

ProglIIIG1 - Act08 - Informe

Modelado y Resolución del Juego de Kakuro como CSP utilizando Python

Grupo 1 - Programación III

Universidad Tecnológica de Pereira

Programa: Ingeniería de Sistemas y Computación

Integrantes:

- Kenneth Santiago Rodríguez Ramírez
- Ramiro Antonio Pinchao Chachinoy
- Brayan Suárez Ceballos

1. Introducción

Kakuro es un juego de lógica numérica en el que se deben completar celdas blancas con números del 1 al 9, de modo que la suma de estos números coincida con las pistas dadas y sin repetir valores en cada grupo. En esta actividad, se modela Kakuro como un Problema de Satisfacción de Restricciones (CSP), implementando una solución automatizada en Python capaz de resolver tableros de dificultad "Muy difícil" del portal Sudokumania.com.ar.

2. Modelado del Problema como CSP

Variables: Cada celda blanca jugable del tablero representa una variable.

Dominios: Cada variable puede tomar un valor del conjunto $\{1, 2, \dots, 9\}$.

Restricciones:

- - La suma de los valores en un grupo horizontal o vertical debe coincidir con la pista proporcionada.
- - Los valores en cada grupo deben ser únicos (restricción tipo all-different).

3. Desarrollo

La solución fue implementada en Python, utilizando clases para modelar los distintos tipos de celdas y estructuras para representar el tablero y las restricciones. Se utilizaron las siguientes técnicas:

- - Backtracking con Forward Checking.

- - Heurística MRV (Minimum Remaining Values).
- - Reducción inicial de dominios.

El tablero se lee desde un archivo de texto plano con un formato definido por el grupo:

- "0" para celdas negras.
- "v|h" para celdas con sumas vertical/horizontal.
- "#" para celdas blancas jugables.

Ejemplo: 0|17, 43|0, 0, #

El resolutor trabaja con tableros de 9x9 y procesa secuencialmente los tres tableros definidos.

4. Resultados

Se ejecutó la solución para tres tableros de dificultad "Muy difícil" con los códigos:

- - KK5HVVCE
- - KK5IXCOC
- - KK5LMRDV

El sistema cargó exitosamente los archivos:

ProgIIG1-Act08-KK5HVVCE-Board.txt

ProgIIG1-Act08-KK5IXCOC-Board.txt

ProgIIG1-Act08-KK5LMRDV-Board.txt

En cada caso se mostró el tablero inicial y la solución encontrada, incluyendo el tiempo de ejecución promedio entre 1 y 3 segundos. Las soluciones cumplen con todas las restricciones del juego.

5. Conclusiones

- - Se reafirmó la utilidad del modelado como CSP para estructurar y resolver problemas que involucran múltiples restricciones interdependientes.
- - La correcta representación del dominio y las restricciones en estructuras de datos apropiadas facilitó la extensión, validación y reutilización del código.
- - Se comprobó que Python es una herramienta adecuada para abordar problemas complejos de satisfacción de restricciones, incluso sin el uso de librerías externas.
- - Se resolvieron correctamente tableros de alta complejidad.

6. Anexos

- Código fuente: [ProgIIG1-Act08-Codigo](#)

- Tableros de prueba en formato .txt: [ProgIIG1-Act08-Boards](#)

- Video de sustentación: [ProgIIG1-Act08-Sustentacion](#)