

# 数独の解法

2014年5月1日  
改訂：2020年9月7日  
改訂：2021年1月7日  
改訂：2021年7月5日  
改訂：2022年8月29日

潮 哲也  
(連絡先 : [tetsuya@ushio.org](mailto:tetsuya@ushio.org))

## 【I. はじめに】

このドキュメントでは、筆者が案出した数独の解法を説明する。数独のルールは、既知とする。

この解法は、既に javascript で実装されており、実装したプログラムは、画面描画に関するプログラムその他関連するプログラムと併せ、  
<https://www.ushio.org/Sudoku/>  
で実際に稼働している。

## 【II. 用語】

まず、用語を定義する。

- (1) 基本用語 (盤面、マス、数字、row、col、box) は、以下のとおりとする。
- ① 数独の問題や解答を示す  $9 \times 9$  の表を「盤面」といい、 $9 \times 9$  の 81 個の場所の一つ一つを「マス (square)」、マスに表示する文字を「数字 (number)」という。
  - ② 横 1 列の 9 個のマスを「row」、縦 1 列の 9 個のマスを「col」、盤面に 9 個ある  $3 \times 3$  のマスの集合 (同じ数字が入ることが禁じられているもの) を「box」という。
  - ③ マスは、row 方向、col 方向にそれぞれ 0 から 8 の順番 (「マスの順番」) を付けることで識別され、この二つの方向のマスの順番の組を「マス番号」という。
  - ④ 数字は、マスに表示される 1 から 9 の文字の順序を表す 0 から 8 の数値で識別され、この数値を「数字順」という。例えば、「1」という文字を表

す数字順は 0 である。

なお、box、square、row 及び column の用語は、末尾の参考に掲げた文献の用語に倣ったものである。

(2) 「確定セット (determined set)」とは、 $n$  個のマスと  $n$  個の数字の集合であり、それら  $n$  個のマスにそれら  $n$  個の数字が入ることが確定したものである。確定セットには、row ベース、col ベース、box ベースの区別はない。ここで、個数  $n$  をその確定セットの「order」という。order=1 の確定セットは、あるマスに入る数字が一意に決まったことを表す。このような (order=1 の) 確定セットを「単一確定セット」という。

プログラム中では、個々の確定セットを表す変数の型名 (javascript でいうプロトタイプの名称) は DSet であり、それまでに発見されている確定セット全体を保管するリストの変数名は dSetList である。

(3) 「排他セット (exclusive set)」とは、排他関係の調査結果であり、確定セットの元情報となるものである。排他セットには、row ベース、col ベース、box ベースの区別がある。

プログラム中では、個々の排他セットを表す変数の型名 (javascript でいうプロトタイプの名称) は ESet である。排他セットは、一時的に使われるものなので、全体を保管するリストはない。

(4) 「表示可能性 (possibility)」とは、あるマスと数字の組合せに対して与えられる性質であり、その数字がそのマスに入る可能性が現時点で残っているかどうかを表すものである。可能性が残っていれば true、残っていなければ false で表す。ある表示可能性の値を true から false にすることを、「可能性を除外 (exclude) する」という。

「可能性リスト (possibility list)」とは、 $9 \times 9$  のマスと 9 種類の数字のそれぞれについて、表示可能性を格納した  $9 \times 9 \times 9$  の要素を持つ 3 次元配列である。配列添字は、マス番号 (row, col) 及び数字順である。

プログラム中では、可能性リストを表す 3 次元配列の変数名は pList である。List と称しているが、プログラム中でのデータ構造は配列である。

(5) 「不定 (indeterminate)」とは、盤面の指定が不十分であり、解答が一意に定まらないことをいう。

(6) 「不能 (inconsistent)」とは、盤面の指定に矛盾があり、解答があり得な

いことをいう。

### 【Ⅲ. 解法の概要】

次に、この解法の概要を説明する。

この解法では、以下の「(1) 可能性サーチ」と「(2) 確定セット登録」を繰り返すことで、単一確定セットを一つ発見して、登録する。つまり、マスに入る数字を一つ決める。

そして、それを繰り返すことで、盤面を完成させる。多くの場合は、これによって解答が求まる。

この二つの繰り返しがそれ以上できなくなった場合は、以下の「(3) 仮置きによる可能性サーチ」を試みる。

#### (1) 可能性サーチ

可能性サーチでは、可能性リストを順にサーチして、true のまま残っている表示可能性を一つ一つ順に取り上げる。取り上げた表示可能性に対して、それまでに発見されている全ての確定セットに照らして検討することで状態判定を行い、あり得ないと判定できればそれを除外 (exclude) する。

ここで、あり得ないと判定して表示可能性を除外する条件は、一例を挙げると、次の条件である。

- (a) まず、数独のルールにより、一つの row に同じ数字が 2 度入ることはないから、次の①の条件は明らかである。
- ① 取り上げた表示可能性のマスと数字を見て、その表示可能性のマスと同一 row のマス (ただし、同一のマスを除く) のいずれかにその表示可能性の数字と同じ数字が確定していれば、その表示可能性は除外できる。
- (b) 上の①で、数字が確定しているということは、order=1 の確定セット (単一確定セット) があることに他ならない。そこで、これを order が 2 以上の場合を含めた確定セットに一般化すると、次の②の条件になる。
- ② 取り上げた表示可能性のマスと数字を見て、その表示可能性のマスと同一 row のマス (ただし、同一のマスを除く) だけを持つ確定セットが

あり、その確定セットがその表示可能性の数字と同じ数字を含んでいれば、その表示可能性は除外できる。

(c) そこで、この解法では、②の条件に基づいて、可能性を除外する。②は①の一般化であるから、これにより、①の条件のものも共に除外されることになる。

## (2) 確定セット登録

表示可能性が除外されたら、その都度、除外された表示可能性のマス及び数字並びにその近隣のマスに残っている表示可能性に注目し、新規に登録できる確定セットの候補（排他セット）がないか、排他関係を調査する。

ここで、排他セットとなるのは、一例を挙げると、次の場合である。

① ある表示可能性が除外されたとき、その表示可能性のマスに残っている数字が $n$ 個であるとすれば、それらの $n$ 個の数字だけを表示可能性として残しているマスが同一 row にちょうど $n$ 個見つければ、それらが確定セットの候補（排他セット）となる。

排他セットが見つかったら、それをそれまでに登録されている全ての確定セットと比較し、例えば同一内容のものが既に登録されていれば除外する等、必要な調整・確認の処理を行う。その結果、登録可能な確定セットを見い出した場合には、それを新しい確定セットとして登録する。

## (3) 仮置きによる可能性サーチ

上の(1)(2)を繰り返すことで、多くの場合は解答が求まる。しかし、難易度の高い問題では、(1)(2)で除外できる表示可能性がなくなってしまう、それ以上進めなくなる場合がある。その場合は、「仮置き」(Assumption)を試みる。

仮置きに際しては、残る表示可能性が二つであるマスから始めて対象となるマスを順に取り上げ、残っている数字を一つ選んで、その表示可能性を仮定した場合（「正」という。）と可能性除外した場合（「副」という。）の二つの場合について、その仮置きが成功するかどうかを調べる。ここで、「仮置きが成功する」とは、仮置きに基づいてこの解法をさらに進めてみて、最終的に

盤面が完成する（後述する《完了》ステータスになる）ことをいう。

#### 【IV. 解法の詳細】

次に、この解法を、javascript で実装した実際のプログラムに即して詳細に説明する。

以下の説明の中の「1」、「1. 1」等の項番は、実際のプログラム中のコメントと一致している。

この解法のプログラムは、以下の「1. 可能性サーチの実行とその結果に基づく状態判定」と「2. 排他関係の調査と確定セットの新規登録」の二つのコンポーネントから構成される。これら二つのコンポーネントには、それぞれ二つずつ（合計四つ）の「代表 function」があり、さらにその下にサブ function がある。四つの代表 function の中で最も複雑なのは、「1. 1 状態判定 startSearch」であり、この代表 function の下のサブ function のうちの三つが、特に「主要サブ function」と指定されている。三つの主要サブ function は、いずれも代表 function 「1. 1 状態判定 startSearch」の内部サブ function であるから、他の代表 function 及びそのサブ function から呼び出すことはできない。

なお、プログラム構造については、別紙の「数独解法プログラムのプログラム構造」に図示している。図中、太枠で示した四つが「代表 function」である。また、「SSB」、「SSL2」、「TS」という略称が示された三つが「主要サブ function」である。

### 1. コンポーネント：可能性サーチの実行とその結果に基づく状態判定

#### 1. 0 このコンポーネントの概要

このコンポーネント（「1. 可能性サーチの実行とその結果に基づく状態判定」）は、「1. 1 状態判定 (startSearch)」と「1. 2 可能性サーチ (searchPossibility)」の二つの代表 function 及びそれらのサブ function から構成される。

このコンポーネントは、ボタンイベントのハンドラから呼ばれることにより

開始する。

このコンポーネントは、【解法の概要】の「(1) 可能性サーチ」を行い、続いて「2. 排他関係の調査と確定セットの新規登録」のコンポーネントを呼び出して【解法の概要】の「(2) 確定セット登録」を行わせる。そして、この動作を繰り返すことによって、1回呼ばれる毎に単一確定セットを一つ発見して登録する。つまり、マスに入る数字を一つ決める。それ以上可能性が除外できなくなった場合は、【解法の概要】の「(3) 仮置きによる可能性サーチ」を試みる。

ボタンイベントのハンドラは、盤面に新たに一つだけ数字を決めようとする場合 (1 Step ボタンが押された場合) にあっては、一度だけこのコンポーネントを呼ぶ。盤面に最終的な解答を求めようとする場合 (Go ボタンが押された場合) にあっては、次に述べる《完了》ステータスが true になるまで繰り返してこのコンポーネントを呼ぶ。

このコンポーネントが呼ばれるときには、引数は指定されないが、呼び元と呼び先で共通に使用する共通変数があり、これによって情報を授受している。

まず、ボタンイベントのハンドラがこのコンポーネントを呼ぶときは、準備処理として、盤面で指定されている全ての数字に対応してそれぞれ単一確定セットを登録し、可能性リストを全てクリア (true) する。また、盤面上のマスを保持する  $9 \times 9$  の 2 次元配列を用意する (実際の変数は、画面描画に関するプログラムで定義されている。)。この配列は、呼出し先の function が《不能》ステータス等のエラーを検知したときにエラー箇所を表示するために使用する。

一方、このコンポーネントの実行が終了して呼び元に戻るときは、確定セットの登録と併せて、コンポーネントの処理結果としての以下のステータスが共通変数にセットされている。

- ① 《完了》ステータス： このコンポーネントの処理結果として 81 個の数字が全て確定していれば true、そうでなければ false。
- ② 《単一確定セット登録》ステータス： サブルーチンとして呼び出す「2. 排他関係の調査と確定セットの新規登録」のコンポーネントにより新たに単一確定セットが登録された場合 (つまり、あるマスに数字が決まった場合) は true、そうでない場合は false。
- ③ 《不定》ステータス： このコンポーネントの調査の過程で不定を検出した場合は true、そうでない場合は false。
- ④ 《不能》ステータス： サブルーチンとして呼び出す「2. 排他関係の調査と確定セットの新規登録」の調査の過程で不能を検出した場合は true、そうでない場合は false。

- ⑤ 《単一確定セット重複》ステータス： 《単一確定セット登録》ステータスの場合において、仮置きにより複数の解が見つかった場合は true、そうでない場合は false。
- ⑥ 《単一確定セット不定》ステータス： 《単一確定セット登録》ステータスの場合において、仮置きにより一方で見つかり、他方で《不定》ステータスが見つかった場合は true、そうでない場合は false。

なお、以上の 6 個のステータスと《確定セット登録》ステータスとを併せて「7 ステータス」という。ただし、《確定セット登録》ステータスは、このコンポーネント内で参照するものであり、呼び出し元に戻るときの値は意味を持たない。

また、このコンポーネントでは、《完了》《単一確定セット登録》《不能》の少なくともいずれかが一つがセットされた場合、サブルーチンとして呼び出す「2. 排他関係の調査と確定セットの新規登録」において、現在のマス番号及び数字順が共通変数の「currentRow」、「currentCol」、「currentNum」にセットされている。この 3 個の値を合せて「3 変数」という。

## 1. 1 代表 function：状態判定 (プログラム中での function 名： startSearch)

### (1) 代表 function の概要

この代表 function (StartSearch) は、このコンポーネント（「1. 可能性サーチの実行とその結果に基づく状態判定」）の入り口であり、ポタニイベントのハンドラから呼び出される。StartSearch は、主要サブ function である startSearch\_body (略称 SSB) を呼び出して、殆どの仕事は、SSB が行う。代表 function である StartSearch 自体の仕事は、SSB のために必要なコンテキスト（これを「リカーシブコンテキスト」という。）を整えることである。

実行中、仮置きに伴って後述の trySeach (略称 TS) から再帰呼び出しが行われる場合もある。この場合は、TS がリカーシブコンテキストを整え、StartSearch を経由せずに SSB を直接呼び出す。

ここでリカーシブコンテキストの具体的内容は、盤面、可能性リスト (pList)、確定セットリスト (dSetList)、3 変数及び 7 ステータスの合計 13

変数であり、変数名としては、recursiveParm 又は rP 等が使われている。このうち「盤面」とは、盤面上のマスを保持する 9×9 の 2 次元配列であり、呼出し先の function が《不能》ステータス等のエラーを検知したときにエラー箇所を表示するために使用する。ただし、再帰呼出しの場合に生じる《不能》ステータス等のエラーは、仮置きが的中しなかったことで生じるものであって本来のエラーではないから、再帰呼出しの場合の「盤面」は、呼出し元で参照することはなく、使い捨てにされる。再帰呼出しでない場合の「盤面」は、画面描画に関するプログラムで定義されている

なお、これらの 13 変数は、SSB が呼ばれる場合にはリカーシブコンテキストとして引数でやり取りされるが、StartSearch が呼ばれる場合には引数でなく、「1. 0 このコンポーネントの概要」で述べた共通変数としてやり取りされる。

盤面に最終的な解答を求めようとする場合 (Go ボタンが押された場合) においては、StartSearch は繰り返し呼び出され、呼出しの都度、数字を一つずつ決めて呼び元に返す。ただし、以前の StartSearch 呼出しにおいて仮置きが行われ、《完了》ステータスに達しているときは、既に最終的な答えが求まっているから、その後の処理を繰り返して行う必要はない。そのため、最初に《完了》ステータスに達した時点での確定セットリストを finishedDSetList という変数にセットしておき、以後、StartSearch は、SSB を呼び出すことなく、finishedDSetList から単一確定セットを一つずつ取り出して呼び元に返す。

この代表 function (StartSearch) は、四つの代表 function の中で最も複雑であって、その下には次の三つの主要サブ function があり、それぞれ略称が与えられている。

- (a) startSearch\_body (状態判定本体) 略称 SSB
- (b) startSearch\_level2 (仮置き) 略称 SSL2
- (c) trySearch (再帰試行) 略称 TS

## (2) SSB の処理内容

代表 function である StartSearch は、その主要サブ function の startSearch\_body (略称 SSB) を呼び出して、殆どの仕事は、SSB が行う。そこで、以下、SSB の処理内容を説明する。



SSB は、「1. 2 可能性サーチ」の代表 function を呼び出すことにより可能性サーチを行うほか、サブ function を使って仮置きによるサーチも行う。そして、単一確定セットを一つ見つけて呼び元に返す。

SSB は、以下の (a) (b) (c) の三つの処理を必要に応じて順に行うことにより、単一確定セットを探す。

(a) 仮置きによらない可能性サーチ (プログラム中のコメントの① - 1)

「1. 2 可能性サーチ」の function を呼び出すことにより可能性サーチを行う。「1. 2 可能性サーチ」からは、表示可能性が除外されたかどうかリターン値で返されるとともに、検出したステータスが該当する 7 ステータスに設定されて返される。

表示可能性が除外されていれば、新たにあるマスに数字が決まったかどうか (つまり《単一確定セット登録》ステータスかどうか) をチェックし、そうであれば、この function (SSB) を終了する。そうでなければ、再度「1. 2 可能性サーチ」の function を呼び出して可能性サーチを更に進める。この処理を単一確定セットが登録できるまで繰り返す (エラーケースを除く)。

表示可能性が除外されていなければ、仮置きによらない可能性サーチはそれ以上進められないから、次の (b) に進む。

(b) 再帰レベル制限超過の処理 (プログラム中のコメントの① - 2)

仮置きによる再帰呼出しのレベルを確認する。80 を超えていれば、それ以上の再帰呼出しを行わず、《不定》として呼び元にリターンする。ただし、そのような事態は想定はされない。

(c) 仮置きによる可能性サーチ (プログラム中のコメントの① - 3)

残る表示可能性が二つであるマスから始めて対象となるマスを順に取り上げ、残っている数字を一つ選んで、その表示可能性を仮定した場合 (「正」という。) と可能性除外した場合 (「副」という。) の二つの場合について、その仮置きが成功するかどうかを調べる。成功するかどうかは、それぞれの表示可能性を仮に置いた後、状態判定 (SSB) を再帰呼出しすることで調べる。再帰呼出しから返された結果が《単一確定セット登録》であった場

合は、更に再帰呼出しと返された結果の確認を繰り返すことでステップを進め、《完了》ステータスにまで到達するかどうかを確認する。《完了》ステータスにまでステップが進められれば、仮置きは成功である。なお、一つのマスについて数字は一つしか選ばないが、表示可能性を仮定した場合と可能性除外した場合について調べるから、これで場合を尽くしている。

仮置きが成功したら、その内容を確定セットリストに登録し、SSB の処理を終了して呼び元に戻る。仮置きを調べる過程で一旦《完了》ステータスにまでステップを進めているが、SSB の呼び元に戻すのは、最初（今回の再帰で最初）に仮置きした一つの単一確定セットだけである。

仮置きによる可能性サーチの実際の処理は、「startSearch\_level2（略称 SSL2）」と「trySearch（略称 TS）」という二つの主要サブ function を使い、以下のような役割分担で行う。なお、TS は、SSL2 の内部サブ function である。

- ① SSL2 は、残る表示可能性が二つであるマスから始めて対象となるマスを順に取り上げ、残っている数字を一つ選んで、それぞれの正副のケースのそれぞれについて、TS を呼び出す。その際、再帰に備えて正副のコンテキスト変数を準備しておく。
- ② TS は、取り上げられたケースについて、リカーシブコンテキストを準備する。その際、「正」であれば数字を仮に確定させることになるから makeDeterminedSet\_body を呼出し、「副」であれば可能性を仮に除外することになるから makeExclusiveSet を呼び出す。

その後、状態判定（SSB）を繰り返し呼出し、最後まで（《完了》ステータスになるまで）ステップを進められるかどうかを確認する。なお、最初（今回のボタン押下後に最初）に《完了》ステータスの状態になったときは、そのときの確定セットリストの内容を finishedDSetList に退避しておく。これは、次の StartSearch 呼出しにおいて再度サーチを行う処理負荷を避けるためである。なお、この finishedDSetList も、共通変数であるが、呼び元のボタンイベントのハンドラでは、値を初期化はするが参照はしない。代表 function の startSearch からリターンした後でも値を保持する必要があるため、startSearch の内部変数でなく共通変数としている。

- ③ SSL2 は、正副の場合のそれぞれの仮置き結果を TS から受け取り、「(3) 正副の仮置き結果に基づく結果判断と処理内容」に従って結果判断とそれに伴う処理を行う。これにより、いずれかが《完了》又は両方が《不

能》の場合は仮置きを終了し、それ以外の場合は次のマスに進むことになる。

なお、「正」の仮置きで《単一確定セット重複》が返された場合は、「副」の仮置きを行うことなく、「正」を答とする。この場合、他に答があることは、「正」の再帰呼出し先で表1の①により既に警告されている。このように「副」の仮置きを省略することは、計算量を削減するために重要である。ただし、省略を行うのは、《単一確定セット重複》の場合だけであって、単に《完了》になっただけでは省略を行わない。というのは、《完了》になっただけでは、一つの解は見つかっているが、それが唯一であるかどうか分かっていないからである。

SSBは、以上の(a)(b)(c)により単一確定セットが見つからなかった場合、すなわち、「(a) 仮置きによらない可能性サーチ」によって可能性サーチがそれ以上進められなくなり、かつ、「(c) 仮置きによる可能性サーチ」によって仮置きが成功しない(《完了》ステータスにならない)場合は、《不定》ステータスをセットし、functionを終了する。

また、SSBは、《単一確定セット》ステータスがセットされてfunctionを終了する場合は、81個のマスが全て確定しているかどうかを判定し、そうであれば、《完了》ステータスをセットする。この処理は、プログラムでは①-1の中に記載されている。なぜなら、仮置きのために①-3が実行された場合であっても、《完了》ステータスは、最終段の再帰呼出しでの①-1で検知されるからである。最終段の再帰呼出し以外での①-3では、《完了》ステータスは、正副の仮置き結果に基づいて、以下の(3)に従って設定される。

### (3) 正副の仮置き結果に基づく結果判断と処理内容

上の(2)(c)の③の中で、SSL2が正副の場合のそれぞれの仮置き結果をTSから受け取った際に行う結果判断とそれに伴う処理内容は、複雑であるから、整理して以下の表1に示す。

表1 正副の仮置き結果に基づく結果判断と処理内容

		正の結果		
		《完了》	《不能》	《不定》
副	《完了》	① 《完了》とし、	④ 《完了》とし、	⑦ 《完了》とし、

の 結 果		正を答とし、他に答 があることを警告 する。	副を答とする。	副を答とし、他に答 があり得ることを 警告する。
	《不能》	② 《完了》とし、 正を答とする。	⑤ 《不能》とする。	⑧ 《不定》とし、 仮置き可能な次の マスに進む。
	《不定》	③ 《完了》とし、 正を答とし、他に答 があり得ることを 警告する。	⑥ 《不定》とし、 仮置き可能な次の マスに進む。	⑨ 《不定》とし、 仮置き可能な次の マスに進む。

表 1 の中で、①②③④⑦の場合に答が求まり、《完了》ステータスとなる。このうち、①③⑦の場合には、警告を出力している。これは、これらの場合には表示される解答が唯一のものではない可能性があるためである。また、②④の場合には、唯一の解答が求まっているが、どのような仮置きを行ったかを知るため、画面上の無名のチェックボックスにチェックを入れることで参考メッセージを出力させることができる。

## 1. 2 代表 function : 可能性サーチ

(プログラム中での function 名 : searchPossibility (略称《SP》))

この代表 function (searchPossibility (略称《SP》) ; なお、略称は、「1. 1 startSearch (状態判定)」の三つの主要サブ function の略称と異なり、《》で囲んで《SP》としている。) は、確定セットを一つ見つけて呼び元に返す。それが単一確定セットでなかった場合には、呼び元から再度呼ばれることで、可能性サーチを更に進める。

《SP》は、 $9 \times 9 \times 9$  の可能性リストを先頭から順にサーチし、true のまま残っている表示可能性について、dSetList 内の各確定セットと順に比較し、下の (1) から (4) のいずれかに該当すれば、そのマス及び数字の表示可能性を除外 (exclude) する。この処理を、何らかの確定セット (単一かどうかにかかわらず) を登録できるまで続ける (エラーケースを除く。)。下の (1) から (4) のいずれかに該当するかどうかのチェック処理は、プログラム中、コメントの「リストの中の確定セットごとの処理①」で示されている。

- (1) その確定セットがそのマスを含み、その数字を含まないことで、そのマスが排除 (eliminate square) される。
- (2) その確定セットがその数字を含み、そのマスと同じ row のマスだけを含み、かつ、そのマスを含まないことで、その数字が排除 (eliminate number) される。
- (3) その確定セットがその数字を含み、そのマスと同じ col のマスだけを含み、かつ、そのマスを含まないことで、その数字が排除される。
- (4) その確定セットがその数字を含み、そのマスと同じ box のマスだけを含み、かつ、そのマスを含まないことで、その数字が排除される。

ある表示可能性が除外された場合、その都度、そのマス番号と数字順を引数として「2. 排他関係の調査と確定セットの新規登録」の function を呼び出す。呼び出された「2」では、排他関係を調査し、登録できる確定セットがあれば、それを登録する。

ただし、例外として、次の二つのケースにおいては、「2. 排他関係の調査と確定セットの新規登録」の function を呼び出しても《不能》が検知されることになるから、これを呼び出すことなく、《SP》が自ら《不能》ステータスをセットし、盤面上のマスのうち《不能》ステータスに関連するマスをマークしてリターンする。ここで、「盤面」とは、画面描画に関するプログラムで定義されている変数であり、本プログラムと共有している。変数名は cr である。なお、このようなエラー検知は、「2. 排他関係の調査と確定セットの新規登録」の function に任せることもできるが、《SP》が自ら行った方がエラー箇所のマスのマークが容易である。

- (1) 既に確定している数字について表示可能性を除外しようとしているケース

このケースの処理は、プログラム中、コメントの「リストの中の確定セットごとの処理②」で示されている。

- (2) 可能性除外の結果として row, col 又は box 内にその数字の可能性がなくなるケース

このケースの処理は、プログラム中、コメントの「リストの中の確定セットごとの処理③」で示されている。

そして、「2. 排他関係の調査と確定セットの新規登録」の function を呼び出した結果、確定セットが登録されるか、《不能》ステータスになれば、この function (《SP》) を終了し、そうでなければ、次の表示可能性 (マス及び数字)

にサーチを進める。

なお、単一確定セットが登録されたとき（つまり、あるマスに数字が決まったとき）は、そのマス及び数字の表示可能性の除外に伴う確定セットのうちでまだ登録されていないものがある可能性があるから、そのマス及び数字の表示可能性を true に戻す。

この function (《SP》) は、可能性が除外された場合に戻り値に true を、そうでない場合に戻り値に false を返す。また、この function からのリターン時は、《単一確定セット登録》ステータスであれば、この function から呼ばれる「2. 排他関係の調査と確定セットの新規登録」の中で、確定したマス番号及び数字順が 3 変数 (「currentRow」、「currentCol」、「currentNum」) にセットされている。「リストの中の確定セットごとの処理②」又は「リストの中の確定セットごとの処理③」で《不能》ステータスをセットしたときには、《SP》自身が 3 変数をセットしている。

## 2. コンポーネント：排他関係の調査と確定セットの新規登録

### 2. 0 このコンポーネントの概要

このコンポーネント (「2. 排他関係の調査と確定セットの新規登録」) は、「2. 1 排他セットの検索 (makeExclusiveSet)」と「2. 2 確定セットの登録 (makeDeterminedSet)」の二つの代表 function 及びそれらのサブ function から構成される。

このコンポーネントは、あるマス及び数字について表示可能性が除外されたとき（つまり、対応する表示可能性が false に変更されたとき）、その都度、そのマス番号と数字順を引数として「1. 可能性サーチの実行とその結果に基づく状態判定」のコンポーネントから呼ばれることにより開始する。

このコンポーネントは、【解法の概要】の「(2) 確定セット登録」を行う。つまり、そのマス及び数字について新たに登録できる確定セットをサーチし、それが見つかれば全て登録する。ただし、単一確定セットが登録できた場合（つまり、あるマスに数字が決まった場合）は、そのマス番号及び数字順を 3 変数 (「currentRow」、「currentCol」、「currentNum」) にセットして、以後の登録は行

わずに、このコンポーネントを終了する。単一確定セットを登録したら、いかなる場合も、1回の呼出しでそれ以上の確定セットを登録することはしない。したがって、このコンポーネントを1回呼出すことで複数個の単一確定セットが登録されることはない。

このコンポーネントは、確定セットの登録と併せて、処理の結果としての以下のステータスをセットして呼出し元で参照できるようにする。

- ① 《確定セット登録》ステータス： このコンポーネントの処理により新たな確定セットが登録できた場合は true、そうでない場合は false。
- ② 《単一確定セット登録》ステータス： このコンポーネントにより新たな確定セットが登録できた場合において、その確定セットの中で単一確定セットがあれば（つまり、あるマスに数字が決まったのであれば）true。そうでなければ false。
- ③ 《不能》ステータス： このコンポーネントの調査の過程で不能を検出した場合は true、そうでない場合は false。

このコンポーネントでは、《単一確定セット登録》《不能》のいずれかがセットされた場合、現在のマス番号及び数字順を3変数にセットする。

## 2. 1 代表 function： 排他セットの検索

（プログラム中での function 名： `makeExclusiveSet`）

この代表 function (`makeExclusiveSet`) は、このコンポーネント（2. 排他関係の調査と確定セットの新規登録）の入り口であり、ある表示可能性が除外されたときに、「1. 可能性サーチの実行とその結果に基づく状態判定」のコンポーネントから呼ばれる。引数は、除外された表示可能性のマス番号及び数字順である。また、仮置き（副）に際して、「1. 1 状態判定」の主要サブ function である `trySearch (TS)` から呼ばれることもある。

この function は、引数で渡された表示可能性のマス及びその近隣のマスに残っている表示可能性に注目し、新規に登録できる確定セットの候補（排他セット）がないか、以下の（1）（2）（3）（4）により排他関係を調査する。

排他セットが見つければ、その都度、「2. 2 確定セットの登録」の function を呼び出すことにより確定セットとしての登録を試みる。

ただし、登録した確定セットの中に単一確定セットがある場合は、あるマスに

数字が決まったことになるから、それ以降の項目は無視してこの function を終了する。

以下のうち (2) (3) (4) については、その後半は、共通処理であるので、プログラム中では内部 function である sub234 によって行っている。

(1) そのマスが表示可能性として持つ残った数字が  $n$  個のとき (検索すべき排他セットの order が  $n$  のとき)、その  $n$  個の数字の集合を  $A$  とし、次の場合は、 $A$  と見つかった  $n$  個のマスの集合を排他セットとする。

(1-1) 第一段：そのままの order での調査

まず、order= $n$  の場合について調べ、次の場合に排他セットとする。

(a) row ベース：  $A$  に含まれる数字だけを表示可能性として持つマスが同じ row にそのマスを含めてちょうど  $n$  個見つかった場合。

(b) col ベース：  $A$  に含まれる数字だけを表示可能性として持つマスが同じ col にそのマスを含めてちょうど  $n$  個見つかった場合。

(c) box ベース：  $A$  に含まれる数字だけを表示可能性として持つマスが同じ box にそのマスを含めてちょうど  $n$  個見つかった場合。

(1-2) 第二段：order を拡大しての調査

上の (1-1) (a) (b) (c) において、見つかったマスが  $n$  個より少ない場合は、order を大きくしてサーチを試みる。つまり、 $A$  に含まれない数字を一つずつ順に取り上げ、 $A$  に加えてみて新たに同様のサーチを行ってみる。

例えば、引数で渡されたマスに表示可能性として残っている数字が (1, 2) である場合 (つまり、 $A$  が (1, 2)、order=2 である場合)、その row の中に他に二つのマスがあってその残っている数字が (2, 3)、(3, 1) であれば、 $A$  に 3 を加えてみることで、(1, 2, 3) を含んだ order=3 の排他セットを作ることができる。

この処理は、プログラム中では内部 function である sub1abc によって行っている。なお、sub1abc の中で再度同様の事態 (新たなサーチで見つかったマスが新たな order より少ない事態) となった場合は、再帰呼出しによって更に数字を加えてみることも可能であるが、現状ではそのような処理は行っていない。



- (2) そのマスが属する row について、その数字を表示可能性として持つ残ったマスが  $n$  個のとき、その  $n$  個のマスの集合を  $B$  とし、その row の中では  $B$  のいずれかのマスだけに表示可能性として持たれている数字がその数字を含めてちょうど  $n$  個見つかった場合、 $B$  と見つかった  $n$  個の数字の集合を排他セットとする (row ベース)。
- (3) そのマスが属する col について、その数字を表示可能性として持つ残ったマスが  $n$  個のとき、その  $n$  個のマスの集合を  $B$  とし、その col の中では  $B$  のいずれかのマスだけに表示可能性として持たれている数字がその数字を含めてちょうど  $n$  個見つかった場合、 $B$  と見つかった  $n$  個の数字の集合を排他セットとする (col ベース)。
- (4) そのマスが属する box について、その数字を表示可能性として持つ残ったマスが  $n$  個のとき、その  $n$  個のマスの集合を  $B$  とし、その box の中では  $B$  のいずれかのマスだけに表示可能性として持たれている数字がその数字を含めてちょうど  $n$  個見つかった場合、 $B$  と見つかった  $n$  個の数字の集合を排他セットとする (box ベース)。

上の (1) (2) (3) (4) の調査において、見つかったマスが  $n$  個より多い場合は、《不能》ステータスをセットし、現在のマス番号及び数字順を 3 変数 (「currentRow」、「currentCol」、「currentNum」) にセットする。この場合、参考のため、盤面上のマスのうち、《不能》ステータスに関連するマスをマークしておく。ここで「盤面」は、コンテキスト変数の中の一つであるが、当初の呼出し (再帰呼出しでない呼出し) の場合の「盤面」は、画面描画に関するプログラムで定義されている変数であり、本プログラムと共有している。変数名は cr である。

なお、以上の処理により、同じ内容の排他セットを何度も繰り返して「2. 2 確定セットの登録」に対して確定セット候補として送ることがある。その場合は、「2. 2 確定セットの登録」が重複を検知して除去する。

## 2. 2 代表 function : 確定セットの登録

(プログラム中での function 名 : makeDeterminedSet)

この代表 function (makeDeterminedSet) は、排他セットが見つかったときに、

「2. 1 排他セットの検索」の function から呼ばれる。引数は、その排他セットである。また、仮置き(正)に際して、「1. 1 状態判定」の主要サブ function である trySearch (TS) から当代表 function のサブ function である makeDeterminedSet\_body が呼ばれることもある。

この function は、指定された排他セットをチェックして必要な調整処理を行い、登録可能な確定セットを見い出せた場合には、新しい確定セットとして登録する。

この処理は、代表 function である外側 function と、そのサブ function である内側 function とで実行される。

#### (1) 外側 function (makeDeterminedSet)

「2. 1 排他セットの検索」の function から呼ばれ、指定された排他セットの order が 4 以下の場合はその排他セット、5 以上の場合には該当するベース (row、col 又は box) に関する補集合である排他セットを元に、確定セット候補を作成する。確定セット候補は、確定セットと同じデータ構造 (クラス) であり、row ベース、col ベース、box ベースの区別はない。

その後、作成した確定セット候補を引数として、内側 function を呼び出す。

#### (2) 内側 function (makeDeterminedSet\_body)

外側 function 又は自分自身から呼ばれ、引数として渡された確定セット候補の登録を試みる。また、仮置きに際して「1. 1 状態判定」のサブ function である trySearch (TS) から呼ばれることもある。

内側 function は、再帰呼び出しのため、外側 function とは別の function としている。

内側 function では、既存の確定セットを順にサーチし、指定された確定セット候補との包含関係を調べ、以下のように処理する。

ここで、確定セット又は確定セット候補 A、B があるとき、「A が B を包含する」とは、B の全てのマスが A に含まれ、かつ、B の全ての数字が A に含まれることをいい、「A と B が同値である」とは、A が B を包含し、かつ、B が A を包含することをいう。

(a) サーチの過程で指定された確定セット候補と同値である既存の確定セットが見つければ、何もせずに内側 function のサーチを終了してリターンする。

(b) サーチの過程で指定された確定セット候補に含まれる既存の確定セットが見つければ、指定された確定セット候補から見つかった確定セットを引いた確定セット候補を作成し、それを引数として再帰的に内側 function を呼出して登録を試み、サーチを終了してリターンする。

ここで、確定セット又は確定セット候補 A、B があるとき、「A から B を引いた確定セット候補」とは、A のセルの中で B に含まれないものと、A の数字の中で B に含まれないものとで構成される確定セット候補をいう。

(c) サーチの過程で指定された確定セット候補を包含する既存の確定セットが見つければ、見つかった確定セットを削除し、見つかった確定セットから指定された確定セット候補を引いた確定セット候補を作成し、それを引数として再帰的に内側 function を呼び出して登録を試み、さらに指定された確定セット候補を引数として再帰的に内側 function を呼び出し、サーチを終了してリターンする。

ただし、1 回目の呼出しの結果、《単一確定セット登録》ステータスがセットされていた場合、2 回目の呼出しは行わない。したがって、この場合には、指定された確定セット候補が登録されないまま、呼び元に戻ることになる。

(d) 既存の確定セットのサーチが途中で終了することなく最後まで到達したときは、指定された確定セット候補を確定セットとして登録する。

この内側 function (makeDeterminedSet\_body) は、確定セットを登録した場合は《確定セット登録》ステータスをセットする。

また、その中で、単一確定セットを登録した場合（つまり、あるマスが決まった場合）は、さらに《単一確定セット登録》ステータスをセットするとともに、現在のマス番号及び数字順を 3 変数（「currentRow」、「currentCol」、「currentNum」）にセットする。

さらに、単一確定セットを登録したときは、それと矛盾する単一確定セットが既に登録されているかどうか検査し、されていれば、《不能》ステータスをセットする。この場合、単一確定セットを登録はしているが、成功裏に登録したわけではないので、《単一確定セット登録》ステータスを false に戻す。3 変数はセットしたままであり、また対応するマスをエラー表示する。

## 【V. サポート用オプション（補足）】

このプログラムには、解法のための本来のプログラムに加えて、以下のサポート用オプションのプログラムが組み込まれている。

### 1. 解法サポート用オプション

問題を解く手順を示し、ユーザが解法を理解することをサポートするためのオプションである。盤面下部のボタン群の右に小さなチェックボックスがあり、これをチェックすることで、次の2点がサポートされる。

#### （1）仮置き状況の表示

チェックボックスにチェックが入ると、仮置きを行ったうえで単一の答えに行きついた場合（つまり、表1の②と④の場合）、正副の結果がどうであったかを表示する。

#### （2）1 Step ボタンのときの詳細オプションの提供

チェックボックスにチェックが入ると、1 Step ボタンの場合に限り、実行に際してさらにメッセージボックスを表示して次の二つの詳細オプションを選択できる。

##### （a）確定セット登録の表示

確定セット（単一確定セットに限らない）が登録されたとき、その都度、内容を表示する。なお、仮置きを行う場合、仮置き中にも表示を行うかどうかは、改めてメッセージボックスにより選択する。

##### （b）1 Step ボタンの処理終了時の確定セットリストと可能性リストの表示

1 Step ボタンの処理が終了したとき（ただし、単一確定セットが見つかった場合と不能になった場合に限り、不定になった場合を除く。）、その時点での確定セットリストと可能性リストの内容を表示する。

なお、このオプションは、（a）の「確定セット登録の表示」が選択されている場合に限って選択できる。

## 2. プログラムの調査サポート用オプション

プログラムの中には、プログラムの調査（デバッグ）用のコードが埋め込まれたままになっている。詳細は省略するが、次の二つの場合があるから、プログラムを見る場合は注意されたい。

(1) コメントアウトされている。

(2) 特定の変数が true の場合だけ動作するようになっている。その変数は、false にセットされているから、必要な場合は、プログラムを修正して true にする。これらの変数は、プログラム先頭の「0. 1 Recursive call によらない共通変数」の中で宣言されている。

以上

Copyright (C)2014, 2020, 2021, Tetsuya Ushio

この解法についての本ドキュメントの記述は、自由に引用していただいて差支えない（商業目的を除く。）。その際には、一報いただくとともに、潮 哲也が開発した解法であることをクレジットしていただきたい。

参考：SUDOKU ON-THE-GO!, Sterling Publishing Co., Inc. 2006 年。  
これは問題集であるが、4 ページから 5 ページに「Introduction」として解き方のテクニックが紹介されている。