

3010 人工知能: 宿題 2

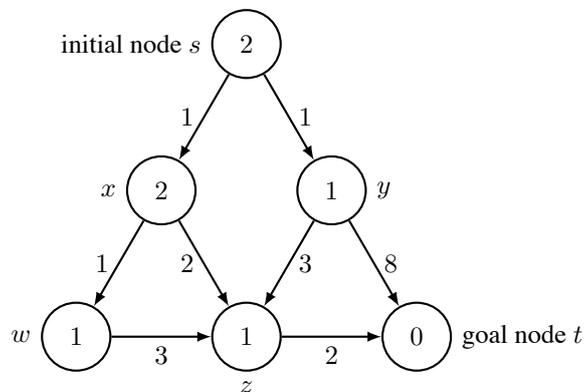
締切: 2019 年 6 月 3 日 (月) 午後 5 時

以下の問題 1-2 に答えよ (裏面にも問題があるので注意). 6 月 3 日午後 5 時までには情報事務室前の宿題提出箱に投函すること. レポートは英語・日本語のいずれかで記述してもかまわない.

注意: 以下では, 次の用語定義にしたがう.

- ヒューリスティック関数 h がある状態空間グラフにおいて適格 (admissible) である, とは, $0 \leq h(v) \leq h^*(v)$ が全ての節点 v で成り立つことを指す. ここで $h^*(v)$ は v から目標節点への経路のうちコスト最小のもののコストを表す.
- ヒューリスティック関数 h がある状態空間グラフにおいて単調 (monotone) である, とは, (i) グラフ中の全ての辺 (v, u) に対して $h(v) \leq h(u) + c(v, u)$ が成り立ち, かつ (ii) $h(t) = 0$ が全ての目標節点 t で成り立つことを指す.

Question 1



上の状態空間グラフは 6 節点 (s, x, y, z, t) からなり, このうち s が出発節点, t が目標節点である. 各節点内に書かれた数値は, その節点のヒューリスティック評価関数 h の値を表しており, また, 各辺の横に書かれた数値は, その辺のコストである. たとえば, $h(s) = 2, h(w) = 1$, であり, 辺 (y, t) のコストは $c(y, t) = 8$ であることが図からわかる. このとき以下の問に答えよ.

1. このヒューリスティック関数 h は単調 (monotone) かどうか, 説明せよ.
2. このヒューリスティック関数 h は的確 (admissible) かどうか, 説明せよ.
3. 上の状態空間グラフに対して, A* アルゴリズム (Figure 1, 次ページ) を実行する. 関数 AStar (左側) の 7-14 行目の繰り返し各回において, 以下を示せ.
 - 8 行目が実行される際に OPEN と CLOSED に, どの節点が入っており, それら節点はどいった g, f 値を持っているか.
 - 10 行目で v としてどの節点を選ばれるか.

Question 2

A* について, 以下の言明はそれぞれ正しいか, 誤りか, 説明せよ.

1. “ヒューリスティック評価関数が適格 (admissible) でなければ, その関数は単調 (monotone) でもない.”
2. “ h を任意の単調なヒューリスティック評価関数とし, $k > 1$ とする. $h'(v) = kh(v)$ なる関数を考える. このとき h' が適格なら, h' は必ず単調である.”

3. “ある状態空間グラフについて, h_1 と h_2 が, いずれも単調な任意のヒューリスティック評価関数であるとき, 各節点 v について $h''(v) = \max(h_1(v), h_2(v))$ と定義された別のヒューリスティック評価関数を考える. このとき, h'' は必ず単調になる.”
(注: $\max(a, b)$ は a と b のうち, 大きい方の値を返す関数である.)
4. “ある状態空間グラフについて, h_1 と h_2 が, いずれも単調な任意のヒューリスティック評価関数であるとき, 各節点 v について, $h'''(v) = h_1(v) + h_2(v)$ と定める. このとき, h''' が適格であるなら, h''' は単調である.

```

1 function AStar( $s$ )
2   OPEN  $\leftarrow$  new PriorityQueue $_f$ 
3    $g[s] \leftarrow 0$ 
4    $f[s] \leftarrow h(s)$ 
5   Insert $_f$ (OPEN,  $s$ )
6   CLOSED  $\leftarrow \emptyset$ 
7   loop do
8     if IsEmpty(OPEN) then
9       return “failure”
10     $v \leftarrow$  DeleteMin $_f$ (OPEN)
11    CLOSED  $\leftarrow$  CLOSED  $\cup \{v\}$ 
12    if IsGoal( $v$ ) then
13      return Solution( $v, s$ )
14    Expand( $v$ )

```

```

1 procedure Expand( $v$ )
2   foreach  $u \in$  Succ( $v$ ) do
3     if  $u \notin$  OPEN  $\cup$  CLOSED then
4        $g[u] \leftarrow g[v] + c(v, u)$ ;  $f[u] \leftarrow g[u] + h(u)$ 
5       Parent[ $u$ ]  $\leftarrow v$ 
6       Insert $_f$ (OPEN,  $u$ )
7     else if  $u \in$  OPEN then
8       if  $g[v] + c(v, u) < g[u]$  then
9          $g[u] \leftarrow g[v] + c(v, u)$ ;  $f[u] \leftarrow g[u] + h(u)$ 
10        Parent[ $u$ ]  $\leftarrow v$ 
11     else
12       if  $g[v] + c(v, u) < g[u]$  then
13          $g[u] \leftarrow g[v] + c(v, u)$ ;  $f[u] \leftarrow g[u] + h(u)$ 
14         Parent[ $u$ ]  $\leftarrow v$ 
15         CLOSED  $\leftarrow$  CLOSED  $\setminus \{u\}$ 
16         Insert $_f$ (OPEN,  $u$ )

```

Figure 1: A* algorithm. OPEN, CLOSED, Parent, g , and f are global variables. See the lecture slides for more detail.