Решение произвольных судоку с помощью SAT

Хритова Екатерина Андреевна
510 группа

Задание 3. Судоку

Решите «самую сложную головоломку судоку» (по мнению The Telegraph):

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

Напишите программу, решающую произвольные судоку с помощью SAT или LP.

Постановка задачи

Головоломка судоку представлена сеткой 9×9, которая состоит из девяти подсеток 3×3 (также называемых коробками). Некоторые элементы сетки заполнены числами от 1 до 9, тогда как другие элементы оставлены пустыми. Головоломка судоку решается путем присвоения пустым элементам чисел от 1 до 9 таким образом, чтобы каждая строка, каждый столбец и каждая подсетка 3×3 содержали каждое из девяти возможных чисел.

Задачу судоку можно обобщить на случай квадрата $n \times n$, заполненного числами от 1 до n.

Кодировка SAT для задачи судоку

Задача SAT представлена с помощью n переменных $x_1, x_2, ..., x_n$, которым можно присвоить значения истинности 0 (ложь) или 1 (истина). Литерал l_i — это либо переменная x_i (т. е. положительный литерал), либо ее дополнение $\bar{x_i}$ (т. е. отрицательный литерал). Дизъюнкт (предложение, клоз) C_j является дизъюнктом литералов, а формула КНФ φ является конъюнкцией дизъюнктов.

$$\varphi = C_1 \wedge C_2 \wedge ... \wedge C_r, \quad C_m = l_{i_1} \vee l_{i_2} \vee ... l_{i_m}$$

Литерал l_i дизъюнкта C_j, которому присвоено истинностное значение 1, удовлетворяет предложению, и говорят, что предложение *выполнено*. *Формула выполняется*, если выполняются все ее условия. Проблема SAT состоит в том, чтобы решить, существует ли истинное назначение переменных, при котором формула удовлетворяется. Введем переменные $x_{i,j,v}$. Переменная $x_{i,j,v}$ принимает истинное значение тогда и только тогда, когда в строке i и столбце j стоит число v. Тогда для кодирования задачи судоку потребуется $9 \cdot 9 \cdot 9 = 729$ переменных.

Необходимо, чтобы каждая строка, каждый столбец и каждая клетка 3×3 (назовем их *объектами*) содержали каждое из девяти возможных чисел. Каждый из этих объектов состоит из 9 клеточек и может быть однозначно описан координатами этих клеток $p_k = (i_k, j_k)$. Тогда:

1. В каждой клетке p_i должна стоять одна из 9 цифр:

$$\vee_{v=1}^{9} x_{p_i,v}$$

2. В каждой клетке одновременно может стоять только одно число:

$$\wedge_{v=1}^{8} \wedge_{u=v+1}^{9} (\bar{x}_{p_i,v} \vee \bar{x}_{p_i,u})$$

Для каждого объекта условие того, что 9 клеток содержат различные значения, можно записать в виде формулы:

$$F(p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9) = \wedge_{i=1}^{8} \wedge_{j=i+1}^{9} \wedge_{v=1}^{9} (\bar{x}_{p_i,v} \vee \bar{x}_{p_j,v})$$

Таким образом, необходимо, чтобы были выполнены следующие условия:

1. Каждое число встречается хотя бы один раз:

$$C_1 = \wedge_{i=1}^{9} \wedge_{j=1}^{9} \vee_{v=1}^{9} x_{i,j,v}$$

2. Каждое число появляется не более одного раза в каждой строке:

$$C_2 = \wedge_{j=1}^{9} (\wedge_{v=1}^{9} \wedge_{i=1}^{8} \wedge_{k=i+1}^{9} (\bar{x}_{i,j,v} \vee \bar{x}_{k,j,v}))$$

3. Каждое число появляется не более одного раза в каждом столбце:

$$C_3 = \wedge_{i=1}^{9} (\wedge_{v=1}^{9} \wedge_{j=1}^{8} \wedge_{k=j+1}^{9} (\bar{x}_{i,j,v} \vee \bar{x}_{i,k,v}))$$

4. Каждое число появляется не более одного раза в каждой подсетке 3×3:

$$C_4 = \wedge_{v=1}^{9} (\wedge_{p=0}^{2} \wedge_{q=0}^{2} \wedge_{i=1}^{3} \wedge_{j=1}^{3} \wedge_{k=j+1}^{3} (\bar{x}_{(3p+i),(3q+j),v} \vee \bar{x}_{(3p+i),(3q+k),v}))$$

$$C_5 = \wedge_{v=1}^{9} (\wedge_{p=0}^{2} \wedge_{q=0}^{2} \wedge_{i=1}^{3} \wedge_{j=1}^{3} \wedge_{k=i+1}^{3} \wedge_{l=1}^{3} (\bar{x}_{(3p+i),(3q+j),v} \vee \bar{x}_{(3p+k),(3q+l),v}))$$

И итоговая формула:

$$\varphi = C_1 \wedge C_2 \wedge C_3 \wedge C_4 \wedge C_5$$

```
# преодразование тройного индекса (i,j,v) в одинарный
def ijv_to_id(i,j,v):
    return i*81 + j*9 + v + 1

# преодразование одинарного индекса в тройной (i,j,v)
def id_to_ijv(ind):
    ind = ind - 1
     i = ind // 81
    ind = ind - 81*i
     j = ind // 9
     v = ind - 9*j
    return i,j,v
```

```
from itertools import product

# получаем клоз C1 (каждое число встречается х.б. один раз)
def get_C1():
     C1 = []
    for i,j in product(range(9), repeat=2):
        C1.append([ijv_to_id(i,j,v) for v in range(9)])
    return C1

# получаем клоз C2 (каждое число появляется не более одного раза в каждой строке)
def get_C2():
     C2 = []
     for j,v in product(range(9), repeat=2):
        for i in range(8):
            for k in range(i+1,9):
                C2.append([-ijv_to_id(i,j,v), -ijv_to_id(k,j,v)])
    return C2

# получаем клоз C3 (каждое число появляется не более одного раза в каждом столбце)
def get_C3():
    C3 = []
    for i,v in product(range(9), repeat=2):
        for j in range(8):
            for k in range(j+1,9):
                C3.append([-ijv_to_id(i,j,v), -ijv_to_id(i,k,v)])
    return C3
```

```
# получаем клоз C4 (каждое число появляется не более одного раза в каждой подсетке)
def get_C4():
    C4 = []
    for v in range(9):
        for p,q,i,j in product(range(3), repeat=4):
            for k in range(j+1,3):
                C4.append([-ijv_to_id(3*p + i,3*q + j,v),\\
                           -ijv_to_id(3*p + i,3*q + k,v)])
    return C4

# получаем клоз C5 (каждое число появляется не более одного раза в каждой подсетке)
def get_C5():
    C5 = []
    for v in range(9):
        for p,q,i,j,l in product(range(3), repeat=5):
            for k in range(i+1,3):
                C5.append([-ijv_to_id(3*p + i,3*q + j,v),\\
                           -ijv_to_id(3*p + k,3*q + l,v)])
    return C5
```

Решение судоку.

```
import pycosat

def solve_sudoku(data):
    C1 = get_C1()
    C2 = get_C2()
    C3 = get_C3()
    C4 = get_C4()
    C5 = get_C5()

    solution = pycosat.solve(C1 + C2 + C3 + C4 + C5 + data)
```

Примеры.

```
PROBLEM:

+-------+-------+-------+
| 8 . . | . . . | . . . |
| . . 3 | 6 . . | . . . |
| . 7 . | . 9 . | 2 . . |
+-------+-------+-------+
| . 5 . | . . 7 | . . . |
| . . . | . 4 5 | 7 . . |
| . . . | 1 . . | . 3 . |
|-------+-------+-------+
| . . 1 | . . . | . 6 8 |
| . . 8 | 5 . . | . 1 . |
| . 9 . | . . . | 4 . . |
+-------+-------+-------+
```

```
SOLUTION:

+-------+-------+-------+
| 8 1 2 | 7 5 3 | 6 4 9 |
| 9 4 3 | 6 8 2 | 1 7 5 |
| 6 7 5 | 4 9 1 | 2 8 3 |
+-------+-------+-------+
| 1 5 4 | 2 3 7 | 8 9 6 |
| 3 6 9 | 8 4 5 | 7 2 1 |
| 2 8 7 | 1 6 9 | 5 3 4 |
+-------+-------+-------+
| 5 2 1 | 9 7 4 | 3 6 8 |
| 4 3 8 | 5 2 6 | 9 1 7 |
| 7 9 6 | 3 1 8 | 4 5 2 |
+-------+-------+-------+
```

```
PROBLEM:

+-------+-------+-------+
| 8 4 . | 7 5 . | . . . |
| . . . | . 6 . | . 8 . |
| . . . | . . 8 | 5 9 2 |
+-------+-------+-------+
| . 3 6 | . 9 . | . 5 . |
| 9 . . | . . 7 | 6 . . |
| 4 . . | 6 . . | . 1 9 |
|-------+-------+-------+
| . . 4 | . 2 . | . . 6 |
| . 8 9 | . . . | . . . |
| 7 . . | . . 6 | . . . |
+-------+-------+-------+
```

```
SOLUTION:

+-------+-------+-------+
| 8 4 2 | 7 5 9 | 3 6 1 |
| 5 9 1 | 3 6 2 | 7 8 4 |
| 3 6 7 | 4 1 8 | 5 9 2 |
+-------+-------+-------+
| 2 3 6 | 8 9 1 | 4 5 7 |
| 9 1 5 | 2 4 7 | 6 3 8 |
| 4 7 8 | 6 3 5 | 2 1 9 |
+-------+-------+-------+
| 1 5 4 | 9 2 3 | 8 7 6 |
| 6 8 9 | 5 7 4 | 1 2 3 |
| 7 2 3 | 1 8 6 | 9 4 5 |
+-------+-------+-------+
```

PROBLEM:

```
+-------+-------+-------+
| 3 . . | . . . | . 6 . |
| 5 7 . | 2 . . | . . . |
| . 8 . | . 5 . | . 4 . |
+-------+-------+-------+
| . . . | . 7 1 | 3 . 9 |
| . . 1 | 4 . 6 | 5 . . |
| . 5 3 | . . . | 6 . . |
|-------+-------+-------+
| . . . | . 8 . | 4 . 7 |
| . 4 . | . . . | . . 1 |
| 9 . . | 7 4 . | . . . |
+-------+-------+-------+
```

SOLUTION:

```
+-------+-------+-------+
| 3 2 9 | 8 1 4 | 7 6 5 |
| 5 7 4 | 2 6 9 | 1 8 3 |
| 1 8 6 | 3 5 7 | 9 4 2 |
+-------+-------+-------+
| 4 6 8 | 5 7 1 | 3 2 9 |
| 2 9 1 | 4 3 6 | 5 7 8 |
| 7 5 3 | 9 2 8 | 6 1 4 |
+-------+-------+-------+
| 6 3 5 | 1 8 2 | 4 9 7 |
| 8 4 7 | 6 9 5 | 2 3 1 |
| 9 1 2 | 7 4 3 | 8 5 6 |
+-------+-------+-------+
```