

Práctica 4 – Problemas de satisfacción de restricciones

Introducción

En esta práctica ensayaremos la formulación y solución de problemas en términos de restricciones. Solucionaremos el problema de coloreado de mapas visto en clase, y también desarrollaremos un programa para resolver Sudokus utilizando programación con restricciones.

Instrucciones

Descargar el zip con el esqueleto de la práctica desde el campus virtual. Descomprimir y correr todos los tests con `mvn test`. Observar que todos los tests en el paquete `ssii.p4.maps` pasan, y algunos tests del paquete `ssii.p4.sudoku` fallan.

Importar el proyecto en Eclipse. Realizar las siguientes modificaciones:

- Examina el código de `aima.core.search.csp.MinConflictsStrategy`, `aima.core.search.csp.BacktrackingStrategy`, y `aima.core.search.csp.ImprovedBacktrackingStrategy` para ver cómo se corresponden con los algoritmos vistos en clase.
- Realiza las siguientes modificaciones sobre las clases `MapCSP` y `MapCSPTest`. Tras las modificaciones asegúrate de que todos los tests pasan, y se ejecutan rápidamente.
 - Añade nuevos tests a `MapCSPTest` en el que se modifica `MapCSP` para incluir una nueva restricción en `MapCSP` para que WA no pueda tener el mismo color que V. ¿Encuentran la misma solución los tres algoritmos `MinConflictsStrategy`, `BacktrackingStrategy` e `ImprovedBacktrackingStrategy`? Añade aserciones en el tests acerca de la solución encontrada con cada algoritmo.
 - Añade nuevos tests a `MapCSPTest` para además de la restricción `WA != V` incluir un nuevo color `ORANGE`, probando con los 3 algoritmos. ¿Encuentran la misma solución los tres algoritmos? Añade aserciones en el tests acerca de la solución encontrada con cada algoritmo.
 - Añade nuevos tests a `MapCSPTest` para además de `WA != V` y un nuevo color `ORANGE`, incluir una nueva región `XX` que se sitúa al lado de `WA` y `V`, probando con los 3 algoritmos. ¿Encuentran la misma solución los tres algoritmos? Añade aserciones en el tests acerca de la solución encontrada con cada algoritmo.
- La clase `SolvedSudoku` representa un puzzle Sudoku resuelto. Examina las reglas del Sudoku, y modifica el método `checkSudokuSolved` de la clase `SolvedSudokuTest` para que compruebe correctamente si un objeto `SolvedSudoku` corresponde a un Sudoku resuelto correctamente. Puedes añadir los métodos que necesites a la clase `SolvedSudoku`, pero no puedes modificar los métodos de test de la clase `SolvedSudokuTest` ni las variables `PUZZLE_*`. Todos los test de `SolvedSudokuTest` deben pasar tras estas modificaciones.
- Modifica el método `setupConstraints` de `SudokuCSP` para añadir las restricciones necesarias para resolver el puzzle Sudoku. Añade los métodos adicionales que necesites a la clase `SudokuCSP`, pero no modifies la clase `SudokuCSPTest`. Tras estas modificaciones todos los test del proyecto deben pasar.

- Añade tests adicionales a la clase SudokuCSPTTest usando los algoritmos `MinConflictsStrategy`, `BacktrackingStrategy` e `ImprovedBacktrackingStrategy` sobre los problemas `easyPuzzle`, `obviousPuzzle`, `harderPuzzle`, `hardestPuzzle`, `noSolutionsPuzzle`, y `noSolutionsPuzzle2` definidos en la clase `SudokuCSPTTest`. Para `ImprovedBacktrackingStrategy` prueba con distintos argumentos de su constructora. Añade aserciones sobre para que puzzles se encuentra solución y para cuales no. Si algún algoritmo no es capaz de encontrar la solución en un tiempo razonable, deshabilita el test correspondiente quitando la anotación `@Test`. Aquí de lo que se trata es de determinar qué algoritmos y con qué parámetros son capaces de ejecutar qué tests.

Presentación de la práctica

Subir a la entrega en el campus virtual un zip con la carpeta del proyecto (la que contiene el archivo `pom.xml`). Solo hace falta que uno de los miembros del grupo realice la entrega en el campus virtual. La práctica deberá estar modificada como sigue:

- El proyecto se puede importar en Eclipse correctamente.
- Todos los tests pasan, y se ejecutan rápidamente.
- Se han realizado todas las modificaciones indicadas en las instrucciones, y pasan todos los tests.
- Se han añadido los tests adicionales indicados en las instrucciones.

También añade un archivo `preguntas.txt` al zip que se entrega en el campus virtual, con respuestas a las siguientes preguntas.

1. ¿Qué nuevos tests de `MapCSPTTest` pasan y cuales no?
2. ¿Qué sudokus tienen solución y cuales no? ¿Todos los algoritmos son capaces de resolver todos los sudokus rápidamente, o algunos no?
3. ¿Qué algoritmo es más rápido? ¿Qué valores de los argumentos de la constructora de `ImprovedBacktrackingStrategy` funcionan mejor?, ¿funcionan igual para todos los problemas?
4. ¿Son deterministas todos los algoritmos?

Para que quede claro que respuesta corresponde a cada pregunta en el archivo `preguntas.txt`, copia en dicho archivo la lista de preguntas y añade cada respuesta a continuación de cada pregunta.