به نام خدا

گزارش پروژه سوم درس مبانی هوش مصنوعی

سید پوریا احمدی ۹۷۲۳۰۰۲

امیر اله ورن ۹۷۲۳۰۱۰

توضیحات کد forward checking :

تابع read_file :

در این تابع فایل (پازل) مورد نظر خوانده شده و مقادیر خوانده شده در ماتریس ذخیره شده و به همراه تعداد ستون ها(که برابر تعداد سطر ها هم هست) برگردانده می شود.

تابع های constraints :

در این توابع محدودیت های برای جدول ما محاسبه می شود .

به ترتیب:

Constraints_row : محاسبه محدودیت برای تعداد ۰ و ۱ ها که تعدادشان برابر و نصف تعداد ستون ها باشد.

Constraints_col : محاسبه محدودیت برای تعداد ۰ و ۱ ها که تعدادشان برابر و نصف تعداد ردیف ها باشد.

Constraints_repeat : چک میکند که سه تا ۰ یا یک پشت هم در جدول نباشد (چه سطری چه ستونی)

Constraint_str : چک میکند که دو ردیف یا دو ستون رشته یکسانی تولید نکنند.

تابع get_possible_values :

در هر مرحله از الگوریتم برای هر خانه خالی جدول مقادیر دامنه را چک میکند که کدام اعداد می تواند در آن خانه قرار بگیرند.

اسم توابع زیر دقیقا کار الگوریتم اصلی backtracking که در کتاب گفته شده را انجام نمیدهند.

Forwardchecking:

در این تابع برای تمام خانه های خالی مقادیر مجاز را پیدا کرده و در dict به صورت سورت شده ذخیره میکند.

Show_table: در هر مرحله جدول را پرینت میکند (امتیازی)

MRV:

در این تابع با توجه به عدد j کار زیر را انجام میدهد.

اگر j=0 باشد یعنی نیازی به backtrack نمی باشد و از لیست سورت شده بالا اولین خانه (یعنی محدود ترین) را انتخاب می کند مقدار دهی کرده و پازل را به روز رسانی میکند.

اگر هم j مخالف صفر باشد یعنی باید backtrack کرد در این صورت از قبل در تابعی دیگر خانه جدول به روز شده و اخرین خانه ای که مقدار دهی شده مقدار - قرار میگیرد و خانه ای دیگر مقدار دهی می شود.

Backtracking :

پس از مراحل اولیه وارد این تابع می شویم و در یک حلقه شروع به پر کردن خانه ها میکنیم اگر تابع به انتها برسد done چاپ می شود و اگر به بن بست بخورد اخرین خانه ک مقدار دهی شده (این مقادیر در لیست u نگه داری می شوند) پاک می شود و u و لیست خانه های مقدار دهی نشده به همراه دامنه آنها به روز رسانی شده و به MRV داده می شود . به همین صورت الگوریتم ادامه پیدا میکند.

توضحات کد MAC :

کلاس CSP:

پیاده سازی قالب CSP در این کلاس انجام شده است. در سازنده این کلاس متغییر ها و دامنه ها و محدودیت ها set می شوند.

متد add_constraint:

به محدودیت های موجود محدودیت اضافه میکند.

متد consistent:

بررسی میکند که ایا اساین کردن یک مقدار به یک متغییر valid هست یا خیر.

متد mrv :

متغییری که کمترین تعداد را در دامنه خود دارد را برمیگرداند.

متد MAC :

با اساین شدن یک مقدار به یک متغییر ، دامنه بقیه متغییر هایی که هنوز مقداری به آنها assign نشده است را به روز می کند.

متد backtracking_search :

این متد الگوریتم backtrack را به صورت بازگشتی اجرا میکند.

کلاس BinaryPuzzleConstraint:

این کلاس موجودیت محدودیت مسئله csp را پیاده سازی میکند.

متد satisfied:

چک میکند ایا اساین کردن یک مقدار به یک متغییر محدودیت را برآورده میکند یا خیر.(محدودیت های مربوط به ستون ها در این متد پیاده سازی شده است.)

تابع assignment_puzzle :

این تابع assignment های انجام شده را به صورت آرایه و به صورت ستونی تبدیل میکند.

تابع satisfy:

این تابع احتمال اینکه یک رشته در آینده با یک رشته دیگری برابر شود را بررسی میکند.

تابع read_file :

در این تابع فایل (پازل) مورد نظر خوانده شده و مقادیر خوانده شده در ماتریس ذخیره شده و به همراه تعداد ستون ها(که برابر تعداد سطر ها هم هست) برگردانده می شود.

تابع generateDomains :

این تابع تمام رشته های باینری که تعداد یک ها و صفر های انها یکسان است را تولید میکند و سپس با توجه به پازل اولیه و محدودیت های ذکر شده برخی موارد را حذف کرده و برای هر متغییر دامنه را تولید میکند.

تابع next_each_other_constraint :

چک میکند که ایا داخل یک رشته سه یک یکسان یا سه صفر یکسان پشت سر هم داریم یا خیر.

تابع same_to_input :

چک میکند که ایا رشته مورد نظر داخل پازل داده شده صدق میکند یا خیر.

مقایسه MAC و Forward Checking :

تفاوت بین forward checking و MAC این است که FC یک متغییر اساین نشده را از نظر سازگاری بررسی میکند. ولی MAC جفت های اساین نشده را برای سازگاری متقابل بررسی میکند.

پس انجام forward checking نسبت به MAC سریعتر است ، اما MAC در هرس فضای جستجو موثرتر است.