正則グラフでアクセス構造を表現する非コミットメント型カードプロトコル

IIJ
Internet Initiative Japan

須賀祐治

2024-05-22

Ongoing Innovation

XOR演算は2-party 2値入力「一致関数」

- 入力が同じなら出力0
- そうじゃなければ出力1(入力が違った)

この考え方は自然に
拡張できますね

$a \backslash b$	0	1
0	**0**	1
1	1	**0**

$a \backslash b$	i_1	i_2
i_1	**0**	1
i_2	1	**0**

2-party 多値入力「一致関数」

- 入力が同じなら出力0
- そうじゃなければ出力1(入力違)

$a \backslash b$	i_1	i_2	i_3	i_4
i_1	0	1	1	1
i_2	1	0	1	1
i_3	1	1	0	1
i_4	1	1	1	0

例えば
4値入力

須賀からの未解決問題2023 (1/2)

- **2-party多値入力の一致関数の実現**
  （ナイーブな手法については提示済）

- (Hamming schemes での構成事例を提示した上で)
  Johnson scheme J(v,k) に
  関連するカードプロトコルの構成
  （もしくは既存のカードプロトコルが
  Johnson schemeと関連している事例の例示）

須賀からの未解決問題2023 (2/2)

- 位数4,6のアソシエーションスキームと関連するカードプロトコルの構成
- 2枚入力 3-valued n-party 一致関数の構成
  （緩い条件：上下カード・非コミットメント型）
- カード入力を制限する一般的な構成
- カード入力として1枚カードを追加することによる新しいプロトコルを構成できるが,
  この拡張方式に呼応する一般的拡張方式の提示

今回扱うカードプロトコル

- 2-party k-valued（2者間 k通りの入力選択肢）
- 非コミットメント型（一発勝負）
- デッキ分割法を適用
  （わりと軽い方法だと信じてる）
- カード入力に「制限」を設けることで
  できることのバリエーションを増やす戦略

なにをやろうとしてるのか？

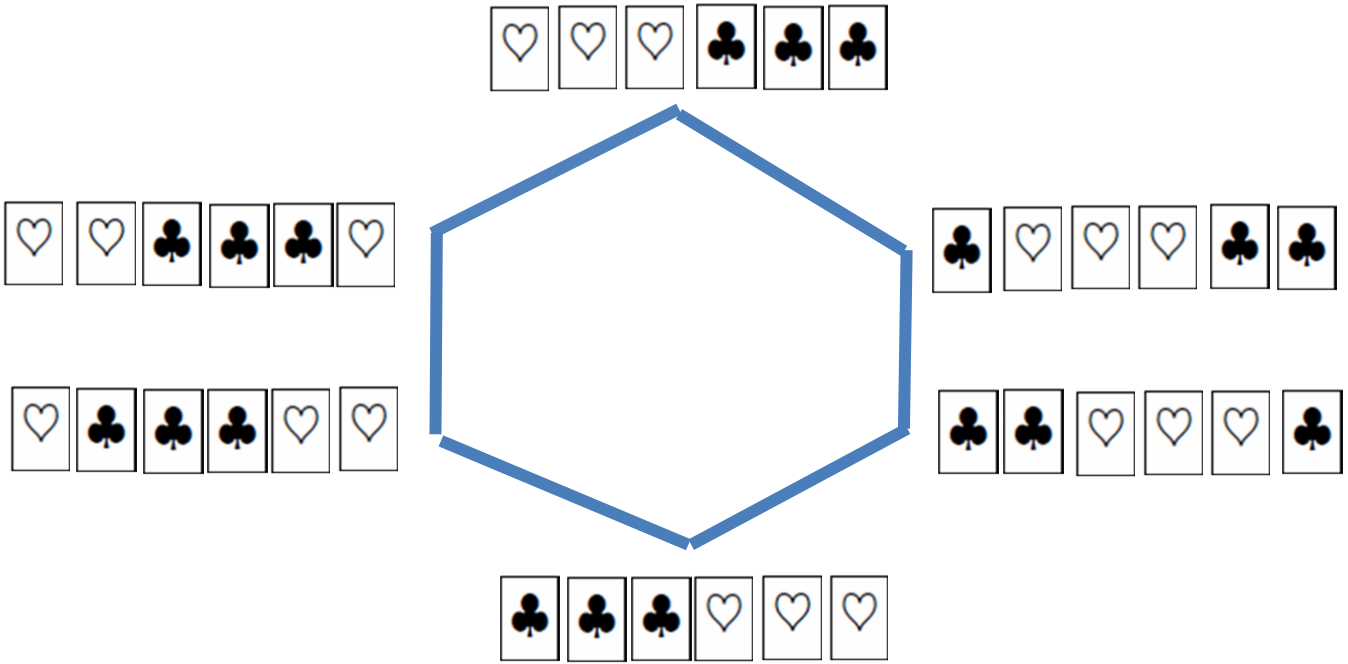
- 2人のプレーヤー・2色カードの利用
- 事前に与えられたグラフの「頂点」を選択してカードを使って入力

なにをやろうとしてるのか？

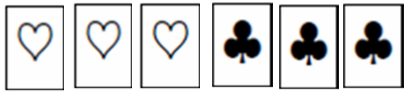
- 2人のプレーヤー・2色カードの利用
- 事前に与えられたグラフの「頂点」を選択してカードを使って入力 
- エンコーディングルールは標準モデルではない

「グラフの頂点を入力する」例

- A given graph = C_6 (6-gon)



なにをやろうとしてるのか？

- 2人のプレーヤー・2色カードの利用
- 事前に与えられたグラフの「頂点」を選択して
  カードを使って入力 
- 2人の入力値の「連結性」または「距離」に
  応じてプロトコル出力が変化する
- 開示（出力）時に自明な状況を除いて
  何を入力したかを秘密にできる

CSEC2024年3月の貢献

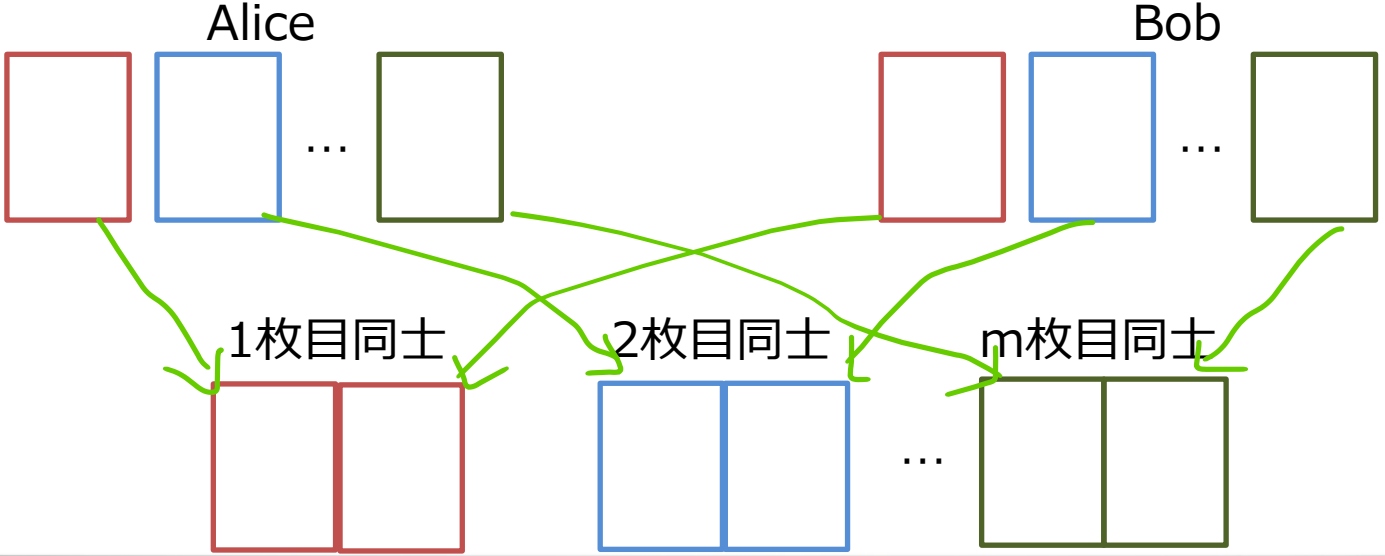
- 与えられた「デッキ」（＝カード束）からの所望の入力を行う際には**ランダムカット**のみを用いる（＝カード入力の「制限」）
- m-デッキ分割法でシャッフル（そこそこ軽い）
- １ユーザm枚２スートカード配布時の完全分類(m=4,5,6,7,8)

m-デッキ分割法

- 1ユーザのカード入力：m枚
- 1枚目同士，2枚目同士,…，m枚目同士の2枚を1つのデッキとして輪ゴムで止めてぶん投げる

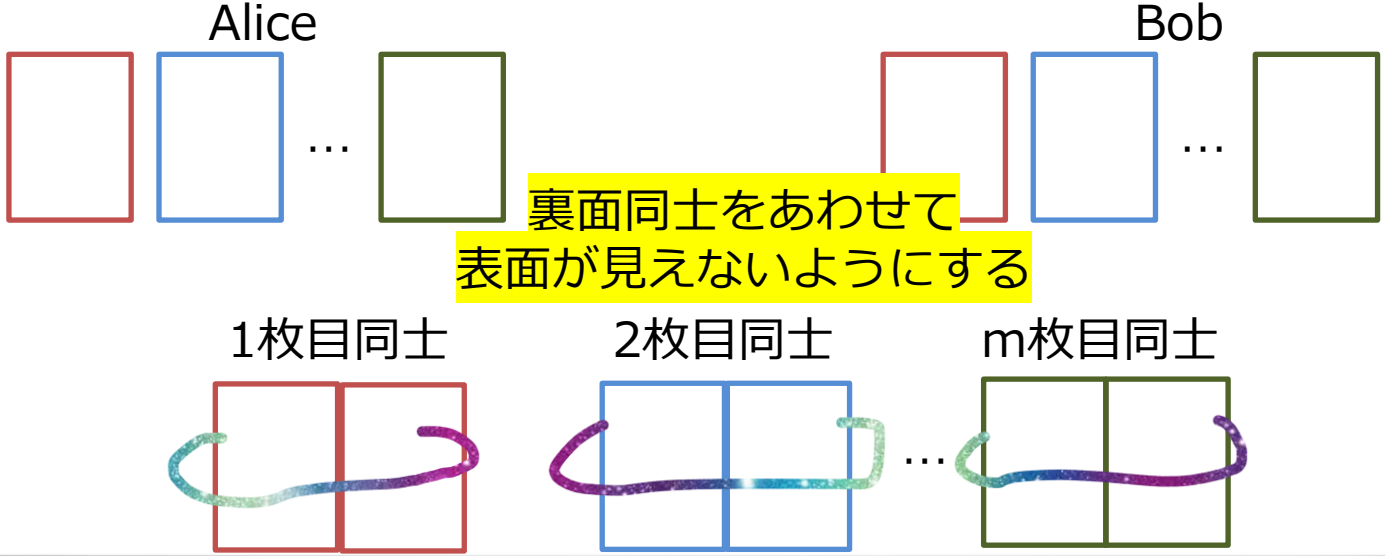
m-デッキ分割法

- 1ユーザのカード入力：m枚
- 1枚目同士，2枚目同士,…，m枚目同士の2枚を1つのデッキとして輪ゴムで止めてぶん投げる



m-デッキ分割法

- 1ユーザのカード入力：m枚
- 1枚目同士，2枚目同士,…，m枚目同士の2枚を1つのデッキとして輪ゴムで止めてぶん投げる



m-デッキ分割法でやっていること

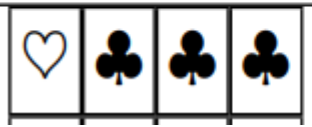
- **以下の3つのシャッフルを同時にやっている**
  - (1) パイルスクランブル（束の入れ替え）
  - (2) 上下シャッフル
  - (3) ランダムカット

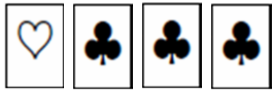
- **ポイント：各デッキは2枚しかない**
- **これが効率的かどうかは主観的だけど**

1ユーザ4枚のケース

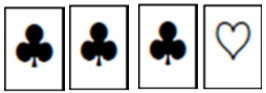
- 下記で網羅してて，あとは「同型」
- （全く定義してませんがスート入替も含めて2パターンしかない）

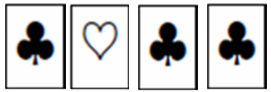
初期状態	距離分布	対応するグラフ
♡ ♣ ♣ ♣	$[0, 1, 1, 1]$	K_4
♡ ♡ ♣ ♣	$[0, 1, 2, 1]$	C_4

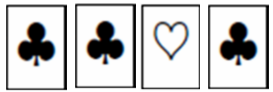
初期状態	距離分布	対応するグラフ
	$[0, 1, 1, 1]$	K_4



初期状態からランダムカットの操作で
入力できるのは4通り







実際には・・・

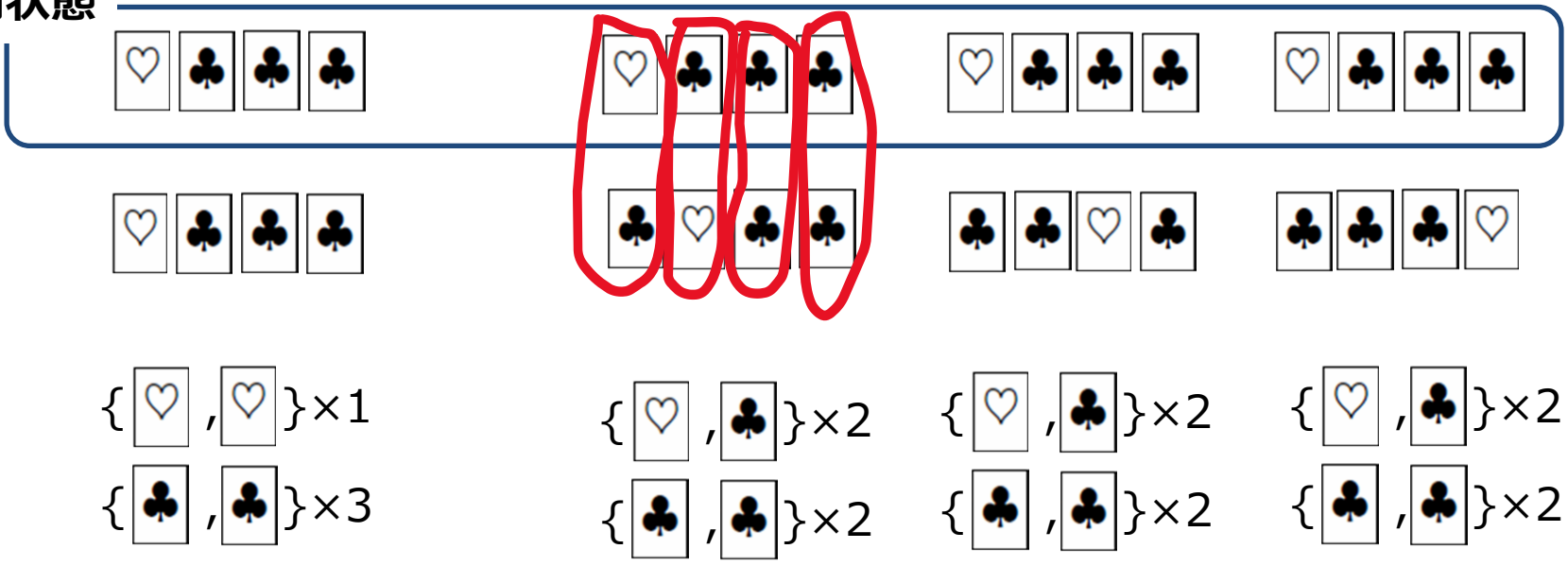
- 初期状態を表に向けて確認
- 裏にしてカード束（重なった状態に）
- 自分か相手かだれかがランダムカット

  （この状態でどこまでシフトしたかわからない）
- 入力者は第3者から分かるようにランダムカットして「所望の」入力にする

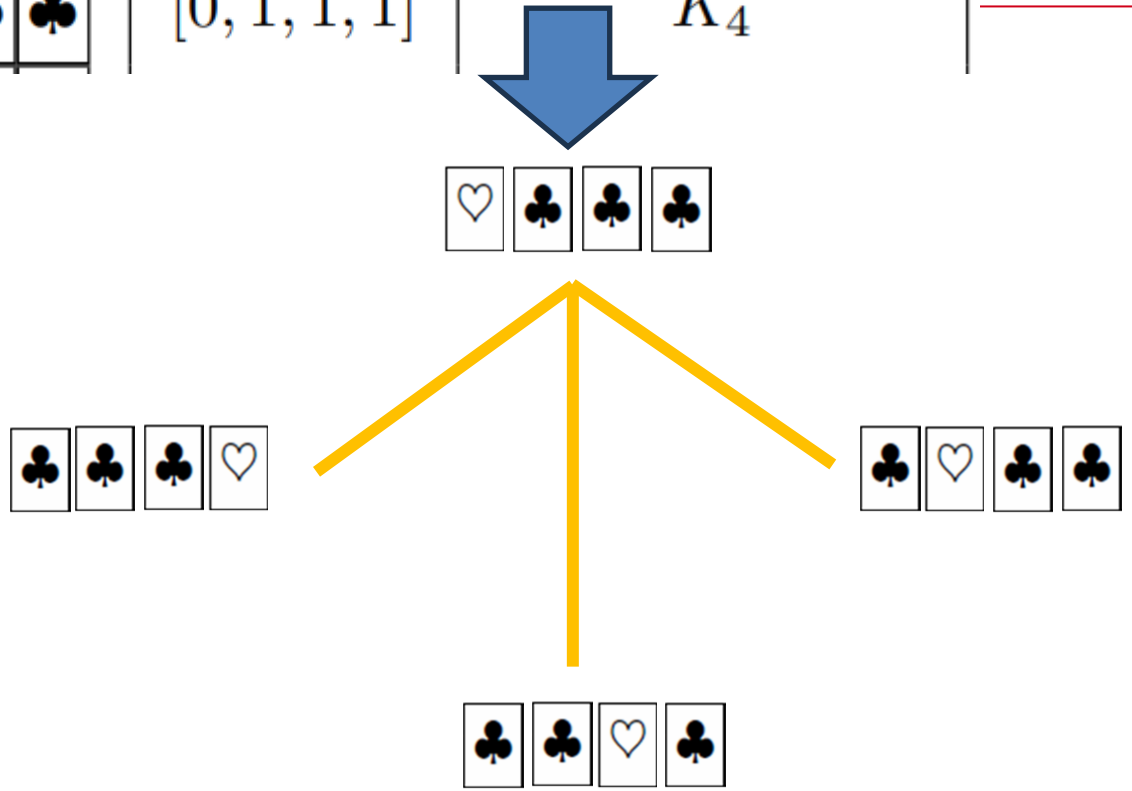
初期状態	距離分布	対応するグラフ
♡ ♣ ♣ ♣	$[0, 1, 1, 1]$	K_4

- **m-デッキ分割法(m=4)を適用した場合**

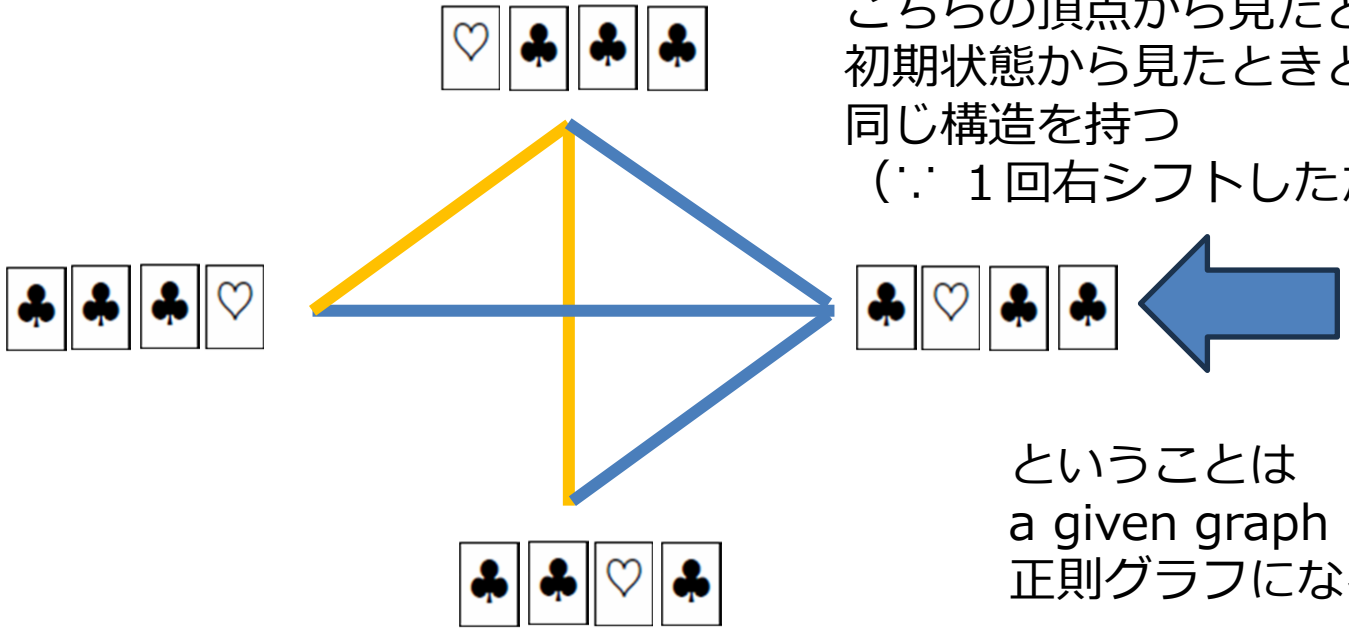
初期状態



初期状態	距離分布	対応するグラフ
♡ ♣ ♣ ♣	$[0, 1, 1, 1]$	K_4



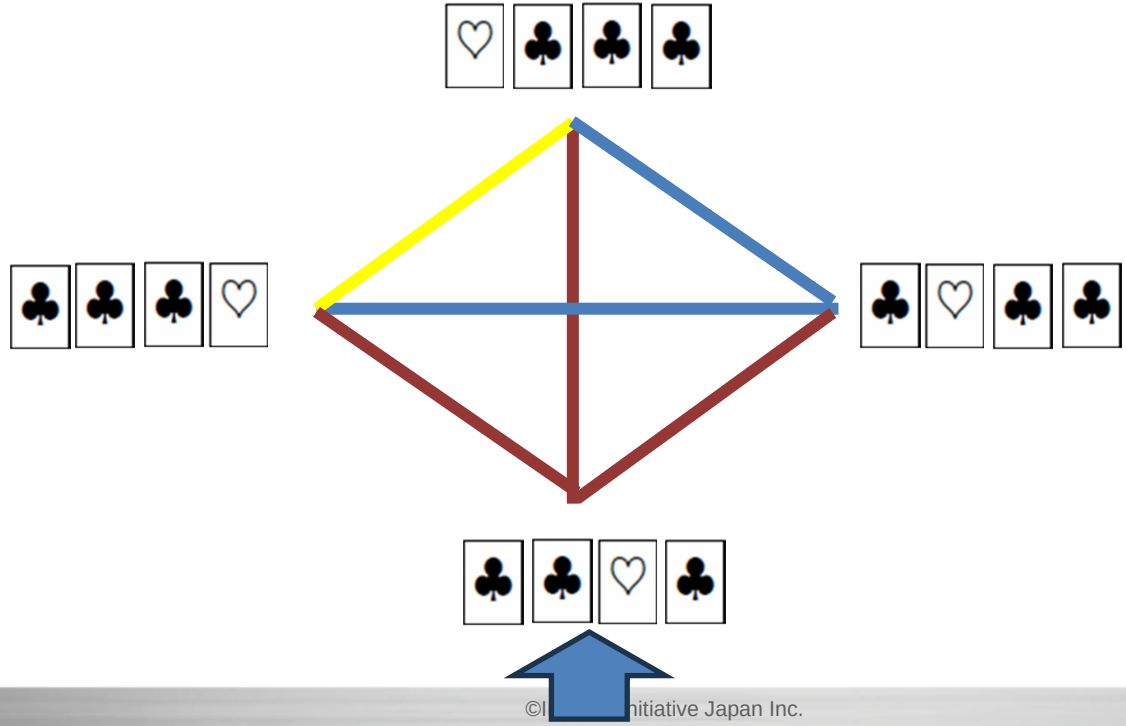
初期状態	距離分布	対応するグラフ
♡♣♣♣	$[0,1,1,1]$	K_4



こちらの頂点から見たときも
初期状態から見たときと
同じ構造を持つ
（∵ １回右シフトしただけ）

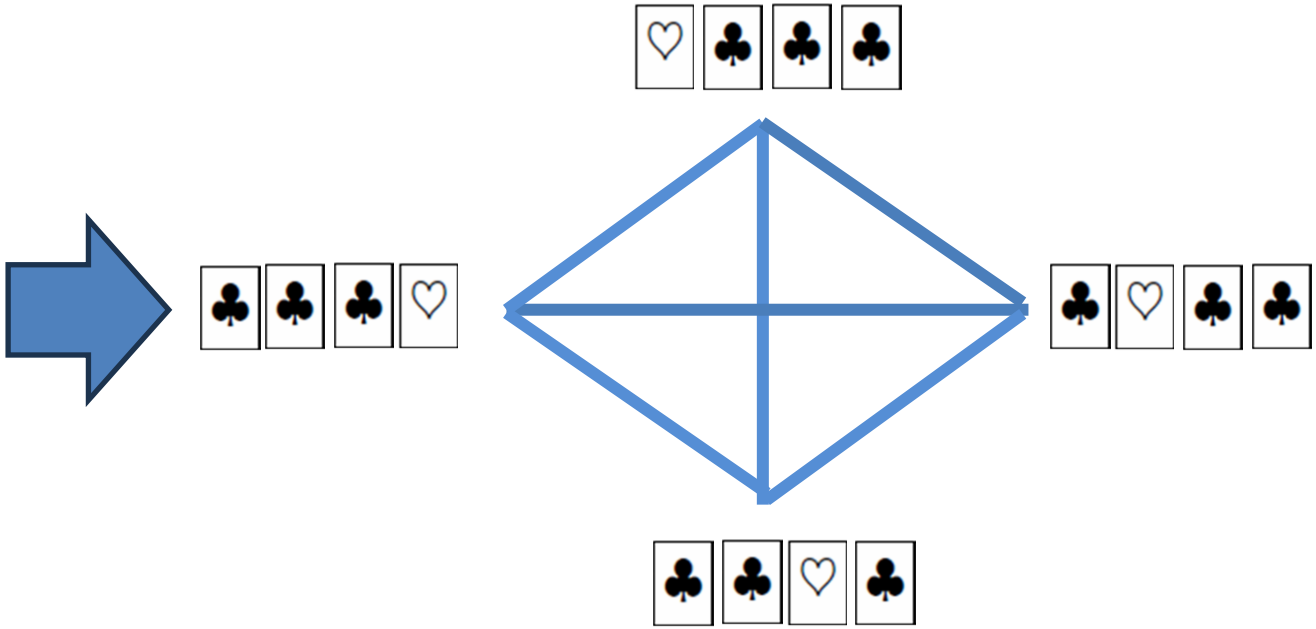
ということは
a given graph は
正則グラフになる

初期状態	距離分布	対応するグラフ
♡ ♣ ♣ ♣	$[0, 1, 1, 1]$	K_4

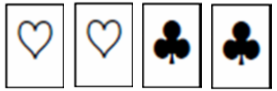


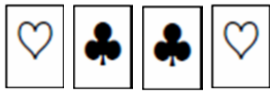
初期状態	距離分布	対応するグラフ
♡ ♣ ♣ ♣	$[0, 1, 1, 1]$	K_4

- **Remark：「一致関数」（完全グラフだから）**

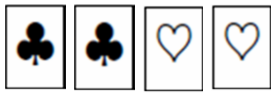


初期状態	距離分布	対応するグラフ
♡ ♡ ♣ ♣	$[0, 1, 2, 1]$	C_4





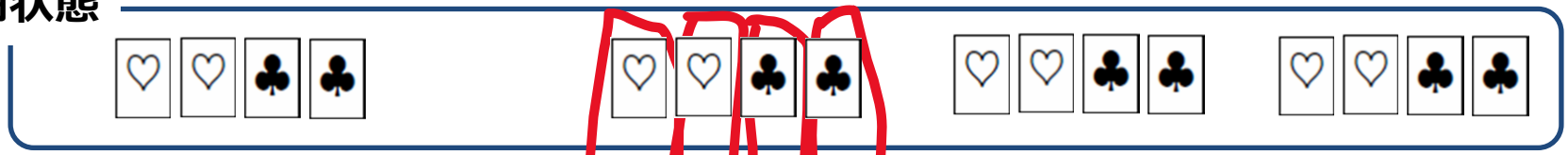
- **Remark：本来なら$_4C_2=6$通りの入力が可能なんだけど，敢えて4通りに「制限」する**



初期状態	距離分布	対応するグラフ
♡ ♡ ♣ ♣	$[0, 1, 2, 1]$	C_4

- **m-デッキ分割法(m=4)を適用した場合**

初期状態



{♡, ♡}×2

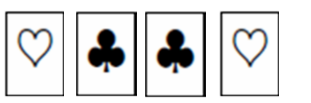
{♣, ♣}×2

{♡, ♣}×2

{♣, ♣}×1

{♡, ♡}×1

{♡, ♣}×4



{♡, ♣}×2

{♣, ♣}×2

{♡, ♡}×1

初期状態	距離分布	対応するグラフ
♡ ♡ ♣ ♣	$[0, 1, 2, 1]$	C_4

Aさん入力 ♡ ♡ ♣ ♣　♡ ♡ ♣ ♣　♡ ♡ ♣ ♣　♡ ♡ ♣ ♣

Bさん入力 ♡ ♡ ♣ ♣　♣ ♡ ♡ ♣　♣ ♣ ♡ ♡　♡ ♣ ♣ ♡

初期状態	距離分布	対応するグラフ
♡ ♡ ♣ ♣	$[0, 1, 2, 1]$	C_4

- **Johnson scheme J(v,k)の定義から**

♡の位置をk-部分集合のindexと同一視する

Aさん入力 ♡ ♡ ♣ ♣

Bさん入力 ♡ ♡ ♣ ♣

距離＝0

$\Omega = \{1, 2, \cdots, v\}$ とし k を $k \leq v/2$ なる自然数

X を Ω の k-部分集合全体の集合

$x, y \in X$ に対して $d(x, y) := k - |x \cap y|$

$g_i := \{(x, y) \mid d(x, y) = i\}$

$(X, \{g_i \mid i = 0, 1, \cdots, k\})$

初期状態	距離分布	対応するグラフ
♡ ♡ ♣ ♣	$[0, 1, 2, 1]$	C_4

- **Johnson scheme J(v,k)の定義から**

♡の位置をk-部分集合のindexと同一視する

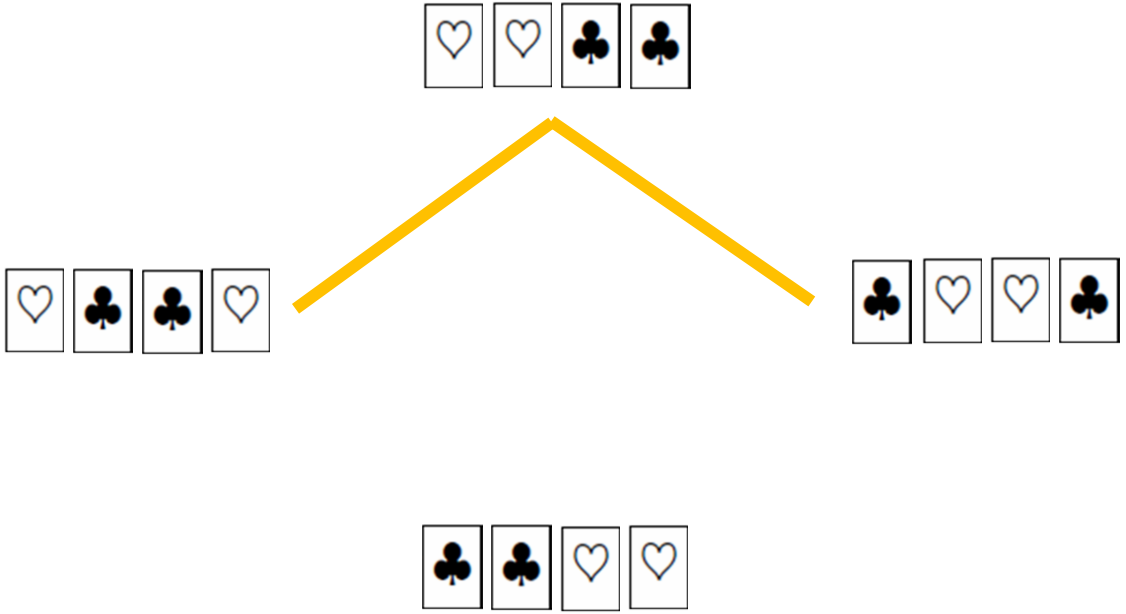
Aさん入力 ♡ ♡ ♣ ♣　　♡ ♡ ♣ ♣　　♡ ♡ ♣ ♣　　♡ ♡ ♣ ♣

Bさん入力 ♡ ♡ ♣ ♣　　♣ ♡ ♡ ♣　　♣ ♣ ♡ ♡　　♡ ♣ ♣ ♡

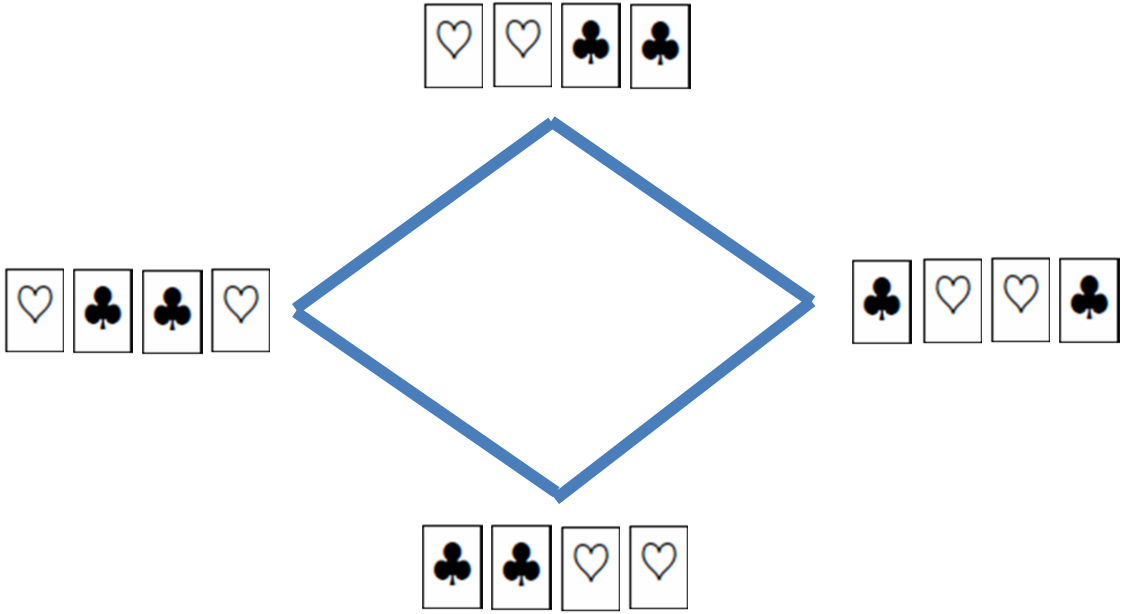
距離＝0　　距離＝1　　距離＝2　　距離＝1

$x, y \in X$ に対して $d(x, y) := k - |x \cap y|$

初期状態	距離分布	対応するグラフ
♡ ♡ ♣ ♣	$[0, 1, 2, 1]$	C_4

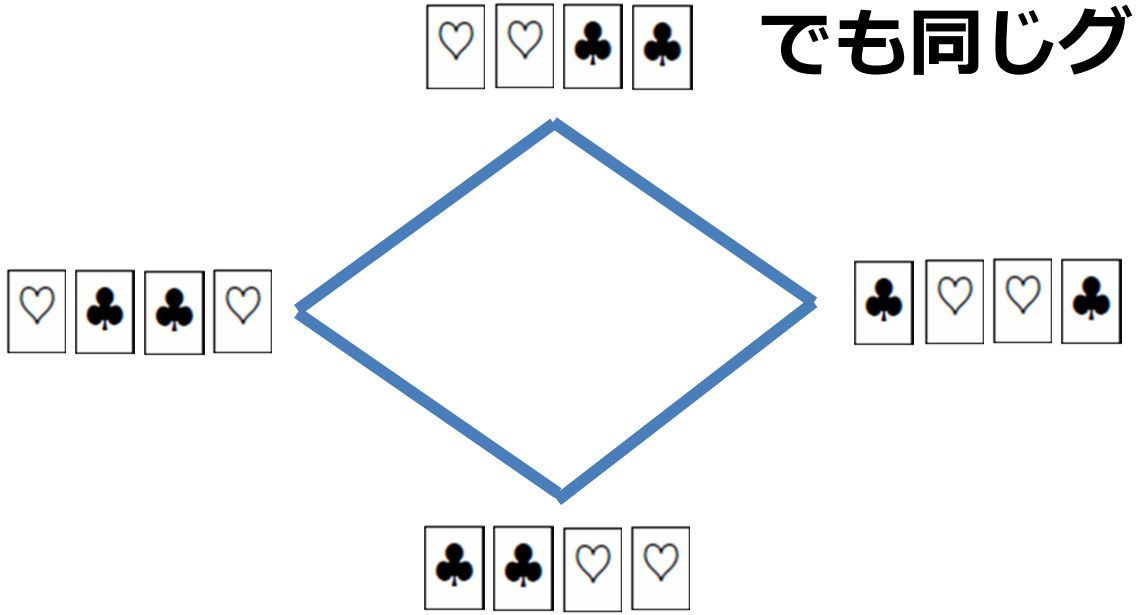


初期状態	距離分布	対応するグラフ
♡ ♡ ♣ ♣	$[0, 1, 2, 1]$	C_4



初期状態	距離分布	対応するグラフ
♡ ♡ ♣ ♣	$[0, 1, 2, 1]$	C_4

- **Remark：距離は高々2までなので「連結性」でも同じグラフになる**

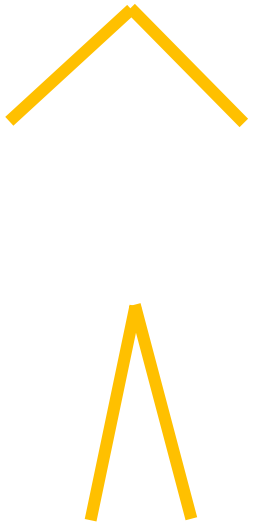


1ユーザ5枚のケース

初期状態	距離分布	対応するグラフ
♡ ♣ ♣ ♣ ♣	$[0,1,1,1,1]$	K_5
♡ ♡ ♣ ♣ ♣	$[0,1,2,2,1]$	C_5
♡ ♣ ♡ ♣ ♣	$[0,2,1,1,2]$	C_5

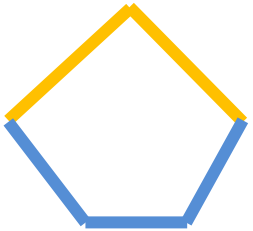
1ユーザ5枚のケース

初期状態	距離分布	対応するグラフ
♡ ♣ ♣ ♣ ♣	$[0,1,1,1,1]$	K_5
♡ ♡ ♣ ♣ ♣	$[0,1,2,2,1]$	C_5
♡ ♣ ♡ ♣ ♣	$[0,2,1,1,2]$	C_5



1ユーザ5枚のケース

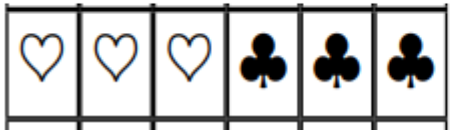
初期状態	距離分布	対応するグラフ
♡ ♣ ♣ ♣ ♣	$[0,1,1,1,1]$	K_5
♡ ♡ ♣ ♣ ♣	$[0,1,2,2,1]$	C_5
♡ ♣ ♡ ♣ ♣	$[0,2,1,1,2]$	C_5

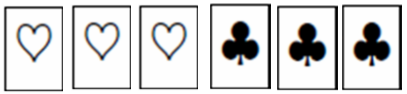


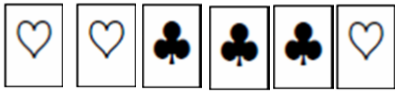
1ユーザ6枚のケース

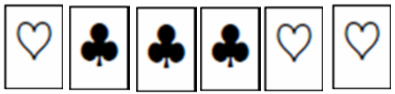
初期状態	距離分布	対応するグラフ
♡ ♣ ♣ ♣ ♣ ♣	$[0,1,1,1,1,1]$	K_6
♡ ♡ ♣ ♣ ♣ ♣	$[0,1,2,2,2,1]$	${C_6}^\sharp$
♡ ♣ ♡ ♣ ♣ ♣	$[0,2,1,2,1,2]$	$\overline{K_{3,3}}$
♡ ♡ ♡ ♣ ♣ ♣	$[0,1,2,3,2,1]$	C_6
♡ ♡ ♣ ♡ ♣ ♣	$[0,2,2,1,2,2]$	$\overline{K_{2,2,2}}$

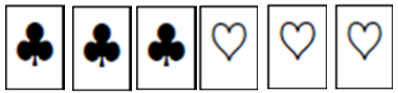
$C_6\#$: 距離に応じて出力が変化するのではなく
グラフ C_6 のノードの「隣接性」だけを見ている

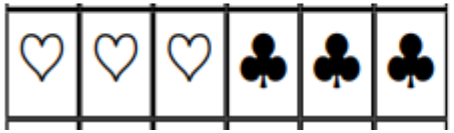
初期状態	距離分布	対応するグラフ
	$[0, 1, 2, 3, 2, 1]$	C_6

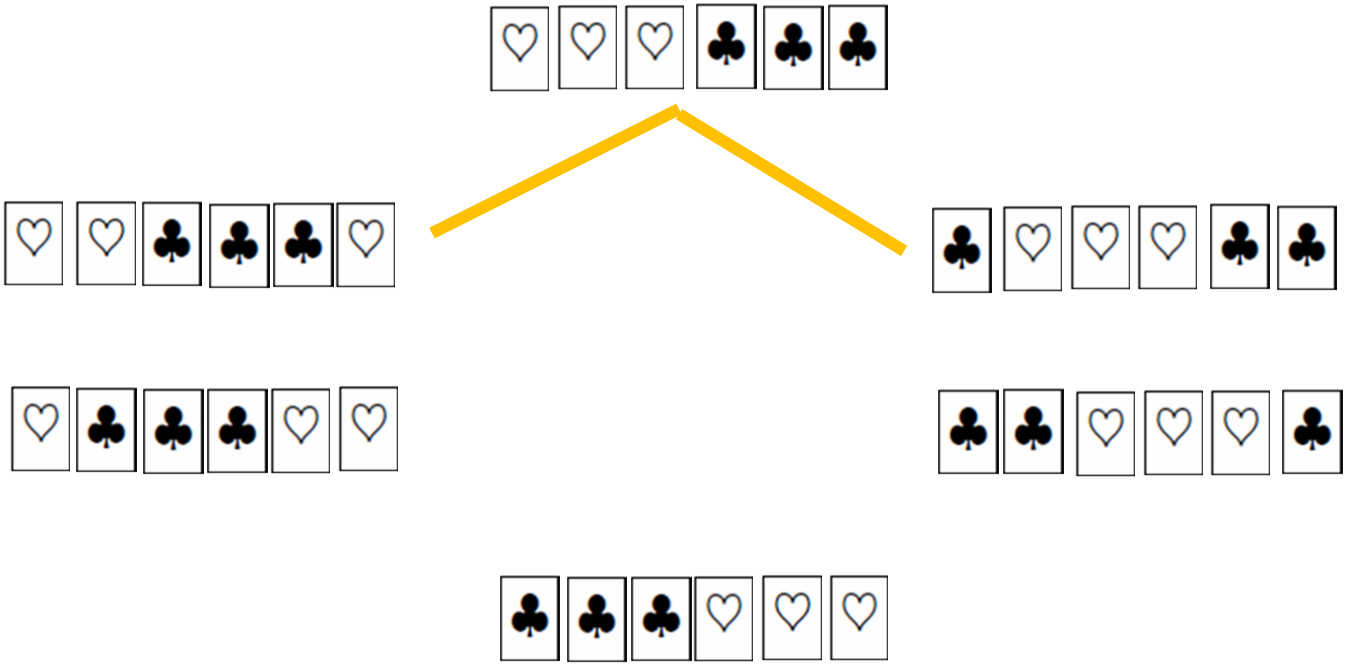


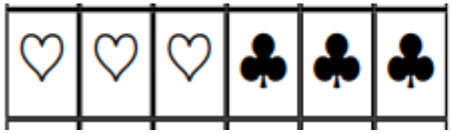


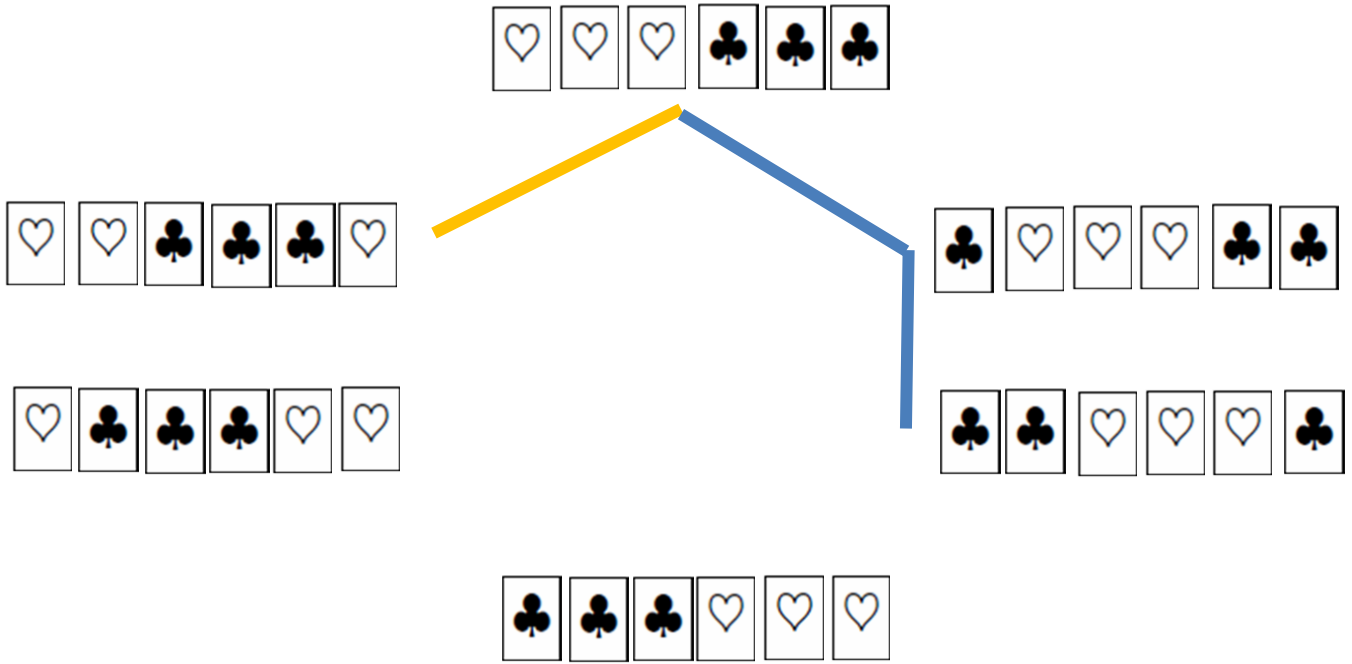




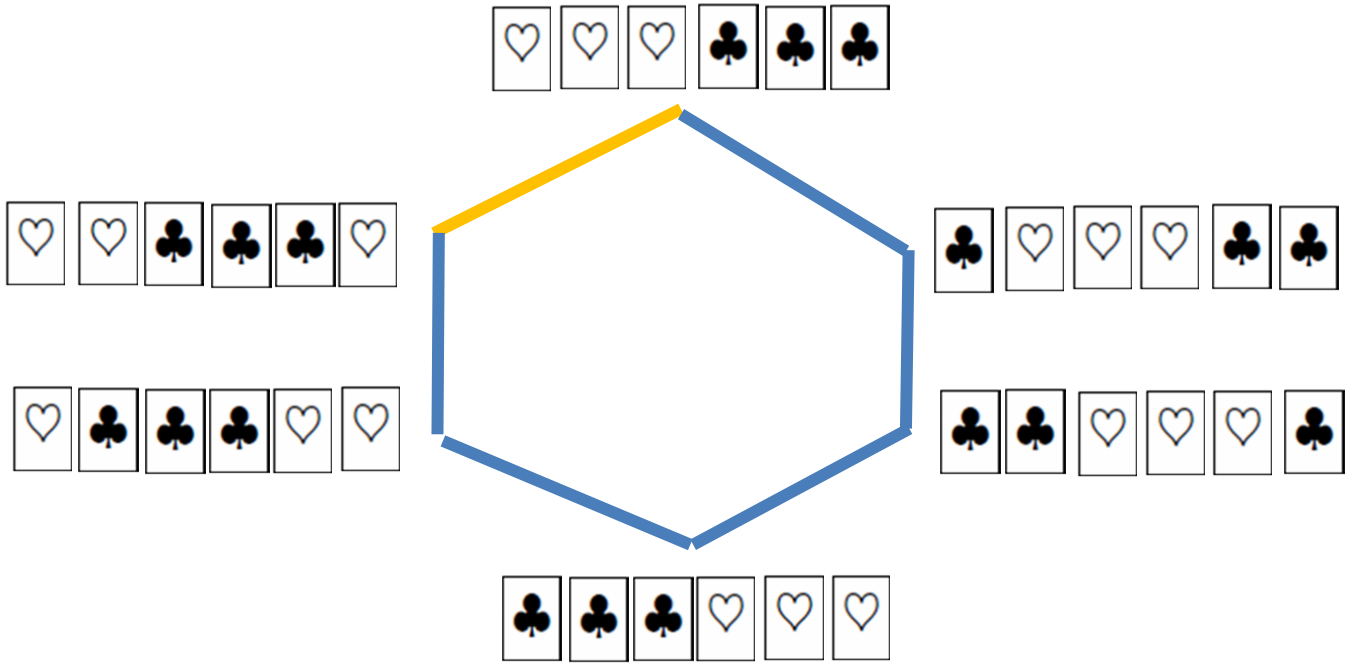
初期状態	距離分布	対応するグラフ
	$[0, 1, 2, 3, 2, 1]$	C_6

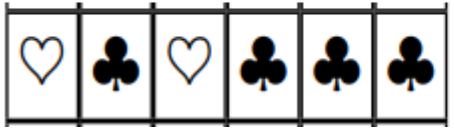


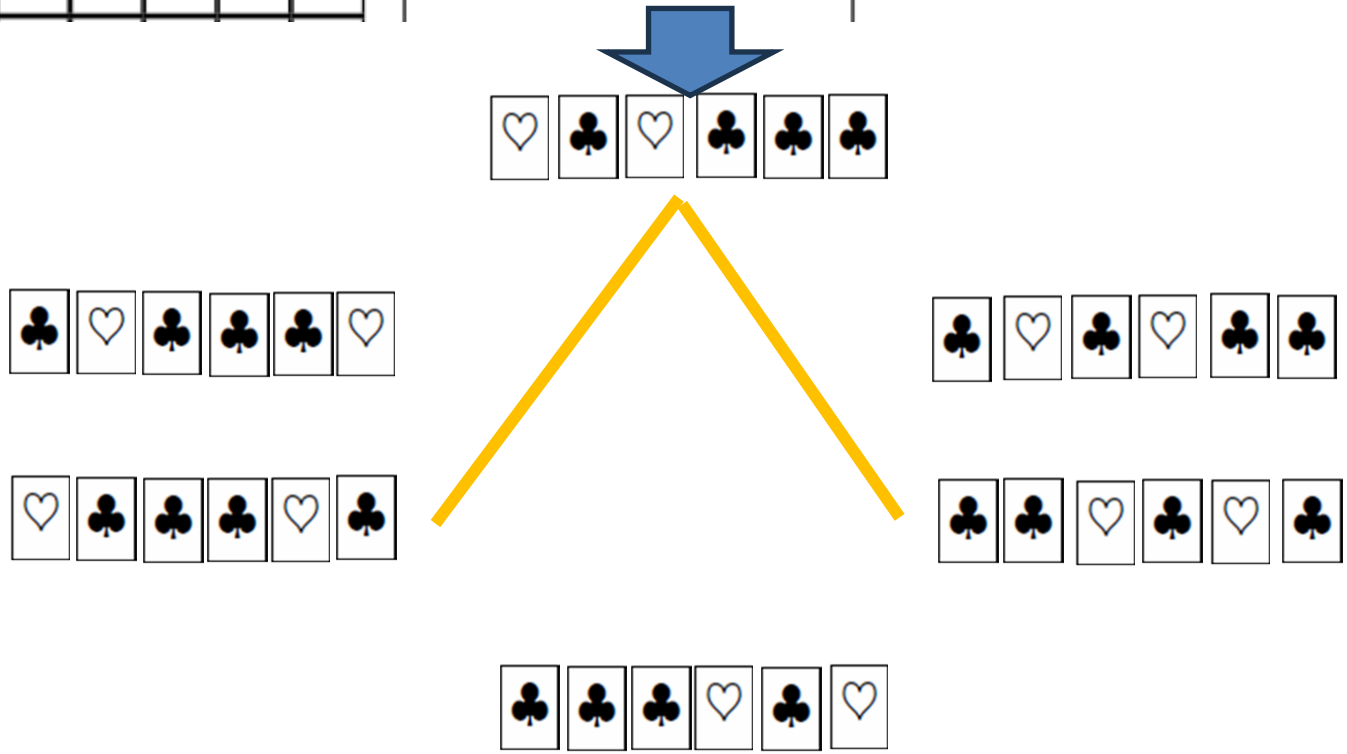
初期状態	距離分布	対応するグラフ
	$[0, 1, 2, 3, 2, 1]$	C_6



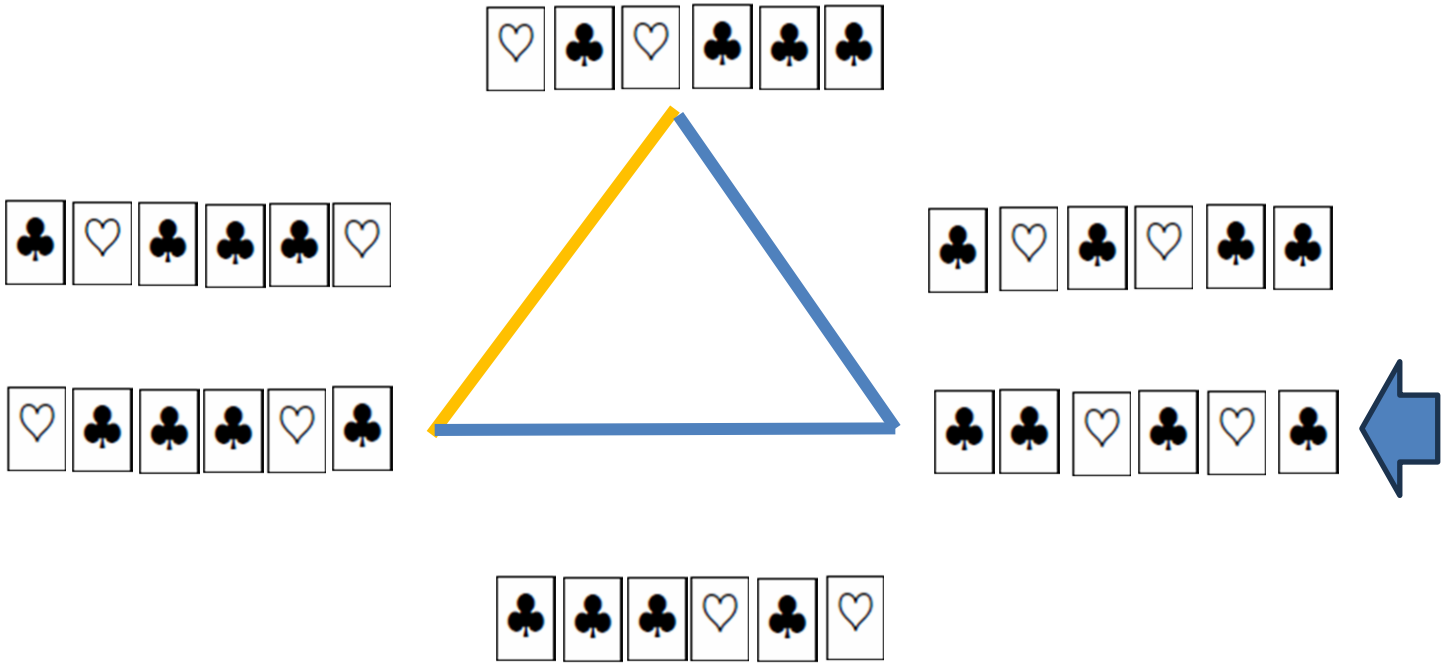
初期状態	距離分布	対応するグラフ
♡♡♡♣♣♣	$[0, 1, 2, 3, 2, 1]$	C_6



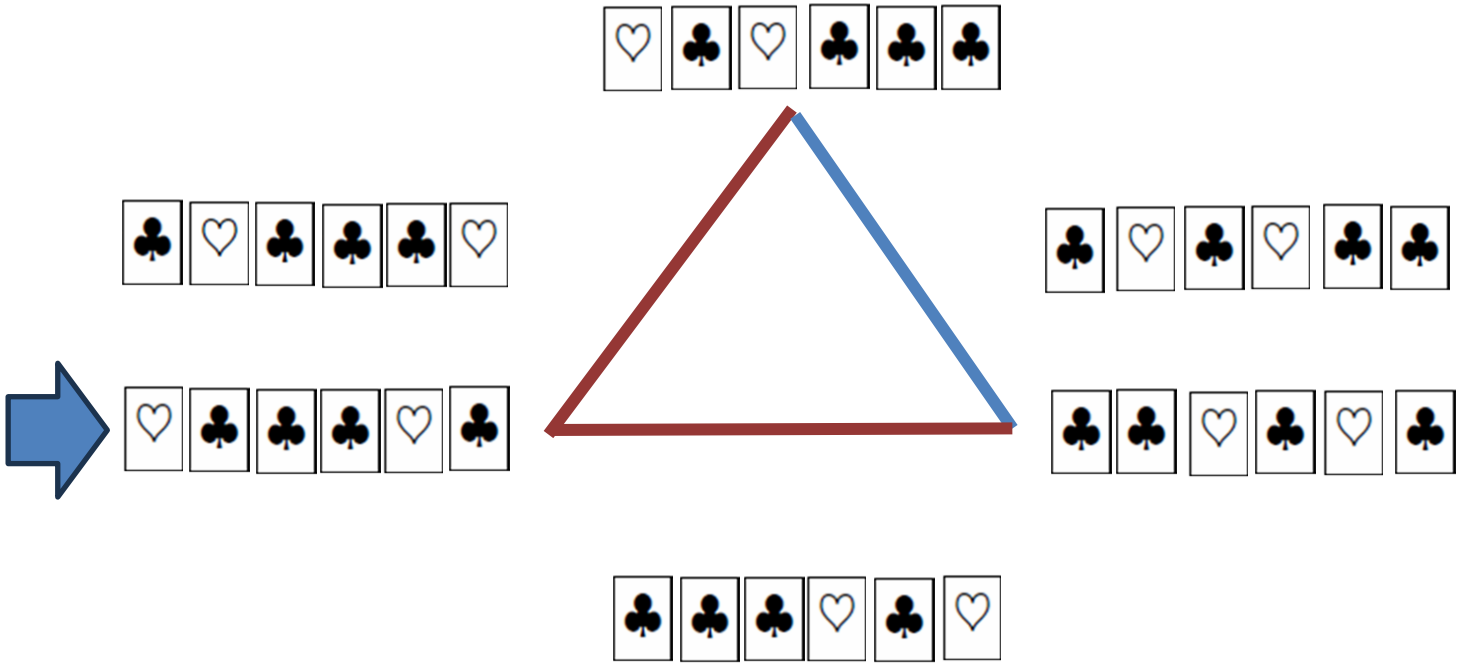
初期状態	距離分布	対応するグラフ
	$[0, 2, 1, 2, 1, 2]$	$\overline{K_{3,3}}$



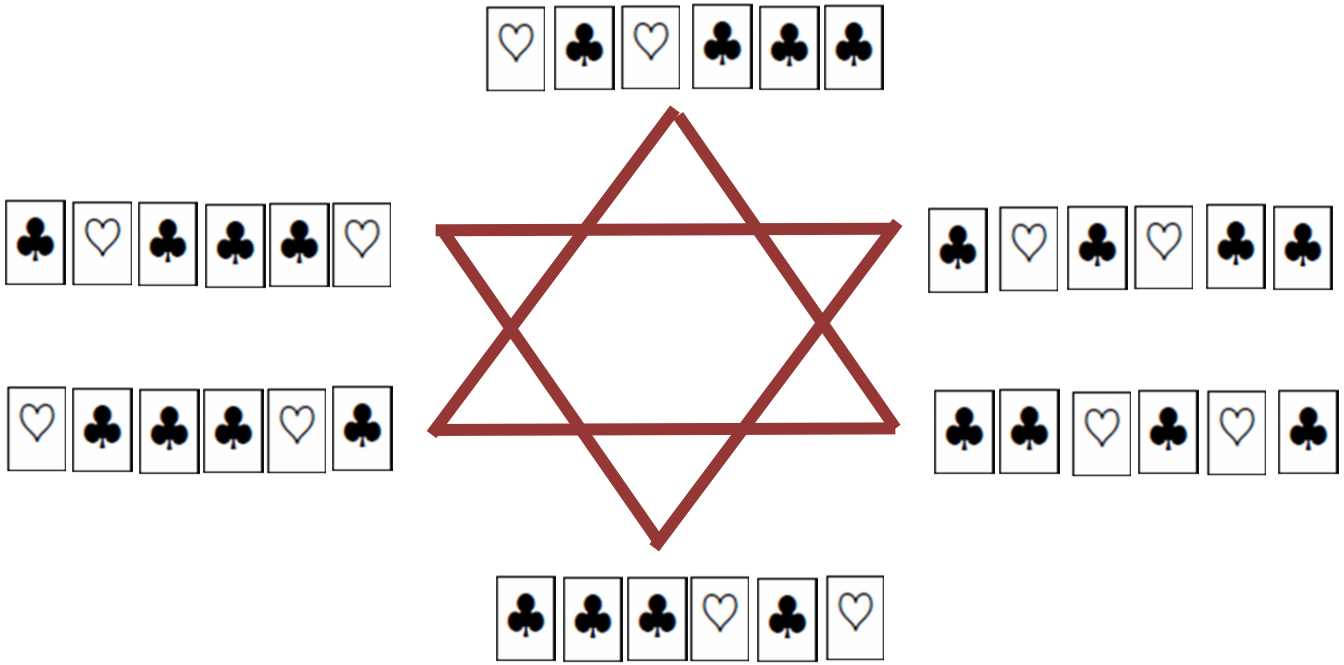
初期状態	距離分布	対応するグラフ
♡ ♣ ♡ ♣ ♣ ♣	$[0, 2, 1, 2, 1, 2]$	$\overline{K_{3,3}}$



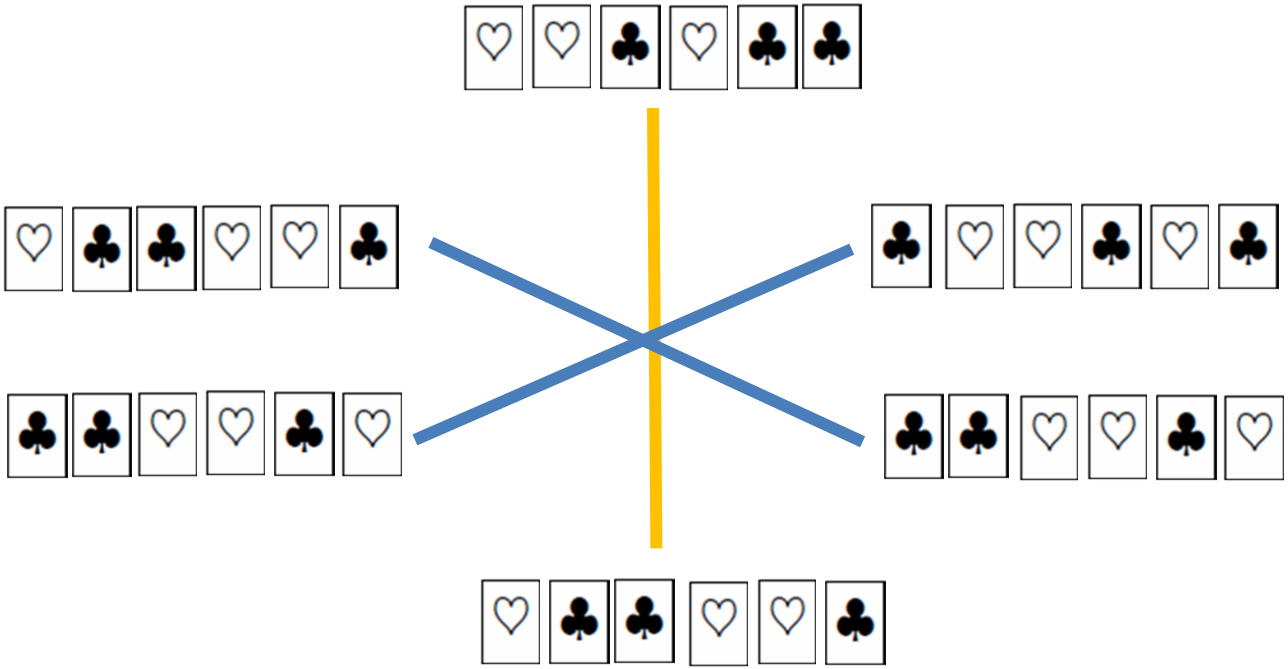
初期状態	距離分布	対応するグラフ
♡ ♣ ♡ ♣ ♣ ♣	$[0, 2, 1, 2, 1, 2]$	$\overline{K_{3,3}}$



初期状態	距離分布	対応するグラフ
♡ ♣ ♡ ♣ ♣ ♣	$[0, 2, 1, 2, 1, 2]$	$\overline{K_{3,3}}$



初期状態	距離分布	対応するグラフ
♡ ♡ ♣ ♡ ♣ ♣	$[0, 2, 2, 1, 2, 2]$	$\overline{K_{2,2,2}}$



（再掲）1ユーザ6枚のケース

初期状態	距離分布	対応するグラフ
♡ ♣ ♣ ♣ ♣ ♣	$[0,1,1,1,1,1]$	K_6
♡ ♡ ♣ ♣ ♣ ♣	$[0,1,2,2,2,1]$	${C_6}^\sharp$
♡ ♣ ♡ ♣ ♣ ♣	$[0,2,1,2,1,2]$	$\overline{K_{3,3}}$
♡ ♡ ♡ ♣ ♣ ♣	$[0,1,2,3,2,1]$	C_6
♡ ♡ ♣ ♡ ♣ ♣	$[0,2,2,1,2,2]$	$\overline{K_{2,2,2}}$

$C_6\#$: 距離に応じて出力が変化するのではなく
グラフ C_6 のノードの「隣接性」だけを見ている

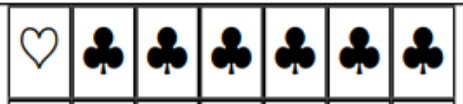
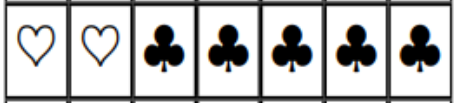
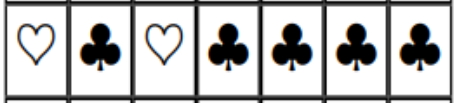
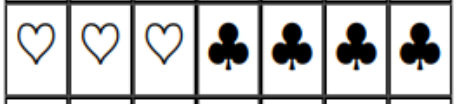
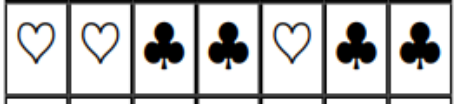
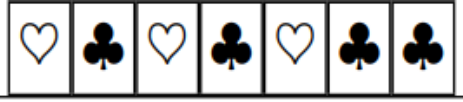
位数6でかつ対称なアソシエーションスキーム

- 対称：2人の入力の順番で結果が変わらない

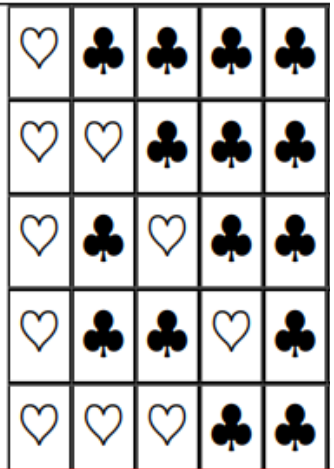
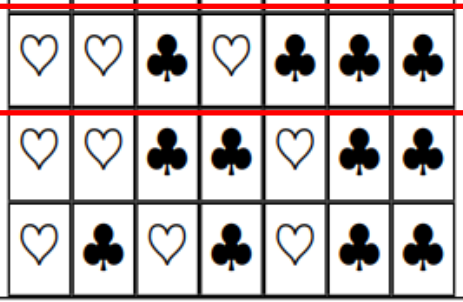
$$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 2 & 2 & 2 & 2 \\ 1 & 0 & 2 & 2 & 2 & 2 \\ 2 & 2 & 0 & 1 & 2 & 2 \\ 2 & 2 & 1 & 0 & 2 & 2 \\ 2 & 2 & 2 & 2 & 0 & 1 \\ 2 & 2 & 2 & 2 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 1 & 3 & 2 & 2 \\ 1 & 0 & 1 & 2 & 3 & 2 \\ 1 & 1 & 0 & 2 & 2 & 3 \\ 3 & 2 & 2 & 0 & 1 & 1 \\ 2 & 3 & 2 & 1 & 0 & 1 \\ 2 & 2 & 3 & 1 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 1 & 2 & 2 & 2 \\ 1 & 0 & 1 & 2 & 2 & 2 \\ 1 & 1 & 0 & 2 & 2 & 2 \\ 2 & 2 & 2 & 0 & 1 & 1 \\ 2 & 2 & 2 & 1 & 0 & 1 \\ 2 & 2 & 2 & 1 & 1 & 0 \end{bmatrix}$$

K_6 $\overline{K_{2,2,2}}$ 6-gon $\overline{K_{3,3}}$

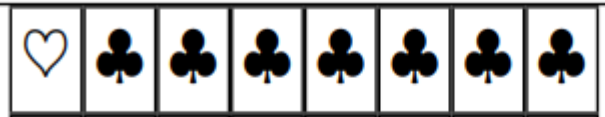
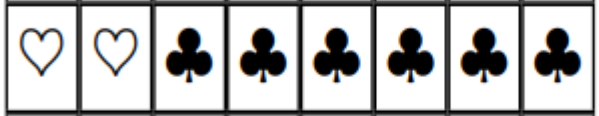
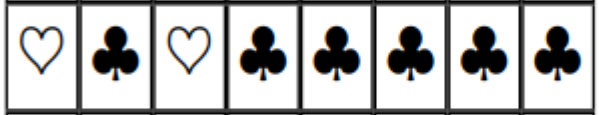
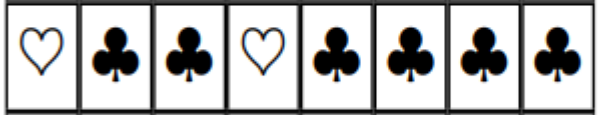
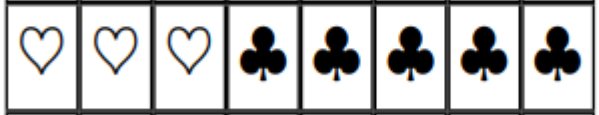
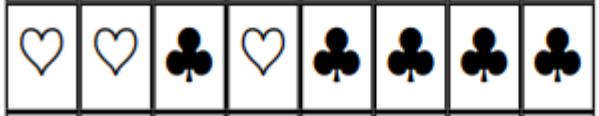
1ユーザ7枚のケース

初期状態	距離分布	対応するグラフ
	$[0,1,1,1,1,1,1]$	K_7
	$[0,1,2,2,2,2,1]$	${C_7}^\sharp$
	$[0,2,1,2,2,1,2]$	${C_7}^\sharp$
	$[0,2,2,1,1,2,2]$	${C_7}^\sharp$
	$[0,1,2,3,3,2,1]$	C_7
	$[0,2,2,2,2,2,2]$	K_7
	$[0,2,3,1,1,3,2]$	C_7
	$[0,3,1,2,2,1,3]$	C_7

1ユーザ7枚のケース

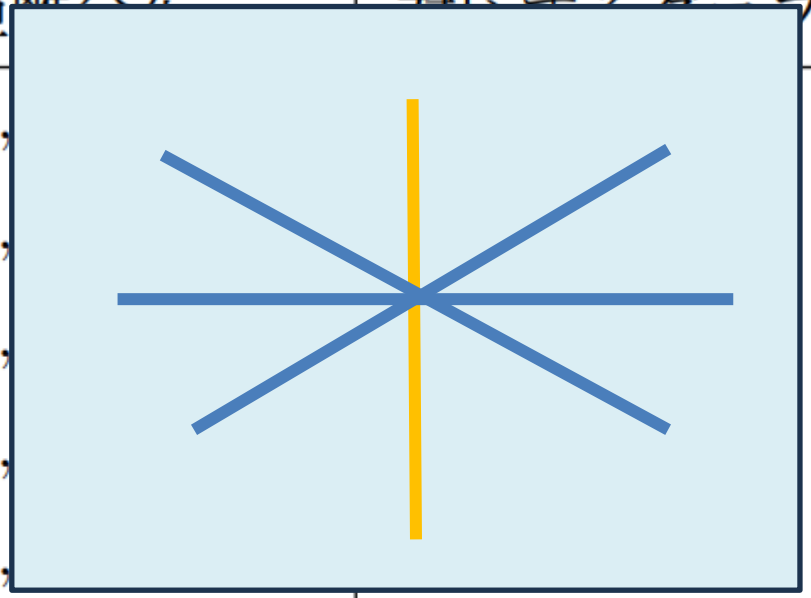
初期状態	距離分布	対応するグラフ
	(7,3,1)-difference set in Z_7 を用いて説明することができる．Difference set は位数 v の可換群 G 上で定義できる．(v,k,λ)-difference set in G は k-部分集合 D で零点でない g はちょうど λ 回 $x-y \in D$ が出現するように定義された概念である．定義より $\lambda(v-1)=k(k-1)$ を満たすことが知られている．	
	$[0,2,2,2,2,2,2]$	K_7
	$[0,2,3,1,1,3,2]$	C_7
	$[0,3,1,2,2,1,3]$	C_7

1ユーザ8枚のケース

初期状態	距離分布	対応するグラフ
	$[0,1,1,1,1,1,1,1]$	K_8
	$[0,1,2,2,2,2,2,1]$	${C_8}^\sharp$
	$[0,2,1,2,2,2,1,2]$	$\overline{K_{2,2}+K_{2,2}}$
	$[0,2,2,1,2,1,2,2]$	${C_8}^\sharp$
	$[0,1,2,3,3,3,2,1]$	
	$[0,2,2,2,3,2,2,2]$	$\overline{K_{2,2,2,2}}$
	$[0,2,3,2,1,2,3,2]$	

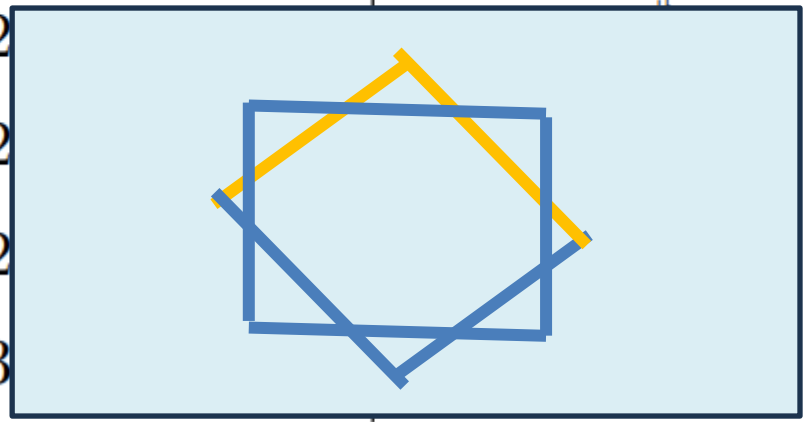
1ユーザ8枚のケース

初期状態	距離分布	対応するグラフ
♡ ♣ ♣ ♣ ♣ ♣ ♣ ♣	[0, 1, 1,	
♡ ♡ ♣ ♣ ♣ ♣ ♣ ♣	[0, 1, 2,	
♡ ♣ ♡ ♣ ♣ ♣ ♣ ♣	[0, 2, 1,	
♡ ♣ ♣ ♡ ♣ ♣ ♣ ♣	[0, 2, 2,	
♡ ♡ ♡ ♣ ♣ ♣ ♣ ♣	[0, 1, 2,	
♡ ♡ ♣ ♡ ♣ ♣ ♣ ♣	[0, 2, 2, 2, 3, 2, 2, 2]	$\overline{K_{2,2,2,2}}$
♡ ♡ ♣ ♣ ♡ ♣ ♣ ♣	[0, 2, 3, 2, 1, 2, 3, 2]	

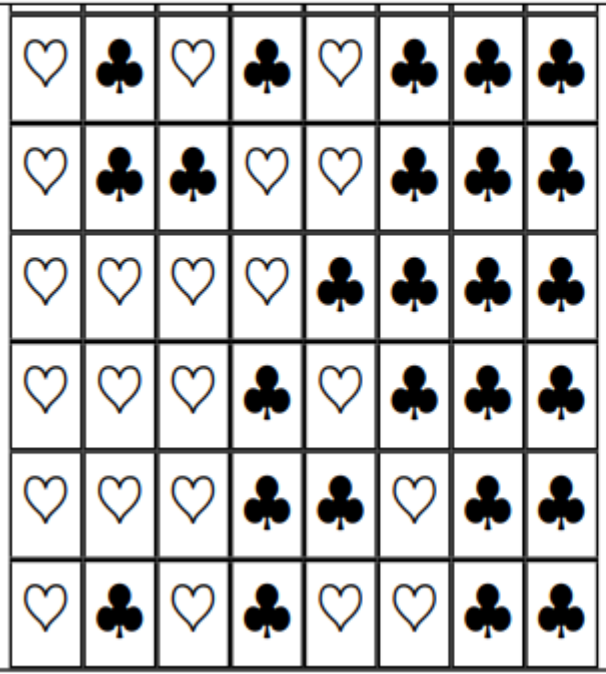


1ユーザ8枚のケース

初期状態	距離分布	対応するグラフ
♡ ♣ ♣ ♣ ♣ ♣ ♣ ♣	$[0,1,1,1,1,1,1,1]$	K_8
♡ ♡ ♣ ♣ ♣ ♣ ♣ ♣	$[0,1,2,2,2,2,2,1]$	$C_8{}^\sharp$
♡ ♣ ♡ ♣ ♣ ♣ ♣ ♣	$[0,2,1,2,2,2,1,2]$	$\overline{K_{2,2}+K_{2,2}}$
♡ ♣ ♣ ♡ ♣ ♣ ♣ ♣	$[0,2,2$	
♡ ♡ ♡ ♣ ♣ ♣ ♣ ♣	$[0,1,2$	
♡ ♡ ♣ ♡ ♣ ♣ ♣ ♣	$[0,2,2$	
♡ ♡ ♣ ♣ ♡ ♣ ♣ ♣	$[0,2,3$	

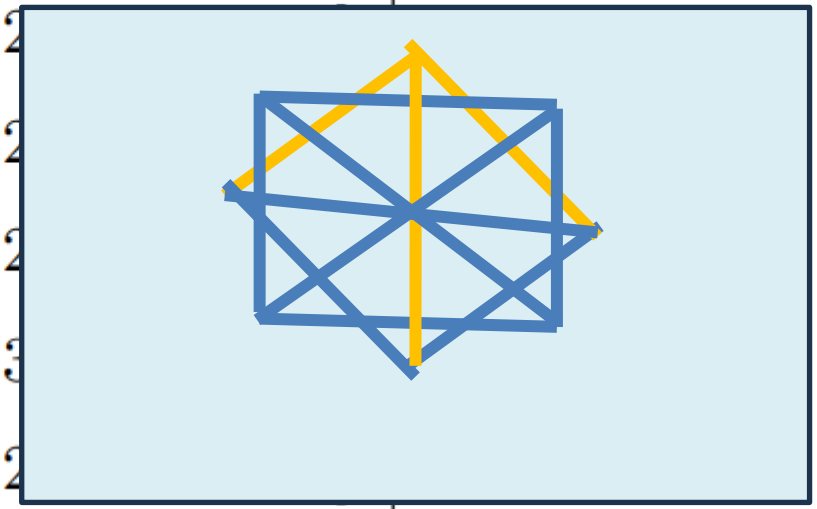


1ユーザ8枚のケース

初期状態	距離分布	対応するグラフ
	$[0,3,1,3,1,3,1,3]$	$\overline{K_{4,4}}$
	$[0,3,2,1,3,1,2,3]$	
	$[0,1,2,3,4,3,2,1]$	C_8
	$[0,2,2,3,2,3,2,2]$	${C_8}^\sharp$
	$[0,2,3,2,2,2,3,2]$	$\overline{K_{2,2}+K_{2,2}}$
	$[0,3,2,2,2,2,2,3]$	${C_8}^\sharp$

1ユーザ8枚のケース

初期状態	距離分布	対応するグラフ
♡ ♣ ♡ ♣ ♡ ♣ ♣ ♣	$[0,3,1,3,1,3,1,3]$	$\overline{K_{4,4}}$
♡ ♣ ♣ ♡ ♡ ♣ ♣ ♣	$[0,3,2$	
♡ ♡ ♡ ♡ ♣ ♣ ♣ ♣	$[0,1,2$	
♡ ♡ ♡ ♣ ♡ ♣ ♣ ♣	$[0,2,2$	
♡ ♡ ♡ ♣ ♣ ♡ ♣ ♣	$[0,2,3$	
♡ ♣ ♡ ♣ ♡ ♡ ♣ ♣	$[0,3,2$	

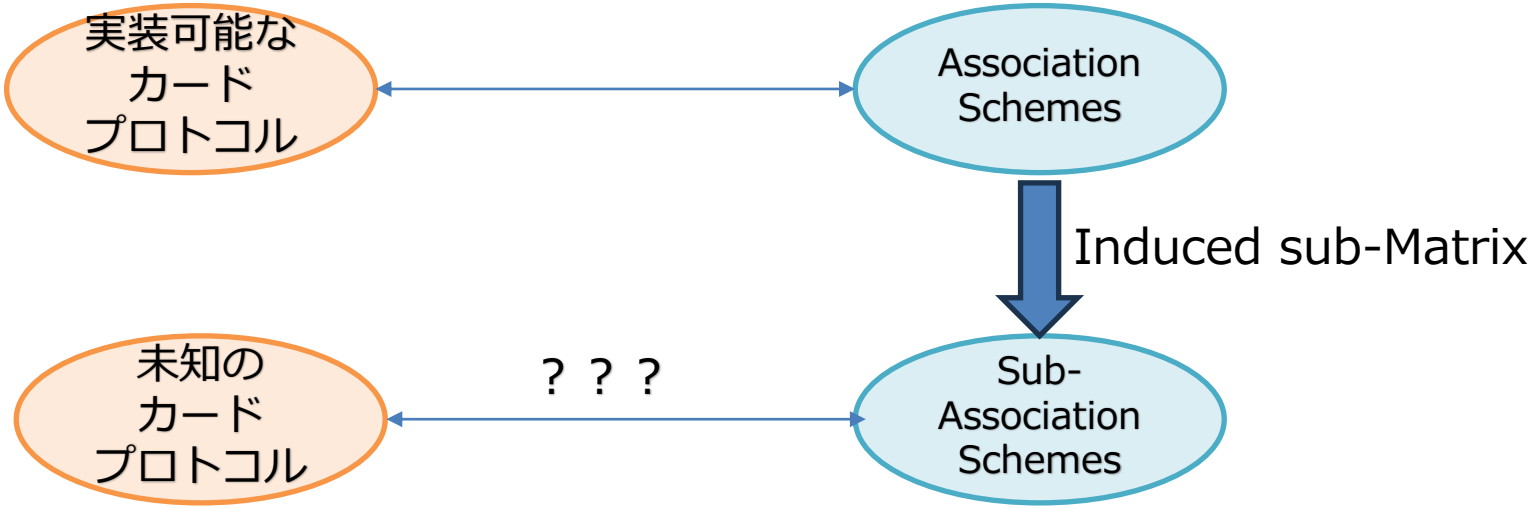


まとめに代えて

- デッキ分割法を適用することで現れる
  正則グラフの分類を紹介した
- アソシエーションスキーム以外の代数的構造は
  1ユーザ6枚配布時にはじめて出現する
- 7枚のとき面白い事例あり（K_7）
- 8枚もそこそこ面白い
- 9枚はかなり面白い（CSS2024）

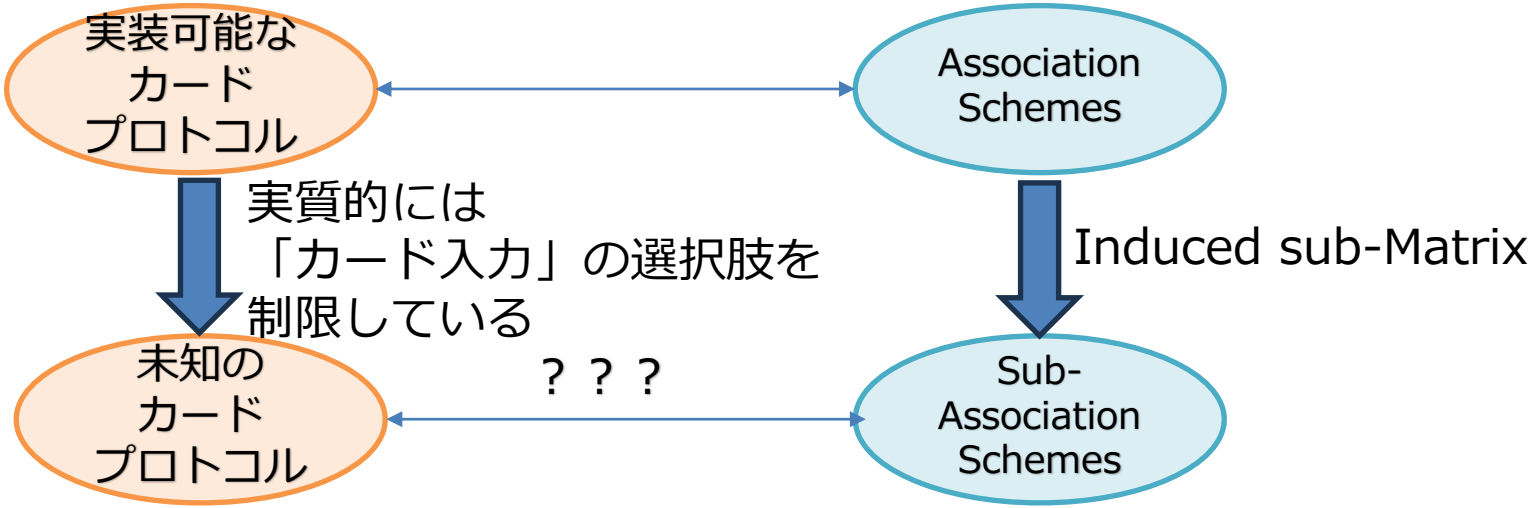
少し抽象度を上げた説明

- 構成できなかったカードプロトコルが
  実装できるかもしれない予感



少し抽象度を上げた説明

- 構成できなかったカードプロトコルが
  実装できるかもしれない予感



須賀からの未解決問題2023 (1/2)

- 2-party多値入力の一致関数の実現
  （ナイーブな手法については提示済）
- (Hamming schemes での構成事例を提示した上で)
  Johnson scheme J(v,k) に
  関連するカードプロトコルの構成
  （もしくは既存のカードプロトコルが
  Johnson schemeと関連している事例の例示）

→ 両者ともナイーブな方法以外は未解決のまま

須賀からの未解決問題2023 (2/2)

- 位数4,6のアソシエーションスキームと関連するカードプロトコルの構成→位数7まで解決
- 2枚入力 3-valued n-party 一致関数の構成
  →未着手（忘れてた）
- カード入力を制限する一般的な構成→RCのみ
- カード入力として1枚カードを追加することによる新しいプロトコルを構成できるが,
  この拡張方式に呼応する一般的拡張方式の提示
  →何言ってるかわかんない（思い出せない）

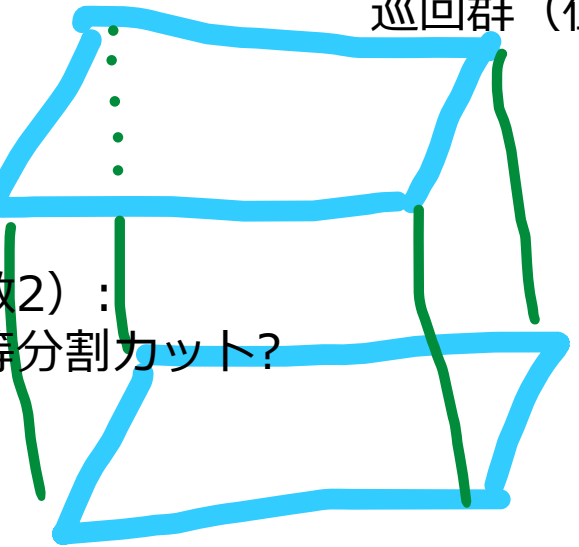
未解決問題2024

- 位数8以上のSymmetric association schemes に呼応するカードプロトコルが構成可能か？
- Johnson schemeに着想を得た「ランダムカットで入力制限」する以外のバリエーションではどのようなグラフが構成できるか？
- RCで制限されたカード入力の集合を巡回群の元と思えば，与えられた有限群に呼応するシャッフルが必ず存在するのか？

気になっていること（位数8）

- H(2,3) = 立方体のような構造 を分解すると

巡回群（位数4）：4枚のランダムカット

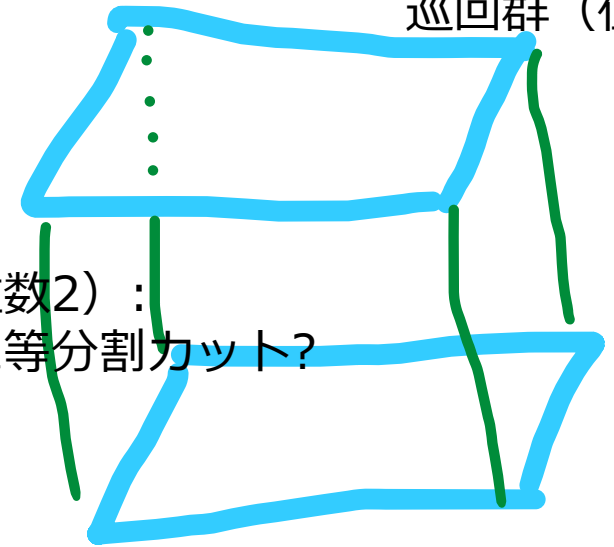


巡回群（位数2）：
ランダム二等分割カット?

[<1,2,3,4> | <5,6,7,8>]
={
{1,2,3,4, 5,6,7,8}, {5,6,7,8, 1,2,3,4},
{2,3,4,1, 6,7,8,5}, {6,7,8,5, 2,3,4,1},
{3,4,1,2, 7,8,5,6}, {7,8,5,6, 3,4,1,2},
{4,1,2,3, 8,5,6,7}, {8,5,6,7, 4,1,2,3},
}

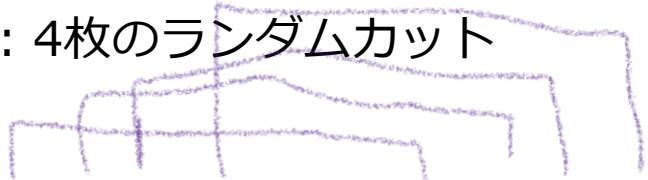
気になっていること（位数8）

- H(2,3) = 立方体のような構造 を分解すると



巡回群（位数2）：
ランダム二等分割カット?

巡回群（位数4）：4枚のランダムカット



[<1,2,3,4> | <5,6,7,8>]

={
{1,2,3,4, 5,6,7,8}, {5,6,7,8, 1,2,3,4},
{2,3,4,1, 6,7,8,5}, {6,7,8,5, 2,3,4,1},
{3,4,1,2, 7,8,5,6}, {7,8,5,6, 3,4,1,2},
{4,1,2,3, 8,5,6,7}, {8,5,6,7, 4,1,2,3},
}

おそらく$Z_2 \times Z_4$ の群構造

- よくわからんくなってますが $Z_2 \times Z_2 \times Z_2$

{
{1,2,3,4, 5,6,7,8}, {5,6,7,8, 1,2,3,4},
{2,1,4,3, 6,5,8,7}, {6,5,8,7, 2,1,4,3},
{3,4,1,2, 7,8,5,6}, {7,8,5,6, 3,4,1,2},
{4,3,2,1, 8,7,6,5}, {8,7,6,5, 4,3,2,1},
}

[[<1,2>|<3,4>] | [<5,6>|<7,8>]]

Ongoing Innovation

IIJ Internet Initiative Japan

Stay safe and healthy