درس مبانی و کاربردهای هوش مصنوعی

گزارش پروژه دوم

آرش حریرپوش
9731505

آرش حریرپوش
9731505

فرموله سازی مسئله:

برای فرموله سازی این مسئله ابتدا در فایل Model.py دو کلاس به نام Node و State تعریف می کنیم؛ کلاس Node مربوط به هر یک از خانه های مسئله می باشد، و مقدار عدد، رنگ و دامنه را برای آن خانه ذخیره می کند؛ همچنین یک متغیر دیگر نیز برای مشخص شدن موقعیت خانه به نام loc در این کلاس قرار دارد؛ در کلاس State، خانه های جدول قرار داده می شود؛ بدین صورت این کلاس یک متغیر state دارد که شامل یک آرایه از Node ها می باشد، که هر یک از آن ها مربوط به یک خانه از جدول می باشند؛ و یک متغیر دیگر replicate_state نیز قرار دارد، که در آن قبل از backtrack مقدار state فعلی را در آن کپی می کنیم؛ یک کلاس CSP نیز وجود دارد، که در آن state فعلی مسئله، محدودیت های موجود برای هر خانه، نود های مقدار داده شده و بقیه موارد قرار دارد، به کمک توابع پیاده سازی شده در این کلاس جستجو انجام می شود؛ و سپس در فایل main پس از خواندن داده ها و ایجاد محدودیت های اولیه، به کمک توابع موجود در فایل CSP.py جستجو انجام می شود.

با توجه به اینکه عمل جستجو و انتشار محدودیت در کلاس CSP انجام می شود، به معرفی توابع این کلاس پرداخته می شود:

- **find_constraints_of_nodes**: به کمک این تابع مشخص خانه هایی که از نظر شماره خانه و یا رنگ خانه باعث ایجاد محدودیت برای یک خانه می شود مشخص می شود؛ به نوعی همانند گراف محدودیت عمل می کند.
- **num_consistency**: به کمک این تابع محدودیت هایی که با انتساب یک عدد به یک خانه برای بقیه خانه ها ایجاد می شود، مشخص می گردد؛ محدودیت هایی که این تابع اعمال می کند برابر است با:
  - در صورتی که شماره خانه انتساب داده شده، مقدار عدد انتساب داده شده از خانه هایی که در سطر ستون آن خانه هستند حذف می شود.
  - در صورتی که مقدار عدد این خانه از خانه دیگر بیشتر باشد اولویت آن نیز باید بیشتر باشد؛ و اگر مقدار عددی این خانه از خانه دیگر کمتر باشد اولویت آن هم باید کمتر باشد.
  - در صورتی که مقدار عدد این خانه برابر بیشینه مقدار باشد؛ باید مقادیر دامنه با کمترین اولویت رنگ از دامنه حذف گردد.
  - در صورتی که مقدار عدد این خانه کمترین مقدار باشد؛ باید مقادیر با بیشترین اولویت رنگ از دامنه حذف گردد.
- **color_consistency**: به کمک این تابع محدودیت هایی که با انتساب یک رنگ به یک خانه برای بقیه خانه ها ایجاد می شود، مشخص می گردد؛ محدودیت هایی که این تابع اعمال می کند برابر است با:
  - هنگامی که به یک خانه رنگ انتساب داده می شود، این رنگ از دامنه بقیه خانه های مجاور آن حذف گردد.

آرش حریرپوش
9731505

- در صورتی که مقدار شماره یک خانه از خانه دیگری بیشتر باشد، مقادیر دامنه ای که اولویت رنگ کمتری نسبت به اولویت رنگ این خانه دارند حذف می گردند.
- در صورتی که مقدار شماره یک خانه از خانه دیگر کمتر باشد، مقادیر دامنه با اولویت رنگ بیشتری از این خانه دارند حذف می گردند.
- در صورتی که یک خانه بیشترین اولویت رنگ را داشته باشد، دامنه هایی که کوچک ترین شماره خانه را دارند از آن حذف می گردد.
- در صورتی که یک خانه کمترین اولویت را داشته باشد، دامنه هایی که بیشترین شماره خانه را دارند از آن حذف می گردد.

- **forward_checking:** به کمک دو تابع عنوان شده در بالا، محدودیت های ناشی از اعمال شماره و رنگ به یک خانه برای بقیه خانه ها اعمال می گردد.
- **check_consistency:** این تابع سازگاری دامنه قرار داده شده برای یک خانه را بررسی می کند.
- **calculate_degree:** این تابع مقدار هیوریستیک درجه را برای یک خانه مشخص می کند.
- **mrv_degree:** این تابع بر اساس هیوریستیک mrv و degree تصمیم می گیرد که کدام خانه مقدار دهی شود، بدین صورت که خانه ای انتخاب می شود که دامنه آن کمترین مقدار را داشته باشد و در صورتی که این مقدار برابر بود بر اساس هیوریستیک درجه تصمیم گیری می کند.
- **back_track_search:** در این تابع به کمک الگوریتم backtrack مسئله مورد نظر حل می شود.

ابتدا برای حل مسئله دامنه های رنگ و عدد را با یک دیگر ترکیب کرده و به عنوان یک مقدار دامنه در نظر می گیریم؛ در صورتی که به یک خانه هیچ مقدار اولیه ای داده نشده باشد، دامنه آن برابر هر ترکیب دو تایی از شماره و رنگ ها می شود؛ اما در صورتی که یکی از این مقادیر مقدار دهی شده باشد، آن مقدار در دامنه اعمال می گردد. همچنین بجای استفاده از اسم رنگ ها از اولویت آن ها استفاده می کنیم و اولویت هر رنگ را به صورت عددی که بزرگ ترین عدد بیشترین اولویت را دارد، نگهداری می شود. برای انجام این کار اسم هر رنگ و اولویت آن در یک دیکشنری نگهداری می شود و با استفاده از آن اولویت ها مشخص می شود.

یک نمونه از خروجی این الگوریتم برابر است با:

آرش حریرپوش
9731505

```
Please Enter m, n:
5 3
r g b y p
1# *b *#
*# 3r *#
*g 1# *#
Search Started....

Final State Has Found In 0.007980108261108398 Seconds.
['1y', '2b', '3r']
['2b', '3r', '1y']
['3g', '1y', '2b']
```