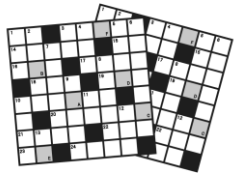
Sumpleteに対する物理的ゼロ知識証明

初貝　恭祐（電気通信大学）

パズルに対するゼロ知識証明



Prover

Verifier

解答があると納得

- Prover
  - 解答を知っている
- Verifier
  - 解答を知らない
  - 解答を計算できない
- 目的
  - パズルには解答が存在することを Verifier に納得させる
  - Verifier に解答を教えない

物理的ゼロ知識証明

⇒ 計算機ではなくカードなどを使って行う

Sumplete とは？

➢問題

目標値

3	5	0	−6	
−3	1	−2	3	−1
5	−7	6	2	8
9	4	3	−8	5
−6	3	−4	−1	−10

➢解答

3	5	0	−6	
✖	1	−2	✖	−1
✖	✖	6	2	8
9	4	✖	−8	5
−6	✖	−4	✖	−10

➢ChatGPTが生成（2023年3月）[1]

➢問題例：4 × 4 サイズの盤面

- マス内に整数
- 各行・列にも整数（目標値）が割り当て

➢解答：整数を削除するマスの位置

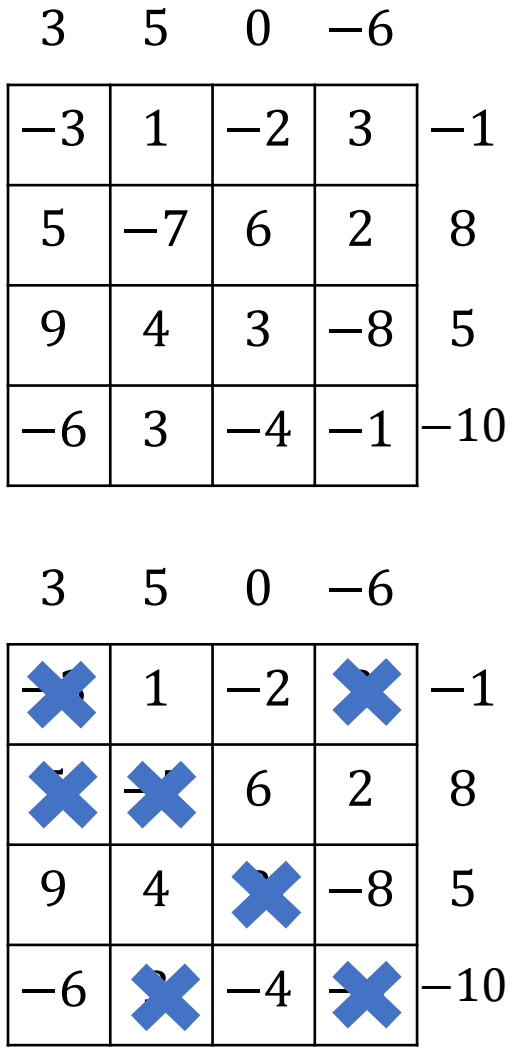
＜条件＞

各行・各列について，

残ったマスの和 ＝ 目標値

[1] P. Penguin: “Chatgpt invented its own puzzle game.”, https:// puzzledpenguin.substack.com/p/chatgpt-invented-its-ownpuzzle-game (2023).

本研究の成果



① Sumplete の NP 完全性の証明
- マスの数字に制限のないSumplete
- マスの数字に制限のあるSumplete

② Sumplete に対する物理的ゼロ知識証明の提案
- 実用性に即した手法
- 定義に即した手法

Sumplete の NP 完全性 – 証明方針

NP完全問題

変換

3	5	0	−6	
−3	1	−2	3	−1
5	−7	6	2	8
9	4	3	−8	5
−6	3	−4	−1	−10

- Sumplete はNP問題

- NP 完全問題から Sumplete へ
  多項式時間で変換可能

⇒ Sumplete は変換前の問題と同程度の難しさ

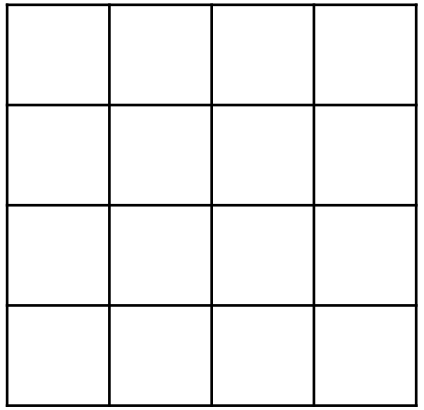
⇒ Sumplete は NP 完全

部分和問題 ⇒ Sumplete の帰着

➢部分和問題

■問題：$A = \{-3,1,-2,3\}, N = -1$
■解答：Yes $(A' = \{1,-2\})$

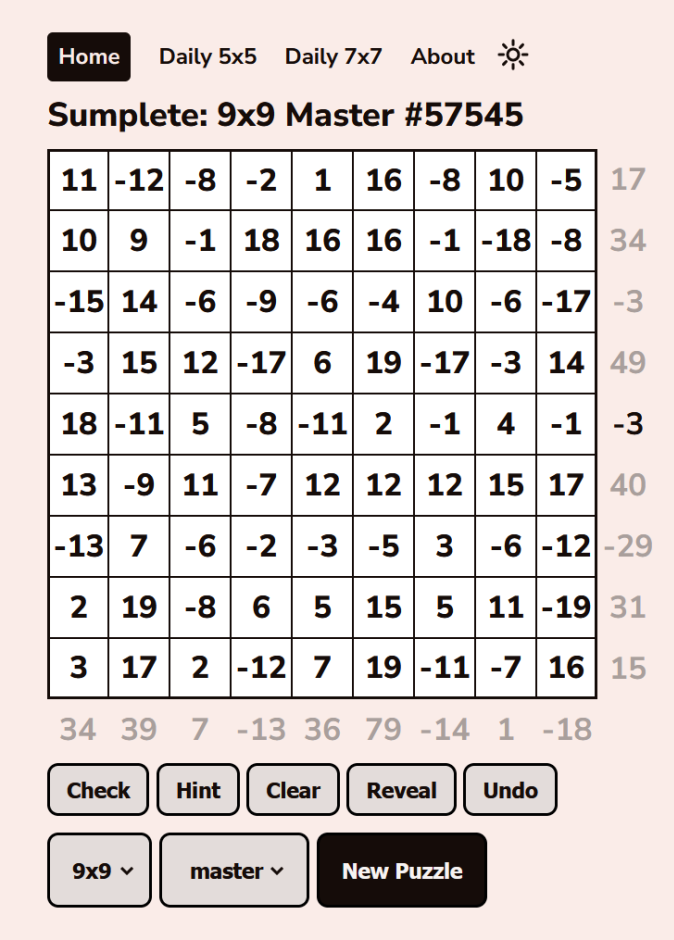
➢Sumplete



制限付き Sumplete の NP完全性

➢オリジナルの Sumplete：マスの数字は
　-20 以上 20 以下
⇒ マスの数字に制限があっても
　Sumplete はNP完全

➢部分和問題から証明できない
　□制限付き部分和問題はNP完全ではない
⇒ 特殊な充足可能性問題から証明



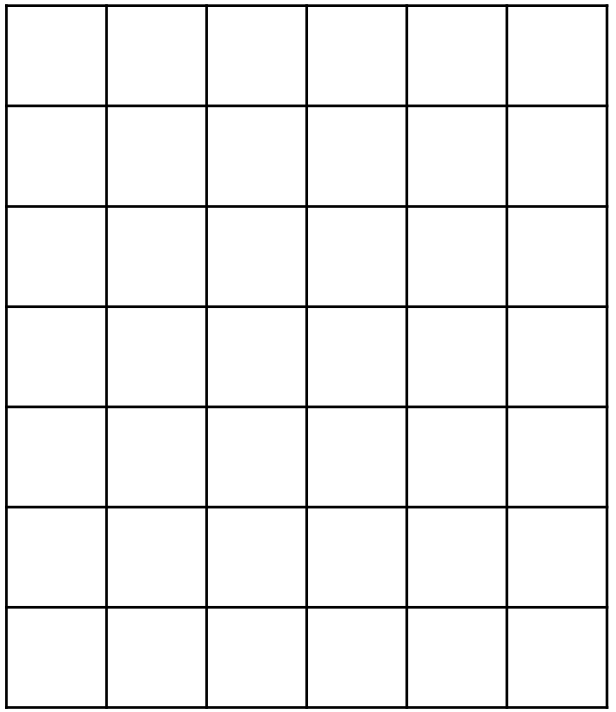
引用：https://sumplete.com

XSAT for 3-CNF_{+}^{3} ⇒ 制限付きSumplete の帰着

- XSAT for 3-CNF_{+}^{3}
  - 各 c に TRUE が 1 個だけ現れる
    変数割り当ては存在するか？
  - 各 c は3変数の論理和
  - 各変数は 3 回ずつ現れる

$$c_1 = x_1 \vee x_2 \vee x_3$$
$$c_2 = x_2 \vee x_3 \vee x_6$$
$$c_3 = x_1 \vee x_4 \vee x_6$$
$$c_4 = x_2 \vee x_5 \vee x_6$$
$$c_5 = x_1 \vee x_4 \vee x_5$$
$$c_6 = x_3 \vee x_4 \vee x_5$$

- 制限付きSumplete
  - $1 \le i \le 6$行目：j 列目 $= \begin{cases} 1 & (x_j \text{ in } c_i) \\ 3 & o/w \end{cases}$
  - 7行目：全て 3



XSAT for 3-CNF_{+}^{3} と (1,3)-Sumplete の解答の対応

➢XSAT for 3-CNF_{+}^{3}

- 各 c に TRUE が 1 個だけ現れる
  変数割り当ては存在するか？
- 各 c は3変数の論理和
- 各変数は 3 回ずつ現れる

$$c_1 = x_1 \vee x_2 \vee x_3$$
$$c_2 = x_2 \vee x_3 \vee x_6$$
$$c_3 = x_1 \vee x_4 \vee x_6$$
$$c_4 = x_2 \vee x_5 \vee x_6$$
$$c_5 = x_1 \vee x_4 \vee x_5$$
$$c_6 = x_3 \vee x_4 \vee x_5$$

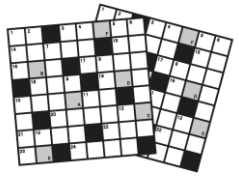
$x_1 = \mathrm{FALSE}$
$x_2 = \mathrm{TRUE}$
$x_3 = \mathrm{FALSE}$
$x_4 = \mathrm{TRUE}$
$x_5 = \mathrm{FALSE}$
$x_6 = \mathrm{FALSE}$

➢制限付きSumplete

- $1 \leq i \leq 6$行目：j 列目 $= \begin{cases} \text{残す} & (x_j = \mathrm{TRUE}) \\ \text{削除} & o/w \end{cases}$
- 7行目：j 列目 $= \begin{cases} \text{残す} & (x_j = \mathrm{FALSE}) \\ \text{削除} & o/w \end{cases}$

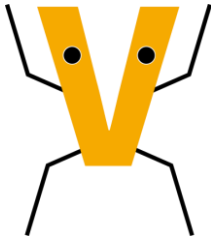
3	3	3	3	3	3	
✗	1	✗	✗	✗	✗	1
✗	1	✗	✗	✗	✗	1
✗	✗	✗	1	✗	✗	1
✗	1	✗	✗	✗	✗	1
✗	✗	✗	1	✗	✗	1
✗	✗	✗	1	✗	✗	1
3	✗	3	✗	3	3	12

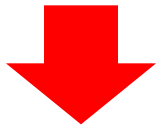
パズルに対するゼロ知識証明



Prover

Verifier





解答があると納得

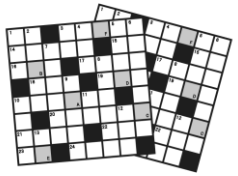
- Prover
  - 解答を知っている
- Verifier
  - 解答を知らない
  - 解答を計算できない

- 目的
  - パズルには解答が存在することを Verifier に納得させる
  - Verifier に解答を教えない

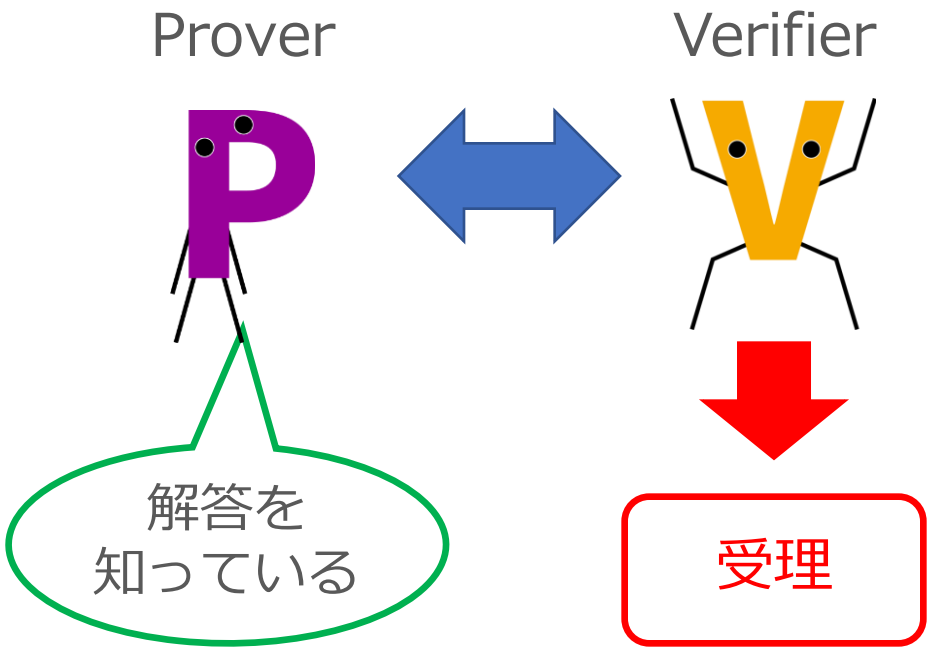
物理的ゼロ知識証明

⇒ 計算機ではなくカードなどを使って行う

パズルに対するゼロ知識証明 – 完全性

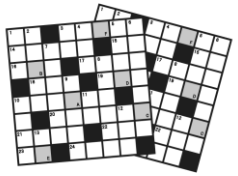


：解答あり

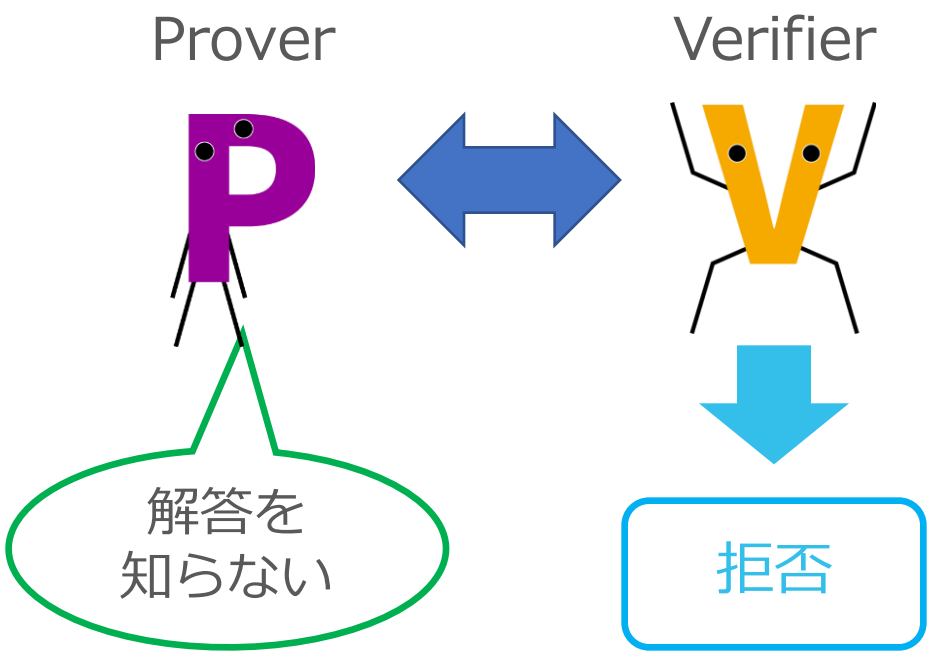


完全性 ($completeness$)
Prover は解答を知っている
⇒ Verifier は必ず受理

パズルに対するゼロ知識証明 – 健全性



：解答なし



健全性 ($soundness$)

Prover は解答を知らない
⇒ Verifier は無視できない確率で拒否

- Prover が解答を知らないにもかかわらず
  Verifier が受理する確率(健全性誤り確率)を p として,

$$0 \leq p < 1$$

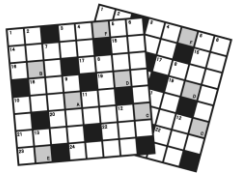
であれば，繰り返し実行することで

全体の健全性誤り確率は 0 に近づく.

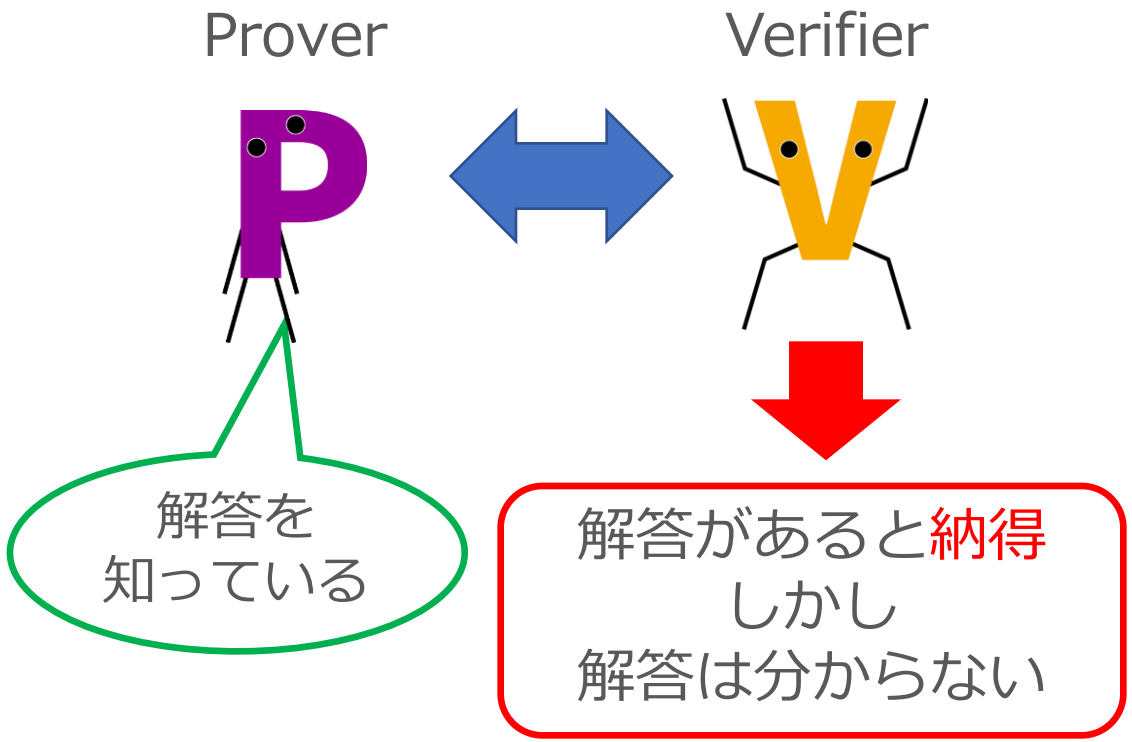
- 物理的ゼロ知識証明：人の手で行う

→ 人の手で繰り返し行うのは非効率的であるから
$p = 0$ が望ましい

パズルに対するゼロ知識証明 – ゼロ知識性



：解答あり



ゼロ知識性 ($zero\text{-}knowledge$)

Verifier は, 解答があること以上の
知識を得られない.

Sumplete のゼロ知識証明

➢問題

3	5	0	−6	
−3	1	−2	3	−1
5	−7	6	2	8
9	4	3	−8	5
−6	3	−4	−1	−10

➢解答

3	5	0	−6	
✖	1	−2	✖	−1
✖	✖	6	2	8
9	4	✖	−8	5
−6	✖	−4	✖	−10

Prover

対話

Verifier

目的

・Prover は Verifier に，次を納得させる
「各行・各列について，
残ったマスの和＝目標値
を満たすようなマスの消し方がある」

・Verifier に，削除するマスの位置は教えない

Sumplete のゼロ知識証明

➢問題

3	5	0	−6	
−3	1	−2	3	−1
5	−7	6	2	8
9	4	3	−8	5
−6	3	−4	−1	−10

➢解答

3	5	0	−6	
✖	1	−2	✖	−1
✖	✖	6	2	8
9	4	✖	−8	5
−6	✖	−4	✖	−10

Prover

対話

Verifier

目的

・Prover は Verifier に，次を納得させる
「各行・各列について，
残ったマスの和＝目標値
を満たすようなマスの消し方がある」

・Verifier に，削除するマスの位置は教えない

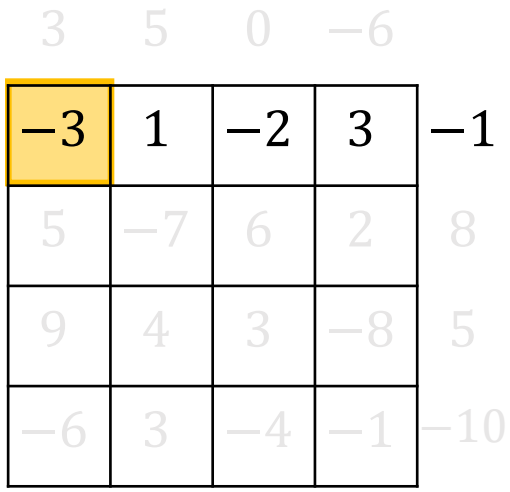
ゼロ知識性を考えない証明

	3	5	0	−6
−3	1	−2	3	−1
5	−7	6	2	8
9	4	3	−8	5
−6	3	−4	−1	−10

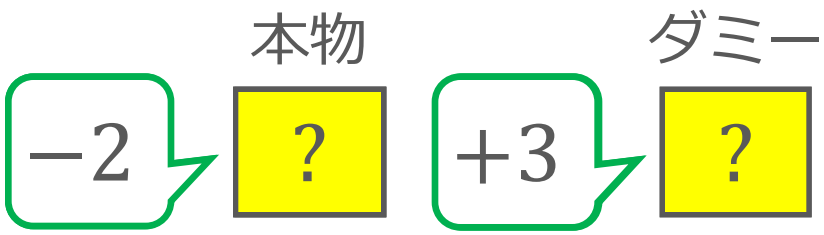
$$-1$$

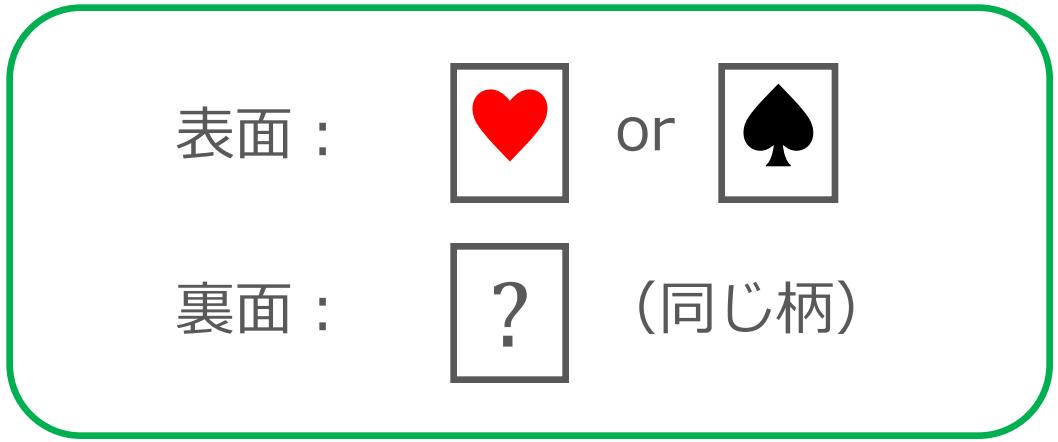
- 任意の行・列で，次を検証
  - 残すマスの和を計算
  - 目標値と比較
- 残すか削除するかを教えずに，
  残すマスだけを加算したい

解答を秘匿するアイディア



- 解答（削除するマス）を隠すために
  - ① カウンターの切り替え：削除マスはダミーに加算
    - ■ どちらのカウンターに加算するかは秘密にする
  - ② 値を隠して加算：最後に本物だけ公開
- このギミックを2種類のカードで実現する





ゼロ知識証明プロトコルの流れ

3	5	0	−6	
−3	1	−2	3	−1
5	−7	6	2	8
9	4	3	−8	5
−6	3	−4	−1	−10

- すべての行/列で検証
  - a. 残すマス ⇒ 本物カウンターに加算
    削除マス ⇒ ダミーカウンターに加算
  - b. 本物カウンター $\stackrel{?}{=}$ 目標値

課題

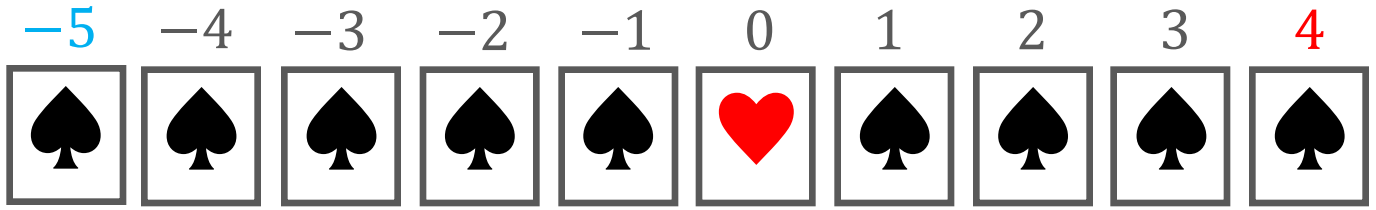
①カウンターの作り方
- 値を隠したまま加算

②カウンターの切り替え方
- どのマスを削除するかは隠す

カウンターの作り方

3	5	0	−6	
−3	1	−2	3	−1
5	−7	6	2	8
9	4	3	−8	5
−6	3	−4	−1	−10

- 注目する行：マスの和の最小値が -5，最大値が 4
- カウンター：$|-5|+1+4=10$ 枚 のカード列
- ハートが左から i 番目：カウンターの値は $i-5-1$



- 既存のアイディアを拡張 [MHMS20]

[MHMS20] D. Miyahara, Y. Hayashi, T. Mizuki and H. Sone, “Practical Card-based Implementations of Yao’s Millonaire Protocol” Theor. Comput. Sci., (2020).

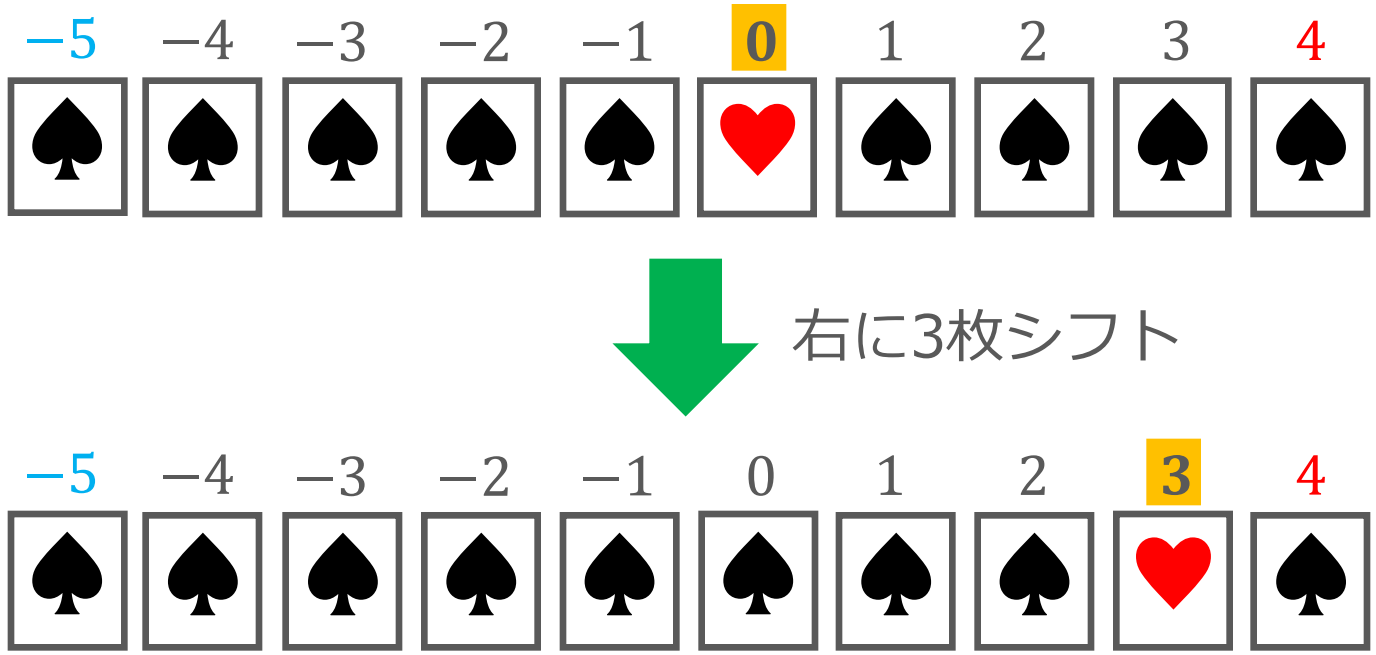
カウンターへの加算

➢加算＝カード列のシフト操作

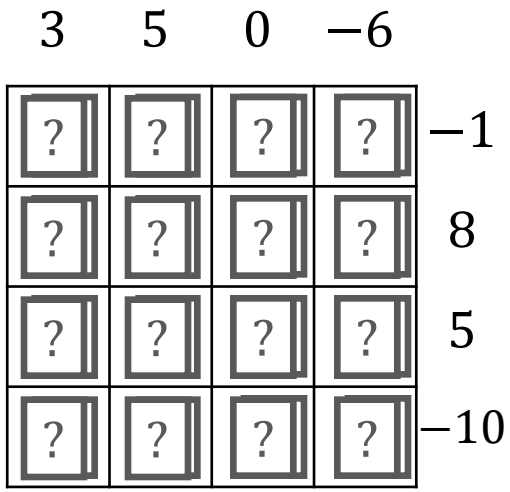
→ カードが裏向き（値が秘密）でも実行可能

➢負数 / 正数の加算：左 / 右にシフト

例）3 を加算



カウンターの切り替え方 – 下準備



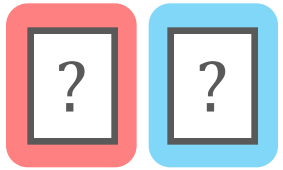
➢目的：残す/削除する マスは，本物/ダミー カウンターに加算
- ■今注目しているマスが残すかどうかは識別できないようにする

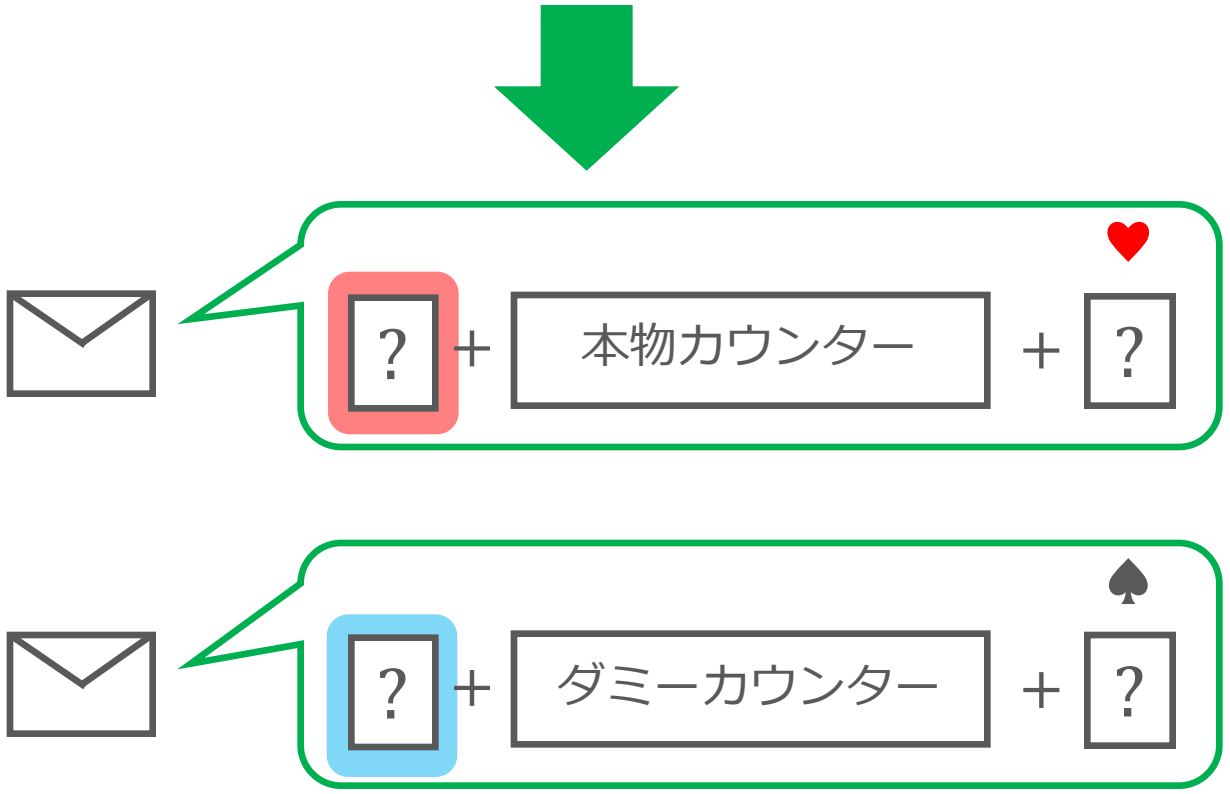
下準備：（削除するマスを知っている）Prover は，
各マスの上にカード組を裏向きに配置

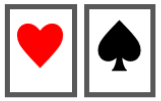
残すマス：♥ ♠

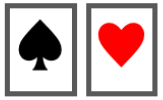
削除マス：♠ ♥

カウンターの切り替え方 – カウンターの配置

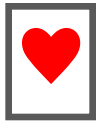
注目マスに配置されたカード組： 



残すマス： 

削除マス： 

① マスからカード組を取り出し，封入

- ■注目マスが残すマス
  - ⇒  の右は**本物**カウンター
- ■注目マスが削除するマス
  - ⇒  の右は**ダミー**カウンター

カウンターの切り替え方 – カウンターへ加算

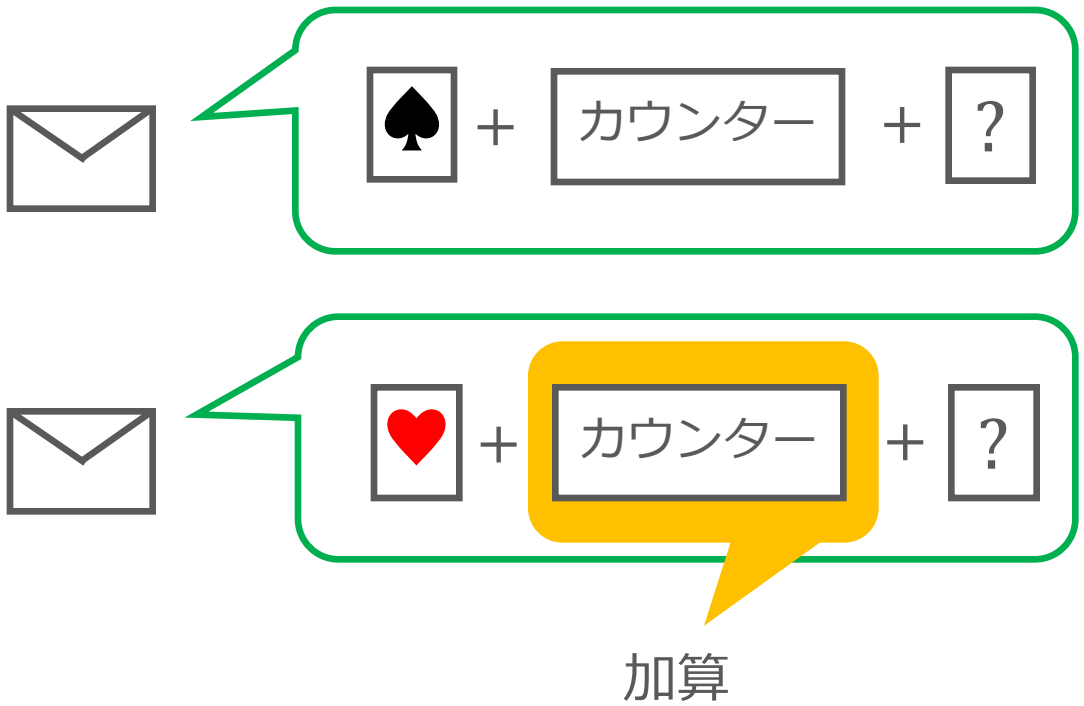
残すマス：♥ ♠

削除マス：♠ ♥

② 封筒をシャッフルし，開封

■左端のカードを表向きにし，

♥ の隣のカウンターに裏向きのまま加算操作



■注目マスが残すマス

⇒ ♥ の右は**本物**カウンター

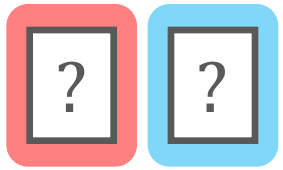
■注目マスが削除するマス

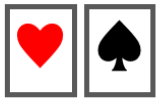
⇒ ♥ の右は**ダミー**カウンター

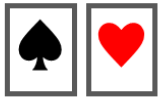
■シャッフルによって，

残すマスか削除するマスかは識別できない

カウンターの切り替え方 – カード組の復元

注目マスに配置されたカード組： 

残すマス： 

削除マス： 

③ カードをすべて裏向きにし，封筒へ戻す

④ 封筒をシャッフルし，開封
- 右端のカードをもとに
  マスへカード組を戻す

提案プロトコルの流れ

3	5	0	-6	
-3	1	-2	3	-1
5	-7	6	2	8
9	4	3	-8	5
-6	3	-4	-1	-10

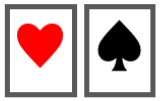
1. すべてのマスにカード組を配置
2. すべての行/列で検証
   a. カウンター作成
   b. 残すマス ⇒ 本物カウンターに加算
      削除マス ⇒ ダミーカウンターに加算
   c. 本物カウンター $\stackrel{?}{=}$ 目標値

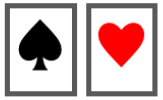
ギミックによって，
解答（削除するマスの位置）は秘匿

提案プロトコルの健全性

健全性 ($soundness$)

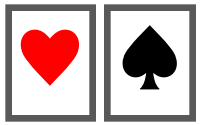
健全性誤り確率は 0（Prover は解答を知らない ⇒ Verifier は必ず拒否）

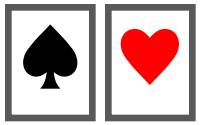
残すマス： 

削除マス： 

対偶「Verifier が証明を受理 ⇒ Prover は解答を知っている」を示す

Verifier が証明を受理
⇒ 任意の行/列について,

Prover が  を配置したマスの和＝目標値

⇒  を配置したマスを削除すれば解答が得られる

⇒ Prover は解答を知っている

提案プロトコルのカード枚数

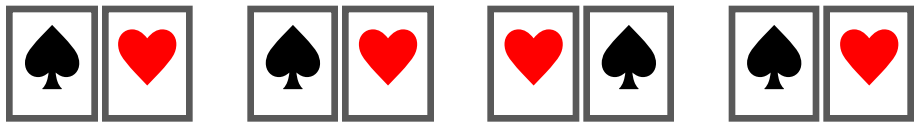
- 1マスにつき2枚 ⇒ $2mn$ 枚
- カウンターの切り替え ⇒ 2 枚
- カウンター作成 ⇒ $2k$ 枚
  - 目標値 k を表現
  - 2個作成

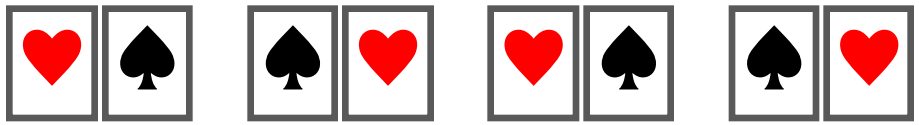
$$2mn + 2k + 2$$
$$= 2mn + 2 \cdot 2^{\log_2 k} + 2 \text{ 枚}$$

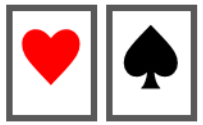
- カード枚数が目標値 k のビット長の指数関数になってしまう！

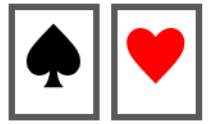
カウンターの様式変更

➢カウンターに2の補数表現を用いる

$2_{(10)} = 0010_{(2)} =$ 

$-5_{(10)} = 1010_{(2)} =$ 

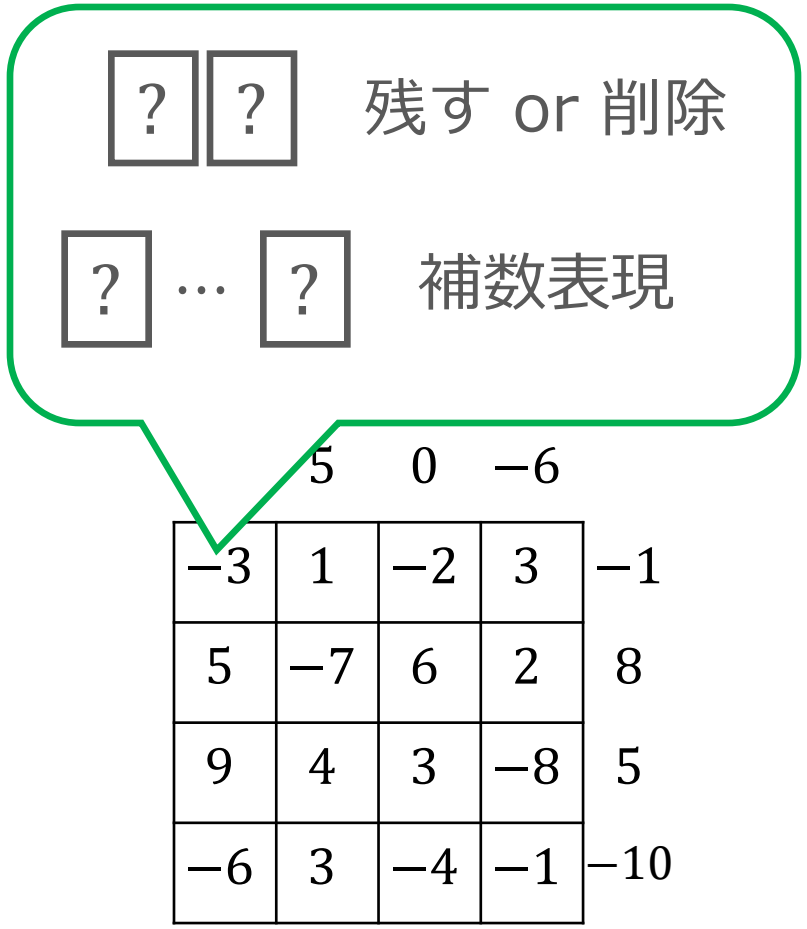
1： 

0： 

➢カウンターの値を秘匿したまま加算：ANDプロトコルとXORプロトコルで実行可能 [NHMS15]

[NHMS15] T. Nishida, Y. Hayashi, T. Mizuki and H. Sone, “Card-Based Protocols for Any Boolean Function” Theory and Applications of Models of Computation, (2015).

2の補数を用いるプロトコルの流れ



1. すべてのマスにカード組を配置
2. マスの値を2の補数で表現
3. すべての行/列で検証
   a. カウンター作成
   b. 残すマス ⇒ 本物カウンターに加算
      削除マス ⇒ ダミーカウンターに加算
   c. 本物カウンター $\stackrel{?}{=}$ 目標値

２の補数を用いた場合のカード枚数

- 1マスに配置するカード　⇒ $mn(2\log_2 k + 2)$ 枚
  - 残す / 削除の指示ペア：２枚
  - マスの数字を2の補数で表現：$2\log_2 k$ 枚
- カウンターの切り替え　⇒ 2 枚
- カウンター作成　⇒ $2 \times 2\log_2 k$ 枚
  - 目標値 k を表現
  - 2個作成

$$mn(2\log_2 k + 2) + 4\log_2 k + 2 \text{ 枚}$$

プロトコルの比較

	カード枚数	シャッフル回数	シフト回数
整数コミットメントを用いる手法	$2mn + 2 \cdot 2^{\log_2 k} + 2$	$4mn$	mn
2の補数表現を用いる手法	$mn(2\log_2 k + 2) + 4 \cdot \log_2 k + 2$	$2mn + 7mn \cdot \log_2 k$	0

- 人の手で行う場合：整数コミットメントの手法が適切
  - シフト操作による加算が分かりやすく，実行が簡単
  - 大きな数字を取り扱うことは少ない
- 定義を遵守する場合：補数表現を用いる手法が適切
  - （理論上）実現可能なカード枚数

まとめ

① Sumplete のNP完全性の証明
- ■部分和問題からの変換
- ■XSAT for 3-CNF_{+}^{3} からの変換

② Sumplete に対する
物理的ゼロ知識証明の提案
- ■ダミーカウンターを利用して
解答を秘匿

46	-30	36	10	-29	-8	38	-3	53	
37	18	-35	7	-15	-18	15	39	11	-15
-8	3	2	15	-30	31	25	-32	27	83
29	11	16	-24	-38	-7	-33	-6	18	-5
-23	34	29	1	33	7	-28	8	15	76
8	-26	-6	30	-4	-1	-15	21	-7	-37
11	39	0	0	-14	-16	9	-11	17	21
31	-38	-20	6	22	-22	39	30	-31	-7
-25	-20	38	33	-27	-18	4	-34	17	40
-10	17	-12	-1	-24	32	-13	-5	-26	-43

付録

➢解答

46	-30	36	10	-29	-8	38	-3	53	
✕	✕	✕	✕	-15	✕	✕	✕	✕	-15
✕	✕	✕	✕	✕	31	25	✕	27	83
29	✕	16	-24	-38	✕	✕	-6	18	-5
✕	34	✕	1	33	✕	✕	8	✕	76
✕	-26	-6	✕	-4	-1	✕	✕	✕	-37
11	✕	✕	✕	✕	-16	9	✕	17	21
31	-38	✕	✕	22	-22	✕	✕	✕	-7
-25	✕	38	33	-27	✕	4	✕	17	40
✕	✕	-12	✕	✕	✕	✕	-5	-26	-43