BMC meets Sudoku

朱倖民

Department of Computer Science and Technology,
Tsinghua University

March 26, 2016

Overview

Contents

Sudoku Game

- 9×9 board
- every cell should be filled in with a number range 1 to 9
- some cells are already given
- numbers 1 to 9 should appear in
  - each row
  - each column
  - each 3×3 box

Formalization

- Let problem P be a 9×9 matrix, each cell $p_{ij} \in \{0, 1, \ldots, 9\}$ for $i = 0, 1, \ldots, 8$.
- Let model B be a 9×9 matrix, where

$$b_{ij} = \begin{cases} x, & p_{ij} = 0 \\ p_{ij}, & p_{ij} \neq 0 \end{cases}$$

  where $x \in \{1, 2, \ldots, 9\}$.
- Find some B, say B^*, satisfying specification $b^*_{i_1 j_1} \neq b^*_{i_2 j_2}$ for all pairs $(i_1 j_1, i_2 j_2)$ that have **conflict**, here, $i_1, j_1, i_2, j_2 = 0, 1, \ldots, 8$.

Formalization

- Let problem P be a 9×9 matrix, each cell $p_{ij} \in \{0, 1, \ldots, 9\}$ for $i = 0, 1, \ldots, 8$.
- Let model B be a 9×9 matrix, where

$$b_{ij} = \begin{cases} x, & p_{ij} = 0 \\ p_{ij}, & p_{ij} \neq 0 \end{cases}$$

  where $x \in \{1, 2, \ldots, 9\}$.
- Find some B, say B^*, satisfying specification $b^*_{i_1 j_1} \neq b^*_{i_2 j_2}$ for all pairs $(i_1 j_1, i_2 j_2)$ that have **conflict**, here, $i_1, j_1, i_2, j_2 = 0, 1, \ldots, 8$.

Definition

Pair $(i_1 j_1, i_2 j_2)$ has **conflict** if $i_1 \neq i_2 \vee j_1 \neq j_2$, and

- $i_1 = i_2$, or
- $j_1 = j_2$, or
- $div(i_1, 3) = div(i_2, 3) \wedge div(j_1, 3) = div(j_2, 3)$.

SMV Description

```
MODULE main
VAR
    board: array 0..80 of 1..9;

ASSIGN
    board[0] := {5};
    board[1] := {3};
    board[2] := {1, 2, 3, 4, 5, 6, 7, 8, 9};
    ...

LTLSPEC
    !G (board[0] != board[1] & board[0] != board[2] & ...);
```

Solution

- If a counter-example is found, then the counter-example itself presents one solution.
- Else, no solution for the problem.

Contents

Motivation

- $b_{02} \in \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$

Motivation

- $b_{02} \in \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- $5, 3, 7, 8, 6, 9 \notin b_{02}$

Motivation

- $b_{02} \in \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- $5, 3, 7, 8, 6, 9 \notin b_{02}$
- $b_{02} \in \{1, 2, 4\}$

Optimizing

Definition

Given a problem P, the **candidate set** for cell p_{ij}

$$CS(p_{ij}) = \begin{cases} U - E, & p_{ij} = 0 \\ \{p_{ij}\}, & p_{ij} \neq 0 \end{cases}$$

where $U = \{1, 2, \ldots, 9\}$ and
$E = \{p_{i'j'} | (ij, i'j')$ has conflict $\wedge\, p_{i'j'} \neq 0, i', j' \in \{0, 1, \ldots, 8\}\}$.

Optimizing

Definition

Given a problem P, the **candidate set** for cell p_{ij}

$$CS(p_{ij}) = \begin{cases} U - E, & p_{ij} = 0 \\ \{p_{ij}\}, & p_{ij} \neq 0 \end{cases}$$

where $U = \{1, 2, \ldots, 9\}$ and
$E = \{p_{i'j'} | (ij, i'j') \text{ has conflict} \wedge p_{i'j'} \neq 0, i', j' \in \{0, 1, \ldots, 8\}\}$.

1. Compute candidate sets for each cell in P;
2. Find all cells p_{ij} which $PS(p_{ij}) = a$ where $a \in \{1, 2, \ldots, 9\}$, $p_{ij} \leftarrow a$;
3. Goto step 1 until P no longer changes.

Contents

Redefine Board

- Given a previous-defined board B.
- Let $S = (s_0, s_1, \ldots, s_{728})$, $s_t \in \{0, 1\}$.
- $s_{9(9i+j)+k} = 1$ iff $b_{ij} = k + 1$. Here, $i, j, k = 0, 1, \ldots, 8$.

Definition

S is a board if

- $|(s_0, s_1, \ldots, s_8)| = 1$, and
- $|(s_9, s_{10}, \ldots, s_{17})| = 1$, and
- $\ldots$
- $|(s_{720}, s_{721}, \ldots, s_{728})| = 1$.

The Rules

- Given the fact $b_{ij} = k + 1$.
- Let $R^{9(9i+j)+k} = (r_0, r_1, \ldots, r_{242})$, $r_t \in \{0, 1\}$.
- Case
  - when $0 \leq t < 81$, $\exists p, q (p, q = 0, 1, \ldots, 8)$ s.t. $t = 9p + q$. $r_t = 1$ iff number $q + 1$ appears on row $\#p$;
  - when $81 \leq t < 162$, $\exists p, q (p, q = 0, 1, \ldots, 8)$ s.t. $t = 81 + 9p + q$. $r_t = 1$ iff number $q + 1$ appears on col $\#p$;
  - when $162 \leq t < 243$, $\exists p, q (p, q = 0, 1, \ldots, 8)$ s.t. $t = 162 + 9p + q$. $r_t = 1$ iff number $q + 1$ appears on box $\#p$.

The Goal

- Construct a 728×243 matrix

$$M = \begin{pmatrix} R^0 \\ R^1 \\ \dots \\ R^{728} \end{pmatrix}.$$

- Our goal is to find a board S, select 81 rows $\#i$ from M if $s_i = 1$ s.t. 1 appears in each column of the selected sub matrix.

Contents

Conclusion

- Tell the model checker what to do rather than how to do.
- Bounded model checking works well for Sudoku.
- The simpler, the faster.

References

- NuSMV 2.6 User Manual.
  http://nusmv.fbk.eu/NuSMV/userman/v26/nusmv.pdf.
- Sudoku as a SAT Problem.
  http://sat.inesc-id.pt/~ines/publications/aimath06.pdf.
- 算法实践——舞蹈链（Dancing Links）算法求解数独.
  http://www.cnblogs.com/grenet/p/3163550.html.
- DLSS. https://github.com/paulzfm/DLSS.

Thank you!

Any questions?