A SAT-based Sudoku Solver

Tjark Weber

webertj@in.tum.de

Club2, November 23rd, 2005
**Sudoku**

*Sudoku* is a placement puzzle

<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>3</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>9</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td></td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>8</td>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td></td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>2</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>9</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>2</td>
<td>1</td>
<td>9</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>9</td>
<td>8</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>9</td>
<td>7</td>
<td>6</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>6</td>
<td>8</td>
<td>5</td>
<td>3</td>
<td>7</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>3</td>
<td>9</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>7</td>
<td>2</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>7</td>
<td>4</td>
<td>1</td>
<td>9</td>
<td>6</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>8</td>
<td>6</td>
<td>1</td>
<td>7</td>
<td>9</td>
</tr>
</tbody>
</table>
Motivation

- *Sudoku* is extremely popular
- more than $6 \cdot 10^{21}$ valid $9 \times 9$ grids
- *Sudoku* is NP-complete
- existing *Sudoku* solvers are often slow or complicated (or both)
Motivation

- *Sudoku* is extremely popular
- more than $6 \cdot 10^{21}$ valid $9 \times 9$ grids
- *Sudoku* is NP-complete
- existing *Sudoku* solvers are often slow or complicated (or both)
- an efficient, yet easy to implement *Sudoku* solver
Tools

To ease implementation:

- **Isabelle/HOL**: an interactive theorem prover for higher-order logic,
  - with support for SAT-based finite model generation

For efficiency:

- **zChaff**: a leading SAT solver
System Overview

Isabelle

Sudoku → Preprocessing → DIMACS CNF → zChaff

Solution → Postprocessing → Assignment

Not solvable → satisfiable?

yes → Solution

no → Not solvable
Implementation in Isabelle/HOL

Def.

valid\( (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9) \equiv \bigwedge_{d=1}^{9} \bigvee_{i=1}^{9} (x_i = d) \)
Implementation in Isabelle/HOL

Def.

\[
\text{valid}(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9) \equiv \bigwedge_{d=1}^{9} \bigvee_{i=1}^{9} (x_i = d)
\]

Def.

\[
\text{Sudoku}(\{x_{ij}\}_{i,j\in\{1,...,9\}}) \equiv \bigwedge_{i=1}^{9} \text{valid}(x_{i1}, x_{i2}, x_{i3}, x_{i4}, x_{i5}, x_{i6}, x_{i7}, x_{i8}, x_{i9})
\]

\[
\bigwedge_{j=1}^{9} \text{valid}(x_{1j}, x_{2j}, x_{3j}, x_{4j}, x_{5j}, x_{6j}, x_{7j}, x_{8j}, x_{9j})
\]

\[
\bigwedge_{i,j\in\{1,4,7\}} \text{valid}(x_{ij}, x_{i(j+1)}, x_{i(j+2)}, x_{(i+1)j}, x_{(i+1)(j+1)}, x_{(i+1)(j+2)}, x_{(i+2)j}, x_{(i+2)(j+1)}, x_{(i+2)(j+2)})
\]
Translation to SAT

- 9 Boolean variables for each grid cell
- Each Boolean variable $p_{ij}^d$ (with $1 \leq i, j, d \leq 9$) represents the truth value of the equation $x_{ij} = d$.
- A clause $\bigvee_{d=1}^{9} p_{ij}^d$ ensures that the cell $x_{ij}$ denotes one of the nine digits.
- 36 clauses $\bigwedge_{1 \leq d < d' \leq 9} (\neg p_{ij}^d \lor \neg p_{ij}^{d'})$ make sure that the cell does not denote two different digits at the same time.
Translation to SAT (2)

Short clauses allow for more unit propagation.

Lemma.

\[
\text{valid}(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9) \equiv \bigwedge_{d=1}^{9} \bigvee_{i=1}^{9} (x_i = d)
\]

\[
\iff \bigwedge_{1 \leq i < j \leq 9} (x_i \neq x_j) \iff \bigwedge_{1 \leq i < j \leq 9} \bigvee_{d=1}^{9} (x_i \neq d \lor x_j \neq d).
\]

Total of . . .

- \(9^3 = 729\) Boolean variables, and
- \(9^2 + 9^2 \cdot 36 + 3 \cdot 9 \cdot 36 \cdot 9 = 11745\) clauses.
Evaluation and Conclusions

- a straightforward translation of a Sudoku into a propositional formula
- easy to generalize to grids of arbitrary dimension
- polynomial in the size of the grid
- easy to modify to enumerate possible solutions
- implementation (using Isabelle/HOL) almost trivial
- 9 × 9 Sudokus are solved within milliseconds