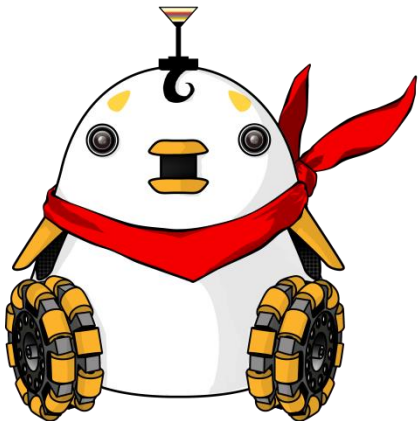


# 人工知能

## 第2章 探索(1): 状態空間と基本的な探索

立命館大学 情報理工学部

谷口彰



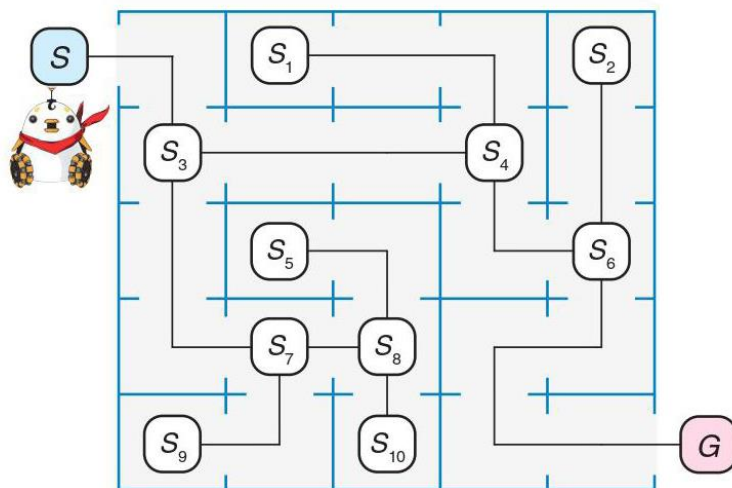
# STORY 状態空間と基本的な探索

- ホイールダック2号はダンジョンに入り，宝箱や出口を見つけなければならない．ホイールダック2号は宝箱に入ったアイテムや財宝を手に入れながら，出口に早くたどり着かなければならない．そして，スフィンクスを倒してさなければならない．
- **ダンジョン内は迷路になっている**．これを闇雲に進んでも，ゴールにたどり着けるのかもしれない．しかし，同じ所をくるくる回ってしまうかもしれないし，行き止まりにぶつかるかもしれない．では，**どのようにすれば効率的かつ確実に宝箱やゴールを見つけることができるのだろうか？**
- ホイールダック2号にまず求められたのは迷路をきちんと探索する能力だった．



# 仮定 状態空間と基本的な探索

- ホイールダック 2 号は迷路の完全な地図を持っているものとする。ただし、**地図上のゴールの位置はわからないものとする。**
- ホイールダック 2 号は迷路の中で**自分がどこにいるか認識できるものとする。**
- ホイールダック 2 号は連続的な迷路の**空間から適切な離散状態空間を構成できるものとする。**
- ホイールダック 2 号は物理的につながっている場所・状態には意図すれば**確定的に移動することができるものとする。**



# Contents

- 2.1 状態空間表現
- 2.2 迷路からの状態空間構成
- 2.3 基本的な探索
- 2.4 ホイールダック 2 号の迷路探索

## 2.1.1 ロボットと状態空間

- ロボットは**センサ系(sensor system)**と**モータ系(motor system)**（もしくはアクチュエータ系）を持つ。
- このような状況を数学的に表現することを目指すのが、広い意味での**状態空間表現(state space representation)**である。

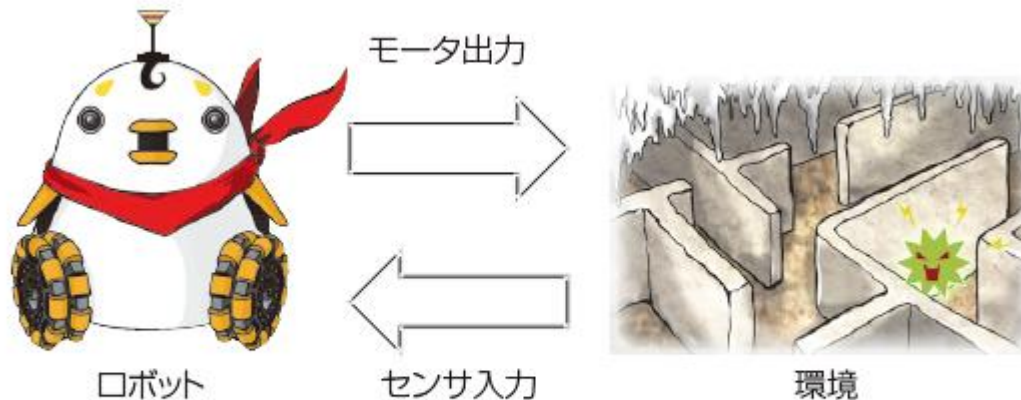


図 2.2 ロボットと環境の相互作用

## 2.1.2 システムのモデル化と不確実性

- **モデル化(modeling)**

- 「このように捉えよう」「このように捉えれば、そんなに間違っていないはずだ」と**システム(system)**を数理的に表現する.

- **不確実性の取り扱い**

- **確定システム**

- 行動後の状態が一通りに決まるシステム
- 例) 投球, ルービックキューブ

- **確率システム**

- 行動後の状態が1通りに決まらず確率的に変化するシステム
- 例) スロットマシン, 麻雀

## 2.1.3 連続システム

ロボシステム制御理論や力学では連続の状態空間で表現することが多い。

状態ベクトル(state vector)  $x_t$  と 行動ベクトル(action vector)  $u_t$  を用いて表現されることが多い。

### 連続システム

$$\text{状態遷移則} \quad x_{t+1} = F(x_t, u_t) \quad (2.1)$$

$$\text{連続状態空間} \quad x_t \in X = \mathbb{R}^n \quad (2.2)$$

$$\text{連続行動空間} \quad u_t \in U = \mathbb{R}^m \quad (2.3)$$

本書では扱わない

状態ベクトル  $x_t = (x_t^{pos}, y_t^{pos}, \theta_t^{pos})$

行動ベクトル  $u_t = (v_t^R, v_t^L)$

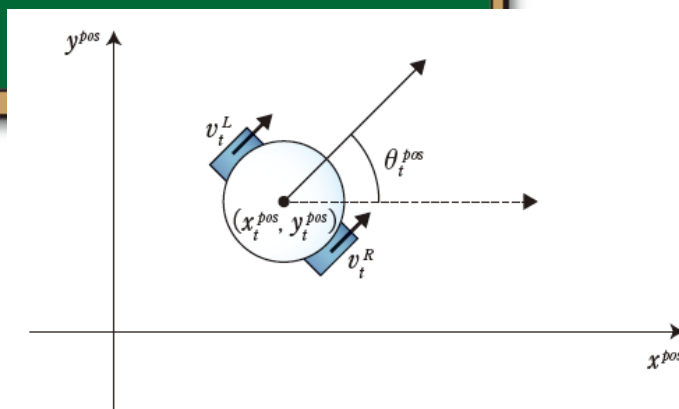


図 2.3 二輪移動ロボットの状態空間

## 2.1.4 離散システム

### 離散システム

$$\text{状態遷移則} \quad s_{t+1} = f(s_t, a_t) \quad (2.4)$$

$$\text{状態集合} \quad s_t \in S = \{1, 2, \dots, \#(S)\} \quad (2.5)$$

$$\text{行動集合} \quad a_t \in A = \{1, 2, \dots, \#(A)\} \quad (2.6)$$



□**離散システム**(discrete system) では, **状態**(state)  $s_t$  も**行動**(action)  $a_t$  も離散的な選択肢のうちの一つとなる.



## 2.1.5 離散システムとグラフ表現

- 離散確定システムでは  $s_{t+1} = f(s_t, a_t)$  によって状態遷移を表すことができる。
- 状態をノード，行動を有向辺で示す。

(例) 感情の状態を「うれしい」「ふつう」「かなしい」の三状態で定義

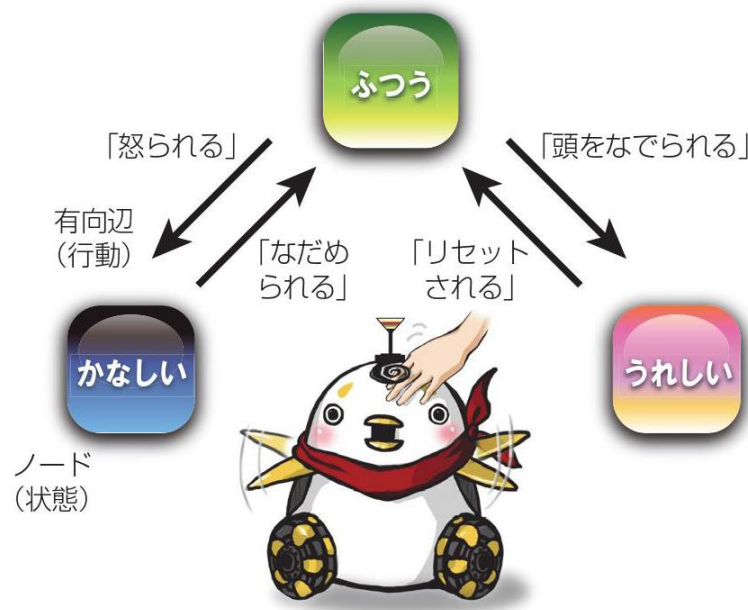


図 2.4 うれしい，ふつう，かなしいの 3 状態からなる感情状態のグラフ表現

# Contents

□2.1 状態空間表現

□2.2 迷路からの状態空間構成

□2.3 基本的な探索

□2.4 ホイールダック 2 号の迷路探索

## 2.2.1 マス(grid)ごとに状態をおく状態空間構成

- 1 マス1 マスを一つの状態として捉える
- ノード間は**無向辺**で結ばれている。

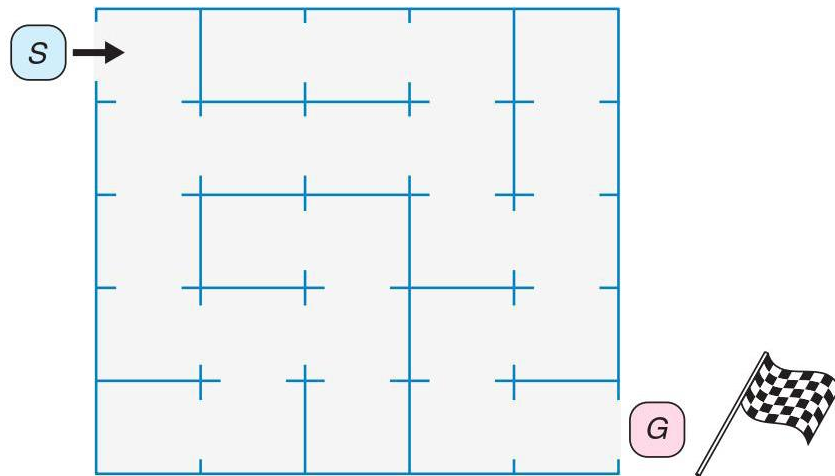


図 2.5 迷路の地図

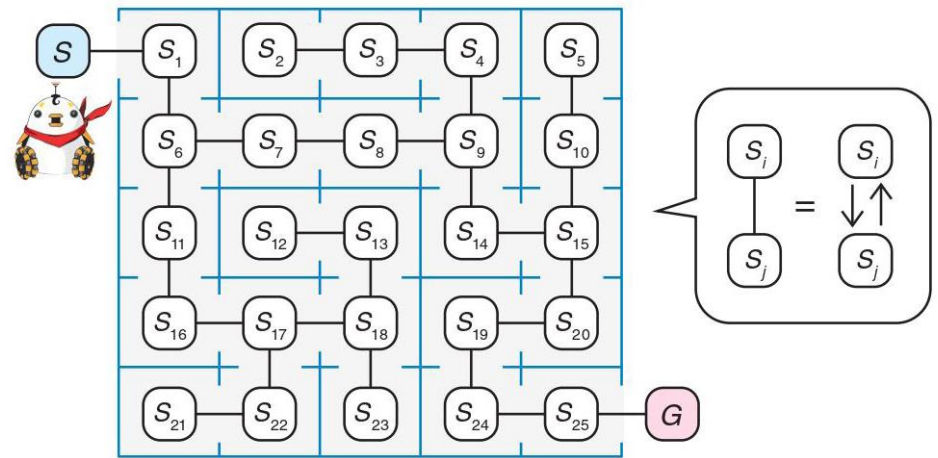
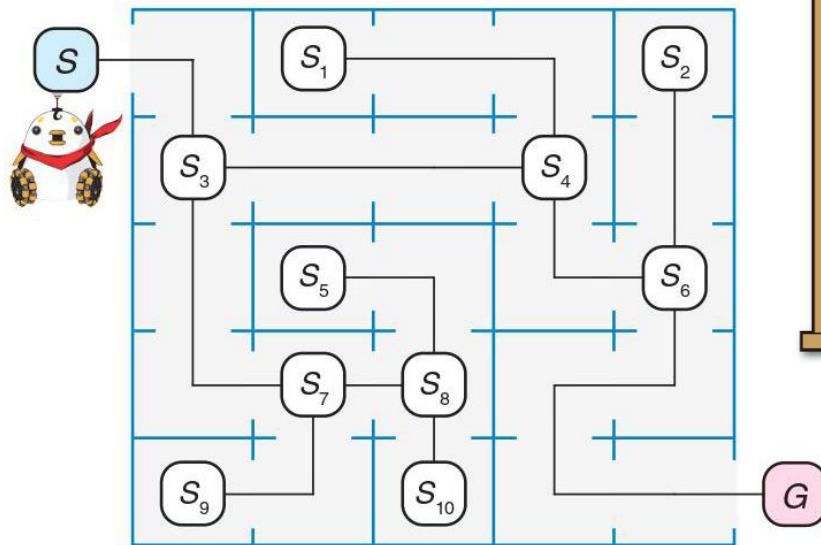


図 2.6 迷路の状態空間 (1 マス 1 状態)

非効率な表現になっている？

## 2.2.2 分岐と行き止まりに状態をおく状態空間構成

□たとえば「分岐」と「行き止まり」についてのみ状態をおいて状態空間を構成してみる.



迷路を表す状態空間の集合を用いた表現

状態集合  $S = \{S, S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8, S_9, S_{10}, G\}$  (2.7)

行動集合  $A = \{(S, S_3), (S_1, S_4), (S_2, S_6), (S_3, S_4), (S_3, S_7), (S_4, S_6), (S_5, S_8), (S_6, G), (S_7, S_8), (S_7, S_9), (S_8, S_{10})\}$  (2.8)

ここで,  $(S_i, S_j)$  は  $S_i$  から  $S_j$  へ移動する行動, および  $S_j$  から  $S_i$  へ移動する行動を表す.

図 2.7 迷路の状態空間（「分岐」と「行き止まり」を状態化）

※状態空間の作り方は物理的環境が決まれば決まるものではなく探索等がしやすいように「工夫して良い」ものである.

## 2.2.3 物体操作タスク(task)の状態空間構成

### □例) 物体操作タスクの状態空間

- 箱とぬいぐるみがあり，これらをおく場所が三箇所あるとする．
- 箱の上にぬいぐるみは乗るが，ぬいぐるみの上には箱は乗らない．
- ロボットは箱かぬいぐるみ，一方のみを持ち上げて任意の場所に移動させることができる．両方を同時に動かすことはできない．

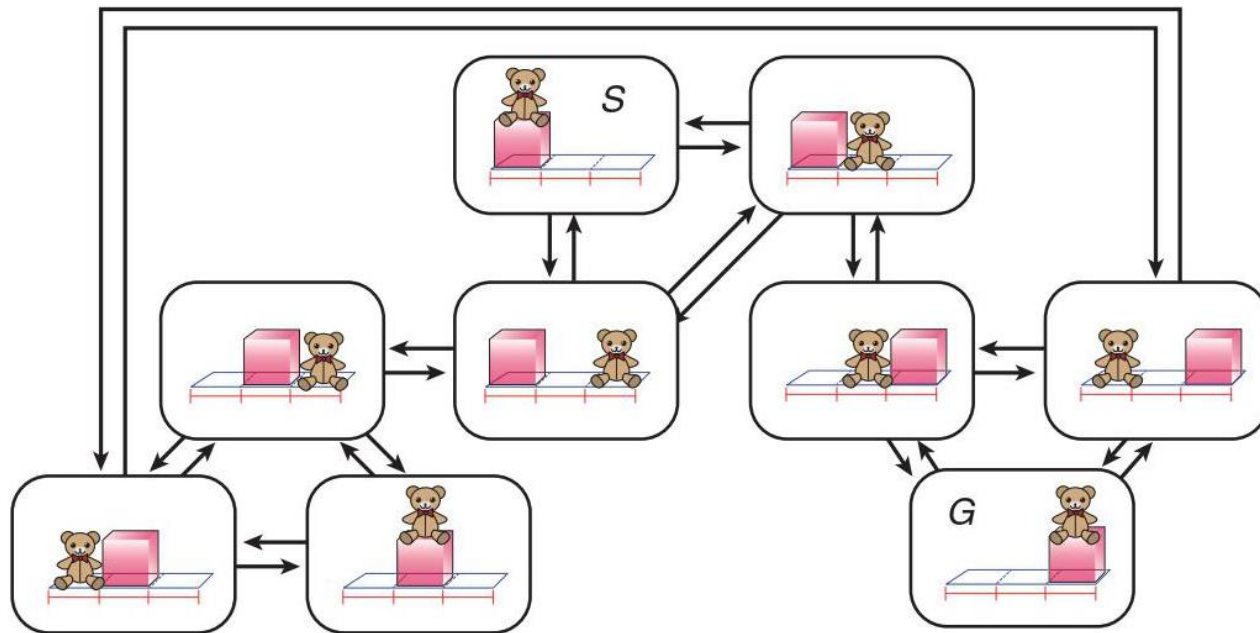


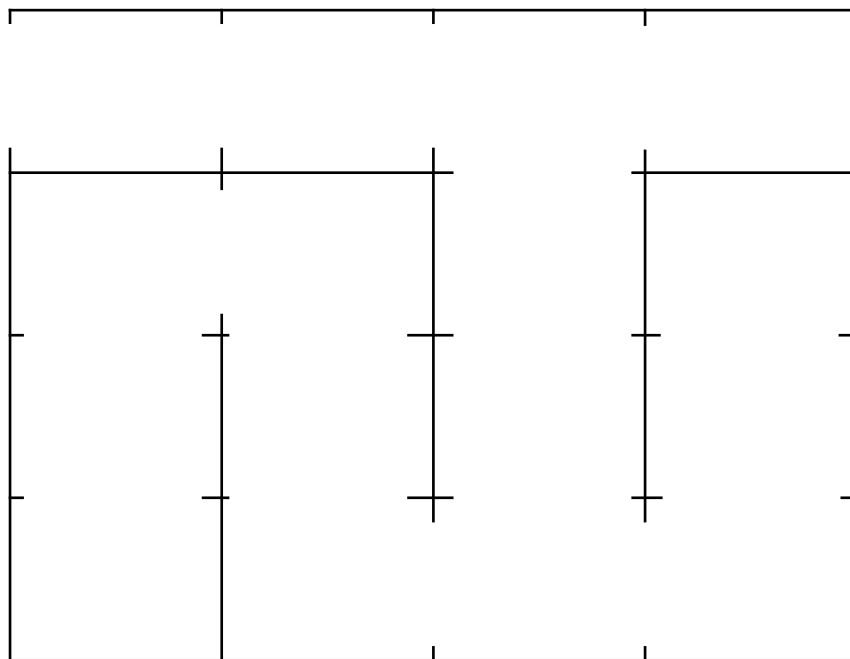
図 2.8 物体操作タスクの状態空間図

## 演習2-1 迷路からの状態空間構成

- 下記の迷路において「分岐」と「行き止まり」についてのみ状態をおいて状態空間を構成し、グラフ表現せよ。



Start



Goal

# Contents

□2.1 状態空間表現

□2.2 迷路からの状態空間構成

□2.3 基本的な探索

□2.4 ホイールダック 2 号の迷路探索

## 2.3.1 知識を用いない探索

□「どこはすでに調べたか」「どこはまだ探していないから調べるべきだ」というような情報を管理し、効率的にしらみつぶしにする必要がある.

### □探索問題

□初期状態から目標状態へ至る行動の系列を求めること

### □解(solution)

□目標状態へ至る行動の系列



## 2.3.2 オープンリストとクローズドリスト

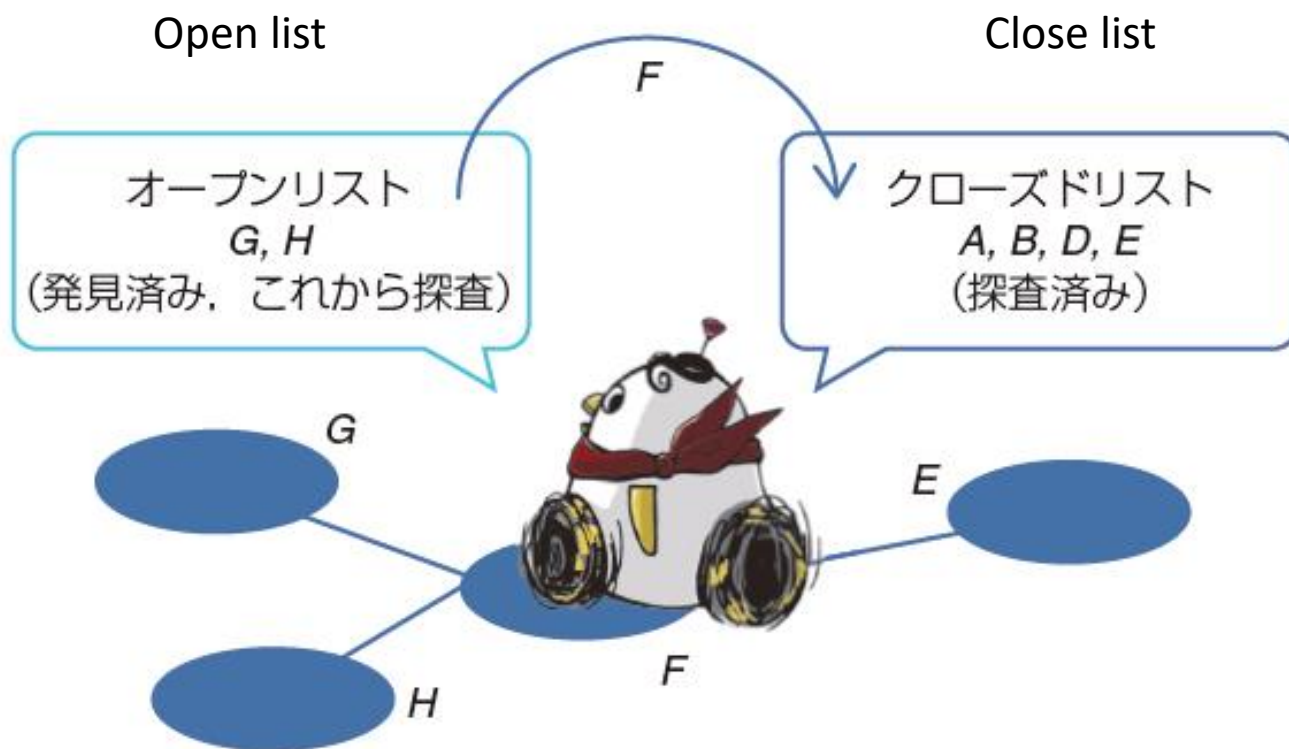
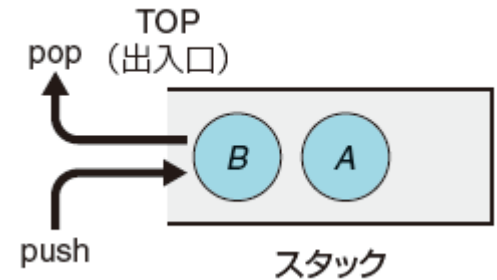


図 2.9

オープンリストとクローズドリスト

## 2.3.3 深さ優先探索



### Algorithm 2.1 深さ優先探索

- ① 初期状態をオープンリストに入れる．クローズドリストを空に初期化する．
- ② while オープンリストが空ではない． do
- ③     オープンリストから先頭の要素  $s$  を取り出す．クローズドリストに  $s$  を追加する ( $s$  を探査することに相当)．
- ④      $s$  が目標状態ならば，解は発見されたとして探索を終了．
- ⑤      $s$  から接続していてまだ探査していない状態をすべてオープンリストの先頭に追加する (**スタックにプッシュ**する)．
- ⑥ end while 探索を終了．

# 深さ優先探索

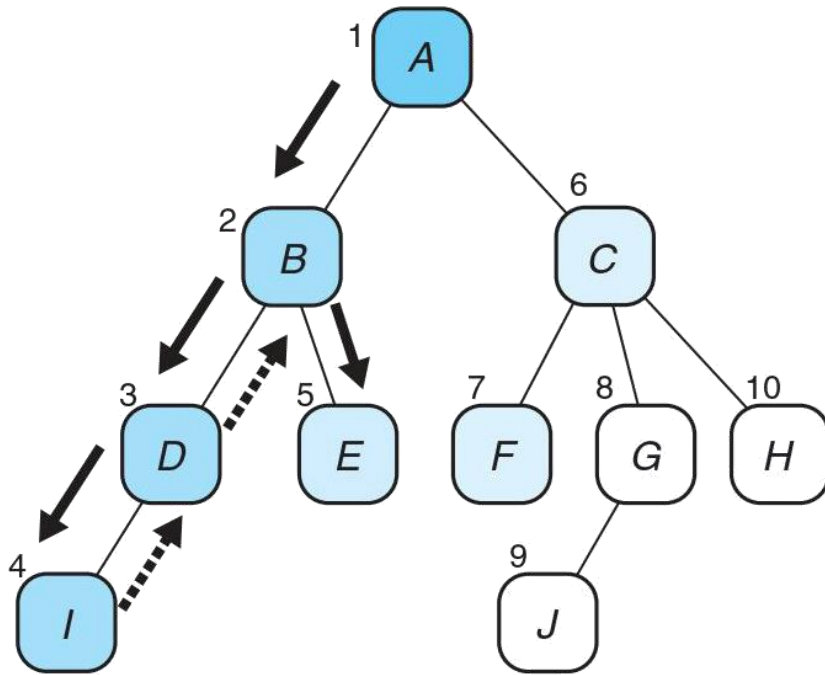


図 2.10 深さ優先探索の例

オープンリスト

クローズドリスト



- オープンリストとクローズドリストの変化を追ってみよう.

# 深さ優先探索の結果

| ステップ | オープンリスト | クローズリスト                      |
|------|---------|------------------------------|
| 1    | A       |                              |
| 2    | B, C    | A                            |
| 3    | D, E, C | A, B                         |
| 4    | I, E, C | A, B, D                      |
| 5    | E, C    | A, B, D, I                   |
| 6    | C       | A, B, D, I, E                |
| 7    | F, G, H | A, B, D, I, E, C             |
| 8    | G, H    | A, B, D, I, E, C, F          |
| 9    | J, H    | A, B, D, I, E, C, F, G       |
| 10   | H       | A, B, D, I, E, C, F, G, J    |
| 11   |         | A, B, D, I, E, C, F, G, J, H |

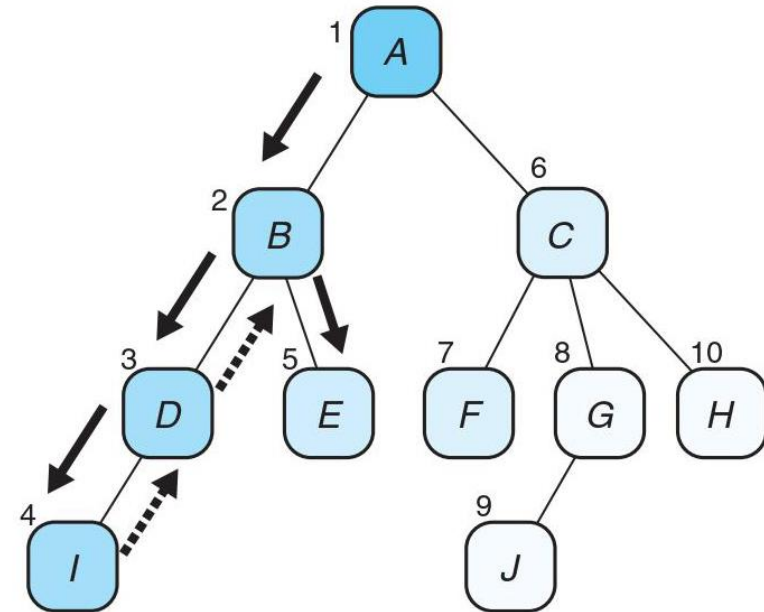
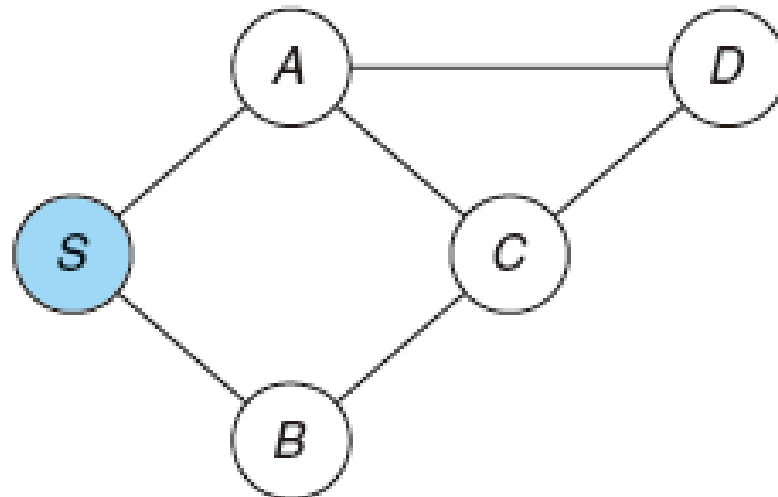


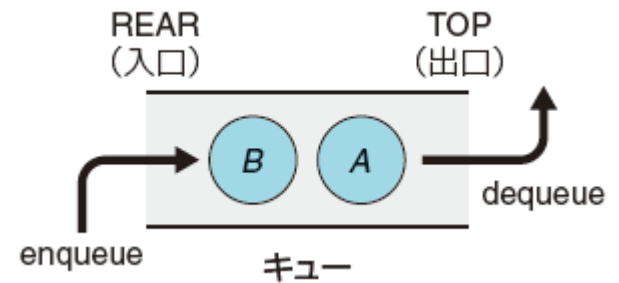
図 2.10 深さ優先探索の例

## 演習2-2 深さ優先探索

- 下図のグラフに関して,  $s$  を初期状態として深さ優先探索を行え. ただしそれぞれについて, オープンリストとクローズドリストの変化も示すこと.
- ただし、アルファベットの並びが前の方から探索する.



## 2.3.4 幅優先探索



### Algorithm 2.2 幅優先探索

- ① 初期状態をオープンリストに入れる．クローズドリストを空に初期化する．
- ② while オープンリストが空ではない．do
- ③     オープンリストから先頭の要素  $s$  を取り出す．クローズドリストに  $s$  を追加する ( $s$  を探査することに相当)．
- ④      $s$  が目標状態ならば，解は発見されたとして探索を終了．
- ⑤      $s$  から接続していてまだ探査していない状態をすべてオープンリストの**末尾**に追加する (**キューにエンキュー**する)．
- ⑥ end while 探索を終了．

# 幅優先探索

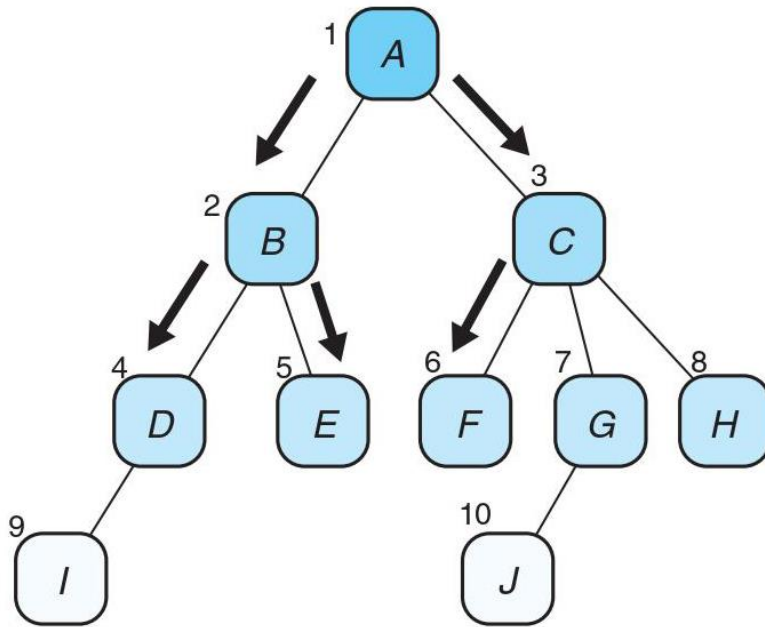


図 2.12 幅優先探索の例

オープンリスト

クローズドリスト



- オープンリストとクローズドリストの変化を追ってみよう.

# 幅優先探索の結果

| ステップ | オープンリスト       | クローズリスト                      |
|------|---------------|------------------------------|
| 1    | A             |                              |
| 2    | B, C          | A                            |
| 3    | C, D, E       | A, B                         |
| 4    | D, E, F, G, H | A, B, C                      |
| 5    | E, F, G, H, I | A, B, C, D                   |
| 6    | F, F, H, I    | A, B, C, D, E                |
| 7    | G, H, I       | A, B, C, D, E, F             |
| 8    | H, I, J       | A, B, C, D, E, F, G          |
| 9    | I, J          | A, B, C, D, E, F, G, H       |
| 10   | J             | A, B, C, D, E, F, G, H, I    |
| 11   |               | A, B, C, D, E, F, G, H, I, J |

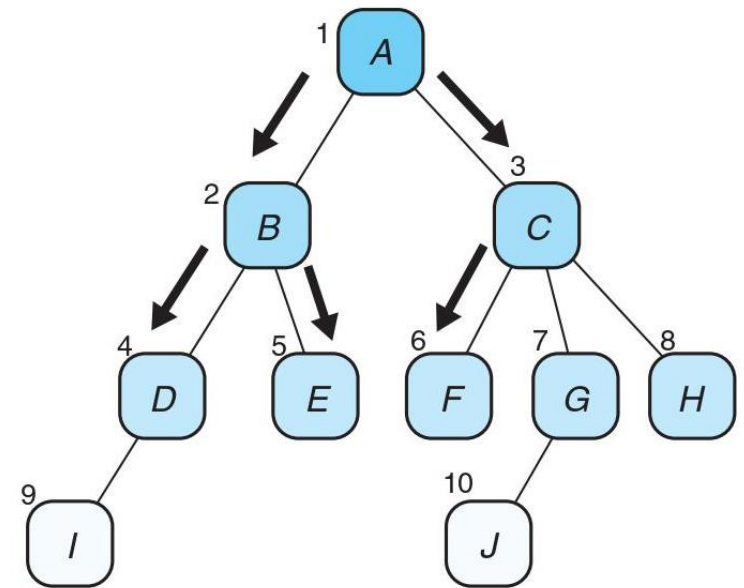
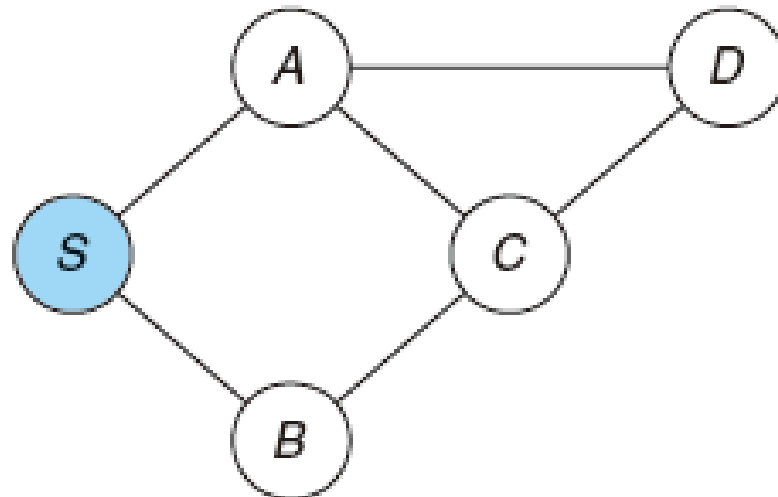


図 2.12 幅優先探索の例



## 演習2-3 幅優先探索

- 下図のグラフに関して,  $s$  を初期状態として幅優先探索を行え. ただしそれぞれについて, オープンリストとクローズドリストの変化も示すこと.
- ただし、アルファベットの並びが前の方から探索する.



# Contents

- 2.1 状態空間表現
- 2.2 迷路からの状態空間構成
- 2.3 基本的な探索
- 2.4 ホイールダック 2 号の迷路探索

# 演習 2-4 宝箱やゴールを求めて迷路を探索するホイールダック2号

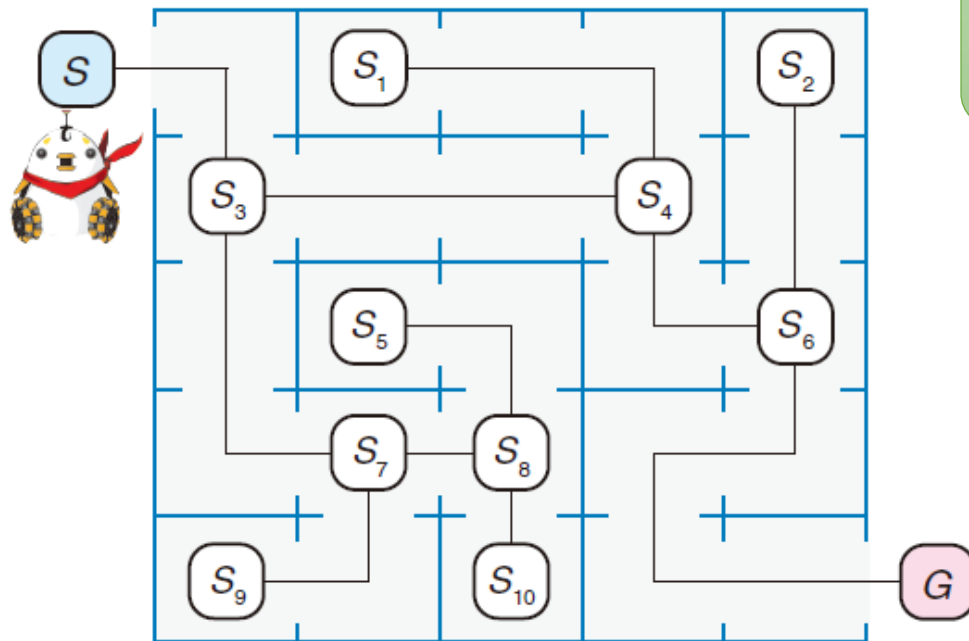


図 2.7 迷路の状態空間 (再掲載)

オープンリスト

クローズドリスト



深さ優先探索, 幅優先探索  
で迷路をぬけてみよう!

## 2.4.2 深さ優先探索と幅優先探索の比較

- 深さ優先探索の特徴

😊 オープンリストに記憶されるノード数があまり多くならないため、**状態空間の大きい探索木**を探索するのに適した手法である。

😞 解が初期ノードから近いところにある場合でも、深さを優先して探索を行なってしまいうため、解を発見するまでに無駄な探索をしてしまう可能性がある。

- 幅優先探索の特徴

😊 初期ノードに近いところから探索するため、**初期ノードから近い解を発見**するのに有効である。

😞 探索木の構造が横に大きいとき、探索のために保持するノード数が多くなってしまい、多くのメモリを必要とする。

## 第2章のまとめ

- 離散システムの状態空間のグラフ表現について学んだ.
- 状態空間表現を得る方法について学んだ.
- 基本的な探索手法として深さ優先探索と幅優先探索について学んだ.
- 深さ優先探索と幅優先探索におけるオープンリストとクローズドリストの管理方法について学んだ

