A SAT-based Sudoku Solver

Tjark Weber

webertj@in.tum.de

TECHNISCHE UNIVERSITÄT MÜNCHEN

Club2, November 23rd, 2005

Sudoku

- *Sudoku* is a placement puzzle

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

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

Motivation

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

- an efficient, yet easy to implement *Sudoku* solver

Tools

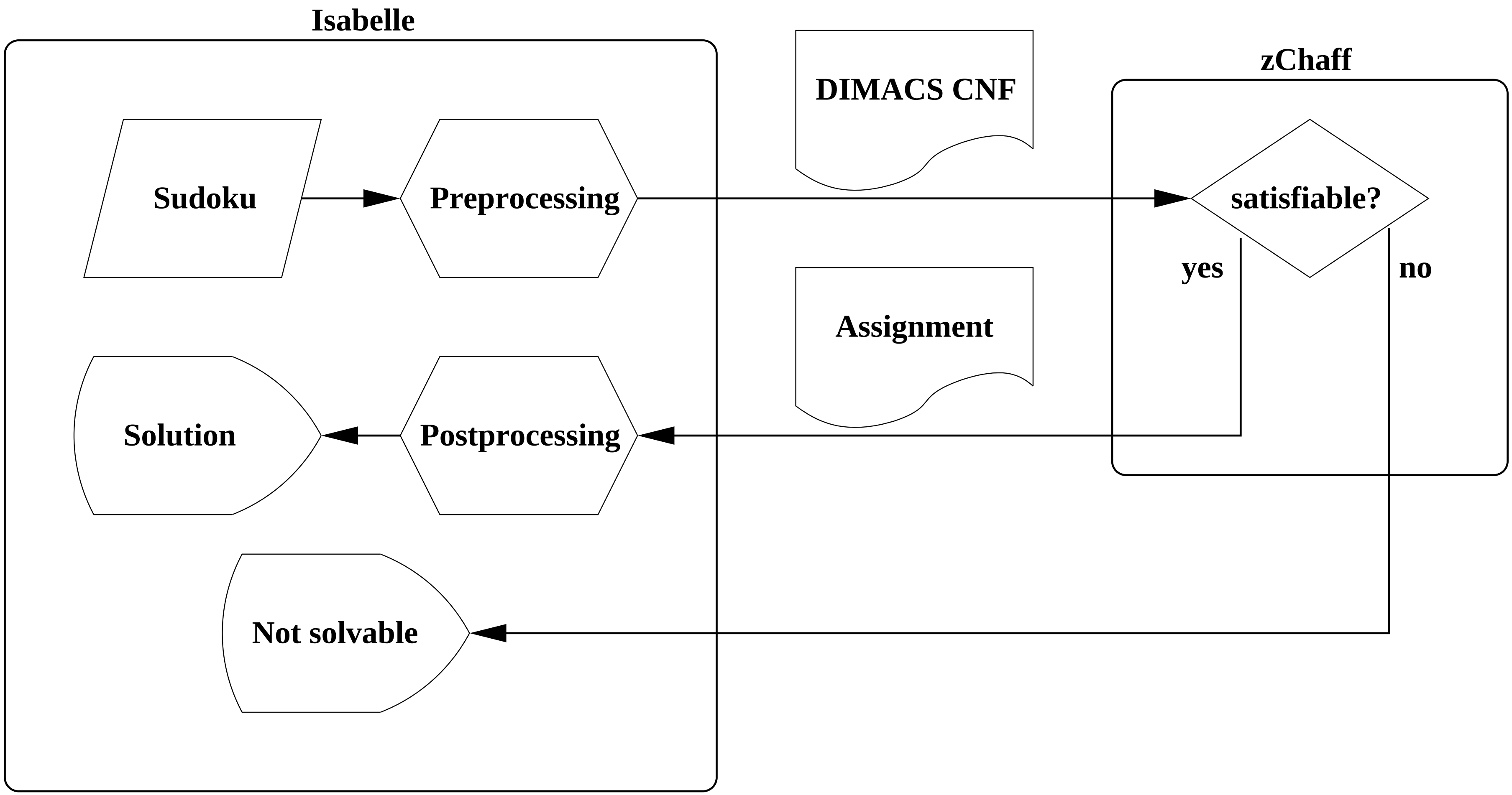
To ease implementation:

- 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



Implementation in Isabelle/HOL

Def.

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

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.

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

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

$$\wedge \bigwedge_{i,j \in \{1,4,7\}} \text{valid}(x_{ij}, x_{i(j+1)}, x_{i(j+2)}, x_{(i+1)j}, x_{(i+1)(j+1)}, x_{(i+1)(j+2)}, x_{(i+2)j}, x_{(i+2)(j+1)}, x_{(i+2)(j+2)})$$

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 $\bigwedge_{1 \leq d < d' \leq 9} (\neg p_{ij}^{d} \vee \neg p_{ij}^{d'})$ 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.

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

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

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 *Sudoku*s are solved within milliseconds