

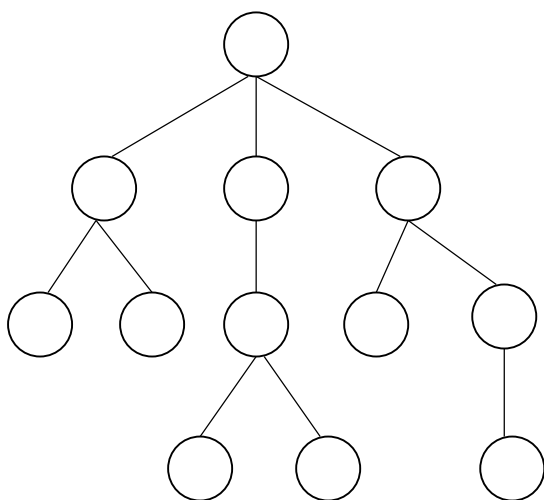
第4講 探索手法(1)

- 探索とは
- 状態空間の探索
- 状態空間の表現
- 探索手法の分類
- コストを考えない探索

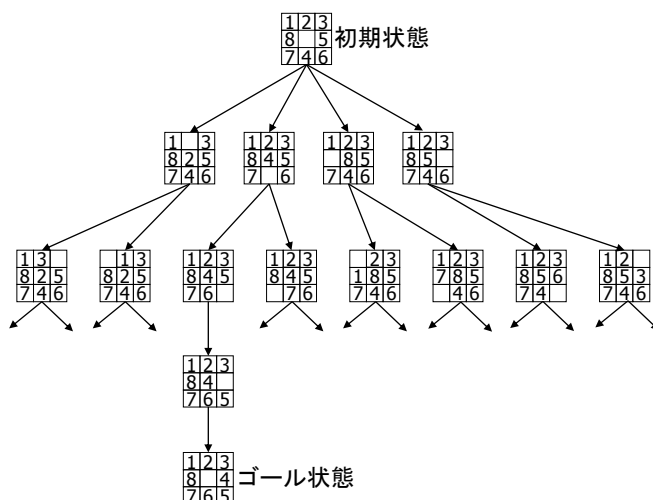
(解の)探索とは

- 解
 - 初期状態から目標状態に到達するまでの経路
- 解の探索
 - 状態空間のなかから、目標状態に到達する経路を見つけ出すこと
- 状態空間
 - 状態をノード、オペレータ適用による状態の遷移を状態(ノード)間のアークとみなし、グラフとして表現

状態空間のグラフ表現



状態空間のグラフ表現 (8パズルの例)



探索手法の分類

- コストを考慮しない探索(しらみつぶし探索)
 - 深さ優先探索(縦型探索)
 - 幅優先探索(横型探索)
 - 反復深化探索
- コストを考慮した探索
 - 経験的知識を用いない探索
 - 分枝限定法
 - 経験的知識を用いる探索
 - 最良優先探索
 - A*アルゴリズム(Aアルゴリズム)

深さ優先探索

- アルゴリズム
 1. 初期ノードをOPENリストに入れる。CLOSEDリストを空に初期化する。
 2. OPENリストが空ならば、探索は失敗し、実行は終了。
 3. OPENリストから先頭の要素Nを取り出し、CLOSEDリストの最後にNを追加する。
 4. 要素Nが目標ノードであれば、探索は成功し、実行を終了する。
 5. 要素Nから1ステップで到達できる子ノードのうちで、OPENリストにもCLOSEDリストにも含まれていないものをすべてOPENリストの先頭に追加する。
 6. 2へ戻る。

幅優先探索

- アルゴリズム

1. 初期ノードをOPENリストに入れる。CLOSEDリストを空に初期化する。
2. OPENリストが空ならば、探索は失敗し、実行は終了。
3. OPENリストから先頭の要素Nを取り出し、CLOSEDリストの最後にNを追加する。
4. 要素Nが目標ノードであれば、探索は成功し、実行を終了する。
5. 要素Nから1ステップで到達できる子ノードのうちで、OPENリストにもCLOSEDリストにも含まれていないものをすべてOPENリストの最後に追加する。
6. 2へ戻る。

OPENリストとCLOSEDリスト

- OPENリスト

- 探索において行き止まりから抜け出せるようにする
- 次に選択できる状態の候補の保持に利用

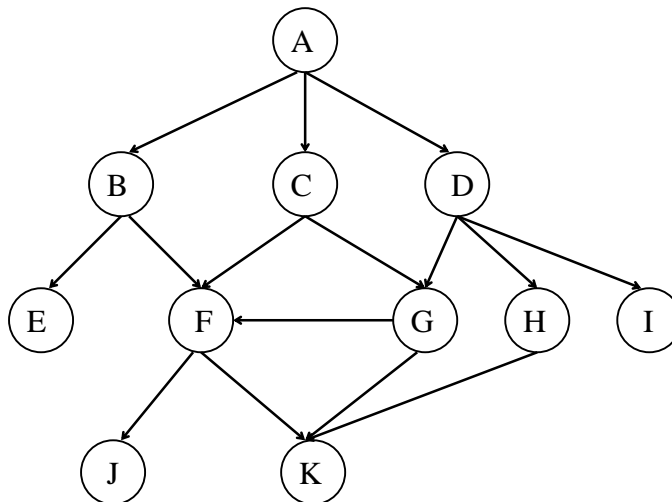
- CLOSEDリスト

- 一度選択したノードはCLOSEDリストに記録しておき、以後選択の候補にならないようにする
- 同じノードを二度選択しないために利用

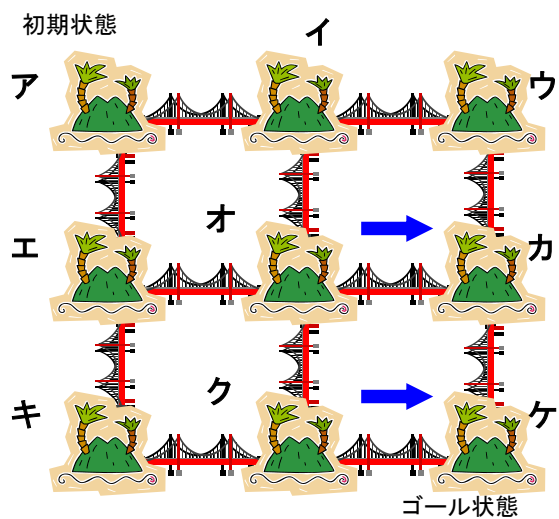
アルゴリズムの完全性

- グラフが有限で、解が存在すれば、必ず解を見つけることができる。
- これをアルゴリズムの完全性という。
- 深さ優先探索、幅優先探索とも完全性が保証されるアルゴリズムである。

グラフの例(1)

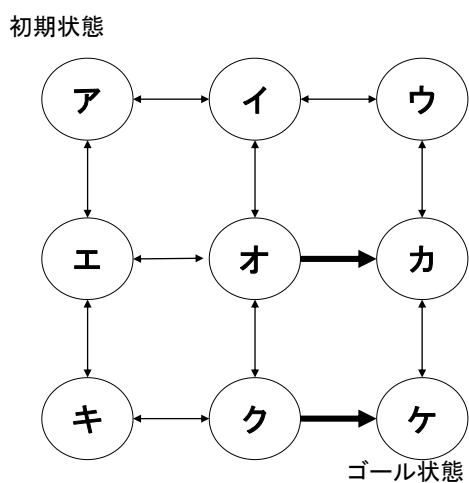


グラフの例(2-1)



島のルート探索

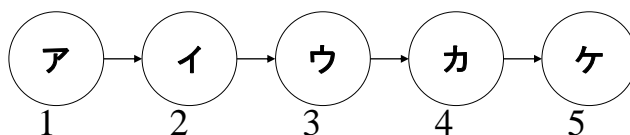
グラフの例(2-2)



グラフの例(2-3)

初期状態

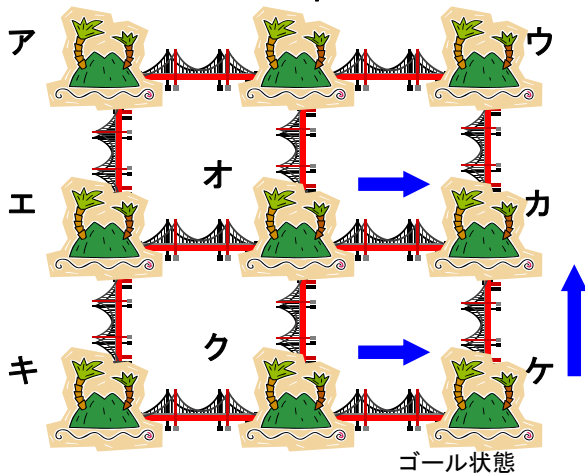
ゴール状態



グラフの例(3)

初期状態

イ



島のルート探索(一方通行の追加)