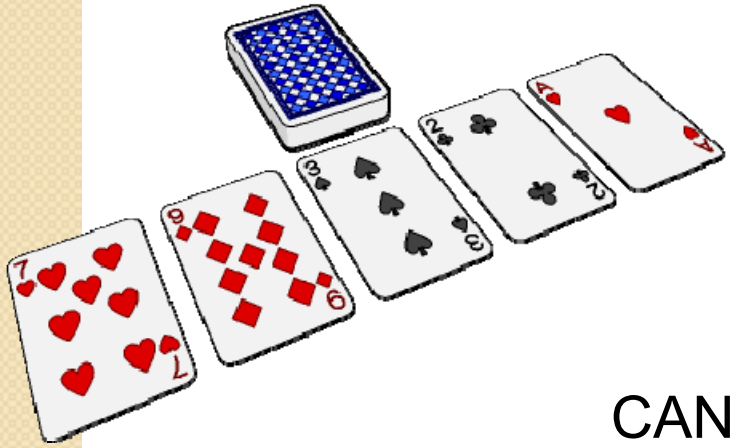
Efficient and Secure Multiparty Computations Using a Standard Deck of Playing Cards

Takaaki Mizuki

Tohoku University



CANS 2016 (16:30-17:00, Nov. 15, 2016, Milan)

Contents

東北大学

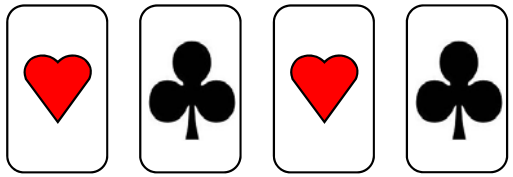
Contents

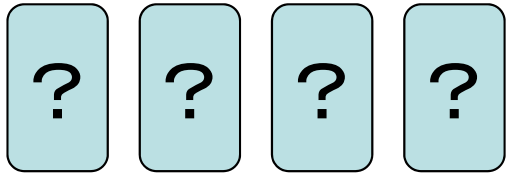
東北大学

It is known that

secure multi-party computation

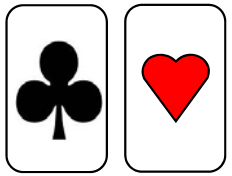
can be performed using red and black cards.

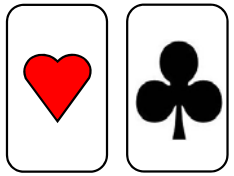




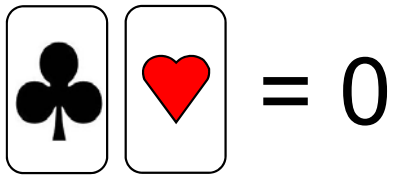
東北大学

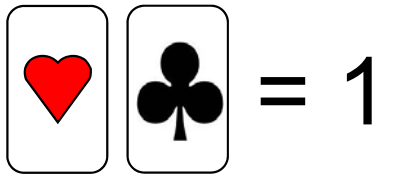
Using two cards of different colors, we can deal with Boolean values, based on the following encoding:

 = 0

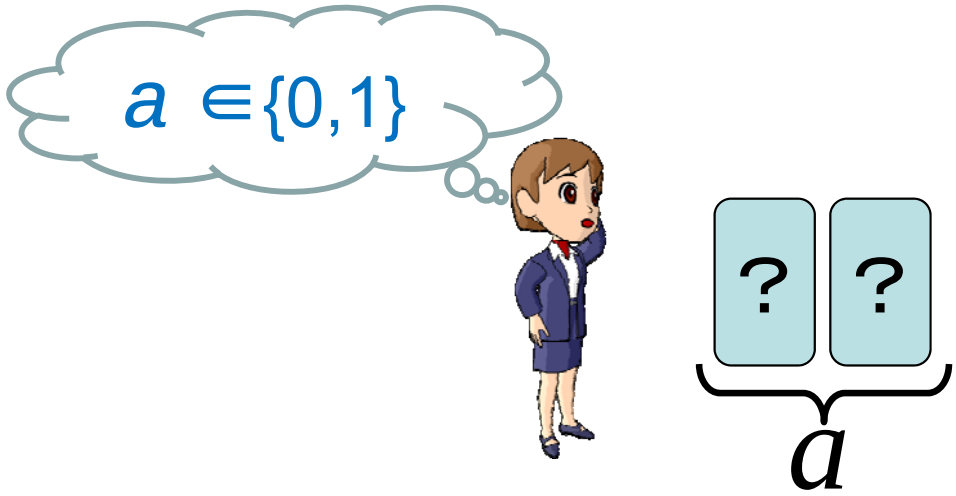
 = 1

東北大学

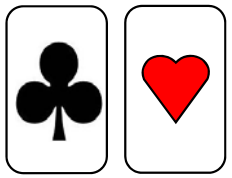


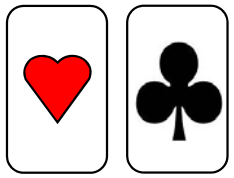


Based on this encoding, a player can commit his/her private input bit to a pair of face-down cards.

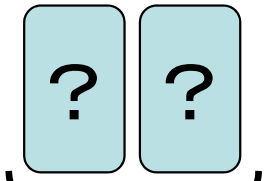


called a ***commitment***


= 0


= 1

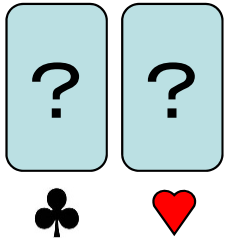
$a \in \{0,1\}$



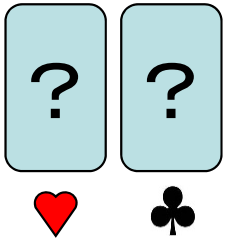
a

called a ***commitment***

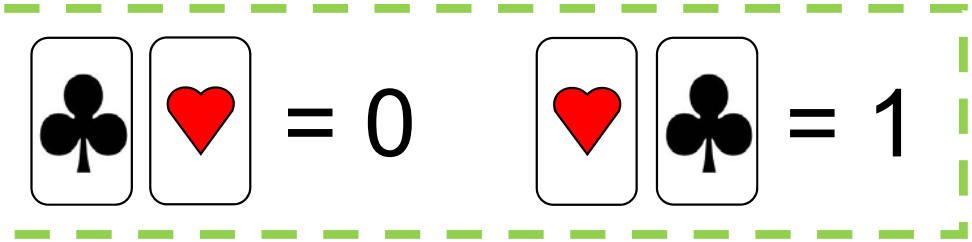
if $a=0$



if $a=1$

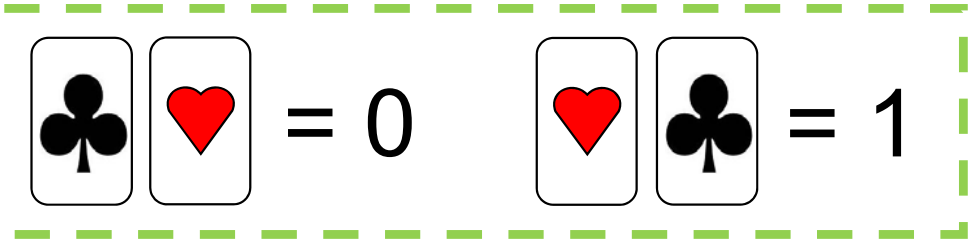


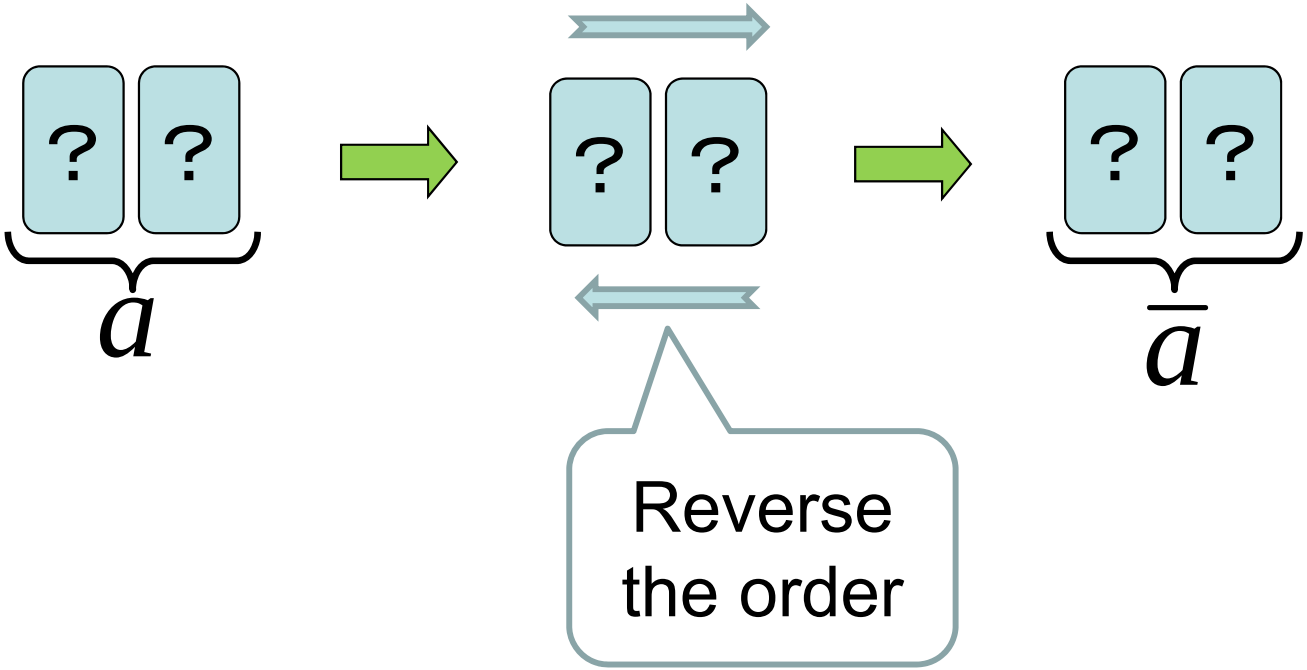
NOT computation



東北大学

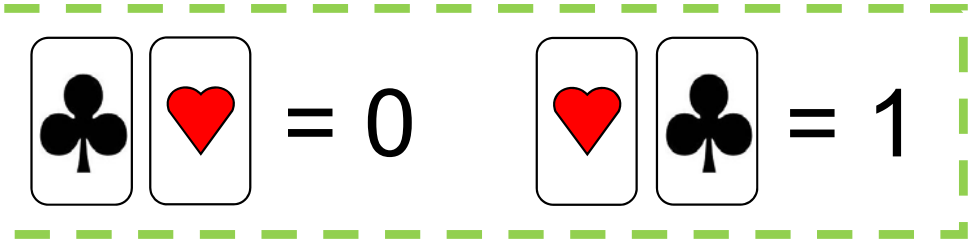
NOT computation

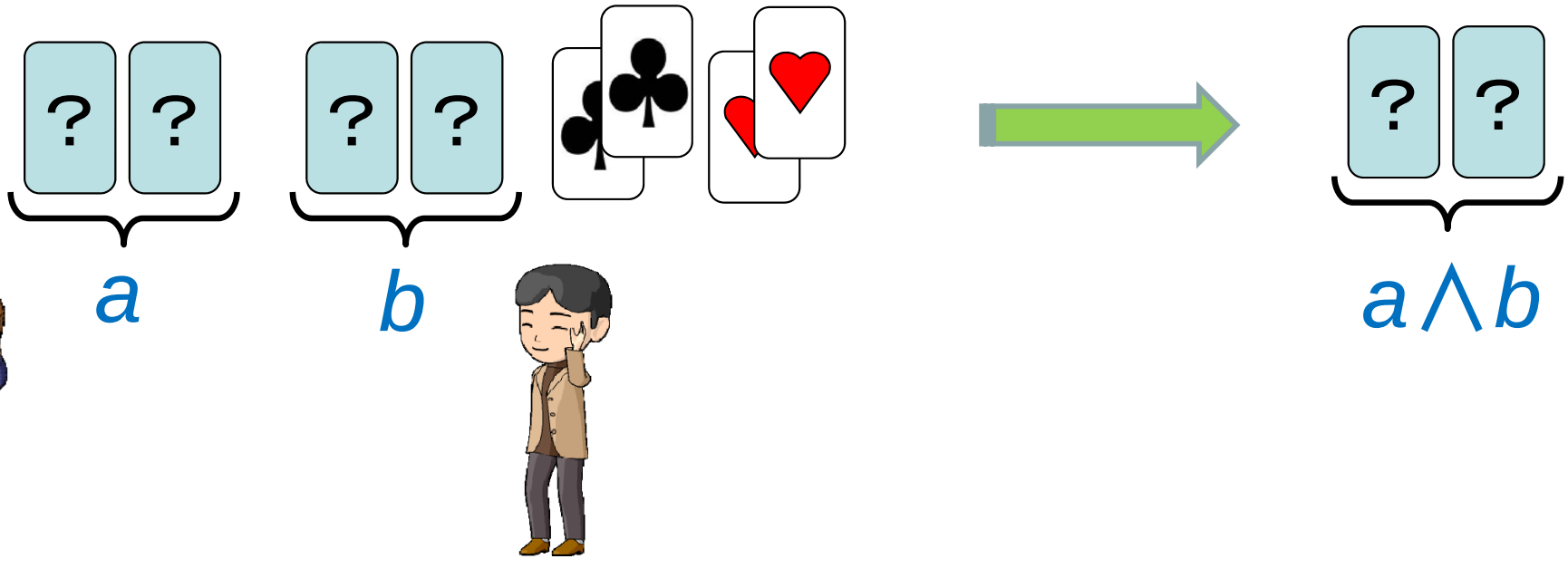




東北大学

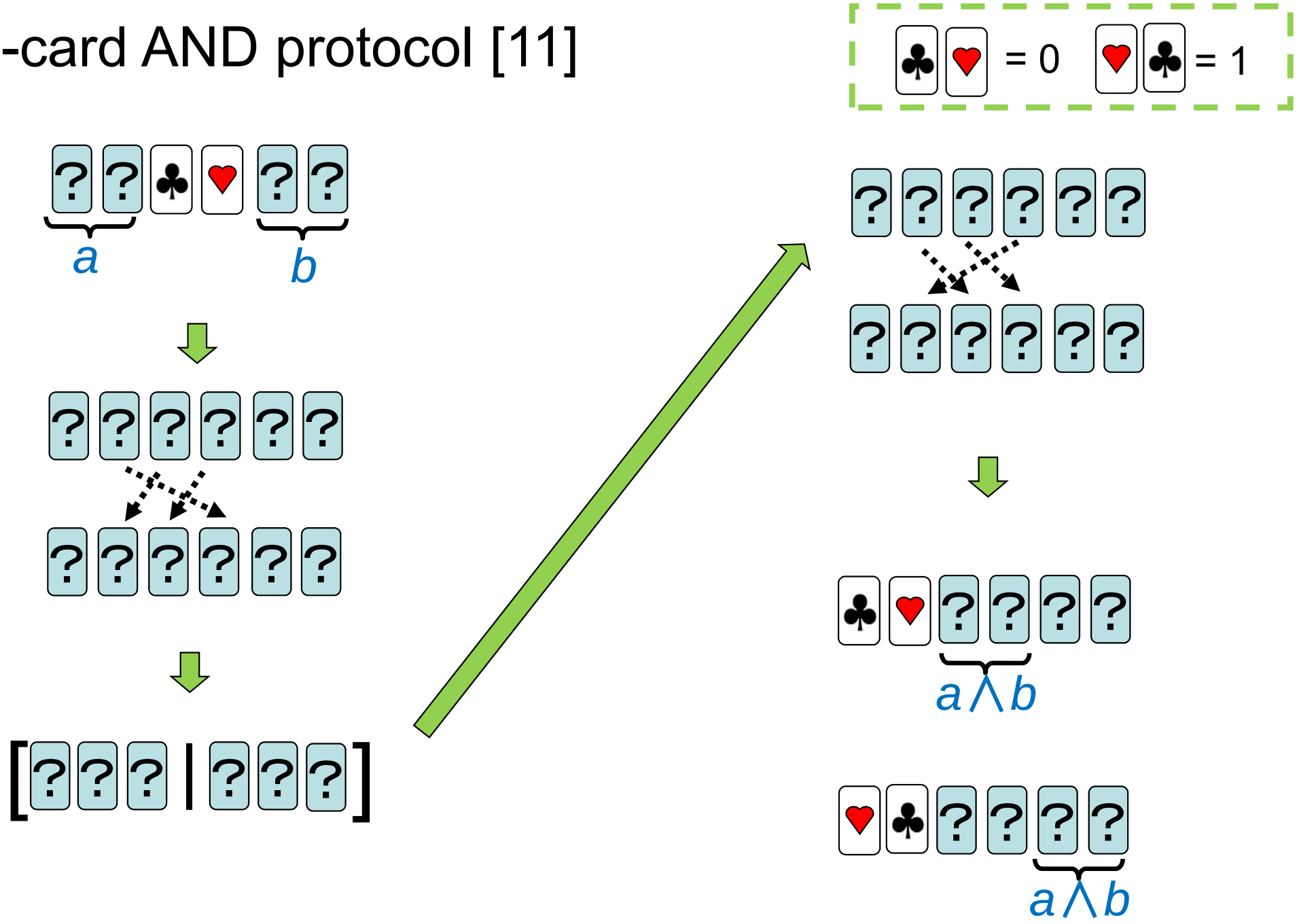
AND computation





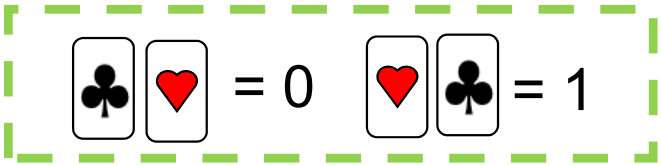
There are many protocols for secure AND computation.

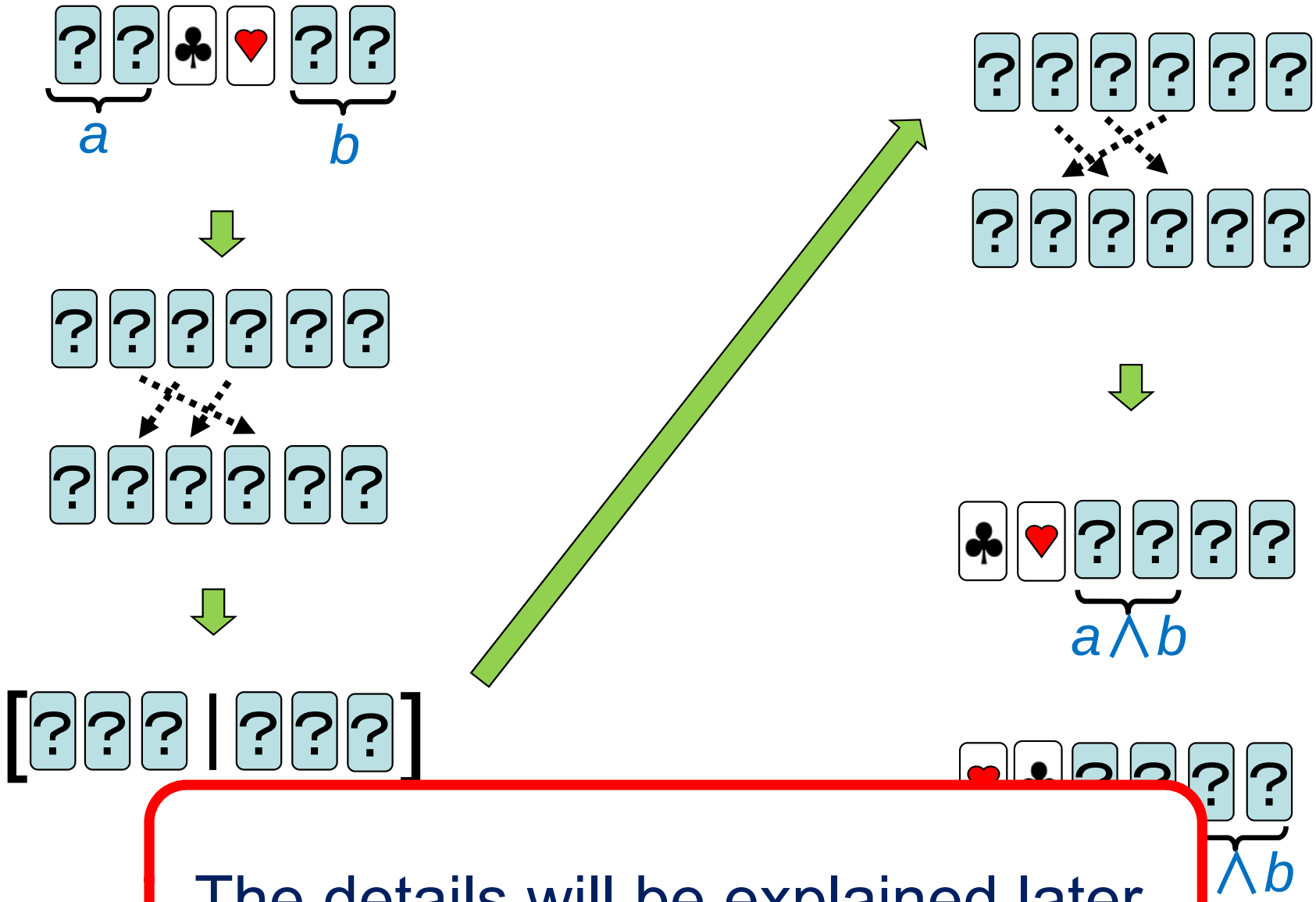
6-card AND protocol [11]



[11] T. Mizuki and H. Sone, Six-Card Secure AND and Four-Card Secure XOR, FAW 2009, LNCS 5598, pp. 358–369, 2009.

6-card AND protocol [11]





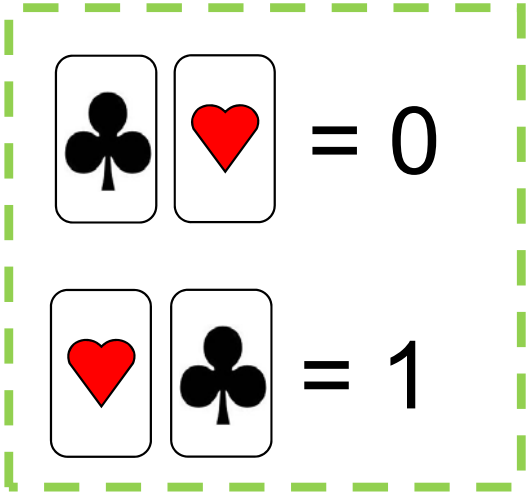
The details will be explained later.

[11] rd Secure XOR,

XOR computation

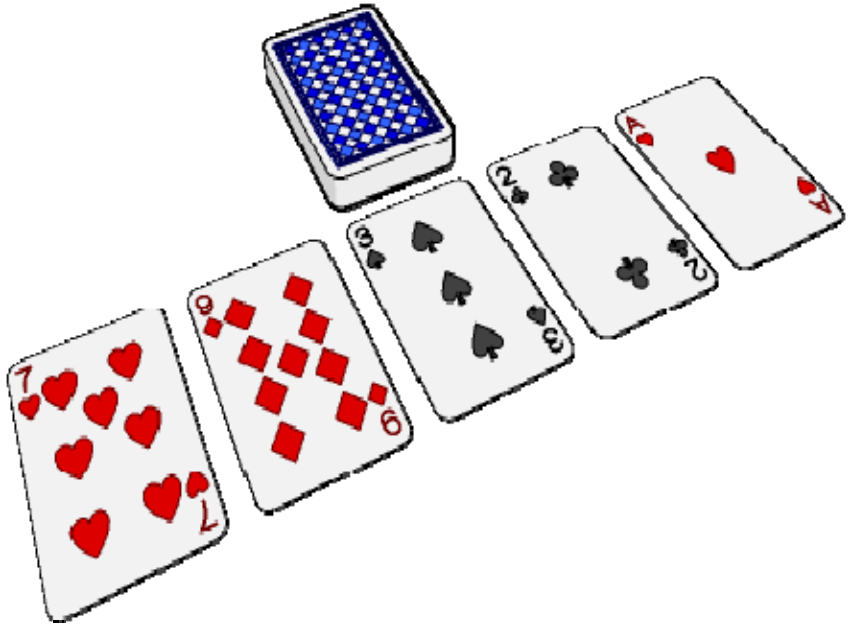
4-card XOR protocol [11]

The details are omitted.



[11] T. Mizuki and H. Sone, Six-Card Secure AND and Four-Card Secure XOR, FAW 2009, LNCS 5598, pp. 358–369, 2009.

These protocols cannot be executed with a standard deck of playing cards.

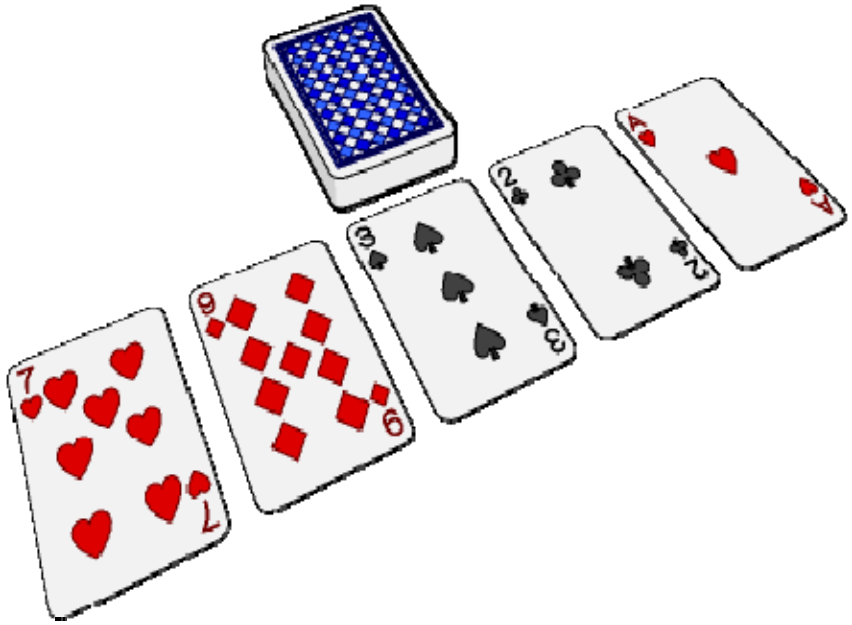


They need custom-made cards.

Almost all existing protocols were designed for custom two-colored cards.

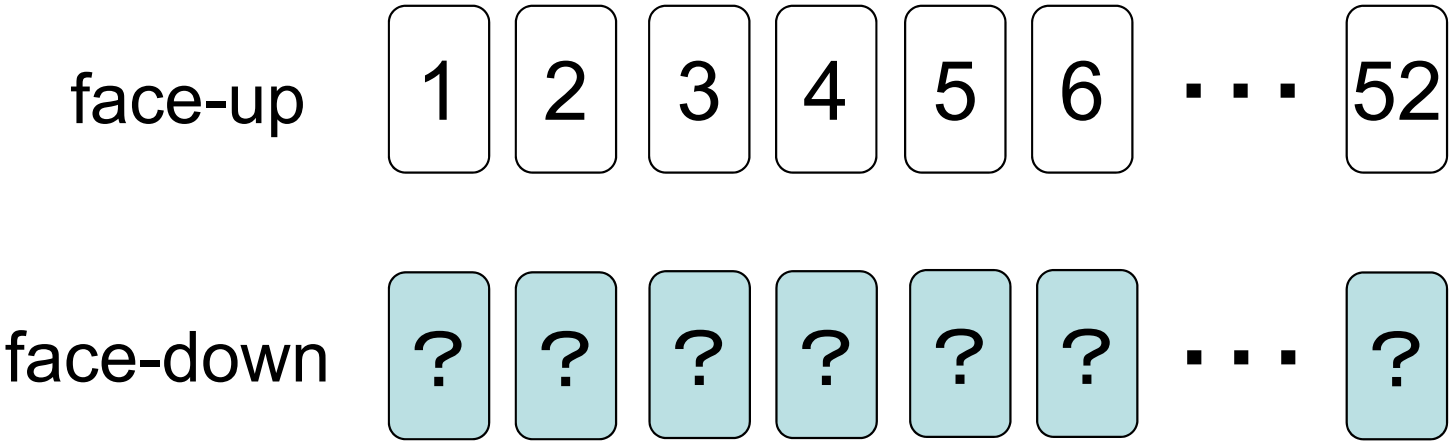
There is one exception: Niemi-Renvall protocols [13] can be executed with standard playing cards.

[13] V. Niemi and A. Renvall, “Solitaire zero-knowledge,” Fundamenta Informaticae, vol. 38, pp. 181–188, 1999



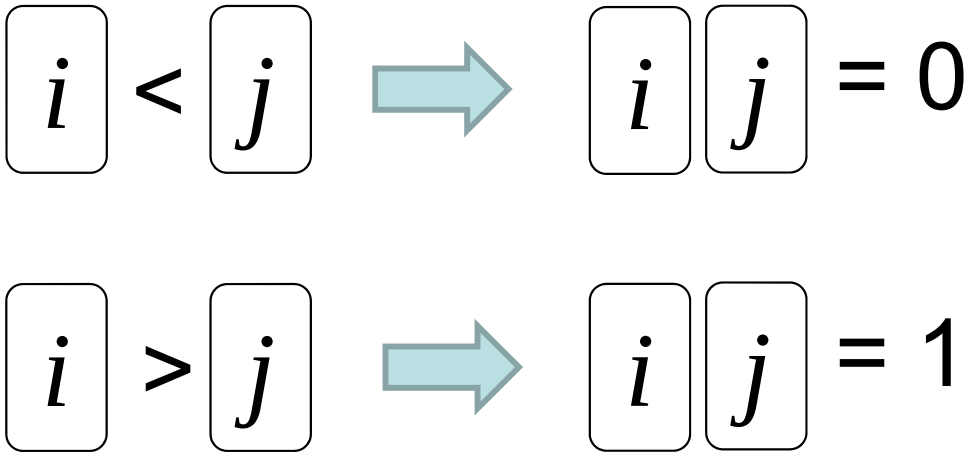
Use of a standard deck of playing cards

We assume the following deck of 52 cards (each card has a unique natural number):



東北大学

Niemi and Renvall [13] considered an encoding rule:



東北大学

Niemi and Renvall [13] considered an encoding rule:

$$\boxed{i} < \boxed{j} = 0$$

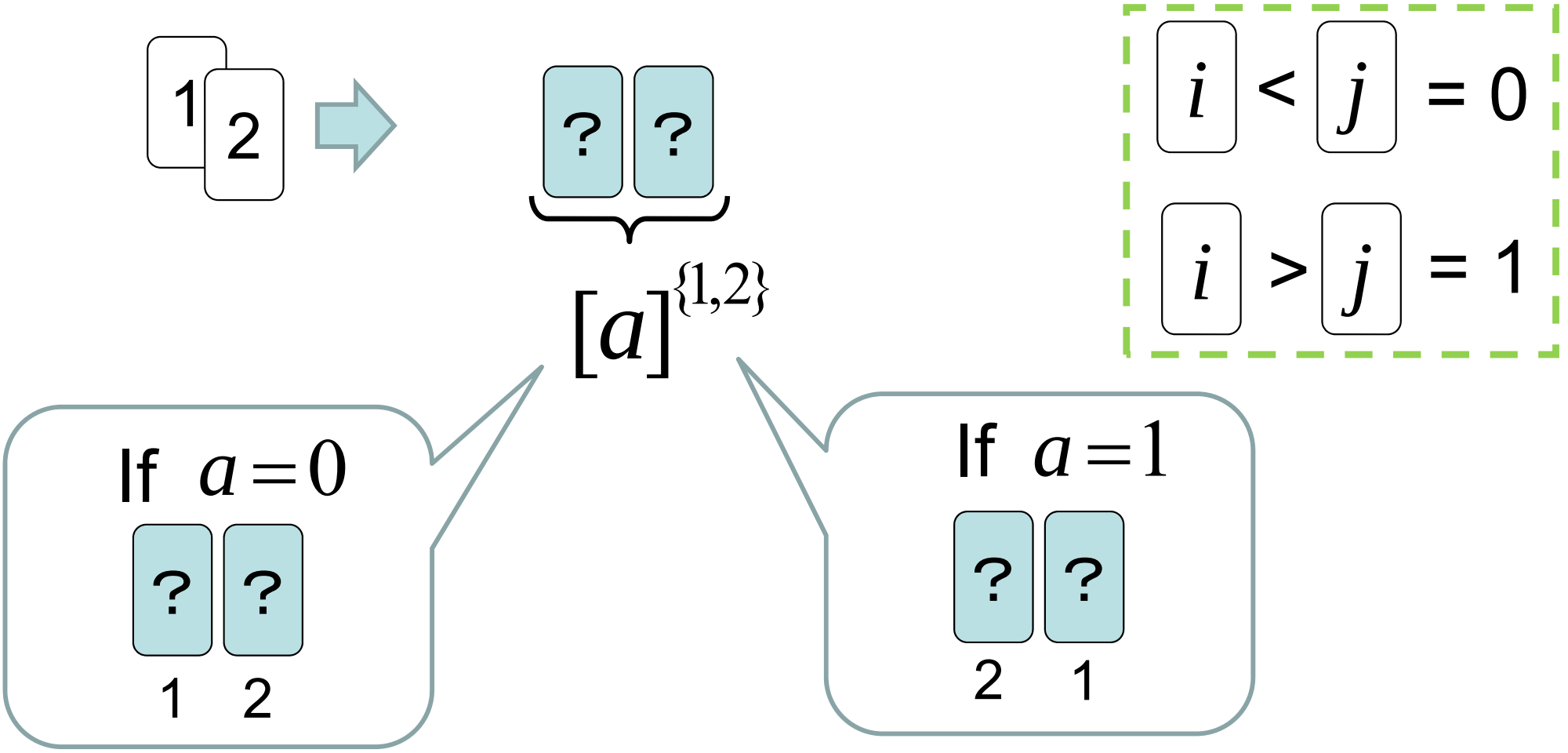
$$\boxed{i} > \boxed{j} = 1$$

Example:

$$\boxed{1}\,\boxed{2} = 0 \qquad \boxed{2}\,\boxed{1} = 1$$

東北大学

We can naturally consider a commitment as well:



We call such a set {1,2} a ***base*** of the commitment.

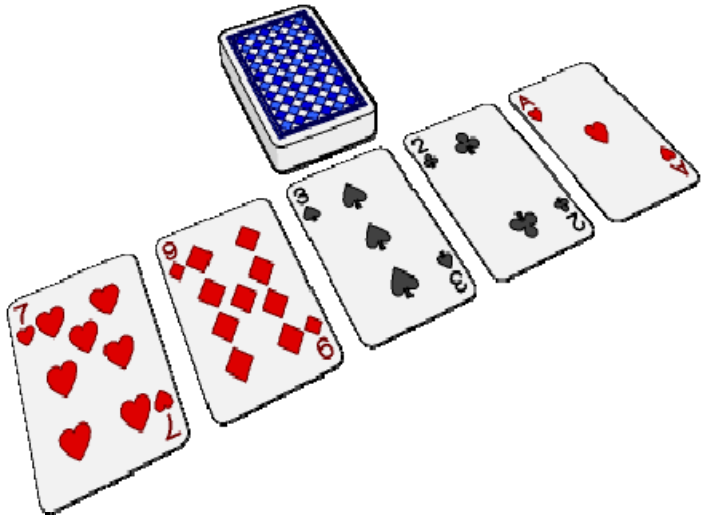
Under this encoding rule on standard playing cards:

- ✓ A NOT computation is trivial.
  → just to reverse the order

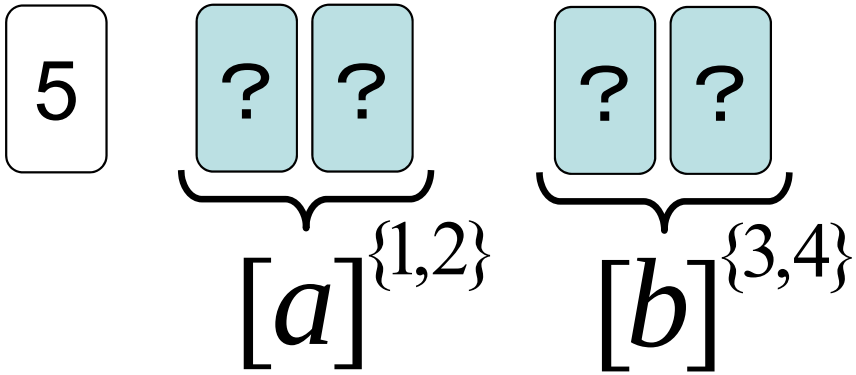
$$\boxed{i} < \boxed{j} = 0$$

$$\boxed{i} > \boxed{j} = 1$$

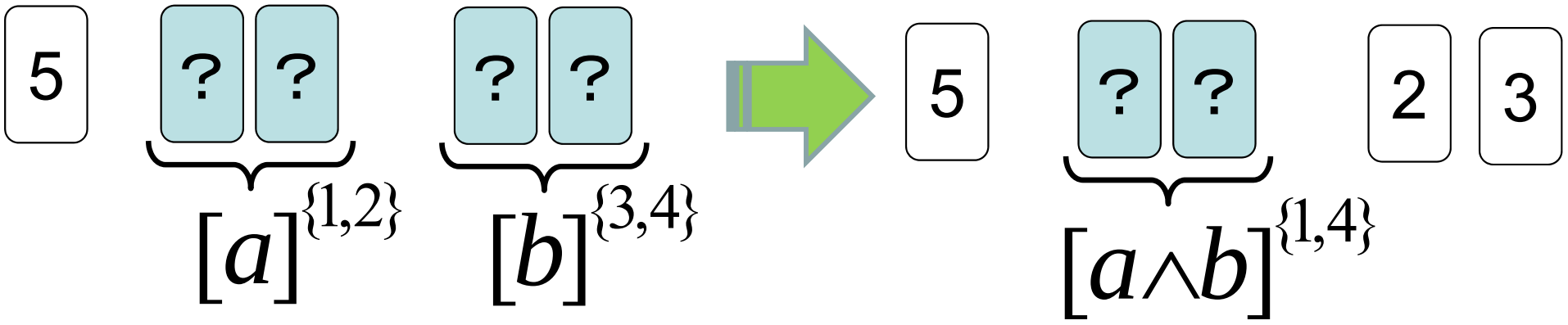
- ✓ Niemi and Renvall [13] constructed:
  - AND protocol
  - XOR protocol



Niemi-Renvall AND protocol [13]



Niemi-Renvall AND protocol [13]



A random cut（= cyclic shuffle）is applied an average of **9.5** times.

Rotate in a circular motion

東北大学

AND computation	# of cards	# of shuffles	
		avg.	fixed
Niemi-Renvall [13]（Sect. 2）	5	9.5	0

東北大学

AND computation

	# of cards	# of shuffles	
		avg.	fixed
Niemi-Renvall [13]（Sect. 2）	5	9.5	0

XOR computation

Niemi-Renvall [13]（Sect. 2）	4	7	0

東北大学

Our results

We will propose more efficient protocols:

- ✓ Utilizing random bisection cuts
- ✓ Simulating the existing protocols [11] + α

based on custom-made cards

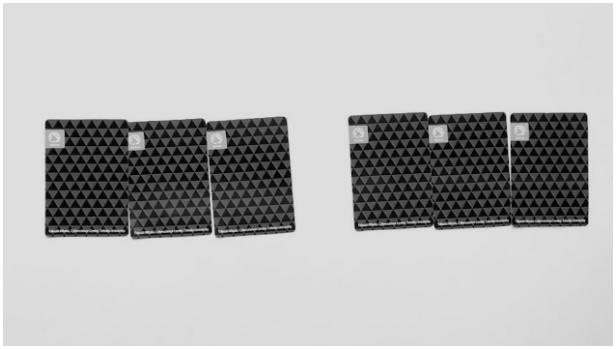
[11] T. Mizuki and H. Sone, Six-Card Secure AND and Four-Card Secure XOR, FAW 2009, LNCS 5598, pp. 358–369, 2009.

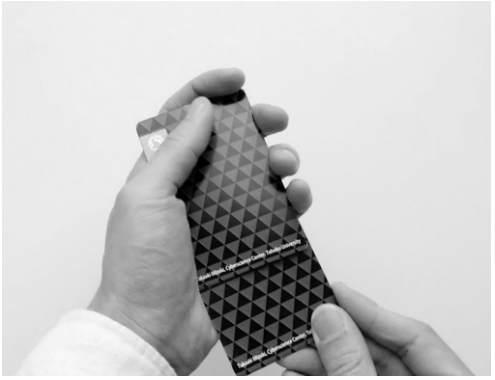
Our results

We will propose more efficient protocols:

- ✓ Utilizing random bisection cuts
- ✓ Simulating the existing p[illegible]cols [11] + α

Switch two halves randomly





東北大学

AND computation

	# of cards	# of shuffles	
		avg.	fixed
Niemi-Renvall [13]（Sect. 2）	5	9.5	0

XOR computation

Niemi-Renvall [13]（Sect. 2）	4	7	0

東北大学

AND computation	# of cards	# of shuffles	
		avg.	fixed
Niemi-Renvall [13]（Sect. 2）	5	9.5	0
Ours（Sect. 3）	8	0	4

XOR computation			
Niemi-Renvall [13]（Sect. 2）	4	7	0

東北大学

AND computation	# of cards	# of shuffles	
		avg.	fixed
Niemi-Renvall [13]（Sect. 2）	5	9.5	0
Ours（Sect. 3）	8	0	4

XOR computation

Niemi-Renvall [13]（Sect. 2）	4	7	0
Ours（Sect. 4）	4	0	1

東北大学

AND computation	# of cards	# of shuffles	
		avg.	fixed
Niemi-Renvall [13]（Sect. 2）	5	9.5	0
Ours（Sect. 3）	8	0	4

XOR computation			
Niemi-Renvall [13]（Sect. 2）	4	7	0
Ours（Sect. 4）	4	0	1

Fewer shuffles and finite-runtime

東北大学

AND computation

	# of cards	# of shuffles	
		avg.	fixed
Niemi-Renvall [13]（Sect. 2）	5	9.5	0
Ours（Sect. 3）	8	0	4

XOR computation

Niemi-Renvall [13]（Sect	4	7	0
Ours（Sect. 4）	4	0	1

Minor issue:
we have 52 cards!

東北大学

Contents

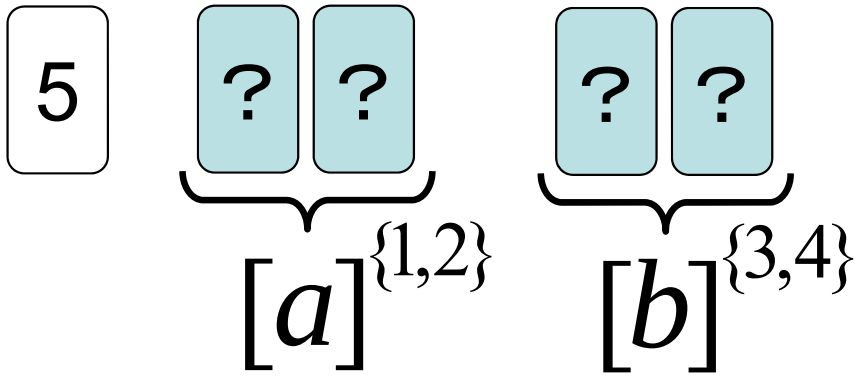
東北大学

AND computation	# of cards	# of shuffles	
		avg.	fixed
Niemi-Renvall [13]（Sect. 2）	5	9.5	0
Ours（Sect. 3）	8	0	4

XOR computation			
Niemi-Renvall [13]（Sect. 2）	4	7	0
Ours（Sect. 4）	4	0	1

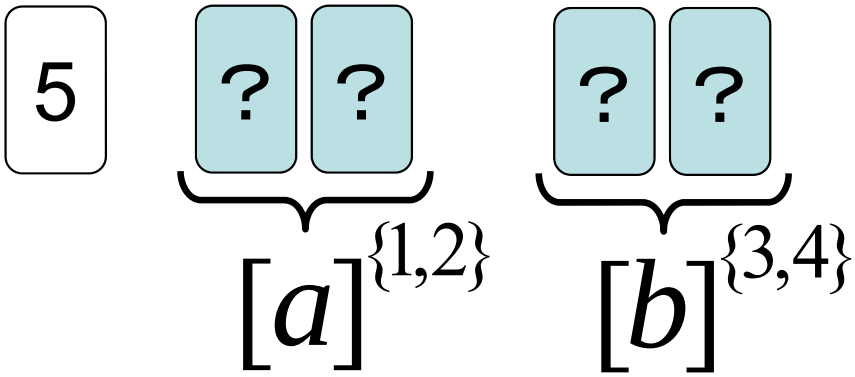
東北大学

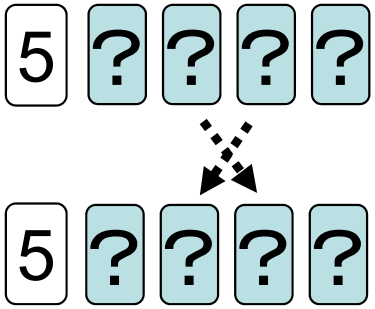
Niemi-Renvall AND protocol [13]



東北大学

Niemi-Renvall AND protocol [13]

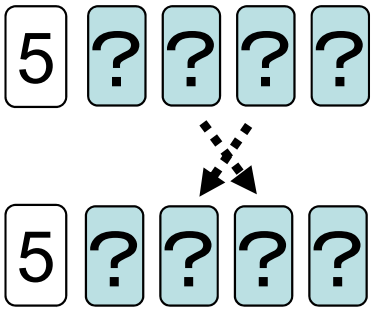




(a,b)	seq. of cards
(0,0)	5 1 3 2 4
(0,1)	5 1 4 2 3
(1,0)	5 2 3 1 4
(1,1)	5 2 4 1 3

東北大学

Niemi-Renvall AND protocol [13]



(*a*,*b*)	sequence
(0,0)	5 1 3 2 4
(0,1)	5 1 4 2 3
(1,0)	5 2 3 1 4
(1,1)	5 2 4 1 3

If we somehow deleted 2 and 3 :

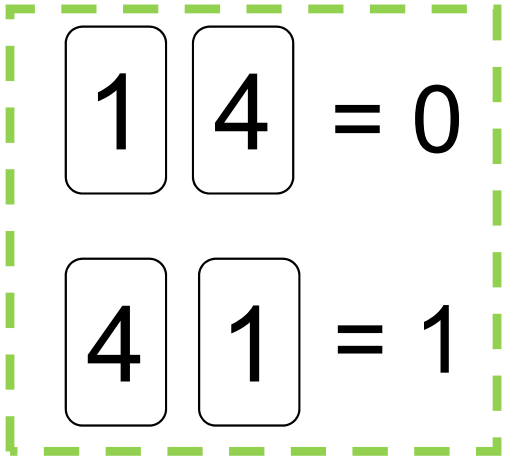
(*a*,*b*)	sequence
(0,0)	5 1 ~~3~~ ~~2~~ 4
(0,1)	5 1 4 ~~2~~ ~~3~~
(1,0)	5 ~~2~~ ~~3~~ 1 4
(1,1)	5 ~~2~~ 4 1 ~~3~~

東北大学

Niemi-Renvall AND protocol [13]

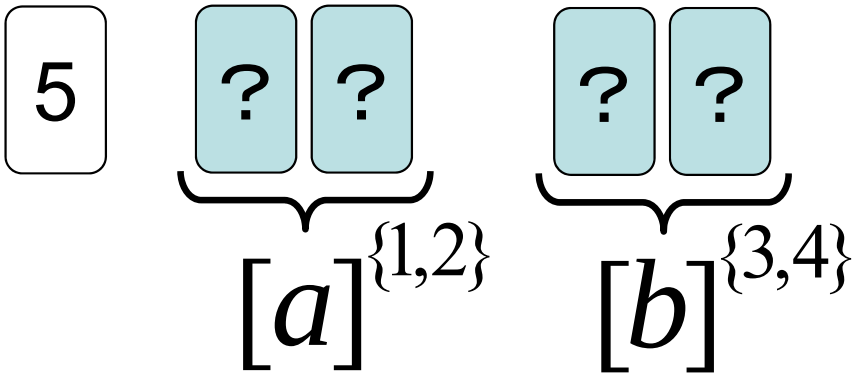
If we somehow deleted 2 and 3 :

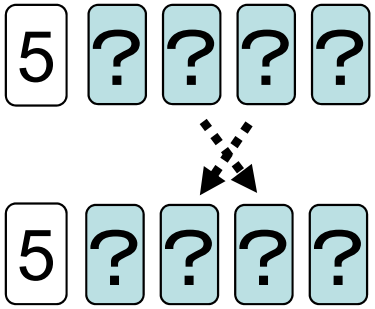
(*a*,*b*)	sequence
(0,0)	5 1 ~~3~~ ~~2~~ 4
(0,1)	5 1 4 ~~2~~ ~~3~~
(1,0)	5 ~~2~~ ~~3~~ 1 4
(1,1)	5 ~~2~~ 4 1 ~~3~~



…we would get a desired commitment!

Niemi-Renvall AND protocol [13]

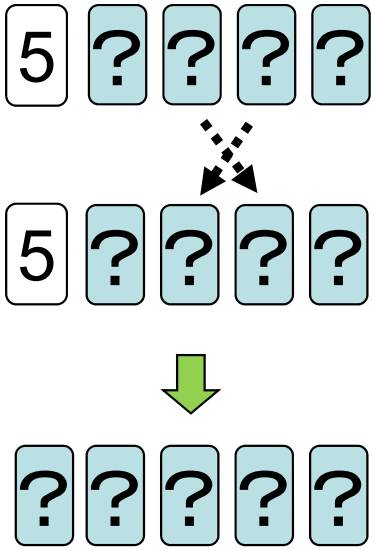




Thus, we want to delete 2 and 3 so that the two cards following 5 will be a desired commitment.

→ Utilizing a random cut（= cyclic shuffle）

Niemi-Renvall AND protocol [13]

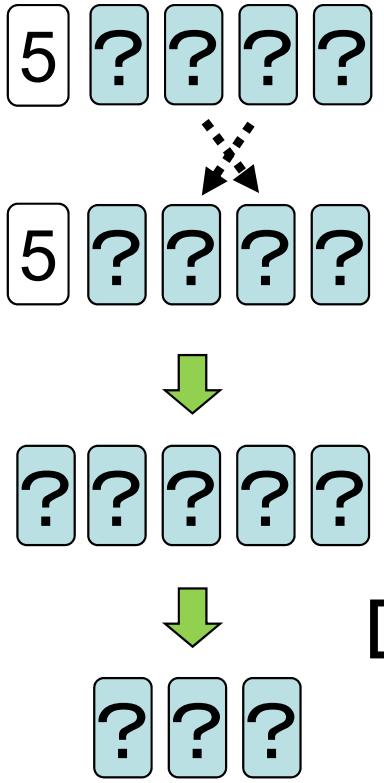


Rotate in a circular motion

Apply a random cut and reveal the first card. Unless the face-up card is 2 or 3 , turn over the card, apply a random cut, and reveal the first card again.

東北大学

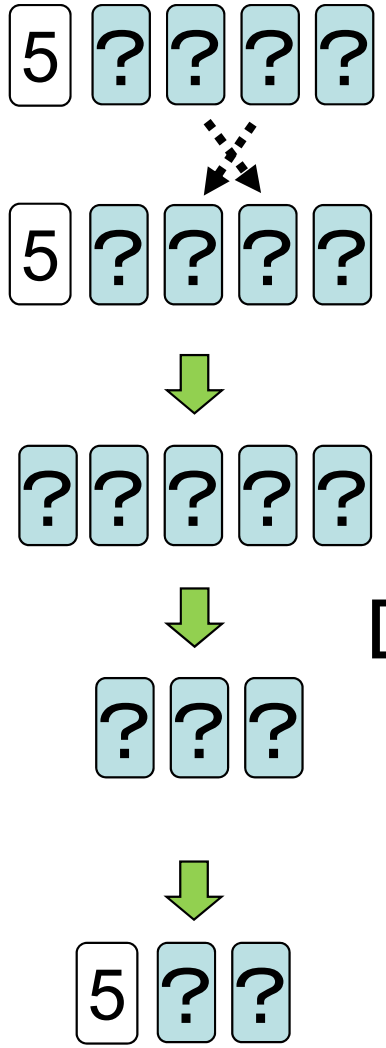
Niemi-Renvall AND protocol [13]



Delete 2 and 3 （Avg. # of 6.5 trials）

東北大学

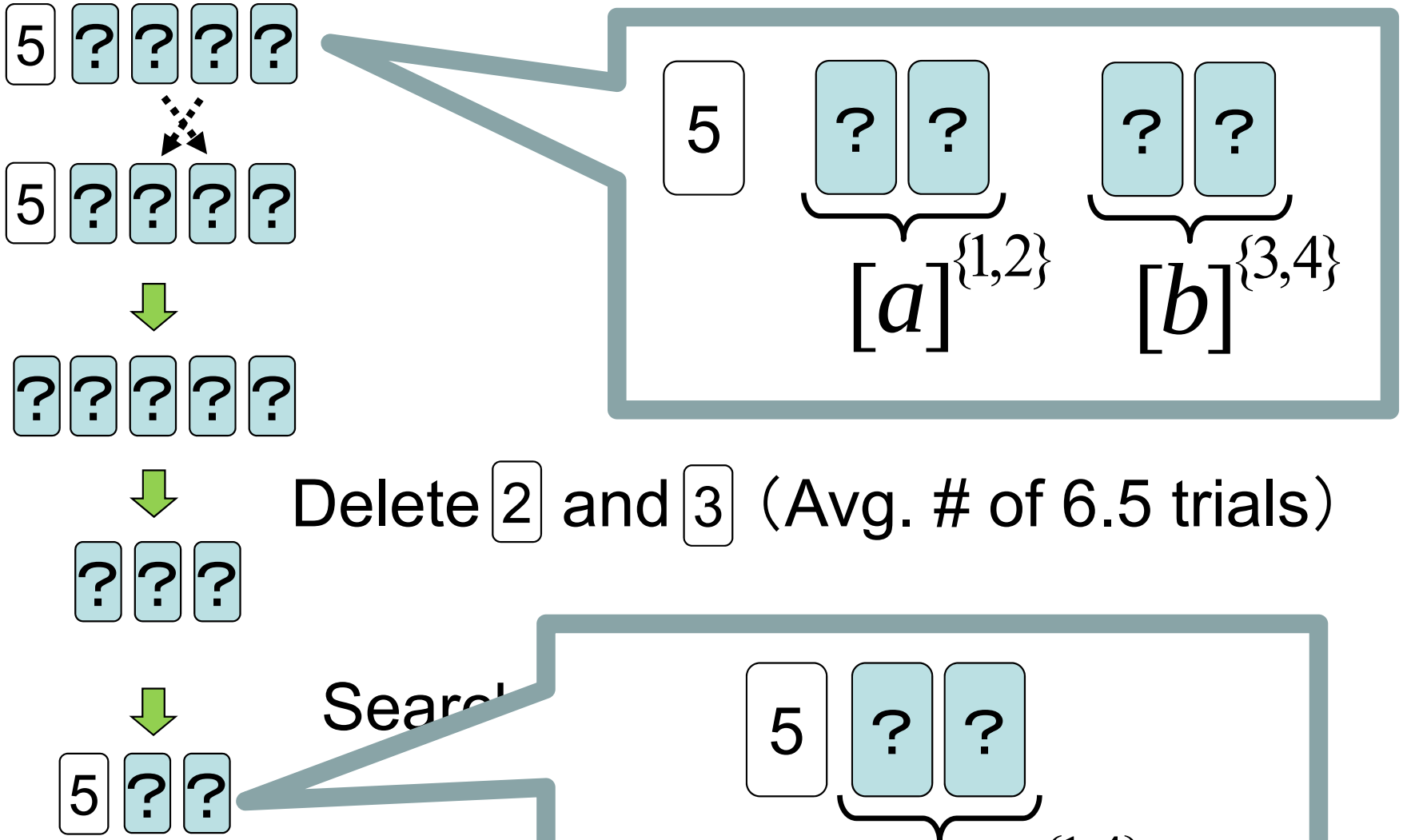
Niemi-Renvall AND protocol [13]



Delete 2 and 3 （Avg. # of 6.5 trials）

Search 5 （ Avg. # of 3 trials ）

Niemi-Renvall AND protocol [13]



AND computation	# of cards	# of shuffles	
		avg.	fixed
Niemi-Renvall [13]（Sect. 2）	5	9.5	0
Ours（Sect. 3）	8	0	4

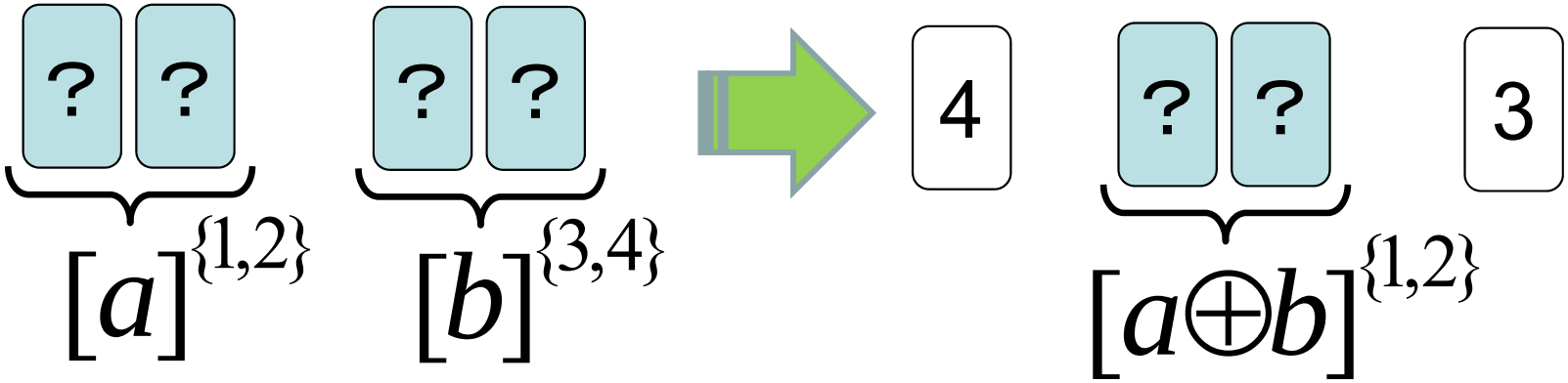
XOR computation			
Niemi-Renvall [13]（Sect. 2）	4	7	0
Ours（Sect. 4）	4	0	1

東北大学

AND computation	# of cards	# of shuffles	
		avg.	fixed
Niemi-Renvall [13]（Sect. 2）	5	9.5	0
Ours（Sect. 3）	8	0	4
XOR computation			
Niemi-Renvall [13]（Sect. 2）	4	7	0
Ours（Sect. 4）	4	0	1

東北大学

Niemi-Renvall XOR protocol [13]



A random cut（= cyclic shuffle）is applied an average of **7** times.

Omit the details

AND computation	# of cards	# of shuffles	
		avg.	fixed
Niemi-Renvall [13]（Sect. 2）	5	9.5	0
Ours（Sect. 3）	8	0	4
XOR computation			
Niemi-Renvall [13]（Sect. 2）	4	7	0
Ours（Sect. 4）	4	0	1

東北大学

Contents

Let me explain "XOR" first.

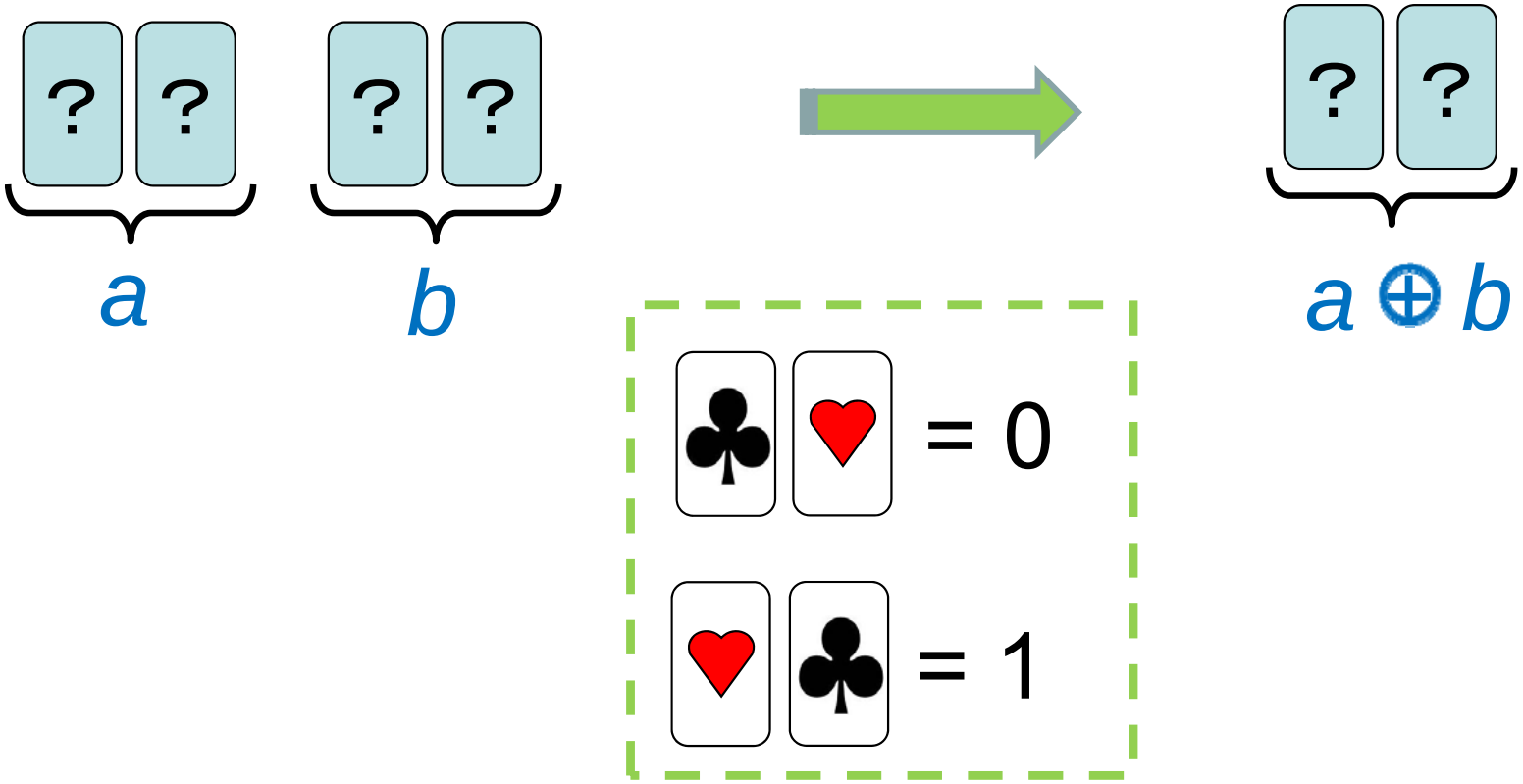
東北大学

Contents

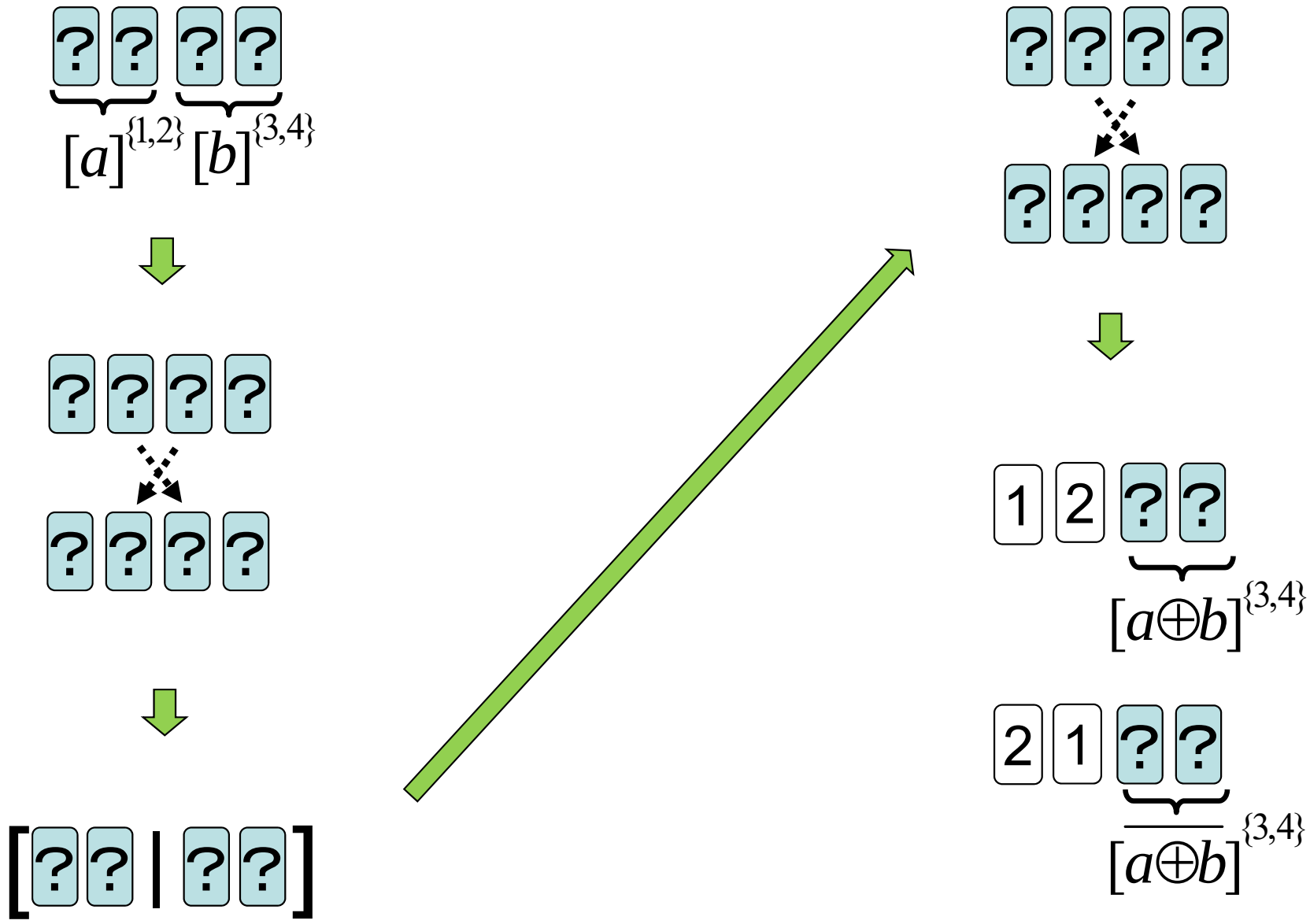
東北大学

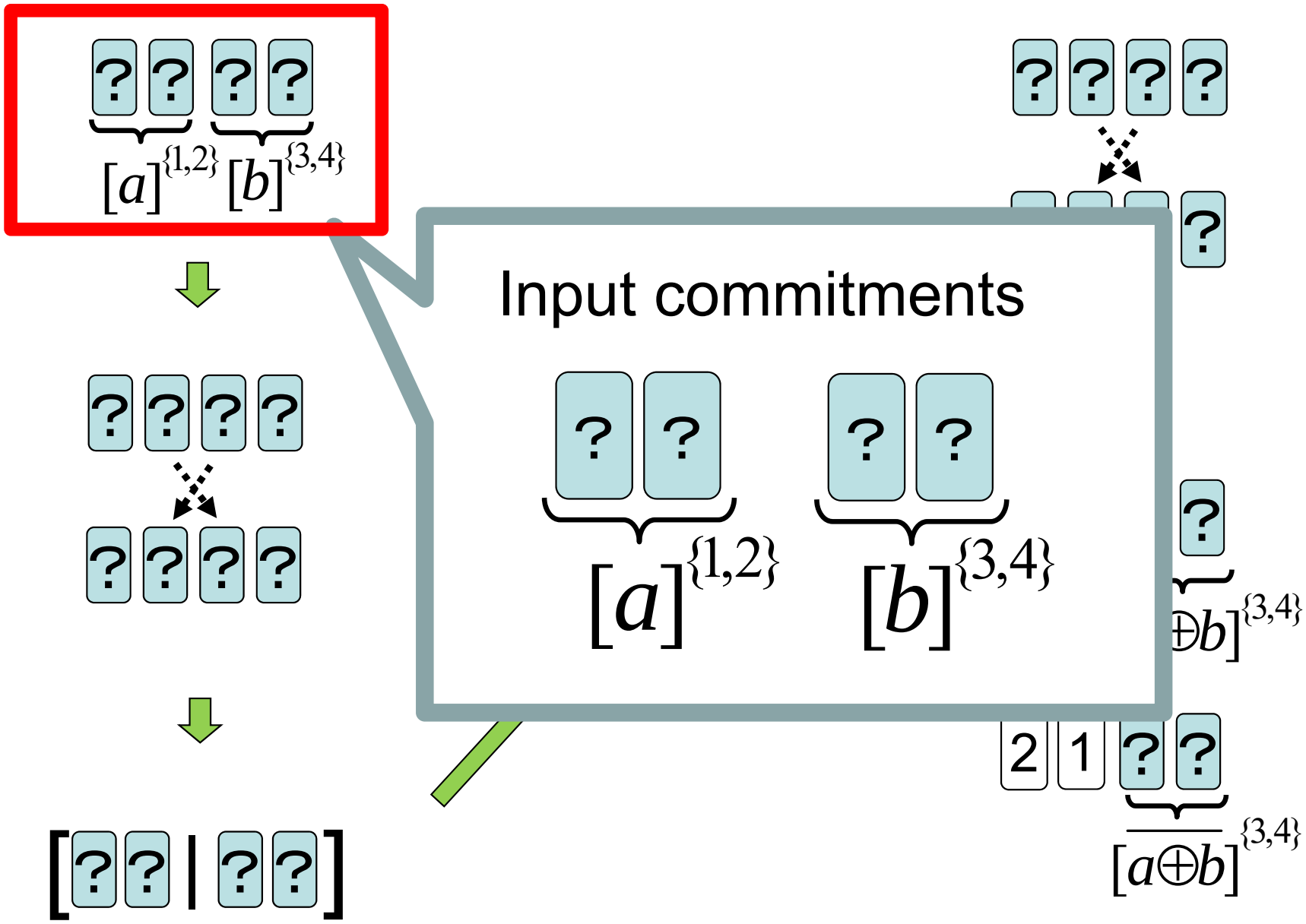
Idea

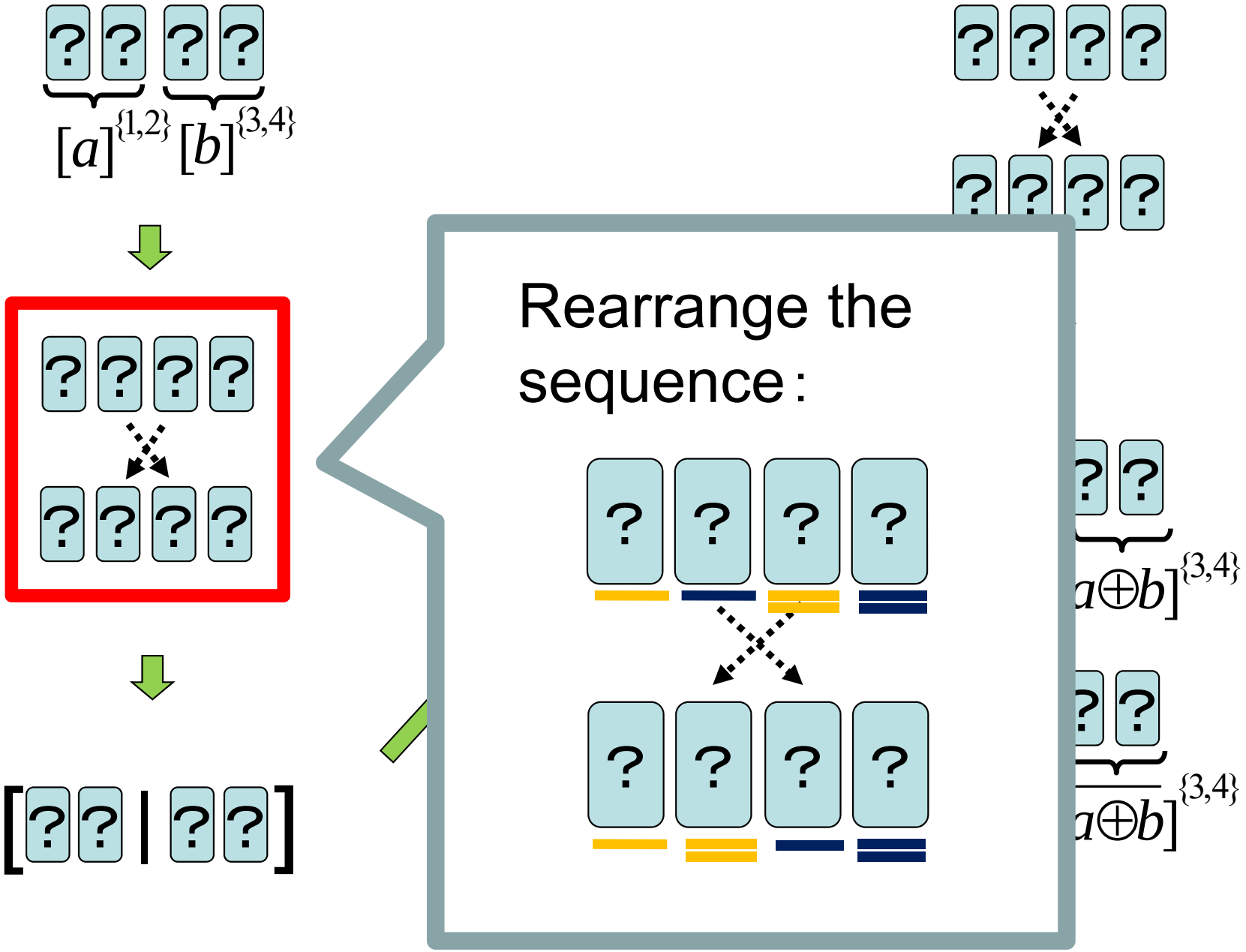
- ✓ Simulating the 4-custom-card XOR protocol [11]

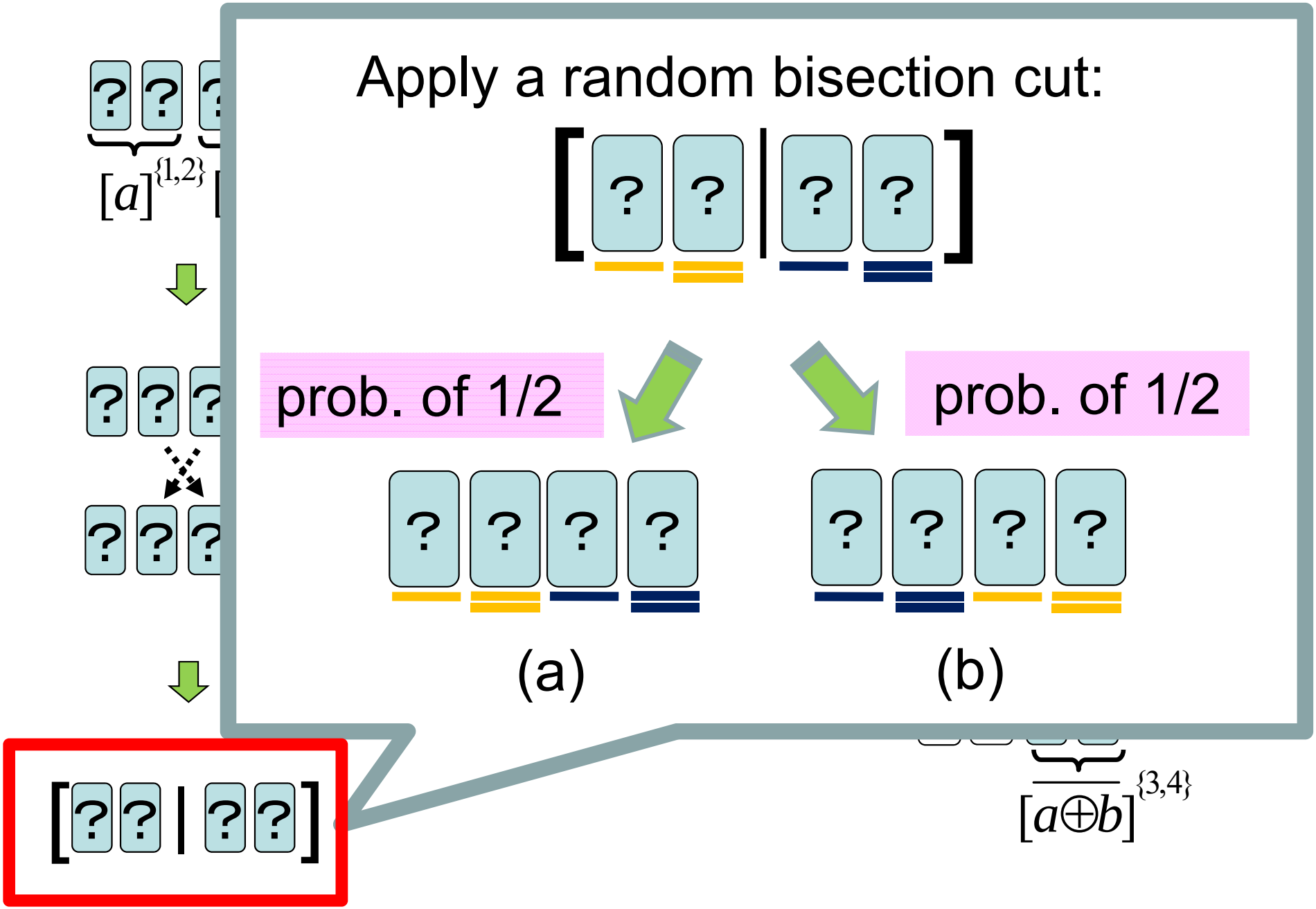


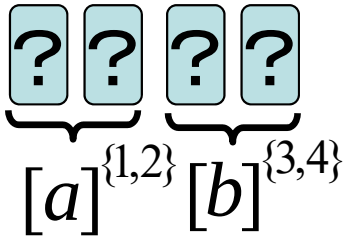
[11] T. Mizuki and H. Sone, Six-Card Secure AND and Four-Card Secure XOR, FAW 2009, LNCS 5598, pp. 358–369, 2009.

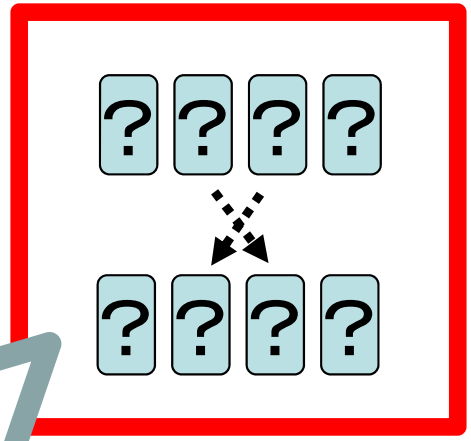

? ? ? ?
[a]{1,2} [b]{3,4}
? ? ? ?
? ? ? ?
[? ? | ? ?]
? ? ? ?
? ? ? ?
1 2 ? ?
[a⊕b]{3,4}
2 1 ? ?
[a⊕b]{3,4}


[a]{1,2} [b]{3,4}
Input commitments
[a]{1,2}
[b]{3,4}
⊕b]{3,4}
2 1
[a⊕b]{3,4}


[a]{1,2} [b]{3,4}
[? ? | ? ?]
Rearrange the sequence :
a⊕b]{3,4}
a⊕b]{3,4}

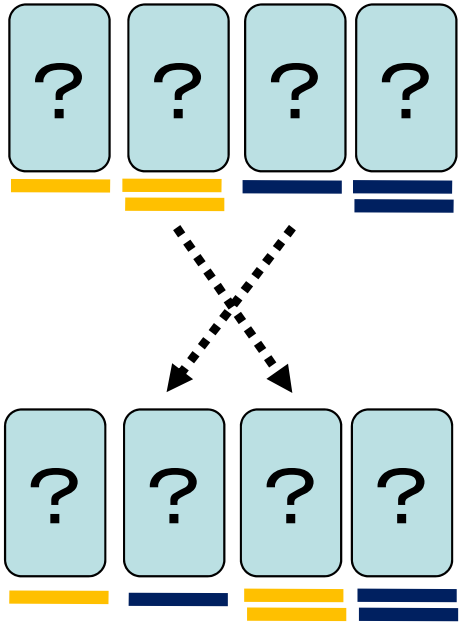

Apply a random bisection cut:
prob. of 1/2
prob. of 1/2
(a)
(b)
$[a]^{\{1,2\}}$
$[a \oplus b]^{\{3,4\}}$



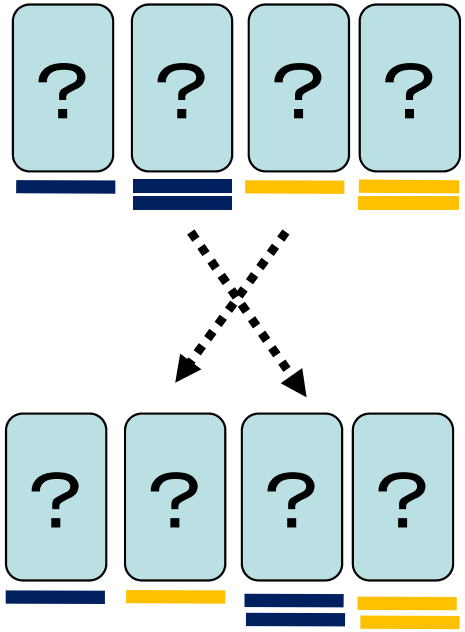


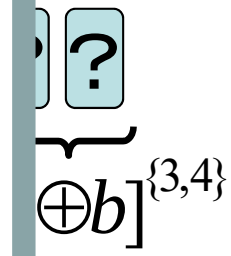
Rearrange the sequence:

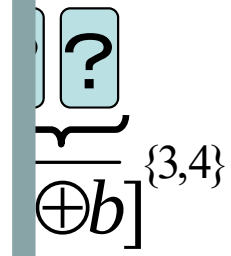
(a)

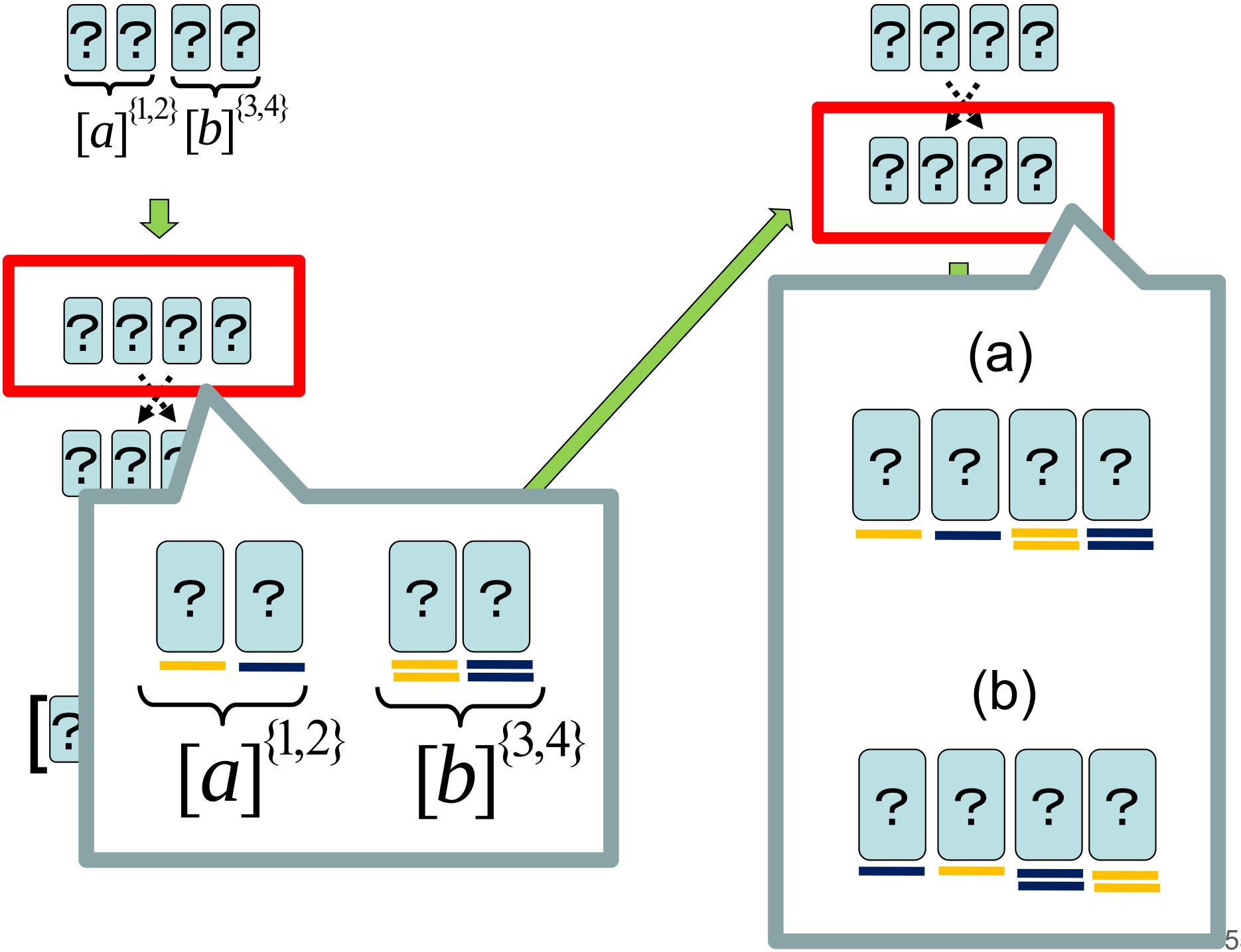


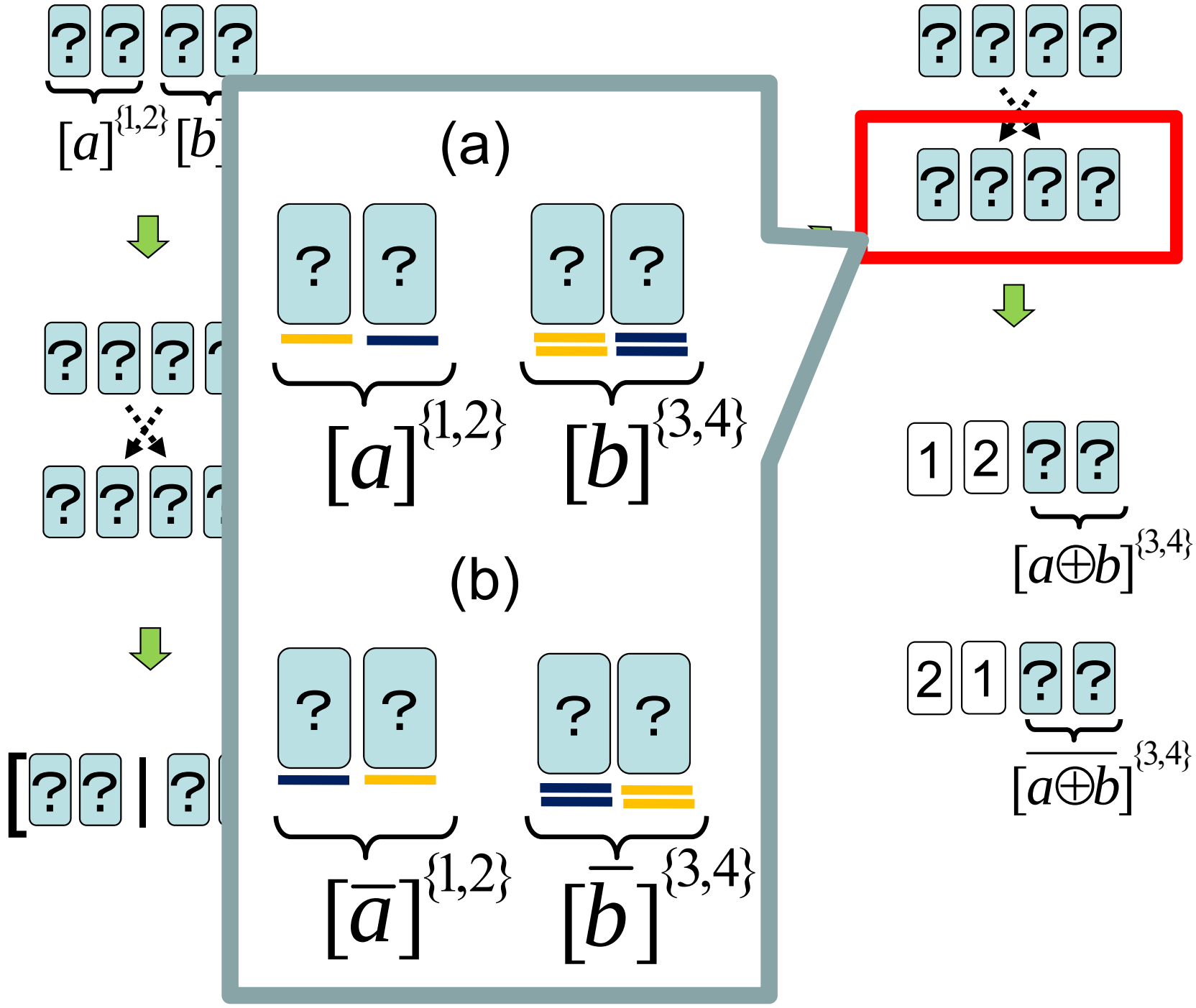
(b)





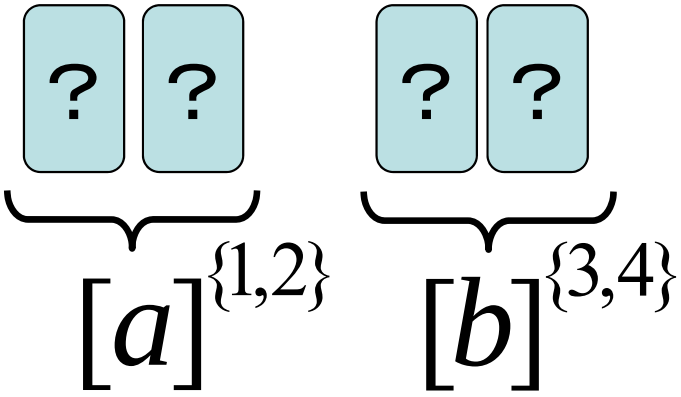



[a]{1,2} [b]{3,4}
[a]{1,2}
[b]{3,4}
(a)
(b)

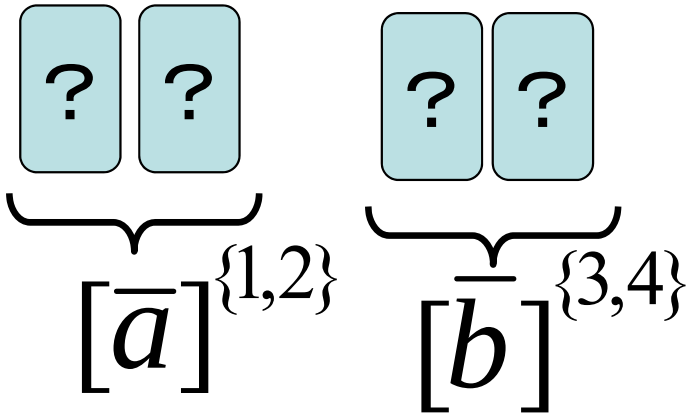

(a)
$[a]^{\{1,2\}}$
$[b]^{\{3,4\}}$
(b)
$[\overline{a}]^{\{1,2\}}$
$[\overline{b}]^{\{3,4\}}$
1 2
$[a \oplus b]^{\{3,4\}}$
2 1
$\overline{[a \oplus b]}^{\{3,4\}}$

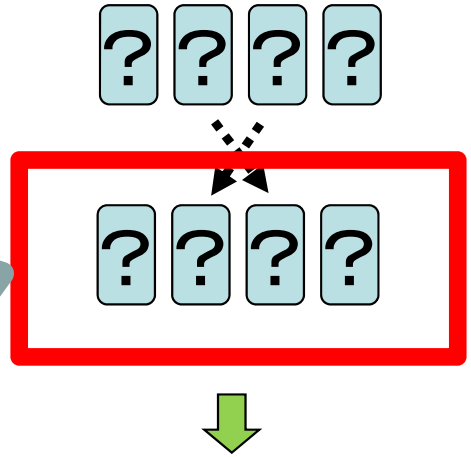
Reveal （no information leaks）

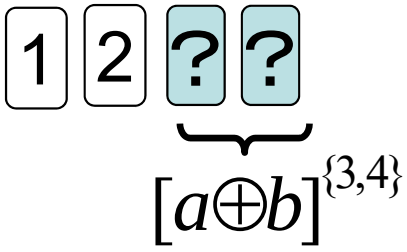
(a)

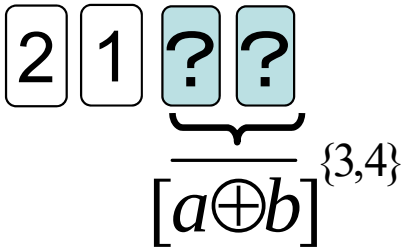


(b)

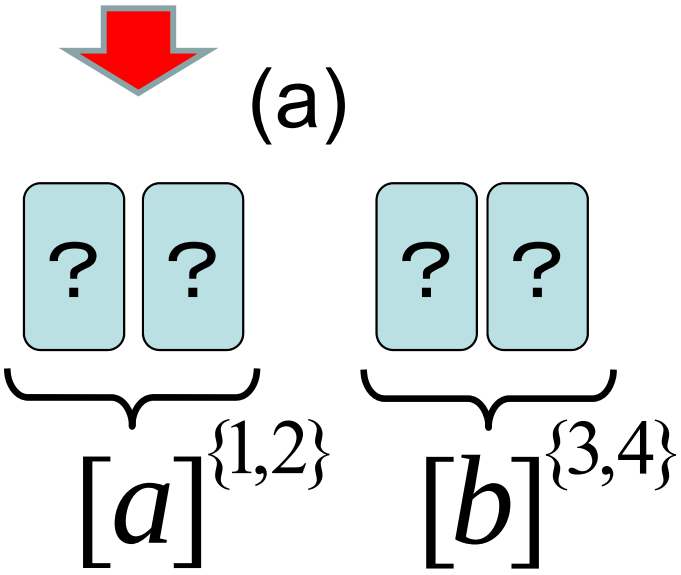


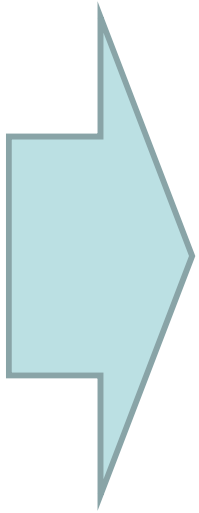


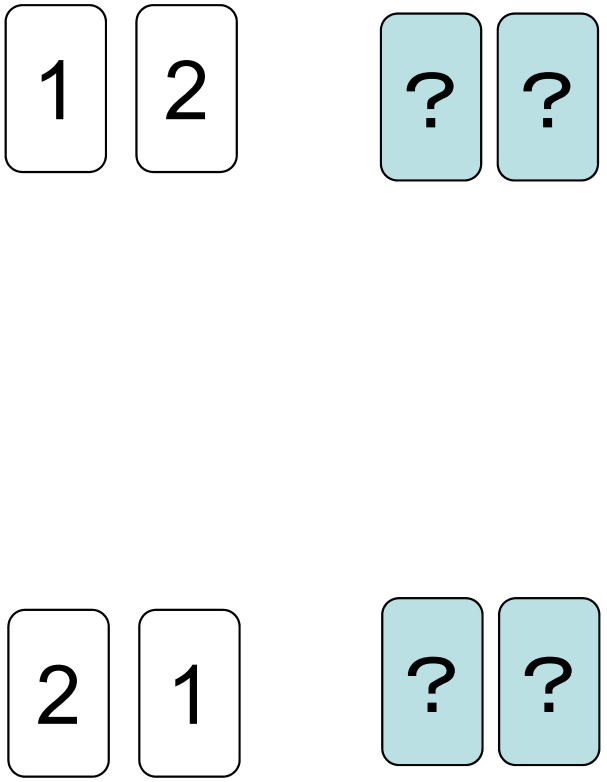


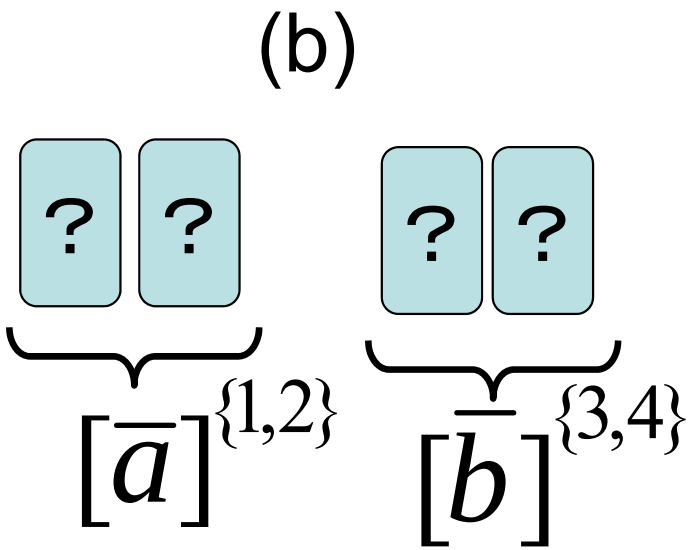


Reveal

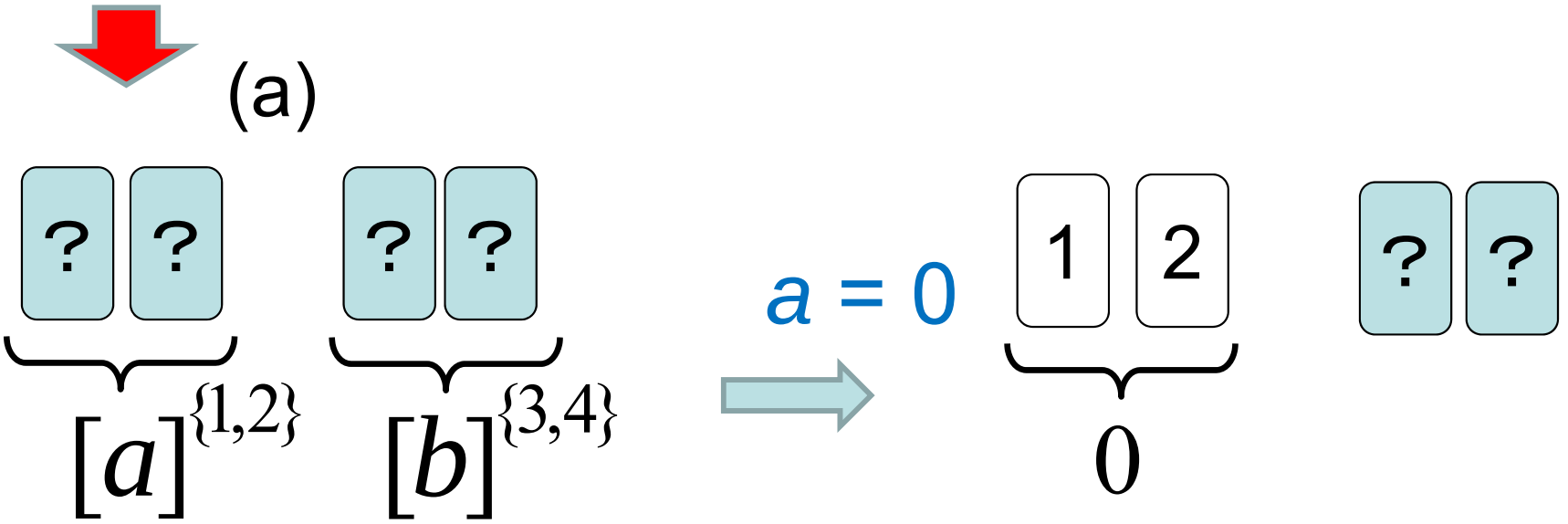






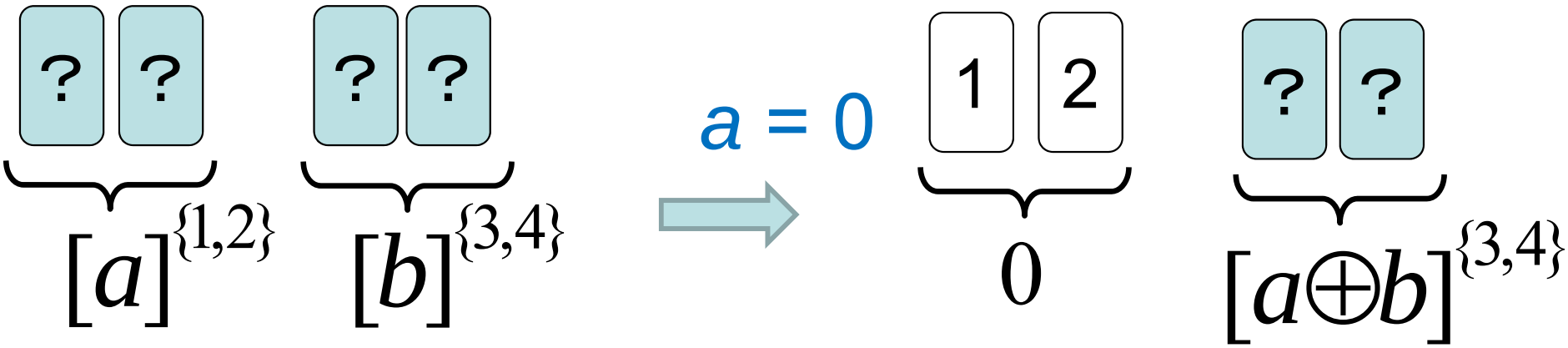


Reveal

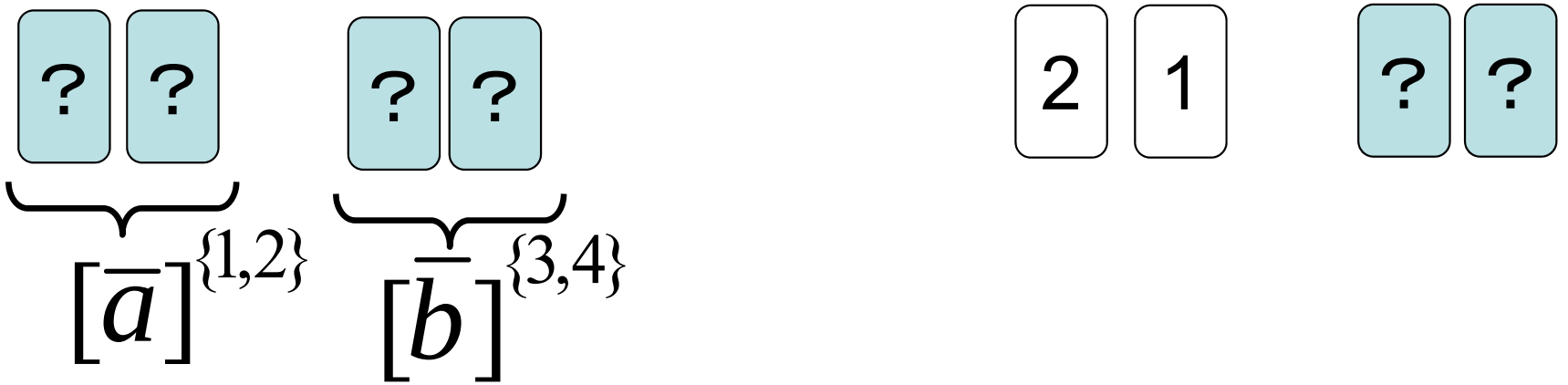


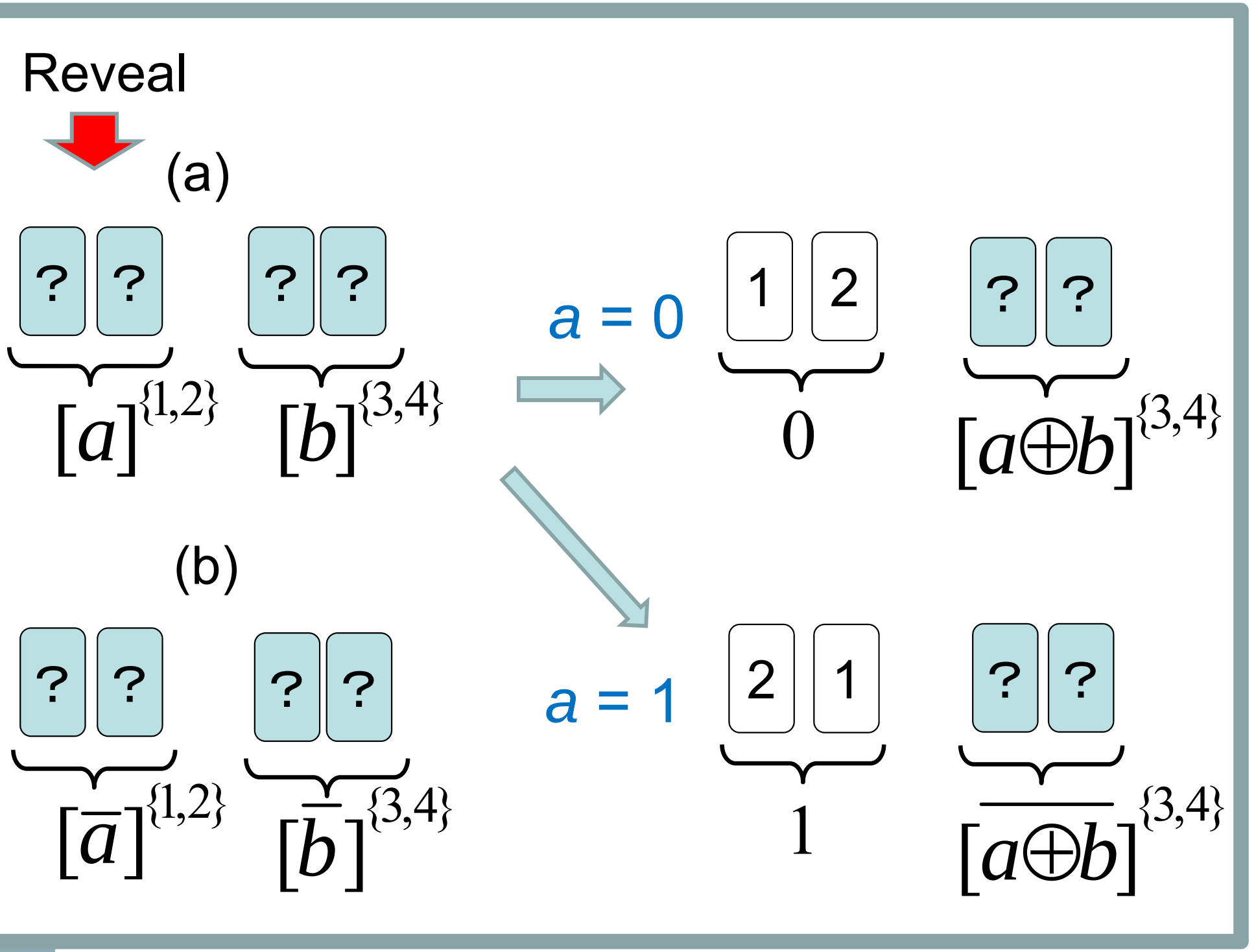
Reveal

(a)

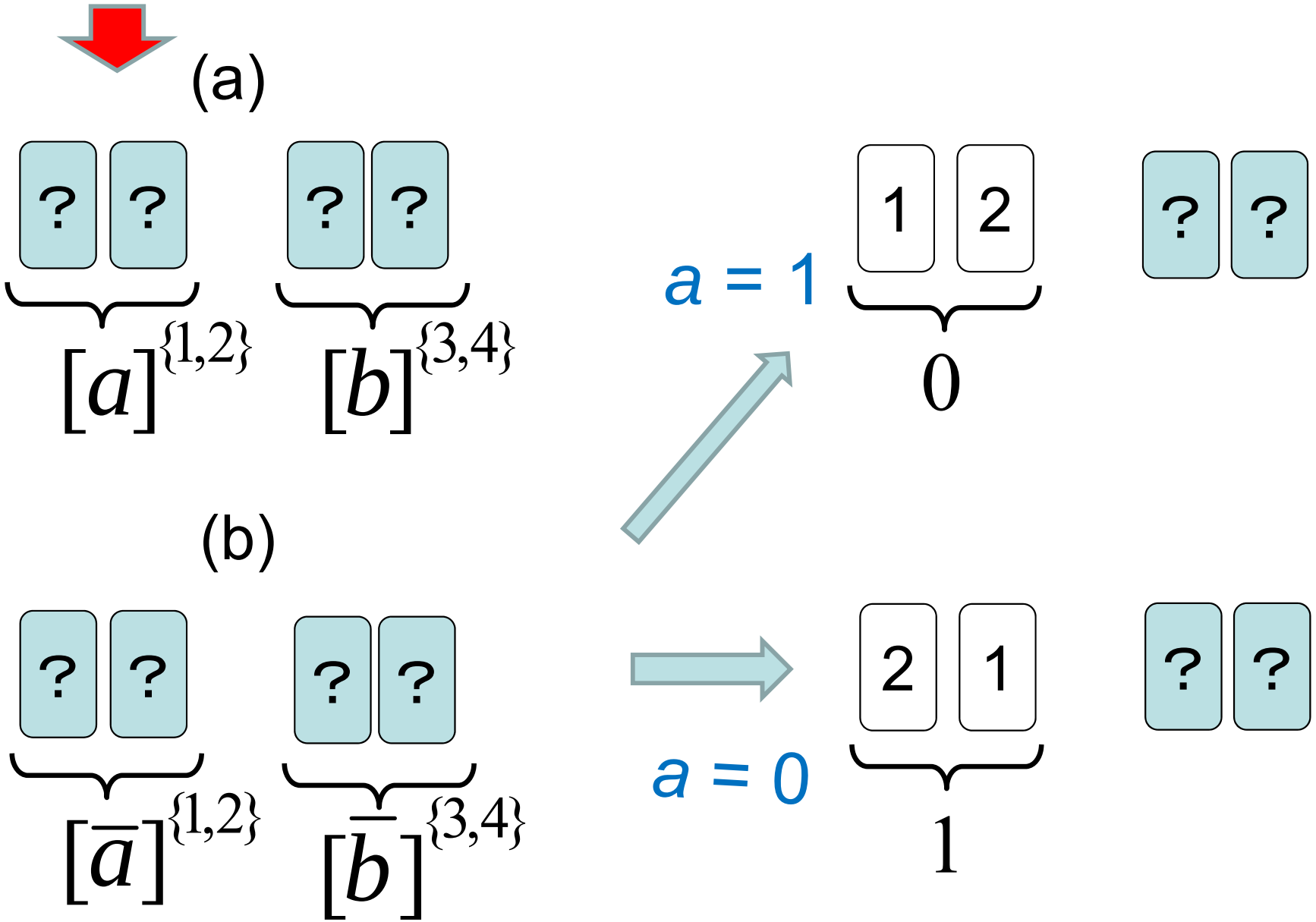


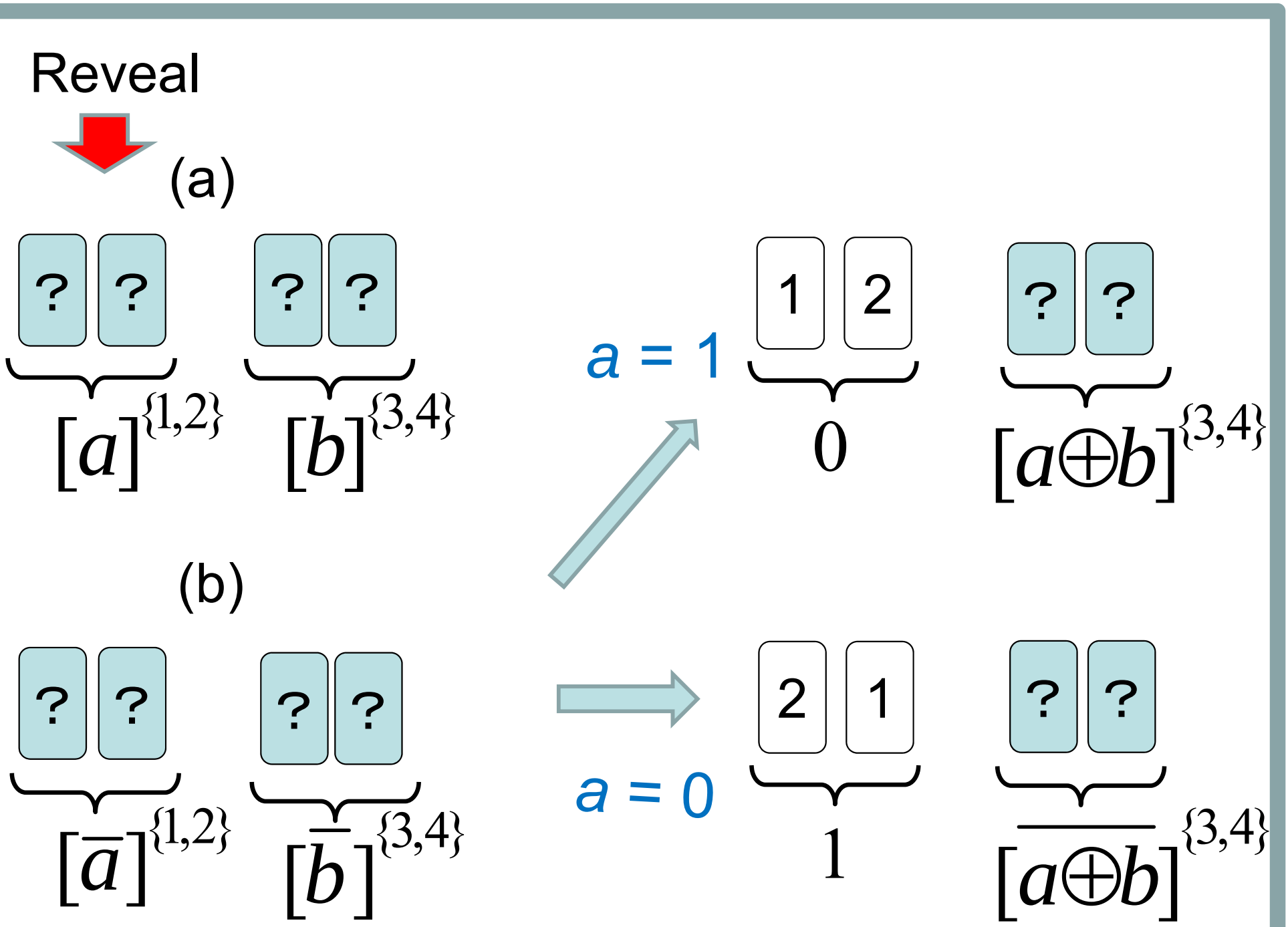
(b)

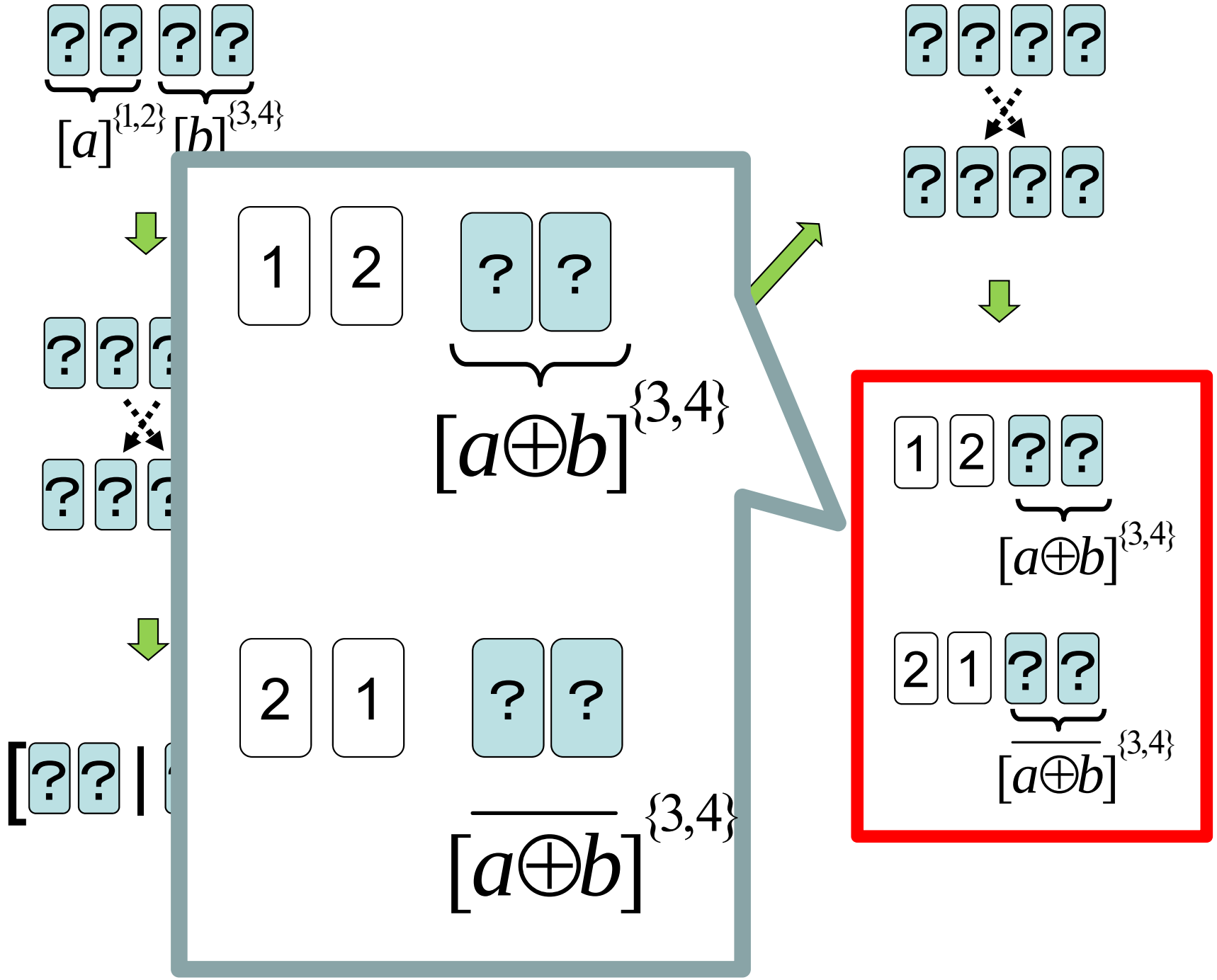


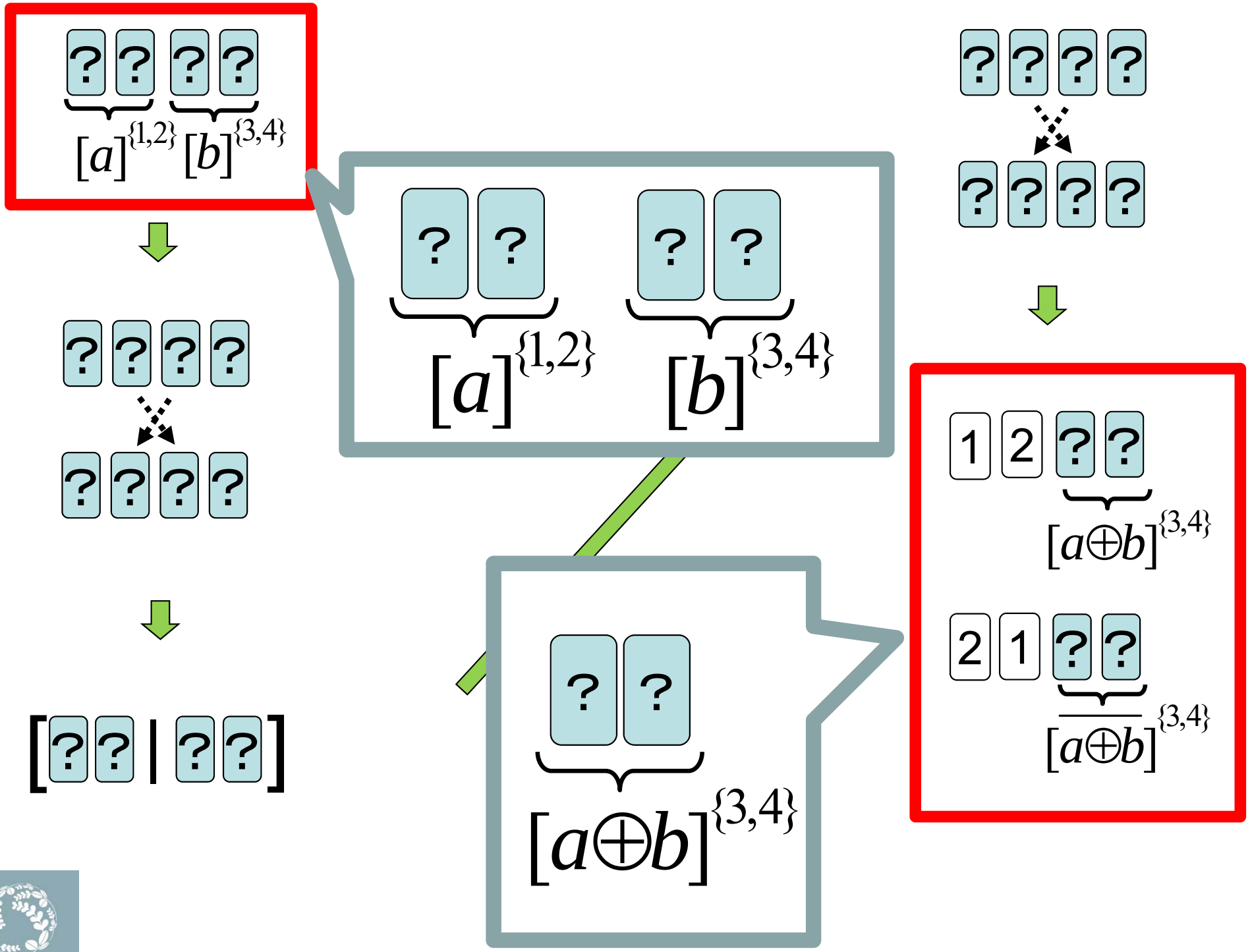

Reveal
(a)
? ?
? ?
[a]{1,2}
[b]{3,4}
a = 0
1 2
0
? ?
[a⊕b]{3,4}
(b)
? ?
? ?
[ā]{1,2}
[b̄]{3,4}
a = 1
2 1
1
? ?
[a⊕b]‾{3,4}

Reveal




Reveal
(a)
? ?
? ?
$[a]^{\{1,2\}}$
$[b]^{\{3,4\}}$
$a = 1$
1 2
0
? ?
$[a \oplus b]^{\{3,4\}}$
(b)
? ?
? ?
$[\overline{a}]^{\{1,2\}}$
$[b]^{\{3,4\}}$
$a = 0$
2 1
1
? ?
$[\overline{a \oplus b}]^{\{3,4\}}$


[a]{1,2}
[b]{3,4}
1 2 ? ?
[a⊕b]{3,4}
2 1 ? ?
[a⊕b]{3,4}
1 2 ? ?
[a⊕b]{3,4}
2 1 ? ?
[a⊕b]{3,4}


[a]{1,2} [b]{3,4}
[a]{1,2} [b]{3,4}
[a⊕b]{3,4}
1 2 [a⊕b]{3,4}
2 1 [a⊕b]{3,4}
[|]

AND computation	# of cards	# of shuffles	
		avg.	fixed
Niemi-Renvall [13]（Sect. 2）	5	9.5	0
Ours（Sect. 3）	8	0	4
XOR computation			
Niemi-Renvall [13]（Sect. 2）	4	7	0
Ours（Sect. 4）	4	0	1

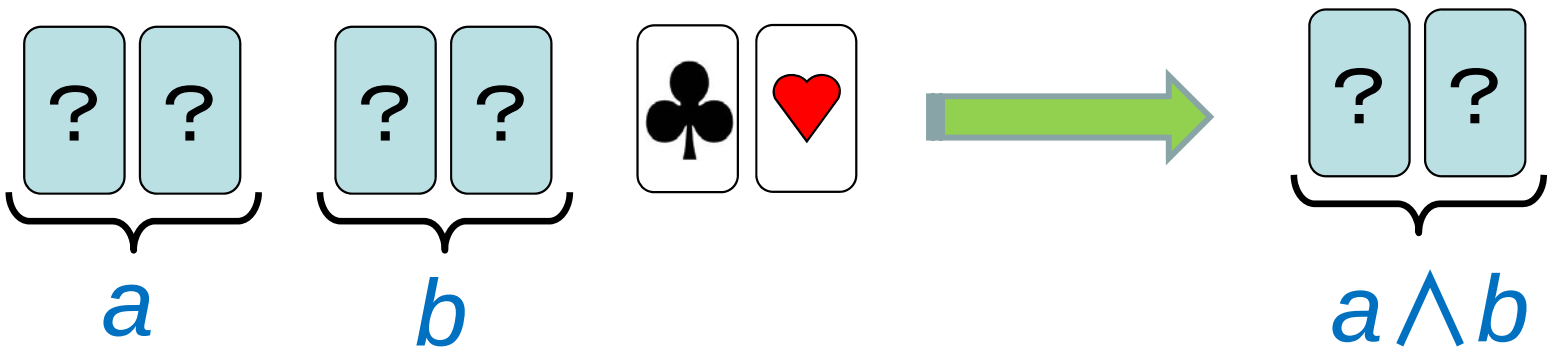
東北大学

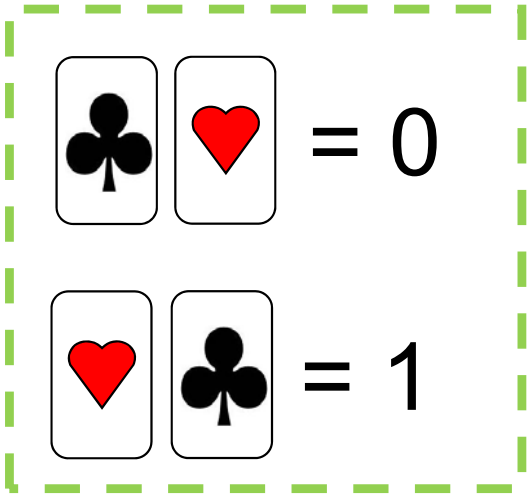
Contents

東北大学

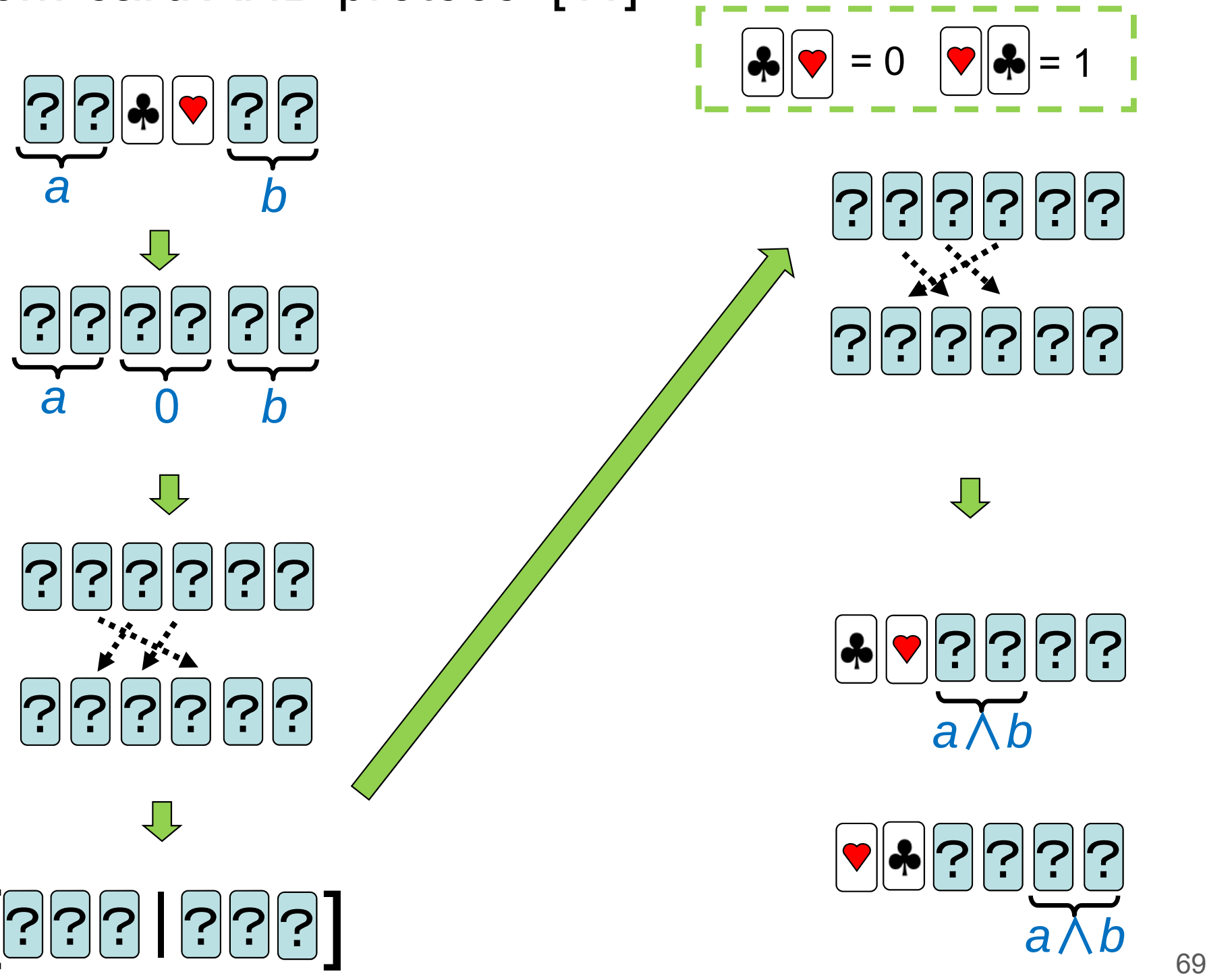
Outline

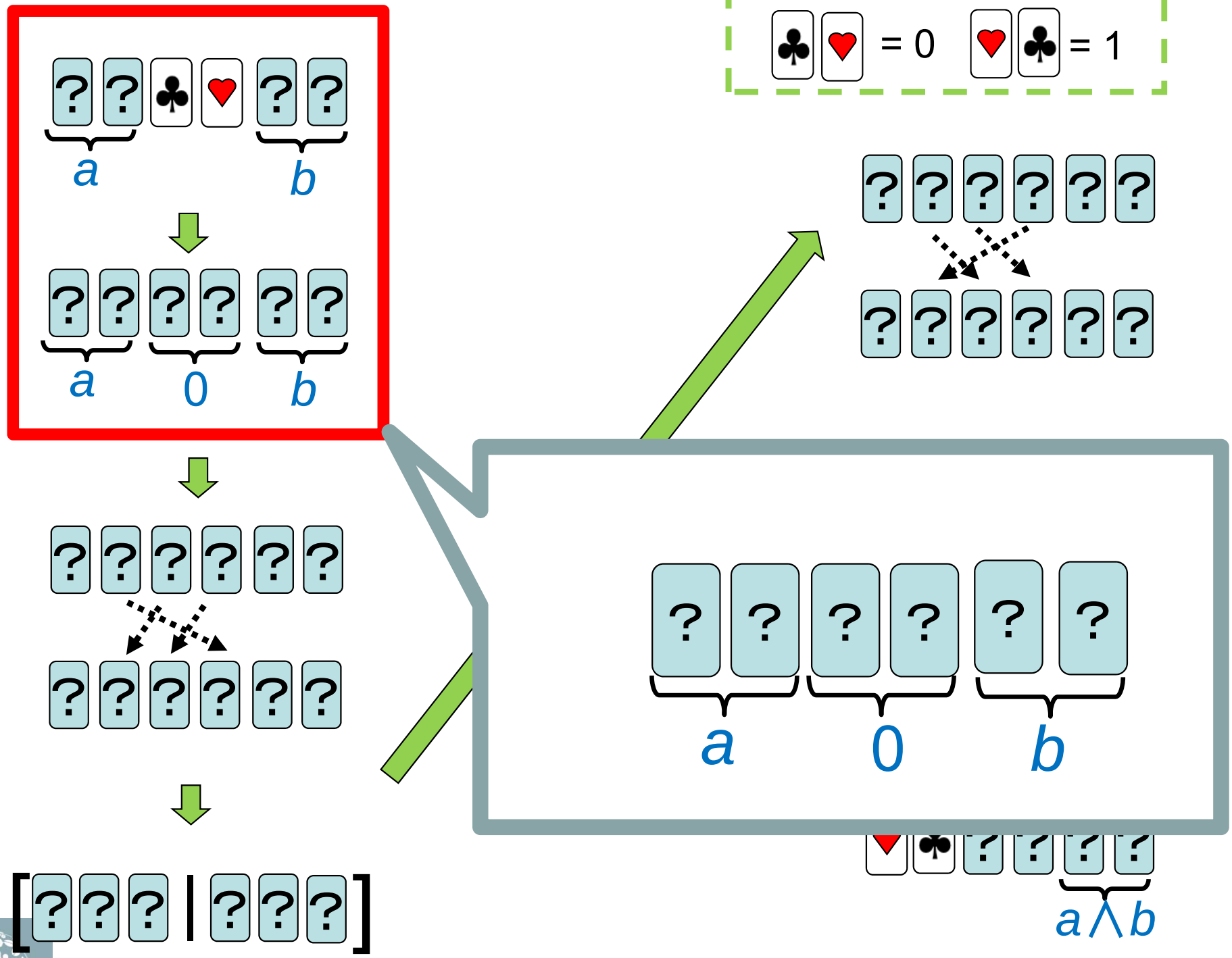
- ✓ Simulating the 6-custom-card AND protocol [11]
- ✓ Straightforward simulation does not work; we need some modification.



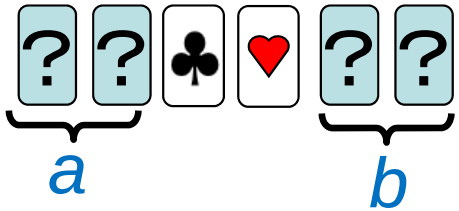


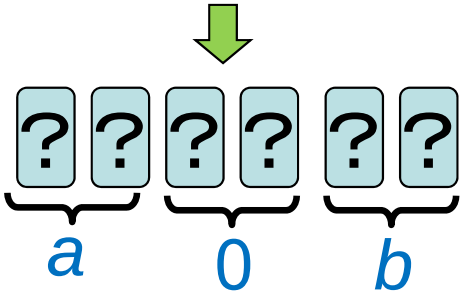
6-custom-card AND protocol [11]

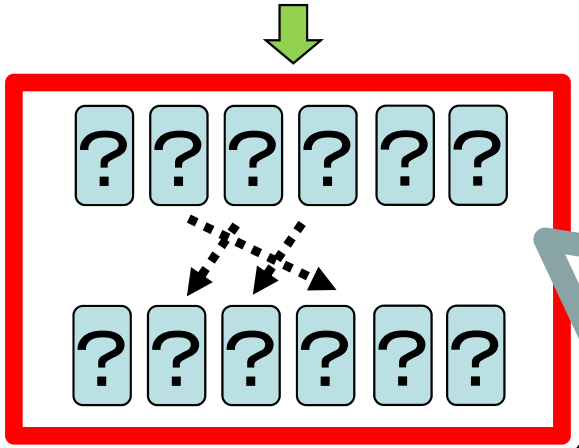


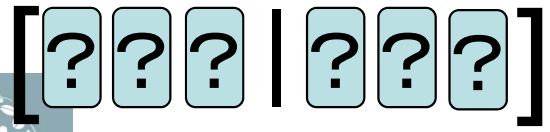


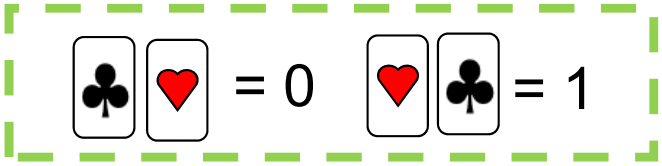
東北大学



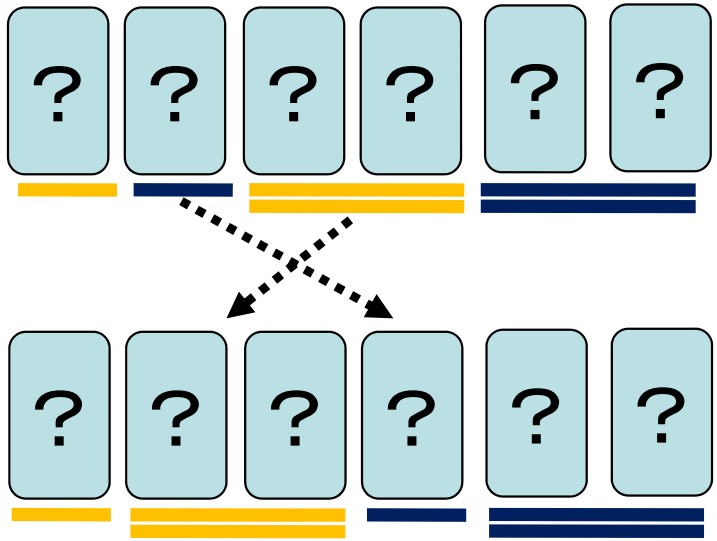




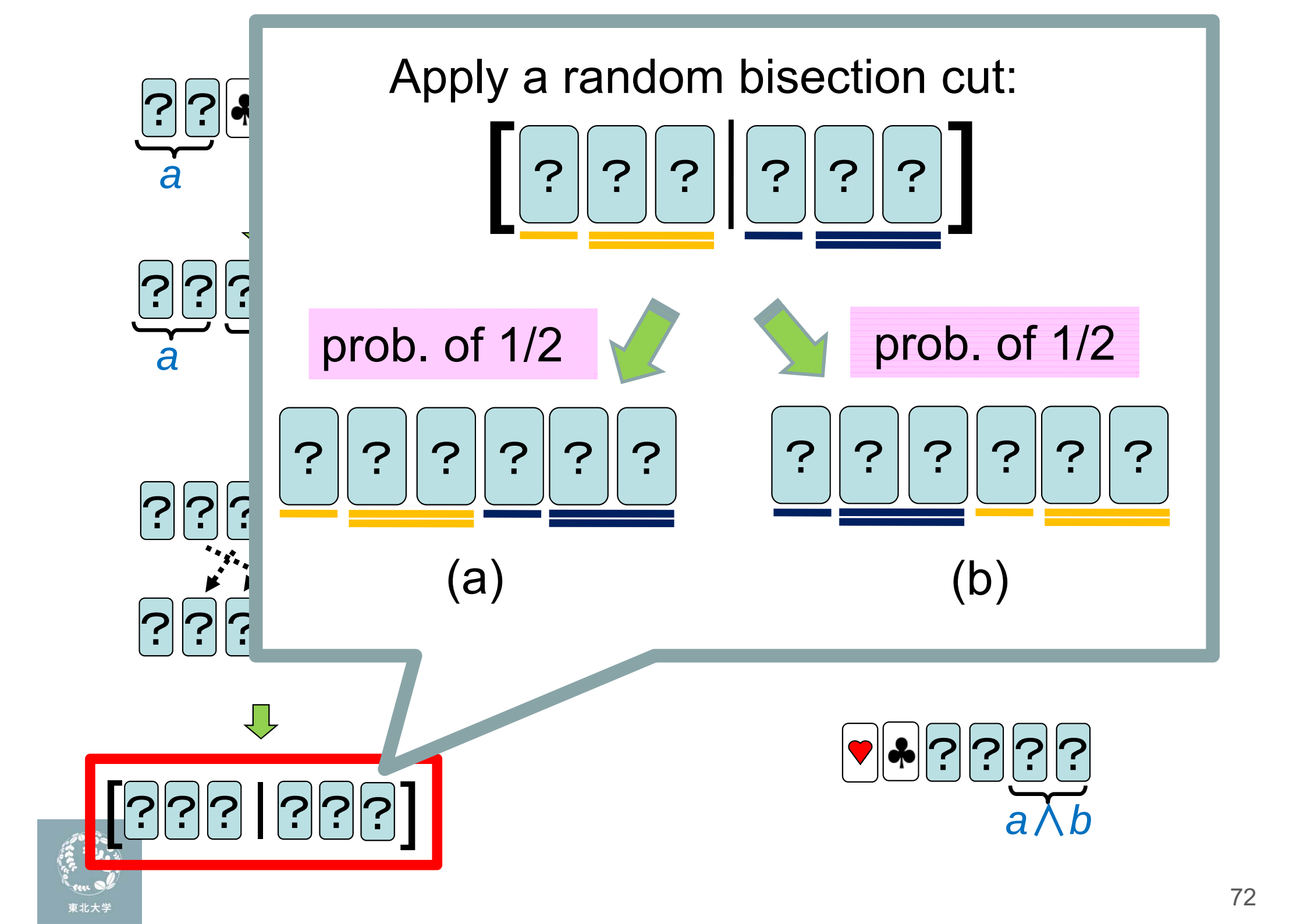


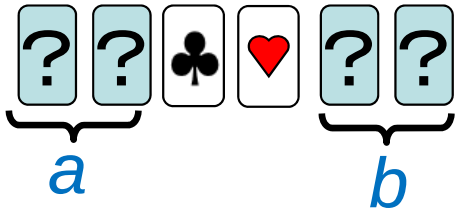


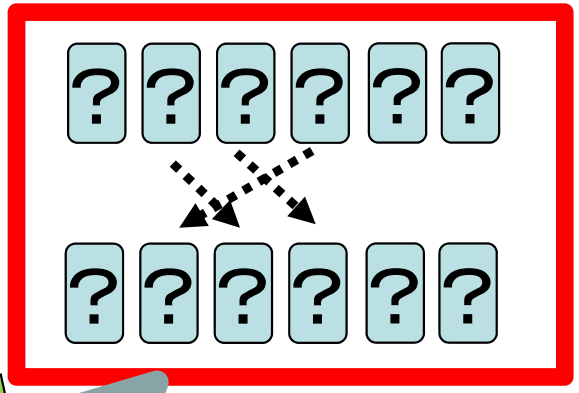
Rearrange the sequence:



$a \wedge b$

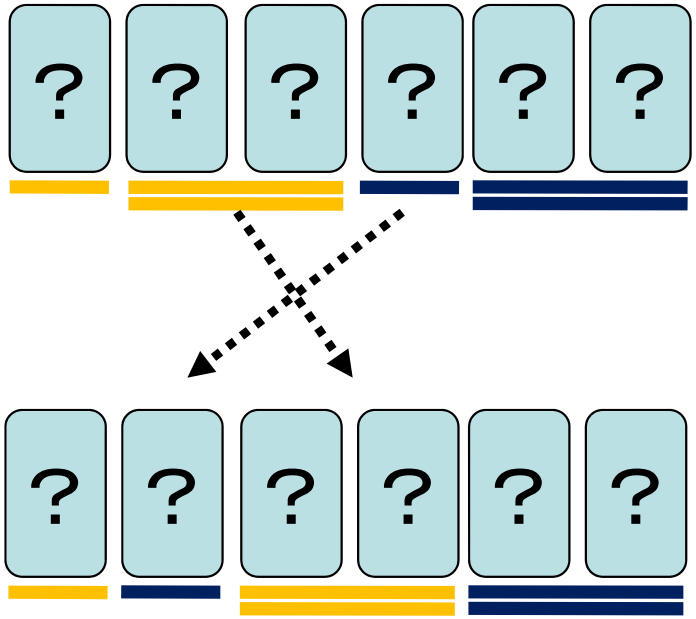

Apply a random bisection cut:
prob. of 1/2
prob. of 1/2
(a)
(b)
a
a
a ∧ b



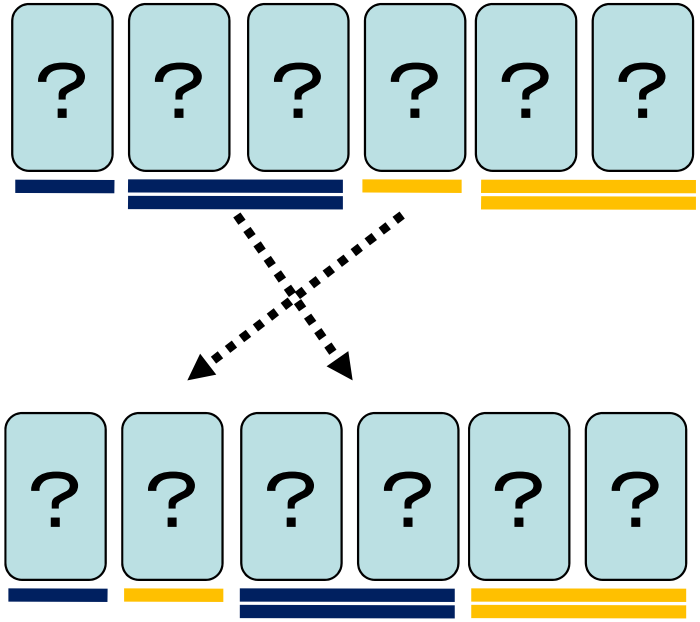


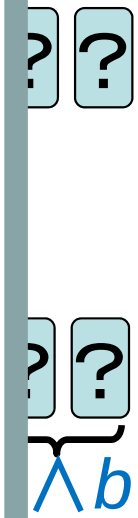
Rearrange the sequence:

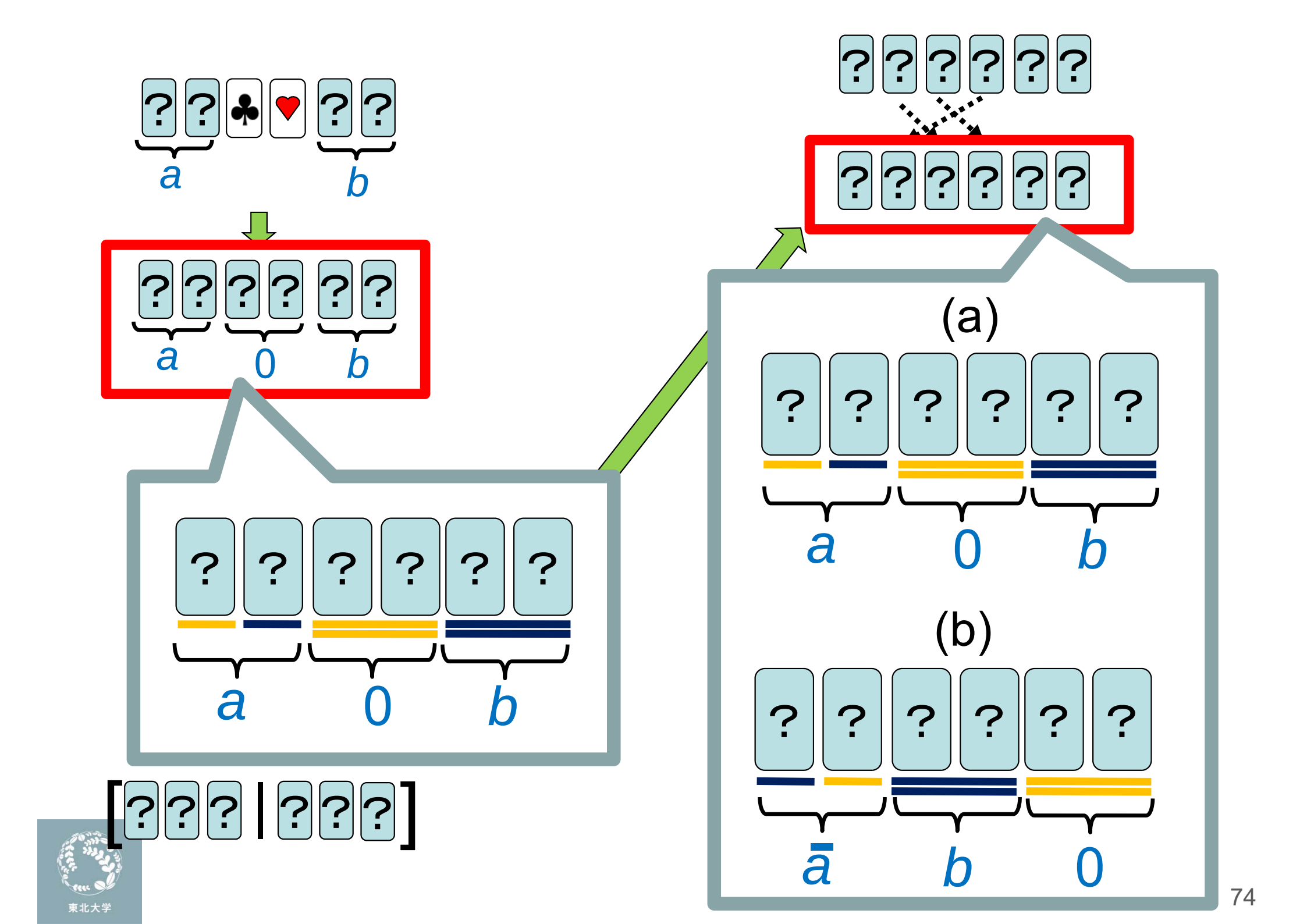
(a)



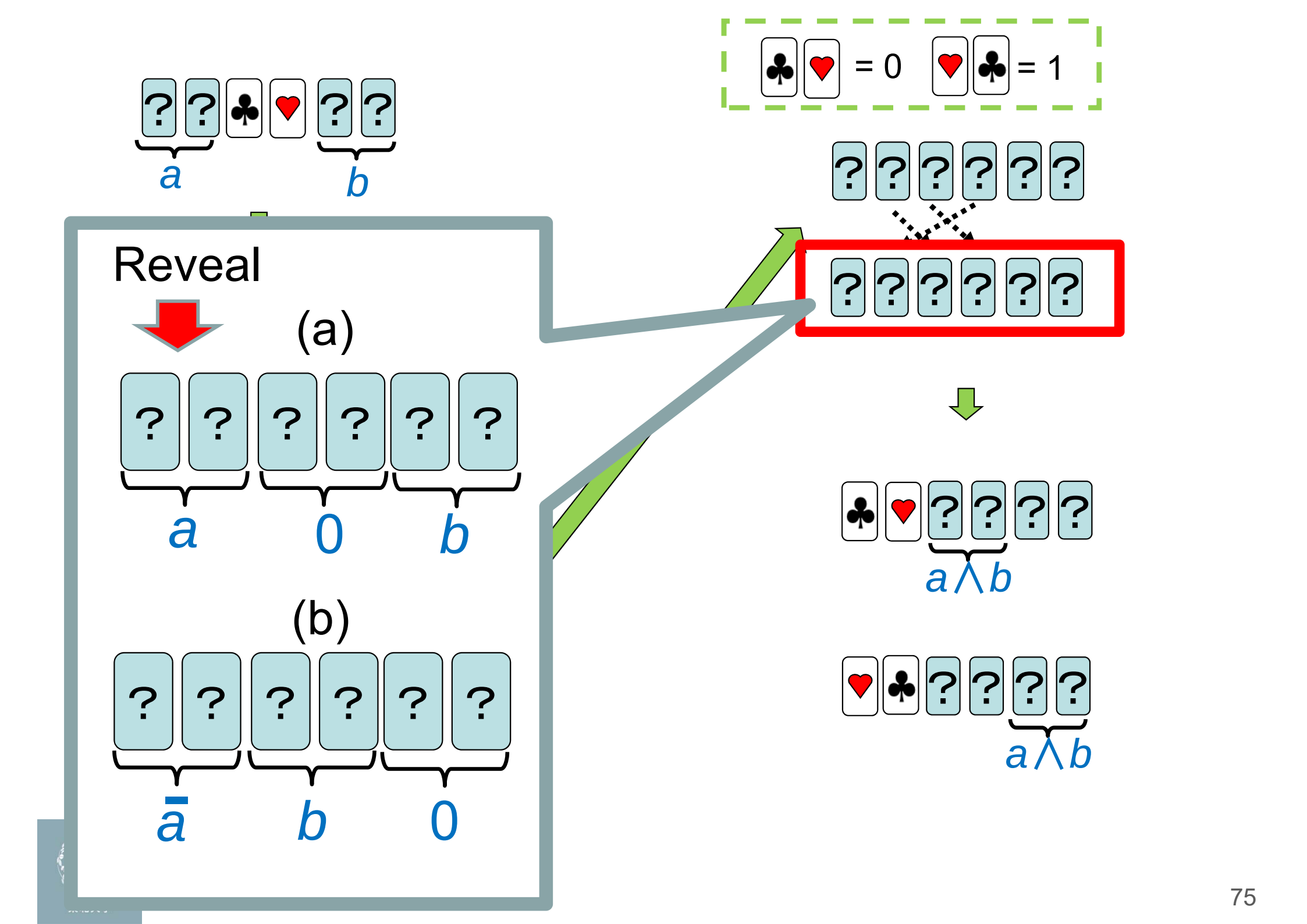
(b)

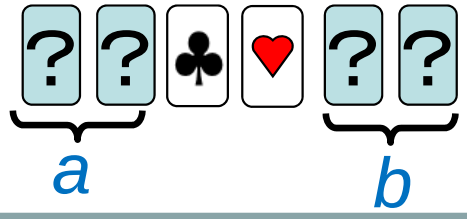


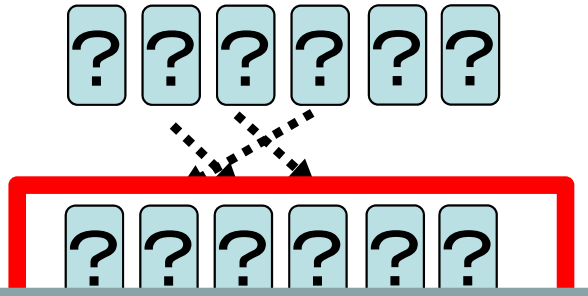




東北大学

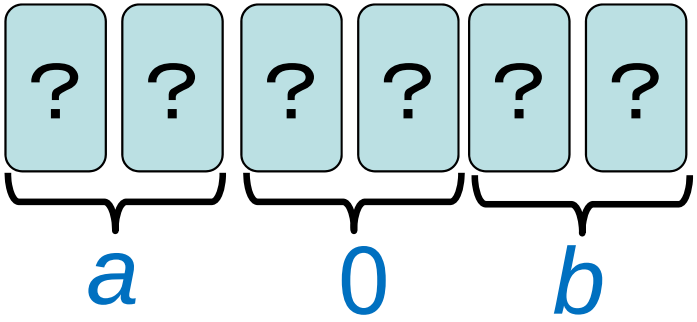

= 0
= 1
a
b
Reveal
(a)
a
0
b
(b)
ā
b
0
a ∧ b
a ∧ b



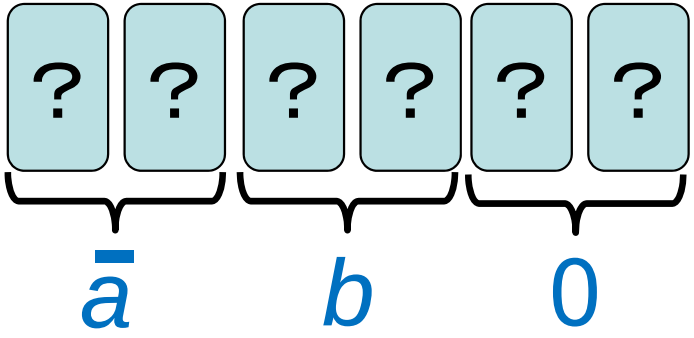


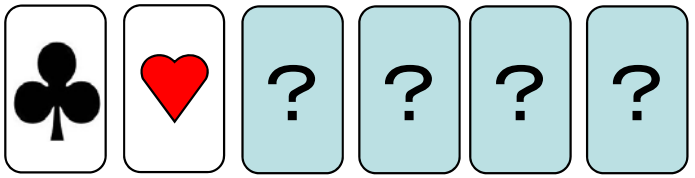
Reveal

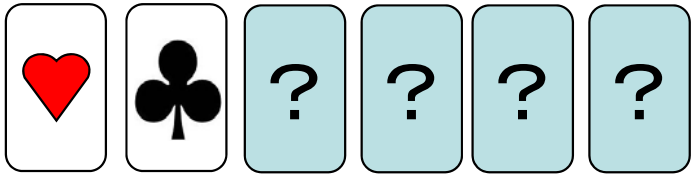
(a)

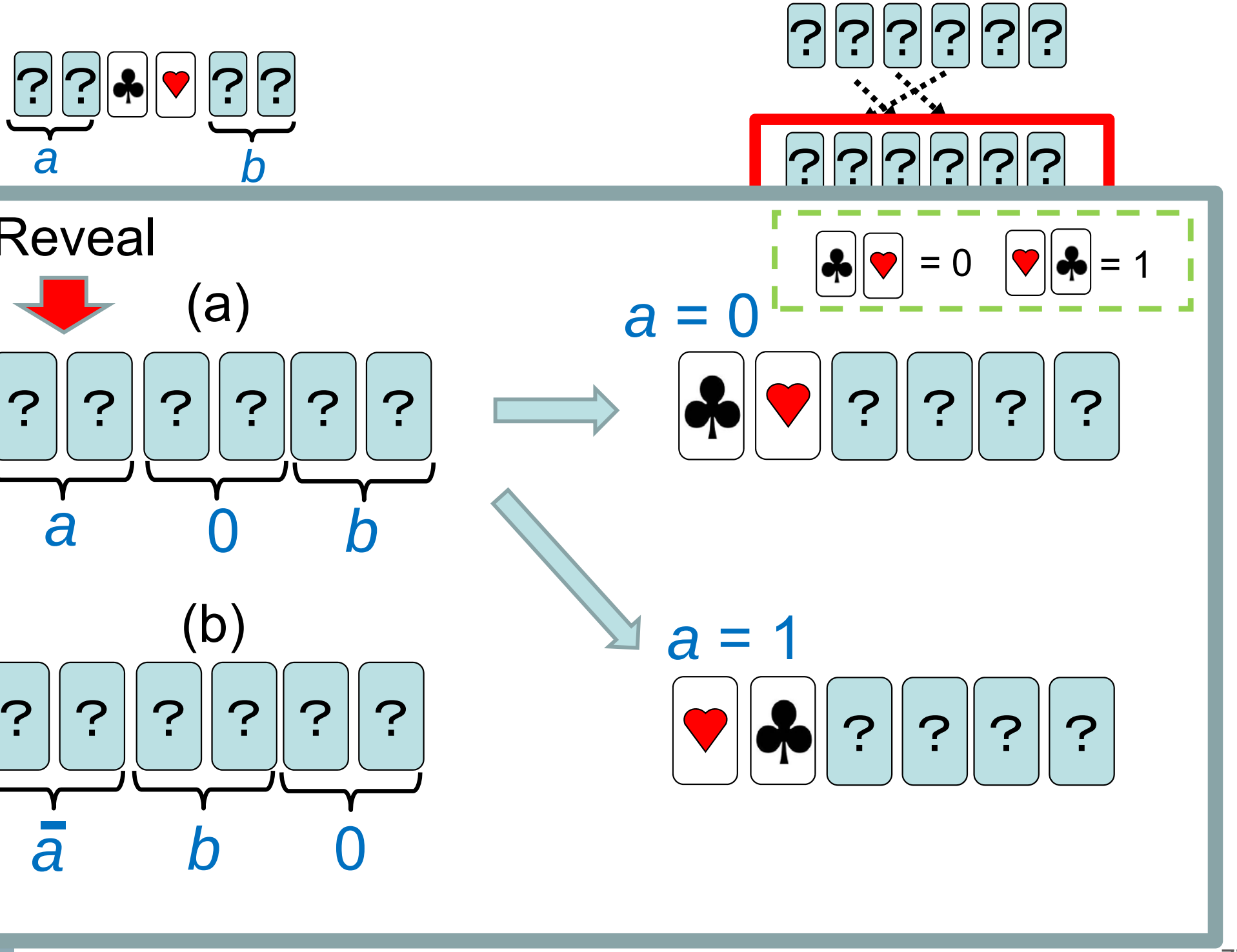


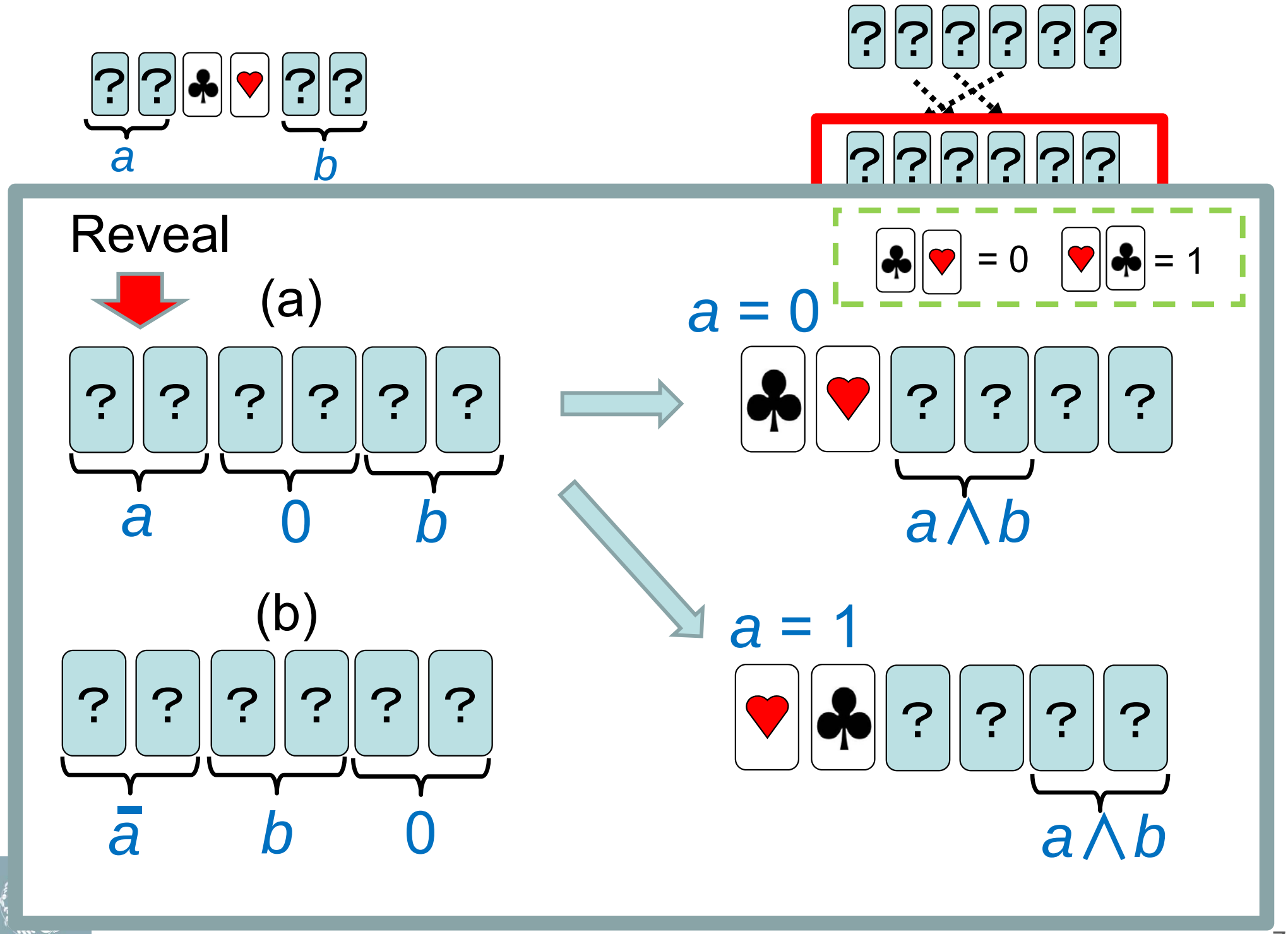
(b)

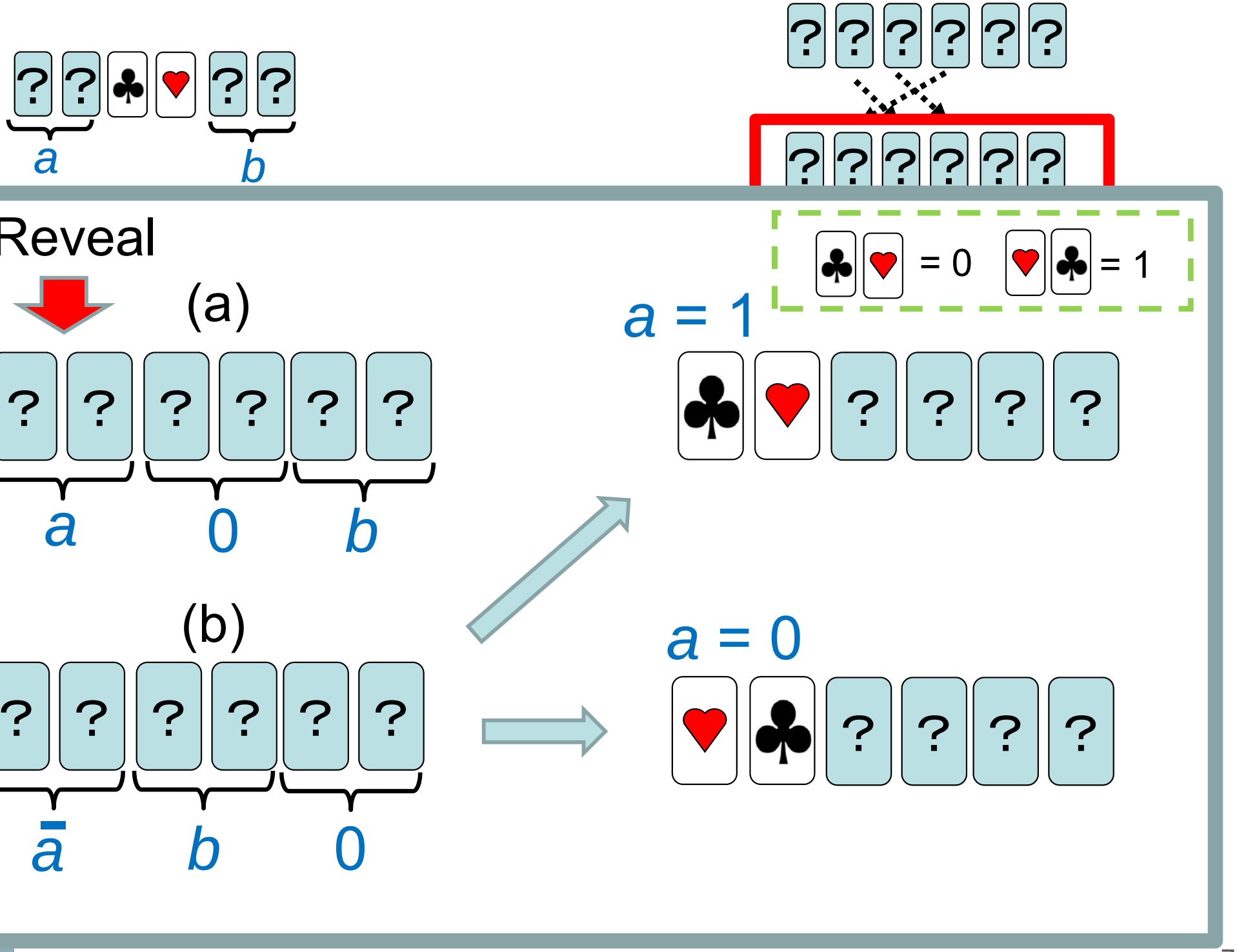


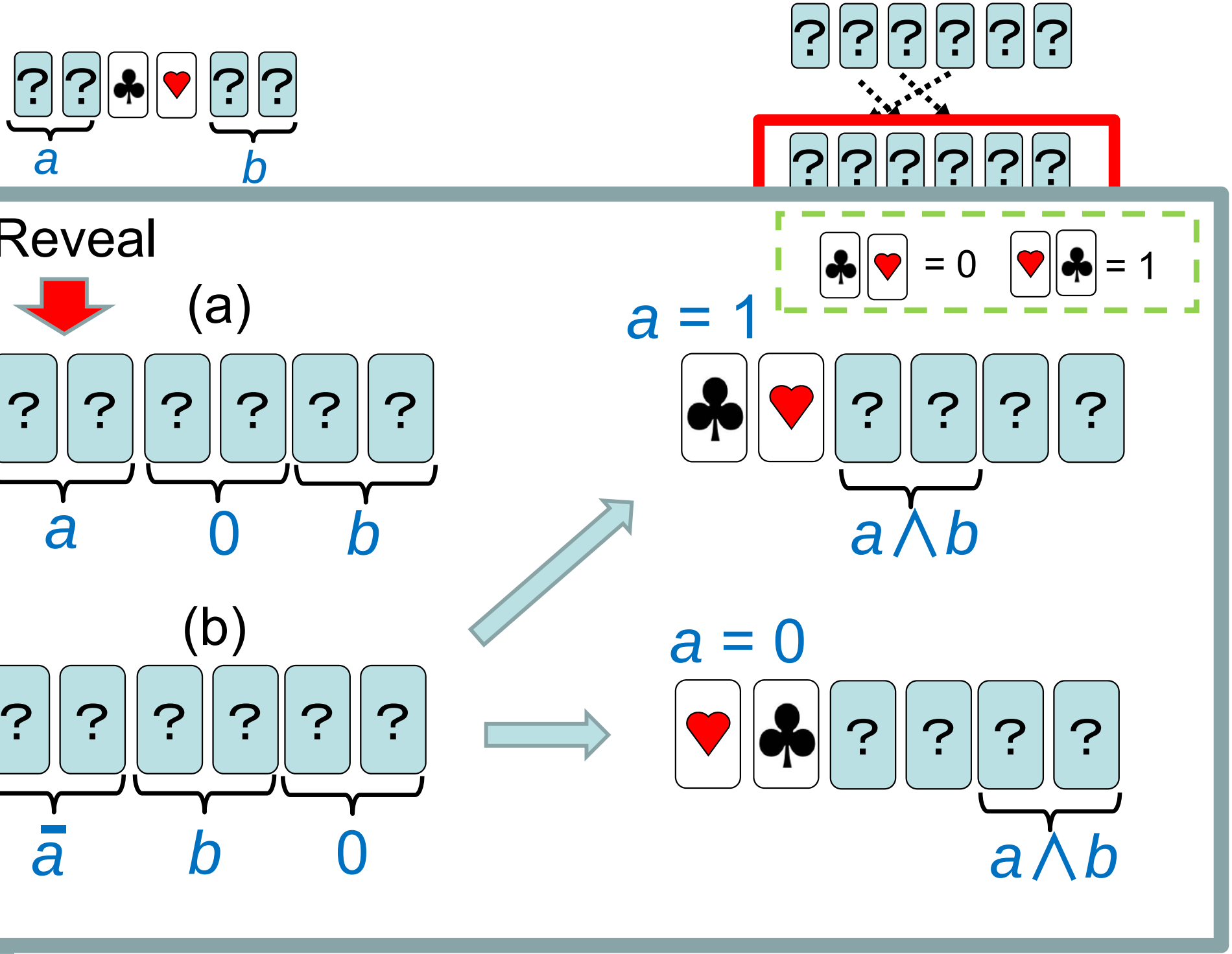


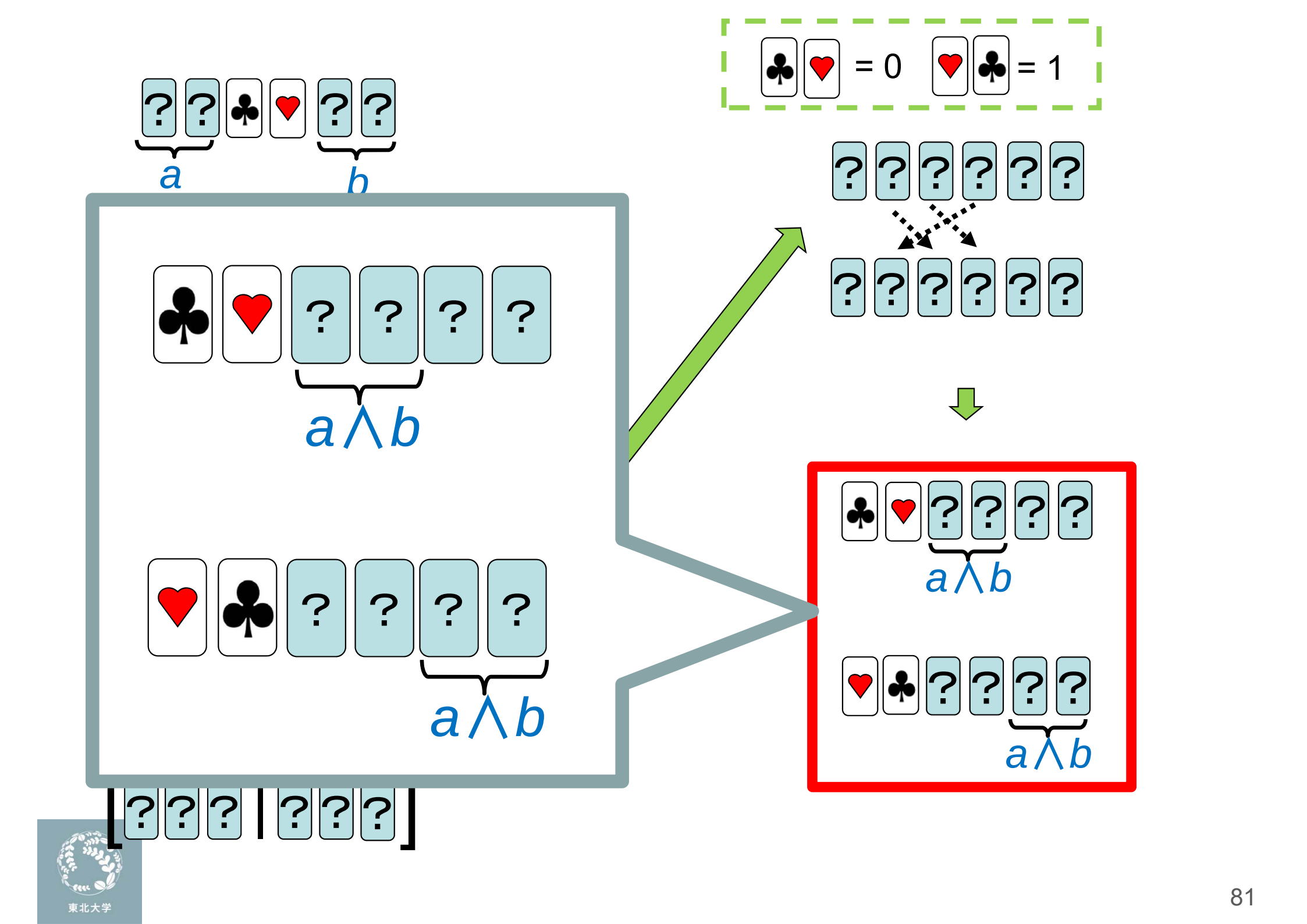


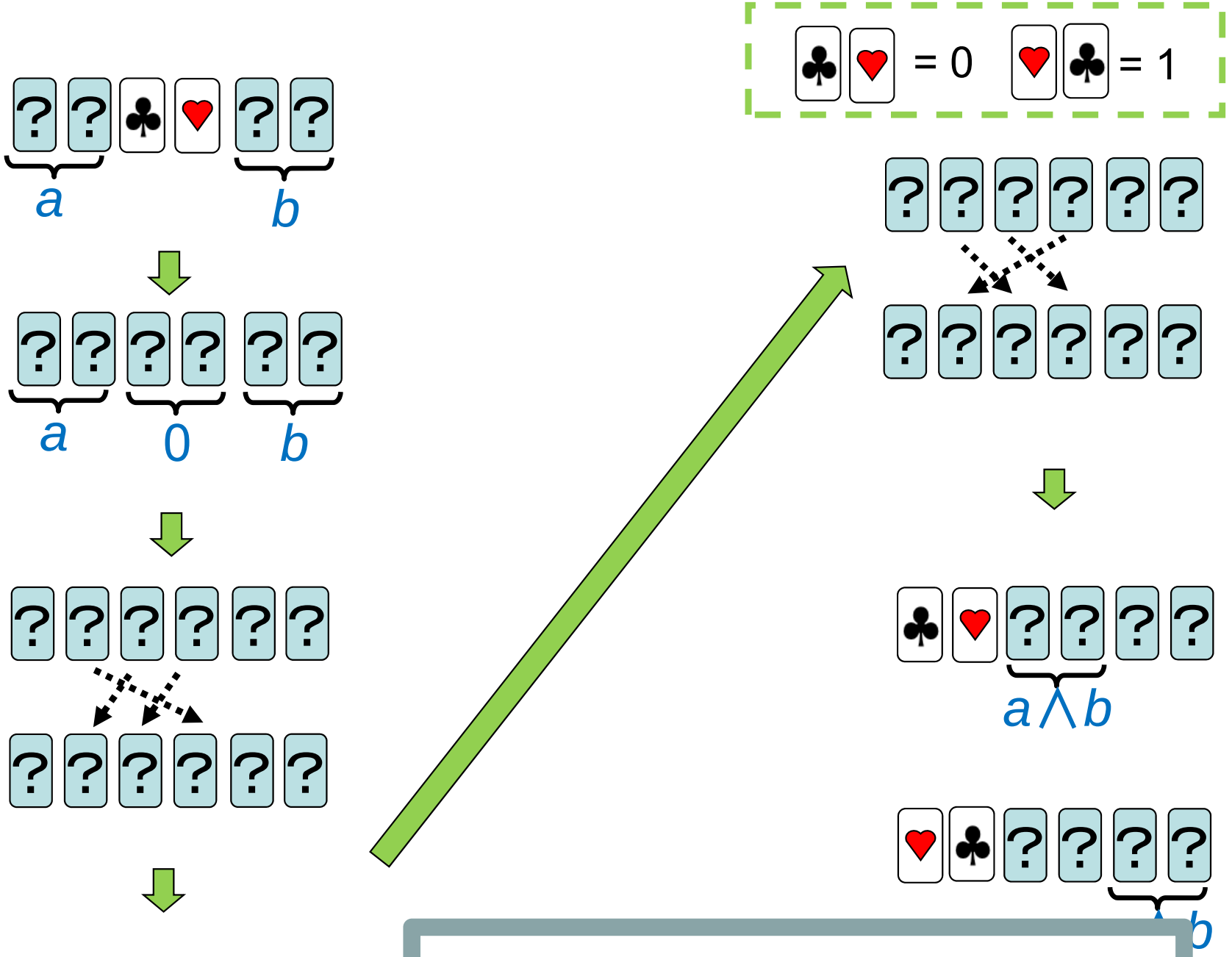

a
b
Reveal
(a)
a
0
b
(b)
ā
b
0
= 0
= 1
a = 0
a = 1


a
b
Reveal
(a)
a
0
b
(b)
ā
b
0
= 0
= 1
a = 0
a ∧ b
a = 1
a ∧ b

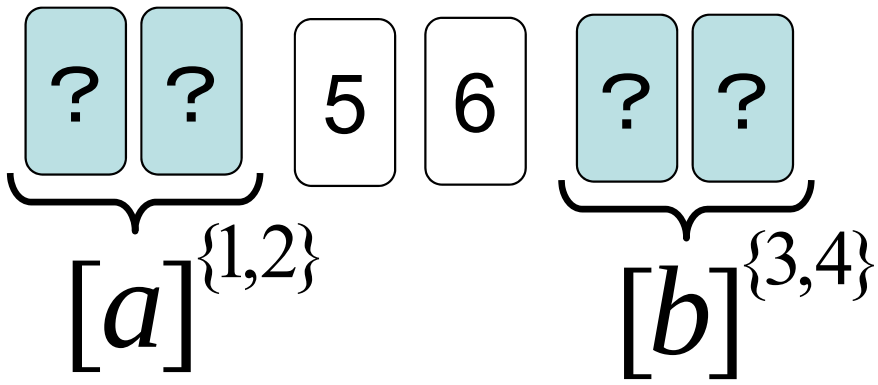

a
b
Reveal
(a)
a
0
b
(b)
ā
b
0
= 0
= 1
a = 1
a = 0


a
b
Reveal
(a)
a
0
b
(b)
ā
b
0
= 0
= 1
a = 1
a ∧ b
a = 0
a ∧ b

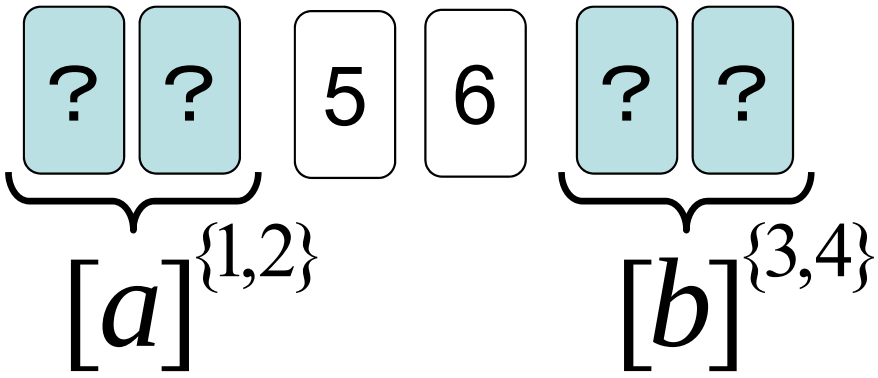

= 0
= 1
a
b
a∧b
a∧b
a∧b
a∧b

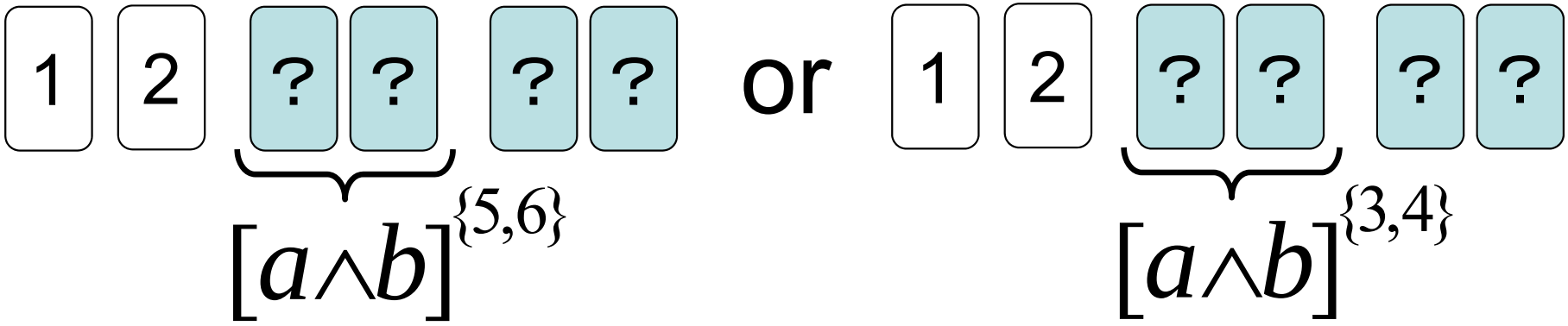


What if we apply this to a standard deck of cards?

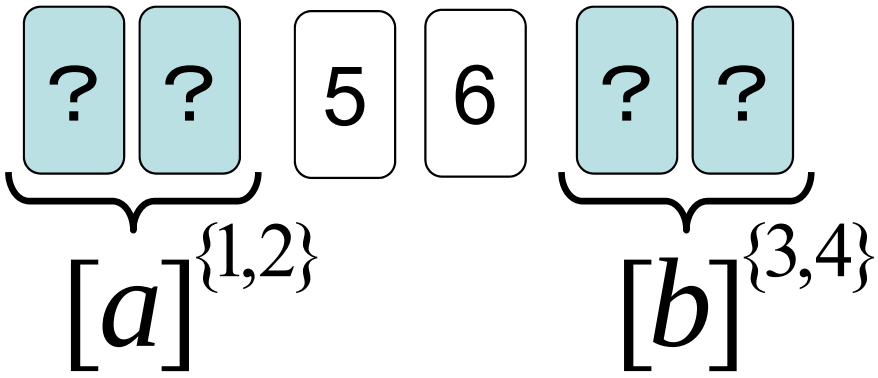


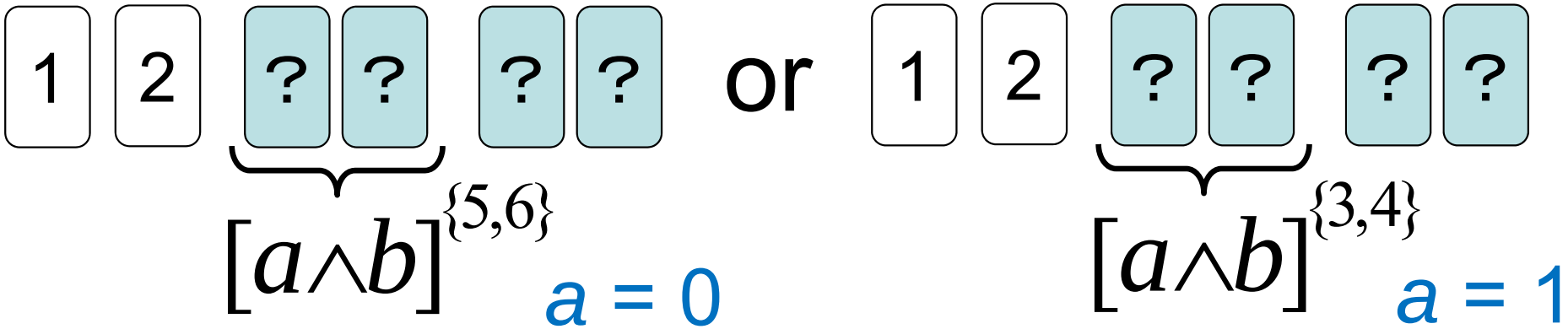
straightforward simulation





東北大学

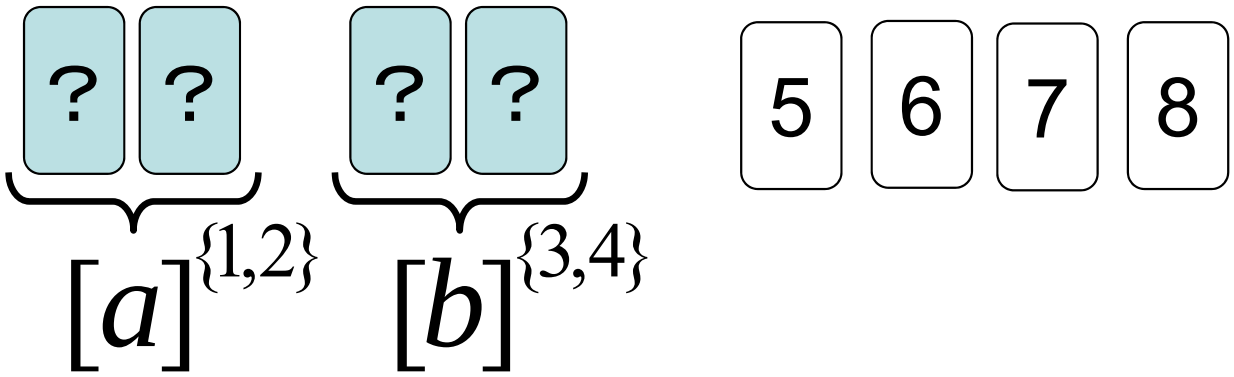




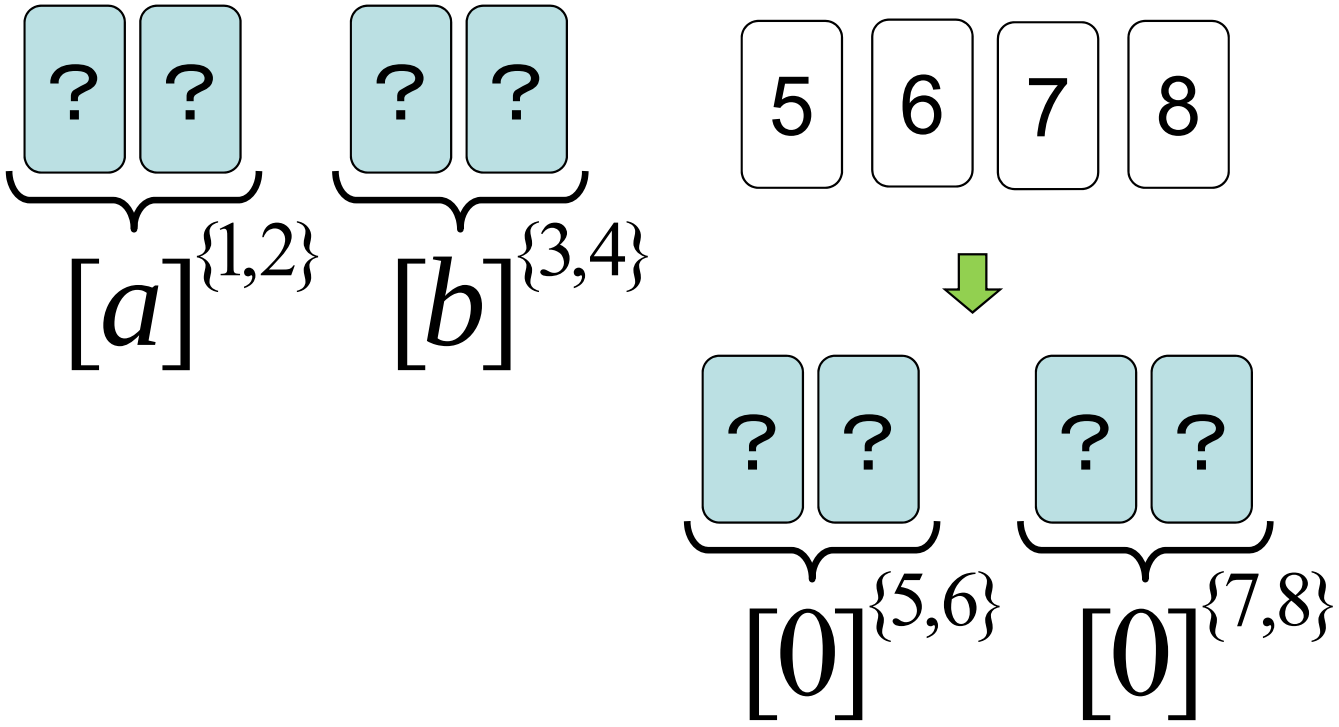
When the base of the commitment is known, the value of a leaks.

→ needs some modification

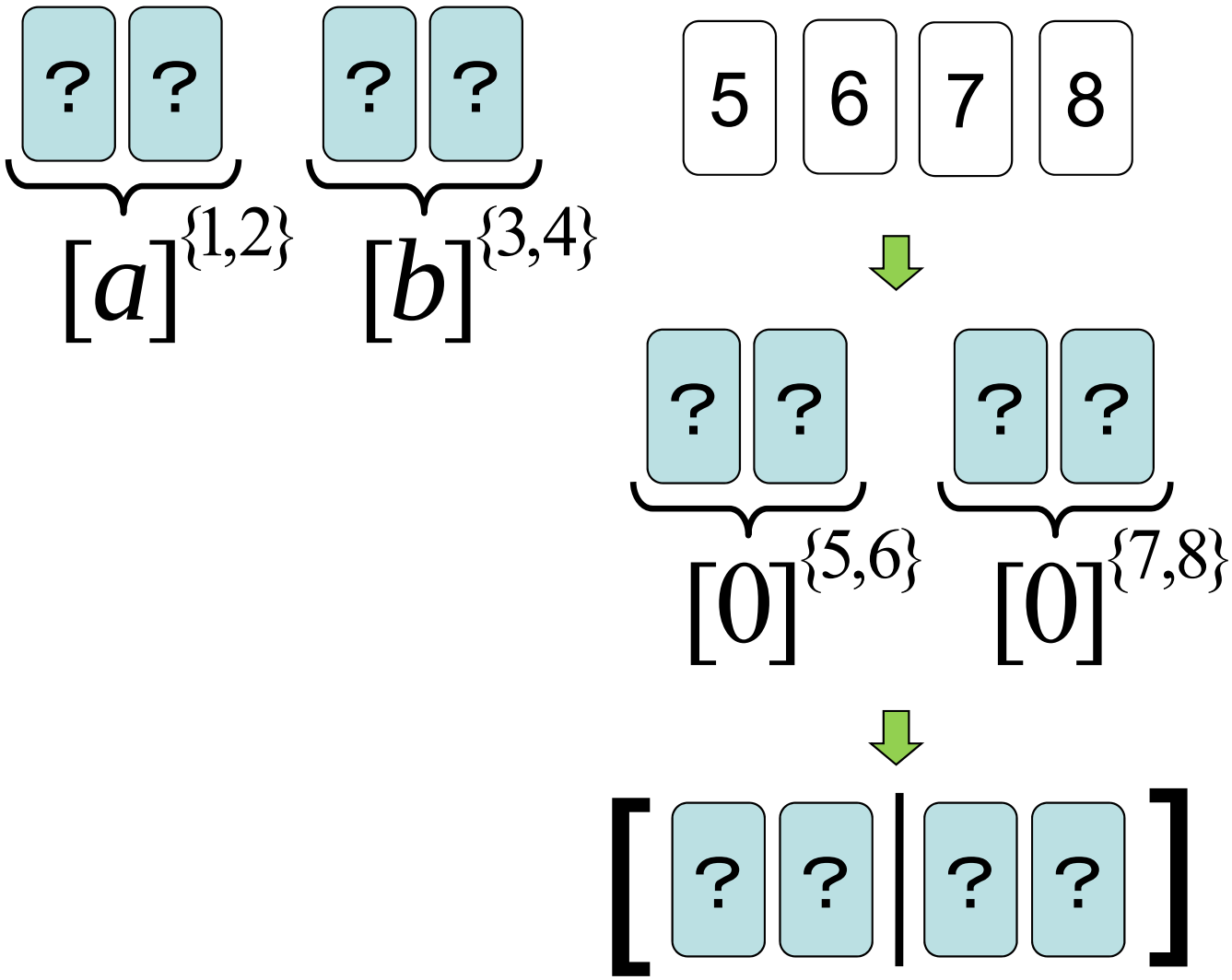
Use 8 cards:

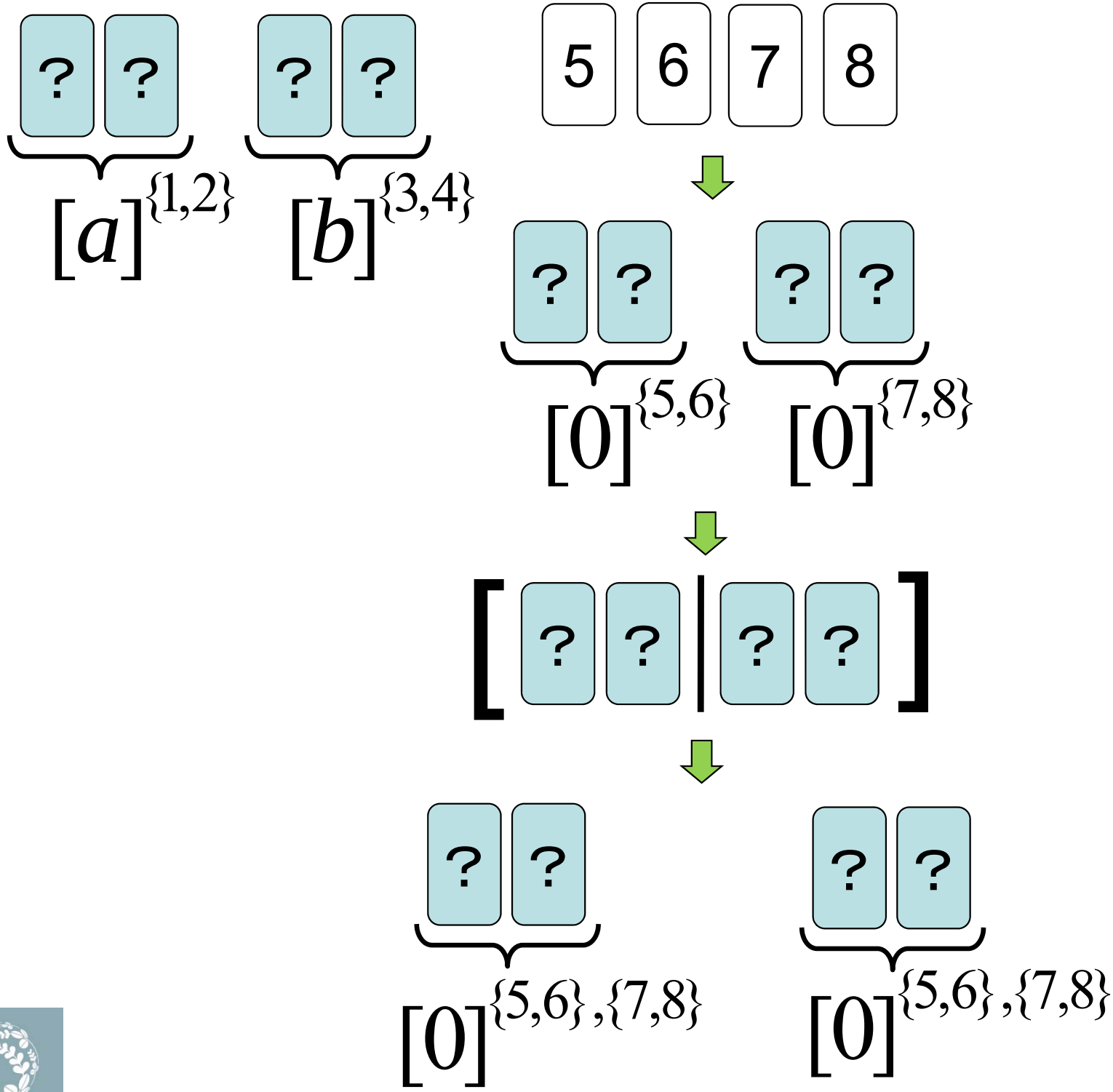


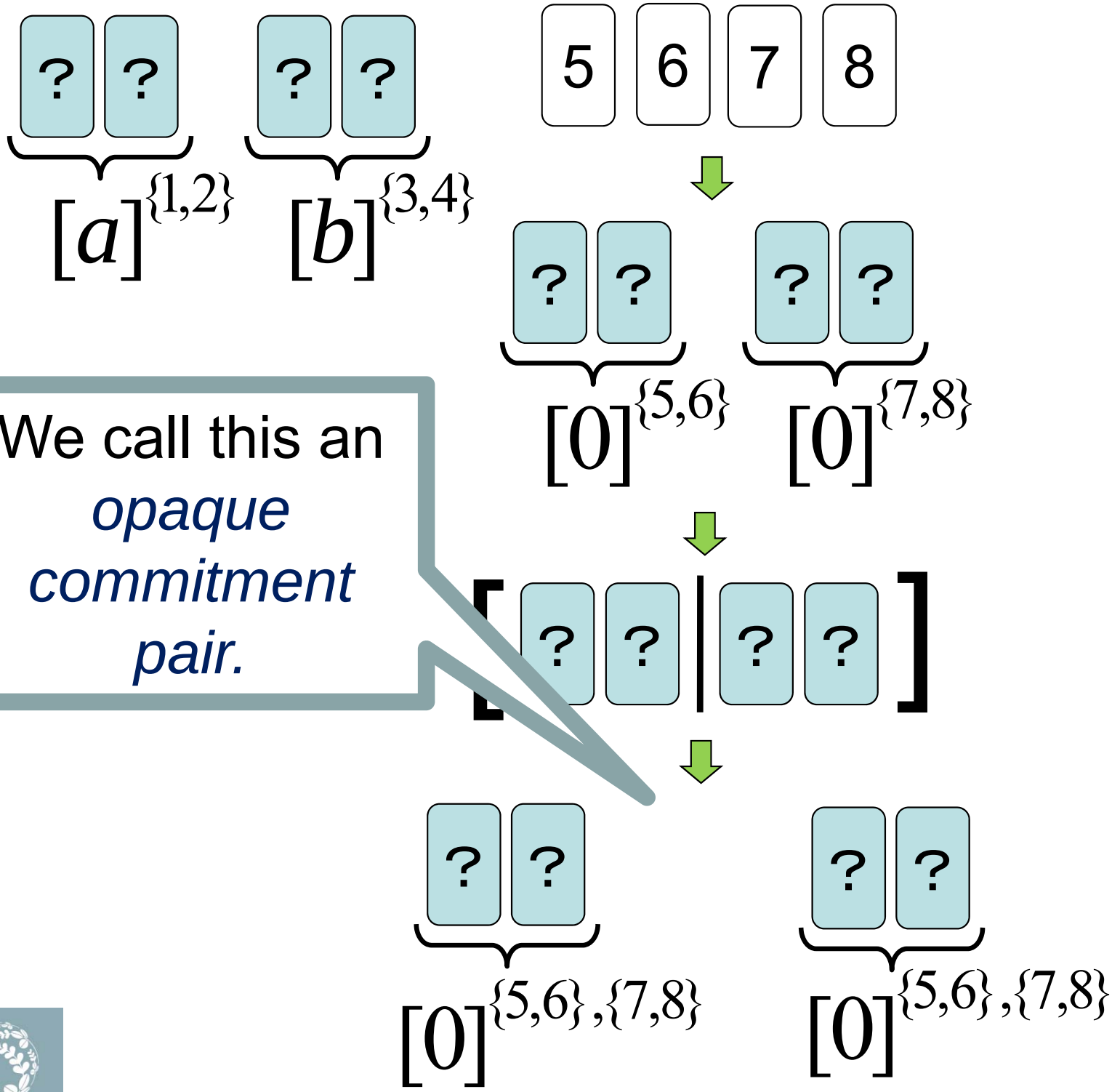
東北大学

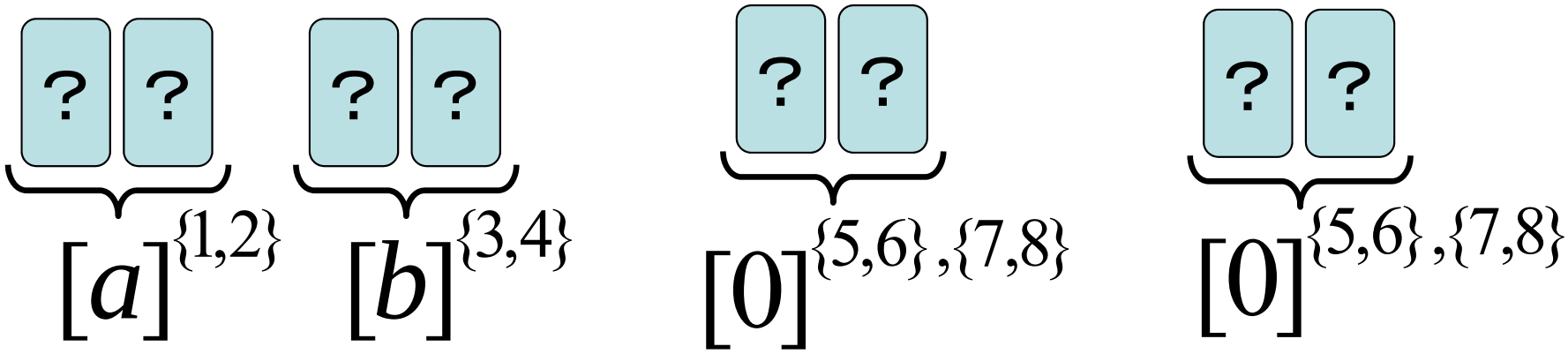


東北大学

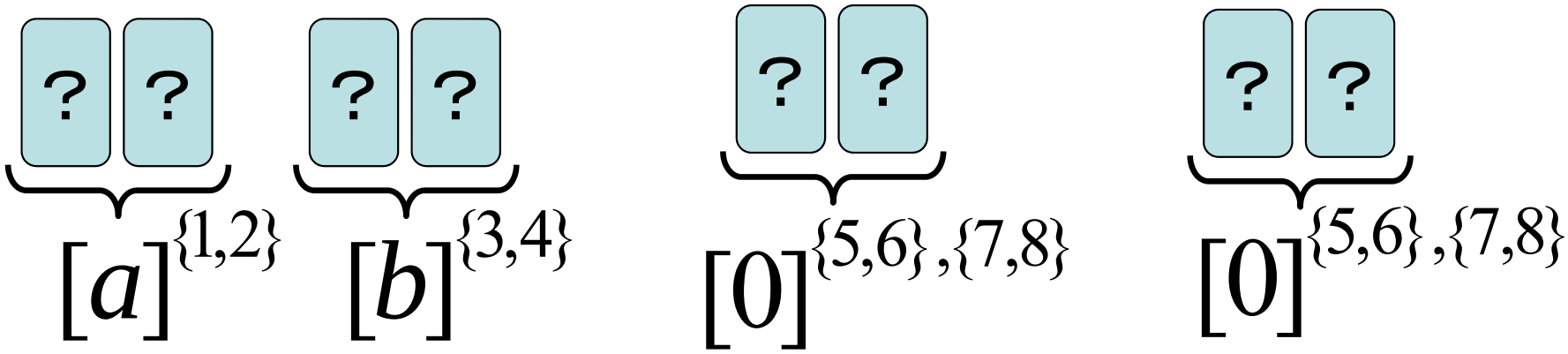

? ? ? ?
[a]{1,2} [b]{3,4}
5 6 7 8
? ? ? ?
[0]{5,6} [0]{7,8}
[? ? | ? ?]


? ?
? ?
[a]{1,2}
[b]{3,4}
5 6 7 8
? ?
? ?
[0]{5,6}
[0]{7,8}
[? ? | ? ?]
? ?
? ?
[0]{5,6},{7,8}
[0]{5,6},{7,8}


? ?
? ?
5 6 7 8
$[a]^{\{1,2\}}$
$[b]^{\{3,4\}}$
$[0]^{\{5,6\}}$
$[0]^{\{7,8\}}$
We call this an *opaque commitment pair.*
$[0]^{\{5,6\},\{7,8\}}$
$[0]^{\{5,6\},\{7,8\}}$

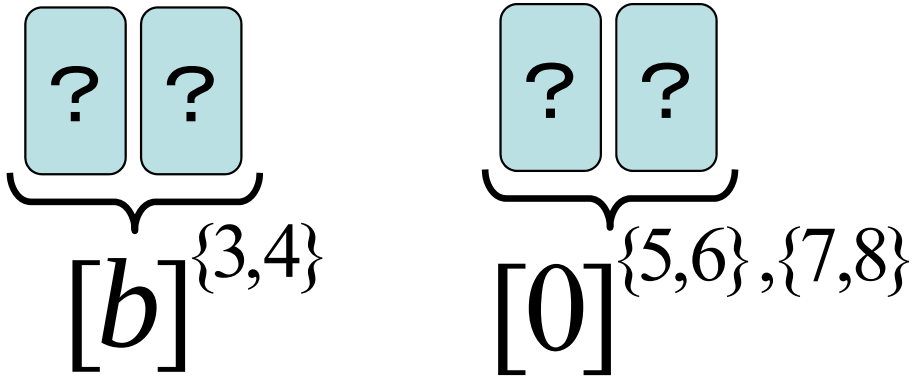


東北大学



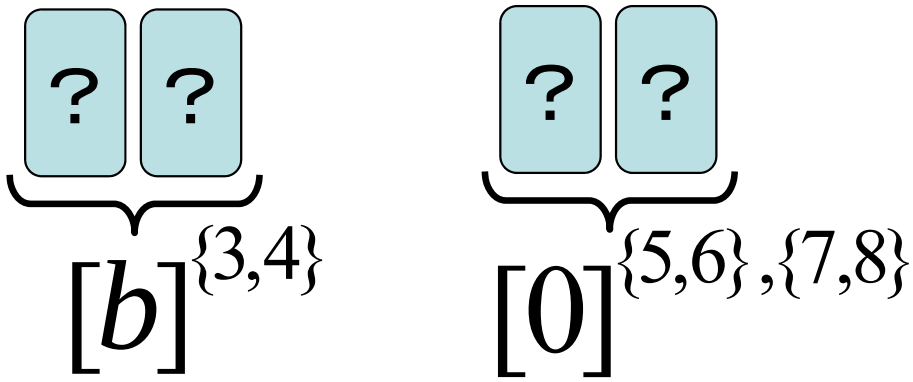
Make the base of the commitment to b opaque.

Make the base of the commitment to $\boldsymbol{b}$ opaque.



東北大学

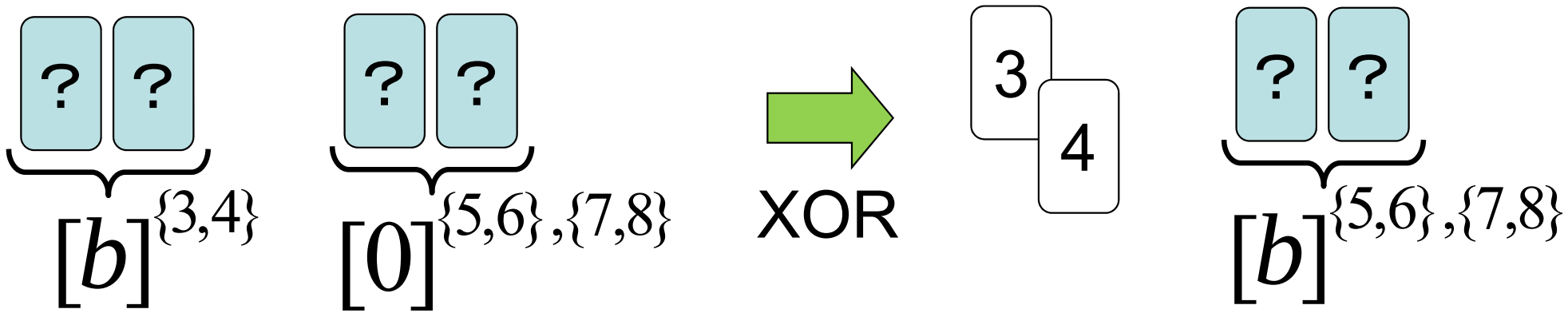
Make the base of the commitment to b opaque.

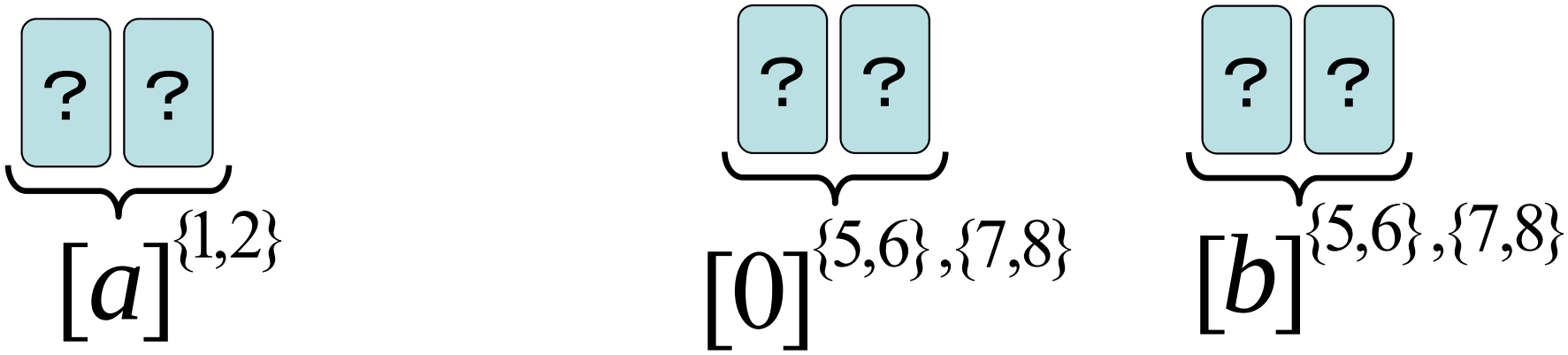


Apply the procedure of the XOR protocol

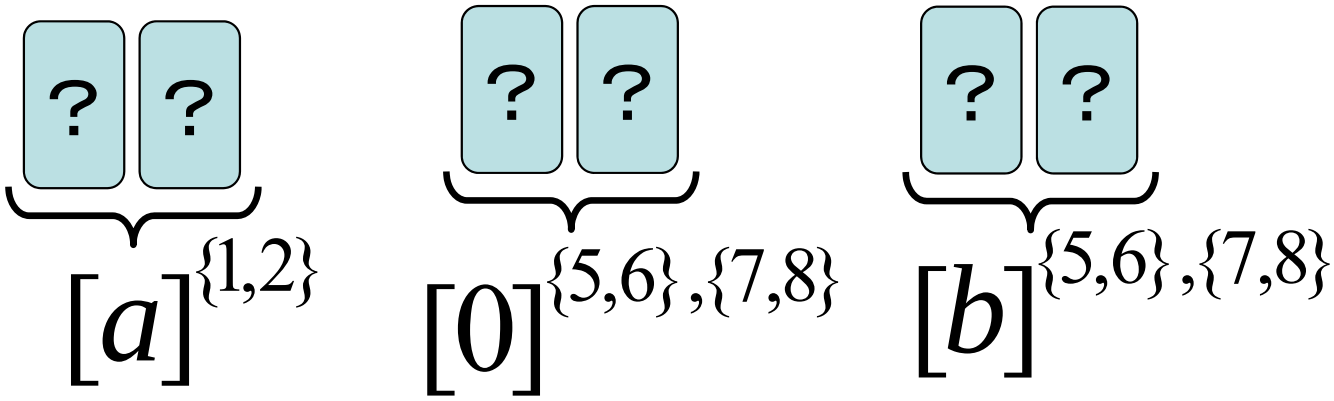
東北大学

Make the base of the commitment to b opaque.

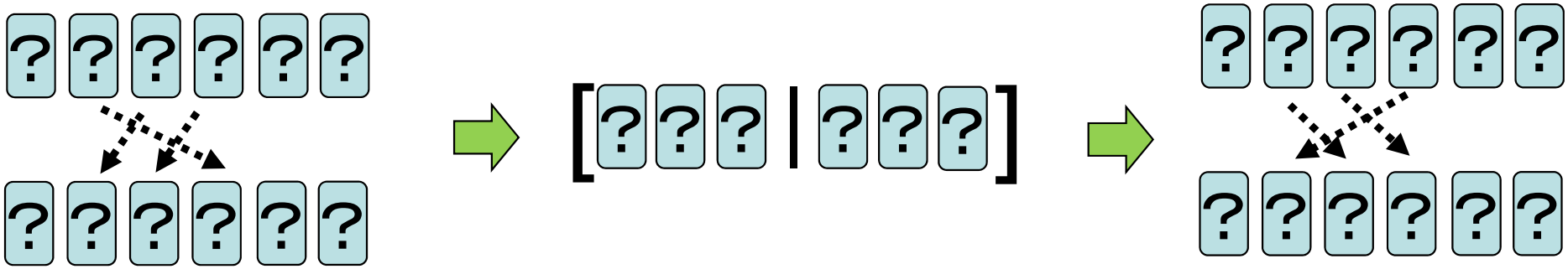




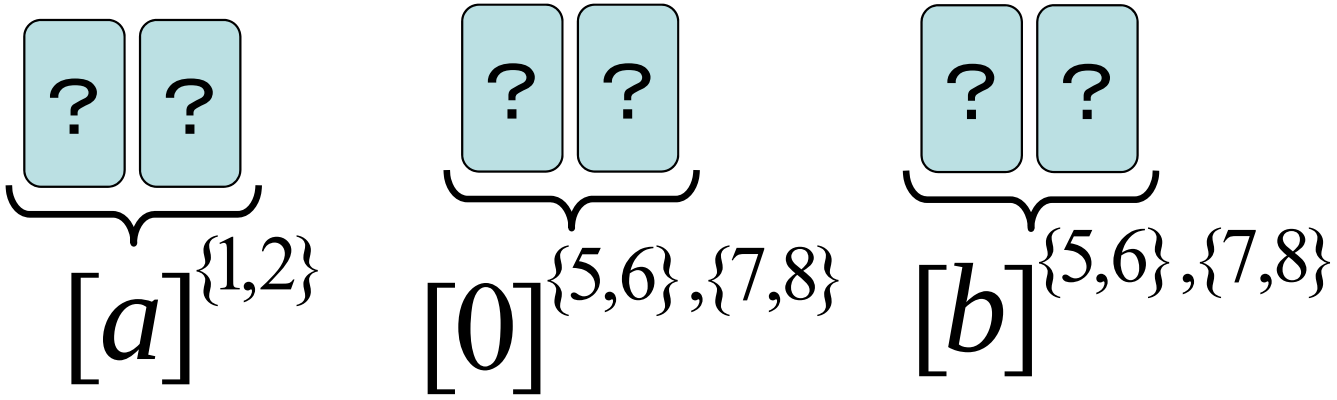
We got an opaque commitment pair to $\mathbf{0}$ and $\boldsymbol{b}$.

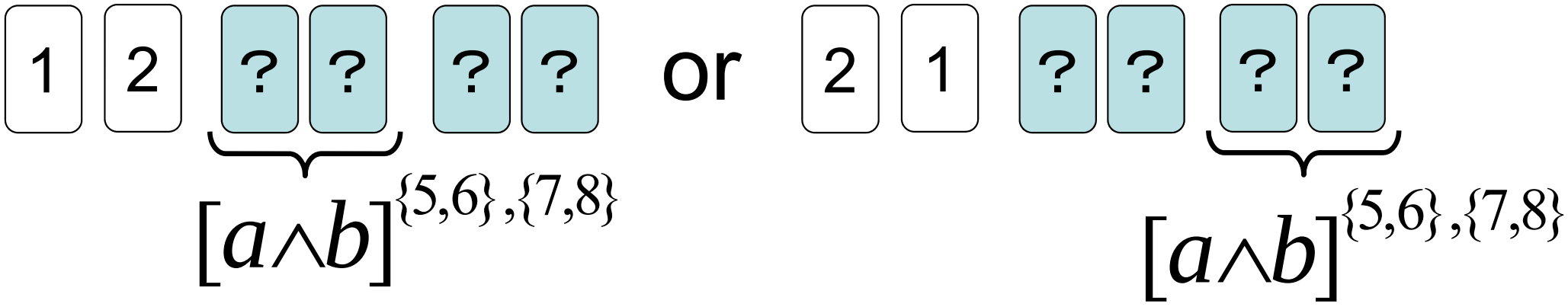


Apply the procedure of the existing AND protocol [11] (introduced before) :



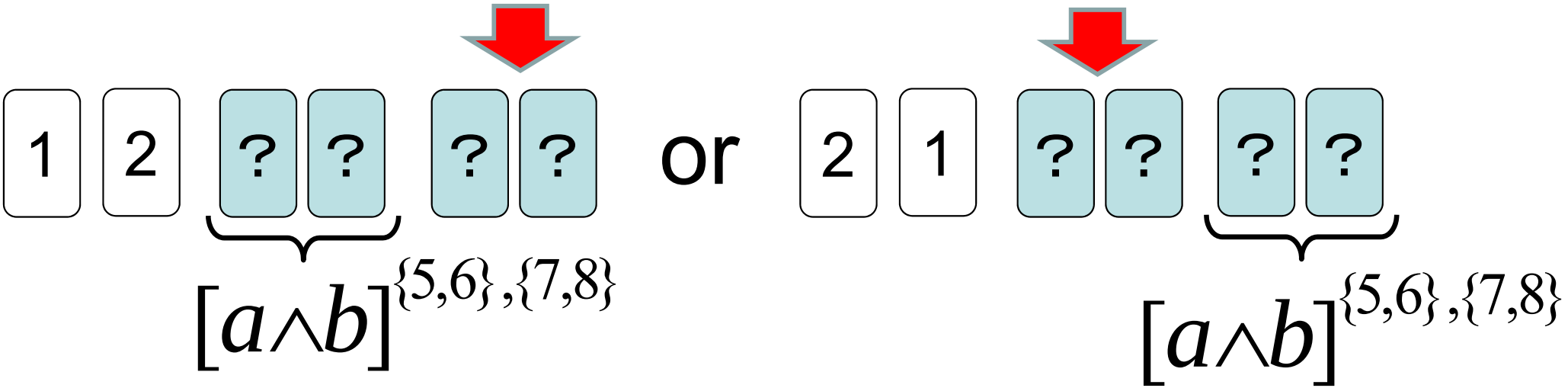
Then,



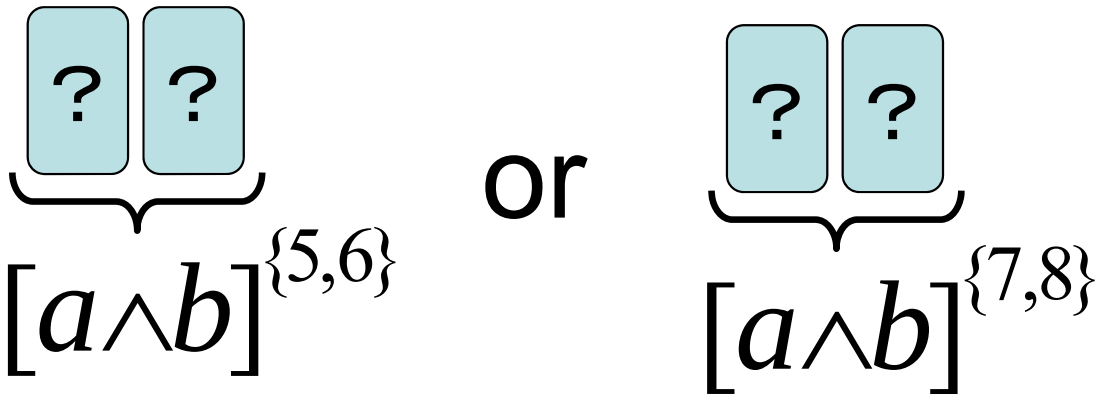


東北大学

Reveal the two cards after shuffling them.



The base of the commitment is found, and we have:



東北大学

How many shuffles（random bisection cuts）?

1. Producing an opaque commitment pair
2. Changing the base
3. Applying the existing AND protocol [11]
4. Finding the base

→ 4 shuffles in total

東北大学

AND computation

	# of cards	# of shuffles	
		avg.	fixed
Niemi-Renvall [13]（Sect. 2）	5	9.5	0
Ours（Sect. 3）	8	0	4

XOR computation

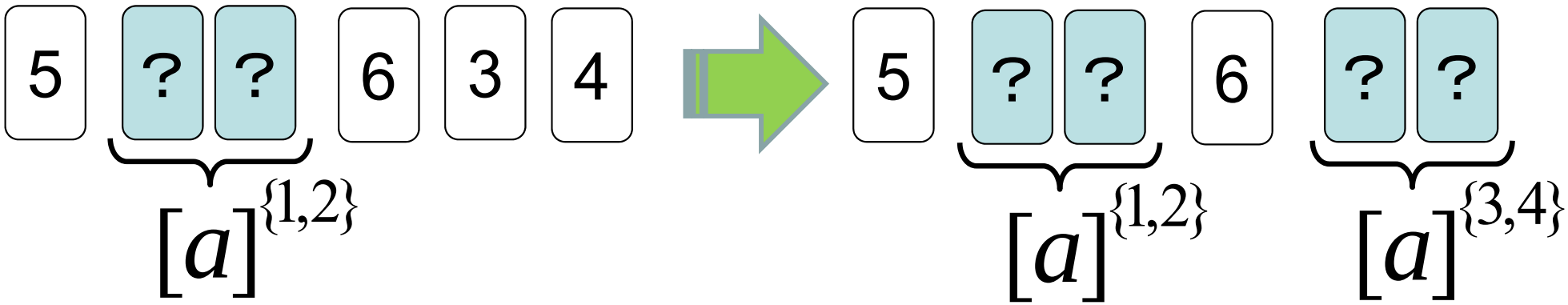
Niemi-Renvall [13]（Sect. 2）	4	7	0
Ours（Sect. 4）	4	0	1

東北大学

Contents

東北大学

Niemi-Renvall copy protocol [13]



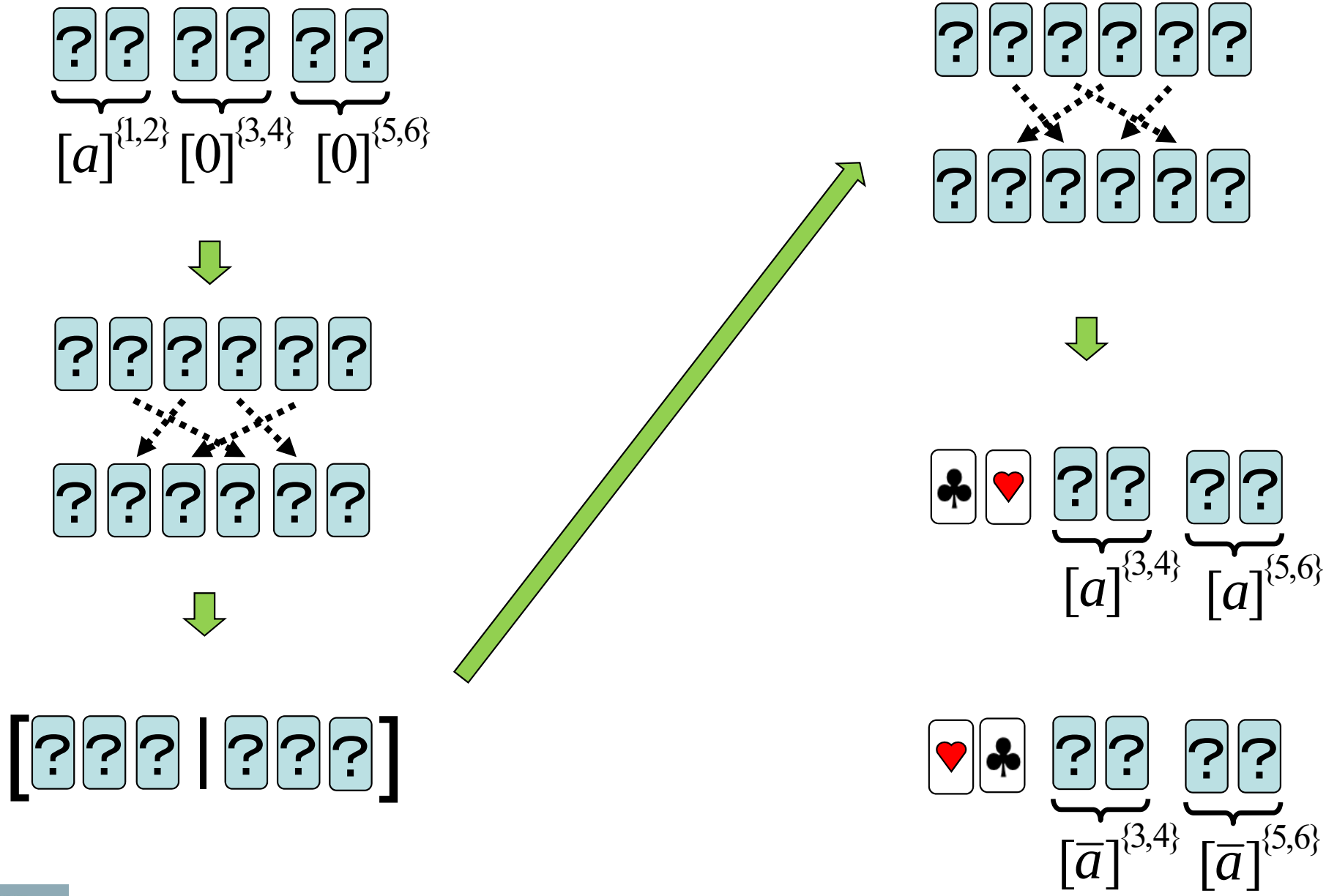
A random cut（= cyclic shuffle）is applied an average of **4.5** times + fixed **1** time.

We omit the details.

# of cards	# of shuffles		
	avg.	fixed	total

Secure copy

Niemi-Renvall [13]（Sect. 2）	6	4.5	1	5.5


[a]{1,2} [0]{3,4} [0]{5,6}
[a]{3,4} [a]{5,6}
[ā]{3,4} [ā]{5,6}

# of cards	# of shuffles		
	avg.	fixed	total

Secure copy

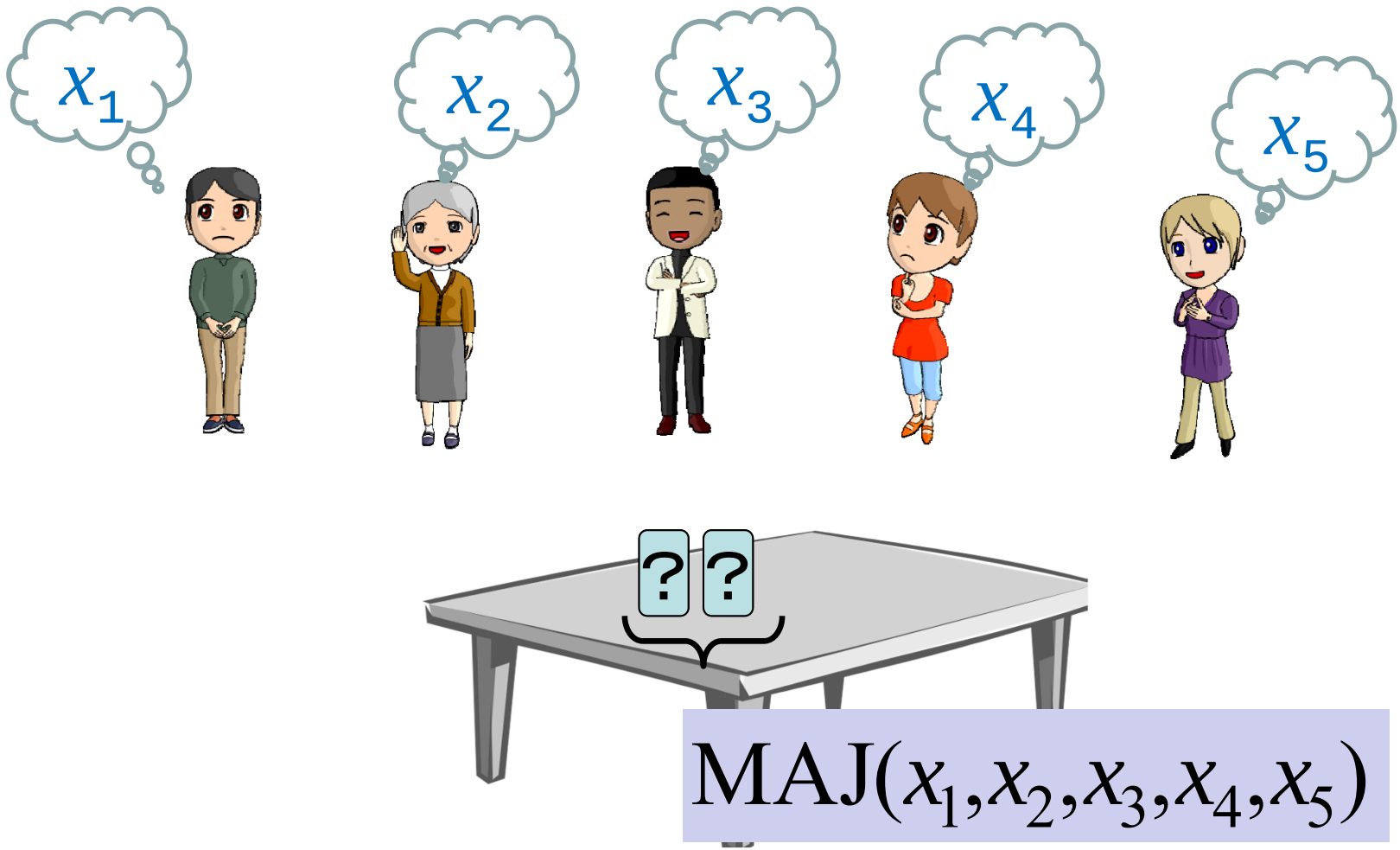
Niemi-Renvall [13]（Sect. 2）	6	4.5	1	5.5
Ours（Sect. 5）	6	0	1	1

東北大学

AND computation	# of cards	# of shuffles		
		avg.	fixed	total
Niemi-Renvall [13]（Sect. 2）	5	9.5	0	9.5
Ours（Sect. 3）	8	0	4	4
XOR computation				
Niemi-Renvall [13]（Sect. 2）	4	7	0	7
Ours（Sect. 4）	4	0	1	1
Secure copy				
Niemi-Renvall [13]（Sect. 2）	6	4.5	1	5.5
Ours（Sect. 5）	6	0	1	1

東北大学

Combining AND/XOR/NOT/copy protocols, one can construct a protocol for any function.



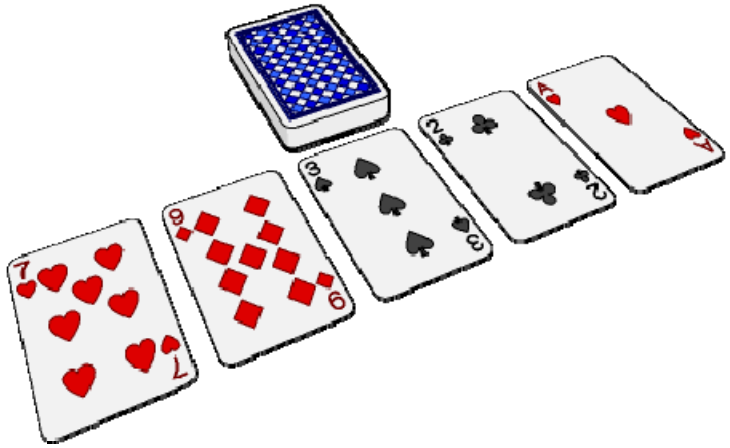
東北大学

Contents

東北大学

Conclusion

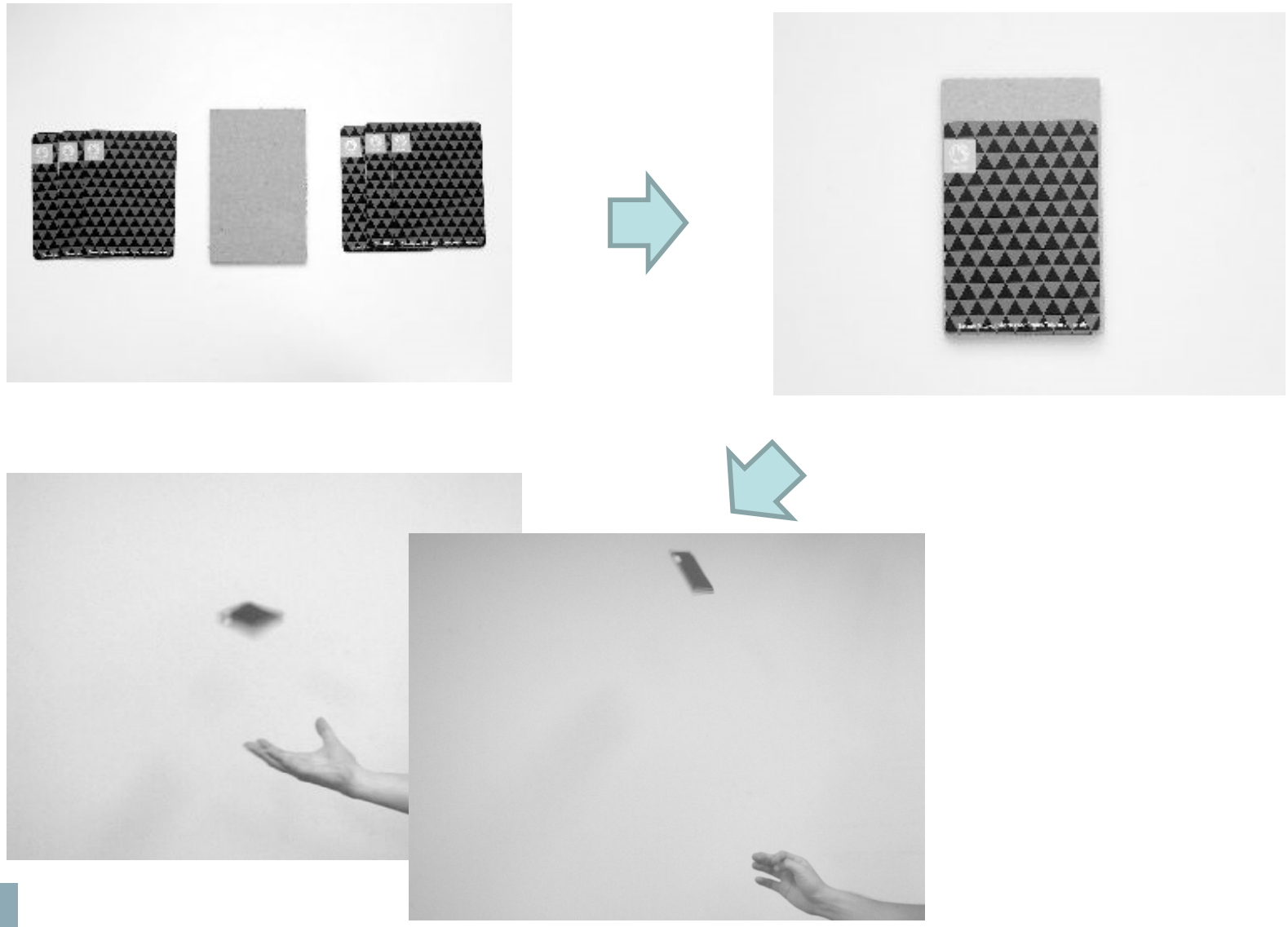
- MPC can be done with standard playing cards.
- We reduced the number of required shuffles.



東北大学

How to implement a random bisection cut?

Spinning throw with a separator and a rubber band:



[Ueda, et al.,TPNC 2016]