

Universidad Simón Bolívar, 13 de junio de 2014

Trimestre: Abril-Julio

Profesor: Blai Bonet

Elaborado por:

- Arturo Voltattorni 10-10774
- Fernando D'agostino 10-10812
- Oswaldo Jiménez 10-10368

### **INFORME - PROYECTO 3. CNF, SAT**

La solución propuesta por el equipo consta de 3 partes fundamentales: un generador de la teoría base, un codificador, y un decodificador.

- La teoría base es generada de acuerdo a las restricciones que caracterizan una configuración válida de números en el sudoku tradicional, considerando en primer lugar que un número no puede repetirse en una fila, en segundo lugar que no puede repetirse en una columna, y en tercer lugar que no puede repetirse en una cuadrícula 3x3. Cada restricción corresponde a una cláusula, interpretadas inicialmente como implicaciones y traducidas subsecuentemente en disyunciones. Se involucran un total de 729 literales (9 posibilidades por cada posición), y en general las cláusulas generadas poseen a lo sumo tamaño  $k=2$ , a excepción de las cláusulas que indican la existencia de un número en una casilla (éstas son de tamaño nueve y corresponden aproximadamente a la centésima parte del total de cláusulas) .
- El codificador se encarga de, dada una instancia de entrada, generar la colección de cláusulas a ser suministradas al solucionador. Para dicha tarea, se extraen las cláusulas unitarias encontradas en el archivo de entrada y se adjunta la teoría base concretada en el paso anterior. La colección se le proporciona al solucionador, recogiendo los resultados que éste arroja.
- El decodificador tiene por objetivo traducir el resultado arrojado por el solucionador a la cuadrícula de sudoku resuelta de la instancia suministrada.

La integración de estos elementos se llevó a cabo a través de código en bash scripting, en el que se compilan y ejecutan los programas pertinentes.

La solución de cada instancia se ve reflejada en un archivo de salida, cuyo nombre es especificado por el usuario al momento de invocar el código bash. Del mismo modo, se muestra en pantalla el tiempo que demora el análisis de cada instancia de sudoku. El cálculo del tiempo total se efectuó en base a lo que se demoró el solucionador zchaff en analizar cada instancia de sudoku, a fin de evaluar su eficacia en función de las cláusulas proporcionadas.

Con respecto al desempeño del solucionador, se concluye que el cuantioso número de cláusulas que le fue suministrado le permite determinar de forma rápida y efectiva las valuaciones que satisfacen las condiciones iniciales fijadas en cada instancia, pues se ven restringidas

considerablemente las posibilidades de valuación y, en consecuencia, se reduce la carga de análisis del solucionador.

El equipo de trabajo pudo comprender la manera en que se solucionan problemas de sudoku usando SAT solvers y una traducción de las instancias a CNF. Asimismo, adjuntamos un esquema explicativo de como el SAT solver ayuda a resolver instancias de sudoku:

Lo primero que se debe resaltar es que se tiene a disposición una teoría que se encuentra en CNF. Dicha teoría se puede dividir en tres partes. La primera, son todas aquellas cláusulas resultantes de las restricciones del propio juego de sudoku, es decir, las restricciones de línea, columna y cuadro, estas cláusulas son de tamaño 2 y todos los literales aparecen negados. En segundo lugar, están las restricciones para que en cada casilla exista al menos un número del 1 al 9, estas cláusulas son de tamaño 9 donde todos sus literales aparecen sin negar. Finalmente, la tercera parte será el conjunto de cláusulas unitarias (sin negar) que son particulares para cada instancia de sudoku.

De aquí, podemos utilizar estas cláusulas unitarias propias para cada instancia en conjunto con las cláusulas asociadas a las restricciones para obtener cláusulas unitarias (negadas) que nos permiten determinar que, por ejemplo, el número 1 está en la casilla (1,1) y que las demás variables asociadas al número 1 en esa fila, columna y cuadro deben ser falsas.

Como estas deben ser falsas, podemos aplicar resolución con las cláusulas de tamaño 9. Reduciendo así el tamaño de las mismas y eventualmente determinando el número correcto para una casilla en particular. De ahí, este procedimiento se repite hasta finalmente llegar a una configuración que cumpla con las restricciones y que indique a través de las variables que sean true los números correspondientes a cada casilla.