

sudoku

February 23, 2020

1 Sudoku

Os puzzles Sudoku são problemas de colocação de números inteiros entre 1 e N^2 numa matriz quadrada de dimensão N^2 , por forma a que cada coluna e cada linha contenha todos os números, sem repetições. Além disso, cada matriz contém N^2 sub-matrizes quadradas disjuntas, de dimensão N , que deverão também elas conter os números entre 1 e N^2 .

Cada problema é dado por uma matriz parcialmente preenchida, cabendo ao jogador completá-la.

O problema pode ser codificado através de um conjunto de N^4 constantes de tipo inteiro, correspondentes às posições da matriz, e escrevendo:

- $2 \times N^4$ desigualdades para os limites inferior e superior das constantes;
- N^2 restrições do tipo “todos diferentes”, uma para cada linha da matriz;
- N^2 restrições do tipo “todos diferentes”, uma para cada coluna da matriz;
- N^2 restrições do tipo “todos diferentes”, uma para cada sub-matriz da matriz.

Acrescem ainda as restrições (igualdades) correspondentes à definição de um tabuleiro concreto.

```
[4]: from z3 import *
```

Vamos necessitar de uma família de variáveis inteiras x_{ij} e, para isso, vamos criar um dicionário do Python.

```
[33]: N = 3

s = Solver()

x={}
for i in range(N*N):
    x[i] = {}
    for j in range(N*N):
        x[i][j] = Int('x'+str(i)+str(j))      # declaração de variáveis
        s.add(And(1<= x[i][j], x[i][j]<=9))    # restrições de valor

# restrições de linha
for i in range(N*N):
    s.add(Distinct([ x[i][j] for j in range(N*N)]))

# restrições de coluna
```

```

for j in range(N*N):
    s.add(Distinct([ x[i][j] for i in range(N*N)]))

# restrições de regiões (sub-matrizes)
for a in range(N):
    for b in range(N):
        s.add(Distinct( [ x[a*N+i][b*N+j] for i in range(N) for j in range(N)]_
→))

s.push()

# restrições de tabuleiro
s.add(x[0][0] == 3)
s.add(x[4][3] == 9)
s.add(x[7][1] == 4)
s.add(x[1][6] == 8)
s.add(x[2][8] == 1)
s.add(x[5][3] == 5)
s.add(x[5][0] == 6)

r = s.check()
if r==sat :
    m = s.model()
    # print(m)
    # print('')
    print('===== SOLUÇÃO =====')
    for i in range(N*N):
        print([ m[x[i][j]].as_long() for j in range(N*N)])

else:
    print("Não tem solução.")

```

```

===== SOLUÇÃO =====
[3, 1, 8, 4, 5, 2, 9, 7, 6]
[4, 6, 5, 7, 1, 9, 8, 2, 3]
[2, 7, 9, 8, 3, 6, 5, 4, 1]
[9, 5, 2, 3, 6, 4, 1, 8, 7]
[7, 3, 1, 9, 2, 8, 4, 6, 5]
[6, 8, 4, 5, 7, 1, 2, 3, 9]
[1, 9, 3, 2, 4, 7, 6, 5, 8]
[5, 4, 6, 1, 8, 3, 7, 9, 2]
[8, 2, 7, 6, 9, 5, 3, 1, 4]

```

[]: