- مسئله:

با استفاده از Z3 جدول سودوکو ارائه شده را حل نمایید .منابع پیشنهادی در ادامه بیان شده است .کد پیاده سازی شده را در گیت هاب به صورت مرحله به مرحله قرار دهید(مبانی مهندسی نرم افزار .) مراحل تحلیل و پیادهسازی بایستی مطابق بیان روشهای رسمی باشد و ارائه کد و توضیح آن بدون رعایت مفاهیم روش های رسمی مورد پذیرش نمی باشد.

				6	1			2
	7						6	
9	2							
		4	5	2		9		
	8	2	1		4	6	3	
		3		7	6	1		
							9	8
	3						4	
6			3	8				

Daily SuDoku: Mon 22-Apr-2024 medium

- بیان مسئله:

الگوی سودوکو یک جدول ۹۸۹است که به نحوی تقسیم شده است که باید اعداد ۱تا۹رادرآن قرار دهیم .بطوری که هر عدد فقط یکبار درهرسطر وستون و بلوک۳۸۳تکرارشود،این معمای عددی بازی منطقی است که مهارت های حل مسئله ومنطقی فرد را به چالش می کشد این مسئله بعنوان یک چالش ذهنی بسیار مفید است .

- نیازهای مسئله:

1. **سودوکو**

درلایه مفهومی سودوکو یک بازی منطقی ومعمایی است که با استفاده از یک جدول۹۸۹ اعداد،چالش حل مسئله را فراهم می کند.مفهوم اصلی سودوکو شامل این است که شما باید اعداد۱تا۹را درجدول قرار دهید،بطوری که هرعدد فقط یکبار درهر سطر،ستون وبلوک۳۸۳تکرار نشود.این بازی از دیدگاه مفهومی یک چالش ذهنی است که نیازمند تمرکز،منطق وروش های حل مختلف است تابه رازحل مسئله برسید.هدف اصلی این بازی ایجاد ترکیبی منطقی وصحیح از اعداد در جدول است که همه قوانین ومحدودیت های آن را رعایت کند.

از نظر سخت افزاری سودوکو یک بازی ساده است که به راحتی می توان بر روی انواع مختلفی از دستگاه ها اجرا کرد ازجمله کامپیوتر ها،تلفن همراه،تبلت ها و حتی کتاب های الکترونیکی .باتوجه به سادگی وپردازش سبک مورد نیاز،حتی دستگاه های باقدرت پردازش پایین نیز قادربه اجرای

بازی های سودوکو هستند که به این معناست که می توانید به راحتی این بازی رابرروی بسیاری از دستگاه های قدیمی یادستگاه هایی با منابع محدود اجرا کرد.

از نظر نرم افزاری در ابتدا نرم افزارهای سودوکو فقط برای حل مسئله سودوکو بودند امابا گذرزمان این نرم افزارها ویژگی های دیگری همچون سطوح مختلف دشواری،تعامل باسایر بازیکنان از طریق ارسال ودریافت امتیازات وحتی تولیدمسائل سودوکو جدید را نیز شامل شوند.به علاوه برخی از این نرم افزارها امکاناتی مثل راهنمایی های حل روش های حل مختلف وتجزیه وتحلیل برای مراحلی که بیشترین چالش رادارند نیز ارائه می دهند.

قاعده و قانون های سودوکو81خانه:

۱.نه ستون و ردیف

۲.نه جدول کوچکتر

۳.اعداد از ۱تا۹درسلول های خالی

۴.درهرردیف،ستون و هربلوک جدول ۳۸۳هرعدد فقط یکبار ظاهر می شود.

۵.قانون ۱:درهرسطر جدول اعداد ۱الی۹بدون تکرار بیاید.

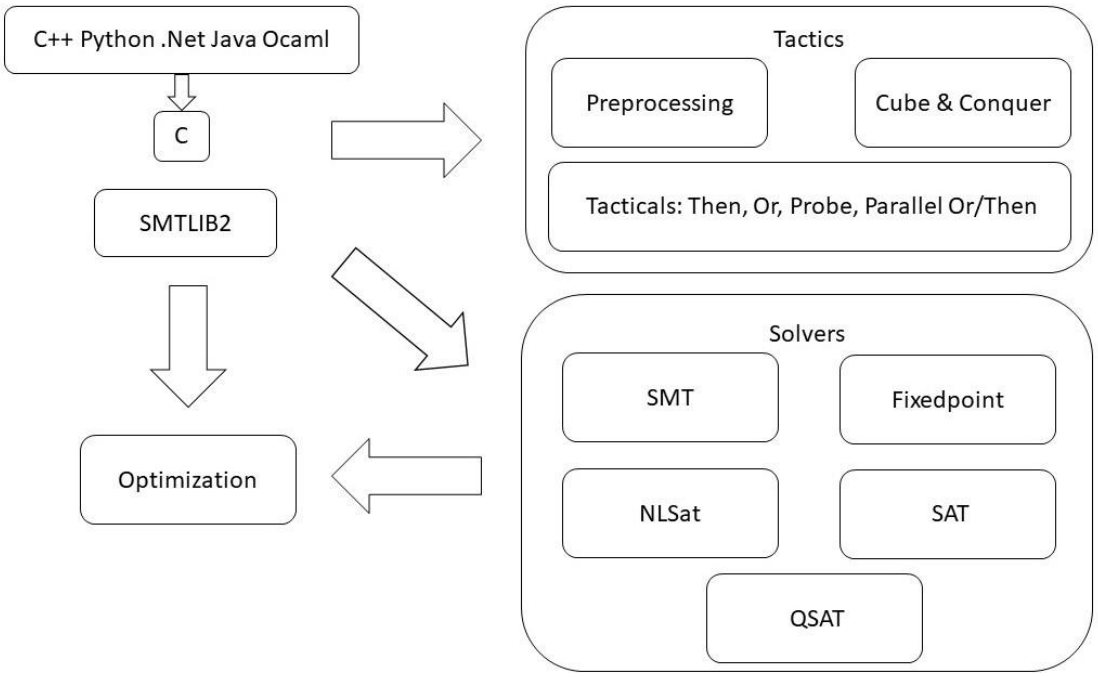
۶.قانون ۲:درهرستون جدول اعداد۱ الی۹بدون تکرار بیاید.

۷.قانون۳:درهرناحیه ۳۸۳جدول اعداد ۱الی ۹بدون تکرار بیاید.

Z3 .2

- ✓ SMT: یک مسئله تصمیم‌گیری برای فرمول‌های منطقی با توجه به ترکیبی از نظریه‌های پس‌زمینه مانند حساب، بردار بیت، آرایه‌ها و توابع تفسیر نشده است. حل SMT از یک رابطه هم افزایی با تجزیه و تحلیل نرم افزار، تأیید و ابزارهای اجرای نمادین برخوردار است.

- ✓ Z3 : یک حل کننده SMT کارآمد با الگوریتم های تخصصی برای حل تئوری های پس زمینه است.



شکل بالا نمودار کلی سیستم Z3 را در نسخه 4.8 نشان می دهد. اجزای کلیدی آن شامل موارد زیر هستند:

- بالا سمت چپ رابط های Z3 را خلاصه می کند: اسکریپت‌های SMT-LIB2 فایل های متنی هستند که Z3 میتواند آنهارا پردازش کند. و تماس های API برای تعامل با Z3 از طریق زبان های برنامه نویسی سطح بالا مانند پایتون

- رابط کاربری پایتون: روش اصلی سند برای تعامل با Z3.

- تاکتیک ها(tactics): ابزاری را برای پیش پردازش ساده سازی و ایجاد اهداف فرعی ارائه می کنند.

- بهینه‌سازی(optimization): سرویس‌ها به کاربران اجازه می‌دهند تا توابع هدف مدول رضایت‌بخشی را برای به حداکثر رساندن یا به حداقل رساندن مقادیر حل کنند.

3. Visual Studio Code

یک ویرایشگر کد منبع برای گنو/لینوکس، او اس ده و ویندوز می‌باشد. این نرم‌افزار توسط مایکروسافت توسعه داده شده و هم‌اکنون به‌طور رایگان در دسترس است.

4. حساب github

یک حساب کاربری است که میتوان آنرا در در پلتفرم گیت هاب ایجاد کرد.
گیت‌هاب یک سرویس میزبانی وب برای پروژه‌هایی است که از سیستم سورس کنترل می‌کند. گیت هاب با استفاده از روبی آن ریلز و ارلنگ ساخته شده است.
سایت گیت‌هاب همه عملکردهای مورد نیاز کاربران در یک جامعه مجازی را در قرار داده‌است.

- نحوه حل مسئله:

برای حل یک جدول سودوکو با استفاده از Z3، ابتدا باید متغیرهای مربوط به خانه‌های مختلف جدول را تعریف کنید. سپس باید شرایط محدودیت‌های جدول سودوکو را به عنوان قیدها اعمال کنید.

به عنوان مثال، برای یک جدول سودوکو 9*9، می‌توانید 81 متغیر مربوط به هر خانه در نظر بگیرید. سپس بحث محدودیت‌هایی مانند اینکه هر عدد از 1 تا 9 باید یکبار در هر سطر، ستون و بلوک 3*3 تکرار شود را به عنوان قیدها اعمال کنید.

سپس با استفاده از Z3، این قیدها را به عنوان ورودی بدهید و سیستم را برای یافتن راه حل منطبق با این قیدها اجرا کنید. اگر سیستم یک راه حل پیدا کند، جدول سودوکو حل شده را خواهید داشت.

به عنوان یک راهنمایی کلی، می‌توان از کتابخانه Z3Py برای اجرای Z3 در پایتون استفاده کرد و با استفاده از آن، قیدها را تعریف و جدول سودوکو را حل نمود.

وظیفه هر خط کد

- from Z3 import * :

تمام توابع کتابخانه Z3 را ایمپورت میکند.

- # sudoku instance, use '0' for empty cells :

لیستی 9*9 را تعریف میکند که وضعیت اولیه سودوکو را نشان میدهد.(خانه های خالی را با 0 نشان میدهیم.)

- instance = (
  (0,0,0,0,6,1,0,0,2),
  (0,7,0,0,0,0,0,6,0),
  (9,2,0,0,0,0,0,0,0),

```
        (0,0,4,5,2,0,9,0,0),
        (0,8,2,1,0,4,6,3,0),
        (0,0,3,0,7,6,1,0,0),
        (0,0,0,0,0,0,0,9,8),
        (0,3,0,0,0,0,0,4,0),
        (6,0,0,3,8,0,0,0,0)
)
```

لیست سودوکو بصورت بالا تعریف میشود.

- cells :

ماتریسی از متغیر های صحیح ایجاد میکند.

- cell_constraints :

اطمینان میدهد هرخانه حتما مقداری بین 1و9 دارد.

- row_constraints :

اطمینان میدهد هر ردیف حتما شامل اعداد منحصر به فرد است (از تابع Distinct استفاده میکند.)

- col_constraints :

اطمینان میدهد هر ستون حتما شامل اعداد منحصر به فرد است(مشابه row_constraints)

- square_constraints

اطمینان میدهد هر مربع 3*3 حتما شامل اعداد منحصر به فرد است.

- sudoku_constraints :

محدودیت های تعریف شده قبلی را ترکیب میکند.(محدودیت روی مقدار هر خانه و منحصر به فرد بودن مقدار هر ردیف، ستون و مربع 3*3)

- instance_constraints :

اطمینان میدهد خانه های از پیش پر شده جدول حتما مقادیر اولیه خودرا حفظ میکنند.

- solver = Solver() :

یک نمونه حل کننده ایجاد میکند.

- solver.add :

تمامی محدودیت های قبلی را به حل کننده اضافه میکند.

- if solver.check() == sat:

بررسی میکند که آیا حل کننده میتواند راه حلی شامل تمام محدودیت ها پیدا کند.(sat ثابتی در Z3 میباشد که نشان دهنده یک حالت قابل قبول است.)

- model = solver.model() :

اگر راه حلی وجود داشته باشد مدل را از حل کننده بازیابی میکند.

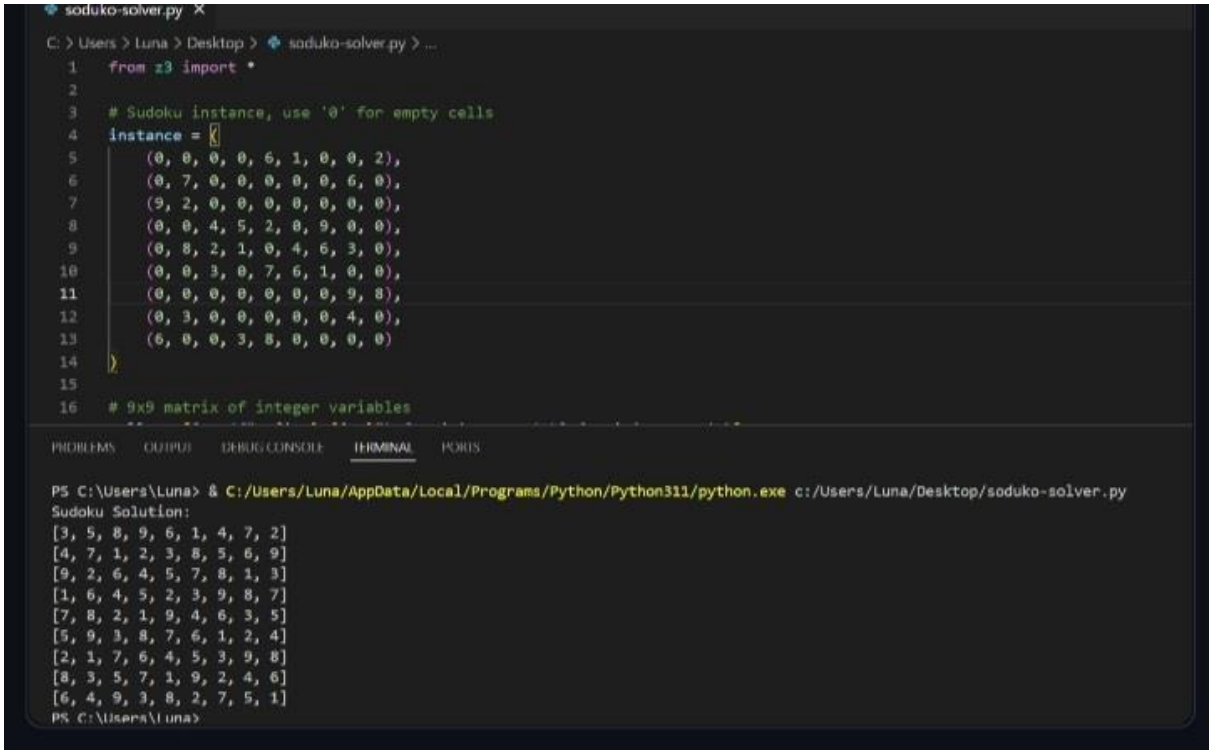
- Solution :

سودوکو حل شده را ایجاد میکند.

- else :

اگر راه حلی پیدا نشود .No solution exists را چاپ میکند.

- پیاده سازی مسئله:


soduko-solver.py
C: > Users > Luna > Desktop > soduko-solver.py > ...
from z3 import *
# Sudoku instance, use '0' for empty cells
instance = (
(0, 0, 0, 0, 6, 1, 0, 0, 2),
(0, 7, 0, 0, 0, 0, 0, 6, 0),
(9, 2, 0, 0, 0, 0, 0, 0, 0),
(0, 0, 4, 5, 2, 0, 9, 0, 0),
(0, 8, 2, 1, 0, 4, 6, 3, 0),
(0, 0, 3, 0, 7, 6, 1, 0, 0),
(0, 0, 0, 0, 0, 0, 0, 9, 8),
(0, 3, 0, 0, 0, 0, 0, 4, 0),
(6, 0, 0, 3, 8, 0, 0, 0, 0)
)
9x9 matrix of integer variables
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
PS C:\Users\Luna> & C:/Users/Luna/AppData/Local/Programs/Python/Python311/python.exe c:/Users/Luna/Desktop/soduko-solver.py
Sudoku Solution:
[3, 5, 8, 9, 6, 1, 4, 7, 2]
[4, 7, 1, 2, 3, 8, 5, 6, 9]
[9, 2, 6, 4, 5, 7, 8, 1, 3]
[1, 6, 4, 5, 2, 3, 9, 8, 7]
[7, 8, 2, 1, 9, 4, 6, 3, 5]
[5, 9, 3, 8, 7, 6, 1, 2, 4]
[2, 1, 7, 6, 4, 5, 3, 9, 8]
[8, 3, 5, 7, 1, 9, 2, 4, 6]
[6, 4, 9, 3, 8, 2, 7, 5, 1]
PS C:\Users\Luna>