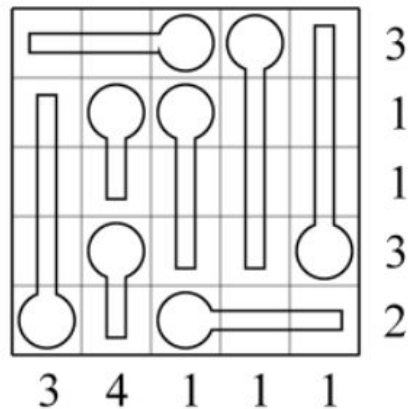


Projecto Termómetros

Problema

Preencher quadrados da matriz, tal que:

- Cada linha tenha N_l quadrados
- Cada coluna tenha N_c quadrados
- Os termómetros estejam preenchidos da base para cima



Solução

- **Naive Solution:**
 - Gerar combinações possíveis
 - Testar se são válidas
- **Good Solution:**
 - Implementar os funções predicados do enunciado

Input

Um puzzle/problema de termómetros é modelado como uma lista com:

- Lista de Termómetros
 - Termómetro é uma lista de posições (L,C)
 - Posição 1 representa base do termómetro
- Lista de valores totais das linhas
- Lista de valores totais das colunas

Predicados a desenvolver

propaga(Puzzle, Posição, Posições)

- Dado um **Puzzle**, e uma **Posição**:
 - Descobrir quais as **Posições** que **Posição** depende

Sugestão:

1. Encontrar o termómetro a que **Posição** pertence
2. **Posições** são as posições desde o início do termómetro até **Posição**
3. Ordenar os resultados!

nao_altera_linhas_anteriores(Poss, Linha, Preench)

- Dada uma possibilidade de preenchimento **Poss** para a linha **Linha**
 - Todas as entradas de **Poss** em linhas anteriores a **Linha**, devem estar preenchidas

Sugestão:


1. Encontrar uma sublista (**Sub**) de **Poss**, com os elementos de linhas anteriores à linha **Linha**
2. Todos os elementos de **Sub**, devem ter sido preenchidos (**Preench**)

verifica_parcial(Puzzle, Preenchidas, Dim, Poss)

Verificar se os valores das colunas são respeitados, dado:

- Um **Puzzle** de dimensões **Dim**,
- Uma lista de peças já **Preenchidas**,
- Uma possibilidade de preenchimento **Poss**.

Sugestão:

1. Unir **Poss** às já **Preenchidas**  **Option**
2. Para cada coluna **J** (**Dim** colunas):
 - a. Ver se **Option** respeita o valor da coluna
 - i. Ver as opções com coluna J

Don't reinvent the wheel!

Predicados Úteis

- `sort(List, ListSorted).`
- `member(E1, List).`
- `union/intersection(List1, List2, Result).`
- `subtract(List1, List2, Result).`
- `nth1(Index, List, Element).`
- `length(List, Len).`
- `append(List1, List2, Result).`
- `findall(E1, Goal, ListMatched).`
- `include(Goal, List1, Result).`
- `foreach(between(1, N, I), Goal(I)).`