

مأنده نادھی ۹۷۱۲۷۶۲۲۳۸

۱ Heuristic evaluation function

h عمق goal بدست آمده توسط الگوریتم BFS روی مساله ی relax شده (حذف صندوقچه و حیوان وحشی و راهزن) است. برای کم کردن زمان محاسبات heuristic evaluation function اقدام زیر صورت گرفته :

• Memoization

در هر مرحله که برای یک node تابع h محاسبه میشود مقادیر h برای گره های موجود در مسیر بهینه ی بدست آمده توسط الگوریتم BFS نیز در $O(1)$ محاسبه و در یک `Hashtable<String,Integer>` ذخیره میشود تا در مراجعات بعدی ابتدا این `HashTable` بررسی گردد.

```
public void add(Node goal) {
    int depth = goal.getDepth();
    Node tempNode = goal;
    Stack<Node> nodes = new Stack<>();
    while (true) {
        nodes.push(tempNode);
        if (tempNode.parentNode == null) {
            break;
        } else {
            tempNode = tempNode.parentNode;
        }
    }
    while (!nodes.empty()) {
        tempNode = nodes.pop();
        mem.put(tempNode.hash(), depth--);
    }
}
```

۲ Admissible

h حاصل relax کردن مساله است و همواره با اضافه شدن سایر موجودیت ها تنها ممکن است مسیر بهینه ی پیدا شده در h افزایش یابد یا ثابت بماند. (این خاصیت relax کردن است چون برخی از محدودیت ها حذف می شوند). بنابراین h قابل قبول است چون همواره از h^* کمتر اند.

۳ Consistency

دبرای consistency باید اثبات کنیم که: (n هر گره دلخواه و n' هر فرزند دلخواه گره n میباشد)
$$h(n) \leq c(n, a, n') + h(n')$$

هزینه ی هر اکشن ۱ می باشد بنابراین داریم :

$$\begin{aligned}h(n) &\leq 1 + h(n') \\ h(n) - h(n') &\leq 1\end{aligned}$$

هر گره (n) در مساله را در نظر میگیریم در فرزندان آن agent میتواند به خانه های زیر برود :

زمین، صندوقچه، راهزن، حیوان وحشی، باتلاق، پل، کلید و قلعه : (n')
در هر یک از انتقال های بالا (از n به n') اگر هر دو گره n و n' به تابع h داده شوند خانه های راهزن و صندوقچه و حیوان وحشی آنها را حذف میکند و مسیر بهینه به هدف را می یابد در نتیجه هر حرکت مجاز از n به n' (حرکت به راست یا چپ یا بالا یا پایین) agent را در map مساله ریلکس شده نیز تنها یک واحد جابه جا میکند حال داریم :

اگر n' در مسیر بهینه ی بدست آمده توسط الگوریتم BFS از گره n به goal قرار داشته باشد آنگاه داریم :

$$h(n) - h(n') = 1$$

(چون مسیر بهینه ی از n' به هدف همان طول مسیر منهای یک را دارد چون موجودیت های مپ h با سایر موجودیت ها تداخلی ندارند)

(منظور از مسیر بهینه، مسیر بهینه ی موجود در مپ ریلکس شده ی h است و نه در مساله ی اصلی.)

اگر n' به گونه ای باشد که مسیر n' به goal به موازات مسیر بهینه ی از n به goal بدست آمده توسط الگوریتم

$$h(n') = h(n) \text{ : BFS باشد در این حالت داریم}$$

(چون n' که یکی از فرزندان n است در مسیری بهینه به موازات مسیر بهینه ی n به goal است)

اگر n' گرهی در خلاف جهت مسیر بهینه ی n به goal باشد (یعنی به طور مثال اگر مسیر بهینه ی بدست آمده از

BFS حرکتی روبه پایین و به سمت چپ را تعیین میکند n' گره بدست آمده از اکشن بالا یا راست است) در این حالت داریم :

$$h(n) - h(n') = -1$$

در هر سه حالت بالا شرط $h(n') \leq h(n)$ برقرار است در نتیجه h سازگار است.