Design & Analysis of Algorithms

Session -29

BRANCH AND BOUND -- THE METHOD

The term branch-and-bound refers to all state space search methods in which all children of the E-node are generated before any other live node can become the E-node.

In branch-and-bound terminology breadth first search(***BFS***)- like state space search will be called FIFO (First In First Output) search as the list of live nodes is a first-in-first-out list (or queue).

A ***D-search*** (depth search) state space search will be called LIFO (Last In First Out) search, as the list of live nodes is a last-in-first-out list (or stack).
Bounding functions are used to help avoid the generation of sub trees that do not contain an answer node.

Least Cost (LC) Search:

•In both LIFO and FIFO branch-and-bound the selection rule for the next E-node is rather rigid.

• The selection rule for the next E-node does not give any preference to a node that has a very good chance of becoming an answer node quickly.

•The search for an answer node can be speeded by using an "intelligent" ranking function for live nodes.

•The next E-node is selected on the basis of this ranking function.

•The ideal way to assign ranks would be on the basis of the additional computational effort (or cost) needed to reach an answer node from the live node.

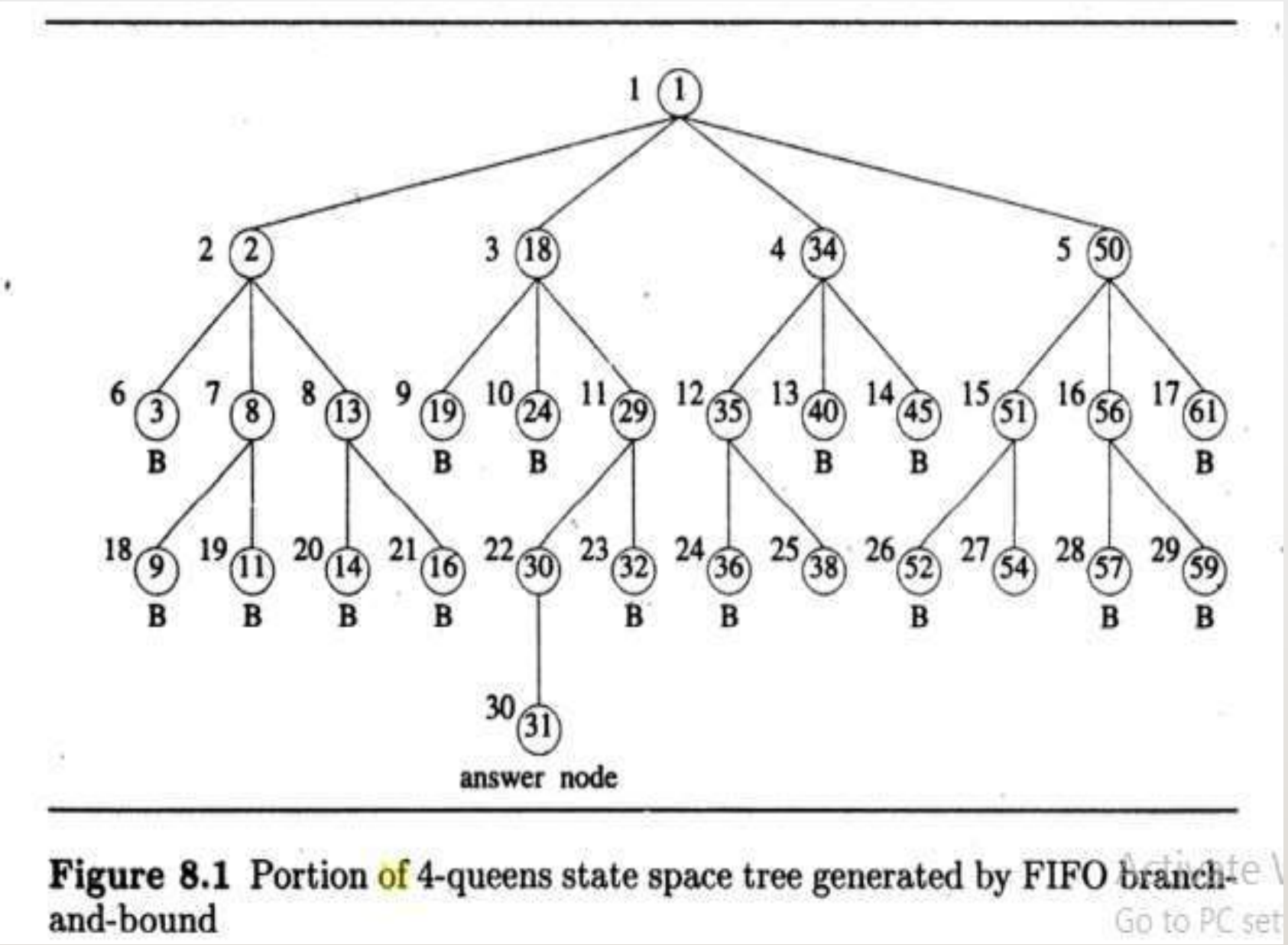


Figure 8.1 Portion of 4-queens state space tree generated by FIFO branch-and-bound

•For any node x, this cost could be
(1) the number of nodes in the subtree x that need to be generated before an answer node is generated or
(2) the number of levels the nearest answer node (in the subtree) is from x.
•Hence, by the time the cost of a node is determined, that subtree has been searched and there is no need to explore x again.
• So search algorithms usually rank nodes only on the basis of an estimate g(.) of their cost.
•Node x is assigned a rank using a function c(•) such that c(x) = f(h(x)) + g(x), where h(x) is the cost of reaching x from the root and f (.) is any nondecreasing function.
• Let g(x) be an estimate of the additional effort needed to reach an answer node from x.

•A search strategy that uses a cost function c(x)= f(h(x))+ g(x) to select the next E-node, would always choose for its next E-node a live node with least cost. Hence, such a search strategy is called an LC-search (Least Cost search).

• BFS and D-search are the special cases of LC-search.

• If g(x)=0 and f(h(x)) = level of node x, then a LC-search becomes BFS.

•If f (h(x))=0 and g(x) > g(y) whenever y is a child of x, then LC search is essentially a D-search.

•An LC-search coupled with bounding functions is called an LC branch-and-bound search.

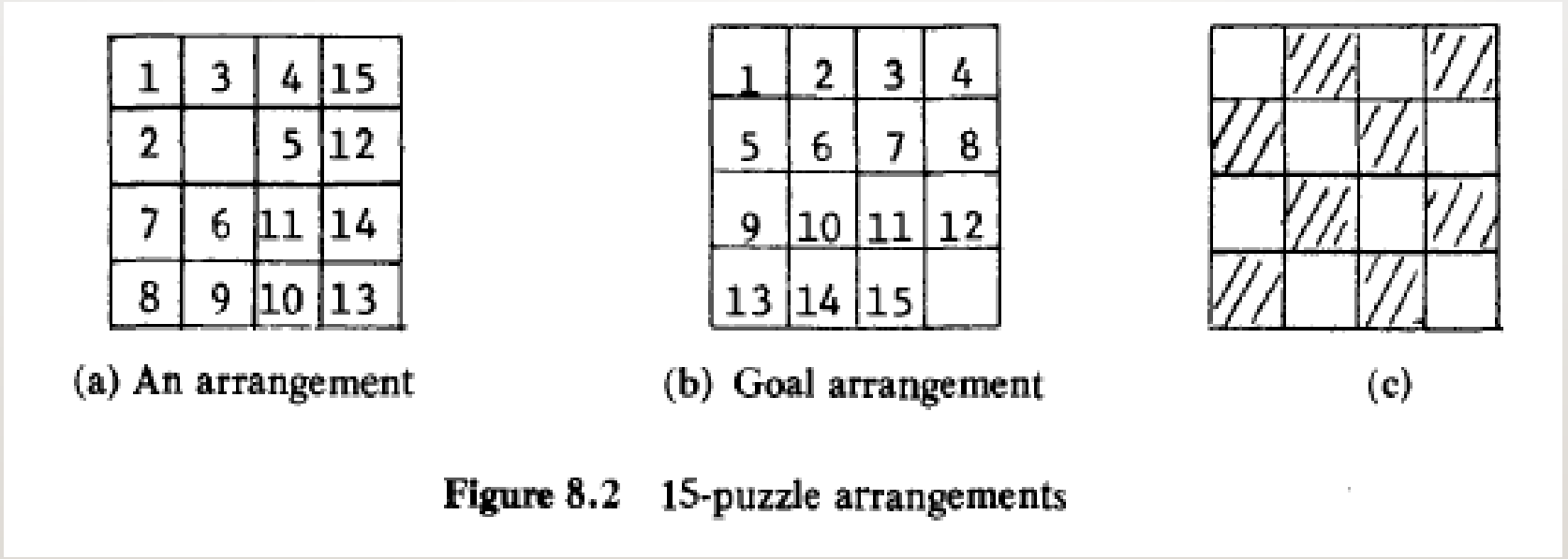
•Generating state space tree using FIFO search with bounding functions is called FIFOBB

•Generating state space tree using LIFO search with bounding functions is called LIFOBB

The 15-puzzle-An Example

The 15-puzzle (invented by Sam Loyd in 1878) consists of 15 numbered tiles on a square frame with a capacity of 16 tiles .

We are given an initial arrangement of the tiles and the objective is to transform this arrangement into the goal arrangement of Figure 8.3(b) through a series of legal moves.



(a) An arrangement (b) Goal arrangement (c)

Figure 8.2 15-puzzle arrangements

The only legal moves are ones in which a tile adjacent to the empty spot (ES) is moved to ES.

Thus from the initial arrangement of Figure 8.2(a), four moves are possible. We can move any one of the tiles numbered 2, 3, 5 or 6 to the empty spot.

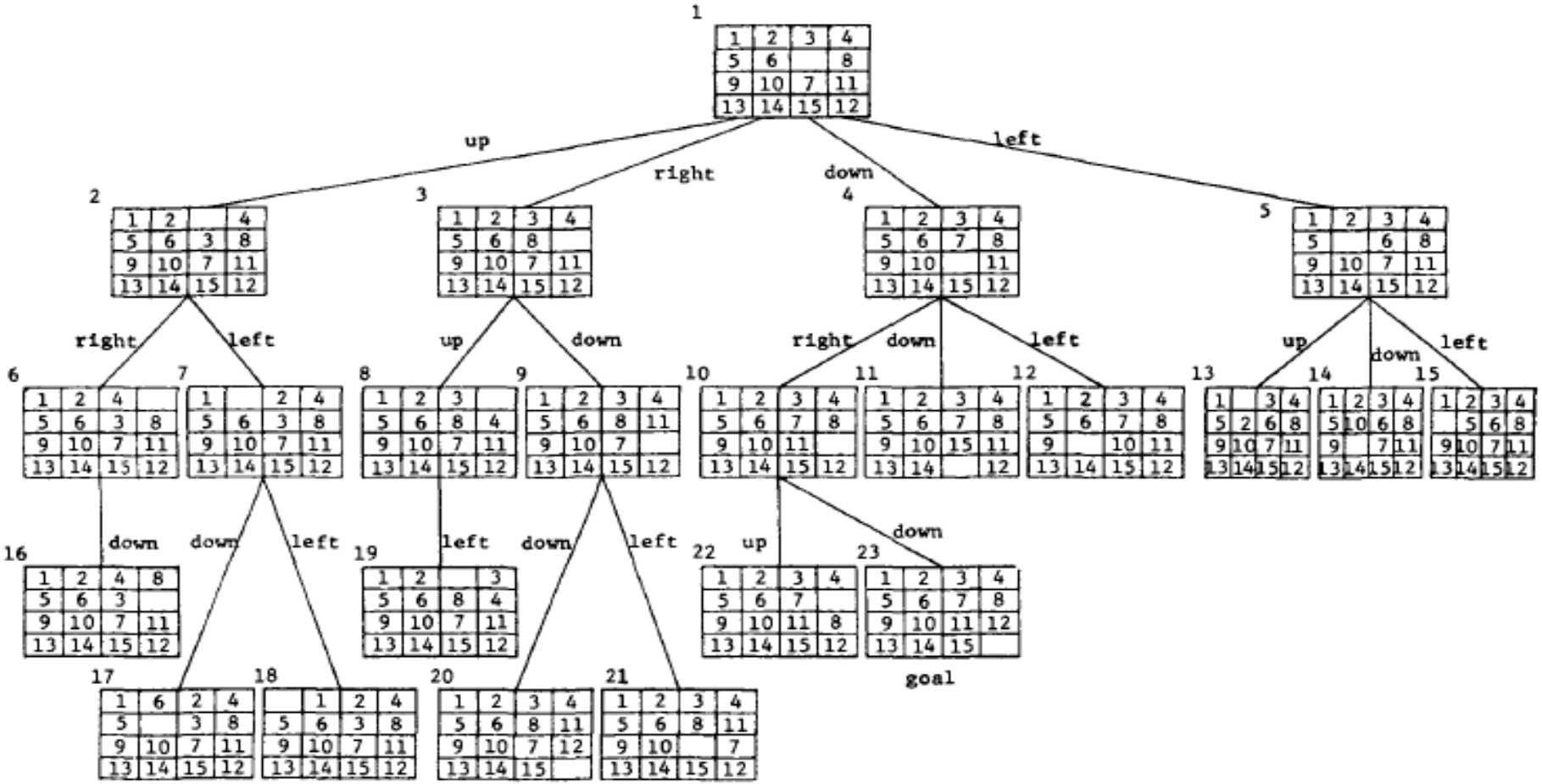
Following this move, other moves can be made. Each move creates a new arrangement of the tiles.

These arrangements will be called the *states of the puzzle.*

The initial and goal arrangements are called the initial and goal states.

A state is reachable from the initial state iff there is a sequence of legal moves from the initial state to this state.

Theorem 8.1 The goal state of Figure 8.2(b) is reachable from the initial state iff $\sum_{i=1}^{16} \text{LESS}(i) + X$ is even.



edges are labeled according to the direction in which the empty space moves

Figure 8.3(a) Part of the state space tree for the 15-puzzle

Control Abstraction for LC-Search

```
1   Algorithm LCSearch(t)
2   // Search t for an answer node.
3   {
4       if *t is an answer node then output *t and return;
5       E := t; // E-node.
6       Initialize the list of live nodes to be empty;
7       repeat
8       {
9           for each child x of E do
10          {
11              if x is an answer node then output the path
12                  from x to t and return;
13              Add(x); // x is a new live node.
14              (x → parent) := E; // Pointer for path to root.
15          }
16          if there are no more live nodes then
17          {
18              write ("No answer node"); return;
19          }
20          E := Least();
21      } until (false);
22  }
```

SAMPLE QUESTIONS

1. List and Explain different types of Branch and Bound Techniques.
2. Explain how FIFOBB is different from LIFOBB?
3. With suitable example explain Control Abstraction for LC-Search?
4. Explain the terminology used in Branch and Bound?