

# Uso de Sat Solvers para resolución de Sudoku

Bernardo Morales      Leonardo Ramos      Rubén Serradas

12 de junio de 2014

## 0.1. Codificador para SAT Solvers

Para poder resolver una gran cantidad de Sudokus en corto tiempo se utilizó un SAT Solver, un programa que resuelve problemas de satisfactibilidad booleana, en nuestro caso **glucose**.

Sin embargo, un programa como **glucose** solo resuelve problemas en formato DIMACS-CNF, es decir, un archivo de entrada en el que cada línea representa una disjunción. Para esto se escribió un Codificador/Decodificador en Python para representar las reglas de Sudoku en formato DIMACS-CNF. Las reglas consideradas fueron:

- Cada casilla contiene un número entre el 1 al 9.
- No puede haber varios números en la misma casilla.
- En la misma fila no puede estar el mismo número.
- En la misma columna no puede estar el mismo número.
- En el mismo bloque no puede estar el mismo número.

El programa `encoder.py` escribe estas reglas en formato DIMACS-CNF en un archivo temporal para que el SAT Solver lo resuelva. Para ejecutar `encoder.py` tan solo es necesario escribir en el terminal `encoder.py` y el programa imprimirá un mensaje de ayuda indicando de que manera se le debe llamar a este. Se probó el programa para que resolviera el archivo “sudoku\_10k.txt”, midiéndose el tiempo que tomaba el SAT Solver en resolver cada una de las diez mil instancias. Los resultados obtenidos se encuentran en la siguiente tabla:

Mínimo (s.)	0.004859
Primer Cuartil (s.)	0.005362
Mediana (s.)	0.005541
Media (s.)	0.005783
Tercer Cuartil (s.)	0.005866
Máximo (s.)	0.103300

Tabla 1: Tabla resumen de los resultados obtenidos para el archivo “sudoku\_10k.txt”

Sumando todos los resultados de las instancias obtenidos por el SAT Solver tenemos un tiempo de 57.8325929642 segundos. La diferencia entre los tiempos obtenidos en cada instancia es mínima como se puede observar por los valores obtenidos en los cuartiles, el tiempo máximo no es muy elevado igualmente.