

Inteligencia Artificial

Proyecto III

Sudoku

Simón Castillo
09-10147

Alejandro Flores
09-10298

Jonathan Queipo
09-10684

Noviembre 22, 2012

El presente proyecto tiene como finalidad resolver instancias de Sudoku, utilizando un solucionador *SAT*. El solucionador seleccionado fue *Glucose 2.1*, primer lugar en el *2012 SAT Challenge* en el área de aplicaciones.

Sea $V_{i,j,k}$ la variable proposicional que representa si la casilla i, j contiene el valor k , donde $i, j, k \in \{1 \dots 9\}$.

- Valores iniciales del tablero (Siendo S_0 el estado inicial).

$$A \equiv \bigwedge_{(i,j,k) \in S_0} V_{i,j,k}$$

- Cada casilla puede tener solo un valor asignado k .

$$B \equiv \bigwedge_{\substack{i,j,k,k' \in \{1..9\}, \\ k \neq k'}} \neg V_{i,j,k} \vee \neg V_{i,j,k'}$$

- Por cada fila, solo una casilla puede tener el valor k .

$$C \equiv \bigwedge_{\substack{i,j,j',k \in \{1..9\}, \\ j \neq j'}} \neg V_{i,j,k} \vee \neg V_{i,j',k}$$

- Por cada columna, solo una casilla puede tener el valor k .

$$D \equiv \bigwedge_{\substack{i,i',j,k \in \{1..9\}, \\ i \neq i'}} \neg V_{i,j,k} \vee \neg V_{i',j,k}$$

- Por cada subcuadrado (3×3), solo una casilla puede tener el valor k .

$$E \equiv \bigwedge_{\substack{k \in \{1..9\}, x, y \in \{0..2\}, \\ i, i' \in I_x, j, j' \in J_y}} \neg V_{i,j,k} \vee \neg V_{i',j',k}$$

Donde:

$$I_i = \{ i * 3 + k \mid k \in [1 \dots 3] \}$$

$$J_j = \{ j * 3 + k \mid k \in [1 \dots 3] \}$$

- Cada casilla (i, j) debe tener un valor $k \in \{1 \dots 9\}$ asignado.

$$F \equiv \bigwedge_{i,j \in \{1..9\}} \bigvee_{k \in \{1..9\}} V_{i,j,k}$$

Finalmente las cláusulas son:

$$A \wedge B \wedge C \wedge D \wedge E \wedge F$$

Para que el solucionador pueda comprender dichas cláusulas deben estar en texto de formato DIMACS, dicho texto lo generamos utilizando un programa (*encoder*) escrito en el lenguaje *Haskell*. Para cada instancia, *Glucose 2.1* nos devuelve la solución, también en formato DIMACS, a dicha instancia del tablero de *Sudoku*, la cual traducimos también con otro programa (*decoder*) escrito en *Haskell*.

A continuación presentamos los resultados de nuestras pruebas:

Instancias de Sudoku	Total (segundos)	Promedio (segundos)
Difíciles (95 instancias)	6.546	0.0689
Sencillas (10000 instancias)	643	0.0643