Uninformed Search Strategies

PoAI

School of Electronic and Computer Engineering
Peking University

Wang Wenmin

Contents

□ 3.4.4 Depth-limited Search

□ 3.4.4 Iterative Deepening Search

1) Depth-limited Search 深度受限搜索

- □ The failure of depth-first search will be happened if in infinite state spaces.

  若状态空间无限，深度优先搜索就会发生失败。

- □ This problem can be solved with a predetermined depth limit l, i.e. nodes at depth l are treated as if they have no successors.

  这个问题可以用一个预定的深度限制 /得到解决，即：深度 /以外的节点被视为没有后继节点。

- □ Disadvantages

  缺点

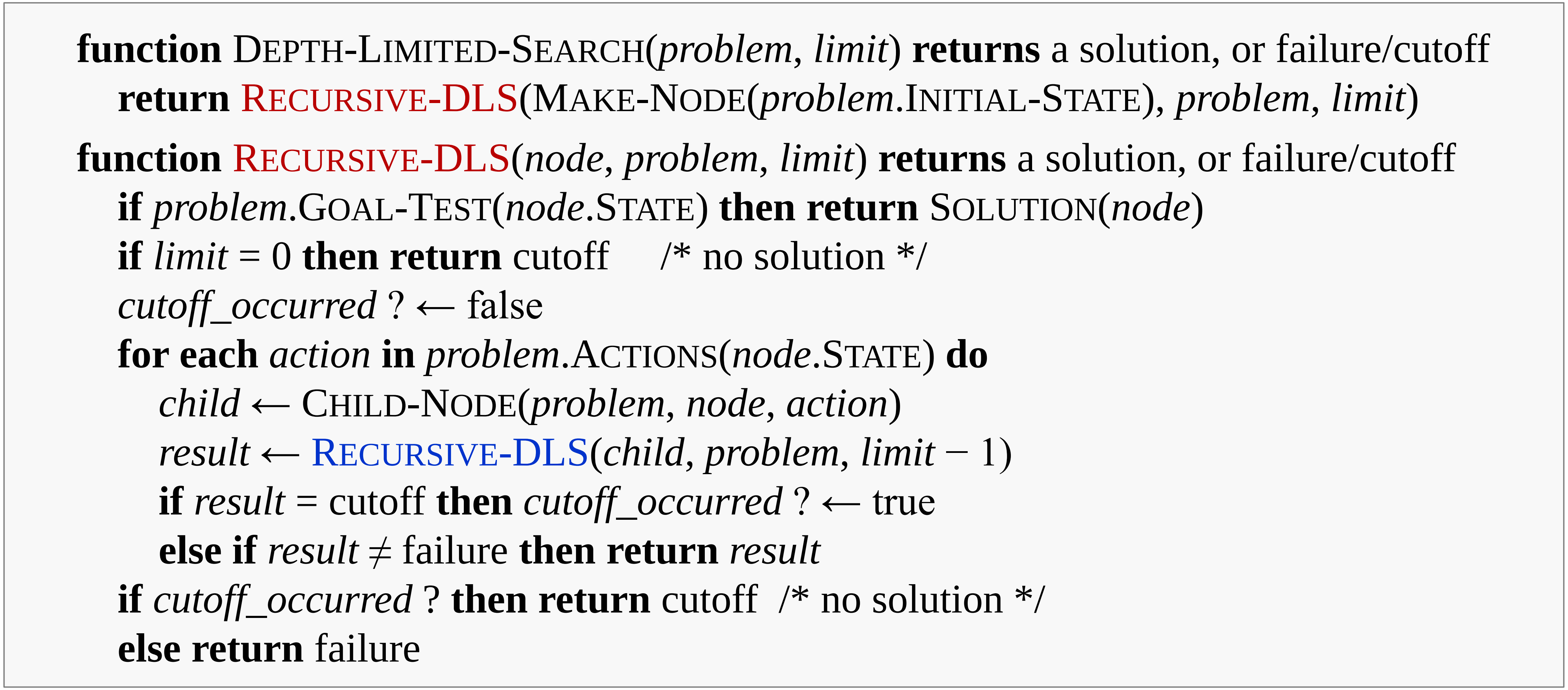
  - ■ It will introduces an additional source of incompleteness if we choose $l < d$, that is, the shallowest goal is beyond the depth limit.

    如果我们选择 $l < d$，即最浅的目标在深度限制之外，这种方法就会出现额外的不完备性。

  - ■ Depth-limited search will also be non-optimal if we choose $l > d$.

    如果我们选择 $l > d$，深度受限搜索也将是非最优的。

Depth-limited Search Algorithm 深度受限搜索算法



```
function DEPTH-LIMITED-SEARCH(problem, limit) returns a solution, or failure/cutoff
    return RECURSIVE-DLS(MAKE-NODE(problem.INITIAL-STATE), problem, limit)

function RECURSIVE-DLS(node, problem, limit) returns a solution, or failure/cutoff
    if problem.GOAL-TEST(node.STATE) then return SOLUTION(node)
    if limit = 0 then return cutoff     /* no solution */
    cutoff_occurred ? ← false
    for each action in problem.ACTIONS(node.STATE) do
        child ← CHILD-NODE(problem, node, action)
        result ← RECURSIVE-DLS(child, problem, limit − 1)
        if result = cutoff then cutoff_occurred ? ← true
        else if result ≠ failure then return result
    if cutoff_occurred ? then return cutoff  /* no solution */
    else return failure
```

A recursive implementation of depth-limited tree search

2) Iterative Deepening Search 迭代加深搜索

- □ It combines the benefits of depth-first and breadth-first search, running repeatedly with gradually increasing depth limits until the goal is found.

  它将深度优先和宽度优先的优势相结合，逐步增加深度限制反复运行直到找到目标。

- □ It visits the nodes in the search tree in the same order as depth-first search, but the cumulative order in which nodes are first visited is effectively breadth-first.

  它以深度优先搜索相同的顺序访问搜索树的节点，但先访问节点的累积顺序实际是宽度优先。

```
function ITERATIVE-DEEPENING-SEARCH(problem) returns a solution, or failure
    for depth = 0 to $\infty$ do
        result $\leftarrow$ DEPTH-LIMITED-SEARCH(problem, depth)
        if result $\neq$ cutoff then return result
```

It repeatedly applies *depth* limited search with increasing limits, in which it calls DEPTH-LIMITED-SEARCH algorithm.

Thank you for your attention!

PoAI