

## Grid-Based Almost Locked Set 入門

Grid-Based Almost Locked Set (GB-ALS) は同じ値の候補数字が  $n$  列 (行) には  $n$  個以上の存在しなければならないという原理を利用して削除可能な候補を探索する手筋です。通常の ALS が候補数字 (値) をベースとした Almost Locked Set を利用するのに対して盤面を  $n$  行  $n$  列の格子 (Grid) として格子上での位置情報をベースとした Almost Locked Set を利用するので GB-ALS と呼ばれます。GB-ALS は Sashimi 系手筋 (Sashimi X-Wing, Sashimi Swordfish, Sashimi Jellyfish など) を一般化したもので Sashimi 系のすべての手筋は GB-ALS として説明可能です。

実例で紹介します。たとえば次の図の 6 行目と 7 行目<sup>1</sup>に注目すると候補数字の 6 は 6 列目、8 列目、9 列目に存在しています。これは、あり得る状態ですが、仮に候補数字の 6 が 6 列目にしか存在しないとすれば 2 行のうち 1 列にしか 6 が存在しないので矛盾です。つまり、この矛盾を発生させた原因に問題があるということです。

#	=====	#
#	. . .   1 2 .   1 2 . # . . .   . . .   . . . # 1 . .   . . .   . . . #	
#	. 6 .   . . .   . . . # . 7 .   . . .   . 3 . # . . .   . 5 .   . 4 . #	
#	. . .   . 8 .   . . . # . . .   . 8 9   . . . # . . . 9   . . .   . . . #	
#	-----	#
#	. . .   . . .   1 . . # . . .   . . .   . . . # 1 . .   . . .   . . . #	
#	. 7 .   4 . .   4 . . # . . 6   . 2 .   . 5 . # . . 6   . . 6   . 3 . #	
#	. . .   . 8 .   . . . # . 8 9   . . .   . . . # . 8 9   . 8 9   . . . #	
#	-----	#
#	. . .   . . .   . . . # . . .   . . .   . . . # . . .   . 2 .   . 2 . #	
#	. 5 .   . 3 .   . 9 . # . . 6   . 1 .   . 4 . # . . 6   . . .   . . . #	
#	. . .   . . .   . . . # . 8 .   . . .   . . . # . 8 .   7 . .   7 . . #	
#	=====	#
#	. . .   . . .   1 2 . # 1 2 .   . . .   . . . # . . .   . . .   1 2 . #	
#	. 4 .   . 6 .   4 . . # . . .   . 5 .   . . . # . 7 .   . 3 .   . . . #	
#	. 8 .   . . .   . . . # . . .   . . .   . 8 9 # . . .   . . .   . . 9 #	
#	-----	#
#	. . . 3   1 2 .   . . . # . . .   . . . 3   . . . # . . .   . 2 .   1 2 . #	
#	. . .   . . .   . 5 . # . 4 .   . . 6   . 7 . # . . 6   . . 6   . . 6 #	
#	. . . 9   . . 9   . . . # . . .   . . . 9   . . . # . 8 9   . 8 9   . . 9 #	
#	-----	#
#	. . . 3   1 2 .   . . . # 1 2 .   . . . 3   . . . # . . .   . . .   1 2 . #	
#	. . .   . . .   . 7 . # . . .   . . .   . . . 6 # . 5 .   . 4 .   . . 6 #	
#	. 8 9   . . 9   . . . # . . .   . 8 .   . 8 9 # . . .   . . .   . . 9 #	
#	=====	#
#	. . .   . . .   . . . # . . .   . . .   . . . # . . .   . . .   . . . #	
#	. 1 .   . 7 .   . 3 . # . . .   . 4 .   . . . 6 # . 2 .   . . 6   . 5 . #	
#	. . .   . . .   . . . # . 8 9   . . .   . 8 9 # . . .   . . 9   . . . #	
#	-----	#
#	. . .   . . .   . . . # . . .   . . .   . . . # . . .   . . .   . . . #	
#	. 2 .   . 5 .   . 8 . # . 3 .   . . 6   . 1 . # . 4 .   . . 6   . . 6 #	
#	. . .   . . .   . . . # . . .   . . . 9   . . . # . . .   7 . 9   7 . 9 #	
#	-----	#
#	. . .   . . .   . . . # . . .   . . .   . . . # . . .   . . .   . . . #	
#	. 4 .   4 . .   . 6 . # . 5 .   . 7 .   . 2 . # . 3 .   . 1 .   . 8 . #	
#	. . 9   . . 9   . . . # . . .   . . .   . . . # . . .   . . .   . . . #	
#	=====	#

1 行は横方向のセルの連なり、列は縦方向のセルの連なり。左上が1行1列で、右下が9行9列。

上の図 (=als-07 の解法途中盤面) の場合、1 行目から順に {1, 478, 47, 2, 5789, 69, 68, 589, 3} の各列に候補数字の 6 が存在しています。6 行目と 7 行目の 6 が存在する列を見ると {69, 689} となっていますが、これは行数と列数の関係に注目すると 2 行の中の 3 列に 6 が存在するという意味で ALS と見なすことができます。5 行 8 列 (以下、r5c8) の 6 は 6 行目に属する r6c9 の 6 とリンク関係にあり、7 行目に属する r7c8 の 6 ともしリンク関係にあります。つまり、もし r5c8=6 と仮定すれば r6c9≠6, r7c8≠6 なので {69, 689} は {6, 6} となりますが、これは 6 行目と 7 行目の 2 行のうち 6 が存在するのが 6 列目と 6 列目、つまり 6 列目だけにしか 6 が存在しないということを表しています。数独のルール上、どの行にも 6 は 1 カ所ずつ存在しなければなりませんから、これは矛盾です。つまり r5c8=6 と仮定したことが矛盾を導いたことになります。故に r5c8≠6 なので、この r5c8 の 6 を削除できます。r7c9 の 6 も同様の理由により削除できます。一般化すると次のようになります。

**定理 1 : n カ所の列 (または行) 内の候補数字 p が n+1 カ所の行 (または列) に存在するとき、これらの要素でないあるセル x に p を仮定すると、n 個の列 (または行) の候補数字 p が n-1 カ所以下の行 (または列) にしか存在できなくなるならば、セル x の候補数字 p を削除できる。**

具体的には 1 から 9 の各候補数字に対して列方向と行方向の両方向から Grid Base で ALS を探し、列方向での GB-ALS であれば、あるセル x に候補数字 p を仮定すると n カ所の列の n-1 カ所以下の行にしか候補数字 p が存在できなくなるセル x の候補数字 p を探索します。行方向での GB-ALS であれば、あるセル x に候補数字 p を仮定すると、n カ所の行の n-1 カ所以下の列にしか候補数字 p が存在できなくなるセル x の候補数字 p を探索します。

GB-ALS は「Almost Locked Set 入門」で紹介した ALS に比べると探索範囲は狭く、ALS では削除できない候補数字も削除できるなど強力です。ただし、ALS ほど多くの盤面で候補を削除できる訳ではないので、やや効力を発揮しにくい手筋です。**ポイントは 2 つの GB-ALS にリンク (= 2 重リンクと呼びます) する候補数字を発見することです。**ただし、このような候補数字すべてが条件を満たす訳ではないので更にチェックを進める必要があります。

## 間接的に矛盾が発生する「隠れ GB-ALS」

厳密には定理 1 の表現でカバーされていますが 1 段階余分にチェックしないと見過ごしてしまうパターンがあります。あるセル x に候補数字 p を仮定した場合、この候補数字 p によって特定の列または行の候補数字 p が確定値となる場合があります。この確定値となる p が、更に他の候補数字 p を削除することで矛盾が発生する場合があります。例で確認してみましょう。次の図は NumberPlace.lisp に添付の例題 (=als-06) です。

#	=====	#
#	. . .   . . .   . . . # . . 3   . . .   . . 3 # . . .   . . .   . . . #	#
#	. 5 6   . 7 .   . 4 . # . . 6   . 2 .   . . . # . 5 .   . 8 .   . 1 . #	#
#	. . 9   . . .   . . . # . . 9   . . .   . . 9 # . . 9   . . .   . . . #	#
#	-----	#
#	. . .   . . .   . . . # . . .   . . .   . . . # . . .   . . .   . . . #	#
#	. 2 .   . 1 .   . . 6 # . 5 .   4 . 6   . 8 . # . 3 .   4 . 6   . 7 . #	#
#	. . .   . . .   . . 9 # . . .   . . 9   . . . # . . .   . . 9   . . . #	#
#	-----	#
#	. . .   . . .   . . . # . . .   . . .   . . . # . . .   . . .   . . . #	#
#	. 8 .   . 3 .   . 5 6 # . 1 .   4 . 6   . 7 . # . 5 6   . 2 .   4 . 6 #	#
#	. . .   . . .   . . 9 # . . .   . . 9   . . . # . . 9   . . .   . . 9 #	#
#	=====	#
#	. . .   . . .   . . . # . . .   . . .   . . . # . . .   . . .   . . . #	#
#	. 5 .   . 2 .   . 7 . # . 4 .   . 8 .   . 6 . # . 1 .   . 3 .   . 5 . #	#
#	. . 9   . . .   . . . # . . .   . . .   . . . # . . .   . . .   . . 9 #	#
#	-----	#
#	. . .   . . .   . . . # . . 3   . . .   . . 3 # . . .   . . .   . . . #	#
#	. 4 .   . . 6   . 1 . # . . .   . 5 .   . 5 . # . . 6   . 7 .   . 2 . #	#
#	. . .   . 8 .   . . . # . . 9   . . 9   . . 9 # . 8 .   . . .   . . . #	#
#	-----	#
#	. . .   . . .   . . . # . . .   . . .   . . . # . . .   . . .   . . . #	#
#	. 5 6   . 5 6   . 3 . # . 2 .   . 7 .   . 1 . # . . 6   4 5 6   4 5 6 #	#
#	. . 9   . 8 9   . . . # . . .   . . .   . . . # . 8 9   . . 9   . . 9 #	#
#	=====	#
#	. . .   . . .   . . . # . . .   . . .   . . . # . . .   . . .   . . . #	#
#	. 3 .   . 4 .   . 8 . # . . 6   . 1 .   . 2 . # . 7 .   . 5 6   . 5 6 #	#
#	. . .   . . .   . . . # . . 9   . . .   . . . # . . .   . . 9   . . 9 #	#
#	-----	#
#	. . .   . . .   . . . # . . .   . . .   . . . # . . .   . . .   . . . #	#
#	. 1 .   . 5 6   . 5 6 # . 7 .   . 3 .   . 4 . # . 2 .   . . 6   . 8 . #	#
#	. . .   . . 9   . . 9 # . . .   . . .   . . . # . . .   . . 9   . . . #	#
#	-----	#
#	. . .   . . .   . . . # . . .   . . .   . . . # . . .   . . .   . . . #	#
#	. 7 .   . . 6   . 2 . # . 8 .   . 5 6   . 5 . # . 4 .   . 1 .   . 3 . #	#
#	. . .   . . 9   . . . # . . .   . . 9   . . 9 # . . .   . . .   . . . #	#
#	=====	#

6が存在するセルを列方向にプロットすると青地のセルは{16, 17, 367}となっています。3列に対して{1, 3, 6, 7}と4つの行に6が存在するのでGrid BaseでAlmost Locked Setを構成しています（GB-ALS）。このときr6c8=6を仮定するとGB-ALSの要素セル内のr6c1とr6c9の6が消えるのでALS自体は{1, 17, 37}となり3列に対して{1, 3, 7}行に6が存在することになります。3列に対して3行に6が存在すること自体は矛盾ではありません。

しかしr6c8=6となることで同じブロック内の黄色字のr5c8とr6c8の6が消えます。その結果r3c7の6が7列目で唯一の6となるので確定値となります。その結果、r3c9の6も消えます。つまりALSは{1, 17, 7}となります。これは3列に対して1行目と7行目の2行にしか6が存在しないということを表していますので矛盾です。このように間接的に矛盾が導かれるパターンを、ここでは「隠れGB-ALS」と呼ぶことにします。手動でチェックする場合は、**定理1にしたがったチェックで一見矛盾が生じていないように見えた場合でも確定値が発生していないかにも注意する必要があります**。NumberPlace.lispでは、このようなパターンを次のように表示します。

0:0> Grid-Based Almost Locked Setにより[@]の位置から候補を削除できます。

[#]は[6]に関して列方向のGrid-Based ALSを構成しています。

r6c8の[@]に[6]が存在するとr3c7の[\$]が列内での確定値となりGB-ALSの3列のうち2行にしか[6]が存在できないの

で矛盾です。

=> r6c8<>6

+-----+-----+-----+		
# - -   # - -   - - -		
- - =   - + -   - + -		
- - +   - + -   \$ - #		
+-----+-----+-----+		
- - -   - - 6   - - -		
- = -   - - -   = - -		
# + -   - - -   + @ #		
+-----+-----+-----+		
- - -   # - -   - + #		
- + +   - - -   - = -		
- = -   - + -   - - -		
+-----+-----+-----+		

この例題の場合 r6c8 の 6 は削除できます。手動でのチェックでは見逃しやすいので特に注意が必要です。ただし見逃したとしても、その後の解法に悪影響を与える訳ではないのであまり気にする必要はありません。

参考 url

<http://www.stolaf.edu/people/hansonr/sudoku/explain.htm> （例題出典もこちら）

[http://sudopedia.org/wiki/Solving\\_technique](http://sudopedia.org/wiki/Solving_technique) （用語の定義等）

([mail2daigo@gmail.com](mailto:mail2daigo@gmail.com)) ■