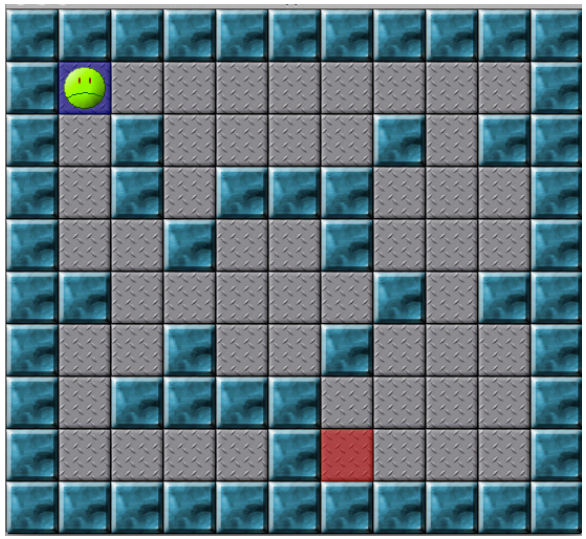
CS 知的システム演習

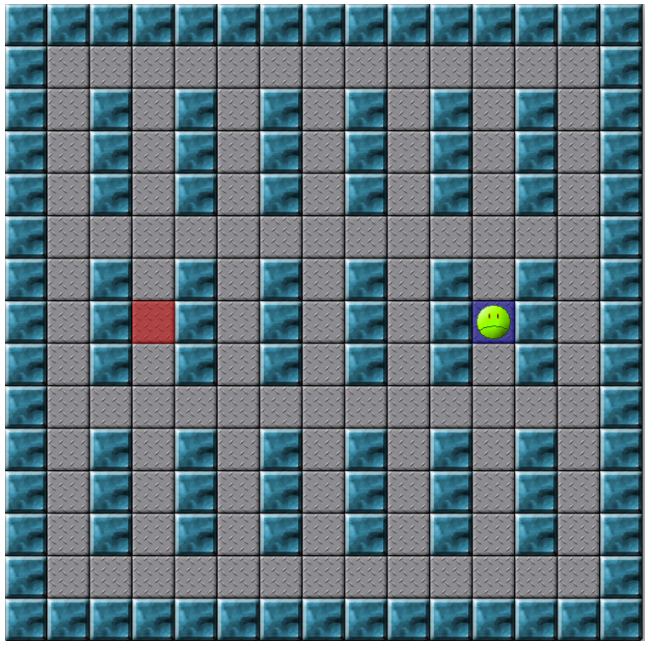
強化学習

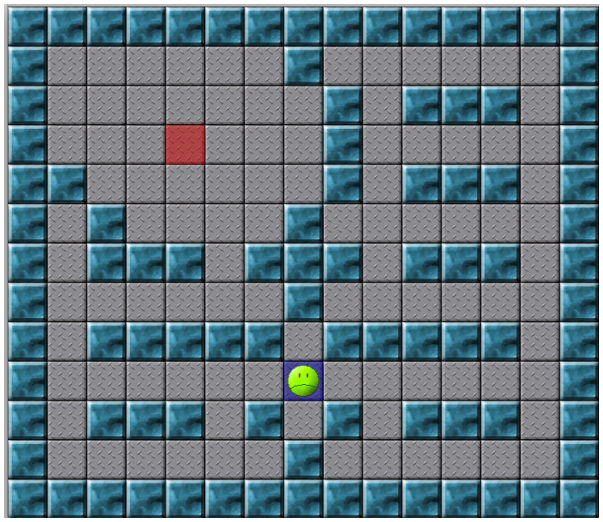
松吉 俊

複雑な迷路を解く

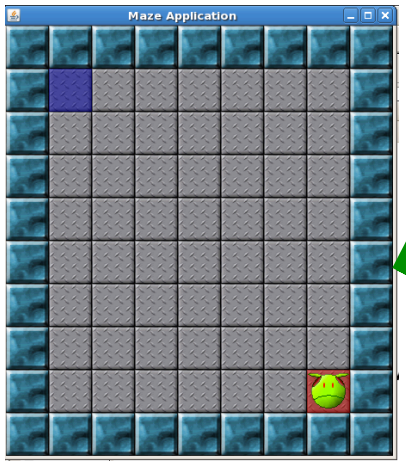
- アルゴリズムをさらに汎用化し、map4.txtとmap5.txtとmap6.txtも解くことができるように改良したい

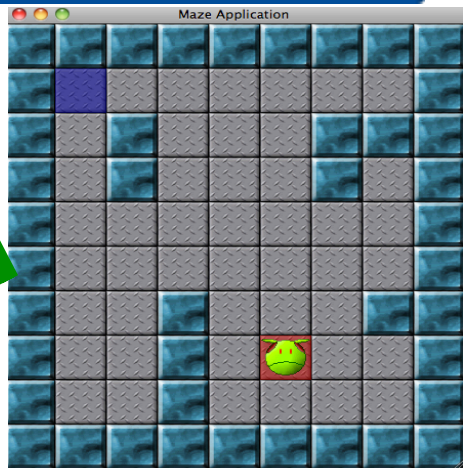




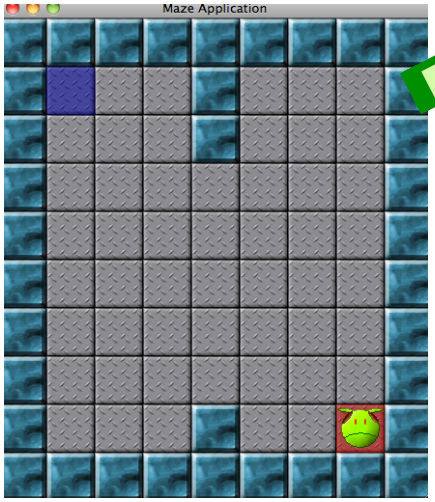


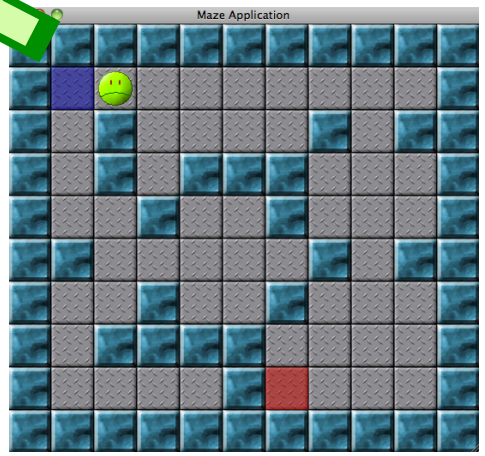
自律移動ロボット





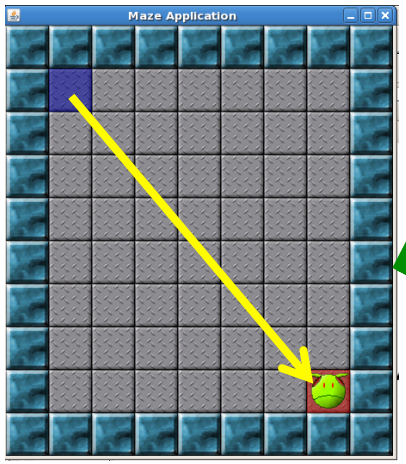
どんな迷路でも
自律的にゴールに
移動したい

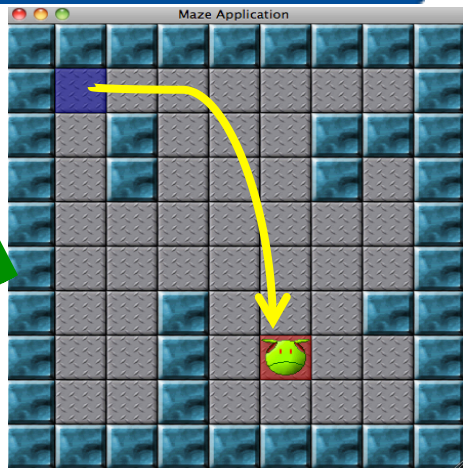




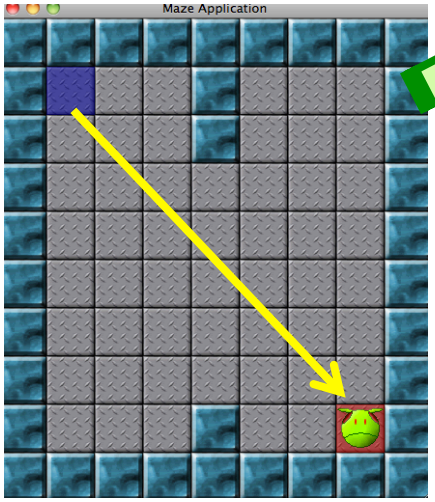
新しい迷路が追加される度に、
人間が、その都度、方法を
指示しないような状態

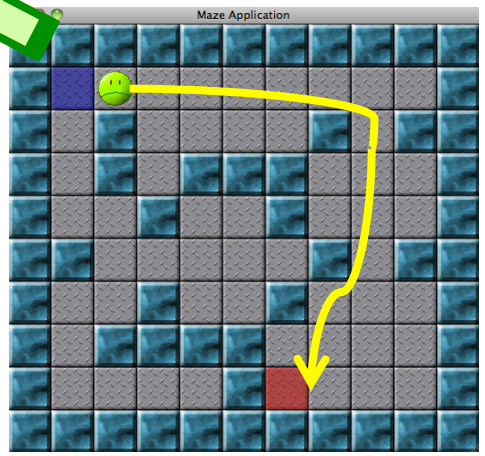
自律移動ロボット





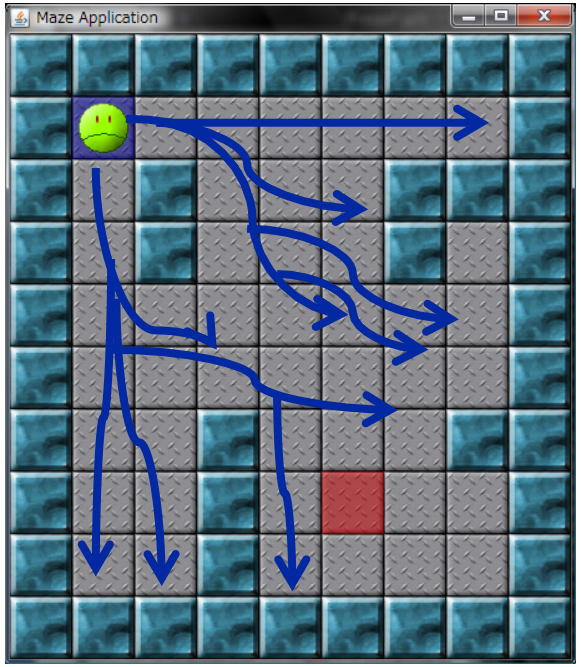
機械学習によって
任意の迷路の最適経路
を獲得する！





最適経路の学習原理

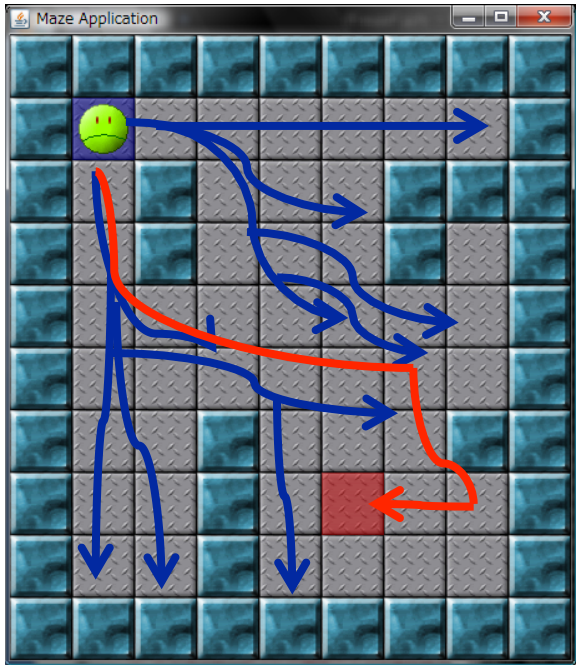
- ヒトや動物の学習機構を真似る！
  - 試行錯誤 (下手な鉄砲を目一杯打つ)
  - 古典的条件付け (パブロフの犬)



試行錯誤でゴールに到る経路を探索

最適経路の学習原理

- ヒトや動物の学習機構を真似る！
  - 試行錯誤 (下手な鉄砲を目一杯打つ)
  - 古典的条件付け (パブロフの犬)



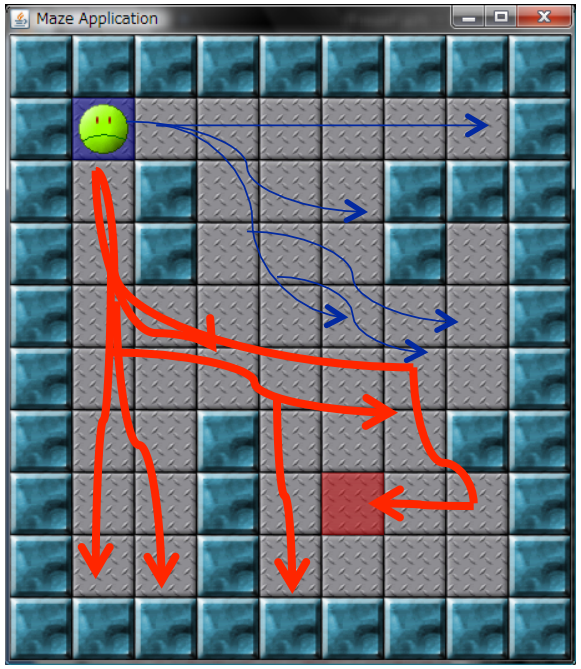
試行錯誤でゴールに到る経路を探索

そのうち経路が見つかるはず
(最適な経路かどうかは気にしない)

この経路をとても褒める

最適経路の学習原理

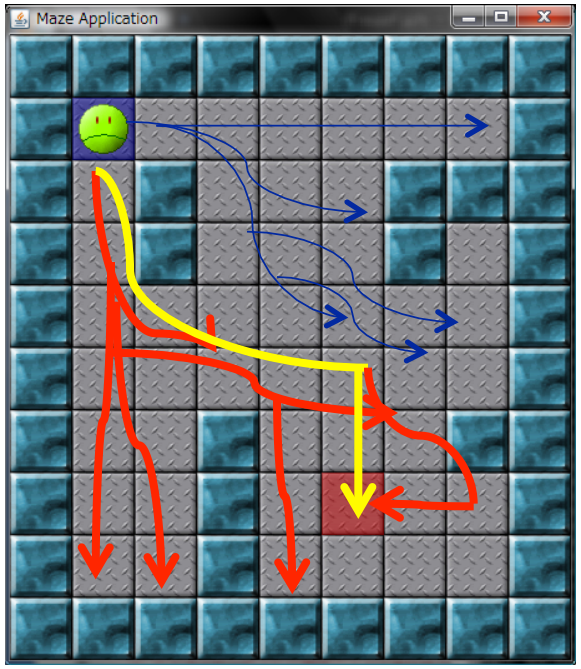
- ヒトや動物の学習機構を真似る！
  - 試行錯誤 (下手な鉄砲を目一杯打つ)
  - 古典的条件付け (パブロフの犬)



試行錯誤だが今度は見つかっている
経路周辺を重点的に探し出す
(また褒めてもらえそうなので)

最適経路の学習原理

- ヒトや動物の学習機構を真似る！
  - 試行錯誤 (下手な鉄砲を目一杯打つ)
  - 古典的条件付け (パブロフの犬)



試行錯誤だが今度は見つかっている
経路周辺を重点的に探し出す

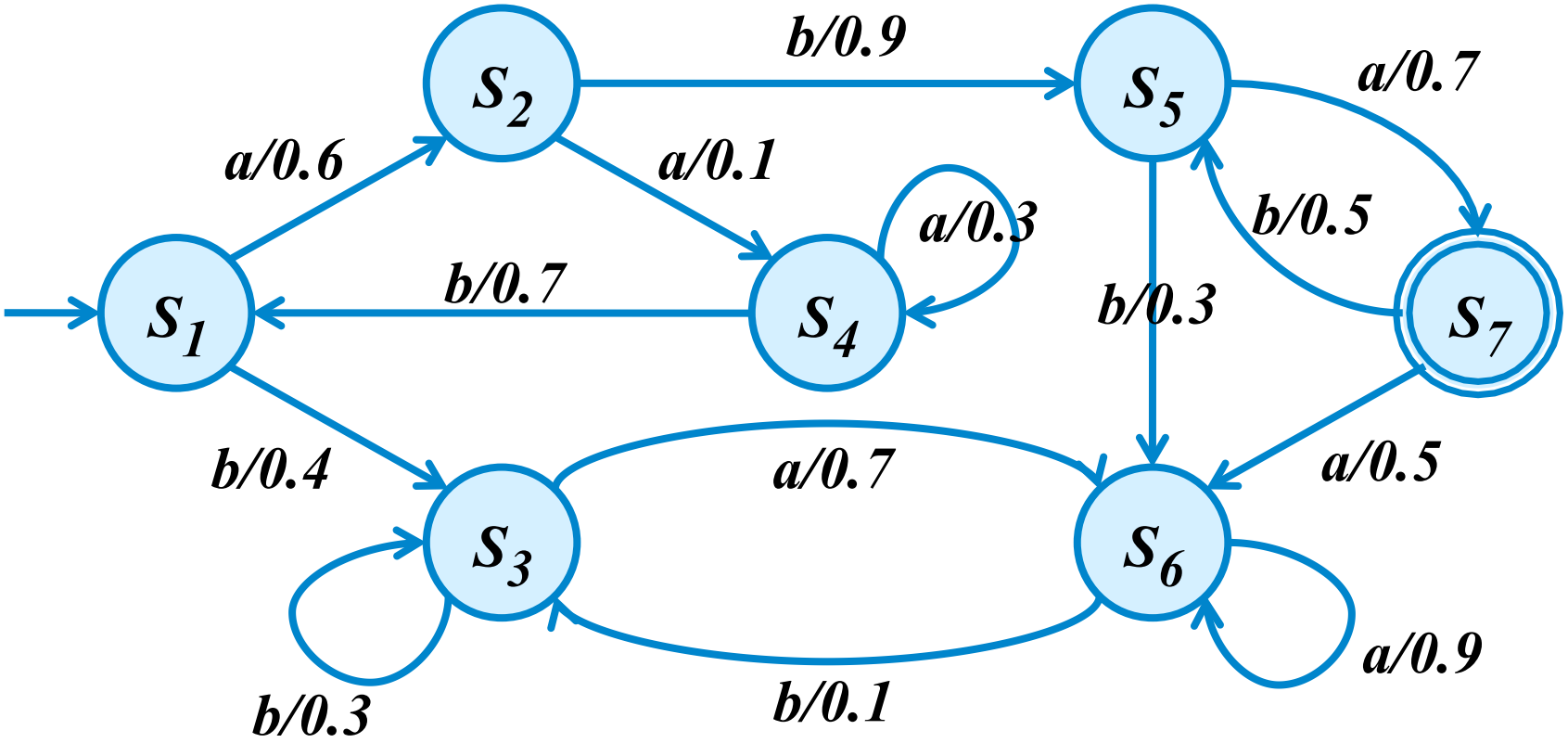
そのうち最適経路が見つかるはず

強化学習

- 目的
- 政策
- 報酬関数
- 行動の評価値: Q値

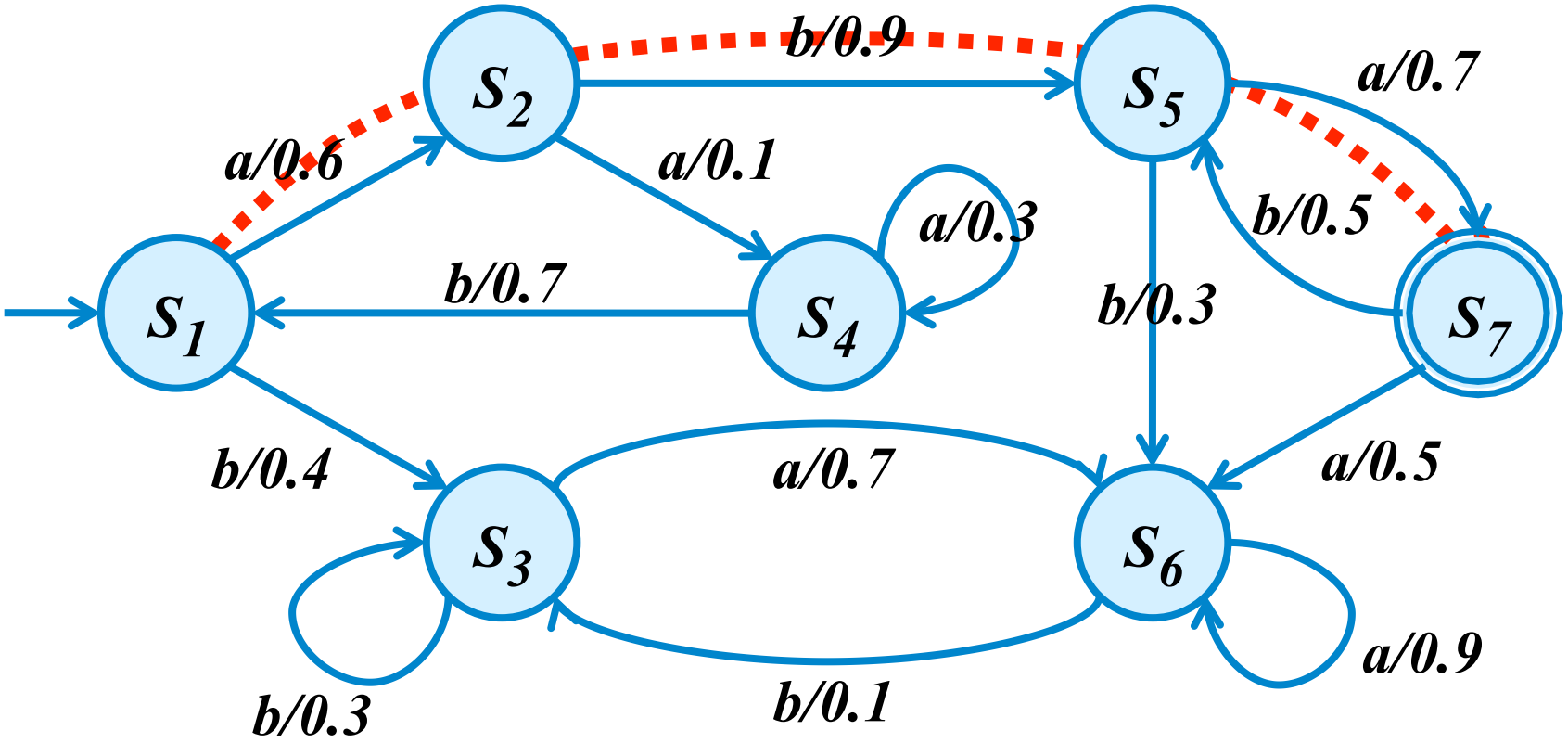
- 強化学習アルゴリズムの例
  - Q学習のアルゴリズム

強化学習の目的



（状態遷移確率付）有限オートマトンにおいて
目標状態に到達するような政策を学習したい！

強化学習の目的

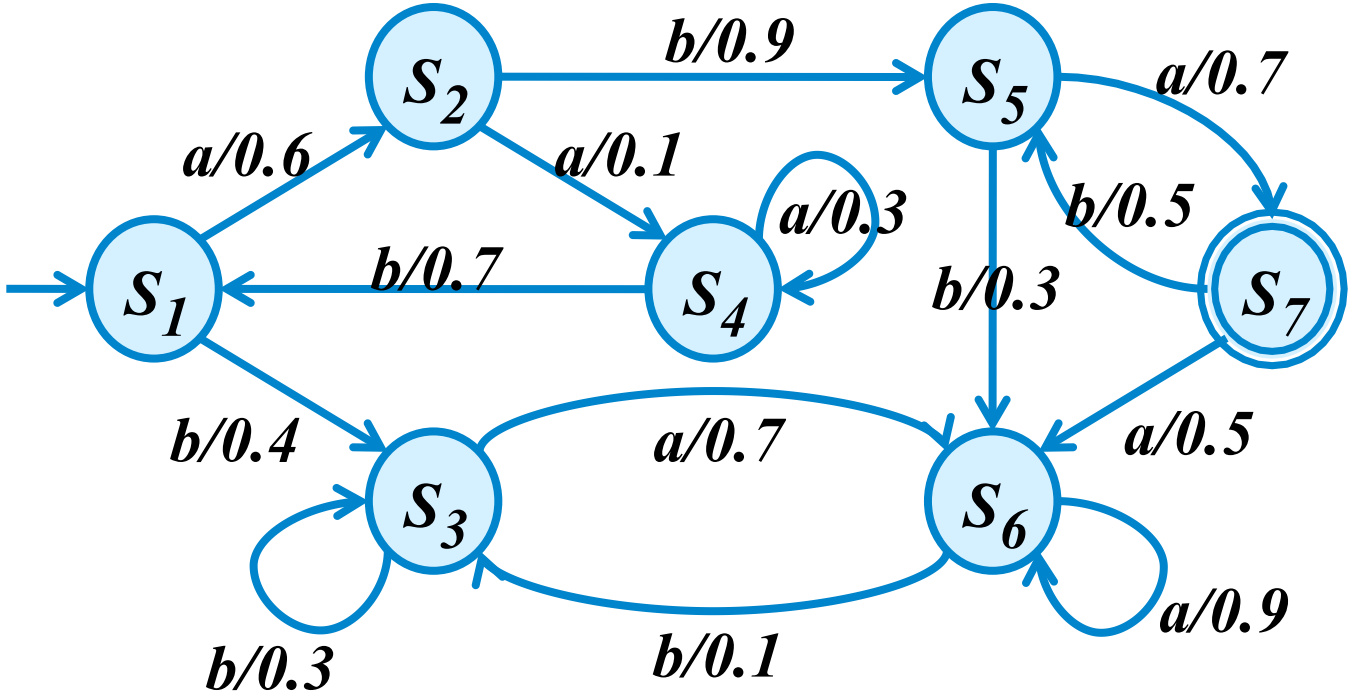


（状態遷移確率付）有限オートマトンにおいて
目標状態に到達するような政策を学習したい！

政策

(本によっては「方策」と呼ぶこともある)

- 状態から行動への写像 π



求めたい政策

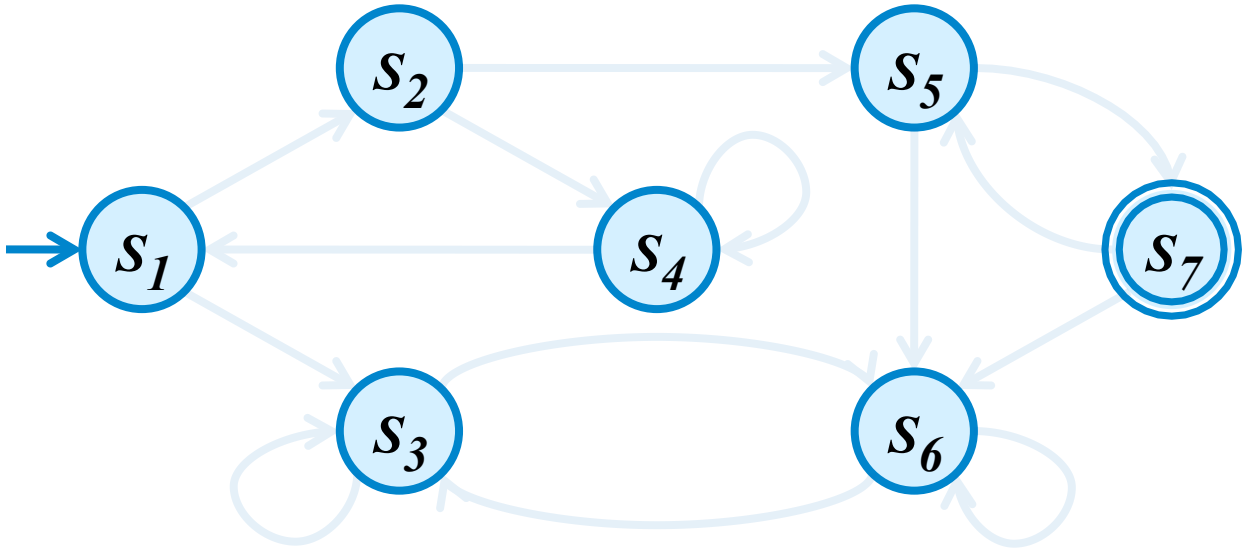
$$\pi(s_1) = a$$

$$\pi(s_2) = b$$

$$\pi(s_5) = a$$

一般的に、遷移関係は未知

- ある状態である行動を実行したときに
  その結果、**どの状態に遷移するかは不明**



求めたい政策

$$\pi(s_1) = a$$
$$\pi(s_2) = b$$
$$\pi(s_5) = a$$

遷移関係は不明！

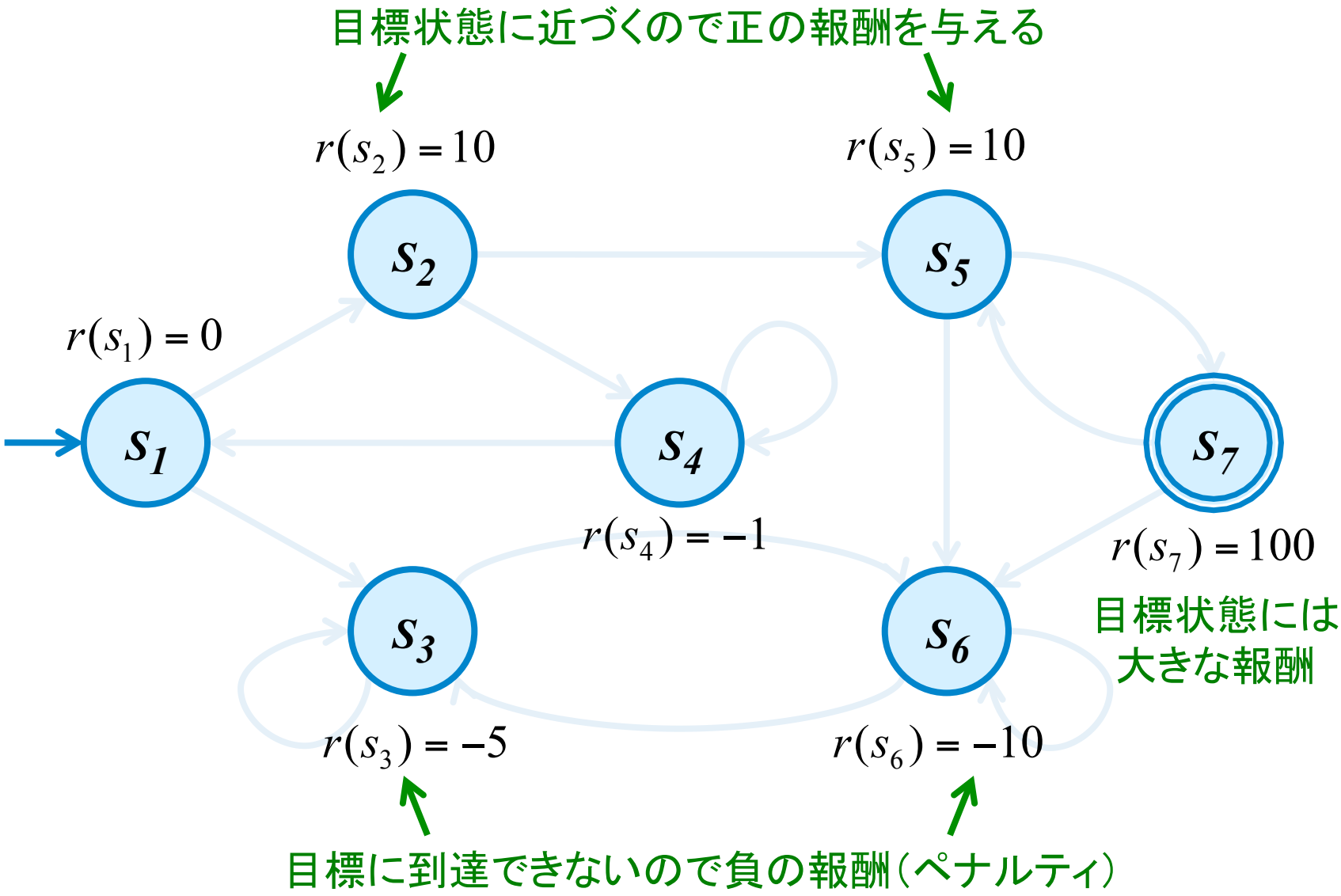
政策を学習するために

報酬関数の導入

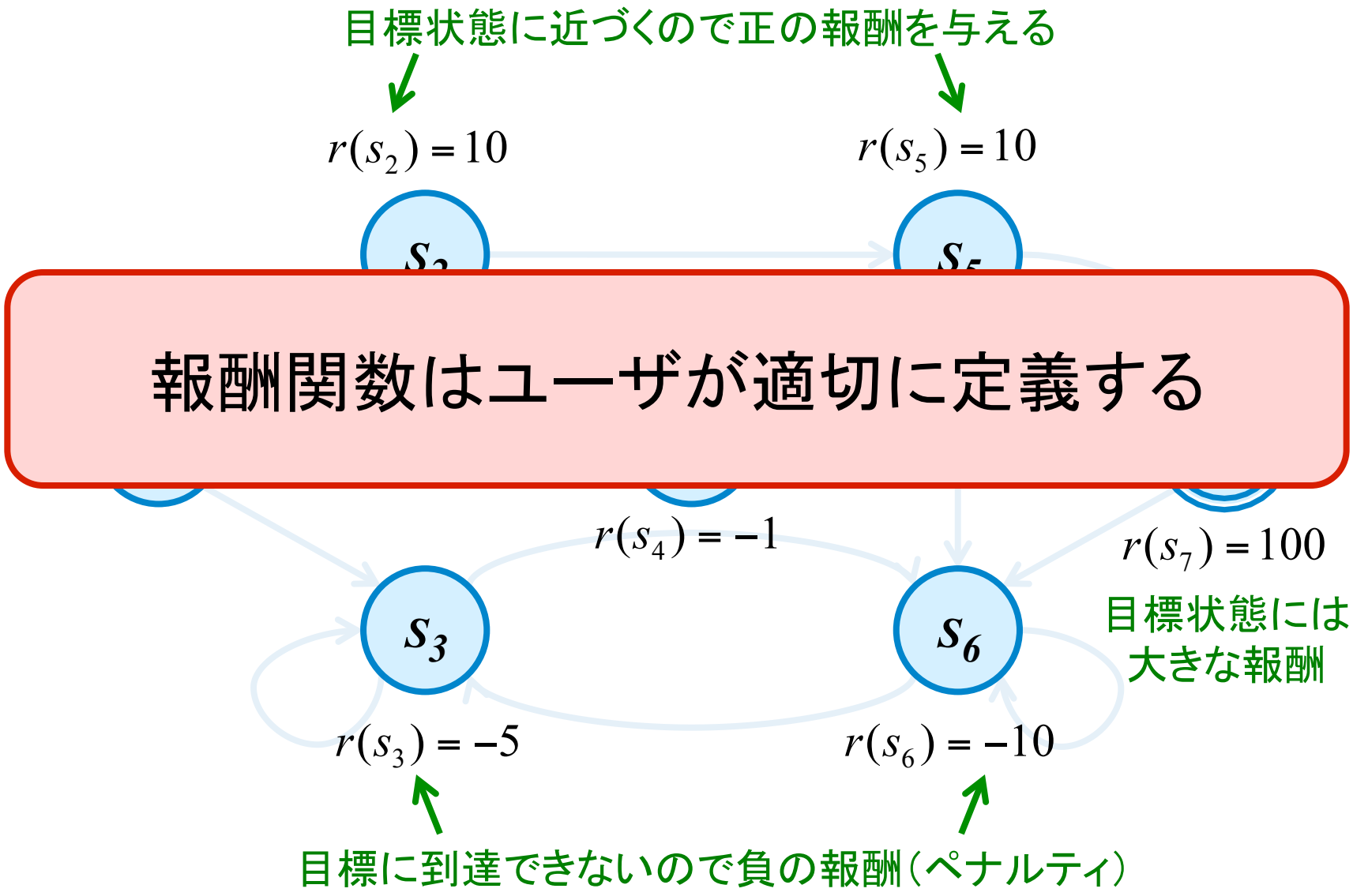
- 報酬関数 $r(s)$
  - 強化学習問題において目標を定義する
  - 状態 s を１個の数字である報酬（reward）に写像する
  - 報酬はその状態の望ましさを表す

強化学習では、最終的に受け取る総報酬を
最大化するような政策を学習する

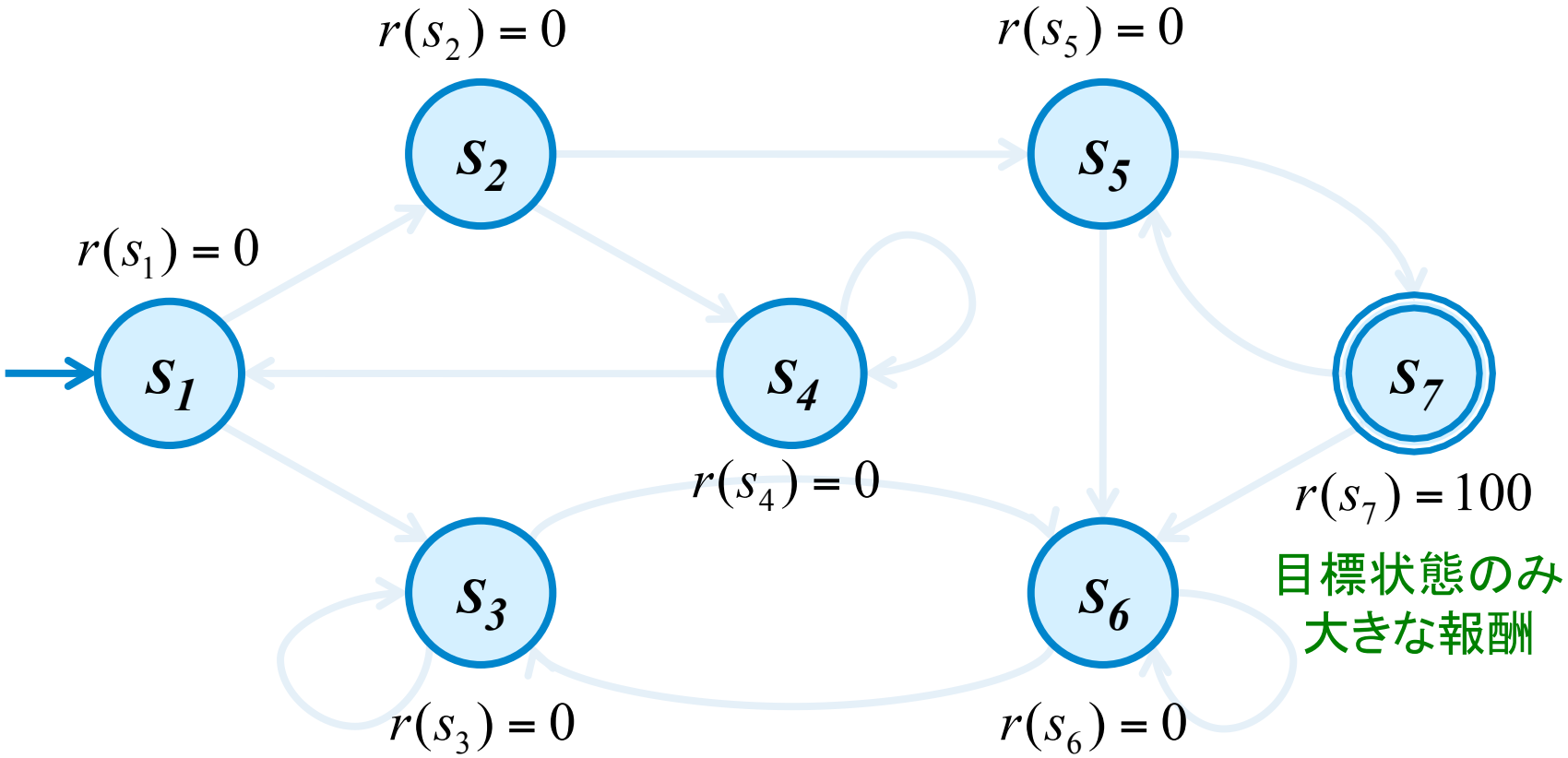
報酬関数の例1


目標状態に近づくので正の報酬を与える
$r(s_2) = 10$
$r(s_5) = 10$
s_2
s_5
$r(s_1) = 0$
s_1
s_4
s_7
$r(s_4) = -1$
$r(s_7) = 100$
目標状態には
大きな報酬
s_3
s_6
$r(s_3) = -5$
$r(s_6) = -10$
目標に到達できないので負の報酬（ペナルティ）

報酬関数の例1

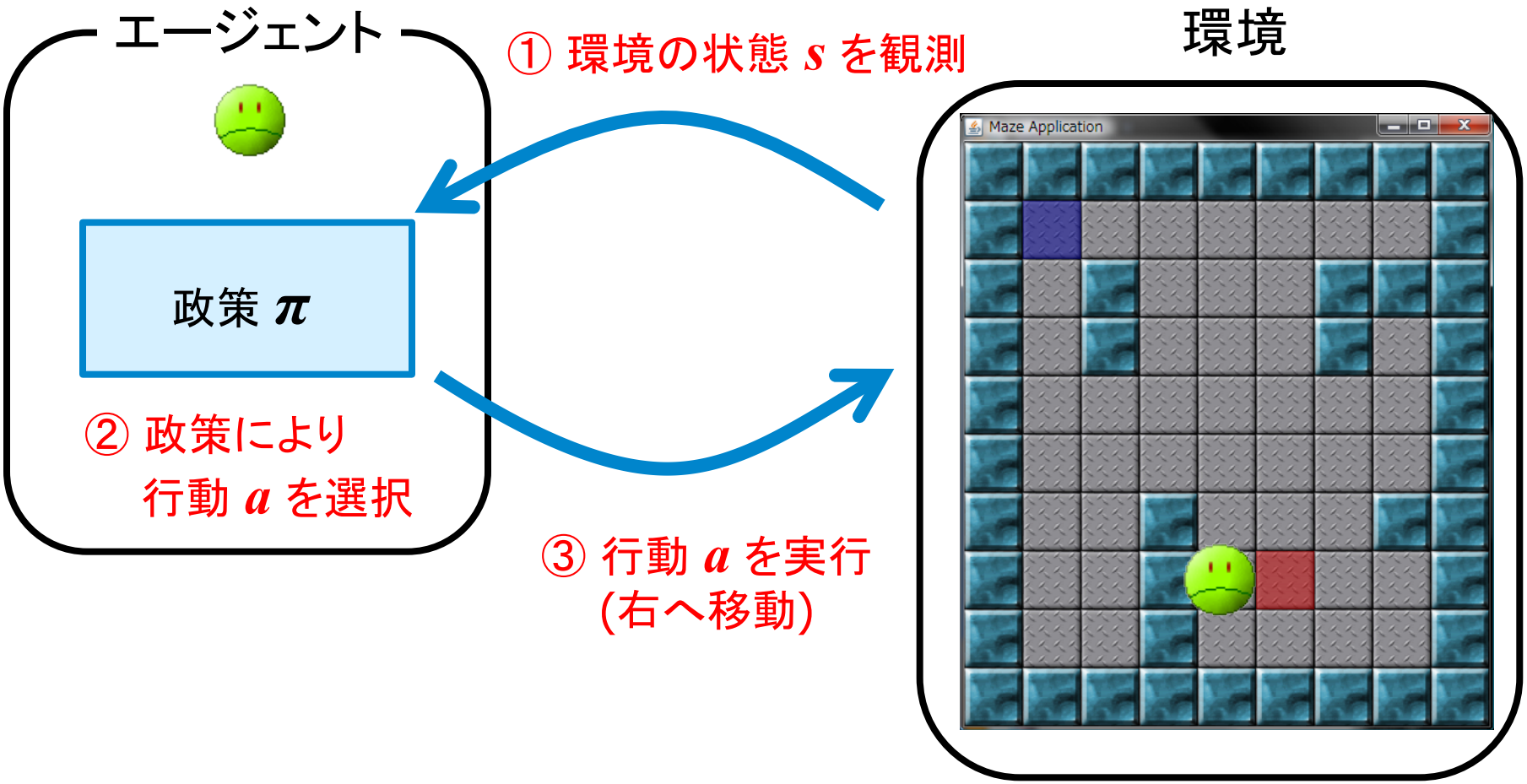

目標状態に近づくので正の報酬を与える
$r(s_2) = 10$
$r(s_5) = 10$
報酬関数はユーザが適切に定義する
$r(s_4) = -1$
$r(s_7) = 100$
s_3
s_6
目標状態には
大きな報酬
$r(s_3) = -5$
$r(s_6) = -10$
目標に到達できないので負の報酬（ペナルティ）

報酬関数の例2

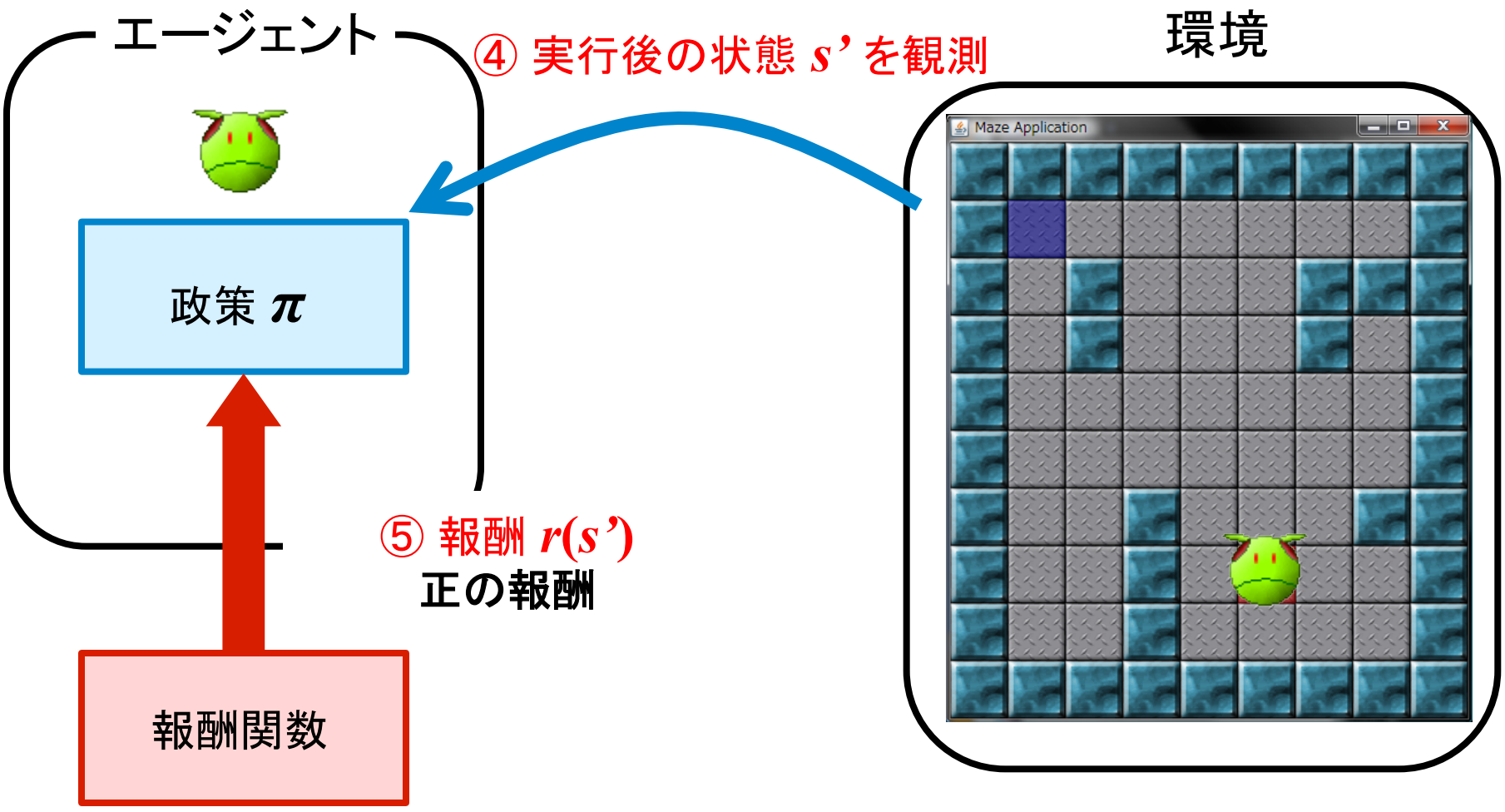


例えば、目標状態のみに報酬を定義しても良い
ただし、学習がうまくいかない場合は、
より良い報酬関数を定義する必要がある

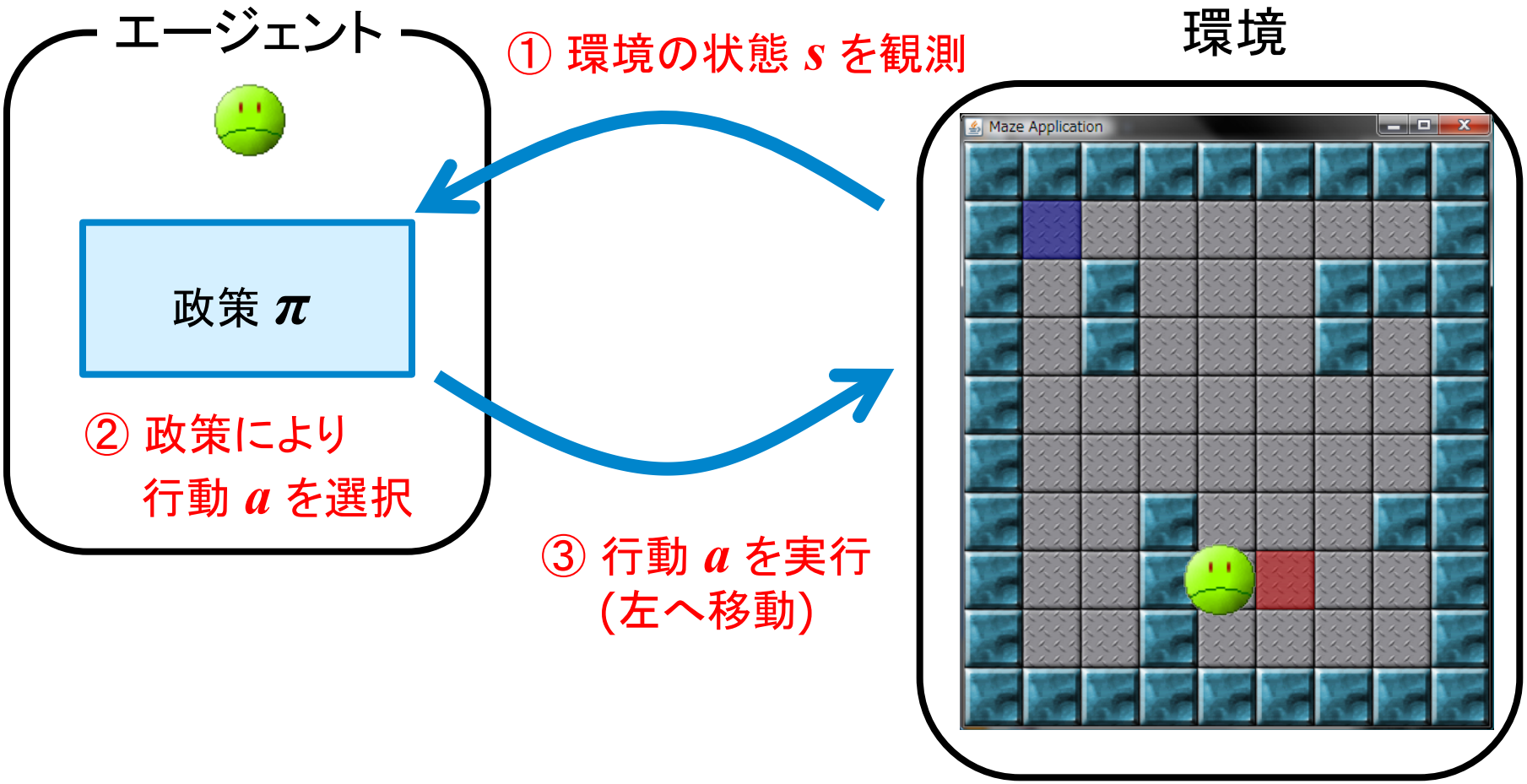
政策と報酬の関係(1)


エージェント
環境
① 環境の状態 s を観測
政策 π
② 政策により
行動 a を選択
③ 行動 a を実行
(右へ移動)
Maze Application

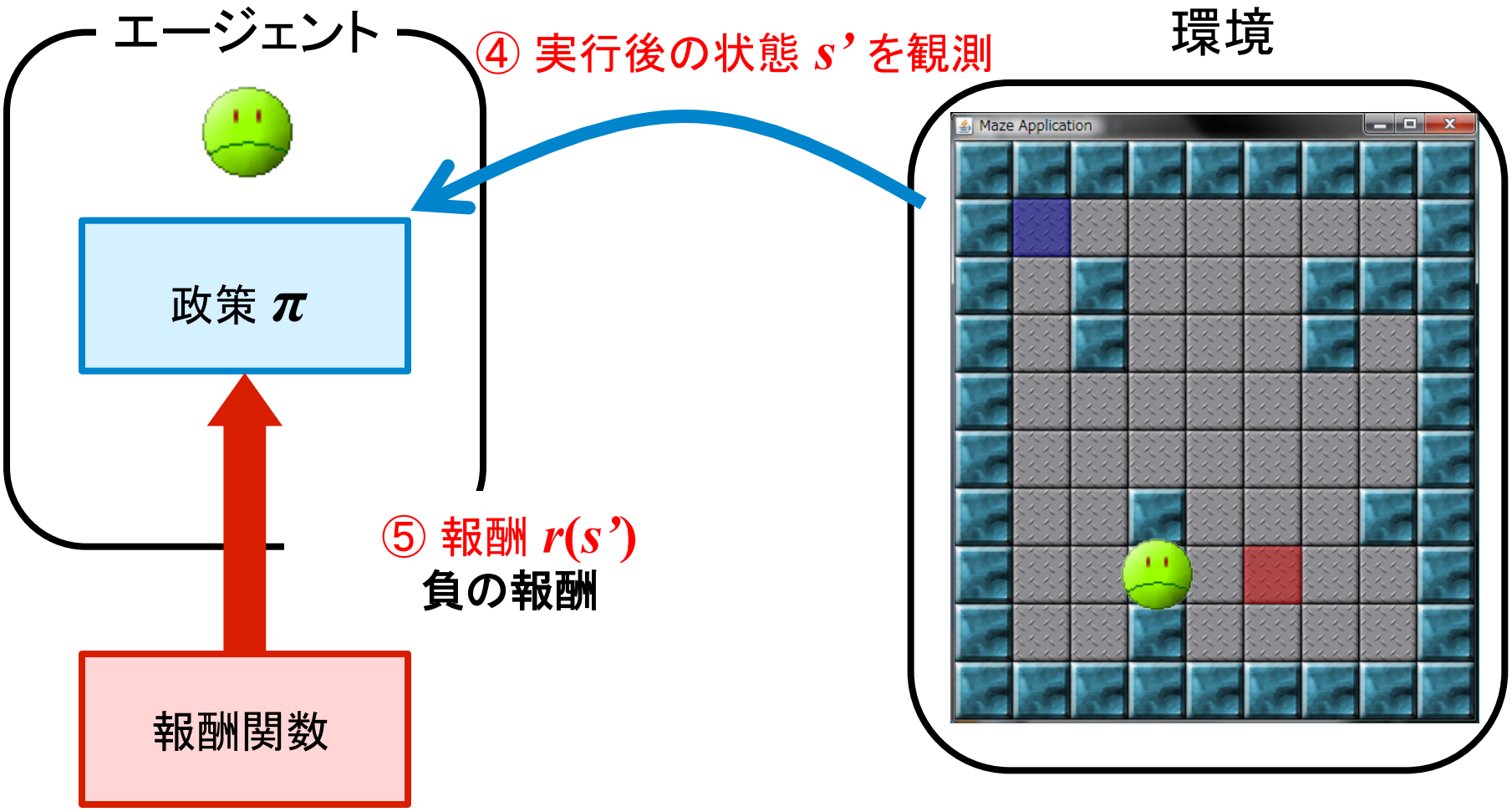
政策と報酬の関係(2)


エージェント
④ 実行後の状態 s’ を観測
環境
Maze Application
政策 π
⑤ 報酬 r(s’)
正の報酬
報酬関数

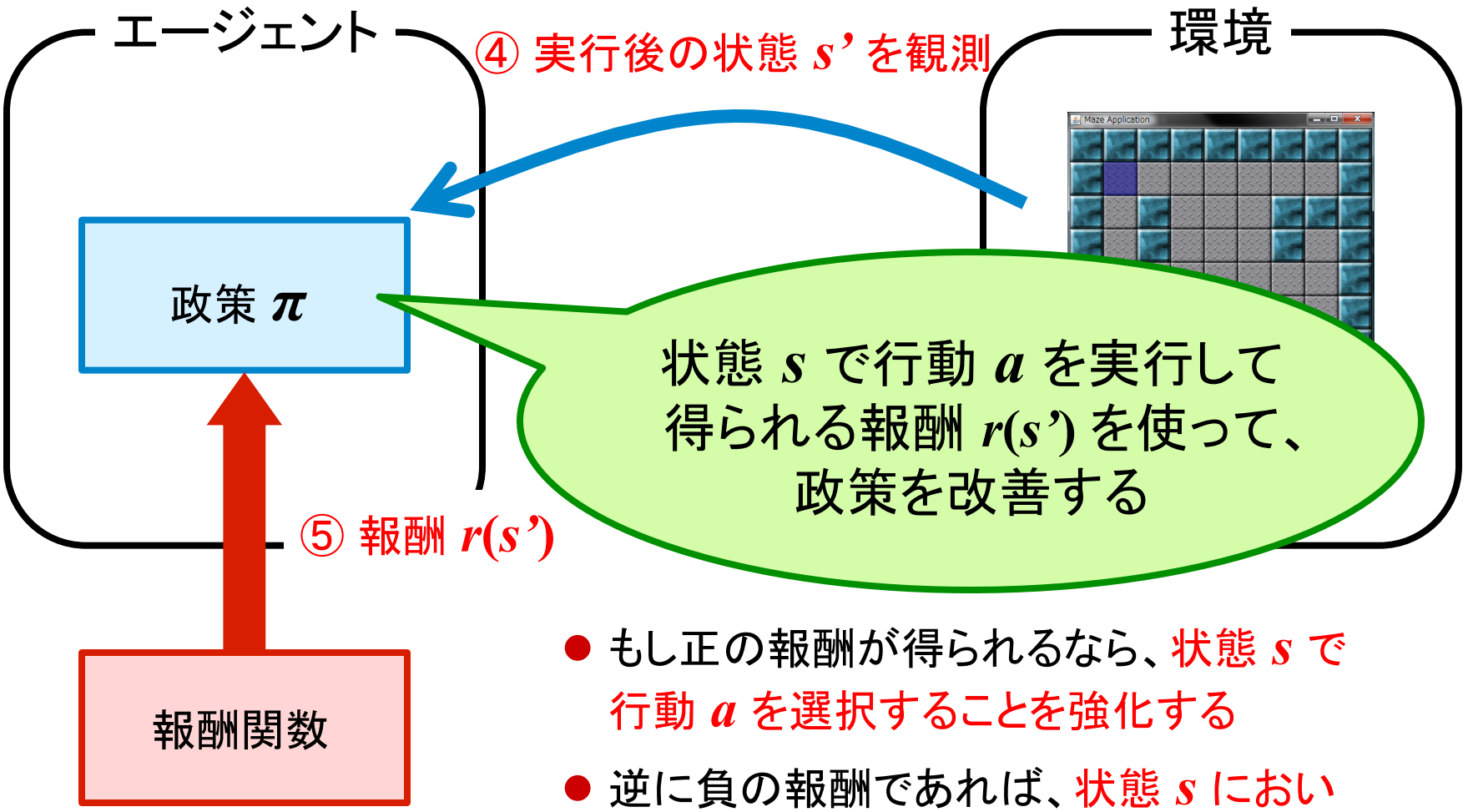
政策と報酬の関係(1')


エージェント
政策 π
② 政策により
行動 a を選択
① 環境の状態 s を観測
③ 行動 a を実行
(左へ移動)
環境
Maze Application

政策と報酬の関係(2')


エージェント
④ 実行後の状態 s' を観測
環境
政策 π
⑤ 報酬 r(s')
負の報酬
報酬関数
Maze Application

政策と報酬の関係(3)



- もし正の報酬が得られるなら、状態 s で行動 a を選択することを強化する
- 逆に負の報酬であれば、状態 s において行動 a が選択されにくくする

政策と報酬の関係(4)

- 状態 s で行動 a を実行した結果...
  - もし正の報酬が得られるなら、状態 s で行動 a を選択することを強化する
  - 逆に負の報酬であれば、状態 s において行動 a が選択されにくくする

ある状態において、どの行動を選択するのが良いのか

を表す行動の評価値を導入する

行動の評価値: Q 値

- $Q(s,a)$
  - 状態 s における行動 a の評価値
  - 正確には、状態 s で行動 a を実行することにより、今後獲得可能な報酬の総量の期待値
    （まさに行動の「良さ」を表している）

強化学習では、この Q 値を徐々に学習していく

Q 値のテーブル: 初期状態

行動

状態

	a_1	a_2	a_3	a_4	....	a_m
s_1	0	0	0	0		0
s_2	0	0	0	0		0
s_3	0	0	0	0		0
s_4	0	0	0	0		0
⋮						
s_n	0	0	0	0		0

$Q(s,a)$

状態 s で行動 a を実行することにより、今後獲得可能な報酬の総量の期待値

初期状態では、$Q(s_i, a_j) = 0$

（つまり Q 値は不明）

Q 値のテーブル: 最終状態

行動

状態	a_1	a_2	a_3	a_4	....	a_m
S_1	10	3	87	-5		17
S_2	32	5	2	78		0
S_3	67	13	23	9		20
S_4	0	-5	94	43		2
⋮						
S_n	17	42	8	32		102

$Q(s,a)$

状態 s で行動 a を実行することにより、今後獲得可能な報酬の総量の期待値

- 例えば $Q(s_1, a_3)$ は、状態 s_1 で行動 a_3 を実行すると、今後報酬を平均して 87 獲得可能であることを表す
- 学習完了後は、各状態において最大の報酬が得られる行動（赤色のマス）を実行すればよい

Q 学習のアルゴリズム

(1) 全ての状態 s と行動 a に対して、$Q(s,a)$ の値を 0 で初期化

(2) 現在の状態が s であるとき、ε-greedy 戦略で行動 a を選択し、それを実行する

(3) 遷移先の状態 s' を観測し、$Q(s,a)$ の値を次式で更新する

$$Q(s,a) \leftarrow Q(s,a) + \alpha[(r(s') + \gamma \max_{a'} Q(s',a')) - Q(s,a)]$$

(4) s' を s としてステップ **(2)** へ戻る

Q 学習のアルゴリズム

(1) 全ての状態 s と行動 a に対して、$Q(s,a)$ の値を 0 で初期化

(2) 現在の状態が s であるとき、ε-greedy 戦略で行動 a を選択し、それを実行する

(3) 遷

$Q(s,$

(4) s'

- ε-greedy 戦略
  - 確率 ε（0.0 ≦ ε ≦ 1.0）でランダムに行動を選択する
  - それ以外では、Q 値が最大の行動を選択する．もしそのような行動が複数あるならば、それらからランダムに選択する

Q テーブルの各マスを埋めるためにもある程度のランダム性は必要

Q 値の時間差分方程式

学習率　　割引率

$$Q(s,a) \leftarrow Q(s,a) + \alpha[(r(s') + \gamma \max_{a'} Q(s',a')) - Q(s,a)]$$

報酬

状態 s' で最大の価値を持つ行動 a' の評価値

状態 s で行動 a を実行することにより、今後獲得可能な総報酬の新しい期待値

新しい期待値と古い期待値との誤差

学習率 α $(0.0 \le \alpha \le 1.0)$

- $\alpha = 0.0$: 学習しない
- $\alpha = 1.0$: 新しい評価値を信じる

割引率 γ $(0.0 \le \gamma \le 1.0)$

- $\gamma = 0.0$: 目先の利益を追求
- $\gamma = 1.0$: 将来性も考慮

最適政策

最終的に受け取る
総報酬の期待値が最大となる政策

Q 学習では、十分に学習を行ったのちに、
各状態においてもっとも大きい Q 値を
とるような行動を選択することで最適政策となる

これを greedy 戦略という

最適政策

最終的に受け取る
総報酬の期待値が最大となる政策

	a_1	a_2	a_3	a_4	……	a_m
S_1	10	3	87	-5		17
S_2	32	5	2	78		0
S_3	67	13	23	9		20
S_4	0	-5	94	43		2
⋮						
S_n	17	42	8	32		102

- 学習完了後は、各状態において最大の報酬が得られる行動（赤色のマス）を実行すればよい

強化学習で迷路を解く

- 次回

- Special thanks:
  - 山本 泰生先生
  - 鍋島 英知先生