4 HEURISTIC SEARCH

The successive stages of **open** and **closed** that generate this graph are:

1. **open = [a4];**
   closed = [ ]
2. **open = [c4, b6, d6];**
   closed = [a4]
3. **open = [e5, f5, b6, d6, g6];**
   closed = [a4, c4]
4. **open = [f5, h6, b6, d6, g6, l7];**
   closed = [a4, c4, e5]
5. **open = [j5, h6, b6, d6, g6, k7, l7];**
   closed = [a4, c4, e5, f5]
6. **open = [l5, h6, b6, d6, g6, k7, l7];**
   closed = [a4, c4, e5, f5, j5]
7. **open = [m5, h6, b6, d6, g6, n7, k7, l7];**
   closed = [a4, c4, e5, f5, j5, l5]
8. **success, m = goal!**

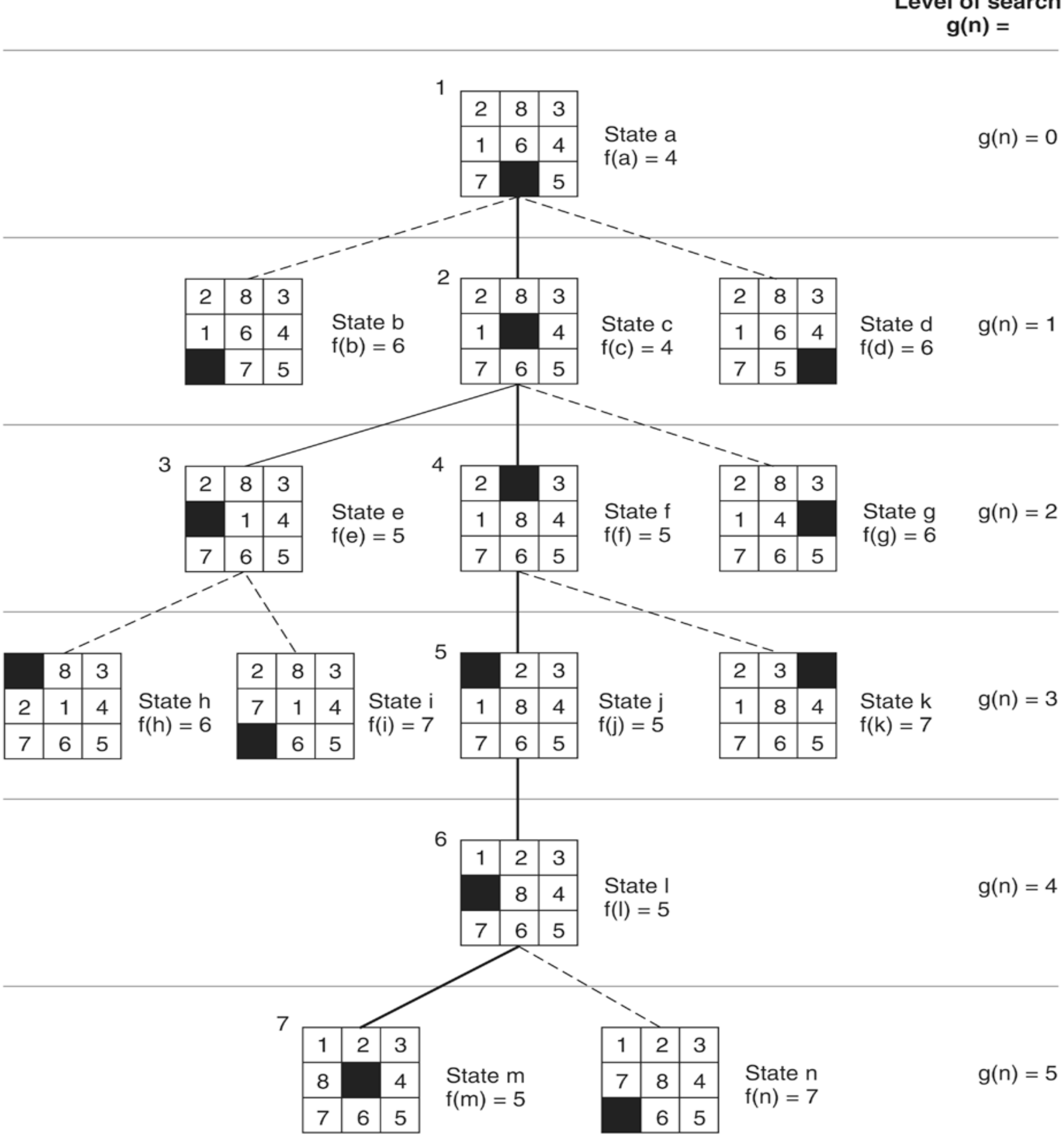


Figure 4.10: State space generated in heuristic search of the 8-puzzle graph.

uger © 2002 Addison Wesley

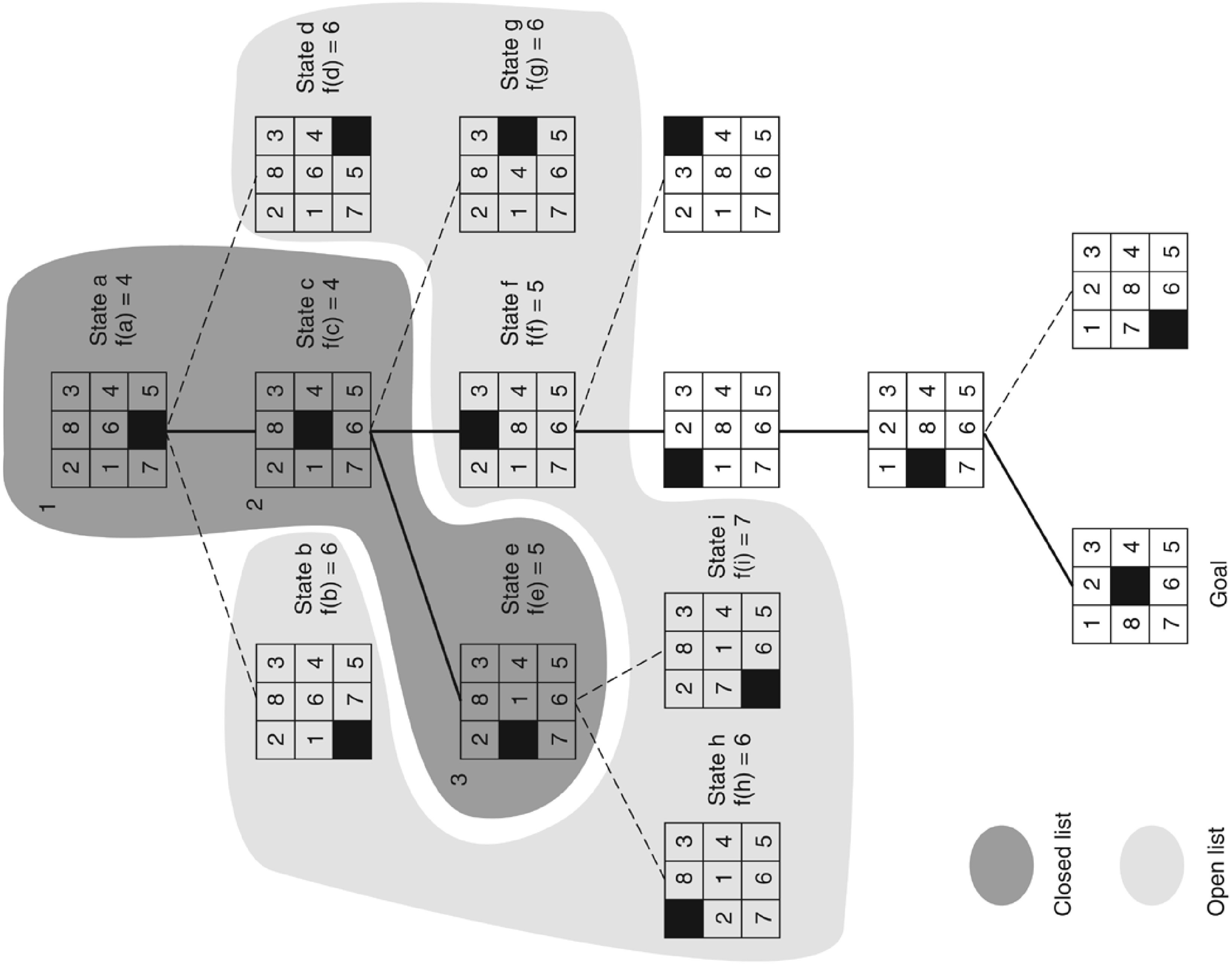


Figure 4.11: **open** and **closed** as they appear after the third iteration of heuristic search.

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 $\mathbf{n}_i$ and $\mathbf{n}_j$, where $\mathbf{n}_j$ is a descendant of $\mathbf{n}_i$,

   $$\mathbf{h}(\mathbf{n}_i) - \mathbf{h}(\mathbf{n}_j) \leq \mathbf{cost}(\mathbf{n}_i,\mathbf{n}_j),$$

   where $\mathbf{cost}(\mathbf{n}_i,\mathbf{n}_j)$ is the actual cost (in number of moves) of going from state $\mathbf{n}_i$ to $\mathbf{n}_j$.

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

DEFINITION

INFORMEDNESS

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

 © 2002 Addison Wesley

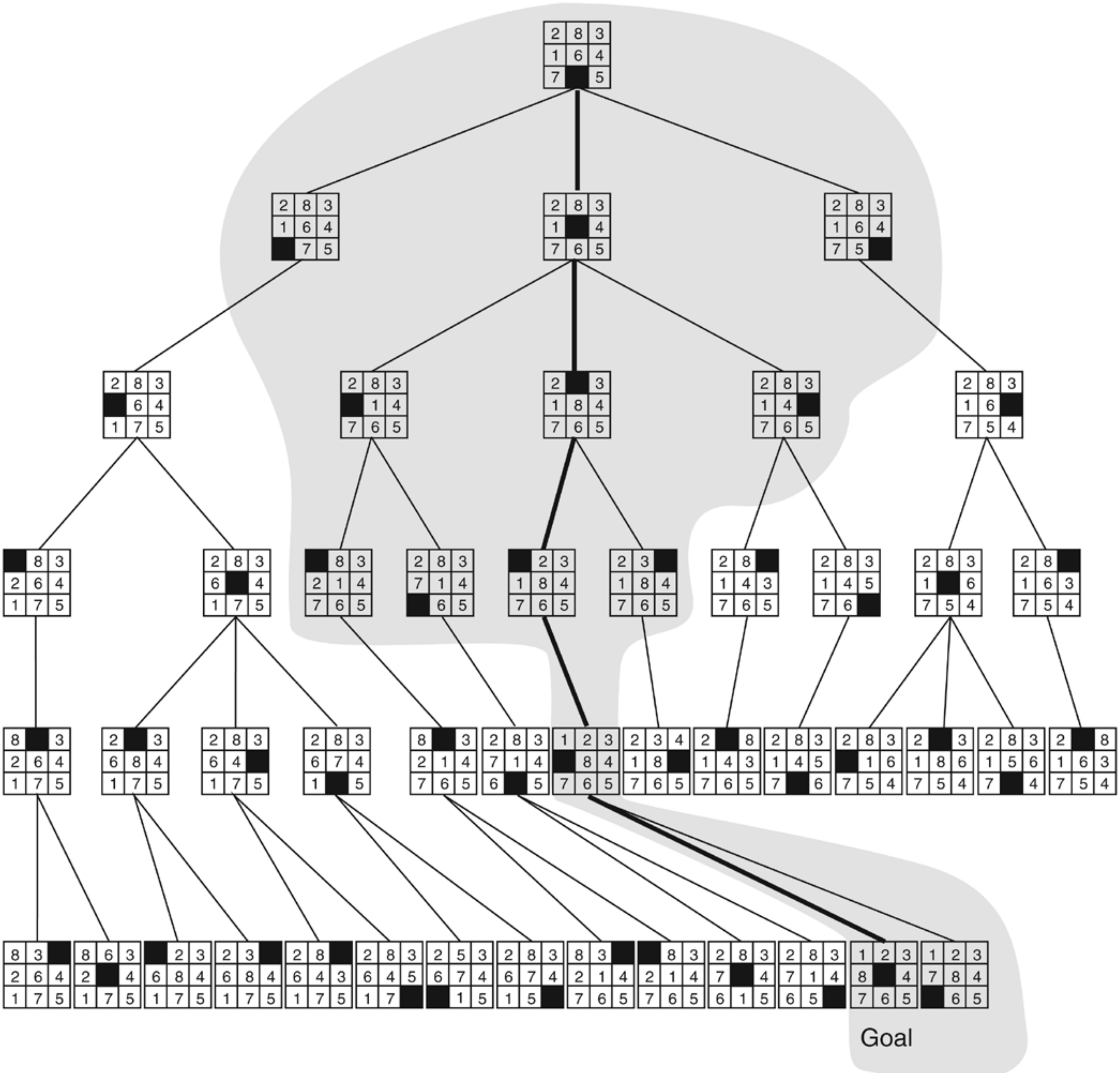


Figure 4.12: Comparison of state space searched using heuristic search with space searched by breadth-first search. The portion of the graph searched heuristically is shaded. The optimal solution path is in bold. Heuristic used is **f(n) = g(n) + h(n)** where **h(n)** is tiles out of place.

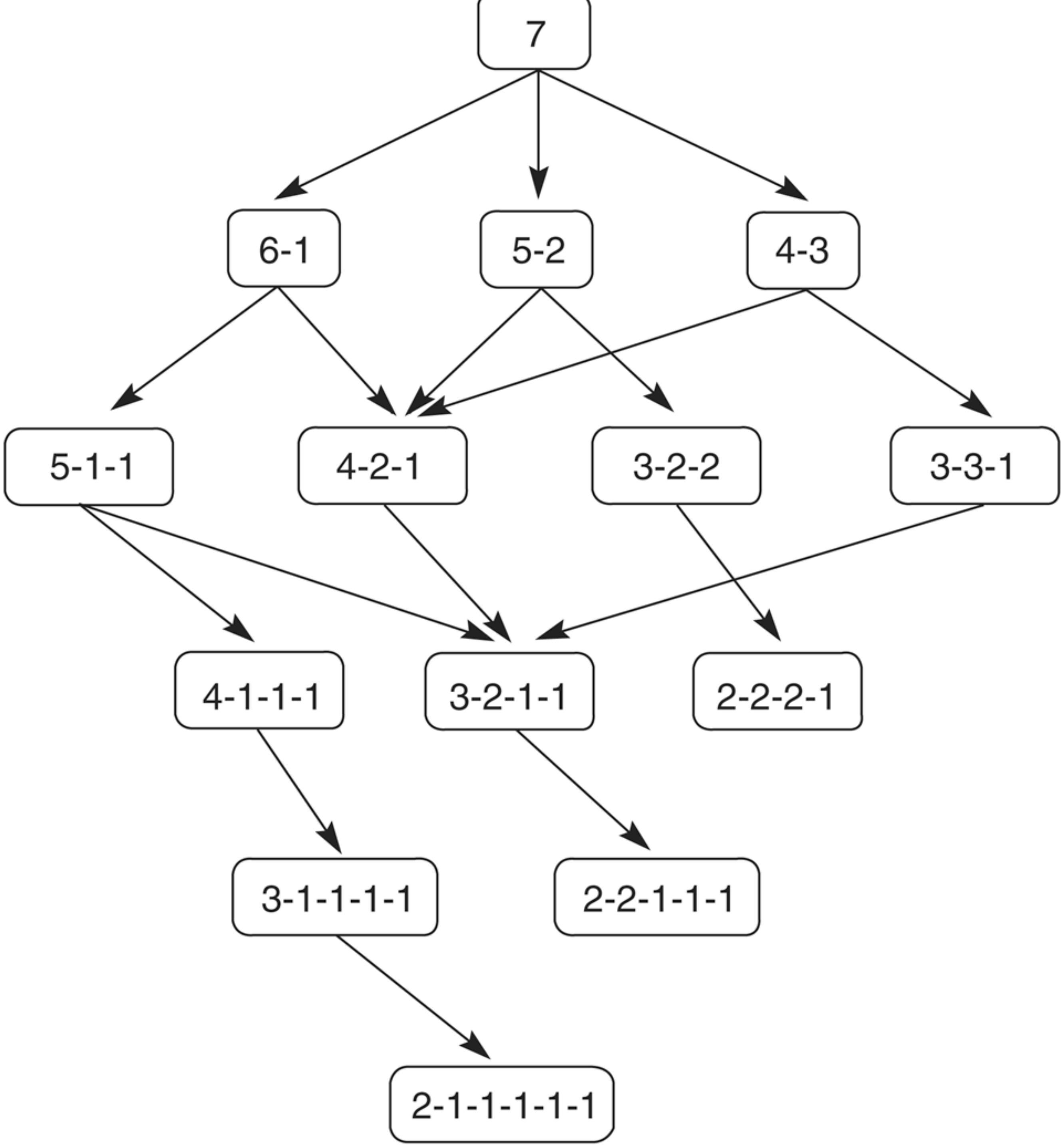


Figure 4.13: State space for a variant of nim. Each state partitions the seven matches into one or more piles.

r © 2002 Addison Wesley

Figure 4.14: Exhaustive minimax for the game of nim. Bold lines indicate forced win for MAX. Each node is marked with its derived value (0 or 1) under minimax.

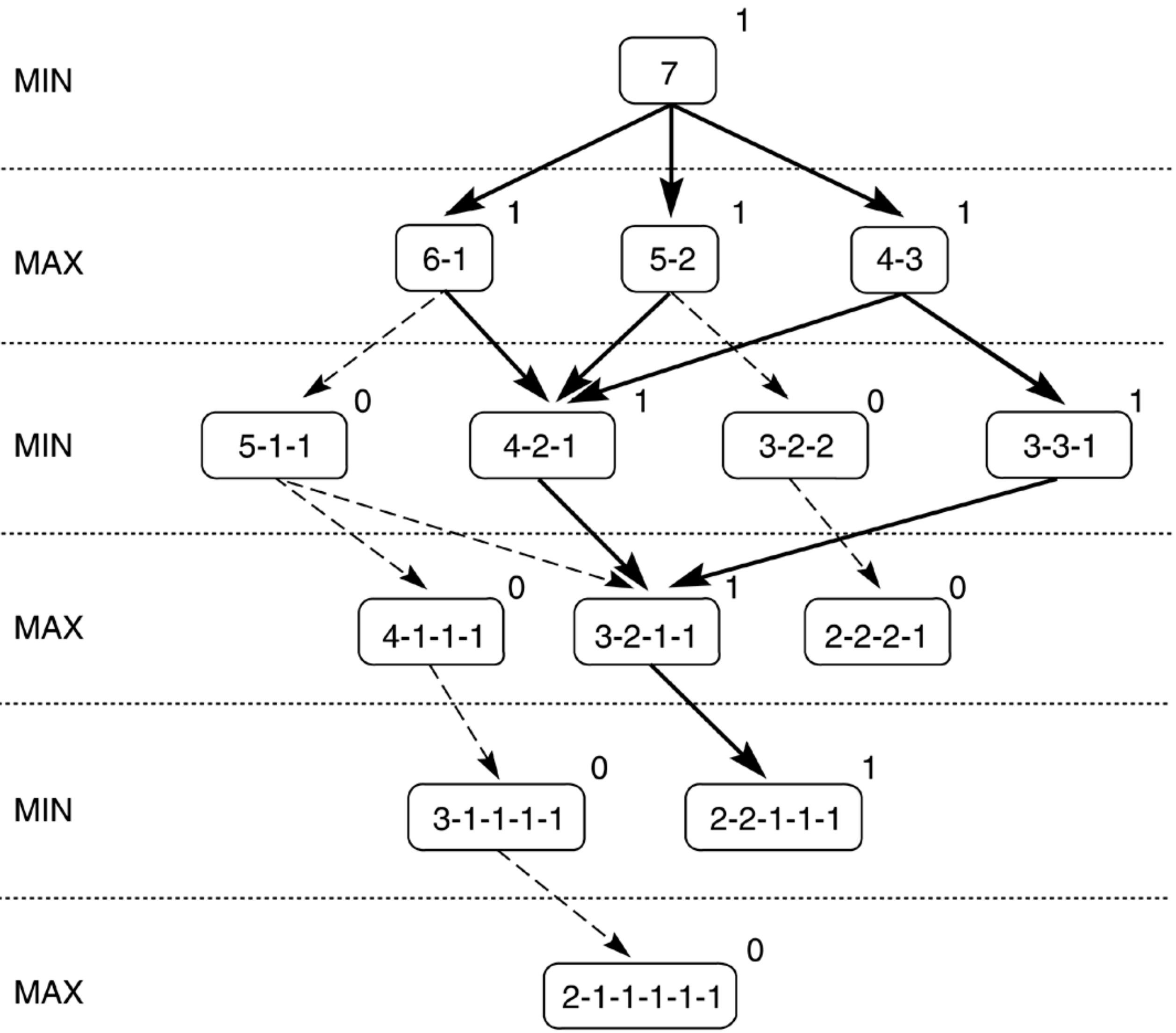
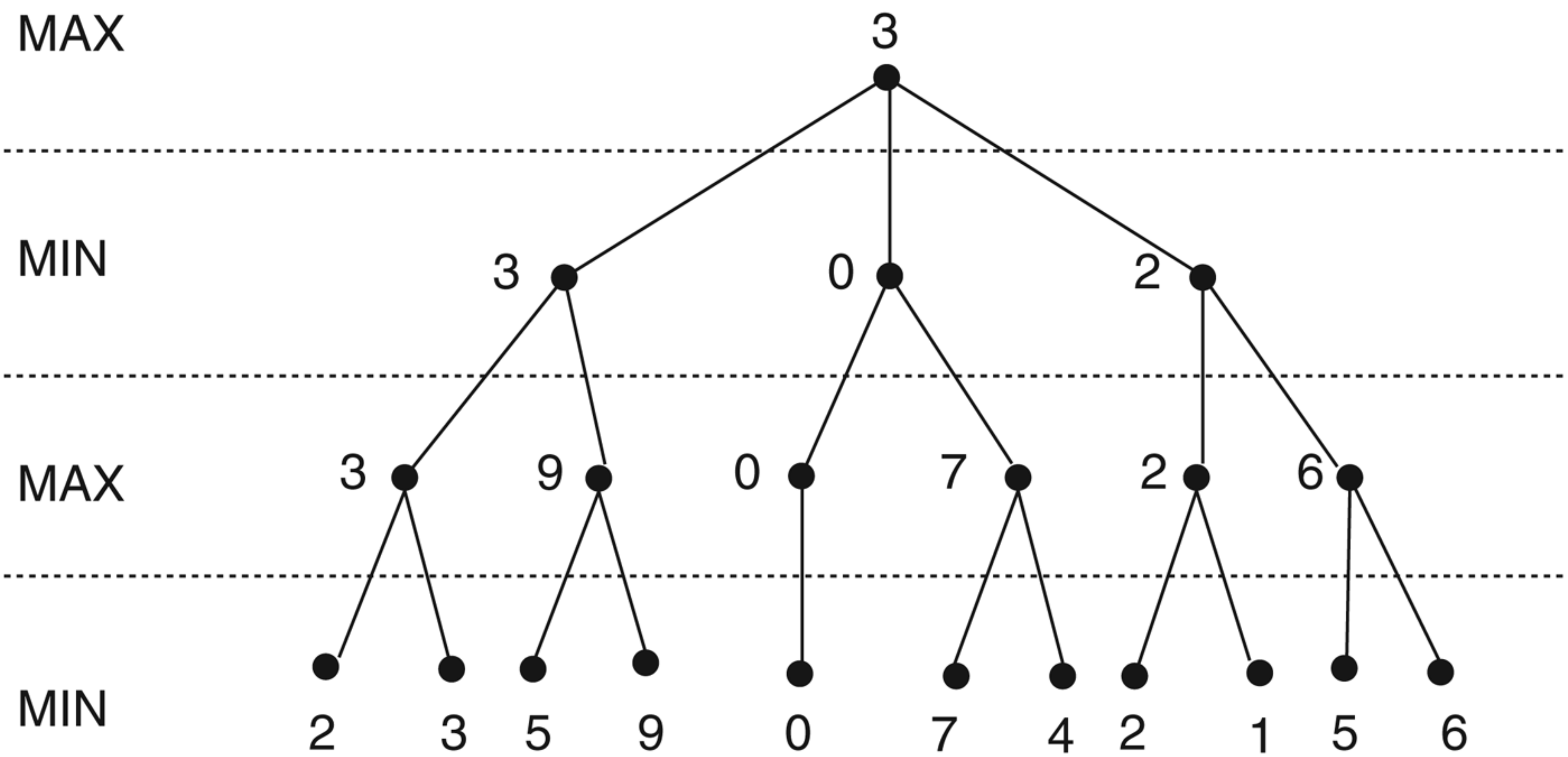
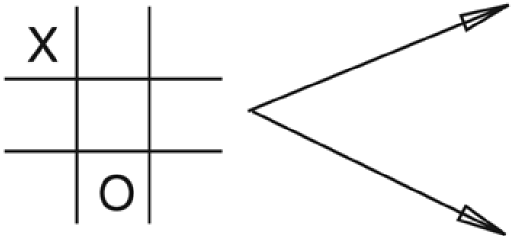


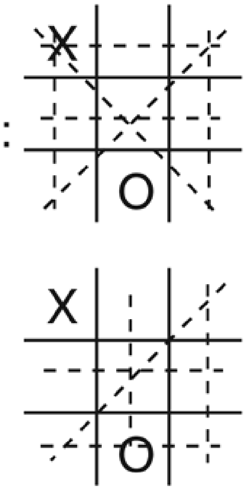
Figure 4.15: Minimax to a hypothetical state space. Leaf states show heuristic values; internal states show backed-up values.



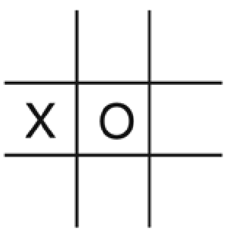


X has 6 possible win paths:

O has 5 possible wins:

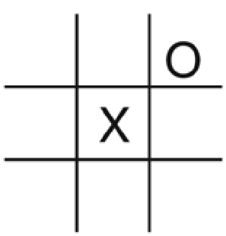


$E(n) = 6 - 5 = 1$



X has 4 possible win paths;
O has 6 possible wins

$E(n) = 4 - 6 = -2$



X has 5 possible win paths;
O has 4 possible wins

$E(n) = 5 - 4 = 1$

Heuristic is $E(n) = M(n) - O(n)$

where $M(n)$ is the total of My possible winning lines

$O(n)$ is total of Opponent's possible winning lines

$E(n)$ is the total Evaluation for state n

Figure 4.16: Heuristic measuring conflict applied to states of tic-tac-toe.

 © 2002 Addison Wesley