ЛР з розділу 4, Щербакова Валерія, ФІ-71

3. Судоку

Сформулювати умови головоломки судоку як задачу пошуку розмітки на напівкільці $\langle\{0,1\},\vee,\wedge\rangle$ та реалізувати алгоритм її розв'язку.

1. На вхід програма приймає табличку з цифрами від 1 до 9 в заповнених клітинках та 0 в якості пустих клітинок.

2. На вихід програма виводить або розв'язок задачі (табличку з цифрами), або повідомлення про те, що розв'язок не було знайдено.

3. Необов'язковою корисною можливістю буде вивід на екран проміжних результатів: табличок, що містять цифри там, де вже обрано відповідь, та пробіли там, де ще є кілька варіантів можливих цифр.

Сформулирую условие головоломки судоку как CSP:

$Z = \langle T, \tau \subseteq T^2, K, g: \tau \times K^2 \to \{0,1\}\rangle$

Множество меток: $K = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$

Множество объектов $T = \{\forall i, j \in I: t_{ij}\}$ (здесь множество индексов $I = K$).

Условие соседства: $\tau\big(t_{ij}, t_{ab}\big) = (⟦i \neq a⟧ \wedge ⟦j = b⟧) \vee (⟦i = a⟧ \wedge ⟦j \neq b⟧) \vee$
$\vee (⟦i \neq a⟧ \wedge ⟦j \neq b⟧ \wedge ⟦s_i = s_a⟧ \wedge ⟦s_j = s_b⟧)$

Пусть множество соседей x_{ij}: $N_{ij} = \left\{\forall a, b \in I,\ \tau\big(t_{ij}, t_{ab}\big) = 1: \boldsymbol{t_{ab}}\right\}$

Вспомогательная функция s_i определяет квадрат:

$$\mathrm{s}_i = \lceil (i-1)/3 \rceil \overset{i \in I}{=} \begin{cases} 0, 1 \leq i \leq 3 \\ 1, 4 \leq i \leq 6 \\ 2, 7 \leq i \leq 9 \end{cases} = ⟦i \geq 4⟧ + ⟦i \geq 7⟧.$$

s_i, s_j	i=1	2	3	4	5	6	7	8	9
j=1	1,1	1,1	1,1	2,1	2,1	2,1	3,1	3,1	3,1
2	1,1	1,1	1,1	2,1	2,1	2,1	3,1	3,1	3,1
3	1,1	1,1	1,1	2,1	2,1	2,1	3,1	3,1	3,1
4	1,2	1,2	1,2	2,2	2,2	2,2	3,2	3,2	3,2
5	1,2	1,2	1,2	2,2	2,2	2,2	3,2	3,2	3,2
6	1,2	1,2	1,2	2,2	2,2	2,2	3,2	3,2	3,2
7	1,3	1,3	1,3	2,3	2,3	2,3	3,3	3,3	3,3
8	1,3	1,3	1,3	2,3	2,3	2,3	3,3	3,3	3,3
9	1,3	1,3	1,3	2,3	2,3	2,3	3,3	3,3	3,3

Таким образом, соседями какого-то элемента x_{ij} считаются все элементы на одной вертикали с ним ($⟦i = a⟧$), все элементы на одной горизонтали с ним ($⟦j = b⟧$) а также все элементы в одном квадрате ($⟦s_i = s_a⟧ \wedge ⟦s_j = s_b⟧$), за исключением x_{ij}.

Ограничение простое – метки у соседей не должны совпадать: $g_{tt'}(k, k') = ⟦k \neq k'⟧$.

Вспомогательная функция индикатора установленной метки: $q_{ij}(x) = ⟦k_{ij} = x⟧$

Теперь следует описание непосредственно алгоритма решения задачи.

На вход передается $\{v_{ij}\} = v \in \widetilde{K}^{9\times 9}$, где $\widetilde{K} = K \cup \{0\}$, где 0 – не заполнена.

1. Задать метки объектов t_{ij} входными данными: $k_{ij} \coloneqq v_{ij}$.

2. Проверить $g_{t_{ij}t_{ab}}(k_{ij}, k_{ab})$ для $\forall k_{ij} \neq 0, \forall k_{ab} \in N_{ij}$. Если хотя бы одно значение g будет равно нулю, завершить алгоритм с результатом – задача не имеет решения по причине некорректного условия задачи.

3. Определить допустимые метки для каждого объекта с $k_{ij} = 0$: $P_{ij} \coloneqq K \setminus \bigcup_{\substack{t_{ab}\in N_{ij} \\ k_{ab}\neq 0}} \{k_{ab}\}$

4. Если существует объект без допустимых значений: $\exists k_{ij} = 0, |P_{ij}| = 0$, то завершить алгоритм с результатом – задача не имеет решения по причине некорректного условия задачи (в процессе решения обнаруживаются противоречащие ограничения).

5. Если существует такой объект, для которого допустимо только одно значение метки $\exists P_{ij}\colon |P_{ij}| = 1$, то, обозначив $P_{ij} = \{p_{ij}\}$, зафиксировать это допустимое значение в разметке: $k_{ij} \coloneqq p_{ij}$ и перейти на шаг 3.

6. Завершить работу алгоритма.

Если $\nexists k_{ij} = 0$, то k – решение задачи.

Если $\exists k_{ij} = 0$, то алгоритм не может решить задачу.