

## 情報Ⅳ

自律型のロボットが  $n \times n$  区画の迷路を走行するマイクロマウスの探索アルゴリズムを考える。次の文章の空欄 (58) (59) から (72) (73) には適切な数字を解答欄にマークしなさい。また、空欄 (74) (75) から (82) (83) にはもっとも適したものを選択肢から選び、解答欄にマークしなさい。

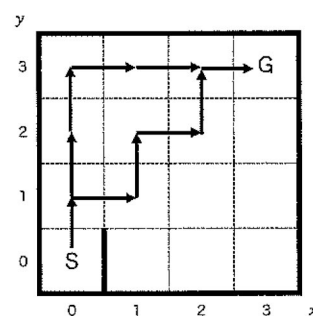
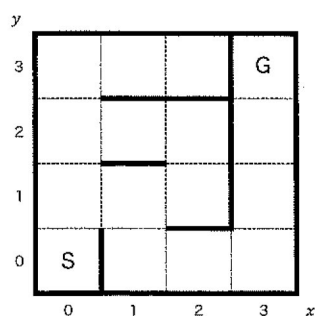
学習指導要領 (3) - 知・技 - イ

学習指導要領 (3) - 思・判・表 - イ

学習内容 (3) - イ アルゴリズムとプログラム

(ア) 実際のマイクロマウス競技では、 $16 \times 16$  (もしくは  $32 \times 32$ ) 区画の迷路中央にゴール地点が存在するため、迷路解析の代表的な手法である左手法等ではゴールまで到達することができない可能性がある。また、走行可能な持ち時間と回数も限られている。このような状況でも迷路の探索と最短経路の導出を同時に行える方法を考えてみよう。

簡単のため、次図 (左) のような  $4 \times 4$  の迷路を想定し、スタート座標  $S(0,0)$ 、ゴール座標  $G(3,3)$  とする。最初の状態ではスタート地点周囲以外の壁はどこにあるかわからないため、スタート地点周囲以外に壁は無いと仮定して最短経路を考えると、次図 (右) のように、一歩進んだ後、ゴールに向かって斜めギザギザに進む、もしくは 3 マス直進し、右折後 3 マス直進する経路などが最短経路となる。



実際には予定していた経路上に壁があれば通れないため、その時点で把握している迷路情報から導き出されるゴールへの最短経路を進み、必要に応じて最短経路を再導出する、次のような手順をとることにしよう。

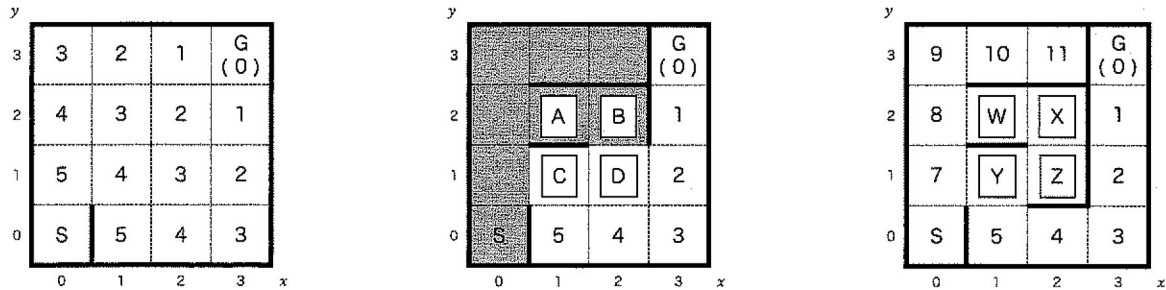
1. ゴールへの最短経路に沿って 1 マス進む
2. 壁を読み取る
3. 最短経路を導き直す

これをゴールに到達するまで繰り返せばよい。

ここで、ゴールの歩数を 0 とし、移動可能な隣接マスを 1、その隣を 2、... と、ゴールからの歩数をマップ化したものを歩数マップと呼ぼう。最初にマウスが認識している壁はスタート地点のもののみであり、この段階での歩数マップは次図 (左) のようになる。

既知の壁情報をもとに歩数マップを作成し、順次現在のマスより歩数の小さいマスへ移動し、1 マス移動するごとに壁情報と歩数マップを更新し、次に進む方向を決定する。グレーのマスを探した時点

での歩数マップの一部は次図（中）のようになる。このときマス A、B、C、D に入る歩数は、それぞれ (58) (59)、(60) (61)、(62) (63)、(64) (65) となる。壁の探索が終わると次図（右）のような歩数マップが完成する。このときマス W、X、Y、Z に入る歩数は、それぞれ (66) (67)、(68) (69)、(70) (71)、(72) (73) となる。タイムアタックでは歩数マップにしたがって進めば最短経路でゴールに到達する。



学習指導要領 (3) - 知・技 - イ  
学習指導要領 (3) - 思・判・表 - イ  
学習内容 (3) - イ アルゴリズムとプログラム

(イ) 上記の方針に従い、8×8 区画の迷路を対象とした探索アルゴリズムを記述すると次のようになる。ここで、 $M_{i,j}$  は座標  $(i,j)$  の歩数を、 $W_{i,j,n}$ 、 $W_{i,j,e}$ 、 $W_{i,j,s}$ 、 $W_{i,j,w}$  はそれぞれ座標  $(i,j)$  の北・東・南・西側の壁情報を保持する変数である。ただし、マップの上方向が北であり、壁情報は 0 (壁なし)、1 (壁あり)、2 (未探索) のいずれかの値をとる。スタート位置は座標  $(0,0)$  とし、スタート位置東側の壁は既知であるとする。

北方向に 1 マス直進する  
到達マスがゴールでなければ次の処理 A を繰り返す  
処理 A の始まり  
壁の情報を取得・更新する  
歩数マップを更新する【歩数マップ作成・更新アルゴリズム】  
前進方向マスの歩数マップの値が最小かつ移動可能であれば 1 マス前進する  
そうでなければ処理 B を実行する  
処理 B の始まり  
左隣接マスの歩数マップの値が最小かつ移動可能であれば左へ転回、1 マス進む  
そうでなければ処理 C を実行する  
処理 C の始まり  
右隣接マスの歩数マップの値が最小かつ移動可能であれば右へ転回、1 マス進む  
そうでなければ処理 D を実行する  
処理 D の始まり  
180 度転回し 1 マス戻る  
処理 D の終わり  
処理 C の終わり  
処理 B の終わり  
進行方向と座標を更新する  
処理 A の終わり

## 【歩数マップ作成・更新アルゴリズム】

ゴール  $M_{m,n}$  の値（歩数）を 0 とし、ゴール以外の全てのマス  $M_{x,y}$  の値を未定義を意味する 255 とする  
処理 A の始まり

変数  $f$ 、変数  $i$  の値をともに 0 にする

条件  $i < 8$  が成立する間、処理 B を実行する

処理 B の始まり

変数  $j$  の値を 0 にする

条件  $\boxed{(74)} \boxed{(75)}$  が成立する間、処理 C を実行する

処理 C の始まり

条件  $\boxed{(76)} \boxed{(77)}$  が成立する場合、処理 H に飛ぶ

条件  $\boxed{(78)} \boxed{(79)}$  が成立する場合、処理 D を実行する

処理 D の始まり

$M_{i,j+1}$  の値が 255 である場合、 $M_{i,j+1}$  の値を  $\boxed{(80)} \boxed{(81)}$  にする

変数  $f$  の値を 1 にする

処理 D の終わり

条件  $\boxed{(82)} \boxed{(83)}$  が成立する場合、処理 E を実行する

処理 E の始まり

$M_{i+1,j}$  の値が 255 である場合、 $M_{i+1,j}$  の値を  $\boxed{(84)} \boxed{(85)}$  にする

変数  $f$  の値を 1 にする

処理 E の終わり

条件  $\boxed{(86)} \boxed{(87)}$  が成立する場合、処理 F を実行する

処理 F の始まり

$M_{i,j-1}$  の値が 255 である場合、 $M_{i,j-1}$  の値を  $\boxed{(88)} \boxed{(89)}$  にする

変数  $f$  の値を 1 にする

処理 F の終わり

条件  $\boxed{(90)} \boxed{(91)}$  が成立する場合、処理 G を実行する

処理 G の始まり

$M_{i-1,j}$  の値が 255 である場合、 $M_{i-1,j}$  の値を  $\boxed{(92)} \boxed{(93)}$  にする

変数  $f$  の値を 1 にする

処理 G の終わり

処理 H: 変数  $j$  の値に 1 を加える

処理 C の終わり

変数  $i$  の値に 1 を加える

処理 B の終わり

変数  $f$  が 1 であれば処理 A の先頭に飛ぶ

処理 A の終わり

【(74) (75) ～ (92) (93) の選択肢】

- (11)  $j < 8$       (12)  $i < 8$       (13)  $j < i$       (14)  $i < j$   
 (15)  $M_{i,j}$  の値が 255 である      (16)  $M_{i,j}$  の値が 0 である  
 (17)  $W_{i,j,n}$  の値が 1 である      (18)  $W_{i,j,n}$  の値が 0 または 2 である  
 (19)  $W_{i,j,e}$  の値が 1 である      (20)  $W_{i,j,e}$  の値が 0 または 2 である  
 (21)  $W_{i,j,s}$  の値が 1 である      (22)  $W_{i,j,s}$  の値が 0 または 2 である  
 (23)  $W_{i,j,w}$  の値が 1 である      (24)  $W_{i,j,w}$  の値が 0 または 2 である  
 (25)  $M_{i,j}$       (26)  $M_{i,j} + 1$       (27)  $M_{i,j} - 1$       (28)  $M_{i+1,j}$   
 (29)  $M_{i+1,j} + 1$       (30)  $M_{i+1,j} - 1$       (31)  $M_{i,j+1}$       (32)  $M_{i,j+1} + 1$   
 (33)  $M_{i,j+1} - 1$       (34)  $M_{i-1,j}$       (35)  $M_{i-1,j} + 1$       (36)  $M_{i-1,j} - 1$   
 (37)  $M_{i,j-1}$       (38)  $M_{i,j-1} + 1$       (39)  $M_{i,j-1} - 1$