مسئله : با استفاده از Z3 جدول سودوکوی داده شده را حل کنیم.

گام اول مدل سازی : سودوکو یک جدول (لیست) دو بعدی است که دارای سلول است و به صورت 3\*3 و 9\*9 و یا بیشتر میتواند باشد و اعداد به صورت تصادفی در سطر و ستون های آن قرار دارند .

نکته : سودوکوی 9\*9 از مربع های 3\*3 تشکیل شده است.

سودوکو دارای یکسری شرط هستش :

1. در یک ردیف یا یک سطر از یک عدد دو بار داشته باشیم
2. اعدادی که وارد می شوند باید در دامنه ی 1 تا 9 باشند
3. اعدادی که به صورت پیش فرض در جدول وارد شده اند نباید تغییر کنند

نتیجه برای حل سودوکو : با توجه به شرط یک و دو ما باید در هر سطر و ستون از عدد 1 تا 9 باید یدونه داشته باشیم که جدول ما حل و کامل شود.

حال که مدل سازی کردیم با توجه به مسئله ما میخواهیم با استفاده از Z3 این مسئله را حل کنیم.

پس در این مرحله ما نیاز داریم که در ابتدا بدونیم Z3 اصلا چی هستش؟

یک کتابخانه عالی برای حل مسائل پیچیده ریاضی است که از بسیاری از زبان های برنامه نویسی مختلف پشتیبانی می کند

برای حل مسئله سودوکو با استفاده از کتابخانه Z3 در اینجا از زبان پایتون استفاده خواهیم کرد.

حال که با Z3 آشنا شدیم برویم سراغ مرحله نصب آن:

نصب z3 :

برای استفاده از کتابخانه z3 کافیست که دستور : pip install z3-solver

را در shell خود وارد کنید تا کتابخانه z3 در سیستم شما نصب شود.

و با این دستور میتوانیم از توابع کتابخانه Z3 استفاده کنیم :

from z3 import \*

گام دوم تعریف متغیر :

برای جدول سودوکو یک متغیر در نظر میگیریم

و برای مقدار دهی آن ، بجای سلول های خالی جدول مقدار صفر را میگذاریم   
و برای سلول های پر شده ی جدول همان مقدارخودشان را قرار میدهیم.

در پایتون با استفاده از تاپل جدول مورد نظرمان را میسازیم به این صورت:

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))

حال وقت آن رسیده که شرط هایی که در مدل سازی مطرح کردیم را به زبان پایتون تبدیل کنیم :

*#  ماتریس 9 در 9 اعداد صحیح*

X = [ [ Int("x\_%s\_%s" % (i+1, j+1)) for j in range(9) ]

      for i in range(9) ]

*# هر سلول دارای مقداری از 1 تا 9 است*

cells\_c  = [ And(1 <= X[i][j], X[i][j] <= 9)

             for i in range(9) for j in range(9) ]

*# هر سطر حداکثر از اعداد 1 تا 9 یدونه دارد*

rows\_c   = [ Distinct(X[i]) for i in range(9) ]

*# هر ستون حداکثر از اعداد 1 تا 9 یدونه دارد*

cols\_c   = [ Distinct([ X[i][j] for i in range(9) ])

             for j in range(9) ]

*# هر مربع 3 در 3 که در جدول وجود دارد از تمام اعداد 1 تا 9 در آن وجود دارد . از هر عدد حداکثر یکبار*

sq\_c     = [ Distinct([ X[3\*i0 + i][3\*j0 + j]

                        for i in range(3) for j in range(3) ])

             for i0 in range(3) for j0 in range(3) ]

sudoku\_c = cells\_c + rows\_c + cols\_c + sq\_c

*# مقادیری که به صورت پیش فرض در جدول وارد شده اند تغییر نمی کنند*

instance\_c = [ X[i][j] == instance[i][j] for i in range(9) for j in range(9) if not (instance[i][j] == 0) ]

گام سوم حل مسئله :

زمان آن رسیده است که با استفاده از کتابخانه Z3 ، جدول سودوکو مورد نظر را حل کنیم.

در کتابخانه Z3 یک تابعی تعریف شده که مسئله مورد نظر را حل میکند ، آن را درون یک متغیر فراخوانی میکنیم :

s = Solver()

حال با استفاده از تابع add مقادیری که تابع solver حل می کند را به جدول سودوکو اضافه میکنیم و در کنار مقادیر پیش فرض جدول سودوکو قرار میدهیم تا جدولمان کامل شود:

s.add(sudoku\_c + instance\_c )

حال یک شرط اضافه میکنیم که بررسی کند آیا جواب هایی که Z3 حل میکند در دامنه 1 تا 9 هستش ؟ اگر که هست آن را به عنوان جواب برامون چاپ کند ،

در غیر این صورت چاپ کند که "مشکلی در حل کردن پیش آمده است"

if s.check() == sat:

    m = s.model()

    r = [ [ m.evaluate(X[i][j]) for j in range(9) ]

          for i in range(9) ]

    print\_matrix(r)

else:

    print("failed to solve")

تبریک می گویم حال مسئله ما به جواب رسیده است.