گزارش پروژه CSP - بازی سودوکو

امیر محمد شریفیان 9622762248

حامد داورمنش 9512762378

**خلاصه ای از مدل سازی :**

**کلاس variable**

برای حل کردن مسیله سودوکو ، هر خانه ی جدول سودوکو را به عنوان یک variable برای مسئله ی ارضای محدودیت در نظر گرفتیم. دامنه ی هر متغیر اعداد یک رقمی بین 1 تا 9 می باشند. از طرفی می دانیم که هر variable می تواند در یک cage نیز حضور داشته باشد ، در نتیجه برای نگه داری آن که هر متغیر متعلق به چه قفسی می باشد ، به هر متغیر یک ارجاع به cage اضافه کردیم .لازم به ذکر است که هر variable می تواند یکی از مقادیر دامنه ی خود را به عنوان مقدار نهایی اختیار کند. در انتها برای آن که بتوانیم تعداد دفعات مقداردهی را در نهایت داشته باشیم ، number of assignments را به تعریف کلاس اضافه کردیم .

public class Variable

{

public List<int> Domain { get; set; }

[Range(1,9)]

public int Value { get; set; }

public int Numbre\_Of\_Assingments { get; set; }

public Cage Cage { get; set; }

public int I { get; set; }

public int J { get; set; }

}

**کلاس cage**

در مسئله داده شده ، مجموعه خانه هایی وجود دارند که مجموع مقادیرشان باید برابر با عدد داده شده باشد. برای مدل سازی این جز ار مسئله کلاس cage را پیاده سازی نمودیم. هر cage می تواند شامل تعدادی variable باشد ، به همین منظور لیستی از ارجاعات به variable در نظر گرفتیم. در پایان هم برای نگه داری مقدار مورد نظر cage متغیر cage\_value را اضافه کردیم. کلاس cage شرح زیر می باشد :

public class Cage

{

public Cage()

{

variables = new List<Variable>();

}

public List<Variable> variables { get; set; }

public int Cage\_Value;

}

**کلاس program**

این کلاس اصلی ترین کلاس برنامه می باشد.اکثر توابع مهم و کاربردی در این کلاس گنجانده شده است . روش های حل مسئله نیز در همین کلاس به صورت تابع های مجزا پیاده سازی شده است. این کلاس مجموعه از ارجاعات به cage ها و variable ها دارد و بدین ترتیب شکل دهنده ی جدول سودوکو ما می باشد.

class Program

{

public static List<Variable> variables = new List<Variable>();

public static List<Cage> cages = new List<Cage>();

}

برای پرهیز از زیاده گویی و رعایت اختصار صرفا به بررسی توابع مهم این کلاس می پردازیم . لیست توابع به صورت کلی :

private static void BACKTRACKING\_SEARCH()

private static bool RECURSIVE\_BACKTRACKING()

این توابع مکانیزم اصلی روش اصلی بازگشتی موله را می سازند. پیاده سازی این توابع متناسب با مفاهیم مطرح شده در کلاس و اسلاید اساتید می باشد.

private static void LCV(Variable v)

private static Variable SELECT\_UNASSIGNED\_VARIABLE()

private static void ForwardChecking(List<Variable> list, int value)

تابع اول وظیفه ی انتخاب عدد از دامنه ی متغیر به روش LCV را برعهده دارد .

تابع دوم وظیفه ی انتخاب variable را به روش MRV را برعهده دارد.

تابع سوم نیز وظیفه ی انتشار محدودیت ها را دارد و لازم به ذکر است به جز توابع بالا که توابع اصلی برنامه می باشند، توابع utility نیز برای انجام کار های مختلف پیاده سازی شده اند. همه ی توابع این کلاس در یک نگاه :

private static void BACKTRACKING\_SEARCH()

private static bool RECURSIVE\_BACKTRACKING()

private static void LCV(Variable v)

private static void Undo\_ForwardChecking(List<Variable> list, int value)

private static Variable SELECT\_UNASSIGNED\_VARIABLE()

private static List<Variable> DEEP\_COPY()

private static bool Completed(List<int> vs)

private static void Print()

private static Forward\_Checking\_Respond Forward\_Checking(Variable var, int value)

private static bool CheckDomain(List<Variable> list, int value)

private static bool AllDiffer(Variable var, int value)

private static List<Variable> GetSameSquere(Variable var, int squere)

private static int Belongs\_To(Variable var)

private static int FindUnAssignedVariable(List<Variable> variables)

public static bool AllHasValue(List<Variable> variables)

علاوه بر کلاس های ذکر شده کلاس های دیگری نیز در پروژه وجود دارد که صرفا حالت utility دارند و نیازی به توضیح ندارند.

**جدول تعداد مقداردهی**

این جدول نیز تعداد مقداردهی برای ورودی های مختلف می باشد.

|  |  |  |  |
| --- | --- | --- | --- |
| Input/Number of Assignments | Easy | Medium | Hard |
| MRV |  | 27143 |  |
| MRV + Forward Checking | 257839 | 13442 | 363540 |
| MRV + ForwardChecking + LCV | 191279 | 10651 | 364270 |
|  |  |  |  |

در پایان لازم به ذکر است ، این پروژه در دو ربان جاوا و سی شارپ نوشته و تحویل داده شده و اطلاعات بخش انتهایی گزارش از نسخه سی شارپ است . (ساختار هر دو کد در کل یکسان است ).