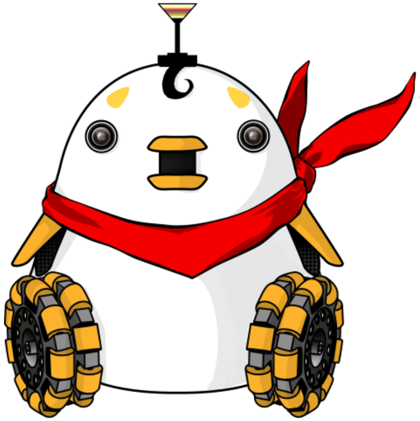
人工知能

第5回 多段決定(1) 動的計画法

立命館大学 情報理工学部 知能情報学科
萩原良信



STORY 多段決定(1)

- 常に状態や状態間のコストが変わらず，ゴールが一つであればA*アルゴリズムでゴールに向かうことができる．しかし，実際にホイールダック2号がとるべき行動は脇目もふらずにゴールに向かうことだろうか．
- ある時刻に現れるアイテムを途中で確保しないといけないし，ある時刻で通りかかる敵を避けないといけないかもしれない．また，ゴールもいくつか存在しえるだろうし，その中でも最も「お得な」ゴールにたどり着くべきだろう．しかし，だからといってすべての行動パターンを試していたのではとてもやっていられない．さてどうすべきか．

仮定　多段決定(1)

- 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.
- 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号は各時刻で各状態間の移動にかかるコストや利得を知っているものとする.
- 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.

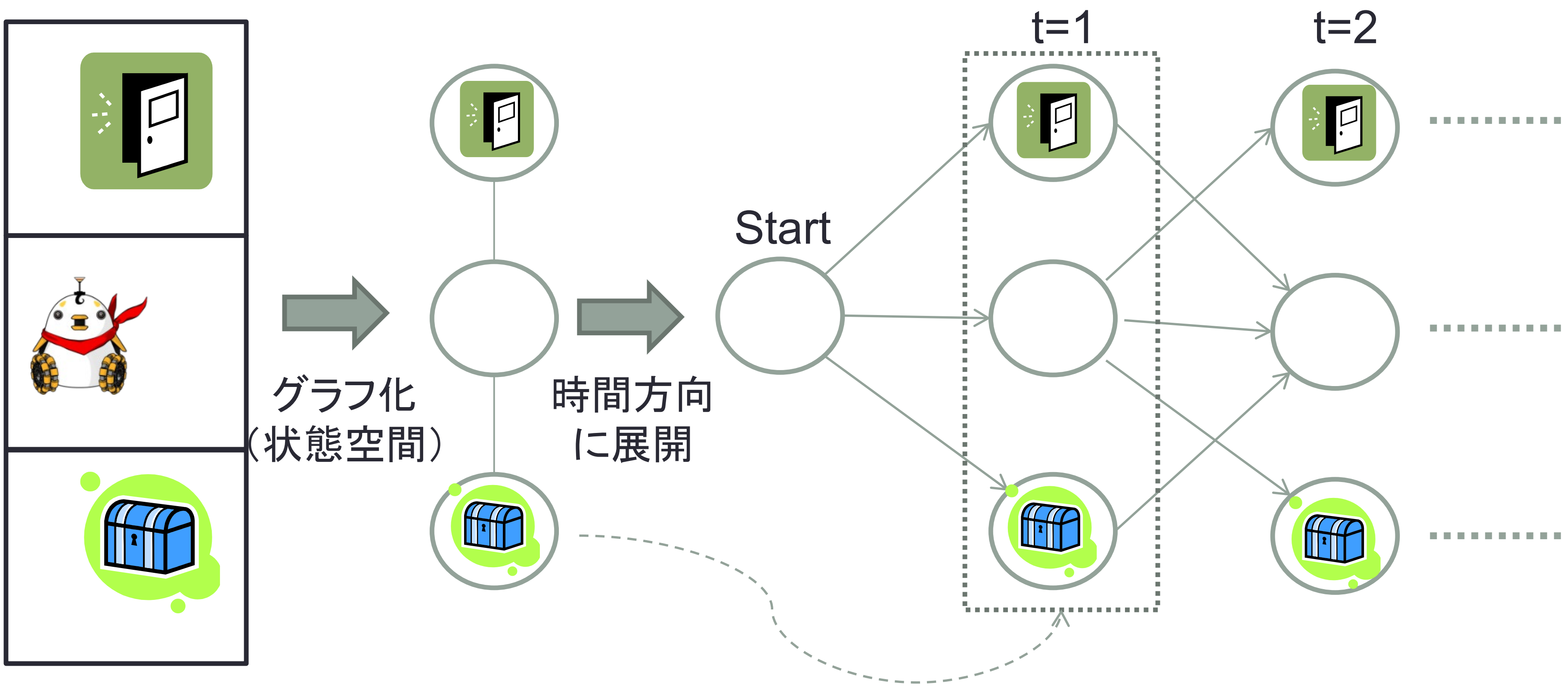
Contents

5.1.1 はじめに

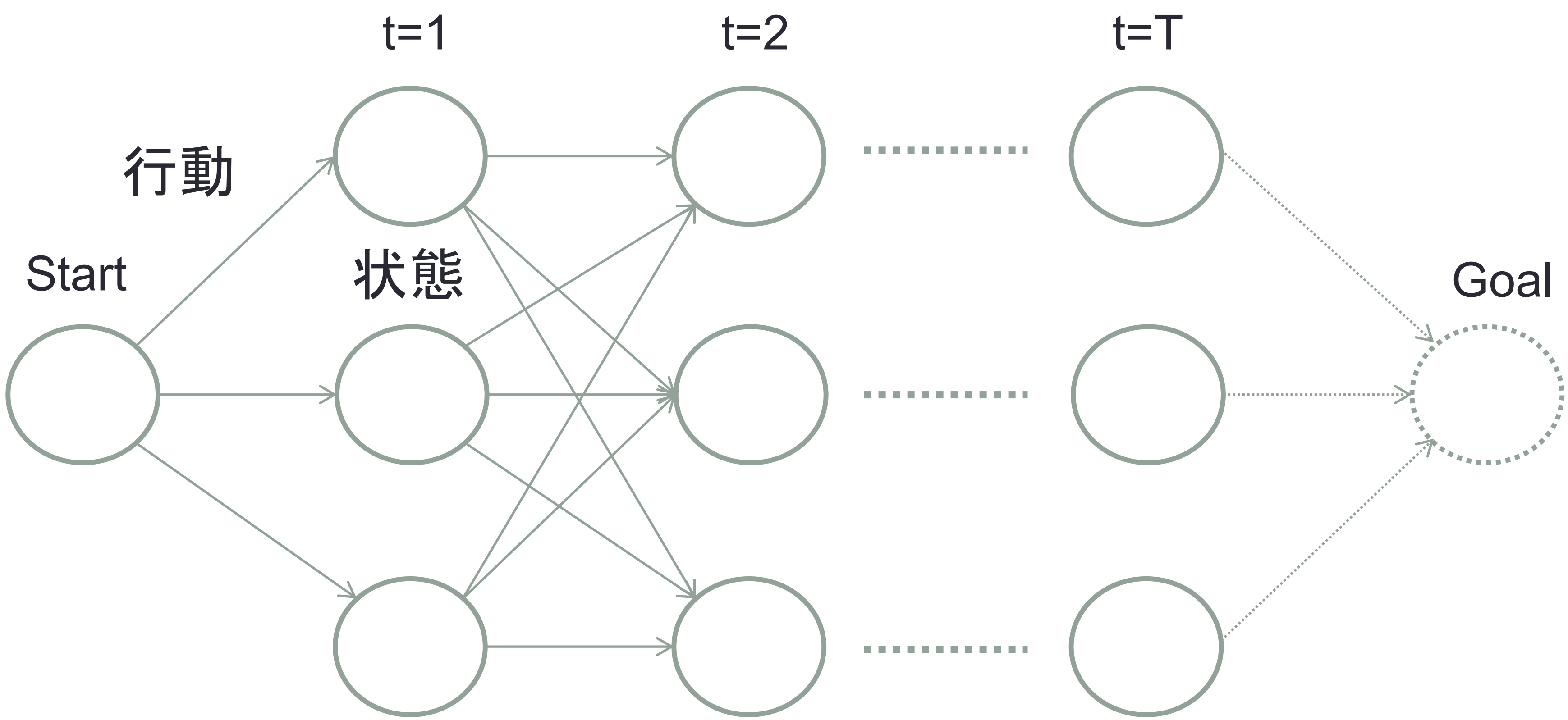
- 時間軸のある意思決定問題を考える．ある時点t で選択した行動が次の時点t+1 の状態を決め，時点t +1 での行動が時点t +2 での状態を決める．
- その上で，各時点での行動選択にもとづいて利得，もしくは費用が発生する．このようなときに時刻T までにかかる費用の和の最小化，もしくは，得られる利得の和の最大化を行う計画問題を**多段決定問題**という．

$$J(s_1, s_2, \ldots, s_T) \longrightarrow \max$$

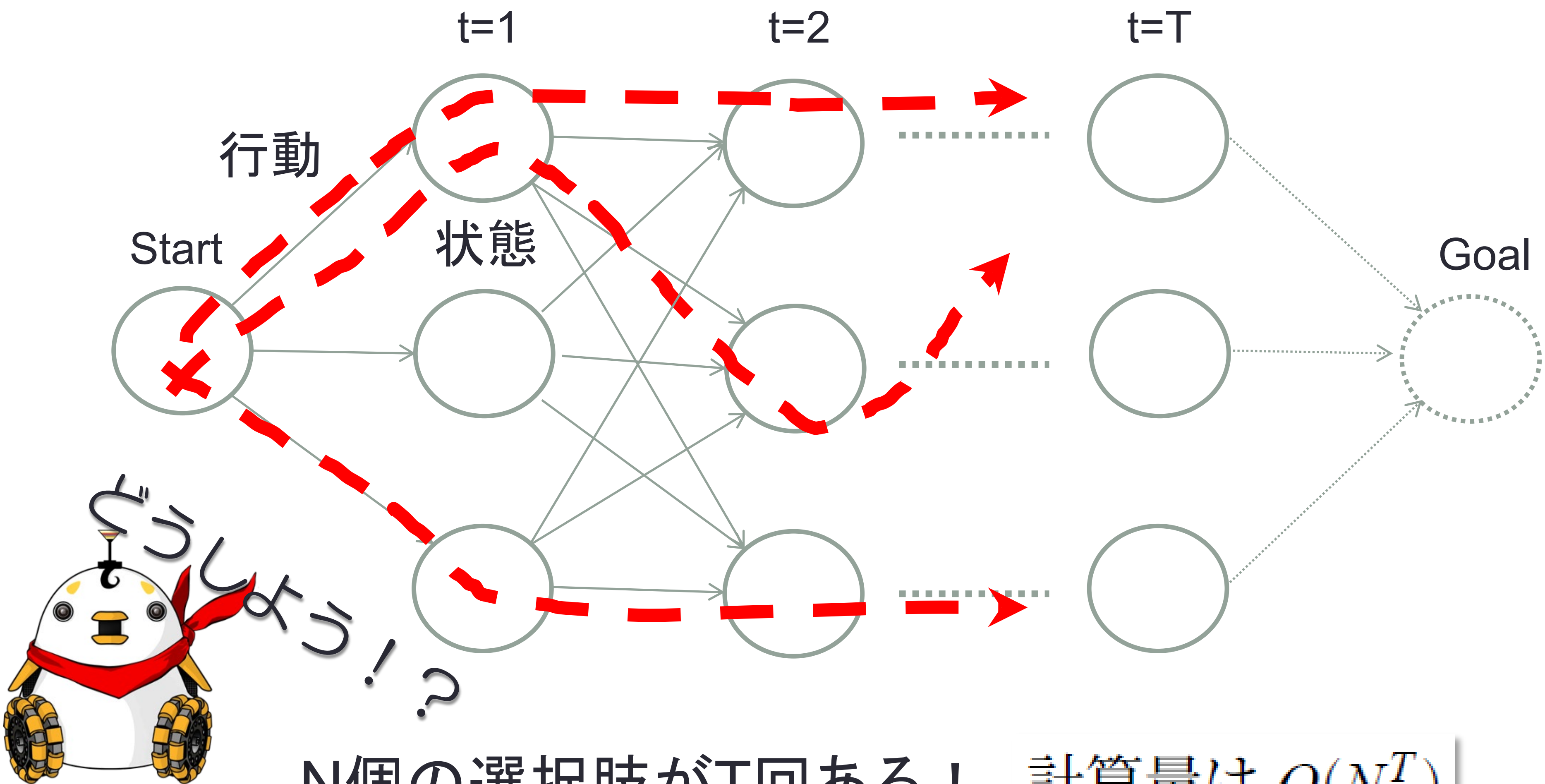
5.1.2 グラフを時間方向に展開する



5.1.2 多段決定問題のグラフ表現



あらゆる経路を列挙的に探索する



N個の選択肢がT回ある！　計算量は $O(N^T)$

Contents

- 5.1 多段決定問題
- 5.2 動的計画法
- 5.3 ホイールダック2号「宝箱を拾ってゴール」
- 5.4 例：編集距離の計算

5.2.1 経路と計算量

- この経路の評価関数をJ とすると．これを最大化することが経路探索の目的となる．

計算量爆発！

$$J(s_1, s_2, \ldots, s_T) \longrightarrow \max$$

計算量は $O(N^T)$

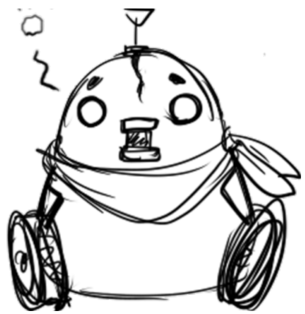
- 動的計画法は多段決定問題において，各評価値が状態の対ごとの二変数関数の和で書けることを利用してこれを効率化するアルゴリズムである．

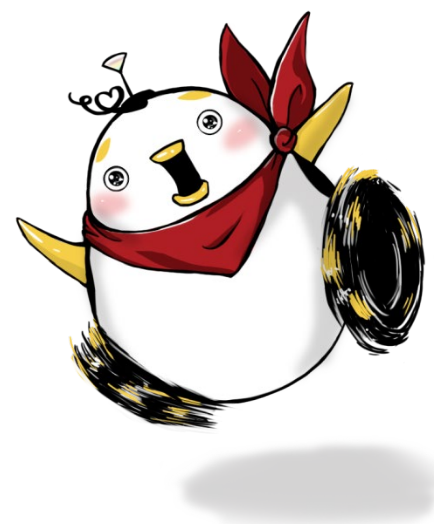
$$J(s_1, s_2, \ldots, s_T) = \sum_{t=1}^{T} h_t(s_{t-1}, s_t)$$

計算量を $O(N^2T)$ まで縮減

指数オーダー⇒2次オーダーのインパクト

- N=100状態，T=34ステップの場合
  - $O(N^T)$
    - 1無量大数回
⇒現実的には終わらない.
  - $O(N^2T)$
    - 34万回
⇒数GHz=数十億Hz　の
　CPUならあっという間

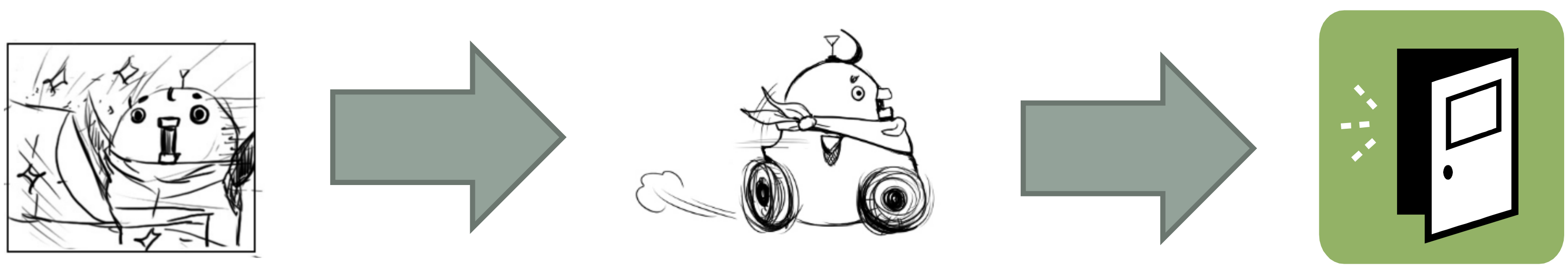




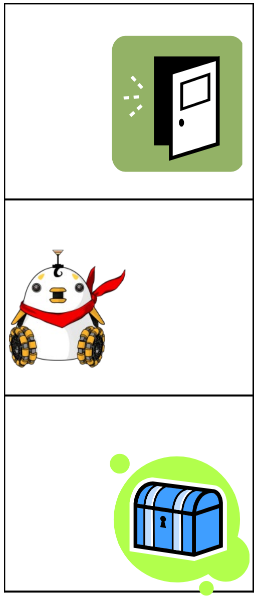
Contents

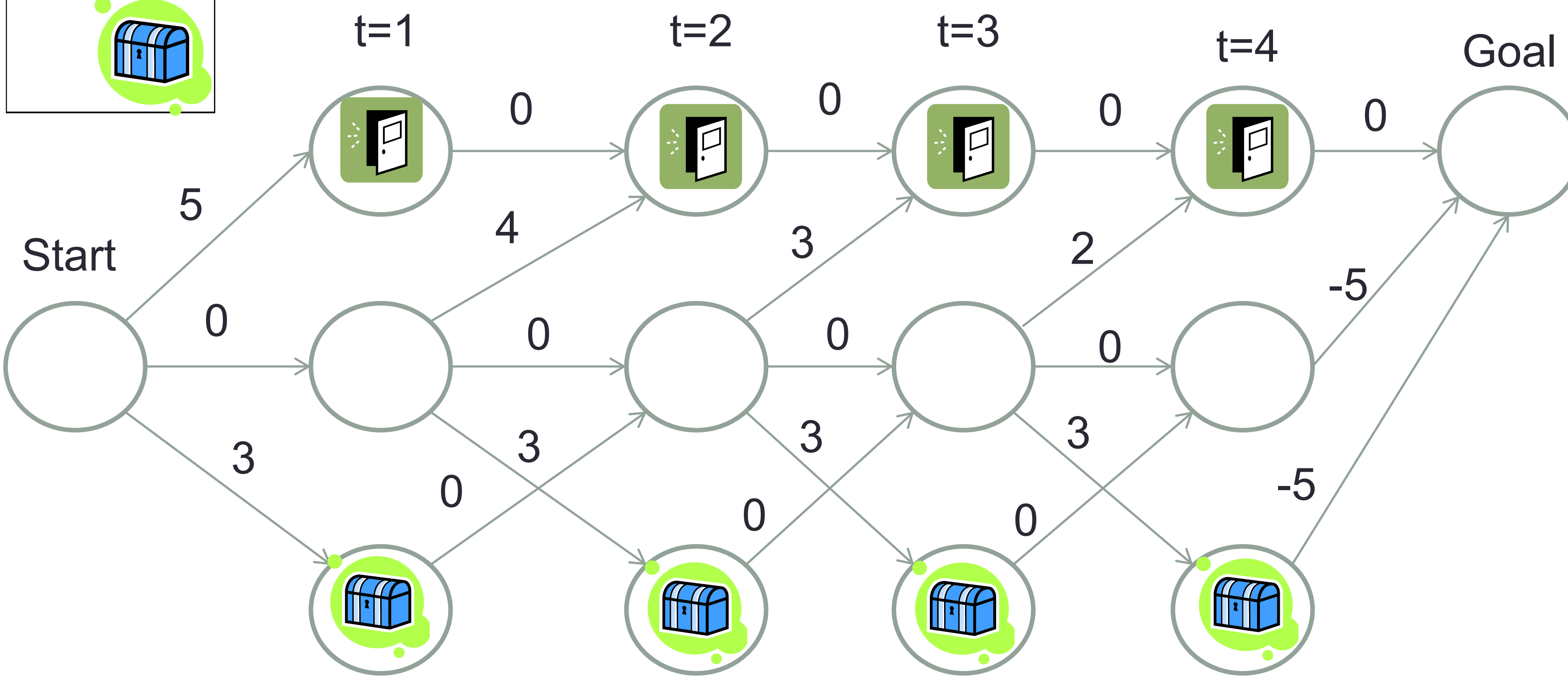
例:「宝箱を拾ってゴール」

- t=4までにゴールできなかった場合，ペナルティとして5の利得を没収される.
- 宝箱をとることは何度でもでき，この時には3の利得を得る.
- また，早くゴールしたほうが利得は高く，ゴールが一時刻遅れるたびにゴール時の利得(5)は減っていく.
- 宝箱の場所にはとどまることはできない．また，一度ゴールすると，ゴールから再度出てくることはできない.



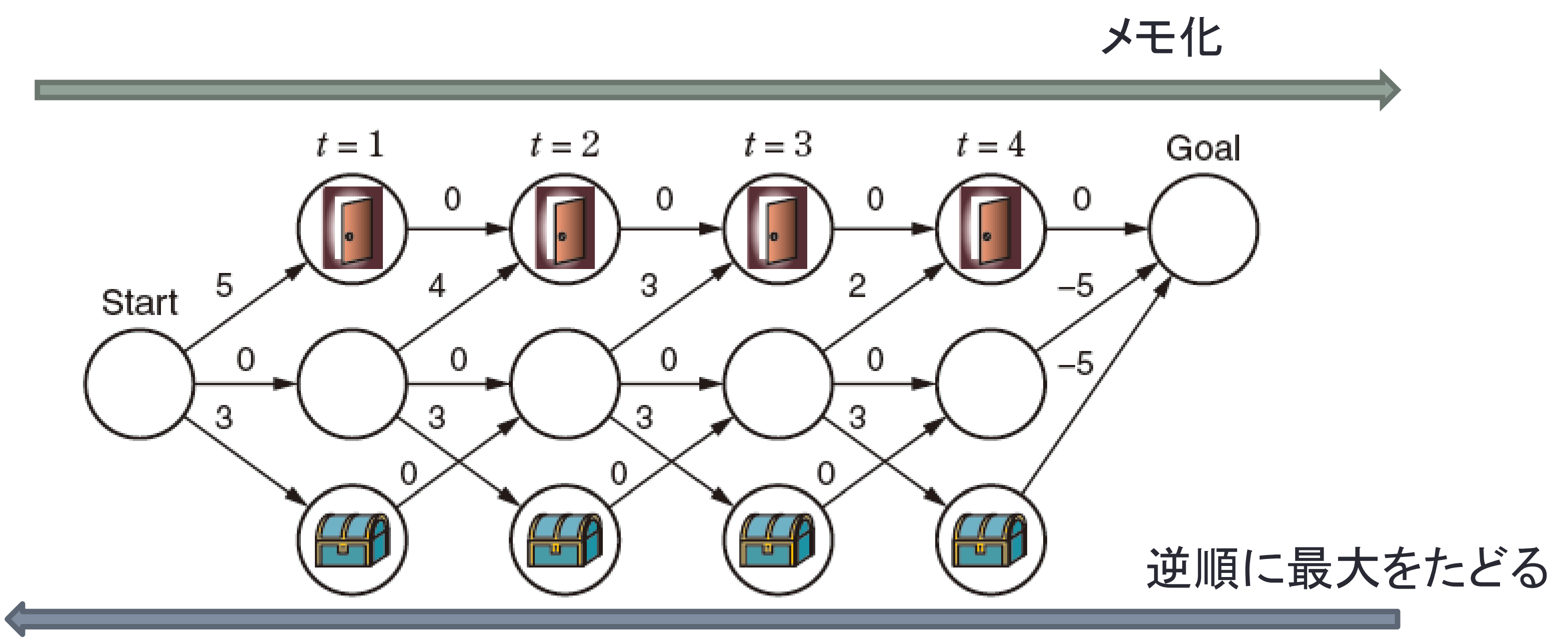
例:「宝箱を拾ってゴール」



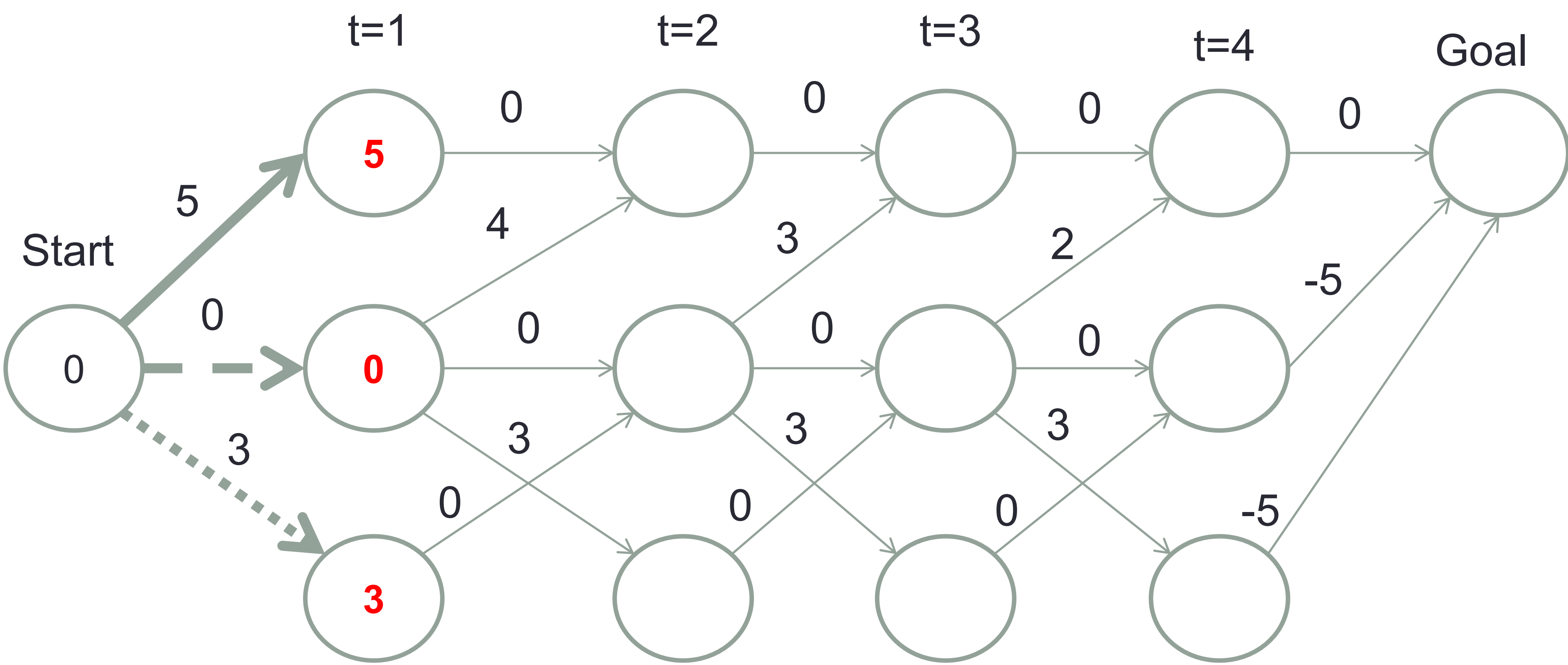


(ポイント) 動的計画法のアルゴリズム

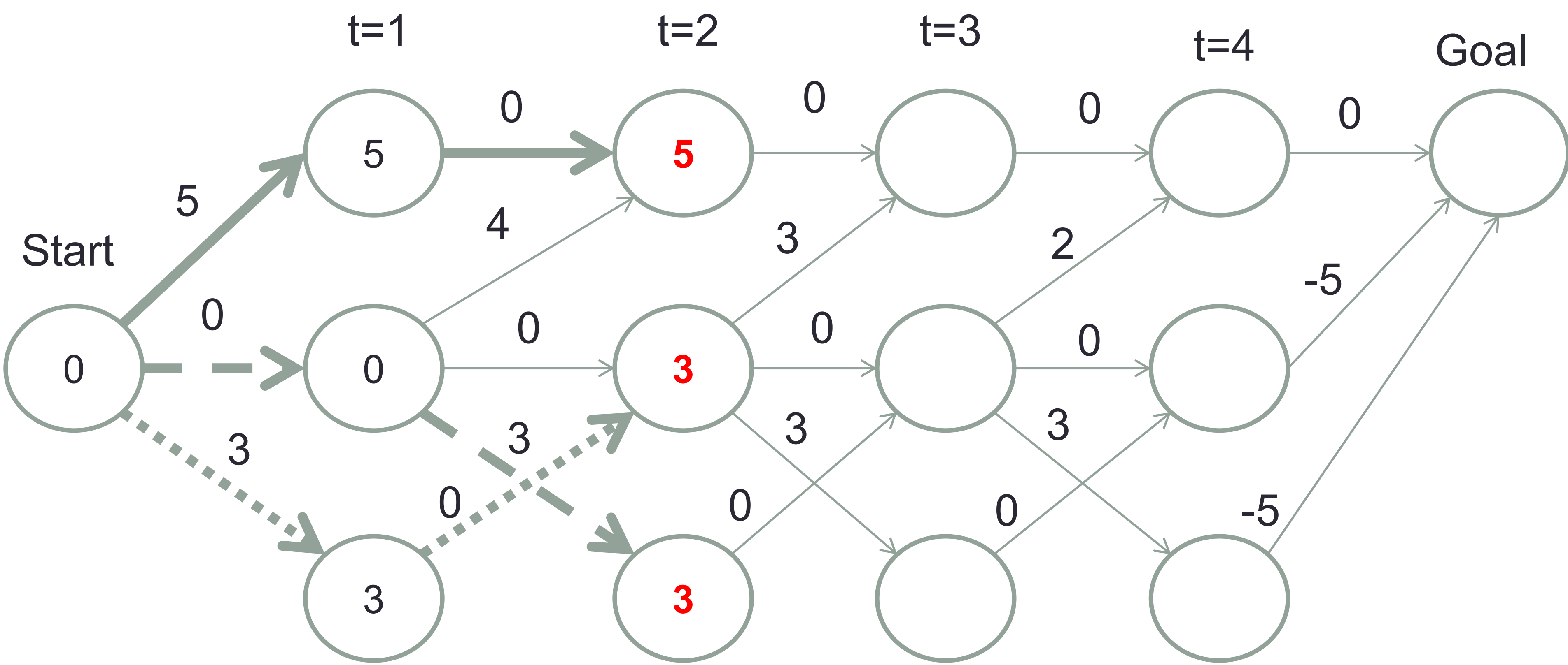
- まず，左から順に各状態までの最適パスを計算し，その時の評価値を状態に記述していく．これを<u>メモ化(Memoization)</u>という．これを繰り返していくことで，最終時刻に至った段階で，これを逆順にたどることで最適なパスがひと通りに決まる．



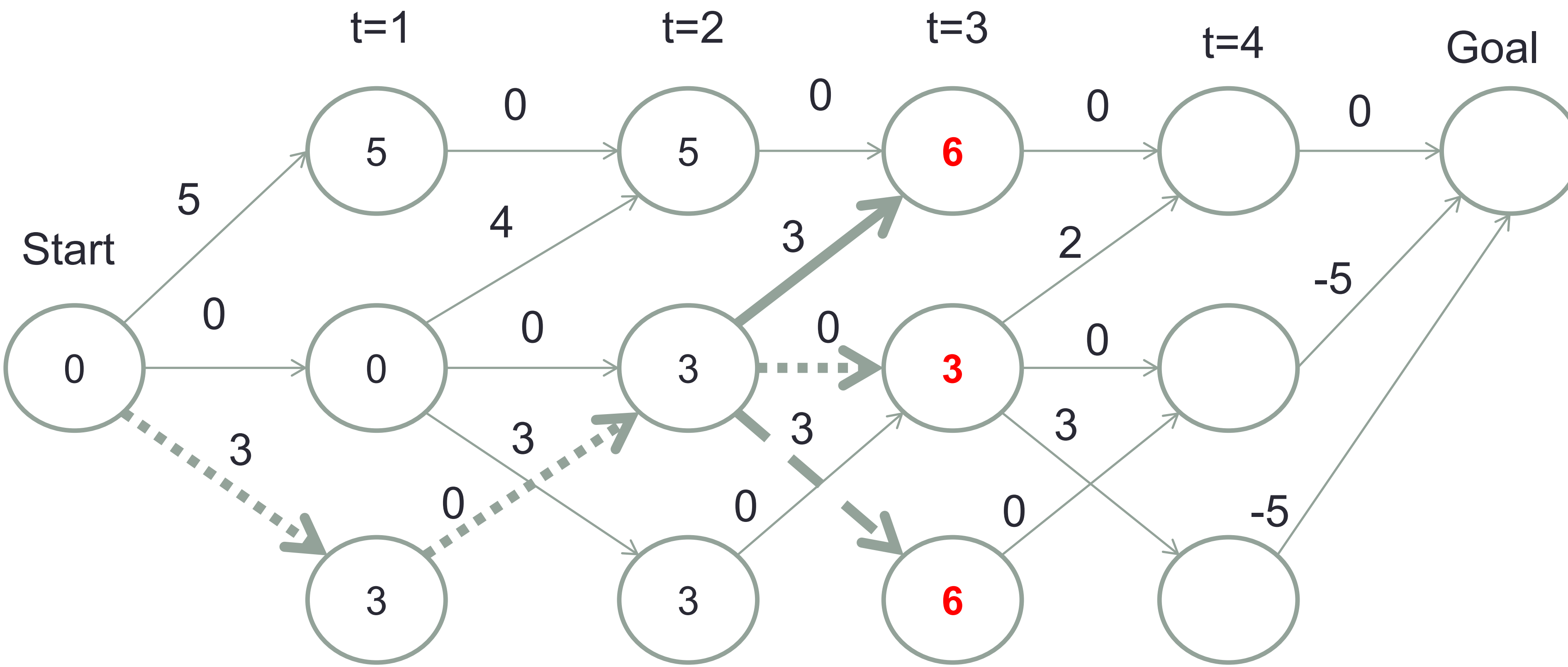
Step 1



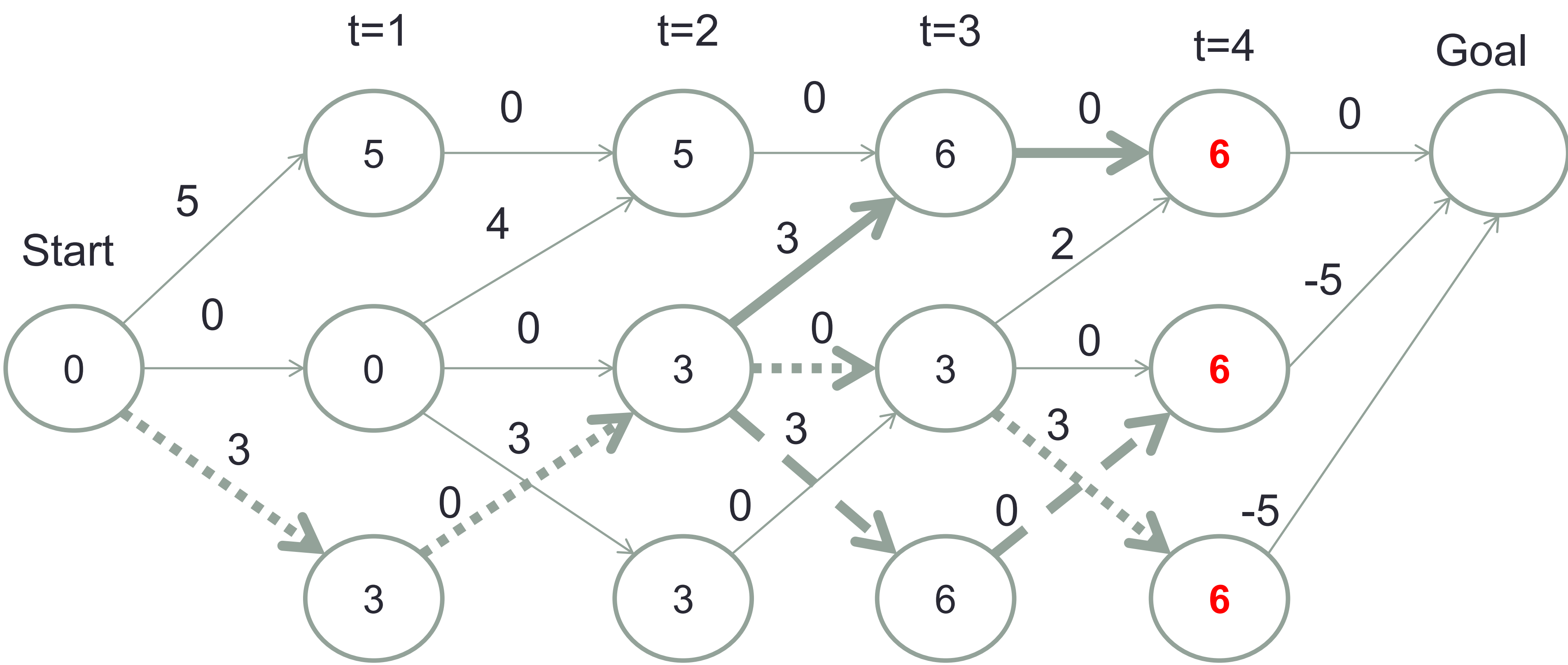
Step 2



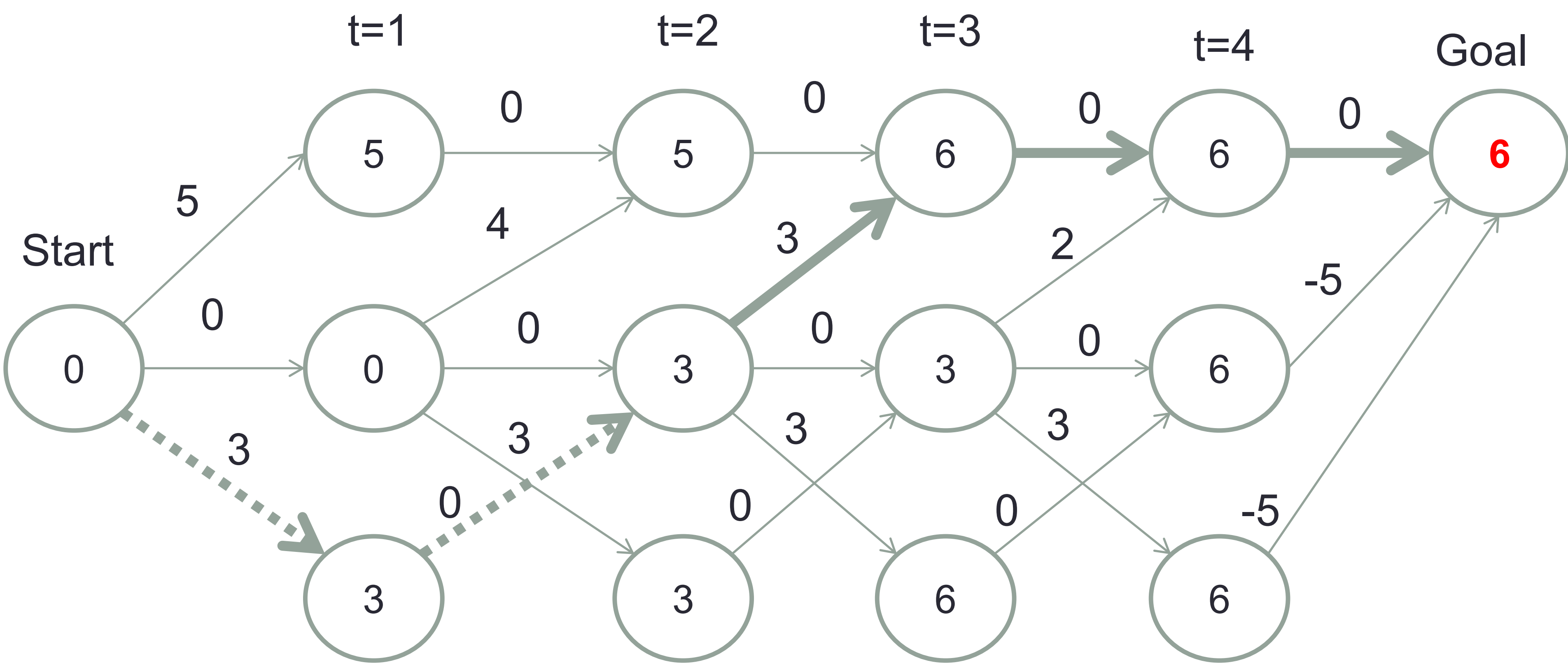
Step 3



Step 4

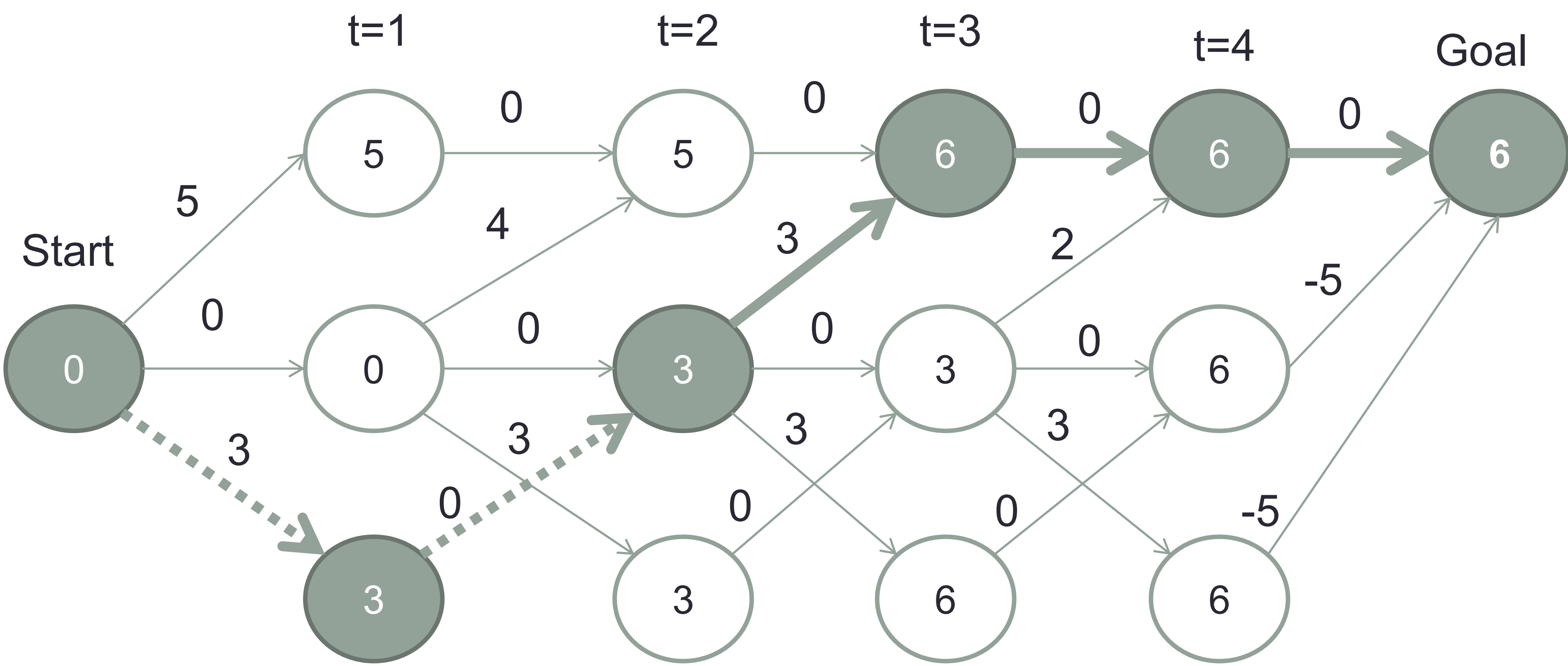


Step 5



t=1
t=2
t=3
t=4
Goal
Start
0
5
0
3
5
0
3
0
4
0
3
0
5
0
3
3
3
0
6
0
3
3
2
0
6
3
0
3
6
0
0
-5
6
6
-5
6
6

最適経路



t=1
t=2
t=3
t=4
Goal
Start
0
5
0
3
5
0
3
0
4
0
3
5
0
3
0
3
3
6
0
3
0
3
6
6
2
0
3
0
6
-5
-5
6
0
6

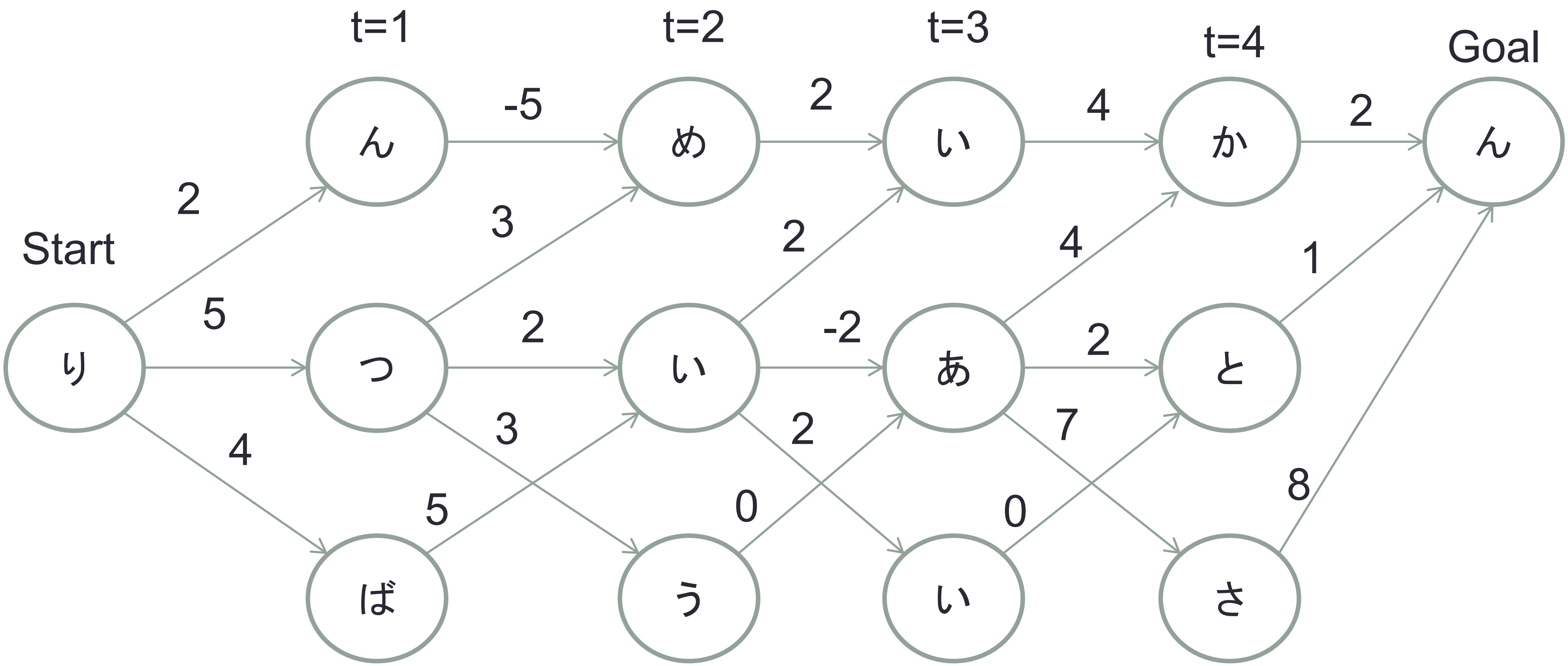
5.2.2 動的計画法のアルゴリズム

Algorithm 5.1 動的計画法

1. for $t = 1$ to T do
2. $$F_t(s_t) = \max_{s_{t-1}}[F_{t-1}(s_{t-1}) + h_t(s_{t-1}, s_t)] \qquad (5.4)$$
   および，その最大値を与える s_t を $\hat{s}_{t-1}(s_t)$ としてメモリに保持する．
3. end for
4. $F_T(s_T)$ を最大にする s_T の値 s_T^* を探索し，その最大値を $J^* \leftarrow F_T(s_T^*)$ とする．
5. for $t = T-1$ to 1 do
6. $s_t^* = \hat{s}_t(s_{t+1}^*)$ を計算する．
7. end for
8. return 経路 $(s_1^*, s_2^*, \ldots, s_T^*)$ および J^* を返す．

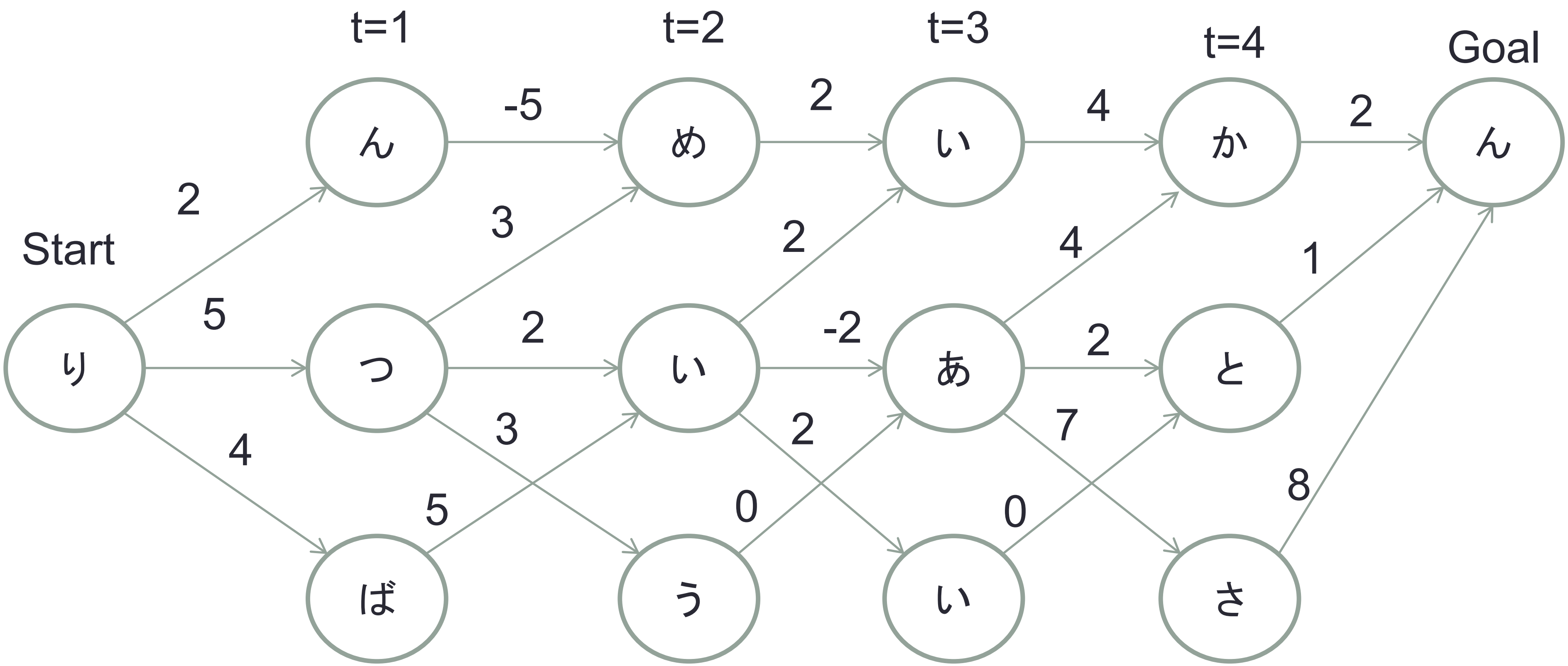
メモ化

演習問題5−1
文字bi−gramによる単語生成



文字のつながりの利得がリンク上に示してある.
最も得点の高くなる経路を見つけよ.

演習問題5-2 アルゴリズムの確認

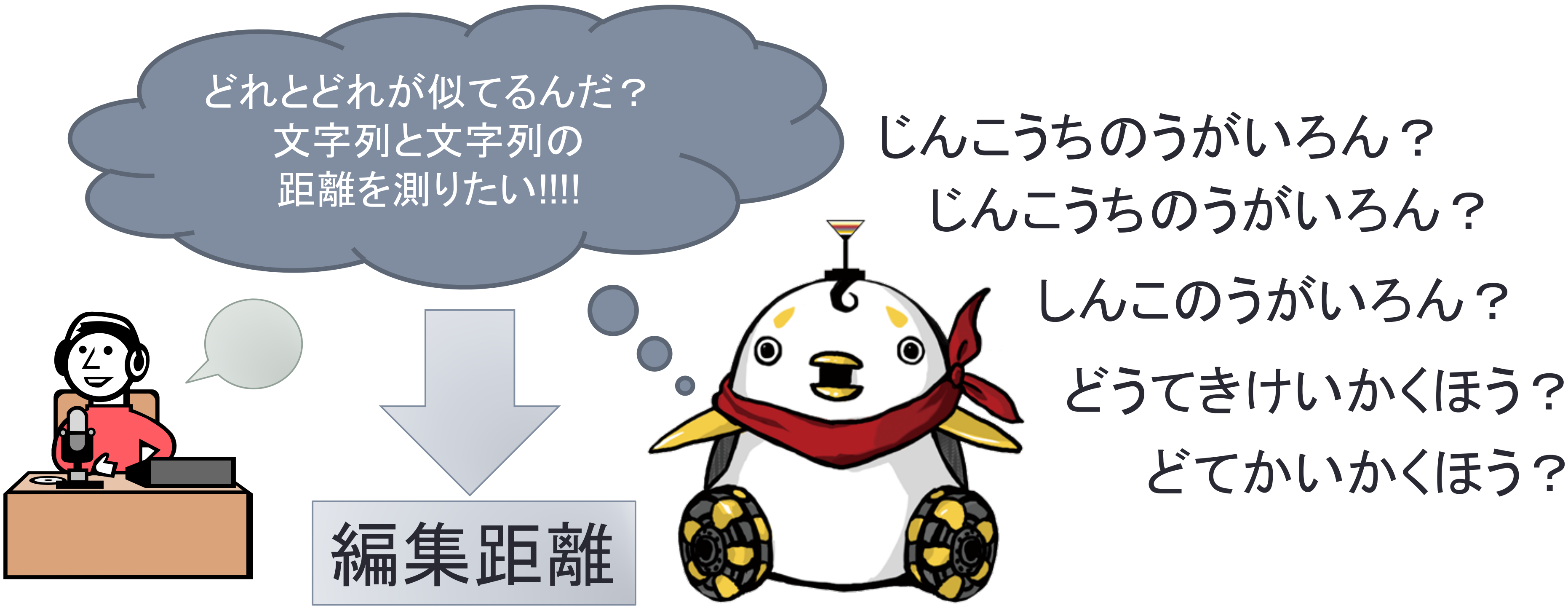


動的計画法のアルゴリズムと演習問題5-1の結果を比較し，最終的なメモリに格納された$F_t(s_t)$とs_{t-1}^(s_t)のリストはどのようになっているか，示せ．

Contents

5.4.1 編集距離の計算

- 動的計画法は確定的システムにおける多段階決定の一般的な解法である.
- ロボットの移動のみならず，様々な多段階決定問題に帰着されうる問題に対して利用することが出来る.



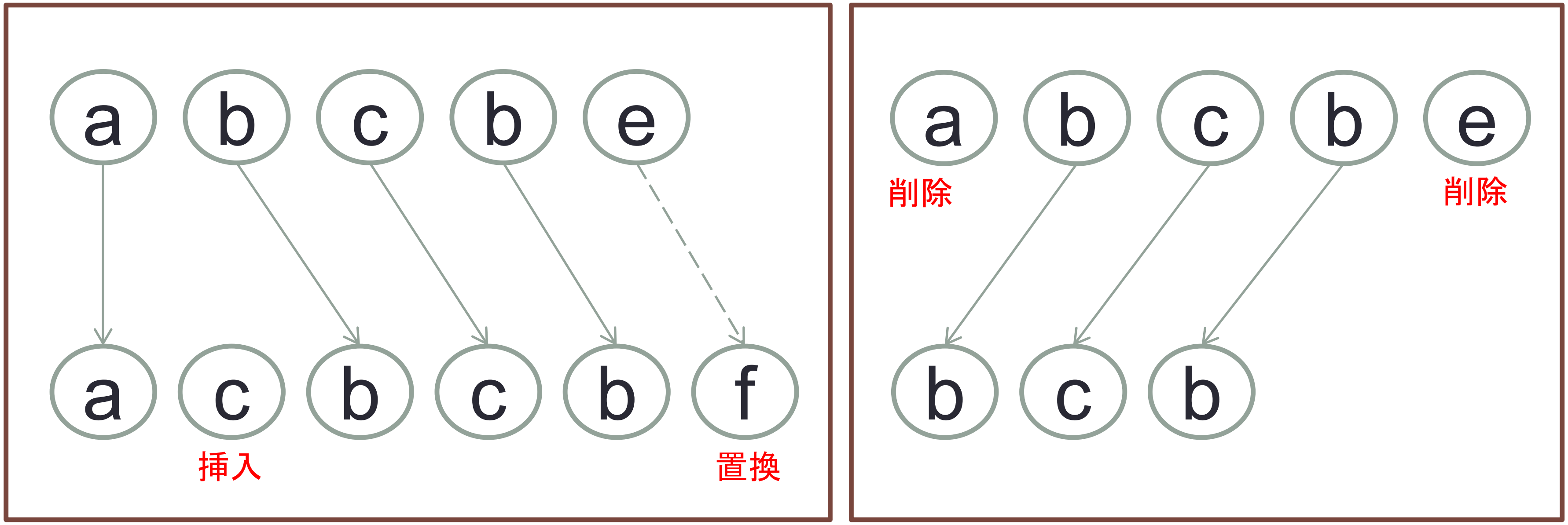
例：編集距離の計算

可插入删除

- 編集距離(edit distance) は文字列と文字列の尺度
- ハミング距離（何文字異なっているか）では文字の置き換えには対応できるが，文字の挿入や削除に対応できない.

文字列 1	文字列 2	ハミング距離	編集距離
abcd	abcd	0	0
abcd	abed	1	1
abcd	acd	3	1
abcd	axbcd	4	1

ストリングマッチング



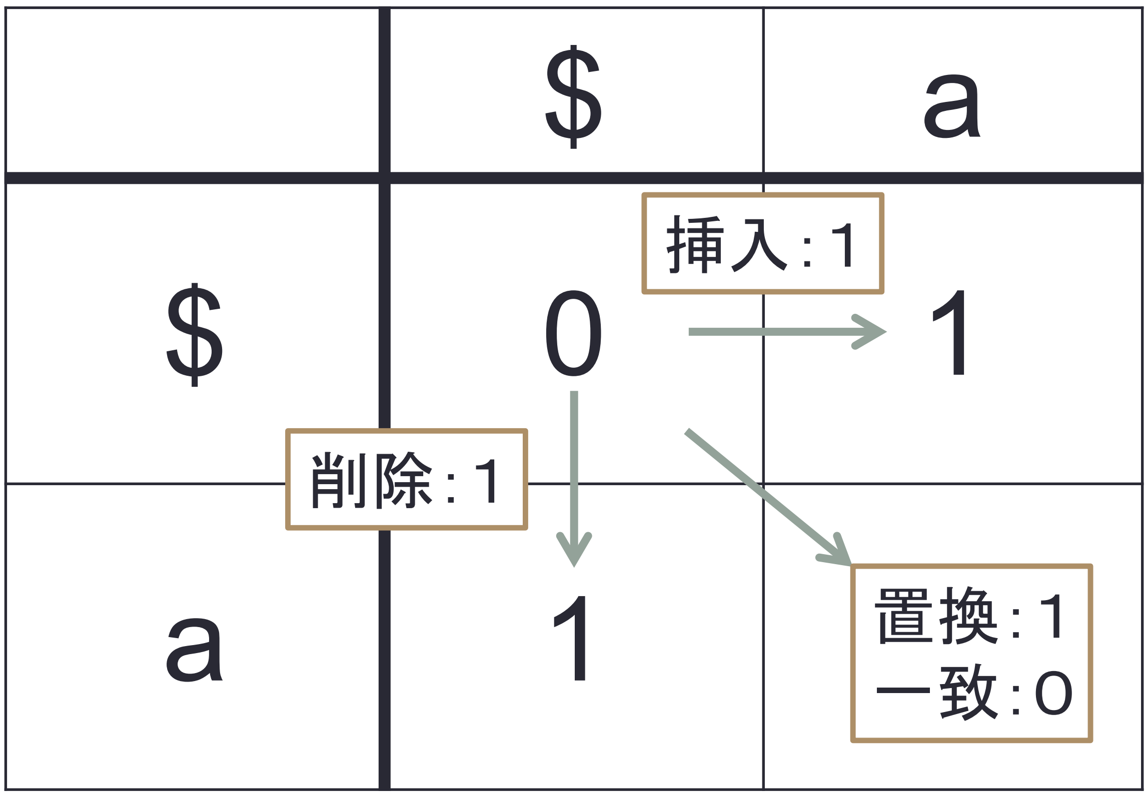
編集距離を計算するための表

Edited String

Original String

	$	a	e	b	c
$	0	1	2	3	4
a	1				
b	2				
c	3				
d	4				Goal

編集距離を計算時の各移動コスト



編集距離の計算結果

Edited String

Original String

	$	a	e	b	c
$	0	1	2	3	4
a	1	0	1	2	3
b	2	1	1	1	2
c	3	2	2	2	1
d	4	3	3	3	2

“e”の挿入

“d”の削除

演習問題 5-3

- 「りつめいかん」と「はつめいか」の編集距離を動的計画法を用いて求めよ.

	$	は	つ	め	い	か
$	0	1	2	3	4	5
り	1					
つ	2					
め	3					
い	4					
か	5					
ん	6					Goal

第5章のまとめ

- 確定システムにおける多段決定問題の定式化を行った.
- 状態空間の時間方向へのグラフ展開について学んだ.
- 動的計画法のアルゴリズムについて学んだ.
- 動的計画法の応用として，ストリングマッチングと編集距離の計算方法について学んだ.