

Subject :

Year :

Month :

Date :

سوال اول : در این کلاس دو کلاس دیگر برای شبیه سازی هر node یک

کلاس با این نام تعریف شده که در آن یک لیست با نام table وجود دارد که

نشان دهنده ردیف ها و در هر ردیف لیستی برای نشان دادن کارت ها وجود

دارد. تابع check - goal با گرفتن یک لیست به عنوان ورودی آزمون هدف را

روی آن انجام می دهد. برنامه پس از دریافت ورودی با قرار دادن حالت اولیه در

لیست frontier که یک نوع queue است و به صورت FIFO عمل می کند

الگوریتم شروع به کار کردن می کند و در هر state با بررسی امکان حرکت مورد نظر

فرزند را تولید می کند و والد را هم در parent در کلاس node قرار می دهد.

آزمون هدف هم در هنگام تولید فرزند انجام می شود به این صورت که ابتدا بررسی می شود

در صورت موجود نبودن در frontier و explored

اگر بود جستجو به پایان می رسد و در غیر این صورت state به frontier اضافه

می شود. در نهایت با سطح نهای به همراه رسته والد های خود (به عنوان مسیر)

نمایش داده می شود.

☒ VANDAT

(یک نکته فراموش نشود هر node پس از بسط داده شدن از frontier به لیست explored منتقل می شود)

آرتاسدن

هروزه اول هوش مصنوعی

Subject:

Year:

Month:

Date:

۹۷۳۱۰۰۶

سوال دوم: متابع و کلاسی node مانند سوال اول ساخته شد و اما متغیری

به نام depth در node وجود دارد که هر depth برابر با depth والد به علاوه یک

می باشد. ابتدا node اولیه بررسی می شود و در صورت هدف نبودن limit برابر با

یک قرار داده می شود. یک حلقه تا هنگام پیدا شدن پاسخ ادامه پیدا می کند و در هر

بار اجرا limit یک واحد افزایش می یابد. در هر بار اجرای حلقه درونی با limit مقایسه

یک لیست explored و یک لیست frontier (stack) که به صورت LIFO


عمل می کنند وجود دارد که در ابتدا هنگام اولیه در frontier وجود دارد. آزمون هدف به هنگام

بسط نمودن انجام می شود. اگر هدف نبود فرزندان آن با بررسی امکان تولید، تولید


می شوند و در پشته frontier قرار می گیرند. برای بسط به یک state ابتدا

بررسی می شود که حتماً بحق نمود مورد نظر از limit کمتر باشد در غیر این صورت

(برابر بودن بحق با limit) فرزندان آن تولید نمی شوند.

 Year :

★ Month :

 Date :

*[Handwritten signature]*

دارد که اولی همان تابع ~~مستقیم~~ مستقیم می باشد که ابتدا اعتبار دارد

کمی توانند در جای درست قرار گرفته باشند و این شمارد (ردیف های خالی نیز

امتیاز مثبت حساب می شوند) به این صورت که اولین کارت شماره تعداد کارت های هر

زنگ باشد و همین طور کارت های بعدی نزولی باشند سپس امتیاز بدست آمده از امتیاز

هدف کم می شود و به عنوان  $\Delta$  داده می شود. (و تابع دیرنفر برای بدست آوردن

f و دیگری برای پیدا کردن بهترین node (کوتاهین f) می باشد.

لیست frontier وجود دارد که هر دفعه state با کمترین  $f(n)$  از آن انتخاب

شده و بسط داده می شود و آزمون هدف هنگام بسط نمود انجام می شود



آرتا اسدی  
۹۷۳۱۰۰۴

پروژه اول هوش مصنوعی

Subject:

Year:

Month:

Date:

مقایسه روش ها:

هر سه روش کامل هستند و هر سه آنها روش بهینه را پیدا می کنند

از نظر پیچیدگی فضایی  $ID_3$  از همه بهتر است زیرا  $O(bd)$  است اما  $BK$  و

$A^*$  نهایی هستند و از نظر پیچیدگی زمانی هم  $A^*$  از همه بهینه تر است

زیرا اثر فضای تابع هیورستیک کوچکتر از  $O(h^* \log h)$  باشد پیچیدگی زمانی

نهایی نسبت به برای  $ID_3$  و  $BK$  نهایی است و از آنجا که پیچیدگی زمانی

متناظر با تعداد کل نودهای تولید شده است می توانیم ببینیم که  $A^*$  از همه

کمتر است.