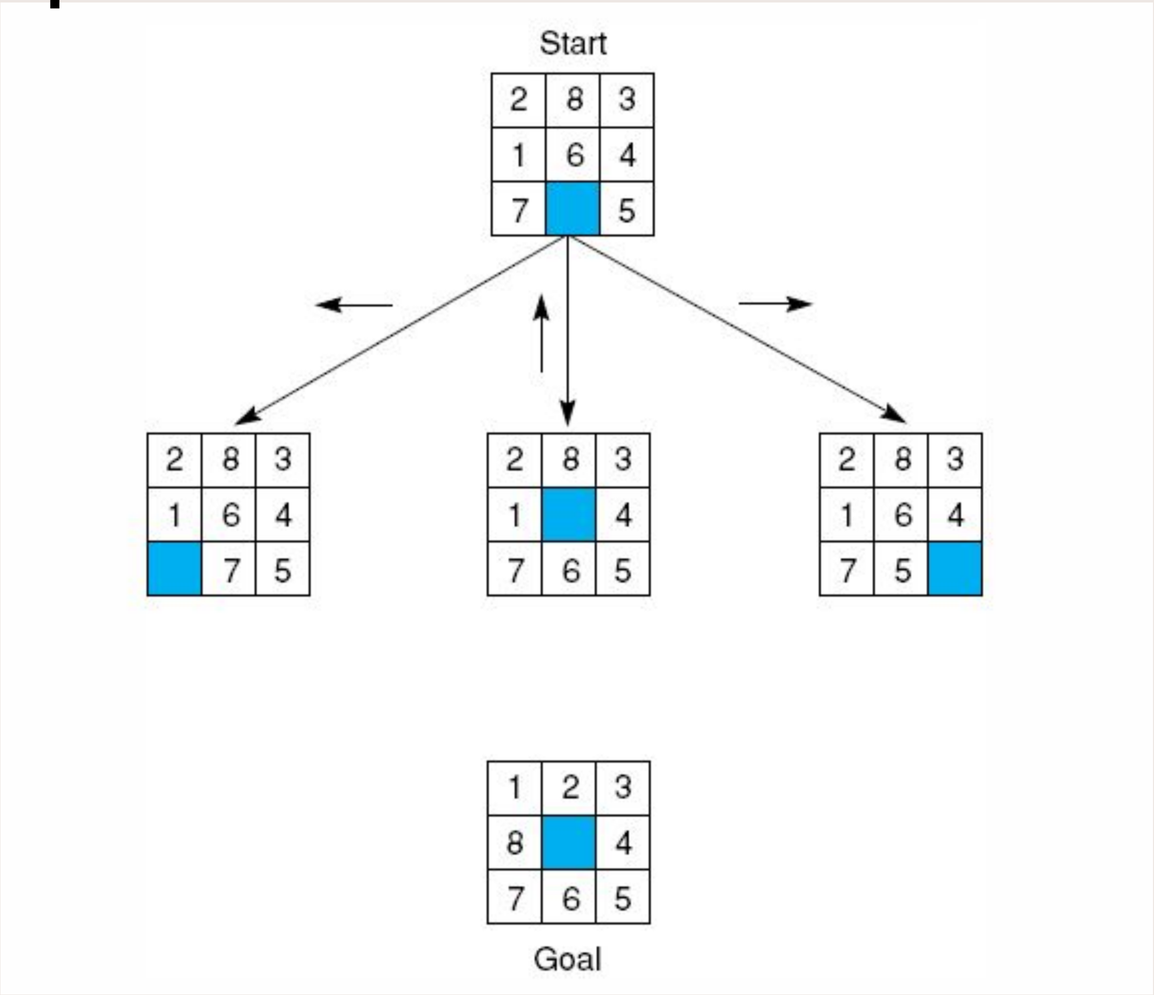


Fig 4.14 Three heuristics applied to states in the 8-puzzle.

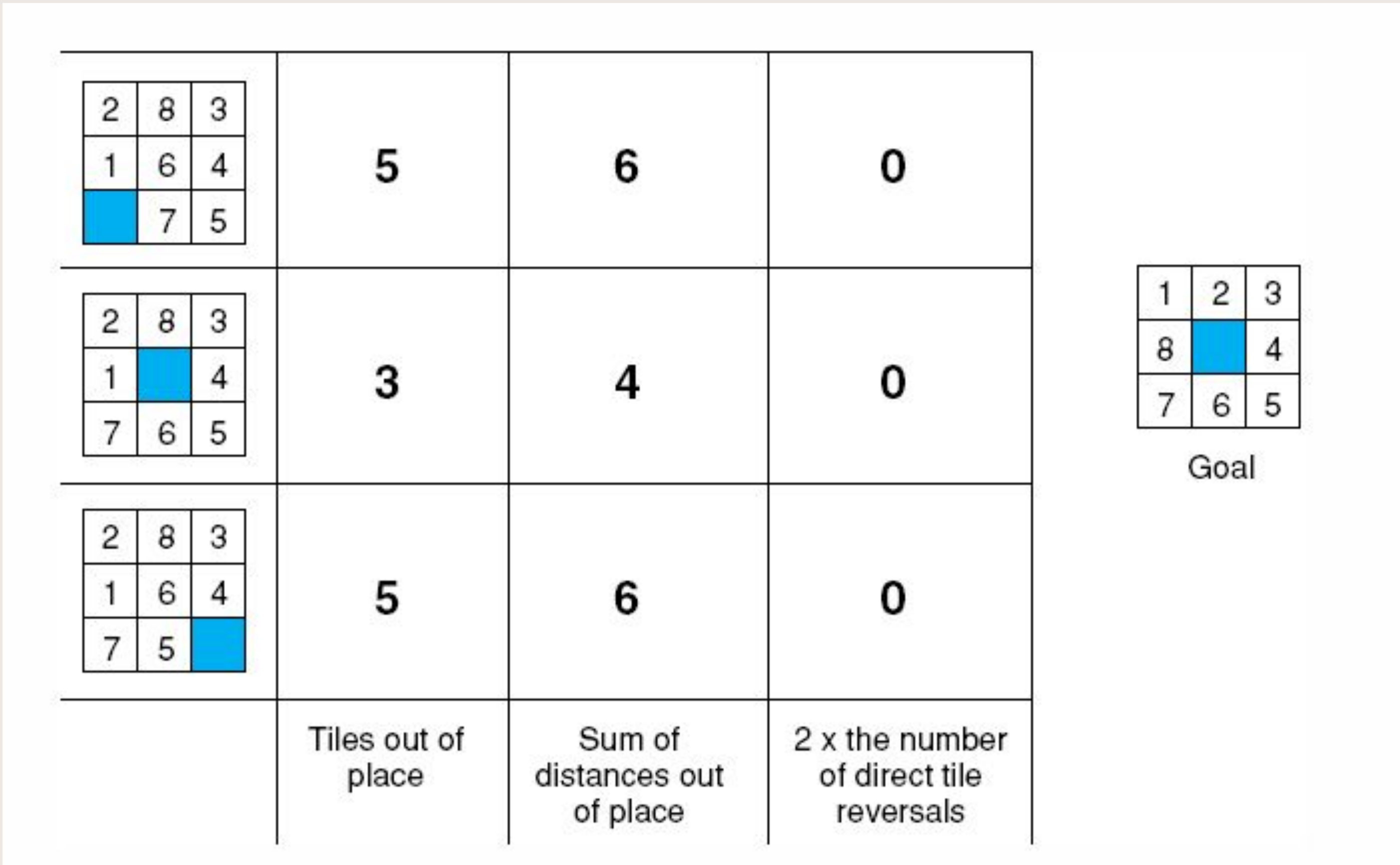


Fig 4.15 The heuristic f applied to states in the 8-puzzle.

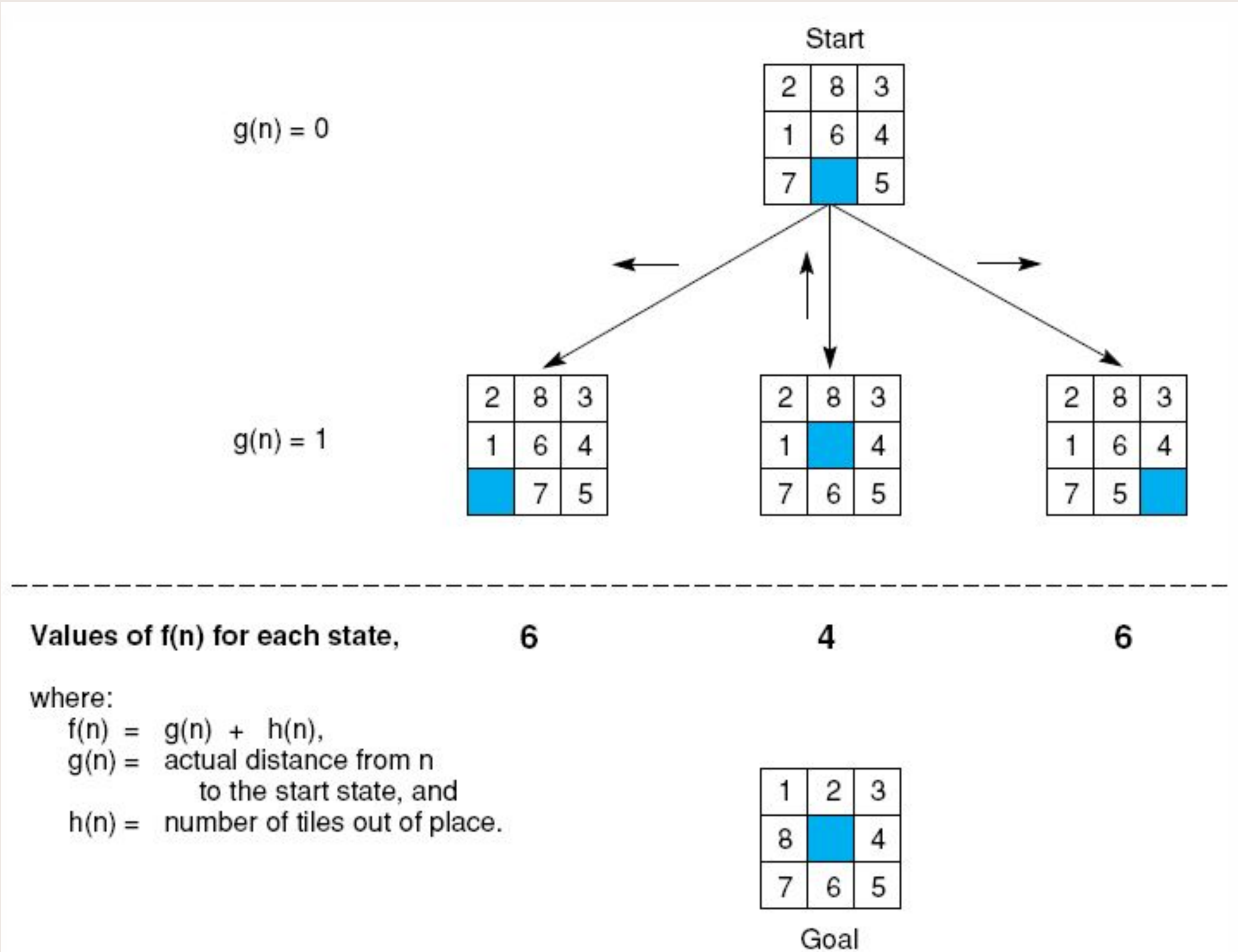


Fig 4.16 State space generated in heuristic search of the 8-puzzle graph.

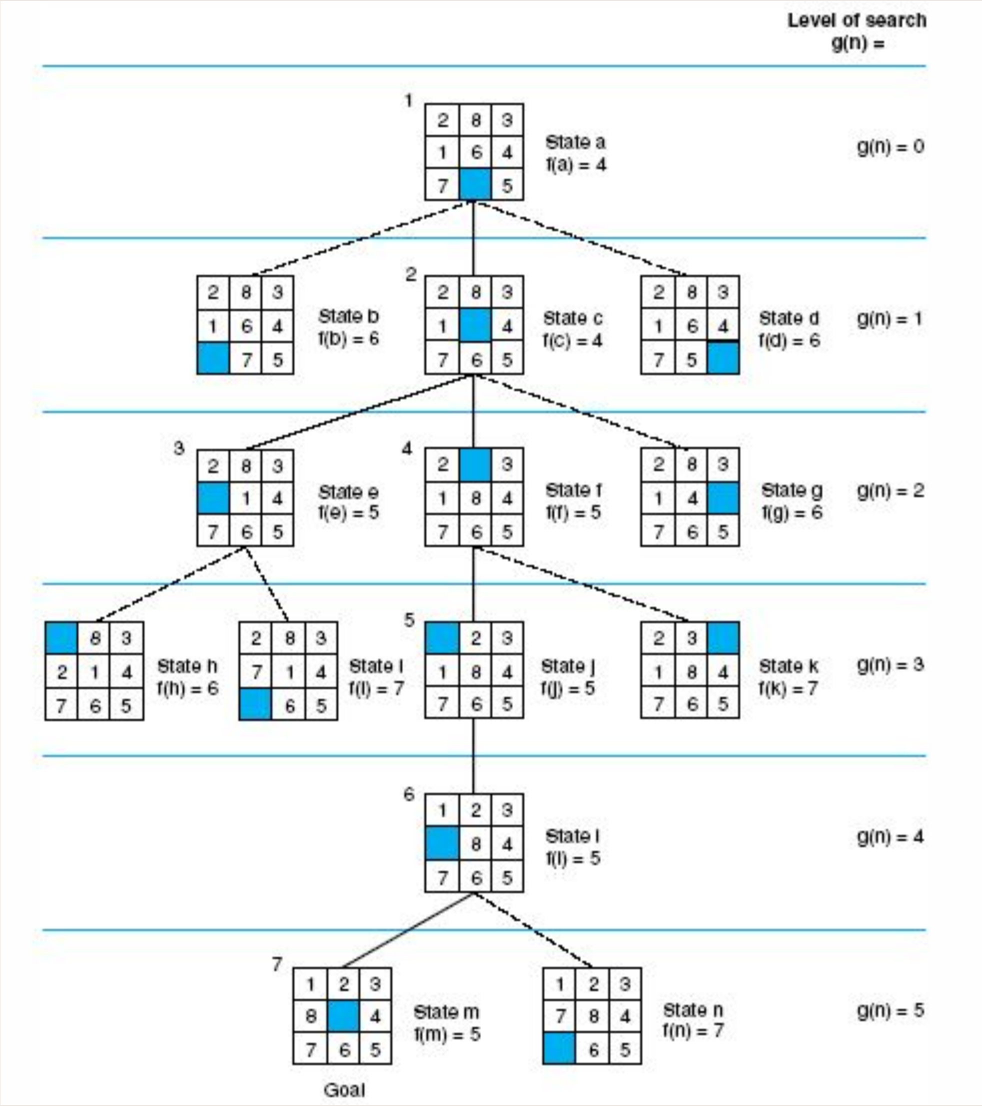
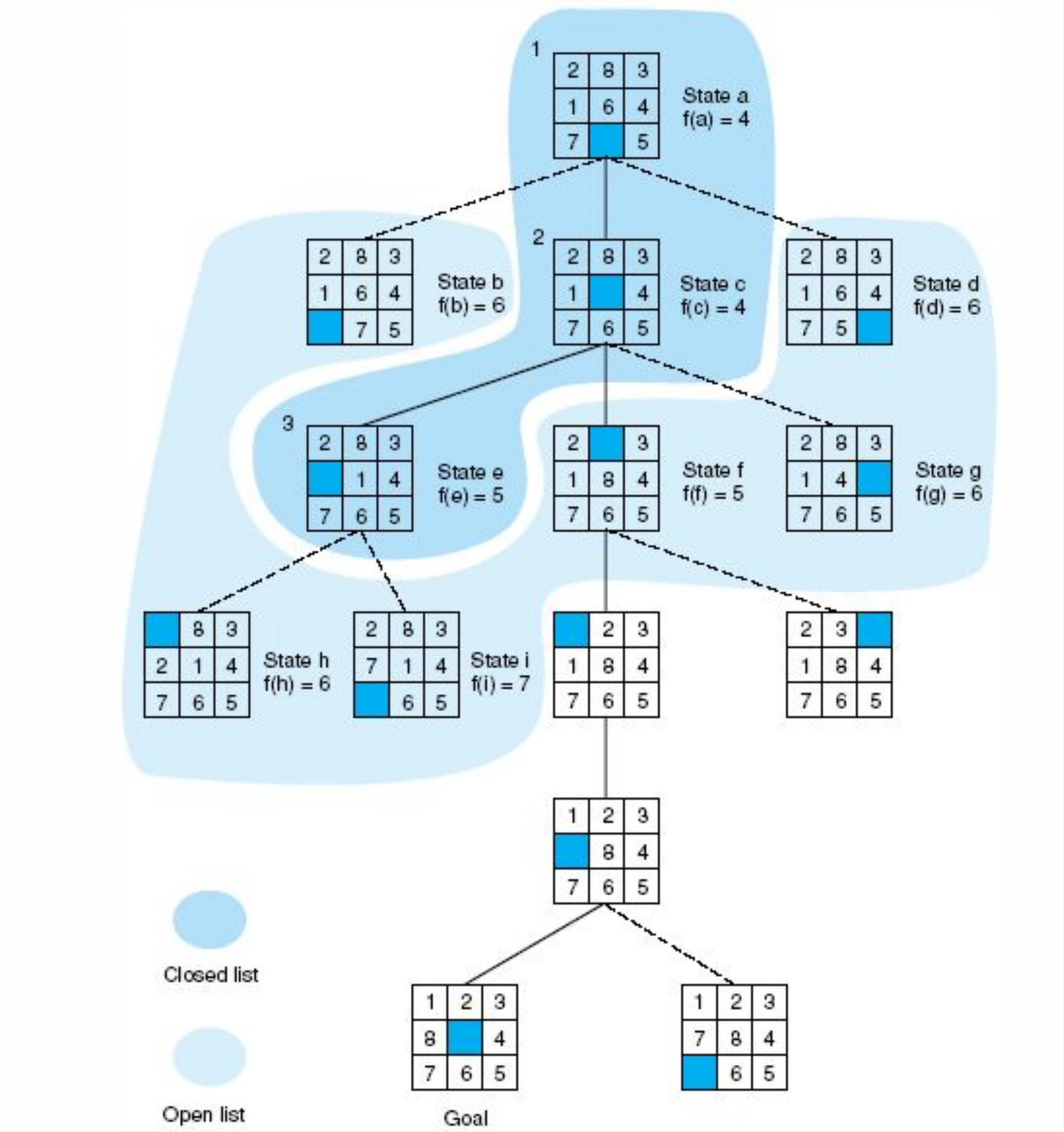


Fig 4.17 Open and closed as they appear after the 3rd iteration of heuristic search



DEFINITION

ALGORITHM A, ADMISSIBILITY, ALGORITHM A*

Consider the evaluation function $f(n) = g(n) + h(n)$, where

> n is any state encountered in the search.
>
> g(n) is the cost of n from the start state.
>
> h(n) is the heuristic estimate of the cost of going from n to a goal.

If this evaluation function is used with the best_first_search algorithm of Section 4.1, the result is called *algorithm A*.

A search algorithm is *admissible* if, for any graph, it always terminates in the optimal solution path whenever a path from the start to a goal state exists.

If algorithm A is used with an evaluation function in which h(n) is less than or equal to the cost of the minimal path from n to the goal, the resulting search algorithm is called *algorithm* A* (pronounced "A STAR").

It is now possible to state a property of A* algorithms:

> All A* algorithms are admissible.

DEFINITION

MONOTONICITY

A heuristic function h is monotone if

1. For all states n_i and n_j, where n_j is a descendant of n_i,

   $h(n_i) - h(n_j) \leq cost(n_i,n_j)$,

   where $cost(n_i,n_j)$ is the actual cost (in number of moves) of going from state n_i to n_j.

2. The heuristic evaluation of the goal state is zero, or $h(Goal) = 0$.

DEFINITION

INFORMEDNESS

For two A* heuristics h_1 and h_2, if $h_1(n) \le h_2(n)$, for all states n in the search space, heuristic h_2 is said to be *more informed* than h_1.

Fig 4.18 Comparison of state space searched using heuristic search with space searched by breadth-first search. The proportion of the graph searched heuristically is shaded. The optimal search selection is in bold. Heuristic used is f(n) = g(n) + h(n) where h(n) is tiles out of place.

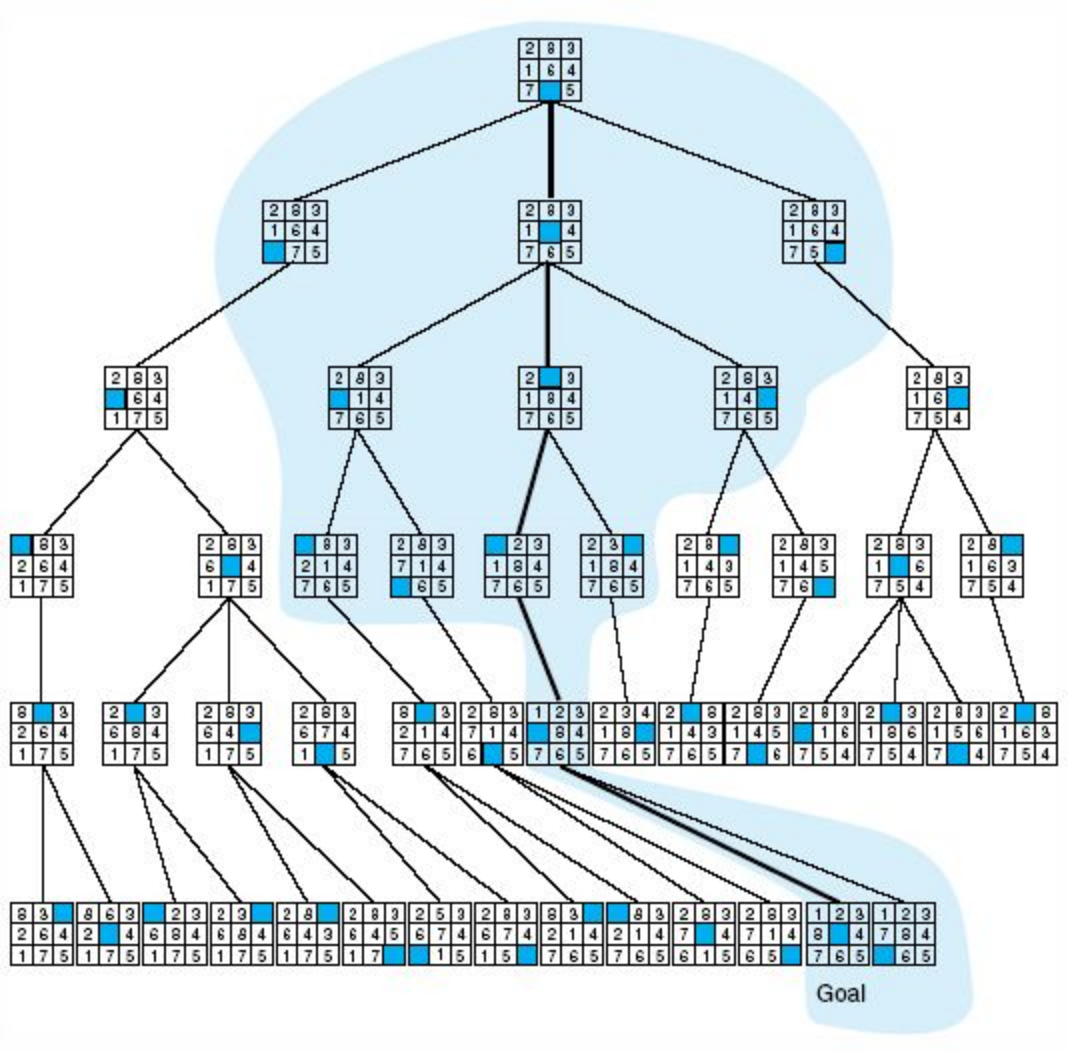


Fig 4.19 State space for a variant of nim. Each state partitions the seven matches into one or more piles.

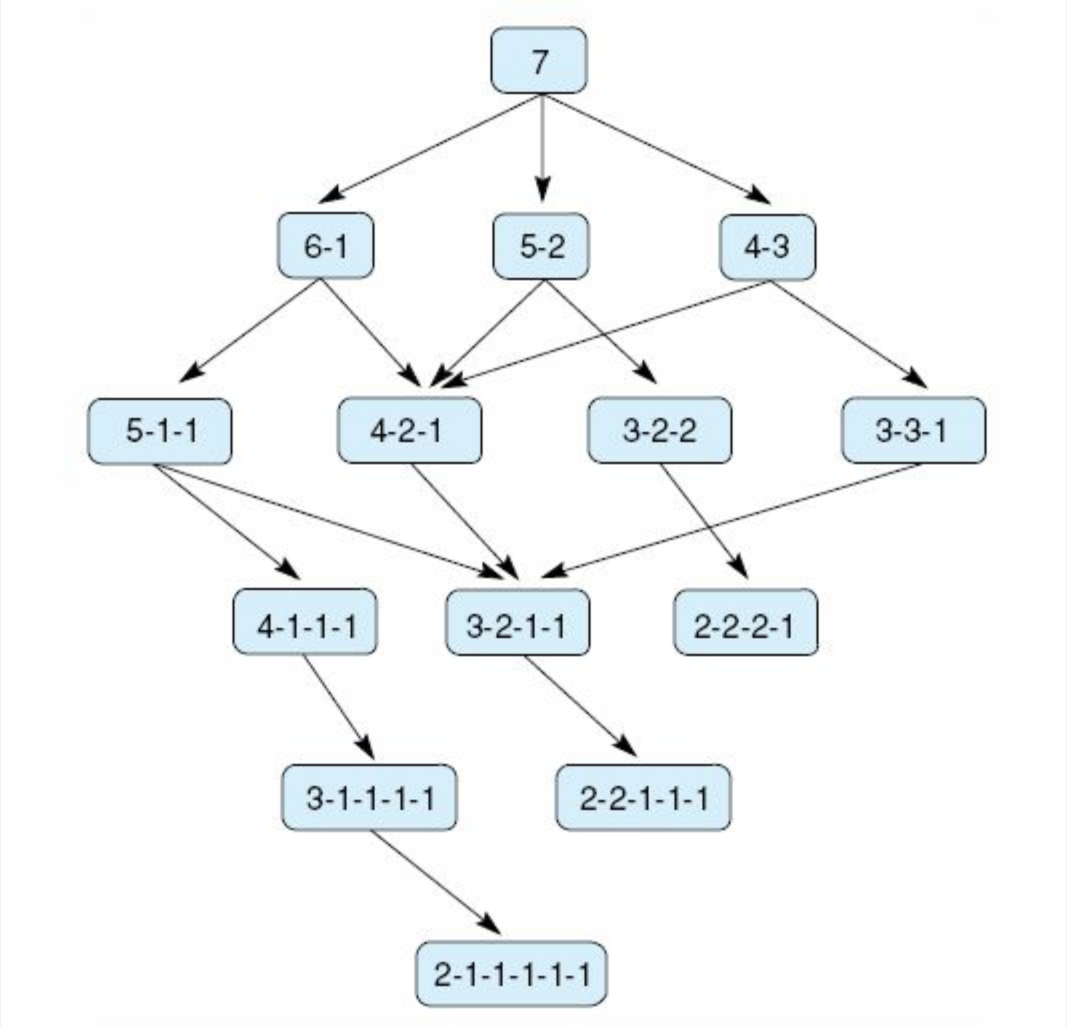


Fig 4.22 Heuristic measuring conflict applied to states of tic-tac-toe.

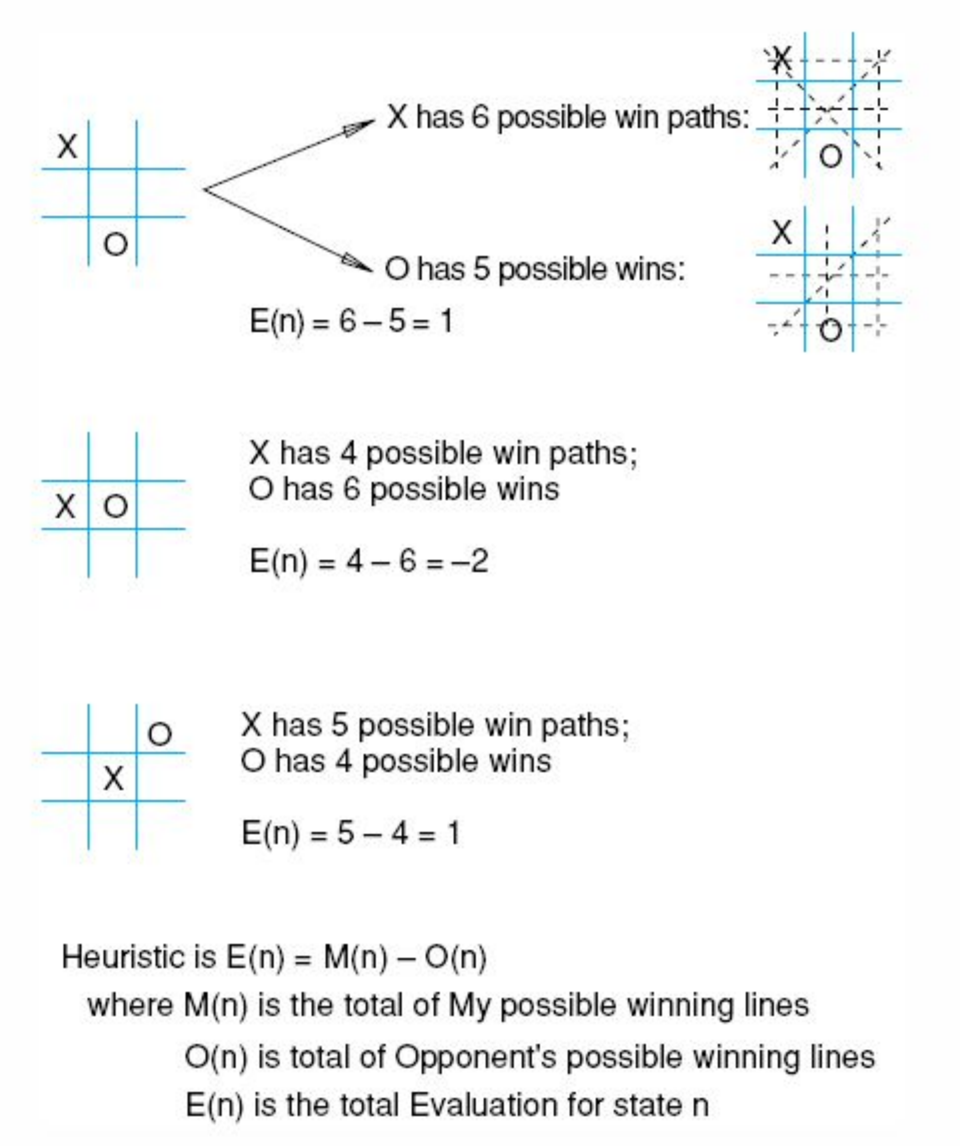


Fig 4.23 Two-ply minimax applied to the opening move of tic-tac-toe, from Nilsson (1971).

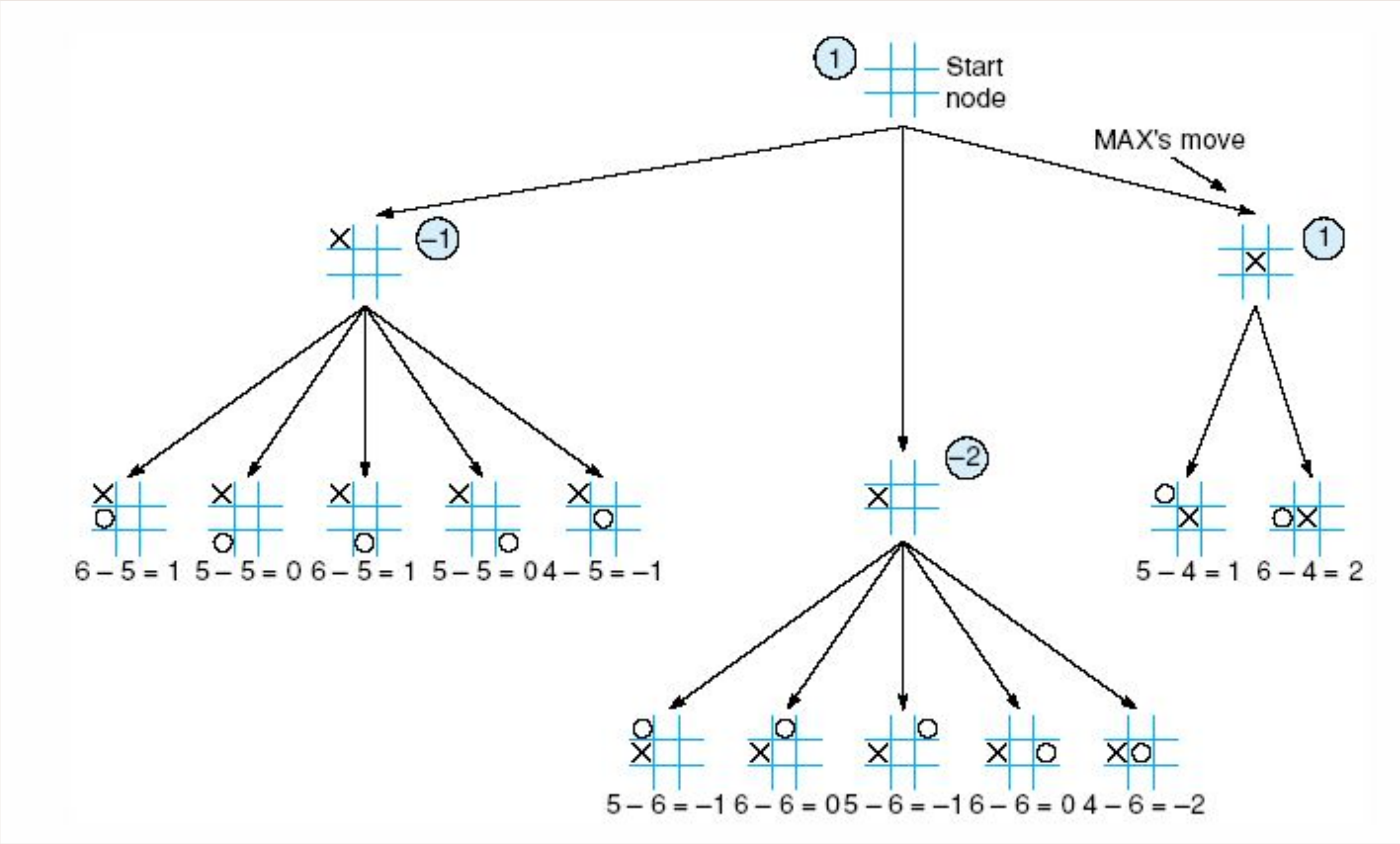


Fig 4.24 Two ply minimax, and one of two possible MAX second moves, from Nilsson (1971).

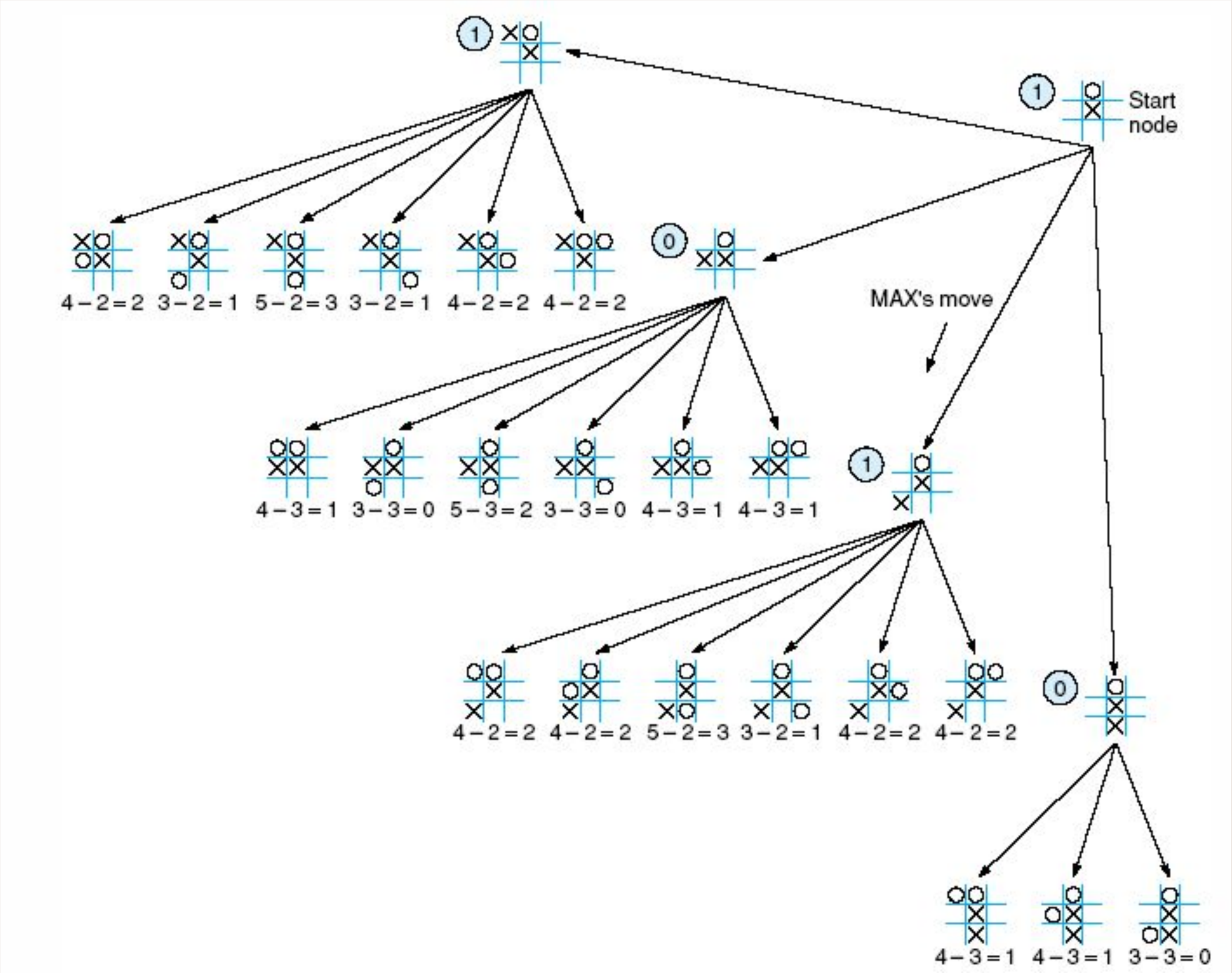


Fig 4.25 Two-ply minimax applied to X's move near the end of the game, from Nilsson (1971).