

Resolviendo Sudoku con SAT

1. Objetivo

El primer objetivo del proyecto es la implementación de un resolutor para el problema de SAT. El segundo objetivo es el hacer un resolutor para Sudoku, que basado en resolutor de SAT.

2. Actividad 1: Resolutor para de SAT

Debe implementar un resolutor del problema de SAT, el cual debe estar basado en el algoritmo DPLL presentado en [3]. El resolutor debe ser llamado `dp11Solver`, el cual se ejecuta con la siguiente línea de comando:

```
>./dp11Solver <archivo-cnf>
```

Donde `archivo-cnf` es un archivo con una fórmula booleana en CNF, en el formato de DIMACS simplificado. Las características de este formato son descritas aquí: <http://www.satcompetition.org/2009/format-benchmarks2009.html>. La salida de `dp11Solver` tiene dos líneas. En la primera debe imprimir la palabra SATISFIABLE o UNSATISFIABLE, si el problema de entrada es satisfacible o no. La segunda línea indica el tiempo usado por el resolutor, en segundos, para obtener la solución.

La librería SATLIB, contiene un buen número de instancias en CNF, que pueden servir como pruebas de su resolutor de SAT. Aquí el enlace: <https://www.cs.ubc.ca/~hoos/SATLIB/benchm.html>.

El SAT solver debe ser programado de forma tal, que pueda ser usado como librería por otro programa. Esto es necesario para la actividad 2.

3. Actividad 2: Resolutor para Sudoku

En esta segunda parte del proyecto, se quiere usar el resolutor para SAT para resolver Sudoku. Es decir, se quiere hacer un resolutor de Sudoku basado en SAT. La Figura 1 muestra los pasos para resolver instancias de Sudoku con un resolutor SAT. En [1] y en [2] se explica como codificar una instancia en Sudoku en una instancia en CNF. A continuación se describen los pasos:

1. El resolutor recibe como entrada una instancia de Sudoku.
2. La instancia se codifica en una instancia en CNF.
3. La instancia en CNF es resuelta por un SAT solver.
4. El resultado del SAT solver se decodifica en una solución de Sudoku
5. La solución de la instancia de Sudoku es mostrada al usuario.

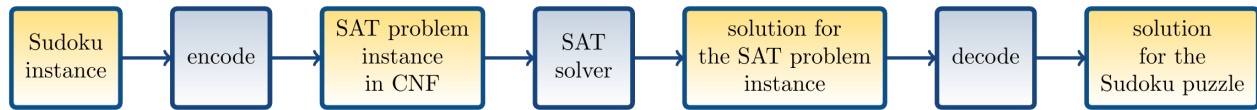


Figura 1: Pasos para resolver una instancia de Sudoku usando un resolutor de SAT. Fuente: [1]

```
>./sudokuSolver [instancia]
```

El resolutor de Sudoku debe ser llamado **sudokuSolver**, el cual se ejecuta con la siguiente línea de **comando**. Donde **instancia** es un string que es una representación de una instancia de Sudoku. Si usa en su implementación un lenguaje interpretado como Java, entonces **sudokuSolver** puede ser un shell script. En el string las filas del Sudoku se colocan en una sola fila, y cuando en la posición de la instancia hay un espacio en blanco se coloca un punto, y en caso contrario el número correspondiente. Por ejemplo, el string

```
1.....2.9.4...5...6...7...5.9.3.....7.....85..4.7.....6...3...9.8...2.....1
```

representa la instancia de Sudoku de la Figura 2.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | | | 2 |
| | 9 | | 4 | | | | 5 | |
| | | 6 | | | | 7 | | |
| | 5 | | 9 | | 3 | | | |
| | | | | 7 | | | | |
| | | | 8 | 5 | | | 4 | |
| 7 | | | | | | 6 | | |
| | 3 | | | | 9 | | 8 | |
| | | 2 | | | | | | 1 |

Figura 2: Instancia de Sudoku

La salida del resolutor debe ser un string con la asignación de todas las casillas, en una sola línea. Por ejemplo, la salida esperada de la 2 es:

```
174385962293467158586192734451923876928674315367851249719548623635219487842736591
```

4. Condiciones de la entrega

El código del proyecto debe incluir un archivo llamado **Leeme.txt**, que indica cómo compilar el proyecto. El código del proyecto y la declaración de autenticidad debidamente firmada, deben estar contenidos en un archivo comprimido, con formato *tar.xz*, llamado *Proy3.X.Y.tar.xz*, donde *X* y *Y*

son los número de carné de los estudiantes. La entrega de proyecto, debe hacerse por medio de la plataforma *Classroom* antes de las 10:00 A.M. del día domingo 06 de abril de 2025.

Referencias

- [1] JUNTILA, T. Solving problems with cnf sat solvers: The sudoku example. <https://users.aalto.fi/~tjunttil/2020-DP-AUT/notes-sat/solving.html>. Visitado el 22 de marzo de 2025.
- [2] LYNCE, I., AND OUAKNINE, J. Sudoku as a sat problem. In *AIEM* (2006), Citeseer.
- [3] RUSSELL, S., AND NORVIG, P. *Artificial Intelligence a Modern Approach*, 4ra ed. Pearson, 2022.