Institute of Electronics, Information, and Communication Engineers

社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
SS2007-50(2007-12)

対話型埋込みによる数独問題の設計ツール

馬野　洋平†　酒井　正彦††　西田　直樹††　坂部　俊樹††　草刈 圭一朗††

†, †† 名古屋大学大学院情報科学研究科
〒 464–8603 名古屋市千種区不老町
E-mail: †umano@trs.cm.is.nagoya-u.ac.jp, ††{sakai,nishida,sakabe,kusakari}@is.nagoya-u.ac.jp

あらまし　近年，論理式の充足可能性判定ツール (SAT ソルバ) の高速化が進み，これを利用した数独パズルの解法が提案されている．本稿では，この解法を応用して試作した，対話的に数独パズルの問題を作成するツールについて報告する．このツールでは,「セルに埋めても矛盾を生じない数字の表示」・「削除しても問題が一意性を保つセルの表示」・「問題を手筋のみで解ける範囲の図示」の三つの主要機能を実装している．前者二つの機能は，SAT ソルバを利用した解法を応用して問題の矛盾・解の一意性を高速に検出することにより実現している．
キーワード　問題作成, 充足可能性, SAT ソルバ

A Tool for Designing Sudoku Problems by Interactive Fill-in Approach

Yohei UMANO†, Masahiko SAKAI††, Naoki NISHIDA††,
Toshiki SAKABE††, and Keiichirou KUSAKARI††

†, ††Graduate School of Information Science, Nagoya University
Furo–cho, Chikusa–ku, Nagoya, 464–8603, Japan
E-mail: †umano@trs.cm.is.nagoya-u.ac.jp, ††{sakai,nishida,sakabe,kusakari}@is.nagoya-u.ac.jp

Abstract　In recent years, several efficient SAT solvers, which decide satisfiability of boolean formulae, have been developed and used in solving Sudoku puzzles. In this paper, we present the interactive tool for designing Sudoku puzzles that we constructed experimentally by using a SAT solver. This tool contains three main functions: 'displaying numbers that can be filled in a cell without a contradiction', 'displaying cells without contributing the uniqueness', 'displaying a partial solution obtained by fundamental techniques.' The implementation of the former two functions relies on efficient checks of a contradiction or uniqueness of the given problem by using a SAT solver.
Key words　problem creation, satisfiability, SAT solver

1. はじめに

近年，数独問題 [7] のコンピュータによる自動生成が行われているが，その自動生成された問題は数独の作家が作成した問題に比べると面白くないという意見がある．実際にニコリでは作家が作成した問題を掲載している [6].

一方で論理式の充足可能性を高速に判定できる充足可能性判定ツール (SAT ソルバ) が発展してきた．これに伴い，数独パズルを SAT ソルバを利用して高速に解くという試みが行われている [1]〜[4].

本稿では，SAT ソルバの高速性を応用して試作した，対話的に数独パズルの問題を作成するツールについて報告する．このツールでは,「セルに埋めても矛盾を生じない数字の表示」・「削除しても問題が一意性を保つセルの表示」・「問題を手筋のみで解ける範囲の図示」の三つの主要機能を実装している．前者二つの機能は，計算量が多い処理である問題の矛盾・解の一意性の検出を SAT ソルバの利用により高速に調べている．本ツールでは表出数字の模様・表出数字の個数・適切な難易度を満たす問題の作成を目指し，それらの条件を満たす問題を短時間で作成可能であるかの実験を行う．

本稿は次のように構成される．2. 節では数独の定義と解法の説明をする．3. 節では作成ツールの設計方針について提示し，4. 節ではツールの実装方法について議論を行う．5. 節では試作したツールを用いての問題作成実験をし，6. 節では本ツールと他のツールとの比較を行う．7. 節では今後の課題について議論する．

NII-Electronic Library Service

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

図 1　数独問題の例

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

図 2　数独（図 1）の解

2. 数独とは

まず数独パズルに関する用語の定義を行う．数独の盤面はセルと呼ばれるマス目が 9×9 の配列で並んでいるものである．

盤面の左上を原点と考えることによりセルの位置を (i, j), $1 \leqq i \leqq 9$, $1 \leqq j \leqq 9$ で表す．各セルには 1 から 9 のいずれかの数字が一つ入るかまたは数字が入っていないことが許され，後者の状態のとき空白セルと呼ぶ．盤面は 3×3 の 9 個のブロックに分けられ，k 番目のブロック $(1 \leqq k \leqq 9)$ は $\{(3((k-1) \texttt{ mod } 3)+i, 3((k-1) \texttt{ div } 3)+j) \mid 1 \leqq i \leqq 3, 1 \leqq j \leqq 3\}$ のセルで構成される．盤面は縦と呼ばれる 9 個の列に分けられ，j 列目の縦は $\{(i,j) \mid 1 \leqq i \leqq 9\}$ で構成される．同様に 1 つの行を横といい，i 行目の横は $\{(i,j) \mid 1 \leqq j \leqq 9\}$ で構成される．解盤面とは，縦・横・ブロックのそれぞれが全て 1 から 9 の数字が重複することなく 1 種類ずつ入った空白セルのない盤面を表す．盤面の空白セルにどのような組み合わせで数字を入れても解盤面とはならないとき，その盤面は矛盾するという．また，盤面から得られる解盤面が一意に存在するときその盤面は一意であるといい，一意でないとは 1 つの盤面から複数の解盤面ができることをいう．一意な盤面を問題といい，問題に出現する数字を表出数字という．問題を解くとは，問題の全ての空白セルに数字を入れることにより解盤面を作成する事をさす．図 1 は数独の問題例であり，その解盤面を図 2 に示す．

数独の問題を人間が解く方法の一つとして，いくつかの手筋 [6] を組み合わせて解く方法がある．これらの手筋は基本的に，縦・横・ブロックにそれぞれ重複した数字は入りえないことを考慮して，特定のセルに入りうる候補をいかにして絞り込むためのものである．以下ではこれらのうち基本的な手筋をいくつか紹介する．

- 同じ列に空白が 1ヶ所

空白セルが 1ヶ所で残りの 8 個のセルには数字が入っている縦（横・ブロック）があれば，その空白セルの数字は残りの 8 個の数字中に含まれない数字に決まる．

- マーキング

1 から 9 のある数字に着目する．その着目した数字が既に入っているセルが属する縦・横・ブロックにはもうその数字は入りえない．そのことから盤上でその数字が出現しているそれぞれのセルの縦と横方向のセル全てに対して，「その数字はこのセルには入り得ない」という事を意味するマークを付ける．そしてマーク付けされていない空白セルが 1ヶ所だけになっているブロックがあれば，そのセル以外には着目した数字は入らないことからそのセルには初めに着目した数字が入る．この方法は，数独の問題を解く上の基本的な手筋として最も多用されている．

3. 作成ツールの設計方針

本節では，本ツールにより作成したい数独問題の目標とツールの概要を述べる．

3.1 作成したい問題

数独の問題として品質の高い問題とは，問題の視覚的な模様が幾何学的など特徴的であり，表出数字の個数が少なく，難易度が適切な問題だと考えた．

- 問題の模様

表出数字がばらばらに並んでいるよりも，盤面が上下左右に対称形であったり回転対称など幾何学的な模様であるほうが，問題を一目見ただけで形状の美しさを把握しやすいことから問題の模様が美しいといえる．このことから問題の価値を大きく左右するのは問題の模様と考えられる．図 1 はこの対称形をした問題となっている．対称形以外にも，文字や記号の形をしている場合などが美しい模様と考えられる．また，どのセルに数字が埋まっているかだけではなく，何の数字が入っているかにも着目できる．隣り合ったセルについて，数字が埋まっている部分は 1,2,3,…のように可能な限り連続した数字の並びになっているとより美しいといえる．

- 表出数字の個数

一般に，表出数字の個数が多い問題は比較的容易に作成可能であるが，個数が少ない問題は作成するのが困難である．問題の解は一意である必要があるため，表出数字の個数が少なくなるに従って解の一意性を保つことが難しくなるからである．その意味で，表出数字の個数が少ないほど問題の価値がより高い．

表出数字の個数が最小の問題として，17 個のものが知られている．表出数字の個数が 16 個の問題が存在するかどうかは分かっていない [5]．

- 適度な難易度

一般にパズルは難易度は高いほうが良いと思われがちだが，その考えは必ずしも正しくはない．人間が解いて楽しさを感じるものが評価され，人間には解けそうもないほど難解な問題は受け入れられない．表出数字の個数を減らすほど難易度は高くなりがちであるため，難易度を一定以下に抑えたまま表出数字の個数を減らすのは，問題を作成する上でも困難である．その意味では，難解な問題よりもむしろ易しめの問題の方が価値が高いといえる．

NII-Electronic Library Service

解き手の技量によっては，簡単な問題だけではなくやや難しめの問題を望むこともありうる．そのため必要に応じて適度な難易度の問題を提供することが理想的である．

3.2 本ツールによる数独問題の作成手順

本ツールでは，全てが空白セルの盤面から開始して，空白セルに一つずつ表出数字を埋めていくことによって問題を作成する．

問題作成の基本手順としては，まず数字を挿入していき一意性を持つまで大まかな形を作る．一意になった後は一意性を保ちつつ数字を追加・削除して模様を調節する．その後数字を追加することにより難易度を下げる．これを問題の模様や難易度に納得するまで続ける．

問題の模様と難易度の調節はどちらも作業方針はほぼ同じであるものの，問題の模様の調節には修正箇所が多いのに対して，難易度の調節の場合には一箇所の変更で大幅に難しさが変わりやすい．よってまず問題の模様をほぼ仕上げた後に難易度の調節を行う．また一意性を保ったまま数字を削減するのは削減できる場所が限られており制約が多いのに対し，一意性を保ったまま数字を追加するのは全く制約がなく容易である．そのことから表出数字を極力削減した状態を維持しつつ調節作業を行い，後から難易度を簡単にするために数字を追加できる余地を残すのがよい．

3.3 作成時に提示すべき情報

次に 3.2 節の作業のために有用なシステムが提示すべき情報について考察する．

- 数字の挿入（削除）時に提示する情報

初めに問題作成者は空白セルに数字を一個ずつ埋めていくことにより，大まかな模様をデザインする．この際各セルに適当な数字を入れていくという方法では，途中で矛盾が発生する可能性がある．そのためにセルの場所を選択してそのセルに数字を挿入するという作業過程において，場所を指定した段階でどの数字ならそこに入れても矛盾が発生しないかの情報を提示することは重要である．この表示された候補の中から数字を選んでいる限り，数字を埋めるセルの順番にかかわらず矛盾を生じることなく模様をデザインすることが可能となる．

問題作成の都合上あえて候補以外の数字を選ぶことも想定しているが，一度矛盾が発生した状態から矛盾しなくなるように修正するのは容易ではないため矛盾しない候補以外の数字を選択するのは推奨できる方法ではない．このようにして作成中の問題に矛盾が発生している場合はその情報を表示する．

セルへの数字の追加は，問題が一意性を持つまで続ける必要がある．逆に数字が入っているセル上の数字を削除すると，一意であった問題が一意ではなくなる可能性がある．よって一つの数字を挿入（削除）した直後に，盤面が一意性であるかどうかを表示する仕組みが有用である．これにより問題として要求される一意性を得ることが容易になる．

- 一意性を維持しつつ表出数字を削減する機能

問題が一意になった後は，表出数字の追加や削除を行って問題の模様を調節する．表出数字の追加については，前項の数字の追加の方法で一意性を有したまま任意の空白セルに数字を埋めることができる．一方表出数字の削除を行う際は，数字を削除することにより問題が一意ではなくなる可能性がある．問題が一意性を有しなくなるのでは再度一意になるまで数字を追加する必要があり作業効率が悪いため，一意性を保ったまま削除可能な表出数字を提示することは有効である．この機能により一意性を保ったまま表出数字を削除することが可能なセルをすぐに把握可能なことから，効率的に問題の模様設計や表出数字の個数を減らすことが行えると思われる．

- 難易度の調節を支援する機能

問題の難易度は一般に表出数字を新たに追加すれば問題が簡単になり逆に削減すれば難しくなるという性質がある．前項の機能を利用して一意性を保ったまま表出数字の追加・削除を行うことにより，模様と共に難易度が変化する．ここでは難易度を易しくするためには，どのセルに表出数字を追加するべきかについて考える．

難易度は 4.4 節で定義するように，どのレベルの手筋を用いるかで決定する．よって表出数字を追加する場所によっては，追加しても同じ手筋を用いている限り問題の難易度に変化がない可能性がある．そのため「あるレベル以下の手筋のみを利用して解ける範囲の図示」を行う機能があるとよいと考えた．この機能は作成した問題に対して指定したレベル以下の手筋のみを用いて解くことを試みて，問題の空白セルのうち解く事ができた部分のセルを表示する．この時指定したレベルより難解な問題の場合は全ての空白セルに数字が埋まらないが，つまりこの時解けなかった空白セルの部分に表出数字を追加すればこの手筋で解くことのできる範囲が一度に広がる可能性が高く，問題の難易度が下がることを意味する．一方このとき数字が埋まる部分に新たに表出数字を追加しても，同じ手筋で解くには難易度は本質的には簡単にならない．このように，あるレベルの手筋のみで解ける範囲を示す機能はどのセルに表出数字を追加するのが望ましいかを調べることが可能になる．

4. 実　　装

本節では，まず SAT ソルバによる数独の解法とそれを応用した一意性の判定法について述べ，前節で述べた機能の実現について説明する．

4.1 SAT ソルバ

問題作成時における矛盾と一意性の検出の際に必要な充足可能性判定には SAT ソルバの minisat v1.14 [14] を用いた．変数・否定記号・和記号・積記号で構成される論理式 $P(x_1, x_2, \cdots, x_n)$ に対して，P を真とする真偽値の割り当て $x_1, x_2, \cdots, x_n$ が存在するとき P は充足可能という．SAT ソルバはこれを判定し，充足可能な場合にはその割り当て方法を結果として出力する．

4.2 論理式への変換による数独の機械的解法

SAT ソルバは CNF(Conjuctive Normal Form) 形式の論理式を入力にとる．以下のようにして数独の問題を CNF 形式の論理式に変換する [1]．

変数 x_{rcv} を，盤面の左上から r 行 c 列のセル (r, c) の数字が v なら `true`, そうでなければ `false` とする．すると，数独の問題は表出数字のあるセル (r, c) について変数 x_{rc1} から x_{rc9} の

NII-Electronic Library Service

真偽値が初めから分かっており，その他の空白セルについての変数 x_{rcv} の値は未確定という状態に対応する．数独の問題を解いて解盤面を求めるとは変数 x_{rcv} の全ての値を求める事に対応し，その時の変数 x_{rcv} の真偽値から解盤面が求められる．すなわち，x_{rcv} の値が `true` の変数だけを取り出すと変数の個数は 81 個となり，取り出された変数名の r 行 c 列のセルの数字が v となる．以下では変数 x_{rcv} の組を $\vec{x} = (x_{111}, x_{112}, \cdots, x_{999})$ と表す．

解盤面を満たすための制約として，以下の 6 つの条件が必要である．括弧の中はその条件を表す名称とする．

- 表出数字が入る ($Assigned$)
- どのセルにも 1 から 9 のいずれかが入る ($Cell_d$)
- どのセルについても，重複した数字は入らない ($Cell_u$)
- どの行についても，重複した数字は入らない (Row_u)
- どの列についても，重複した数字は入らない (Col_u)
- どのブロックについても，重複した数字は入らない (Blo_u)

これらの条件を論理式に対応させると，以下のようになる．

- $Assigned(\vec{x}) = \bigwedge x_{rcv}$
  (表出数字 v が入っているセル (r, c) のみ)
- $Cell_d(\vec{x}) = \bigwedge_{1 \leq r \leq 9} \bigwedge_{1 \leq c \leq 9} \bigvee_{1 \leq v \leq 9} x_{rcv}$
- $Cell_u(\vec{x}) = \bigwedge_{1 \leq r \leq 9} \bigwedge_{1 \leq c \leq 9} \bigwedge_{1 \leq v_1 \leq 9} \bigwedge_{1 \leq v_2 \leq 9, v_2 \neq v_1} (\neg x_{rcv_1} \vee \neg x_{rcv_2})$

Row_u, Col_u, Blo_u についても $Cell_u$ と同様の方法で論理式になる．

これらの 6 つの論理式の論理積 $P(\vec{x}) = Assigned(\vec{x}) \wedge Cell_d(\vec{x}) \wedge Cell_u(\vec{x}) \wedge Row_u(\vec{x}) \wedge Col_u(\vec{x}) \wedge Blo_u(\vec{x})$ は，問題と一対一に対応付けられる．

4.3 矛盾と一意性の検出方法

数独の問題を表す論理式を $P(\vec{x})$ とし，この充足可能性を調べる．充足不能であれば問題に矛盾が生じており，充足可能であれば問題の解が一つ以上存在しており問題に矛盾は発生していないことが分かる．その場合に SAT ソルバから得られる変数の真偽値を $\vec{a} = (a_{111}, a_{112}, \cdots, a_{999})$ とする．

次に，この情報 $\vec{a}$ を用いて一意性の判定を行う．これは $P(\vec{x}) \wedge \neg(\vec{x} = \vec{a})$ で定義される論理式 $Q(\vec{x})$ の充足可能性を判定することにより可能である．ここで $\vec{x} = \vec{a}$ は $\bigwedge_{1 \leq r \leq 9, 1 \leq c \leq 9, 1 \leq v \leq 9} (x_{rcv} = a_{rcv})$ を表す論理式である．$Q(\vec{x})$ が充足不能であれば，$\vec{a}$ 以外には $P(\vec{x})$ の割り当てが存在しないため，数独の問題 $P(\vec{x})$ の解は一意である．$Q(\vec{x})$ が充足可能であれば，$\vec{a}$ 以外にも $P(\vec{x})$ に割り当てが存在し，問題 $P(\vec{x})$ は複数の解盤面を持つ．このように，SAT ソルバを 2 回実行する事により矛盾と一意性の検出が可能である．

4.4 ユーザーへの提示情報の計算法

ここでは 3.2 節で述べた問題作成者に提示する情報を，どのように計算するかについて述べる．

- 数字の挿入（削除）時に提示する情報の計算

数字を入力するためにセルを選択した時にそのセルにどの数字が入る可能性があるかを調べるが，その実現方法は次の通りである．作成途中の数独盤面を表す論理式を $P(\vec{x})$ として選択したセルの場所を r 行 c 列とすると，各 $v = 1, \ldots, 9$ に対してそれぞれ $P(\vec{x}) \wedge x_{rcv}$ の充足可能性を判定する．それにより選択したセルに入れても矛盾しない数字が分かる．なお，セルから縦・横・ブロック方向に同じ数字があるかどうかが自明な場合にはその数字に対しては SAT ソルバを呼び出さないようにしている．これにより，処理時間が削減される．

セルに数字を入力した時に一意性の検出を行う機能は，4.3 節で述べた方法で実装できる．その際，先ほどセルを選択した時の SAT ソルバによって求めた解を再利用することにより SAT ソルバの呼び出しを 1 回で済ませることが可能になる．

- 一意性を維持しつつ表出数字を削減する機能の実装

「数字を削除しても一意性を保つセルの検出」を行う機能は問題中の表出数字それぞれについて，問題からその表出数字を一個削除した盤面の一意性を調べることによって実現している．

- 難易度の調節を支援する機能の実装

「あるレベル以下の手筋のみを利用して解ける範囲の図示」を行う機能は，本ツール内で問題を解くための基本手段である SAT ソルバを用いずに以下のように実現している．各セルに対して 1 から 9 の数字が候補かどうかを示すフラグを用意し，次に述べる手筋の条件に従って候補を絞るという手法をとっている．

– 手筋と難易度

問題を解く際は，解法手筋を繰り返し利用することにより空白セルに数字を入れていく．問題の難易度は，その問題を解くためにどのレベルの手筋を用いる必要があるかで分類する．まず，それぞれの手筋をレベル分けする．ここでは 3 通りに分類する．同じ列に空白が一ヶ所・マーキングによる解法を初級手筋とし，中級・上級手筋も [6] に従って同様に取り決める．

初級手筋のみで問題を解くことができればその問題は最も簡単と取り決め，レベル 1 とする．初級手筋のみで問題を解くことができず，中級手筋を用いる必要があれば，問題はより難しくなりレベル 2 とする．このように使用する手筋に応じて段階的に定義し，解くのに上級手筋が必要であれば更に難しくレベル 3 とする．ここまでの手筋では解けず，以下で定義する仮定法を用いる必要があればその問題はレベル 4 とする．仮定法を使用しても解けないような問題はレベル 5 とする．このレベルの数値は高い方が難解になる．レベル 5 の問題は，多くの人にはほとんど解けないほど難易度が高いと思われる．

– 仮定法

仮定法では，まず特定のセルに着目する．そのセルの縦・横・ブロックの情報などからそこに入りうる数字の候補を列挙し，その数字の候補が複数ある場合，そのうちの一つを正しいと仮定する．そして仮定法以外の他の手筋を用いて数字を入れる作業を継続し，矛盾が生じたら初めに仮定した候補が誤りであった事が判明する．その場合は初めの仮定した数字を候補をから削除することができる．このようにある数字を仮定して矛盾を導くことにより候補を絞り込んでいき，残りの候補が一つだけになればその数字がこのセルに入ることが決定される．

なお，数字を仮定した後に矛盾を生じることなく途中まで進んで再度行き詰った場合は当初の仮定が正しかったのか誤りであったのか区別することができない．この場合は別の数字を仮

NII-Electronic Library Service

定するか，別のセルで仮定法を用いることになる．

4.5 問題作成の流れ

本ツールでは初期状態として，空白の盤面と盤面の矛盾・一意性を示す情報が表示される．問題作成は，この初期状態から以下の作業を対話的に行っていき，作成者（ユーザー）が意図する問題が得られたと判断した時点で終了する．終了の際には，盤面を表現したテキストファイルに結果を保存できる．

問題作成は以下の手順で行う．

（1） セルを選択する．この際，セルに矛盾なく入りうる数字の候補が表示される．

（2） 選択したセルに入れたい数字を選択，もしくはすでに入っている数字を削除する．この際，盤面の矛盾・一意性を示す情報が更新される．

なお，いずれの時点でもメニューからそれぞれの機能を選択することにより，以下の情報を知ることができる．

- 盤面の表出数字を含有する解盤面を求める (盤面が矛盾している場合を除く)
- 手筋のみで解けるセル範囲の図示 (盤面が矛盾している場合を除く)
- 難易度の判定 (盤面が一意の時のみ)
- 数字を削除しても一意性を保つセルの検出 (盤面が一意の時のみ)

5. 問題の作成実験

本ツールを使うことにより，対話型による問題作成がどの程度の時間で作成可能かを実験した．

- 作成条件

以下の条件を満たす問題を著者が作成するのに必要になった時間の測定を行った．

- 問題の模様は，縦・横方向共に対象形または回転対象形
- 削除しても対象性と一意性を保つ表出数字は未出現
- 表出数字は 28 個以下
- 難易度は初級手筋のみで解くことが可能

上記の条件を満たすかどうかは，情報表示の機能を使い適時行った．

- 結果

実験の結果を表 1 に示す．ここで 10 分経過した時点でここで良問とする問題が作成できない場合には作成を中止した．なお，図 1 の問題はこの実験で作成したものの一つである．

作成時間	問題数	割合
～3 分	11	58%
4～5 分	4	21%
6～10 分	0	0%
11 分以上 (中止)	4	21%

表 1 実験結果

- 結果分析

測定結果によると，対話型による問題作成にも関わらず本稿で高品質と考える問題が 5 分以内で作成できている．今回作成した問題の条件は画一的な評価のため簡略化されているため，更に高品質な問題を作成する場合はより時間が必要と思われる．

- システムの反応時間

一般的な PC におけるシステムの一操作あたりの反応時間は以下のとおりである．数独問題を表す論理式を SAT ソルバで解く場合の実行時間は 1 回あたり約 0.01～0.05 秒であり，主に SAT ソルバの呼び出し回数に応じて反応時間が変化する．

数字の挿入のためにセルを選択したときは SAT ソルバは最大 9 回実行され，その候補表示に掛かる合計実行時間は 0.1～0.3 秒程度である．数字を挿入したときの一意性の検出は SAT ソルバの実行は一度であり，所要時間は 0.1 秒以下となっている．「数字を削除しても一意性を保つセルの検出」機能は表出数字の個数に応じて SAT ソルバが呼ばれる回数が変わってくるが，多くの場合表出数字は 20～30 個の時にこの機能を実行すると思われる．すなわち 20～30 回程度 SAT ソルバを呼んでおり，反応時間は 1～2 秒程度である．「あるレベル以下の手筋のみを利用して解ける範囲の図示」機能では 0.01 秒以下で結果が返される．

6. 他の手法との比較

本ツール以外の問題作成ツールには大別して 2 種類ある．いくつかの生成条件を指定すると問題を自動で生成するプログラム [8]～[10] と，対話型で数字を埋めていく際にそれを支援するようなプログラム [11], [12] である．これらのツールの多くは内部の処理方法を公表していないため，以下では処理方法が明らかになっているものとの比較を行う．

6.1 自動作成ツールとの比較

自動生成ツールは予めいくつかの生成条件を指定して，プログラム内部で問題生成の演算を行うことにより指定した条件に合致している問題を探し出すという仕組みになっているものが多い．Sudoku Generator [8] では表出数字の数・対称性の有無・最大難易度等を指定してその条件に合う問題を生成する．最初に解局面を生成し，そこから表出数字を 1 個ずつ削除してその都度一意性が保たれるかチェックを行う．それを一意性を維持する限界まで任意の場所の表出数字をセルから削除していく．これ以上削除できなくなると，一段階元に戻して別の場所の削除を試みるという方式でできるだけ表出数字が少ない問題を探索する．また Puzzle Generator [10] では，表出数字を入れたいセルの場所を，条件として指定する．解盤面を次々に生成 [13] し，そこから指定したセル以外を空白セルに置換して得られる盤面が一意になっているものを探すことによって問題を作成する．この生成手法は，生成条件を厳しくすると計算時間が急激に増えるという欠点がある．本ツールにおいても同様に表出文字の少ない問題を作成することは非常に難しいが，問題の模様については作成途中で作成者が少しずつ模様を変化させることが可能である．

6.2 他の対話型による作成ツールとの比較

ほとんどの対話型ツールでは本ツールと同様に，作成を開始すると全て空白セルの盤面が用意され，表出数字を埋めたいセルを選択して 1 から 9 の数字を入力し，その選択と数字の入力の繰り返しで問題を作成していく．多くのツールは，数字を

NII-Electronic Library Service

						1		
			1					
		2						
	1							
1								

図 3 矛盾の例

入力する際に縦・横・ブロックについて数字の重複が無いかをチェックする [11], [12]. 数字を入力する段階でどこにどの数字が入る可能性があるかが表示されており，その事を利用してこの場合解の一意性のチェック結果も表示されている．しかしながら，矛盾や一意性の検出を正しく行っていない対話型ツールが多い．その原因は数字を入力する際に，縦・横・ブロックについて同じ表出数字は無いことのチェックのみにたよっている事に起因する．例えば図 3 のような場面では縦・横・ブロックのいずれにも同じ数字はない．しかし左上のブロックには，9 つのセルのうちどのセルにも 1 を入れることもできない．この例のように，同じ数字が重複していないことだけでは問題が矛盾していないことのチェックはできない．

本設計ツールでは対話型の作成手法を採用したが，SAT ソルバを利用する事により矛盾と一意性の検出を正しく行っている．作成途中にどの数字が入りえるかを示したり，表出数字を削減する機能や難易度の検出を設けることにより問題のデザインのしやすさも他の対話型ツールに比べて向上がみられる．

7. おわりに

本稿では，数独の問題を対話型による方法によって作成する手法について提示した．ここでは問題の模様・表出数字の個数の少なさ・難易度の適切さの 3 つを品質が高いと考えた．問題作成ツールにおいて矛盾と一意性を正しく検出するためには数独の問題を論理式に変換して SAT ソルバを利用する方法が有用であった．本ツールにより，ここで品質が高いと想定した問題を比較的短時間で作成することが可能であった．

今後より良いツールを作成するための改善案として，次のような方法が考えられる．問題作成者が想定する問題の模様について，セル内の数字は指定せずに表出数字として埋めたいセルの場所だけを指定する．そしてそれを問題として満たすような数字の入れ方はあるかを検索する．更にそのような数字の入れ方が複数存在する場合は，希望する難易度の問題を出力する．この手法により問題のデザインは非常に簡素化することが可能であるが，その存在するかどうかの計算は時間が掛かると思われるためそれをどう解決するかが課題である．

今回実装しなかったが，以下のような機能があったほうが良いと思われる．

- 1 から 9 の数字はそれぞれをどのように入れ替えても本質的に全く同じ問題であることから，数字の入れ替え機能があれば容易に連続した数字の並びを実現できる．
- 問題図全体を対称移動や回転させる機能により，対称移動させたい場合に必要な手間を削減する．
- 問題作成初期において表出数字の数が少ないうちは特定のセルにどの数字が入るかは大きな問題ではないことから，表出数字としたい場所をある程度指定してその指定した場所に問題が矛盾しないような任意の数字を自動的に入れていく．こうすれば初期段階においては具体的な数字を指定する必要がなくなるため，ある程度問題の作成に掛かる時間を短縮できると思われる．

謝辞　本研究は一部，科研費#18500011, #17700009, #16650005 の補助を受けている．

文　献

[1] Gihwon Kwon, Himanshu Jain. Optimized CNF Encoding for Sudoku Puzzles. In 13th International Conference on Logic for Programming Artificial Intelligence and Reasoning LPAR, 2006.
[2] Ines Lynce, Joel Ouaknine. Sudoku as a SAT problem. In The Proceedings of AIMATH'06, 2006.
[3] Tjark Webber. A SAT-based Sudoku Solver. In The Proceedings of LPAR'05, 2005.
[4] Helmut Simonis. Sudoku as a Constraint Problem. In CP Workshop on Modeling and Reformulating Constraint Satisfaction Problems, 2005.
[5] Gordon Royle. Minimum Sudoku. http://people.csse.uwa.edu.au/gordon/sudokumin.php
[6] ニコリ編. ニコリ「数独」名品 100 選. 文藝春秋, 2006.
[7] ニコリ公式パズルガイド「数独」. http://www.nikoli.co.jp/ja/puzzles/sudoku/
[8] Mogens Kjaer. Sudoku Generator. http://www.lemo.dk/sudoku/
[9] Thomas Hinkle. GNOME Sudoku. http://gnome-sudoku.sourceforge.net/
[10] 稲葉直貴, 藤原博文. Puzzle Generator. http://puzzle.gr.jp/
[11] Angus Johnson. Simple Sudoku. http://www.angusj.com/sudoku/
[12] Mark Huckvale. Su Doku Puzzle Creator. http://www.phon.ucl.ac.uk/home/mark/sudoku/
[13] 戸神星也. 数独の解生成と解に対する番号付け. http://doorgod.org/sudoku/
[14] Niklas Een, Niklas Soerensson. MiniSat. http://www.cs.chalmers.se/Cs/Research /FormalMethods/MiniSat/

NII-Electronic Library Service