

Trabalho 2: Satisfação de Restrições

O **sudoku** é um puzzle numérico bem conhecido onde é apresentada uma grelha 9×9 parcialmente preenchida com números de 1 a 9. As regras do puzzle proibem que o mesmo número ocorra repetido numa linha, coluna ou "bloco". O **objetivo** é preencher todas as 81 posições da grelha.

Por exemplo:

Puzzle	Posições
_ _ 5 _ 8 _ _ 7 _	11 12 13 14 15 16 17 18 19
7 _ _ 2 _ 4 _ _ 5	21 22 23 24 25 26 27 28 29
3 2 _ _ _ _ _ 8 4	31 32 33 34 35 36 37 38 39
-----+-----+-----	-----+-----+-----
_ 6 _ 1 _ 5 _ 4 _	41 42 43 44 45 46 47 48 49
_ _ 8 _ _ _ 5 _ _	51 52 53 54 55 56 57 58 59
_ 7 _ 8 _ 3 _ 1 _	61 62 63 64 65 66 67 68 69
-----+-----+-----	-----+-----+-----
4 5 _ _ _ _ _ 9 1	71 72 73 74 75 76 77 78 79
6 _ _ 5 _ 8 _ _ 7	81 82 83 84 85 86 87 88 89
_ _ 3 _ 1 _ 6 _ _	91 92 93 94 95 96 97 98 99

Existem nove "blocos", que são os quadrados 3×3 com centros em 22, 25, 28, 52, 55, 58, 82, 85, 88. Isto é, os "blocos" são os nove conjuntos de posições [11, 12, 13, 21, 22, 23, 31, 32, 33], [14, 15, 16, 24, 25, 26, 34, 35, 36], ... (mais sete conjuntos)

1. **Represente**, em Prolog, este problema como um problema de restrições. Em particular, indique como representa os **estados**, as **variáveis** e as **restrições**.
2. Considere o **estado inicial** descrito acima. Resolva este problema com o algoritmo **backtrack** (implementado em Prolog).
3. Resolva o problema anterior com o algoritmo **forward check** (implementado em Prolog).
4. Procure, na *web*, outros quatro problemas **sudoku** e use os programas que escreveu para os resolver. Indique a **origem** dos problemas, como definiu os **estados iniciais** e as **instruções** que usou para encontrar as soluções.