COCO2P Package の紹介

Introduction of COCO2P Package

宮本 泉 [*1]
MIYAMOTO IZUMI

Abstract

We will briefly introduce COCO2P Package, which is a GAP-package for the computation with color graphs, especially with coherent configurations. The author is particularly interested in computing automorphisms and isomorphisms of them. Some experiments of computing automorphism groups are shown. Fusions of color graphs are computed by COCO2P. We can see whether a color graph is a coherent configuration or not by COCO2P function IsWLStableColorGraph and also see whether a coherent configuration is Schurian or non Schurian by computing automorphism group of it. We will give an example of a fusion scheme which is WL-stable but not a coherent configuration from [3].

1 はじめに

COCO2P Package by Mikhail Klin, Christian Pech, Sven Reichard.
パッケージの説明：GAP-package for the computation with coherent configurations

COCO パッケージは、現在のバージョンが COCO2P-0.17 となっている。GAP が認定して GAP ファイルといっしょに配布されるパッケージには、(まだ、) 含まれていない。自分でダウンロードして、GAP システムの pkg ディレクトリに COCO2P ファイルを置く。 コンパイルは不要。そして、GAP を起動して、

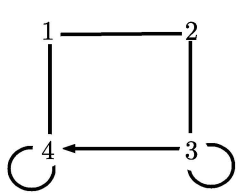
```
gap> LoadPackage(”coco2p”);
```

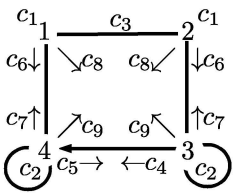
により、使用できるようになる。このとき、GRAPE パッケージが、先に、Load される。

2 COCO2P マニュアルの Contents その 1

[*1] E-mail: imiyamoto1@gmail.com

【例】カラーグラフ 頂点集合 $\{1,2,3,4\}$, 頂点のカラー $\{c_1, c_2\}$, カラー辺集合 $\{c_3, c_4, \cdots, c_9\}$





$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} c_1 & c_3 & c_8 & c_6 \\ c_3 & c_1 & c_6 & c_8 \\ c_9 & c_7 & c_2 & c_4 \\ c_7 & c_9 & c_5 & c_2 \end{pmatrix} \end{array}$$

GAP による実行例

```
gap> mat:=[["c1","c3","c8","c6"],
>          ["c3","c1","c6","c8"],
>          ["c9","c7","c2","c4"],
>          ["c7","c9","c5","c2"]];;
gap> cgr:=ColorGraphByMatrix(mat);
<color graph of order 4 and rank 9>
gap> AdjacencyMatrix(cgr);
#頂点もカラーも番号 (自然数) で処理される。この配列表示の定義は後で。
[ [  1,  3,  8,  6 ],
  [  3,  1,  6,  8 ],
  [  9,  7,  2,  4 ],
  [  7,  9,  5,  2 ] ]
gap>AdjacencyMatrix(cgr,[6]);AdjacencyMatrix(cgr,[1,2]); #カラー番号別の配列表示
[ [  0,  0,  0,  1 ],   [ [  1,  0,  0,  0 ],
  [  0,  0,  1,  0 ],     [  0,  1,  0,  0 ],
  [  0,  0,  0,  0 ],     [  0,  0,  1,  0 ],
  [  0,  0,  0,  0 ] ]    [  0,  0,  0,  1 ] ]
```

【定義】 order：グラフの頂点の個数、rank：カラーの個数

COCO2P パッケージによる設定の便利な点：

カラーグラフの出力形式について、上に示したように、　例えば、グラフを行列表示すると、order が大きいときは画面での表示に支障がでる。また、その後で、行列から rank を確認することになる。一方で、簡単な例では行列表示して見てみたい。

GRAPE パッケージについて：

グラフに関する計算をする GAP 認定パッケージ。GAP 本体といっしょに配布されている by Leonard H. Soicher。この中の、**同型計算**のために使っている nauty package は B. D. McKay による。この部分のみ**コンパイルが必要**。その他、GAP とのインターフェイスに関しては、Steve Linton, Alexander Hulpke などの GAP 開発者がかかわっている。

COCO2P では、GRAPE はコンパイルしないでも使える。

グラフについての GRAPE の説明：

finite directed graph で、loop は許すが、multiple edge は無いものを取扱う。

多くの function は、simple graph (loop 無しで、 [x, y] が edge なら [y, x] も edge) を対象とする。

一般的に、グラフの細かい説明はややこしい。以下の例などでは、GRAPE はとりつきにくい。

カラーグラフにすれば、グラフに関する細かい定義は不用になる（COCO の説明)。

【Example1】

```
gap> A := [ [ 0, 1, 0 ],
            [ 1, 0, 0 ],
            [ 0, 0, 1 ] ];;
gap> gamma := Graph( Group(()), [1..3], OnPoints,
> function(x,y) return A[x][y]=1; end, true );;
```

行列 A を adjacency matrix とするグラフを作るのに手間がかかる。グラフ gamma から、行列 A を復元する function が見つからない。

【Example2】

```
gap> EdgeOrbitsGraph( Group((1,3),(1,2)(3,4)), [[1,2],[4,5]], 5 );
rec( isGraph := true, order := 5, group := Group([(1,3),(1,2)(3,4)]),
     schreierVector := [-1,2,1,2,-2], adjacencies := [[2,4,5],[ ]],
     representatives :=[1,5], isSimple := false )
```

isSimple=false$\Longleftrightarrow$[1,5] は edge なのに [5,1] は edge ではない。(有向辺, directed edge)

COCO の歴史：

- Faradzev, I.A., Klin, M.H., Computer package for computations with coherent configurations. In: ISSAC 1991, Proc. Symposium on Symbolic and Algebraic Computation, pp. 219-221.
- Pasechnik, D., Kini, K., Cohcfg, a GAP package for coherent configurations (preliminary version) (2010) (コンパイルが必要な単独のパッケージ。GAP package になっていない。)

  置換群 $\longrightarrow$coherent configuration $\longrightarrow$ 自己同型群の計算のみ

- Pasechnik, D., Kini, K., Cohcfg, A GAP Package for Computation with Coherent Configurations, International Congress on Mathematical Software, ICMS 2010, pp 69-72

(以上、ICMS2010 の文献より)

3　今回の紹介について

サイズが大きい coherent configuration の同型計算で、自前のプログラムでは困難な場合がでてきたとき、COCO2P を知って、実験に使うことを試みた。

【定義】coherent configuration $(\Omega, \{A_i\}_{i=1,2,\cdots,d})$ on $\Omega = \{1, 2, \cdots, n\}$ (combinatorial property)

1. $A_i (i = 1, 2, \cdots, d)$ は、n 次正方 0, 1 matrix (d： rank)
2. $A_1 + A_2 + \cdots\cdots + A_d = J$ (all 1 matrix)
3. $A_1 + A_2 + \cdots + A_r = I$ (単位行列, $\exists r < d$)($\Longleftarrow$ 頂点集合に対応)
4. ${}^tA_i = A_j$ for some j (color mate という。$j = i^*$ で表す。)(逆向きの辺への対応とみなせる。)
5. $A_iA_j = \sum_{k=1}^{d} p_{ij}^k A_k$ ($\{A_i\}_{i=1,2,\cdots,d}$ は A_i 達の生成する algebra の basis となる。)

【定義】 A_i を color i に関する adjacency matrix という。
【定義】 coherent configuration の relation matrix を、$A = 1A_1 + 2A_2 + \cdots + dA_d$ で定める。([1] など)

Color Graph の作り方

- 行列 A から、ColorGraphByMatrix(A) (最初の例)
- 置換群 G からは、ColorGraphByOrbitals (G [, option...])

【定義】*orbital* : G を Ω 上の置換群としたとき、G を $\Omega \times \Omega$ に $[i,j]^g = [i^g, j^g]$, $g \in G$ で作用させたときの orbit。

その他にもあるが、上の 2 つがわかり易い。良く知られたグラフを作る関数も数多くある。

COCO2P における coherent configuration -カラーグラフとして-
coherent configuration の定義における adjacency 行列達 $A_1, A_2, \cdots, A_d$ に対して、

$$\begin{aligned} A &= 1A_1 + 2A_2 + \cdots + dA_d \\ \mathrm{cgr} &= \mathsf{ColorGraphByMatrix}(A) : \text{coherent configuration} \\ (\text{このとき} \quad A &= \mathsf{AdjacencyMatrix}(\mathrm{cgr}) \quad \text{が成立する。}) \\ \text{また、置換群 } G \text{ に対して} \quad \mathrm{cgr}' &= \mathsf{ColorGraphByOrbitals}(G) : \text{coherent configuration} \end{aligned}$$

AutGroupOfCocoObject(cgr) : カラーグラフの自己同型群を計算する。Color 番号は、固定したままで取り扱う。このときは、カラーグラフであれば、coherent configuration である必要はない。
AutomorphismGroup(cgr) : AutomorphismGroup は、GAP 本体にもある ” 汎用的 ” な関数で、例えば、群の自己同型群を求めるときにも使える。COCO2P Package では、カラーグラフにも使えるように設定している。
カラーグラフ cgr が、置換群 G を使って、ColorGraphByOrbitals(G) で作られるときは、G が cgr の自己同型群には含まれるという情報を使って自己同型群計算が行われる。
Color 番号の付け替えも考慮する関数を使うときは、coherent configuration であることを、明示的に、IsWLStableColorGraph を使って true の判定が必要。マニュアルでは、

This function returns true if cgr is stable under the Weisfeiler-Leman algorithm, that is, whether it is the color graph of a coherent configuration.

となっている。同型計算は、Weisfeiler-Leman algorithm を基本にしている。

COCO の同型計算プログラム pbag にある説明 (抜粋)
(Partition Backtracking Algorithm using GAP by Ch. Pech)
first version 1997/3, pbag.g, v 1.3 in GAP 1999/08.
stabilizer chain of a sub-group of its automorphism group 2007/11,
hashing by GAP4 2008/11, hashtable code changed in GAP 4.5 2012/11

参考資料 : I.Faradzev (in COCO), J.S.Leon (1984), G.Tinhofer (1997)

・This preliminary version is not yet optimized for speed.
・For CGRs graphs up to 200 vertices were handeled very well; this means in a few seconds.

4 自己同型群の計算例

サイズが小さい場合の計算状況

【例】 as06[3] アソシエーションスキームの分類番号、6 次の 3 番目

講演者の program "auto" の出力結果：[[(), (5,6)], [(), (4,5), (4,6)], [(), (2,3)(4,6)],
[(), (1,2)(4,6), (1,3)(4,6), (1,4,3,5,2,6), (1,5,2,4,3,6), (1,6)(2,5)(3,4)]]

AutGroupOfCocoObject の出力結果：Group([(5,6), (2,3), (4,5), (1,2), (1,4)(2,5)(3,6)])

【注意】 association scheme の分類リスト as では、regular な置換群のつくるものは除外している。

サイズが小さい場合の計算時間の比較

	個数	AutCoco	auto	auto2		個数	AutCoco	auto	auto2
as24	735	18404	4243	3590	Tr24	25000	299052	101156	191660
as25	43	4501	250	422	Tr25	211	4898	1207	1526
as26	32	7078	205	484	Tr26	96	2554	817	786
as27	497	547340	3427	20388	Tr27	2392	66493	14124	21158
as28	181	7496	1409	1583	Tr28	1854	49076	14518	15678
as29	25	28219	126	902	Tr29	8	8156	125	93
as30	239	12873	2870	2155	Tr30	5712	174267	50041	66230
as32	18159	1015293	557947	531845	Tr：TransitiveGroup				

計算時間はミリ second, auto2 は auto の改良版

計算状況の例が示すように、AutCoco では群論を細かく利用している。その効果は、サイズが小さい場合では認められない。

自己同型群の計算例　サイズが大きい場合

PrimitiveGroup(n,i) GAP の置換群データ、n 次数, $1 \leq i \leq$NrPrimitiveGroups(n)

群が 2 重可移の場合は、rank 2 の自明な coherent configuration になるので、計算から除外する。

G :=PrimitiveGroup(n,i);

cgr:=ColorGraphByOrbitals(G);

mat:=AdjacencyMatrix(cgr); (講演者の mat 計算プログラムより COCO の方が倍くらい速い)

AutoGroupOfCocoObject(cgr); $\#G$ を KnownGroupOfAutomorphisms として利用

講演者のプログラム：auto(mat); auto2(mat); auto2(mat,G);

次数 n の範囲	個数	AutCoco	auto	auto2	aut2 with G
PrimitiveGroup31-100	421	6949	90620	36300	6477

次数 n の範囲	個数	AutCoco	aut2 with G
PrimitiveGroup101-200	435	15,250	19,989
PrimitiveGroup201-300*	563	63,120	398,790
PrimitiveGroup301-400**	409	538,652	1,815,842
PrimitiveGroup401-500	252	64,160	84,808

*：243 次全部で、AutGroupOfCocoObject は 15,100,796 ミリ second かかっている。

PrimitiveGroup(243,30)$\cong 3^5 : \mathrm{M}_{11}$ 一個で実験したとき、15,055,437(約 4 時間***) であった。

残りの 243 次すべてで 1816 ミリ second

：400 次のとき、PrimitiveGroup(400,3) で、auto2 は計算が終わらなかった (10 時間以上*)。この場合は除外してある。

***：COCO2P の pbag の説明にも書かれているが、backtrack をするときの base points の選び方で計算時間が劇的に変わる。頂点やカラー番号のつけ方で base points が変わる。

自己同型群の計算例　サイズが大きい場合 その２

G =Stabilizer(PrimitiveGroup$(n,i),n)$ 次数 $n-1$, $1 \leq i \leq$NrPrimitiveGroups(n)

G が primitive になる場合は除いた。G が IsSemiRegular の場合も除いた。

次数 n の範囲	個数	AutCoco	aut2 with G
PrimitiveGroup31-100	372	35,690	10,640
PrimitiveGroup101-200	271	277,034	43,801
PrimitiveGroup201-300	442	2,482,530	601,741
PrimitiveGroup301-400	242	8,247,121	1,561,384
PrimitiveGroup401-427	-	1,297	3,687

いくつかの特徴的な場合

n	AutCoco	auto2 with G	n	AutCoco	auto2 with G
243	64,441	131,105	364	974,610	653,266
273	1,896,188	44,905	381	2,880,390	56,219
341	1,012,719	307,624	400	2,712,562	264,066

AutCoco と講演者の aut2 with G の計算時間は大差無いようであるが、計算困難な場合が異なる。

5　COCO2P マニュアルの Contents その２

```
2 Structure Constants Tensors
 2.1 Introduction
 2.2 Functions for the construction of tensors 2.2-1～2.2-2
 2.3 Functions for the inspection of tensors 2.3-1～2.1-13
 2.4 Testing properties of tensors 2.4-1～2.1.4
 2.5 Symmetries of tensors 2.5-1～2.5-6
 2.6 Character tables of structure constants tensors 2.6-1
```

【例】
```
gap> T:=StructureConstantsOfColorGraph(cgr);
<Tensor of order 4>
gap> ClosedSets(T);
[ [ 1 ], [ 1 .. 4 ], [ 1, 3 ], [ 1, 4 ] ]
```

adjacency 行列において、$A_iA_j = \sum_{k=1}^{d} p_{ij}^k A_k$ の p_{ij}^k を structure constant という。

$\langle A_1 \rangle$, $\langle A_1, A_3 \rangle$, $\langle A_1, A_4 \rangle$ が、$\langle A_1, \cdots, A_4 \rangle$ の subalgebra になる。

```
3 WL-Stable Fusions Color
 3.1 Introduction
 3.2 Good sets 3.2-1～3.1-4
 3.3 Orbits of good sets 3.3-1～3.1-11
 3.4 Fusions 3.4-1～3.1-6
 3.5 Orbits of fusions 3.5-1～3.1-11
4 Partially ordered sets
 4.1 Introduction
 4.2 General functions for COCO2P-posets 4.2-1～4.2-11
 4.3 Posets of color graphs 4.3-1～4.3-3
5 Color Semirings
 5.1 Introduction 5.1-1～5.1-3
```

群 $G \supseteq H$ 部分群 subgroup が考えられる。coherent configuration の sub-coherent configuration は、fusion scheme と呼ばれる。それを下に示す。

Fusion Scheme

【例】 D が C の fusion scheme

C：A_1, A_2, A_3, A_4 を adjacency 行列とする rank4 の coherent configuration とする。

D：$A_1, A_2, A_3 + A_4$ が rank 3 の coherent configuration となる。

```
gap> D:=ColorGraphByFusion(C,[[1],[2],[3,4]]); #color graph として構成
```

- $\langle A_1, A_2, A_3 + A_4 \rangle \subset \langle A_1, A_2, A_3, A_4 \rangle$　（3 次元）subalgebra になる。
- AutomorphismGroup(C)$\subseteq$ AutomorphismGroup(D)

fusion scheme 関係は poset になる。

下の計算例では、regular な置換群からできるものを除いた 181 個の as28 から ColorGraphByMatrix により作られる coherent configuration のリストを cgr28 として、この集合の fusion scheme 関係による poset を求めている。

```
gap> scip28:=SubColorIsomorphismPoset(cgr28); #計算時間 1,665,703 ミリ seconds
<object>   #28 次アソシエーションスキーム全体 (regular 群が作る場合は除く) の作る poset
gap> PredecessorsInCocoPoset(scip28,176); #as28[176]
[ 155, 110, 75 ] #poset として含まれる極大 fusion scheme
```

【例】 as28[176] の fusion scheme 全体（poset の包含関係を示す資料あり [1, 2]）

```
gap> cgr28[176];
<color graph of order 28 and rank 16>
gap> IdealInCocoPoset(scip28,176);
[ 1, 2, 3, 7, 10, 59, 72, 75, 76, 79, 87, 91, 95, 110, 111, 113, 114, 138,
  145, 149, 155, 176 ]
gap> ColorIsomorphicFusions(cgr28[176],cgr28[114]);
[ <FusionOrbit of length 3> ]
gap> last[1];;
gap> AsList(last);
[ <fusion of order 16 and rank 7>, <fusion of order 16 and rank 7>,
  <fusion of order 16 and rank 7> ]
gap> List(last,AsPartition);
[ [ [ 1 ], [ 2 ], [ 3, 4 ], [ 5, 6, 7, 8 ], [ 9, 10, 11, 12 ], [ 13, 16 ],
      [ 14, 15 ] ],
  [ [ 1 ], [ 2, 4 ], [ 3 ], [ 5, 6, 7, 8 ], [ 9, 11 ], [ 10, 12 ],
      [ 13, 14, 15, 16 ] ],
  [ [ 1 ], [ 2, 3 ], [ 4 ], [ 5, 7 ], [ 6, 8 ], [ 9, 10, 11, 12 ],
      [ 13, 14, 15, 16 ] ] ]
gap> ColorGraphByFusion(cgr28[176],last[1]);
<color graph of order 28 and rank 7>
gap> IsWLStableColorGraph(last);
true
```

```
gap> ColorIsomorphismColorGraphs(last2,cgr28[114]);
(5,13,21)(6,16,26,11,15,27,7,17,28,8,19,23,9,20,24,10,18,25,12,14,22)
```

【注】 これは、頂点番号の対応で、カラー番号の対応は、(関数が見つからないので) relation matrix を関数 AdjacencyMatrix によって直接見て確認。

6 Fusion Scheme の例 ([3] より)

GAP の PrimitiveGroup のデータや、それらから GAP 関数 SubidirectProducts を使って作った置換群から coherent configuration を構成して、その fusion scheme を [3] でいくつか計算している。そこでは、群からは作られない (non Schurian) scheme の例を探して紹介している。

その中にある、WL-stable なのに coherent configuration にならない例をここに参照する。

COCO2P マニュアルには、WL-stable ならば、カラーグラフは coherent configuration になるとある。

```
gap> G:=PrimitiveGroup(81,135);
3^4:Sym(6)
gap> SDs:=SubdirectProducts(G,G);;Length(last);
10 # SDs の群は、並び順や共役で動いて、毎回、ちがう群が出てくる。
gap> SD; # SDs の中の一つの群
<permutation group of size 58320 with 3 generators>
gap> CG:=ColorGraphByOrbitals(SD);
<color graph of order 162 and rank 14>
gap> OutValencies(CG); #この条件を満たすように SD を選ぶ
[ 1, 30, 20, 30, 15, 6, 60, 15, 60, 6, 1, 30, 20, 30 ]
```

下の ptn0 の取り方として、[[1], [3, 4], [2], $\cdots$]; としても [[1], [2, 3], [4], $\cdots$]; としても、OutValencies は [1, 50, 30, $\cdots$]; となるが、WL-stable となるのは、そのうちの 1 つだけ。

【注】 [1, 2, 3, 4] の部分だけで次数 81 の association scheme になるが、そのときはどちらの fusion も association scheme になる。

```
gap> ptn0:=[ [ 1 ], [ 3, 4 ], [ 2 ], [ 5, 7 ], [ 6 ], [ 8, 9 ],
 [ 10 ], [ 11 ], [ 13, 14 ], [ 12 ] ];;
gap> ptn:=[ [ 1 ], [ 3, 4 ], [ 2 ], [ 5, 6 ], [ 7 ], [ 9, 10 ],
 [ 8 ], [ 11 ], [ 13, 14 ], [ 12 ] ];;
gap> FS0:=ColorGraphByFusion(CG,ptn0);;FS:=ColorGraphByFusion(CG,ptn);;
gap> IsWLStableColorGraph(FS0);IsWLStableColorGraph(FS);
true    #FS0 が WL-stable となるように ptn0 を選んだうえで
true    #FS が WL-stable で下の valency となる ptn を選ぶ。
gap> OutValencies(FS); ColorMates(FS);
[ 1, 30, 50, 21, 60, 15, 66, 1, 30, 50 ] #rank 10
(4,6)(5,7) #4 番と 6 番カラーは 21,15 と valency が異なる。5 番 7 番も同様。
```

$\Longrightarrow$4、6、5、7 番カラーは color mates がない。

relmat(FS0),relmat(FS) をそれぞれの relation matrix とし、それらを SD の orbit で区分けした行列で示すと次のようになる。

$$\text{relmat(FS0)}=\begin{array}{c} \\ 1 \\ \vdots \\ 81 \\ \\ 1' \\ \vdots \\ 81' \end{array}\begin{array}{c} \begin{array}{cccccc} 1 & \cdots & 81 & 1' & \cdots & 81' \end{array} \\ \begin{pmatrix} \text{rank3} & & & \text{rank2} & & \\ \text{OutValencies} & & & \text{OutValencies} & & \\ 1,30_1,30_2+20 & & & 60+15,6 & & \\ & & & & & \\ \text{rank2} & & & \text{rank3} & & \\ \text{OutValencies} & & & \text{OutValencies} & & \\ 60+15,6 & & & 1,30_1,30_2+20 & & \end{pmatrix} \end{array}$$

SD の orbit 2 個:
$\{1,2,\cdots,81\}$,
$\{1',2',\cdots,81'\}$

【注】 `AutomorphismGroup(FS0)=SD` となるので、fusion scheme `FS0` は群からは作られない coherent configuration になる。relmat(FS0) から、strongly regular graph と strongly regular design が得られる。

$$\text{relmat(FS)}=\begin{pmatrix} \text{rank3} & \text{rank2} \\ \text{OutValencies} & \text{OutValencies} \\ 1,30_1,30_2+20 & 60,15+6 \\ \text{rank2} & \text{rank3} \\ \text{OutValencies} & \text{OutValencies} \\ 60+6,15 & 1,30_1,30_2+20 \end{pmatrix}$$

coherent configuration にはならない
rank 10
OutValencies
$1,30,50;60,21;$
$66,15;1,30,50$

ColorMates $i \leftrightarrow i^*$ の対応以外の coherent configuration の条件は満たす。
FS の 10 個の adjacency matrix の作る algebra $\langle A_1,\cdots,A_{10}\rangle$ は 10 次元。

```
gap> Alg:=Algebra(Rationals,List([1..14],u->AdjacencyMatrix(CG,[u])));
<algebra over Rationals, with 14 generators>
gap> AlgFS:=Subalgebra(Alg,List([1..10],u->AdjacencyMatrix(FS,[u])));
<algebra over Rationals, with 10 generators>
gap> Dimension(AlgFS);
10
```

coherent configuration CG の adjacency 行列が作る 14 次元 algebra のなかで、fusion scheme は 10 次元 subalgebra を生成する。

coherent configuration の定義のうち、WL-stable は、adjacency matrix 達がそれらの生成する algebra の basis になる条件の確認をしているらしい。この例では、${}^tA_i$ で示される逆向きの辺は存在しない場合になっている。

参 考 文 献

[1] A. Hanaki and I. Miyamoto, Classification of association schemes of small order, Disc. Math. 264(2003) 75–80.

[2] M. Klin, M. Muzychuk, Ch. Pech, A. Woldar and P.-H. Zieschang, Association schemes on 28 points as fusions of a half-homogeneous coherent configuration, Europ. J. Comb. 28(2007) 1994–2025.

[3] 宮本 泉, 群から作られない fusion scheme の計算, 第 31 回有限群論草津セミナー報告集 (2019) 57–62.