12th Annual Conference on Theory and Applications of Models of Computation
18–20 May 2015, National University, Singapore

東北大学

Card-Based Protocols for Any Boolean Function

Takuya Nishida, Yu-ichi Hayashi,
Takaaki Mizuki, Hideaki Sone

Tohoku University

[Monday 18 May 2015 14:50-15:15]

12th Annual Conference on Theory and Applications of Models of Computation
18–20 May 2015, National University, Singapore

東北大学

Card-Based Protocols for Any Boolean Function

What is Card-Based Protocol?

[Monday 18 May 2015 14:50-15:15]

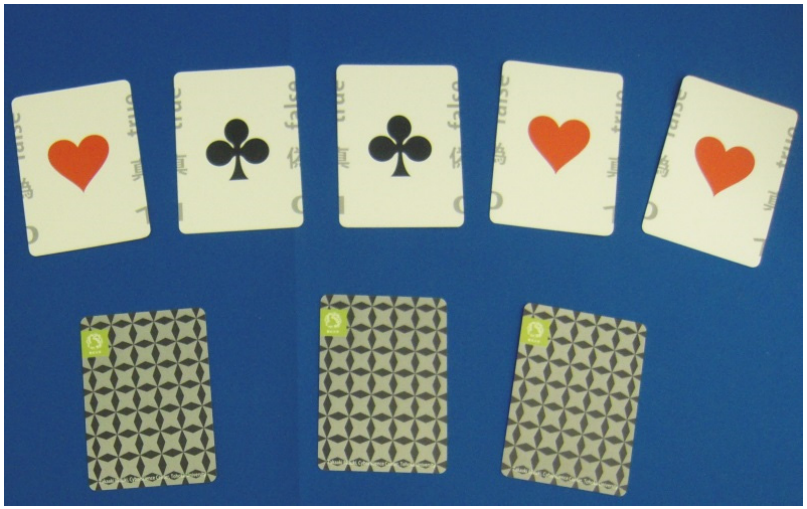
What is Card-Based Protocol?

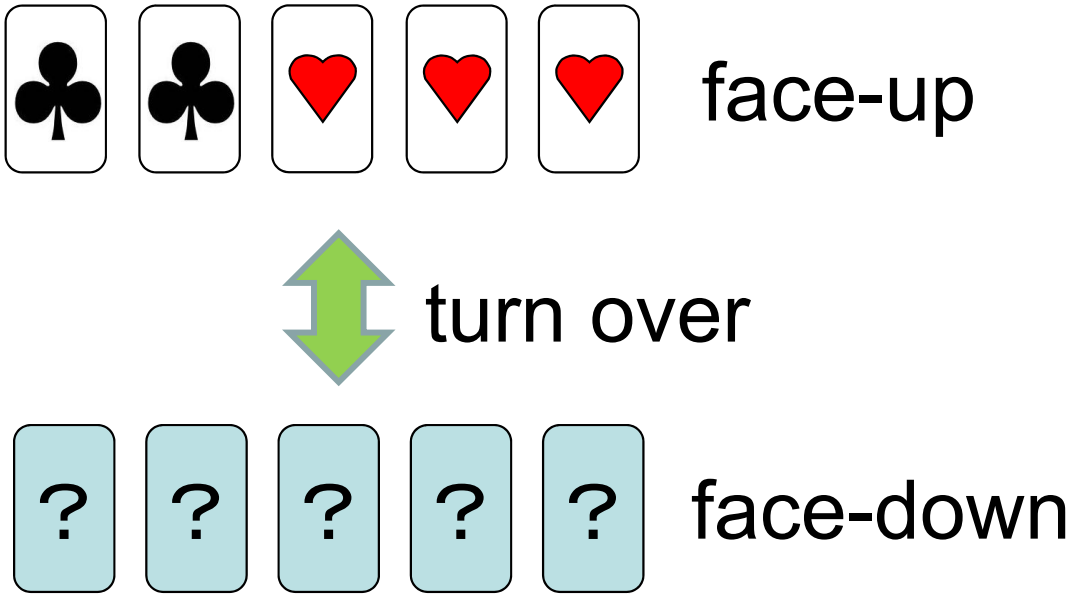
東北大学

Card-based protocols achieve

secure multi-party computation (MPC)

using a number of physical cards of 2 colors.





What is Card-Based Protocol?

東北大学

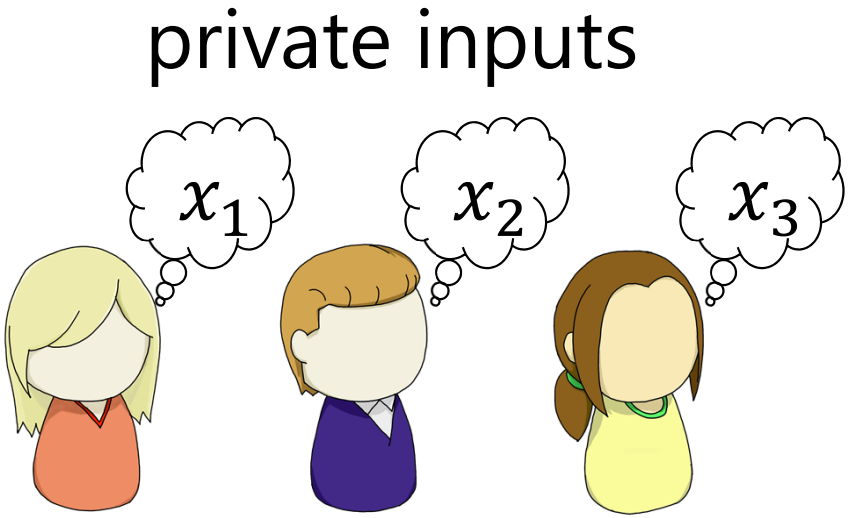
Card-based protocols achieve

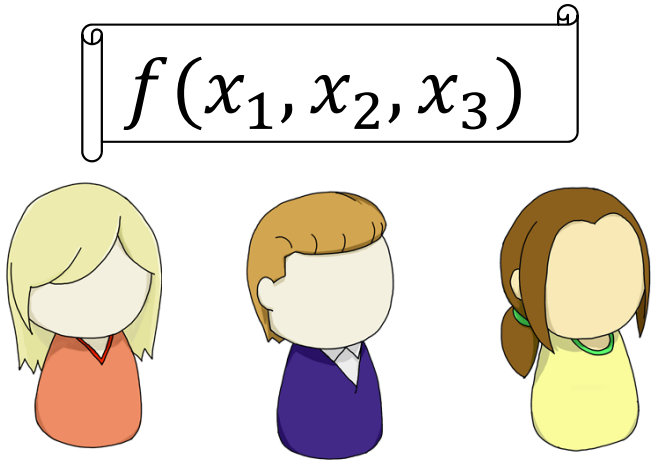
***secure multi-party computation* (MPC)**

using a numb of physical cards of 2 colors.

What is secure multi-party computation?

What is secure multi-party computation?





With keeping their inputs secret, all players know only the value of a function (e.g., AND, MAJ).

Title:

Card-Based Protocols for Any Boolean Function

We propose a general approach to constructing a card-based protocol for any given function.

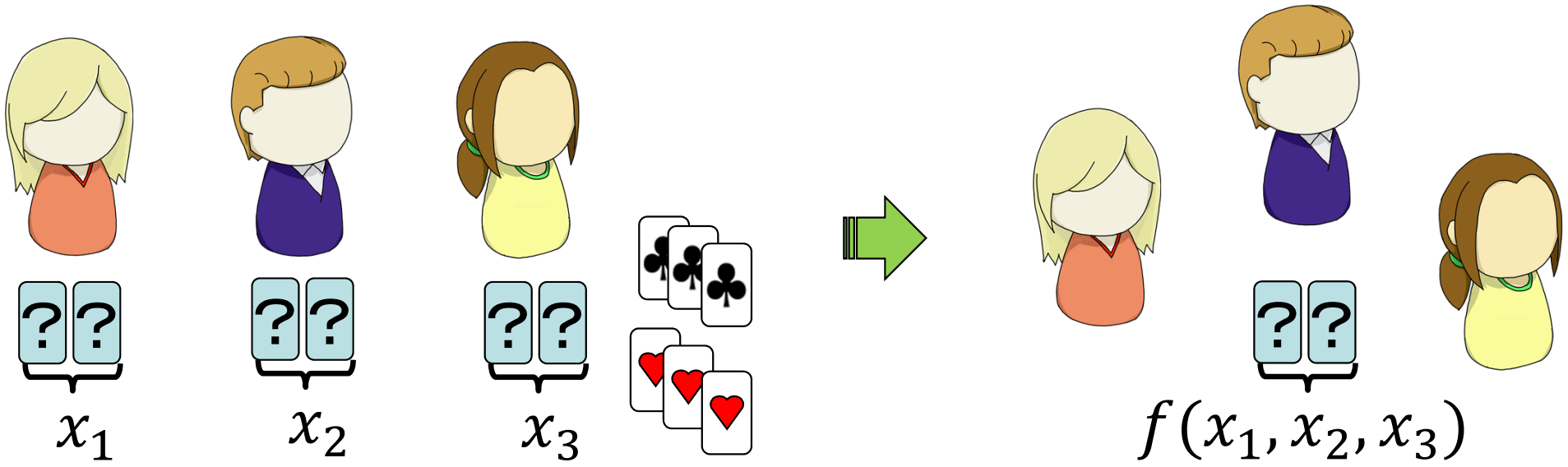
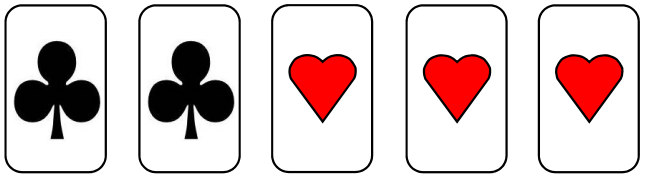


Table of Contents

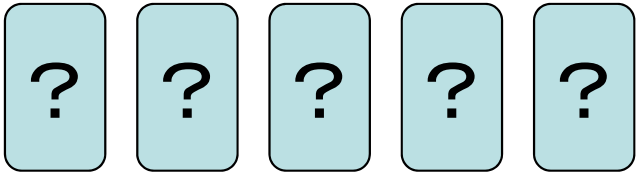
Table of Contents

Preliminary Notations



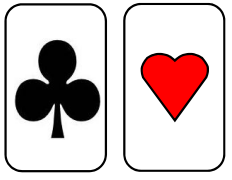
face-up

turn over

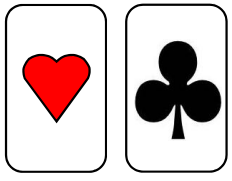


face-down

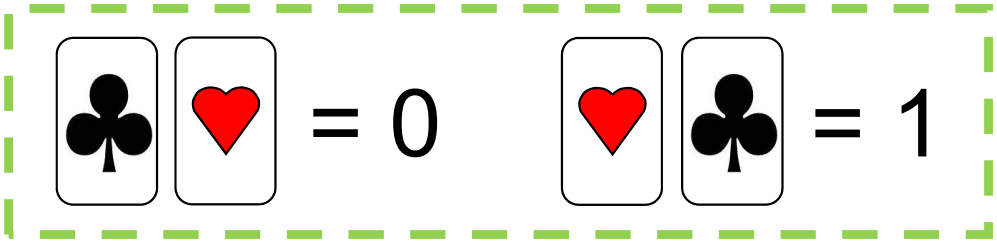
To deal with Boolean values, this encoding is used:



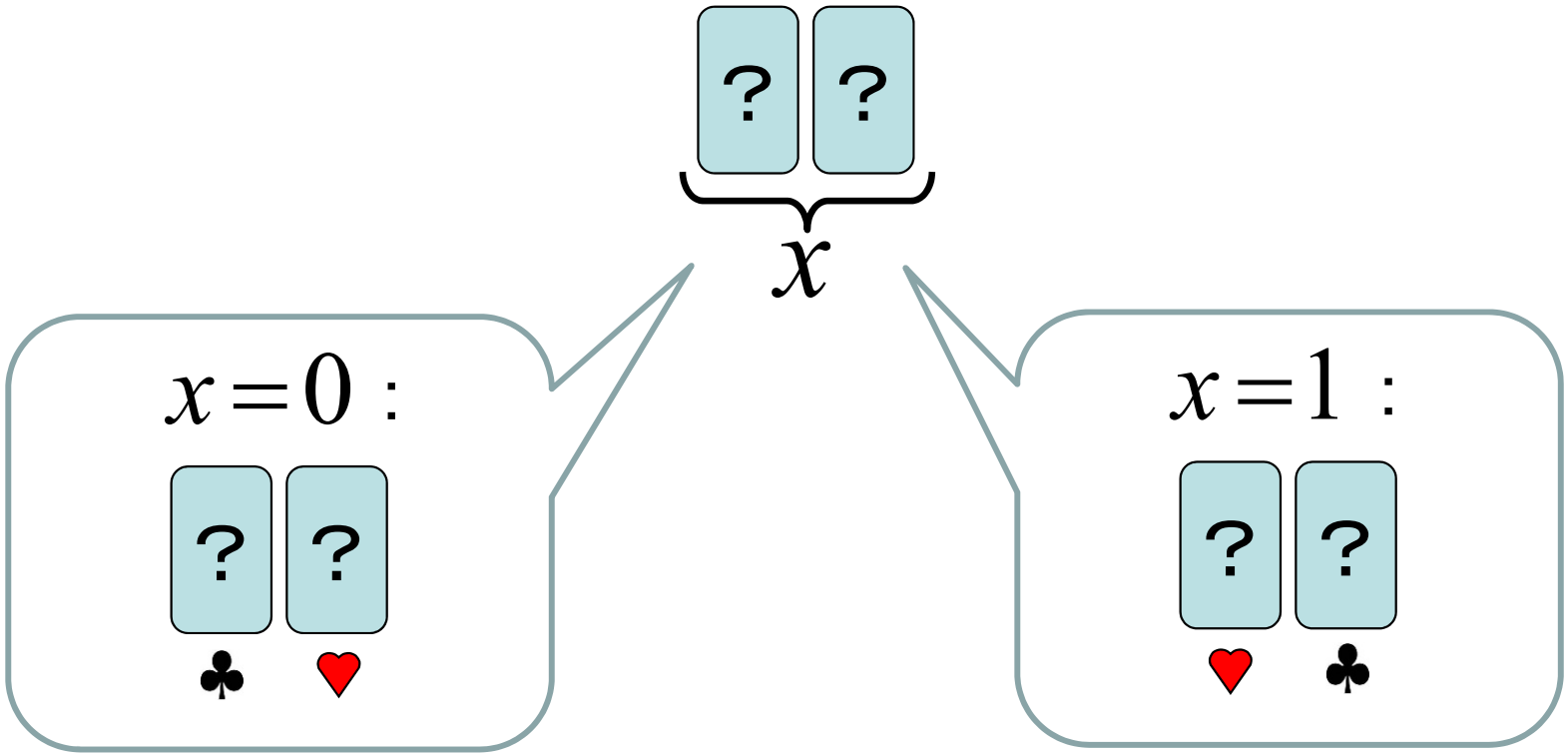
= 0

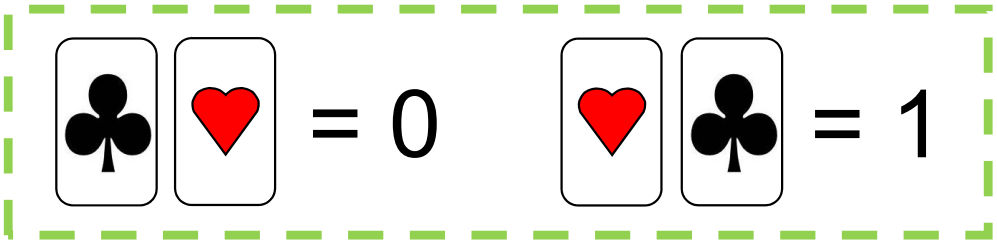


= 1

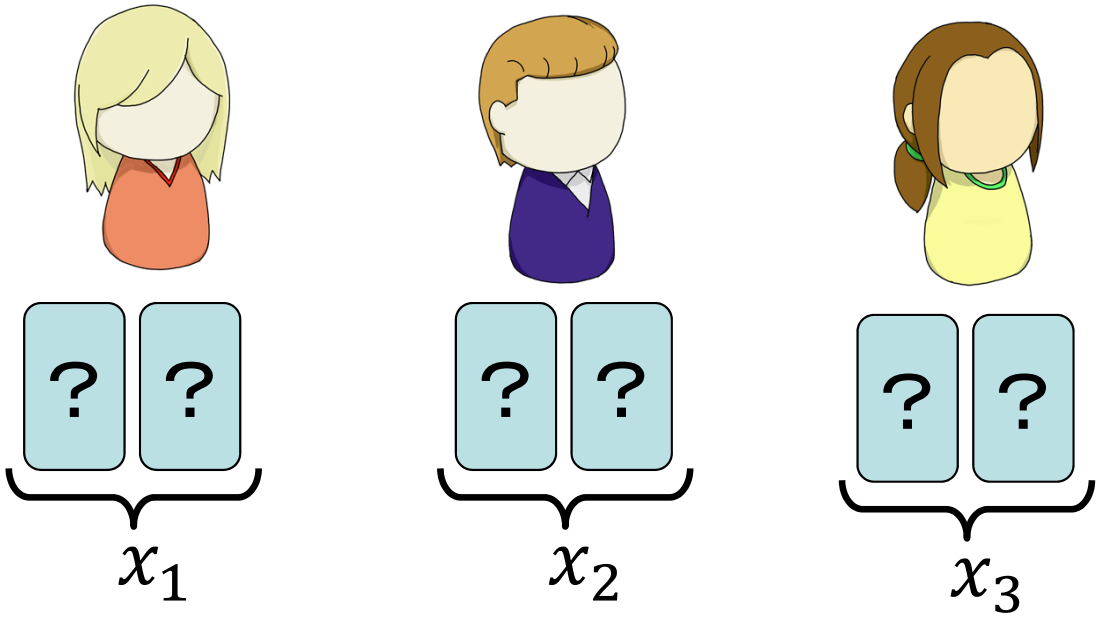


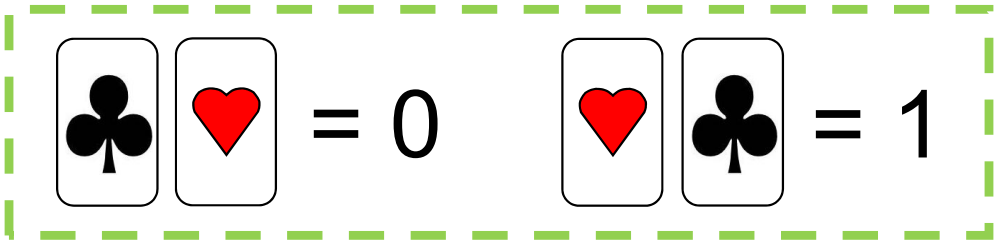
A ***commitment*** to a bit is a pair of face-down cards following the encoding.



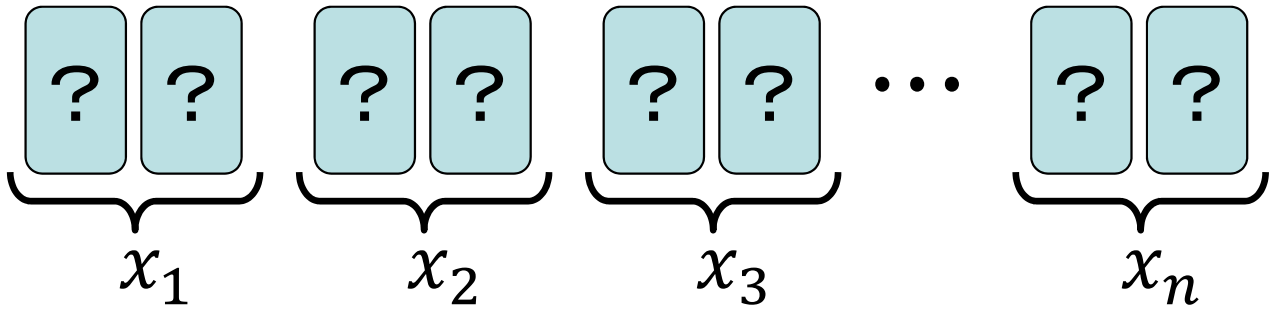


Using 2 cards, each player can arrange a commitment to his/her private input bit.

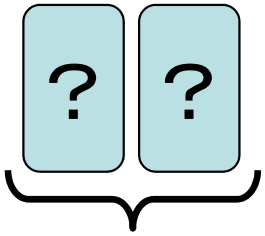




Given commitments to private input bits



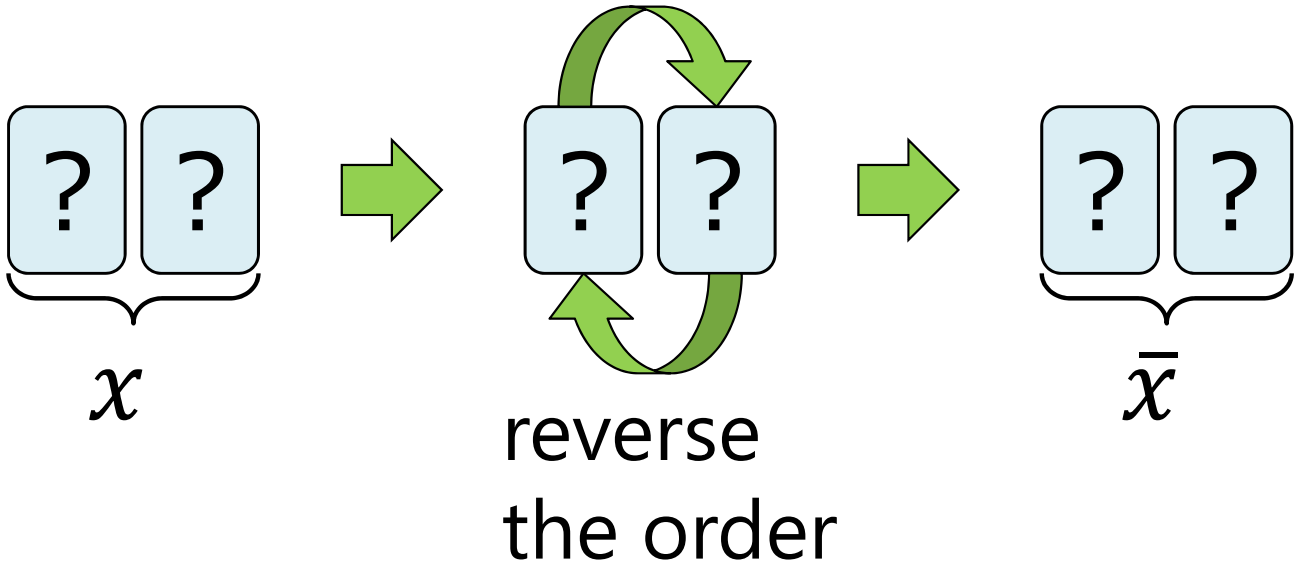
along with additional cards, a card-based protocol produces a commitment to the value of a function

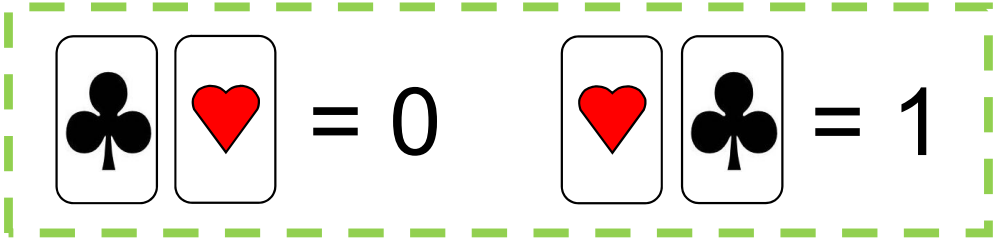


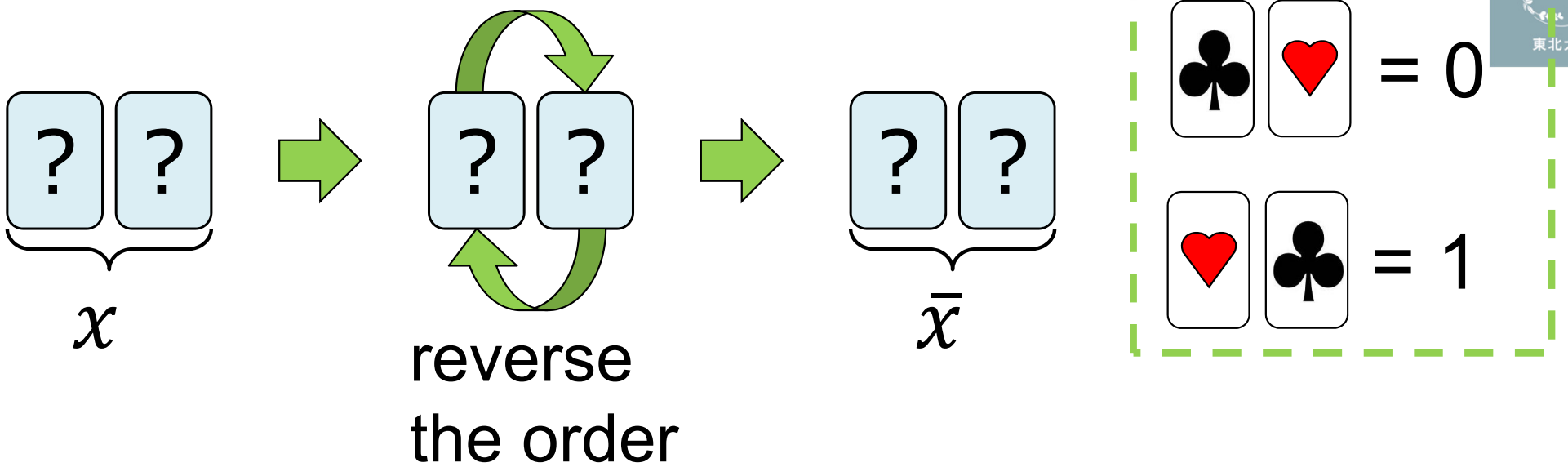
$f(x_1, x_2, \ldots, x_n)$

The simplest protocol: NOT computation

Swapping two cards results in a commitment to the negation of the bit.

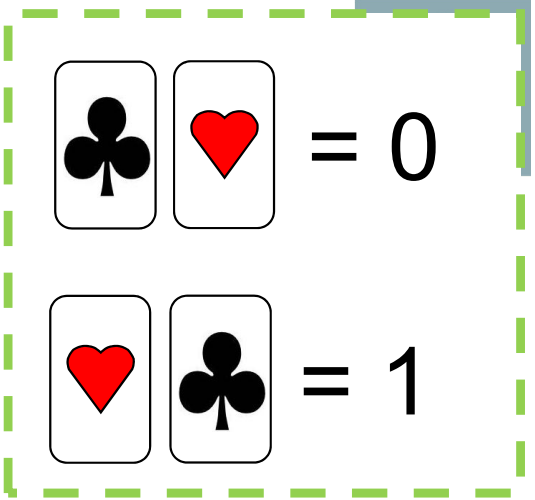




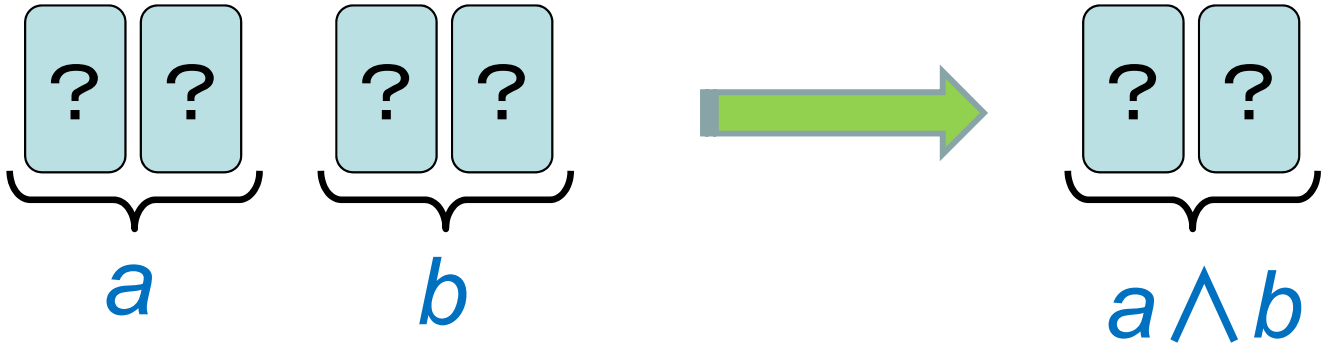


With keeping the value of x secret, we can get a commitment to the negation $\bar{x}$ of x .

- Secure **NOT** computation is trivial.



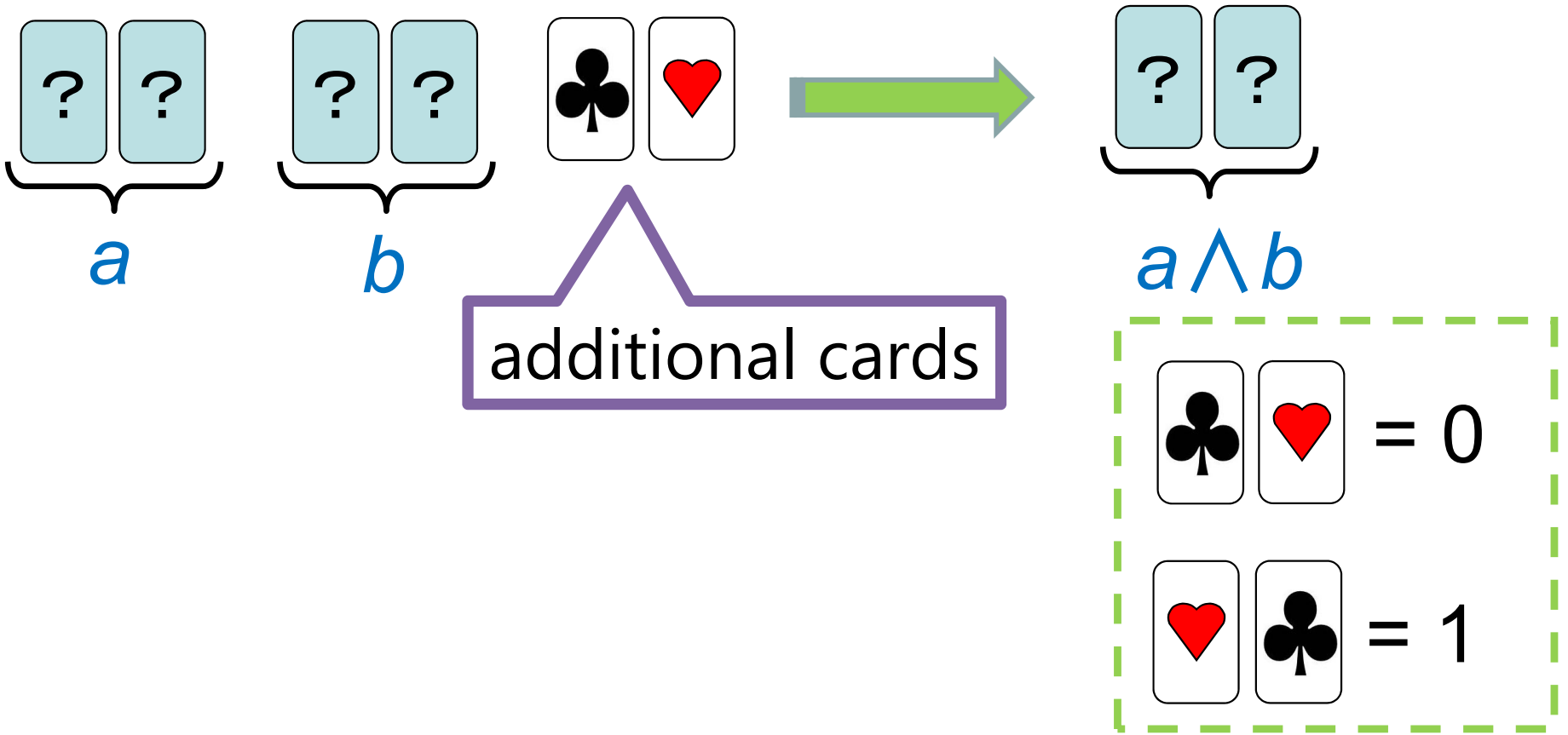
➢ How about secure **AND** computation?



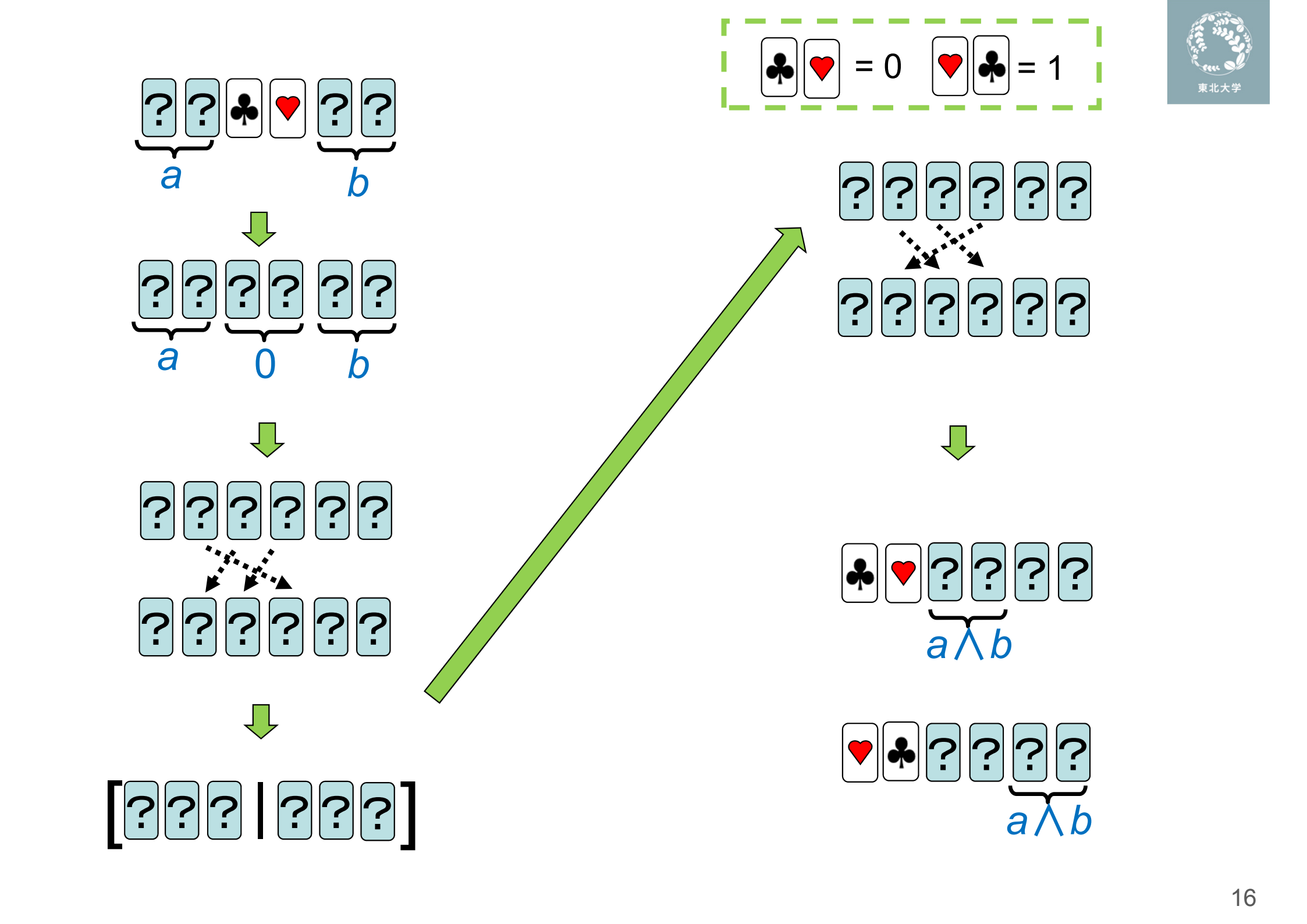
With keeping the values of a and b secret, we want to get a commitment to $a \wedge b$.

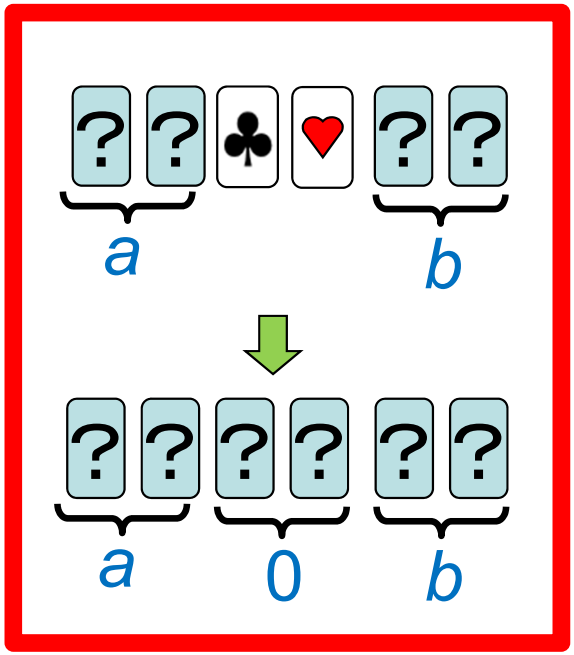
The most efficient existing AND protocol

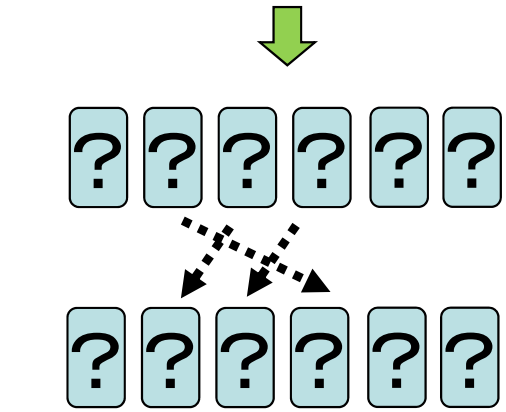
A commitment to $a \wedge b$ can be obtained using 2 additional cards [10].

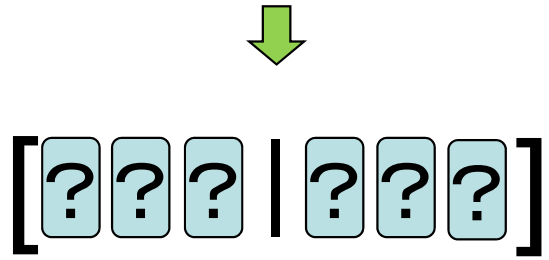


[10] T. Mizuki and H. Sone, Six-Card Secure AND and Four-Card Secure XOR, FAW 2009, LNCS 5598, pp. 358–369, 2009.

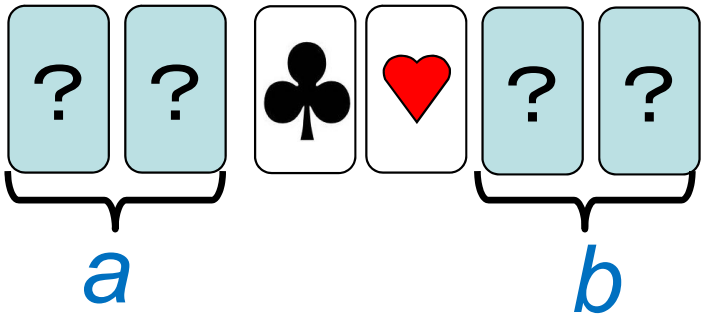

= 0
= 1
a
b
a
0
b
a ∧ b
a ∧ b



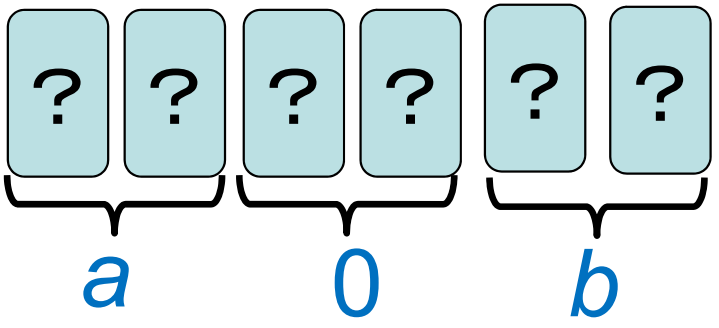




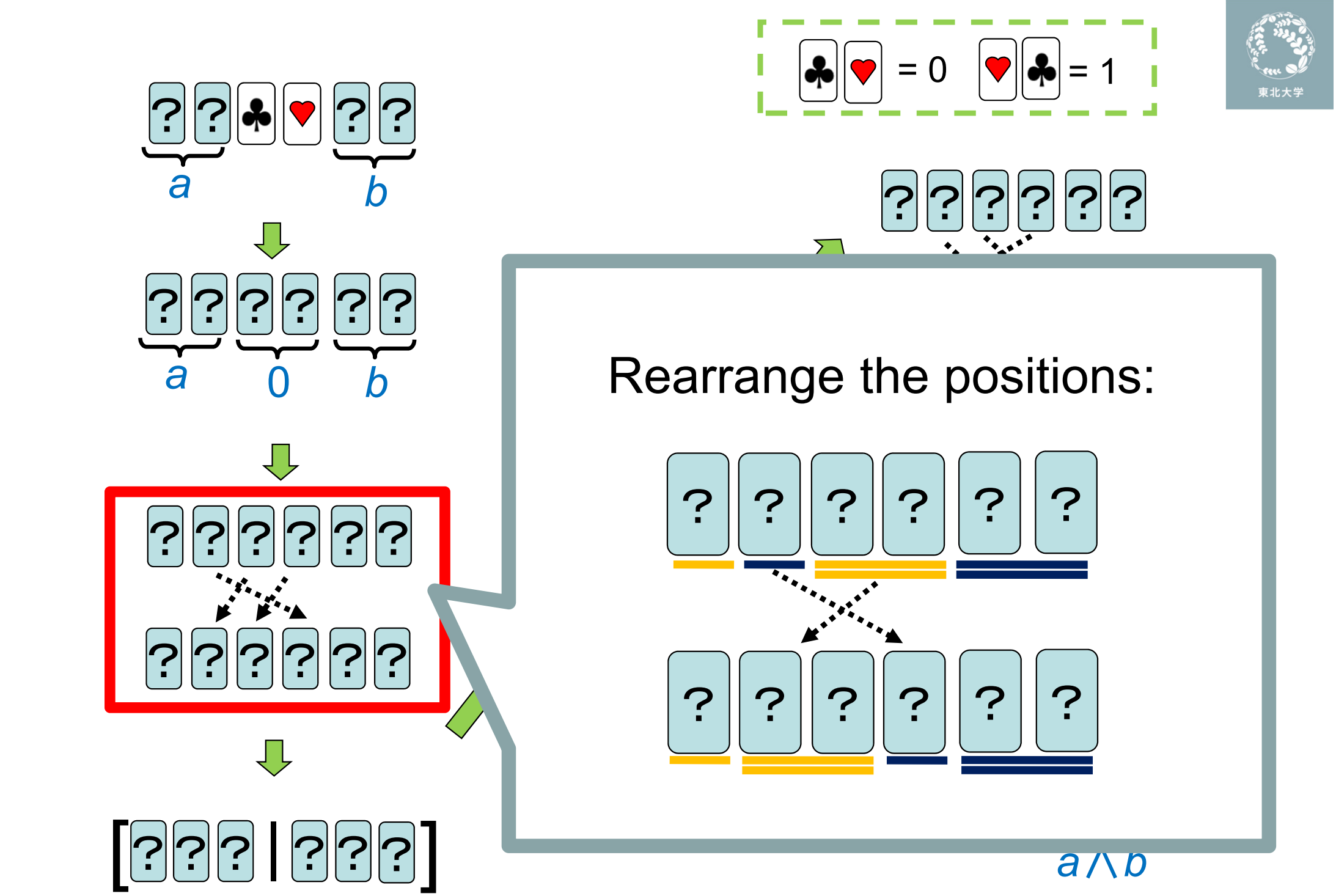
= 0 = 1

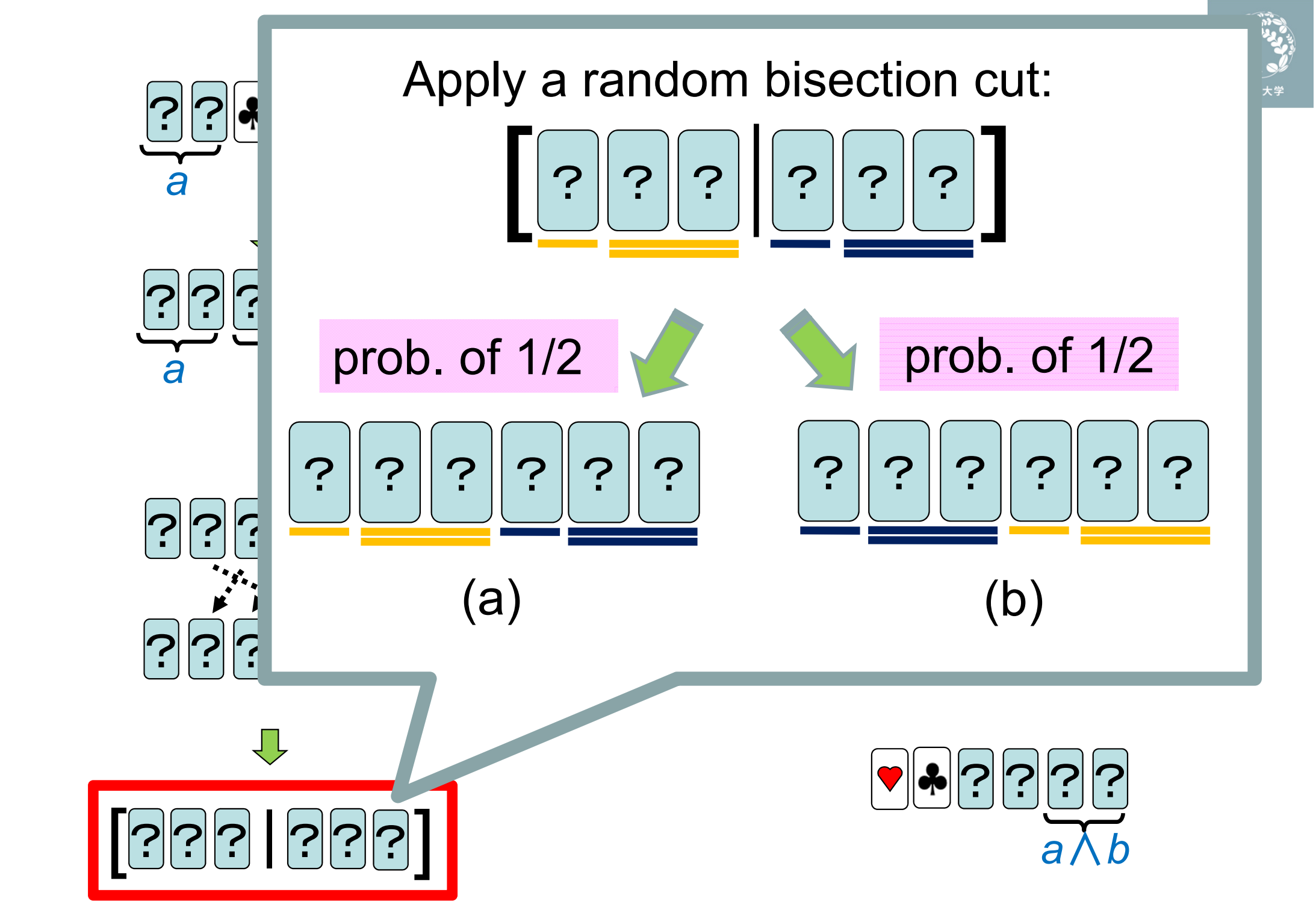


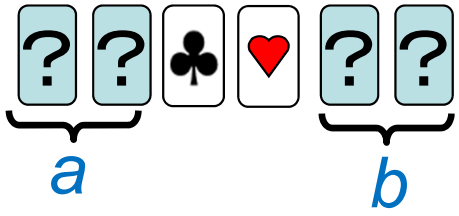
Turn over the two middle cards:

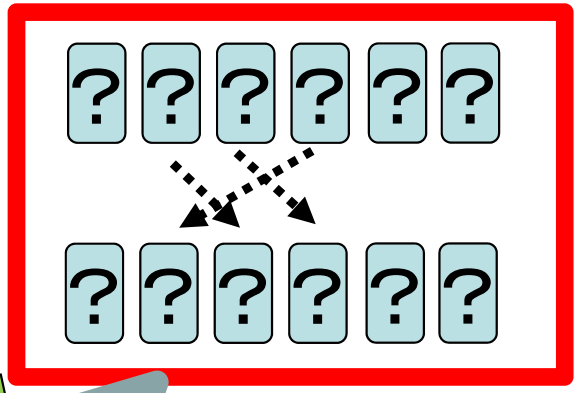


$a \wedge b$


= 0
= 1
a
b
a
0
b
Rearrange the positions:
a /\ b

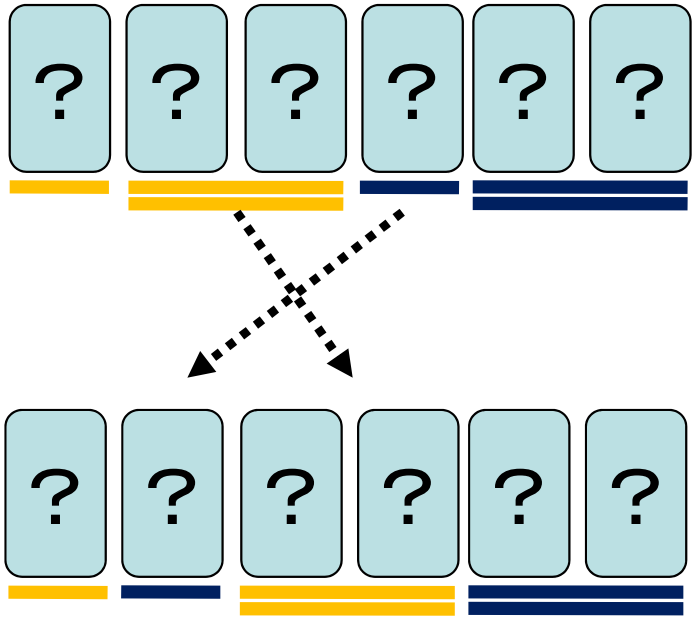

Apply a random bisection cut:
prob. of 1/2
prob. of 1/2
(a)
(b)
a
a
a ∧ b



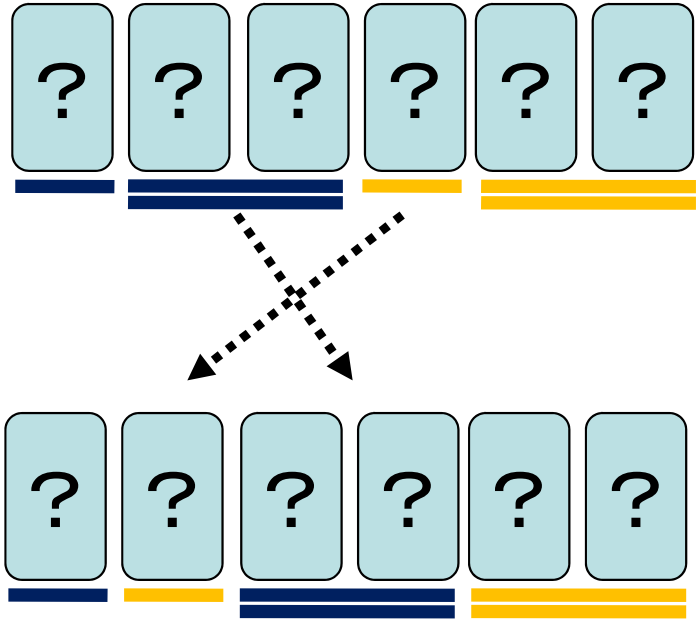


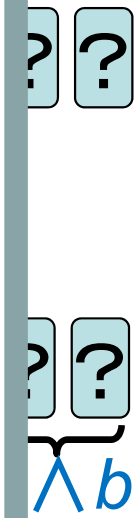
Rearrange the positions:

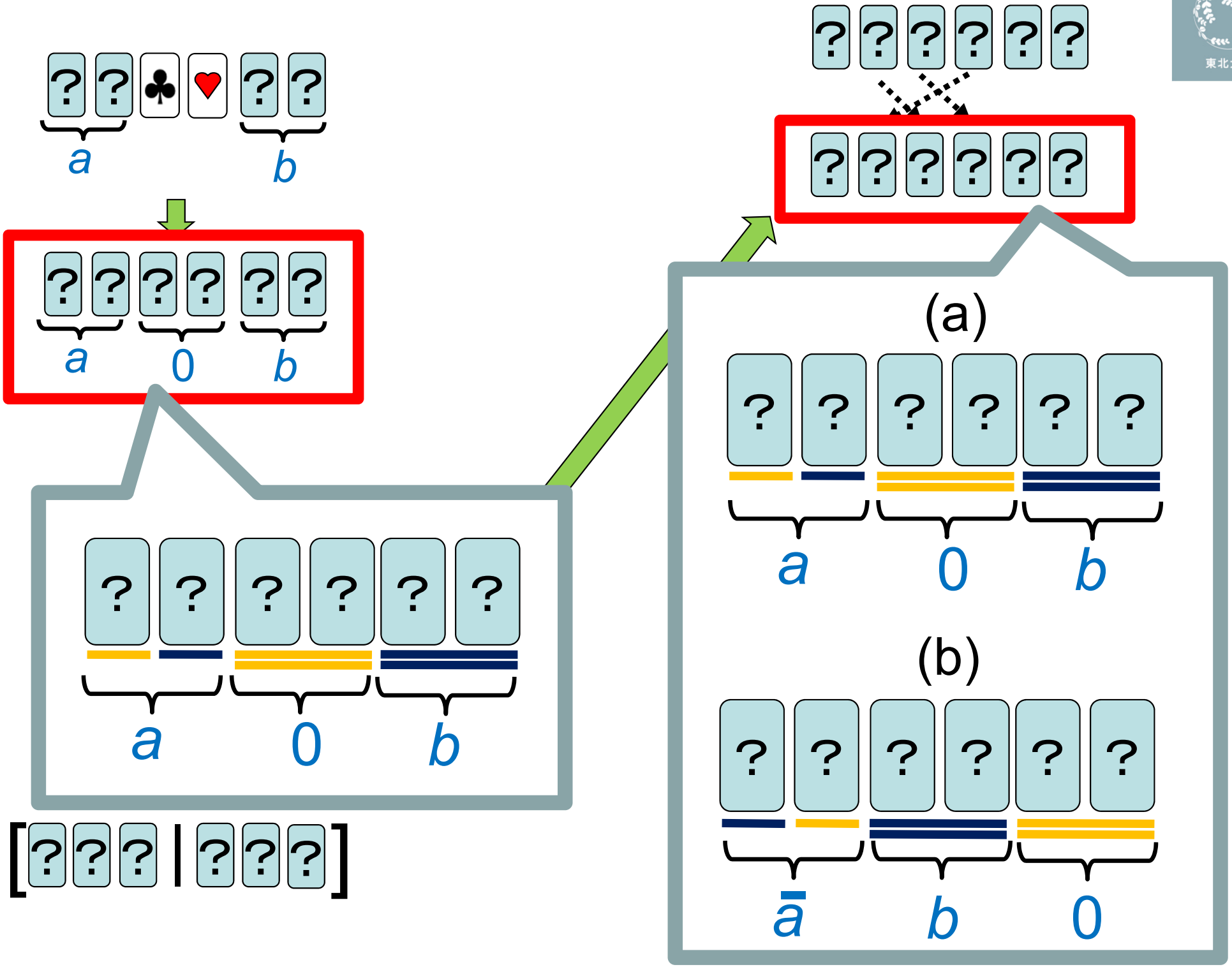
(a)

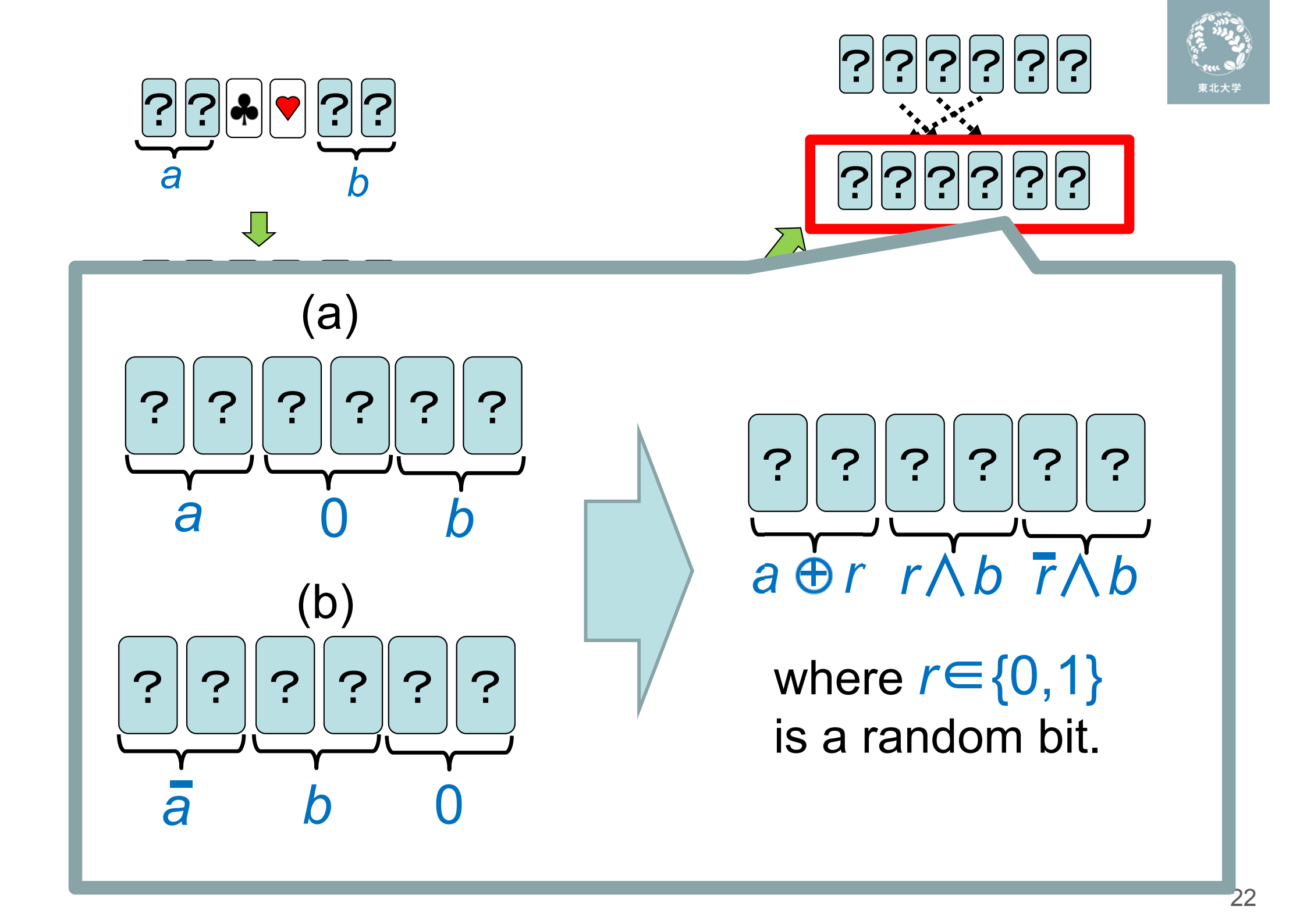


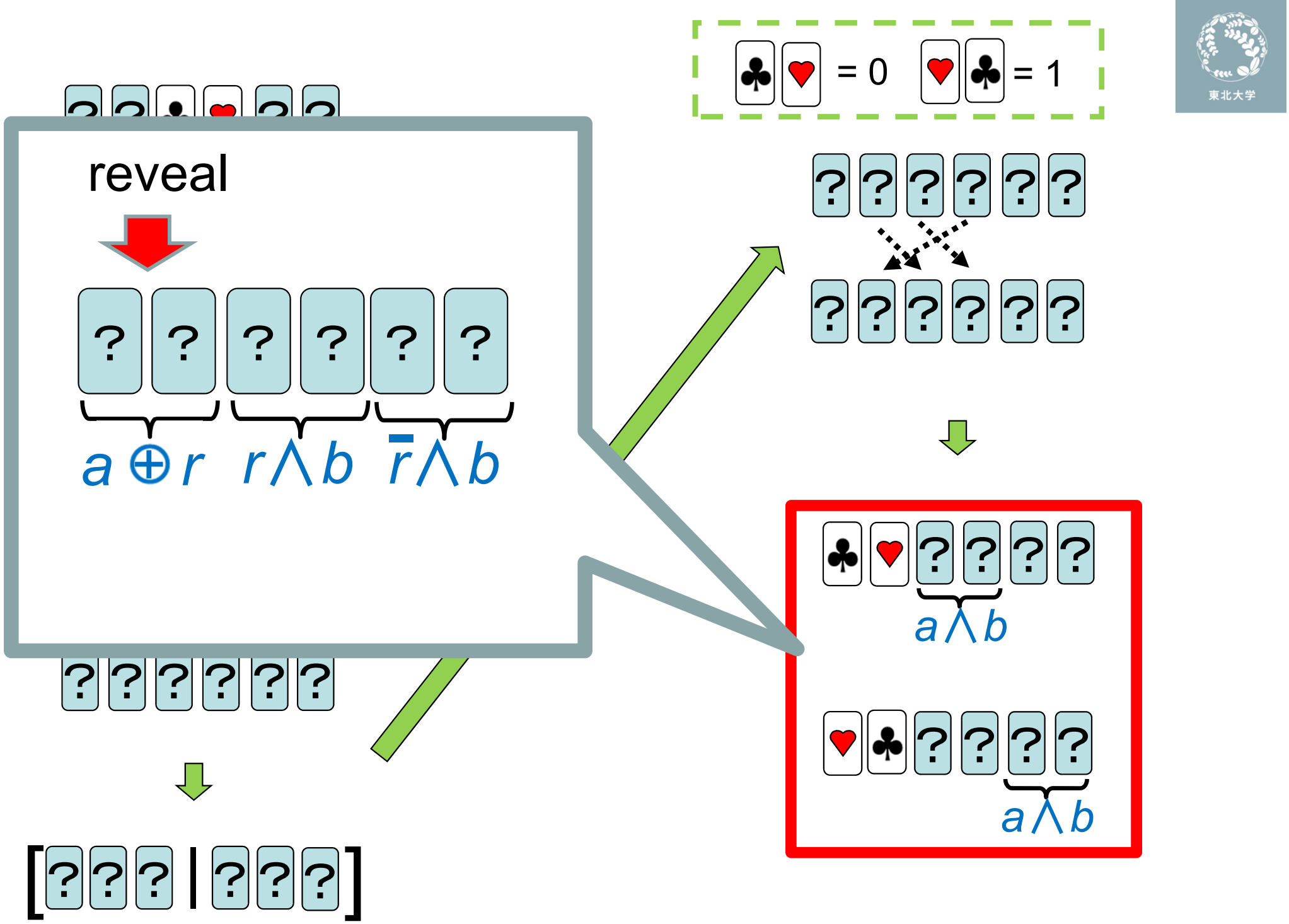
(b)

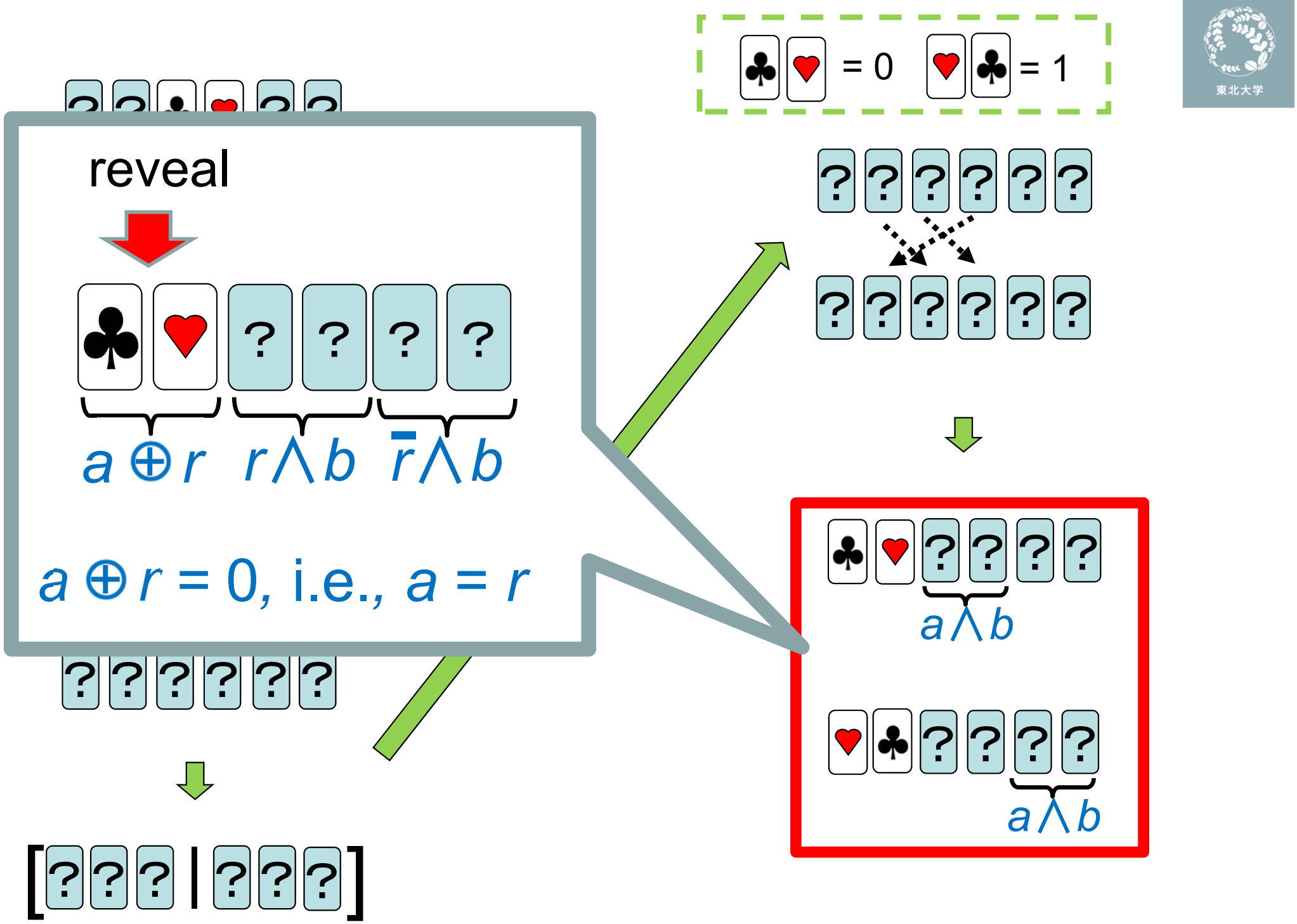


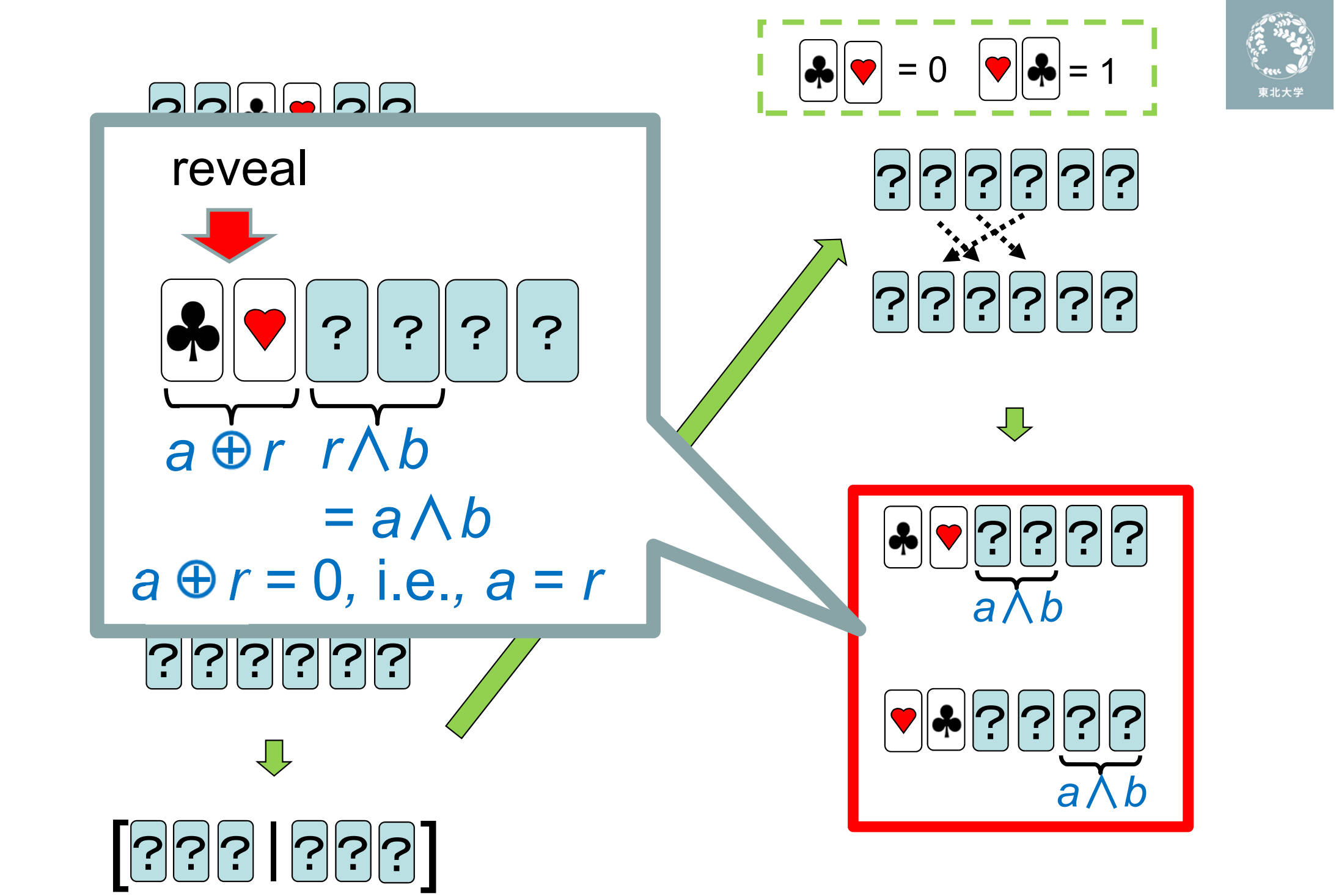


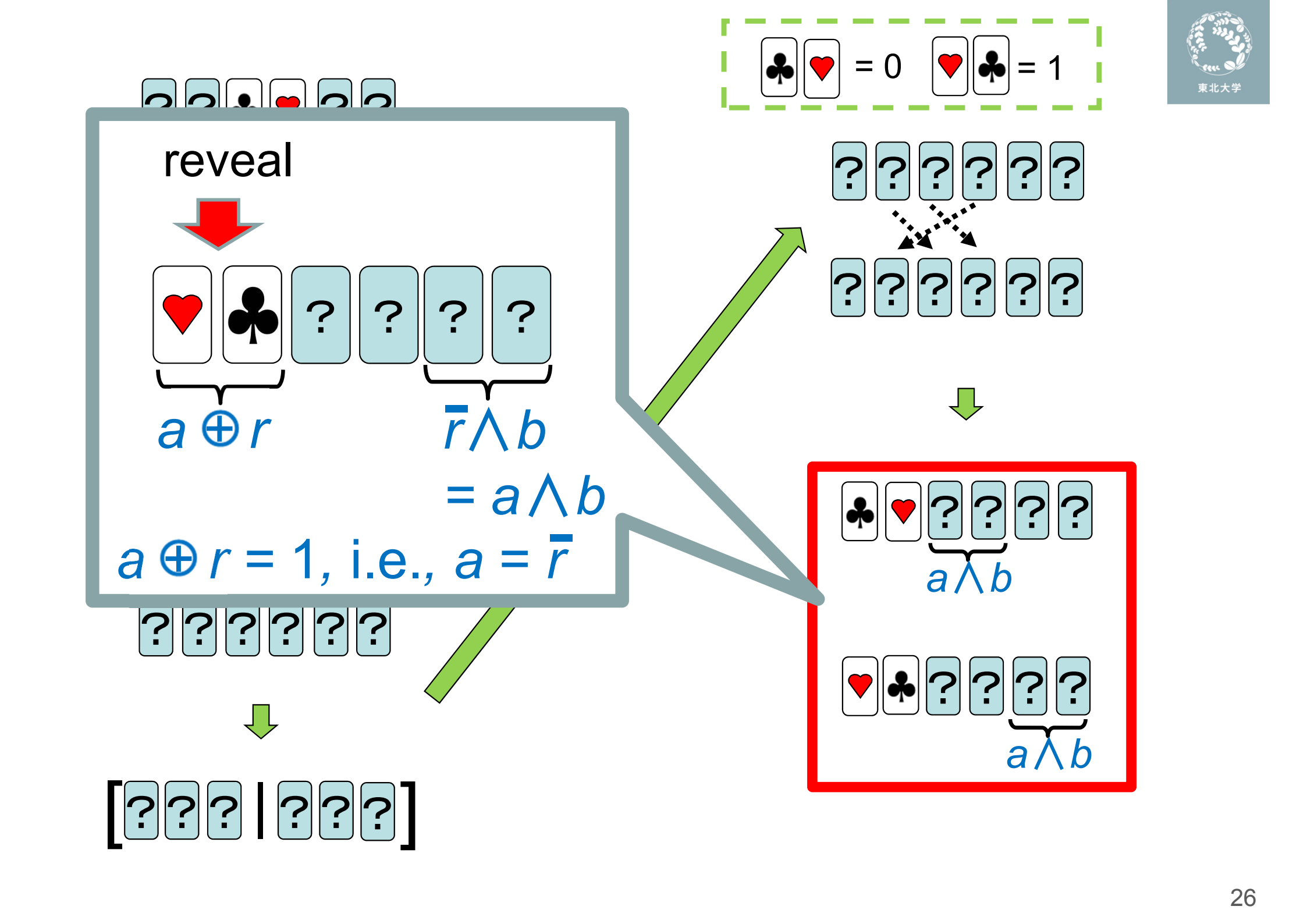

a
b
a
0
b
a
0
b
[|]
(a)
a
0
b
(b)
ā
b
0

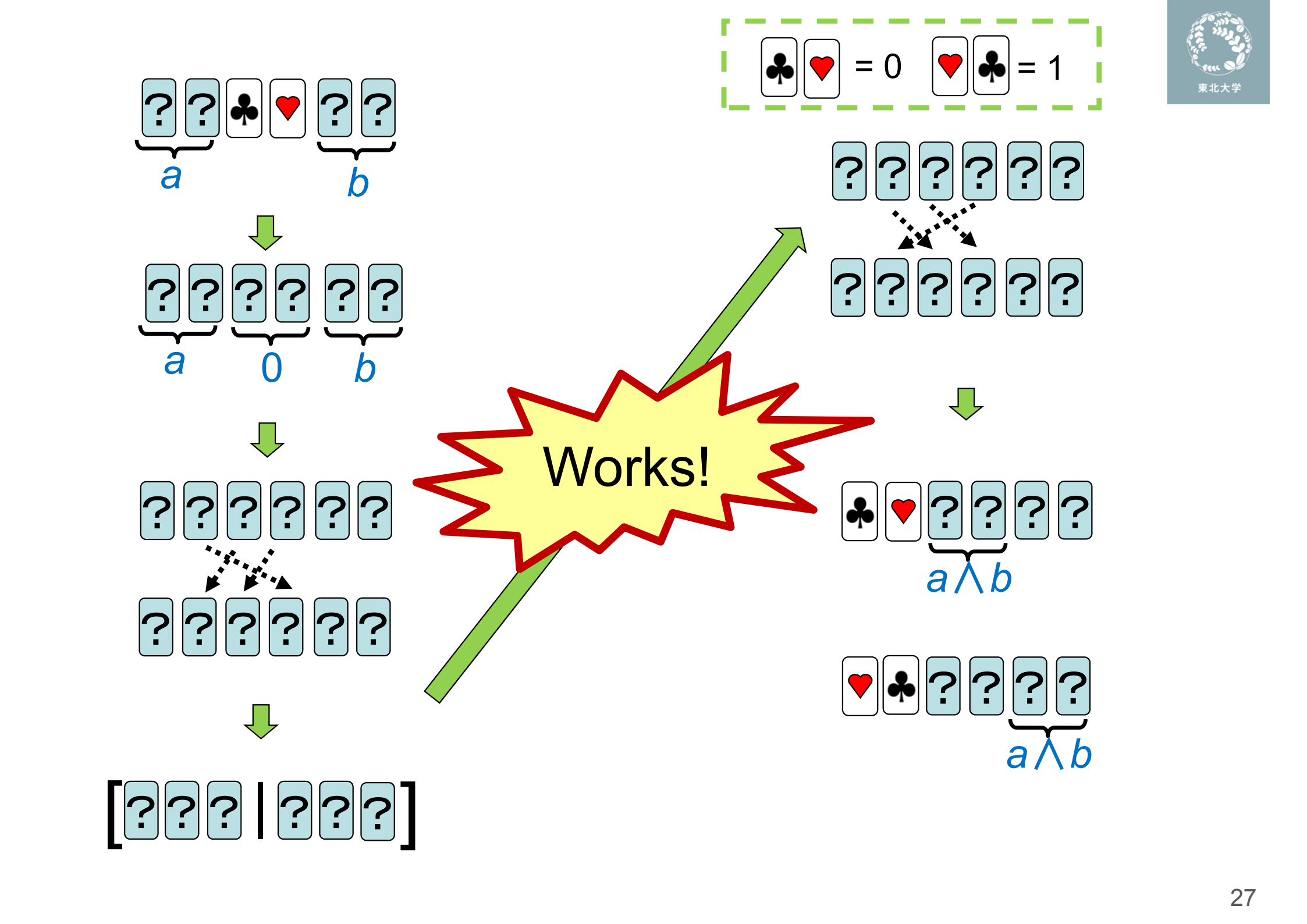

a
b
(a)
a
0
b
(b)
ā
b
0
a ⊕ r
r ∧ b
r̄ ∧ b
where r∈{0,1}
is a random bit.


= 0
= 1
reveal
$a \oplus r$
$r \wedge b$
$\bar{r} \wedge b$
$a \wedge b$
$a \wedge b$

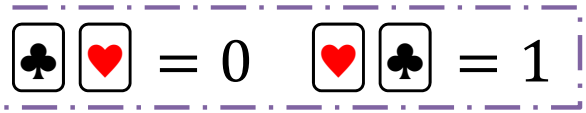

= 0
= 1
reveal
$a \oplus r$
$r \wedge b$
$\bar{r} \wedge b$
$a \oplus r = 0$, i.e., $a = r$
$a \wedge b$
$a \wedge b$


= 0
= 1
reveal
a ⊕ r
r ∧ b
= a ∧ b
a ⊕ r = 0, i.e., a = r
a ∧ b
a ∧ b

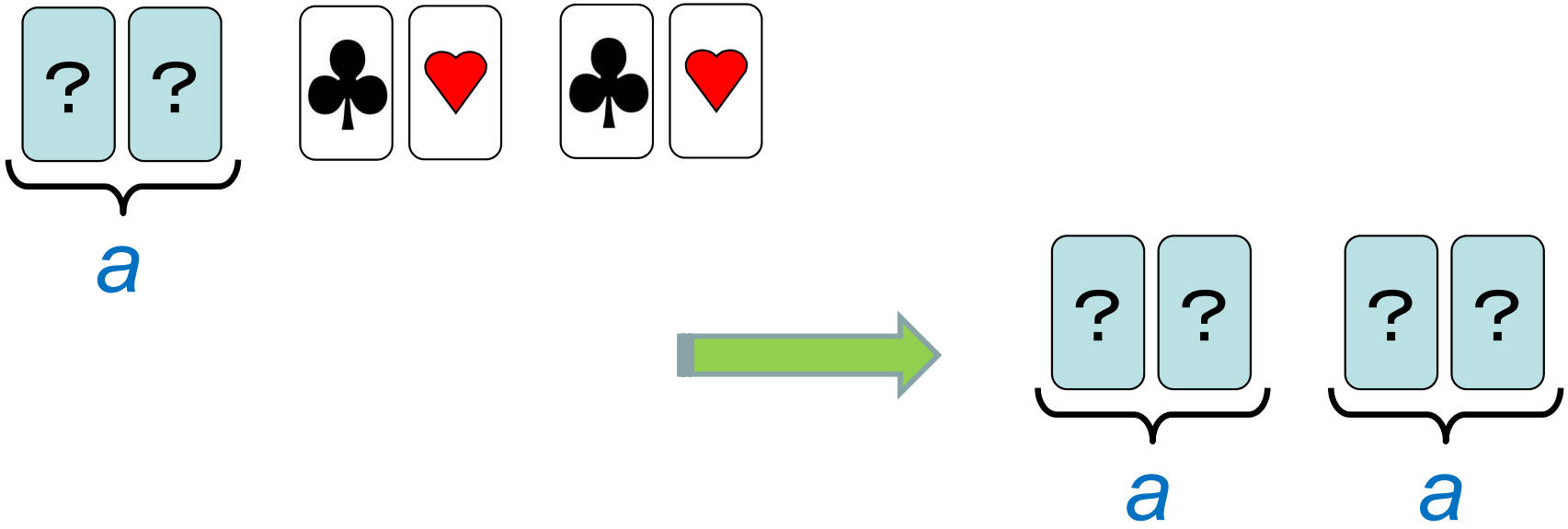

= 0
= 1
reveal
$a \oplus r$
$\bar{r} \wedge b$
$= a \wedge b$
$a \oplus r = 1$, i.e., $a = \bar{r}$
$a \wedge b$
$a \wedge b$


= 0
= 1
a
b
a
0
b
Works!
a ∧ b
a ∧ b

The Copy Protocol [10]

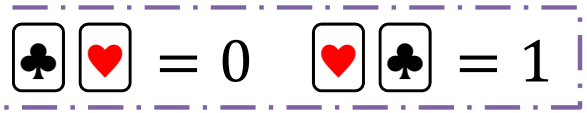


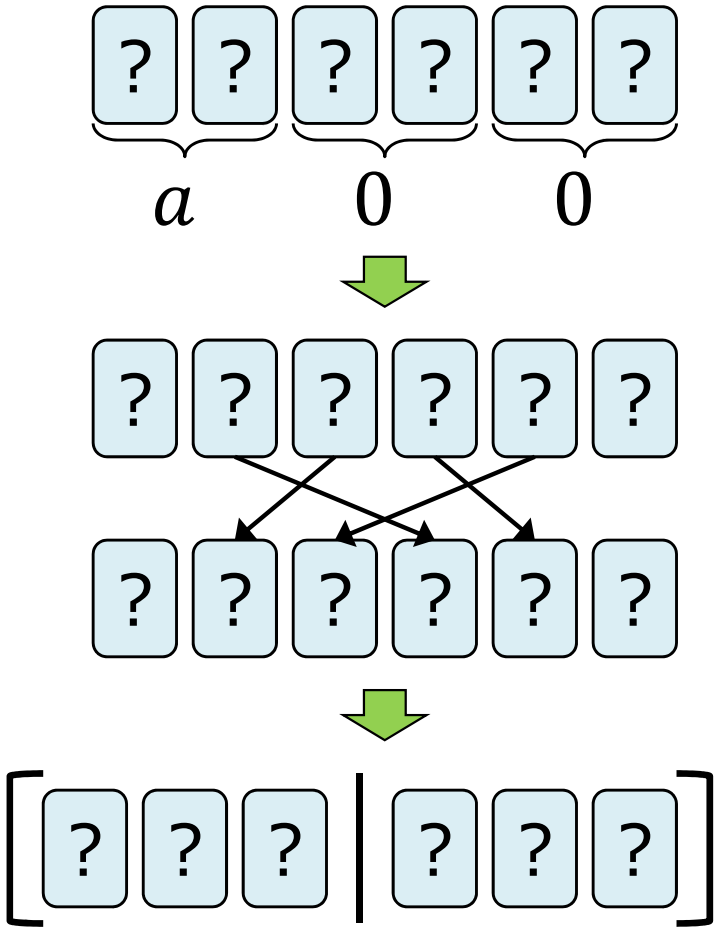
A commitment can be copied with 4 additional cards [10].

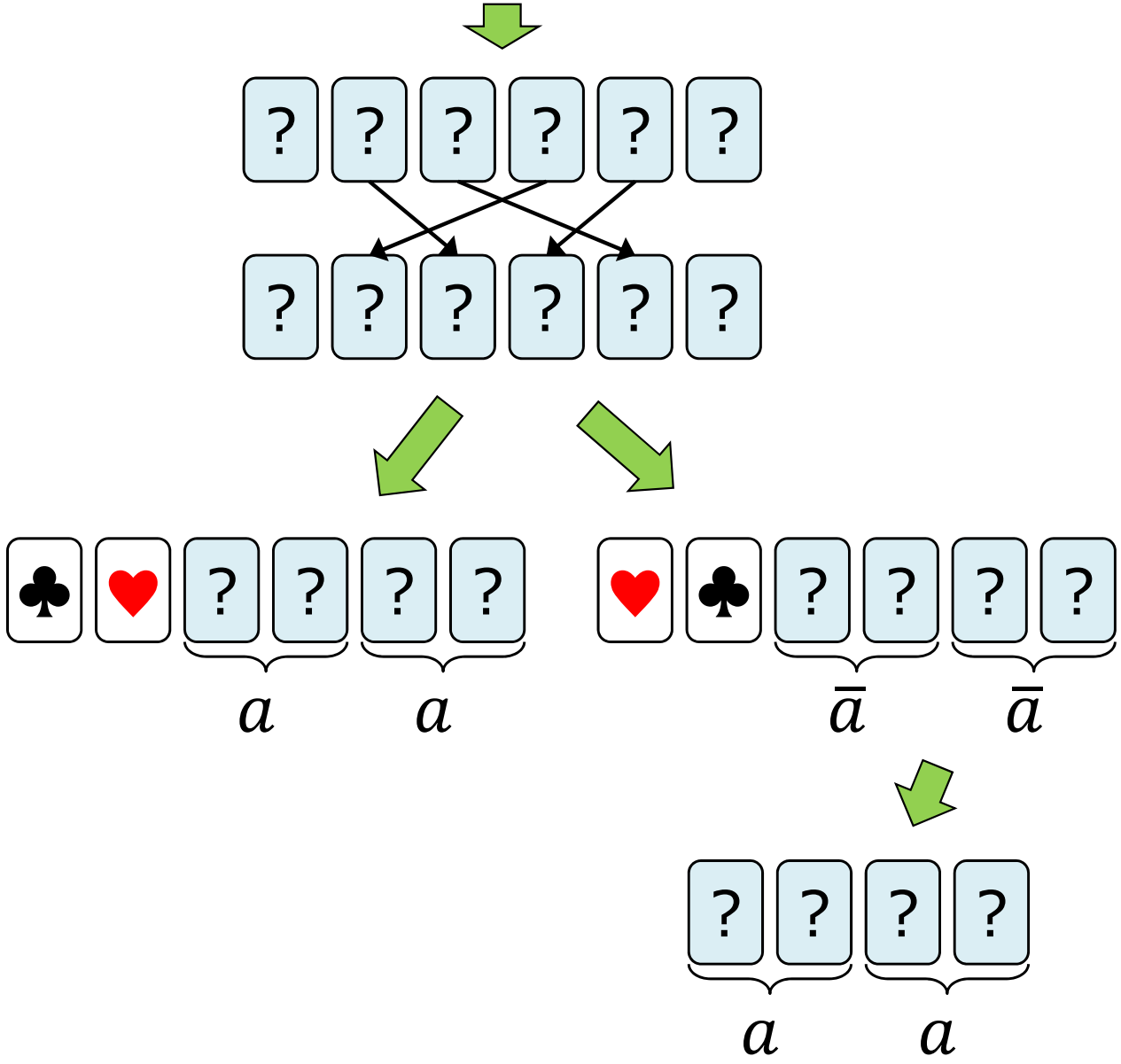


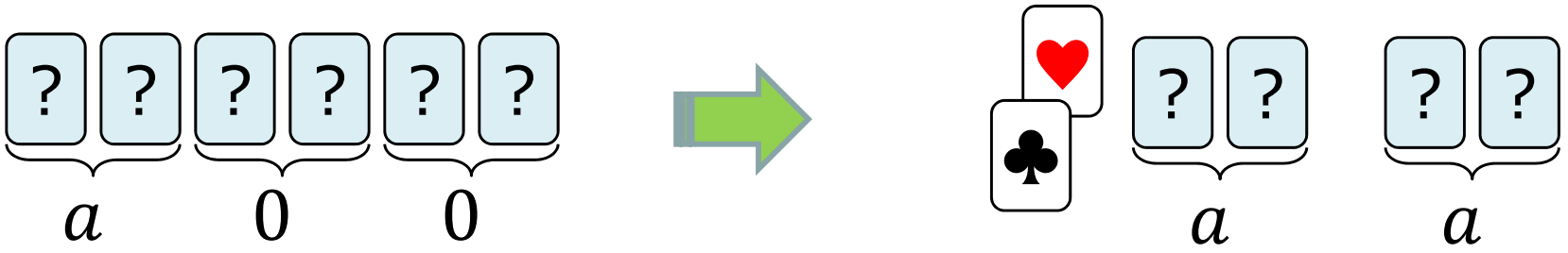
[10] T. Mizuki and H. Sone, Six-Card Secure AND and Four-Card Secure XOR, FAW 2009, LNCS 5598, pp. 358–369, 2009.

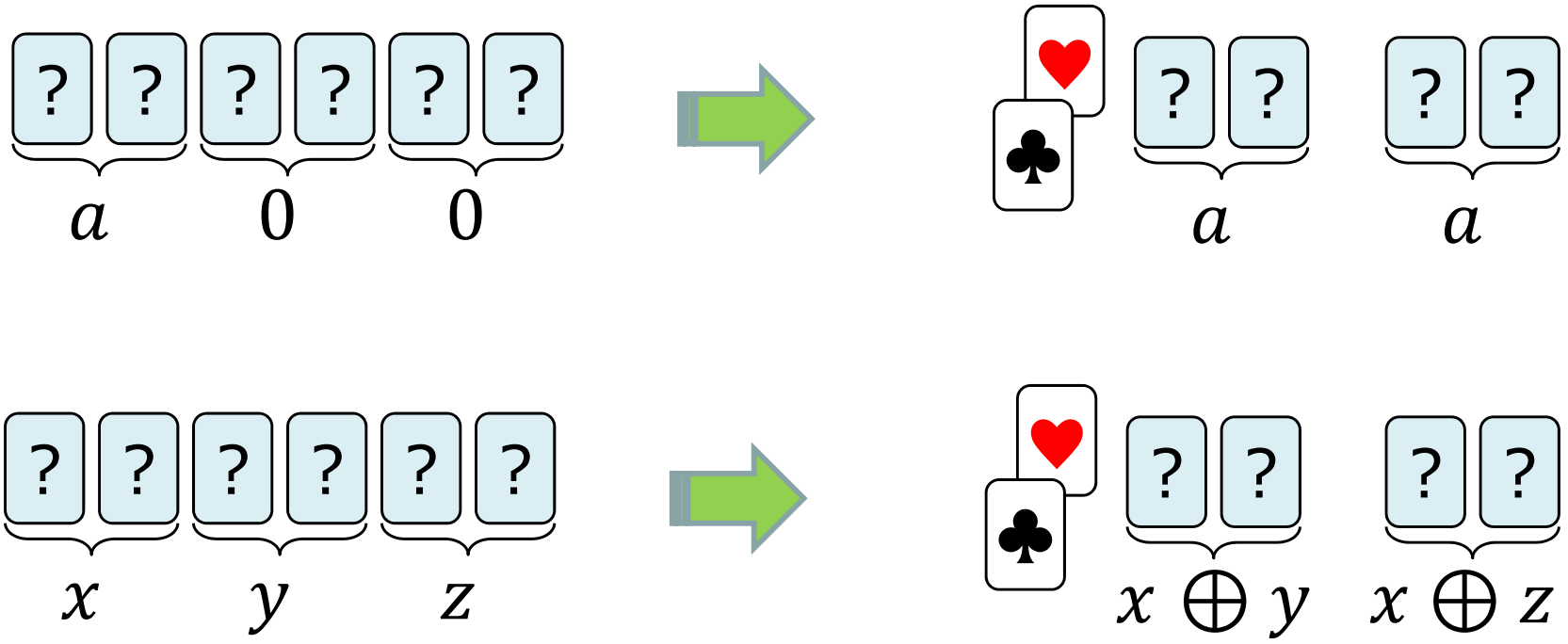
The Copy Protocol [10]

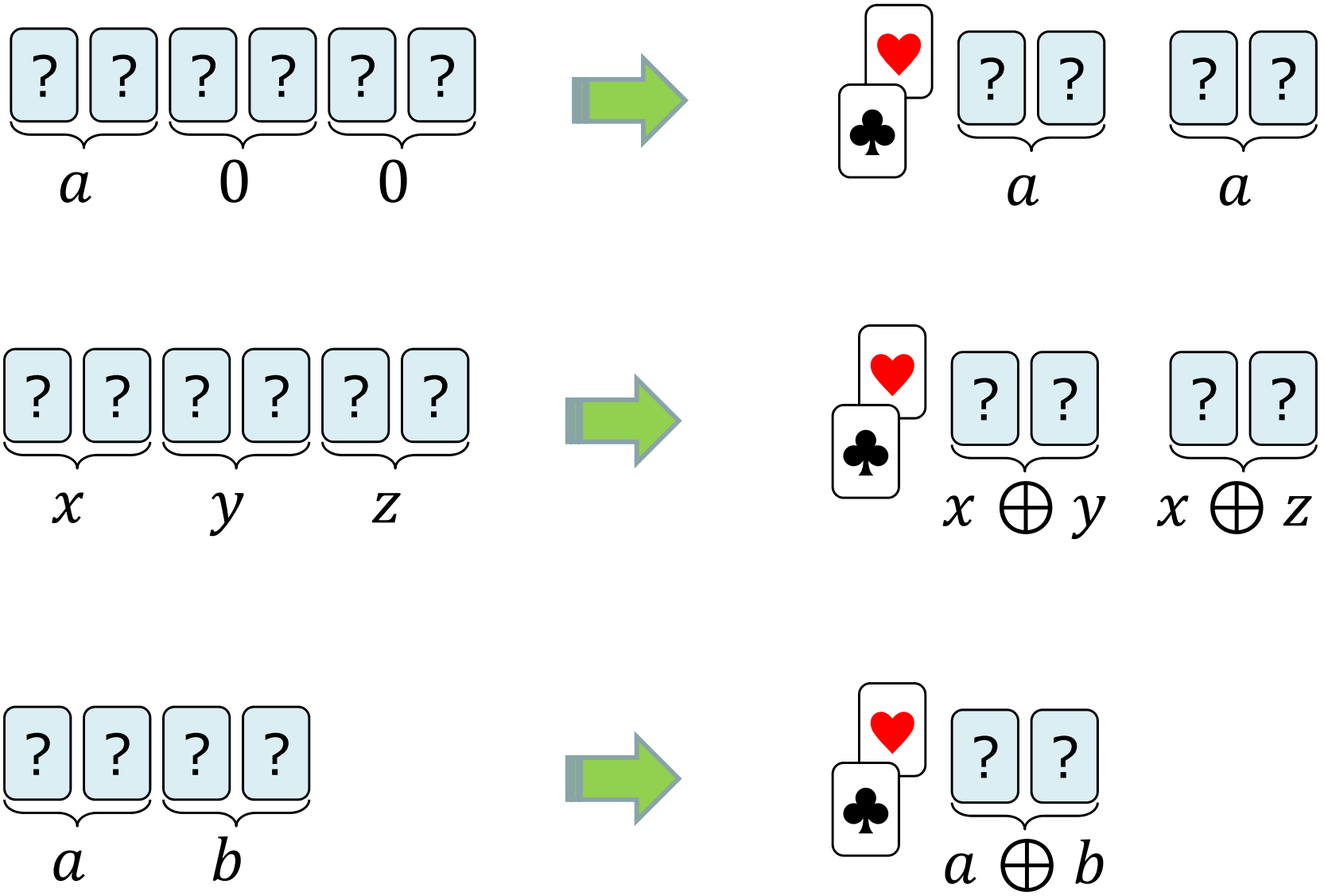






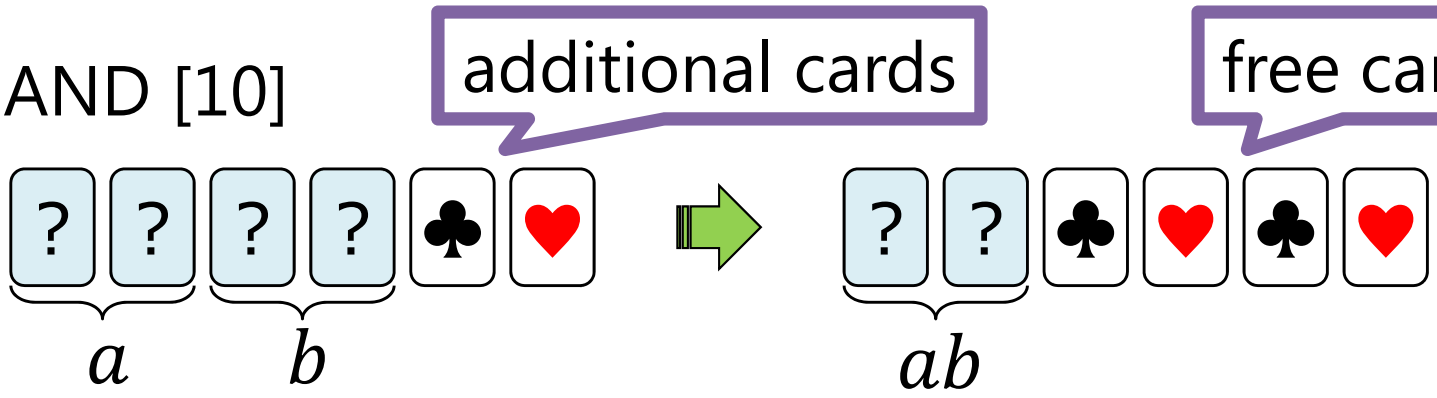

?
?
?
?
?
?
a
0
0
?
?
?
?
a
a


? ? ? ? ? ?
a 0 0
? ? ? ? ? ?
x y z
? ? ? ? ? ?
a a
? ? ? ? ? ?
x ⊕ y x ⊕ z

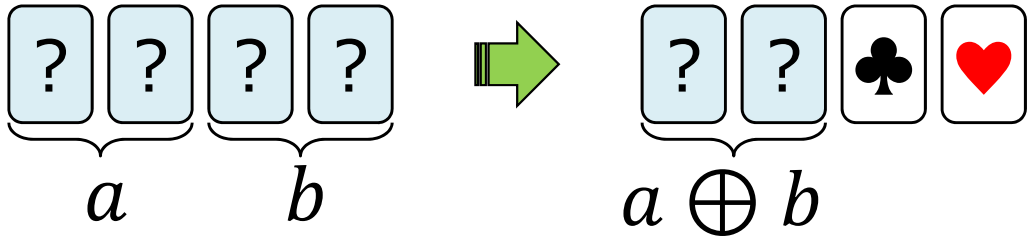


Secure XOR can be done with no additional card [10].

Secure AND [10]

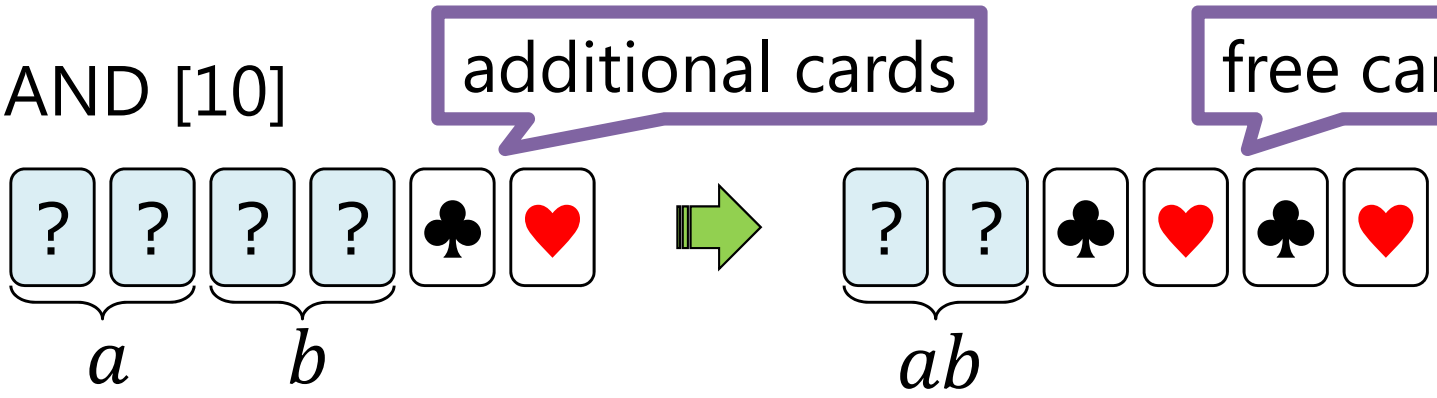


Secure XOR [10]

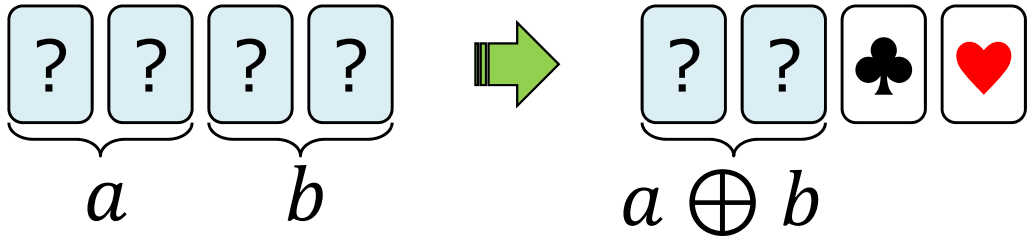


[10] T. Mizuki and H. Sone, Six-Card Secure AND and Four-Card Secure XOR, FAW 2009, LNCS 5598, pp. 358–369, 2009.

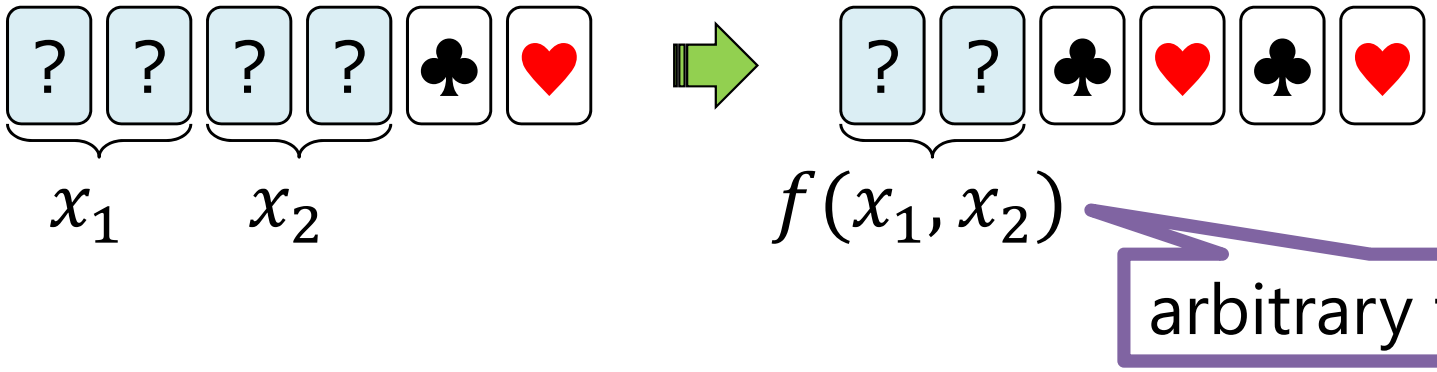
Secure AND [10]



Secure XOR [10]

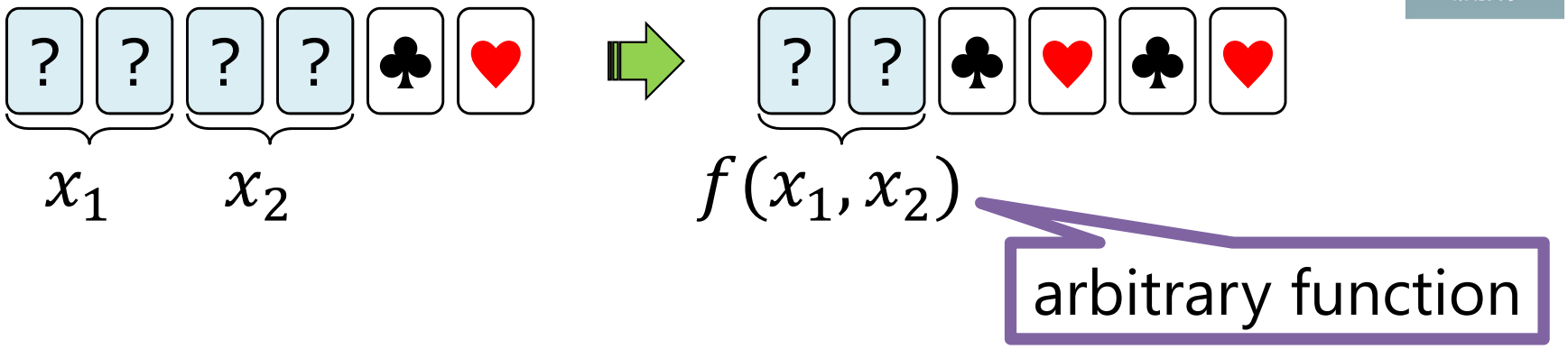


Theorem 1 ([10])

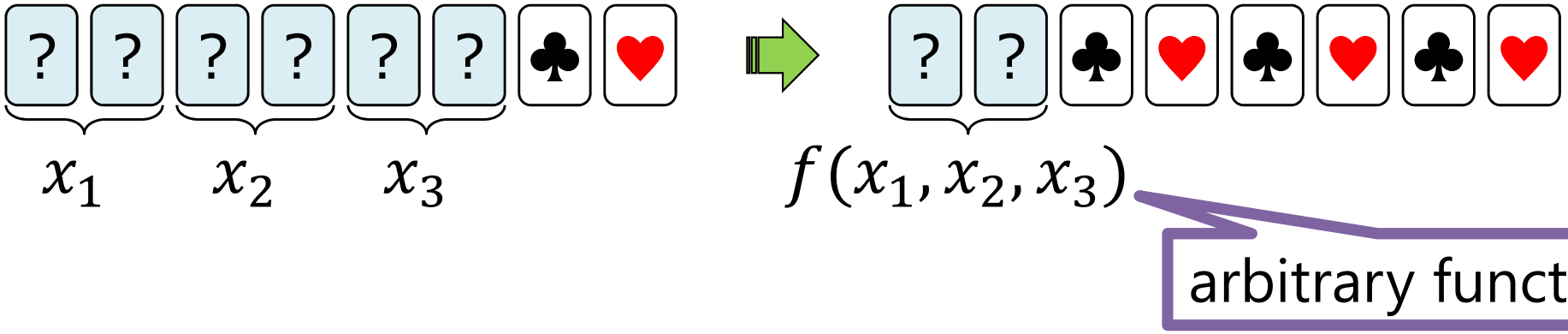


[10] T. Mizuki and H. Sone, Six-Card Secure AND and Four-Card Secure XOR, FAW 2009, LNCS 5598, pp. 358–369, 2009.

Theorem 1 ([10])

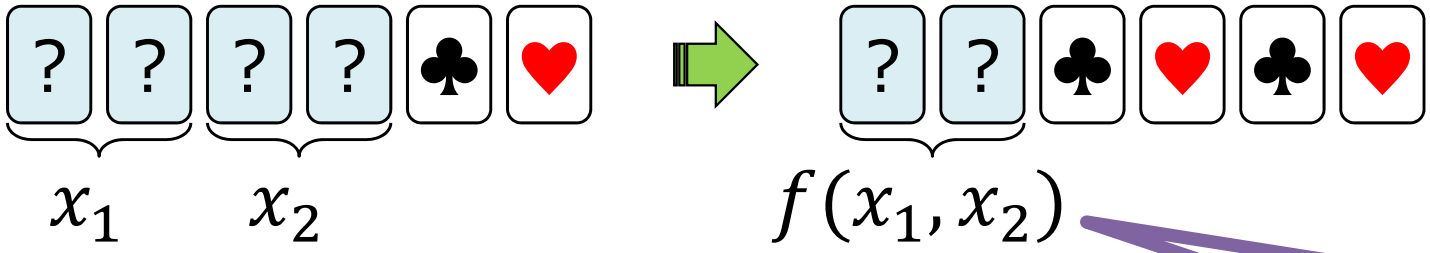


Theorem 2 ([13])

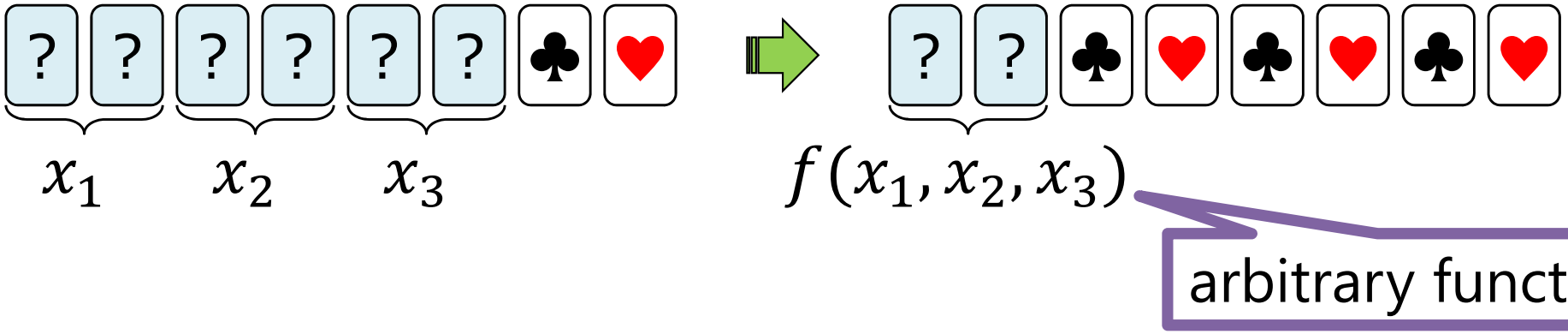


[13] T. Nishida, Y. Hayashi, T. Mizuki, and H. Sone, Securely computing three-input functions with eight cards, IEICE Trans. Fundamentals, vol.E98-A, no.6, 2015.

Theorem 1 ([10])



Theorem 2 ([13])



2 additional cards are sufficient for $n \leqq 3$

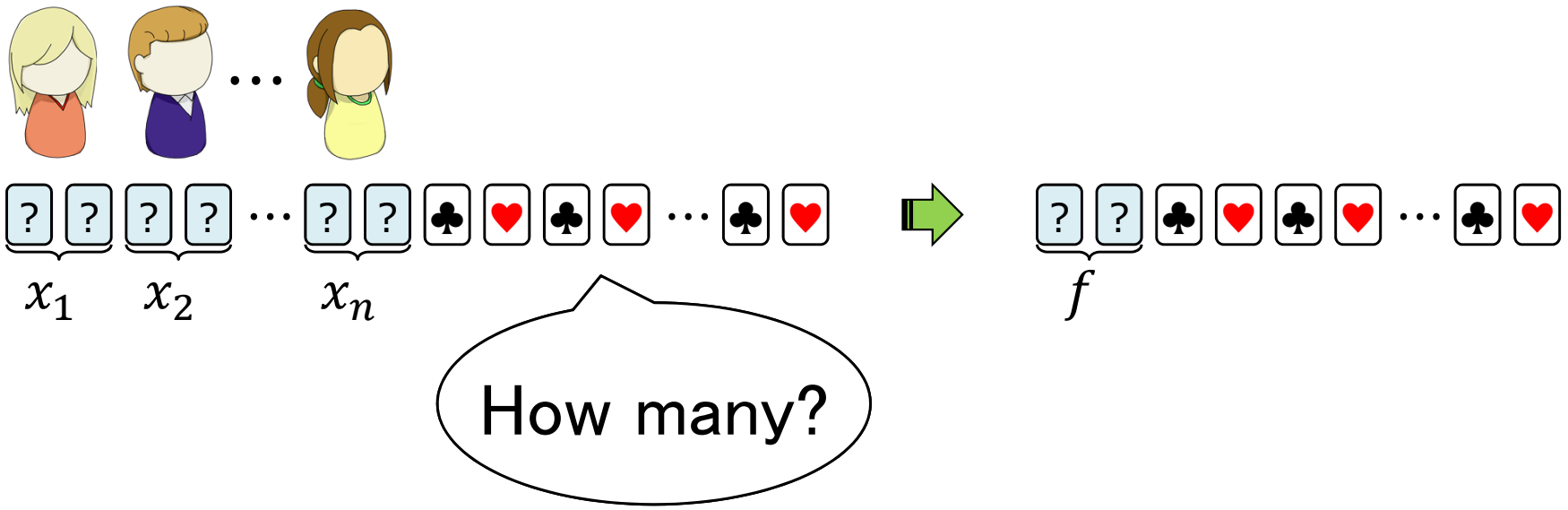
東北大学

➢ How about functions having 4 or more variables?

Since we have AND, XOR, NOT, and copy protocols, any Boolean function $f(x_1, x_2, \ldots, x_n)$ can be securely computed as long as there are plenty of additional cards.

Open Problem

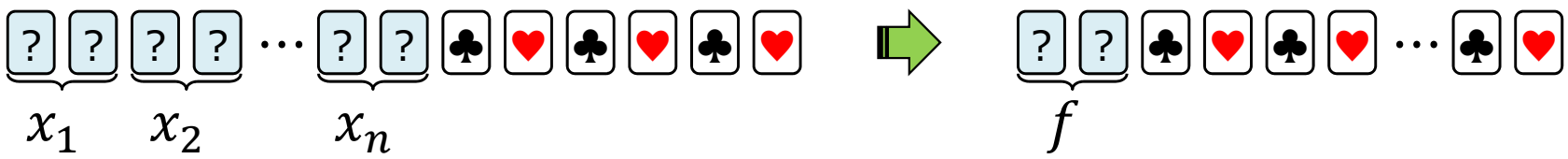
- An explicit sufficient number of additional cards has not been revealed thus far.



Our Results

We show sufficient conditions on the numbers of additional cards.

6 cards are enough



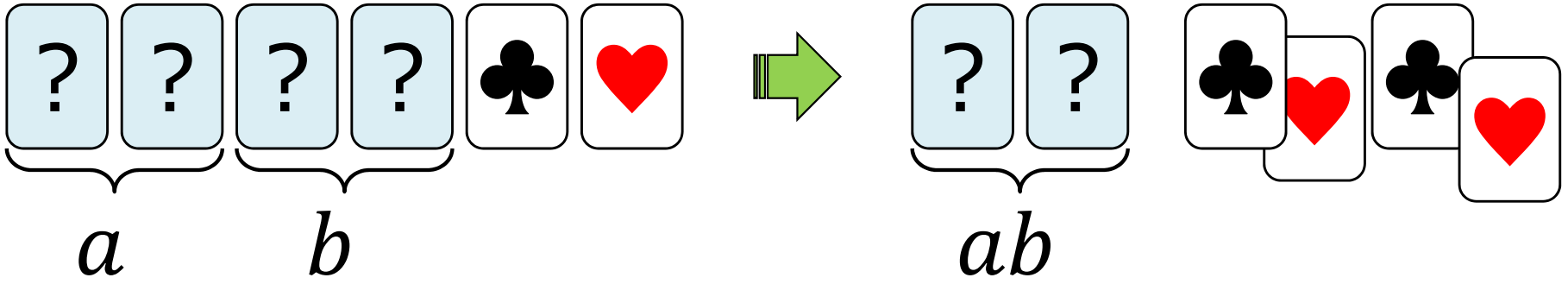
2 cards are enough

symmetric function

Table of Contents

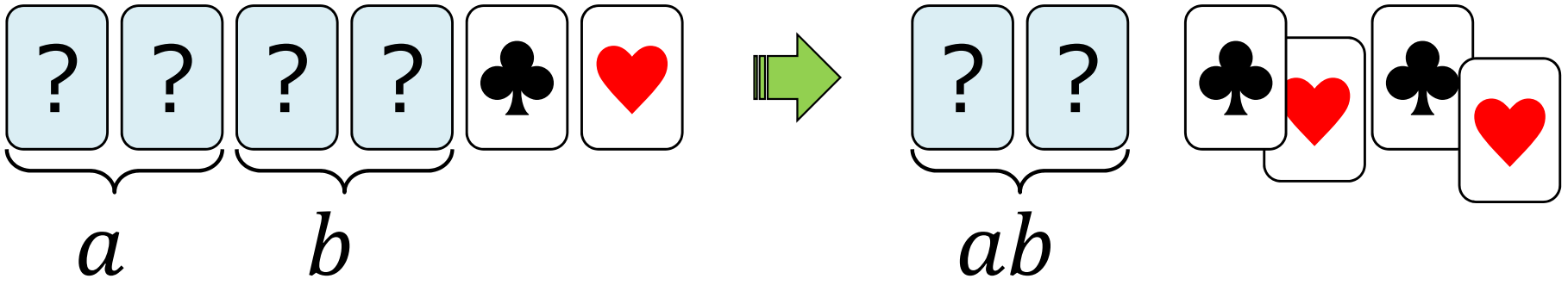
2.1 Improved AND Protocol

The existing AND protocol [10]

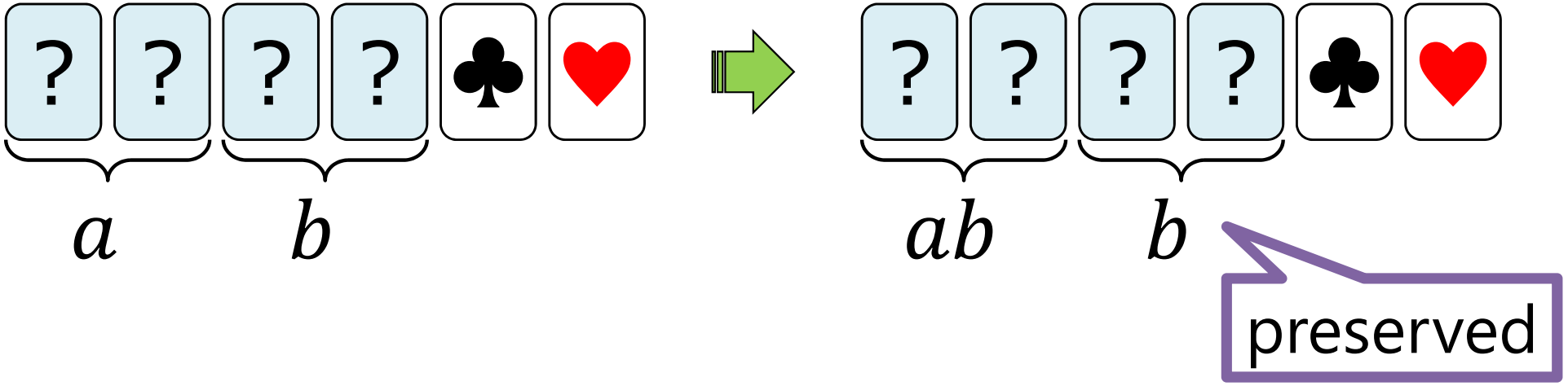


[10] T. Mizuki and H. Sone, Six-Card Secure AND and Four-Card Secure XOR, FAW 2009, LNCS 5598, pp. 358–369, 2009.

The existing AND protocol [10]



Our improved AND protocol

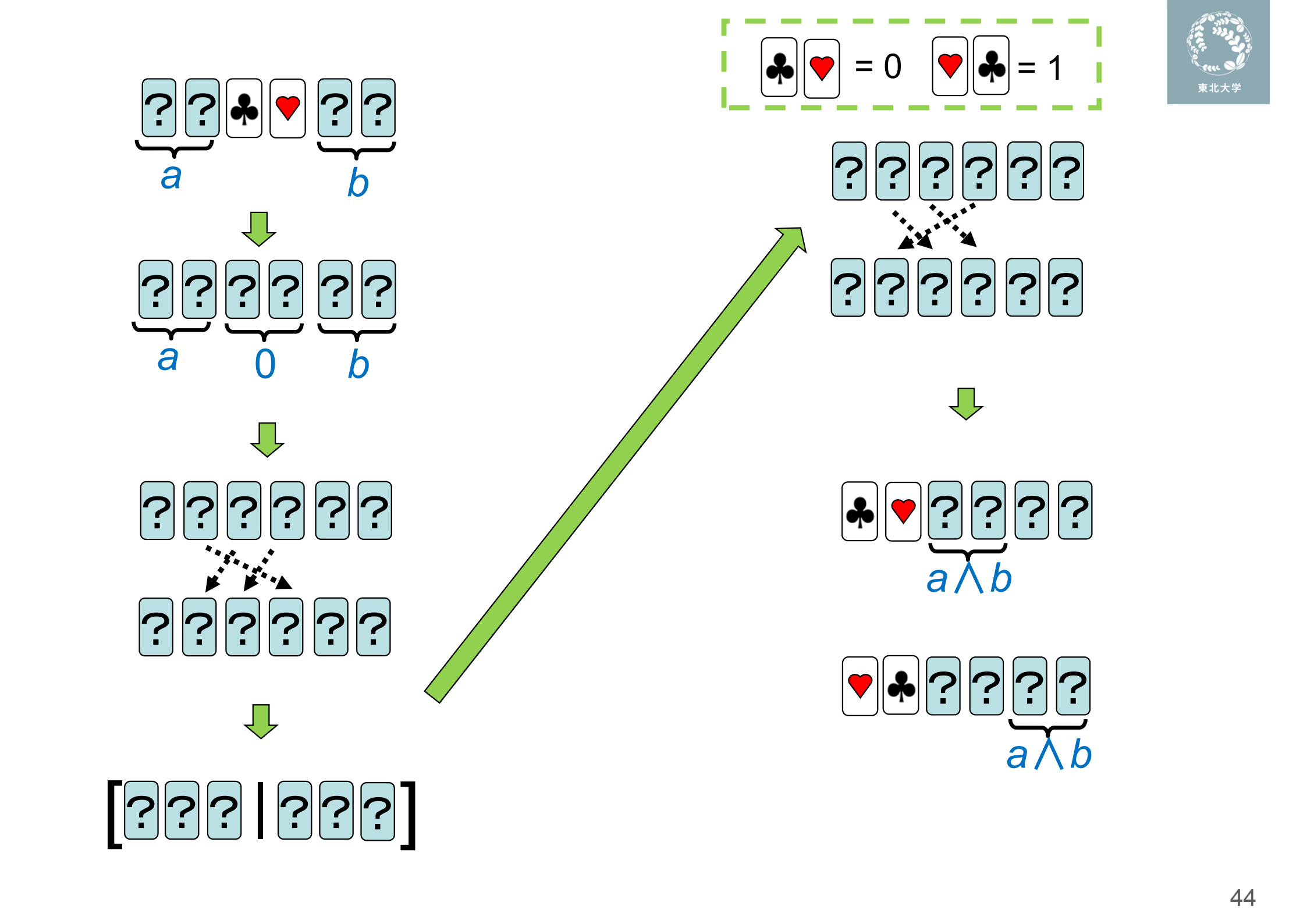


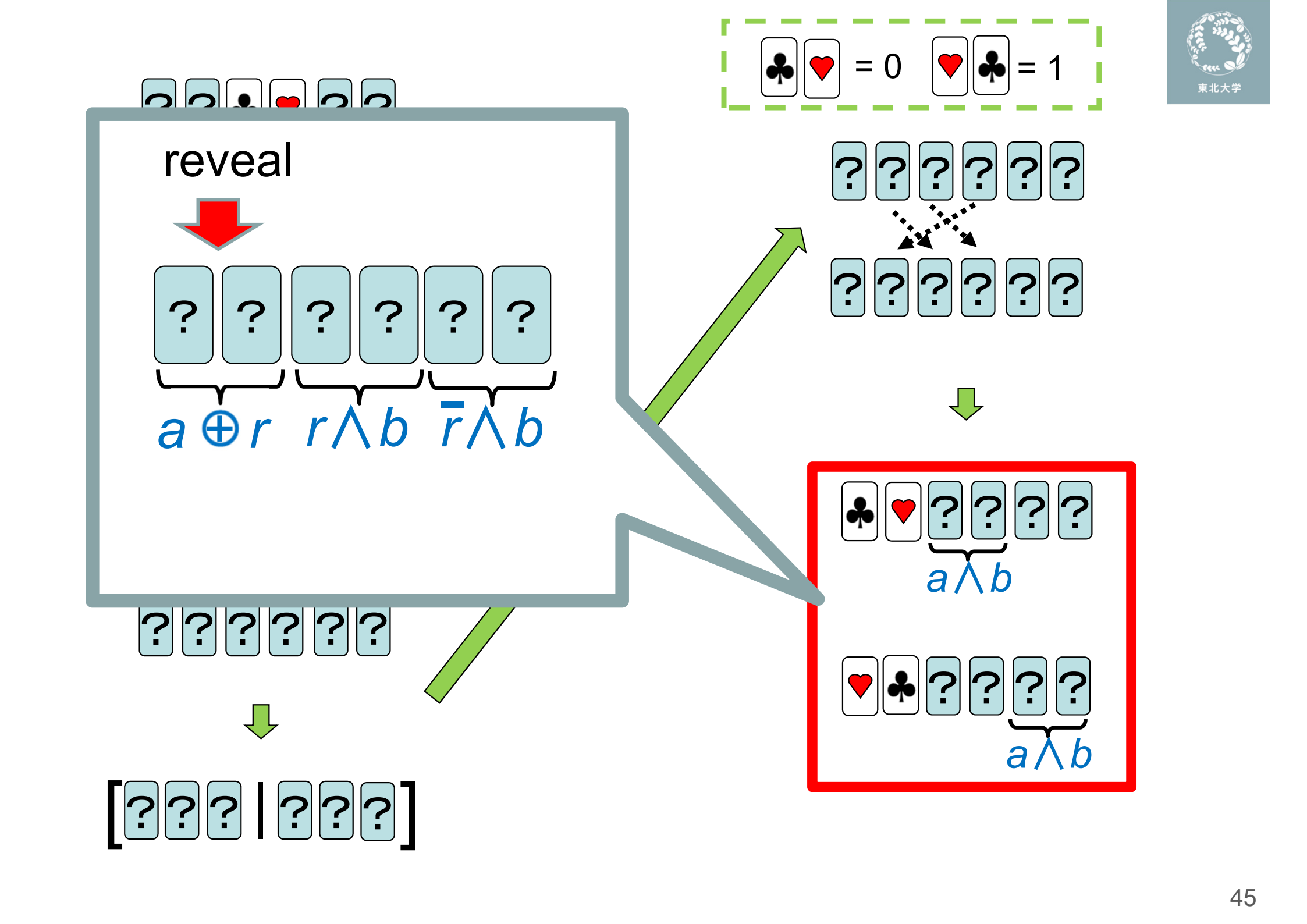
[10] T. Mizuki and H. Sone, Six-Card Secure AND and Four-Card Secure XOR, FAW 2009, LNCS 5598, pp. 358–369, 2009.

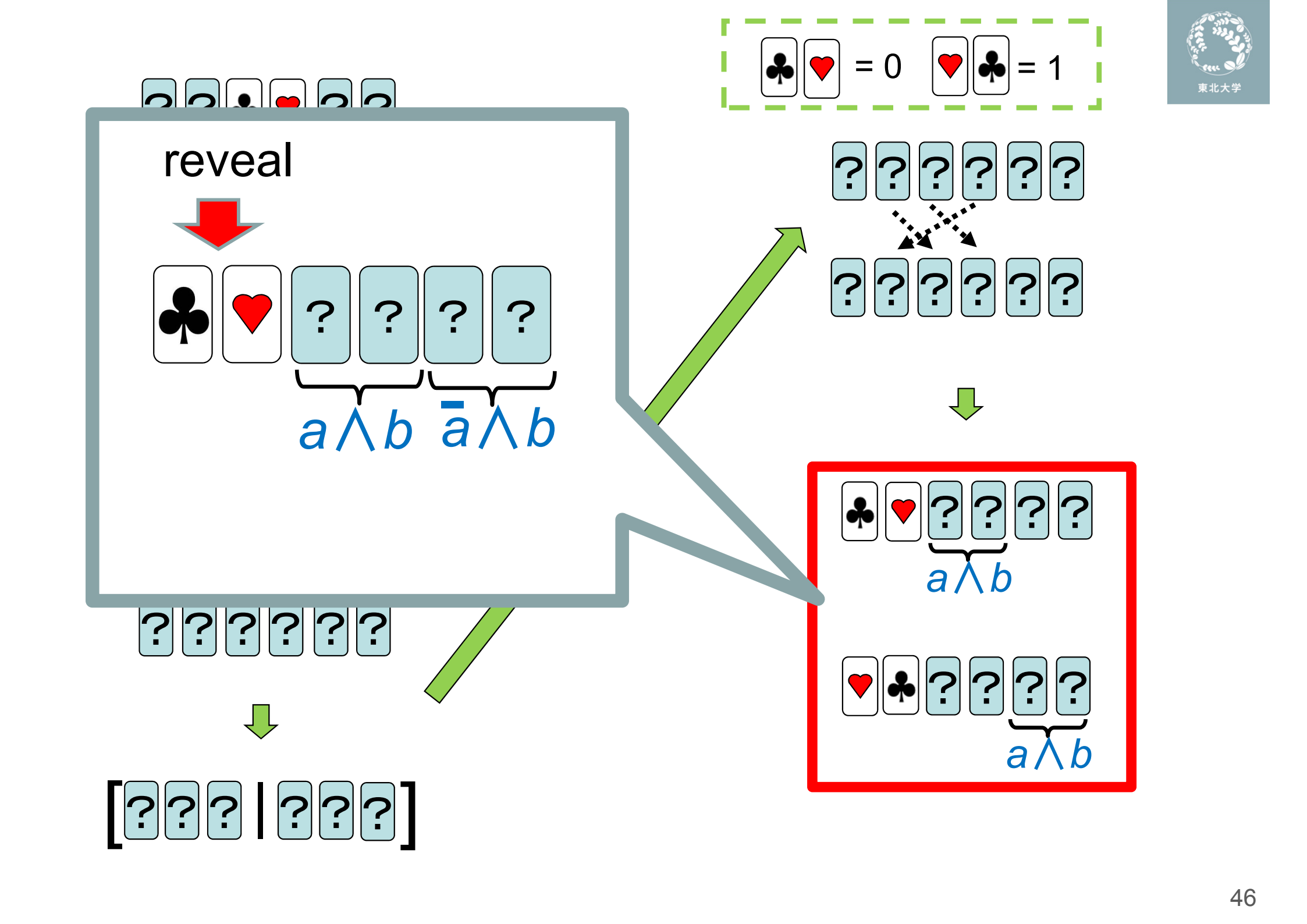
Idea:

(1) Revisit the existing AND protocol

(2) Identity: $ab \bigoplus \bar{a}b = b$

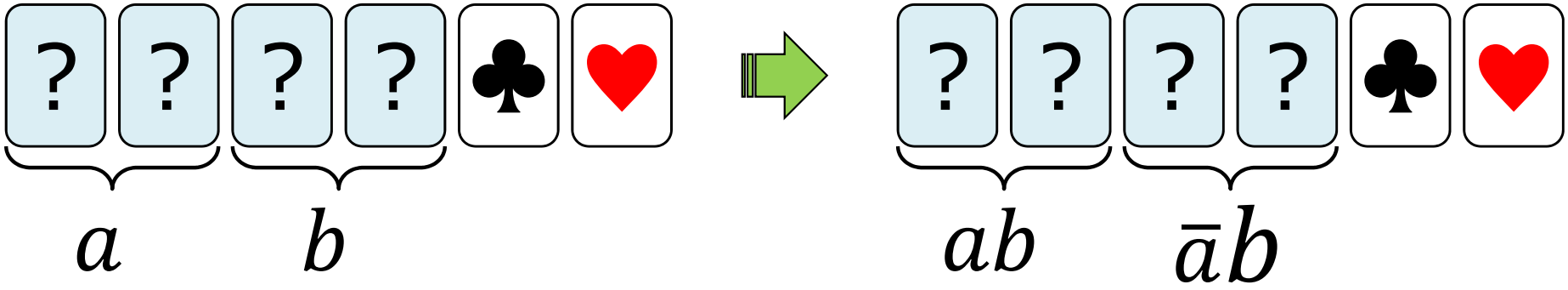

= 0
= 1
a
b
a
0
b
a ∧ b
a ∧ b


= 0
= 1
reveal
$a \oplus r$
$r \wedge b$
$\bar{r} \wedge b$
$a \wedge b$
$a \wedge b$


= 0
= 1
reveal
$a \wedge b$
$\bar{a} \wedge b$
$a \wedge b$
$a \wedge b$

Idea:

(1) Revisit the existing AND protocol

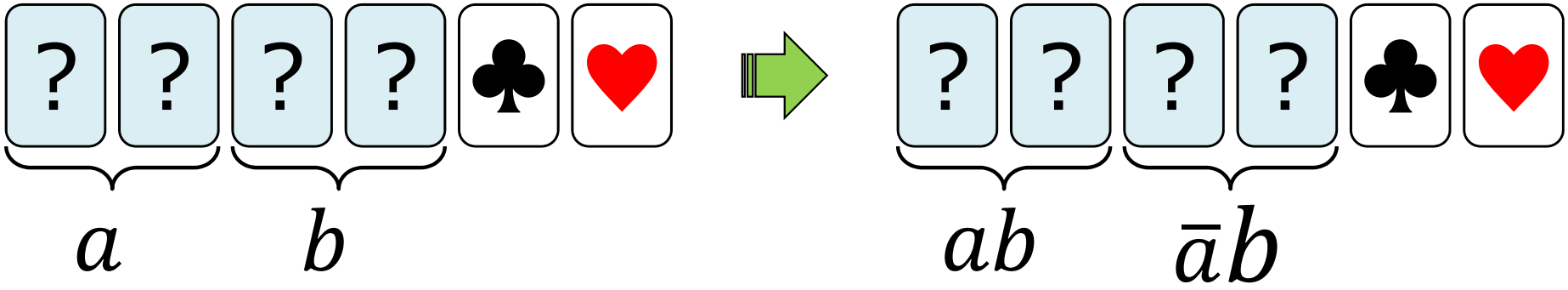


(2) Identity: $ab \oplus \bar{a}b = b$

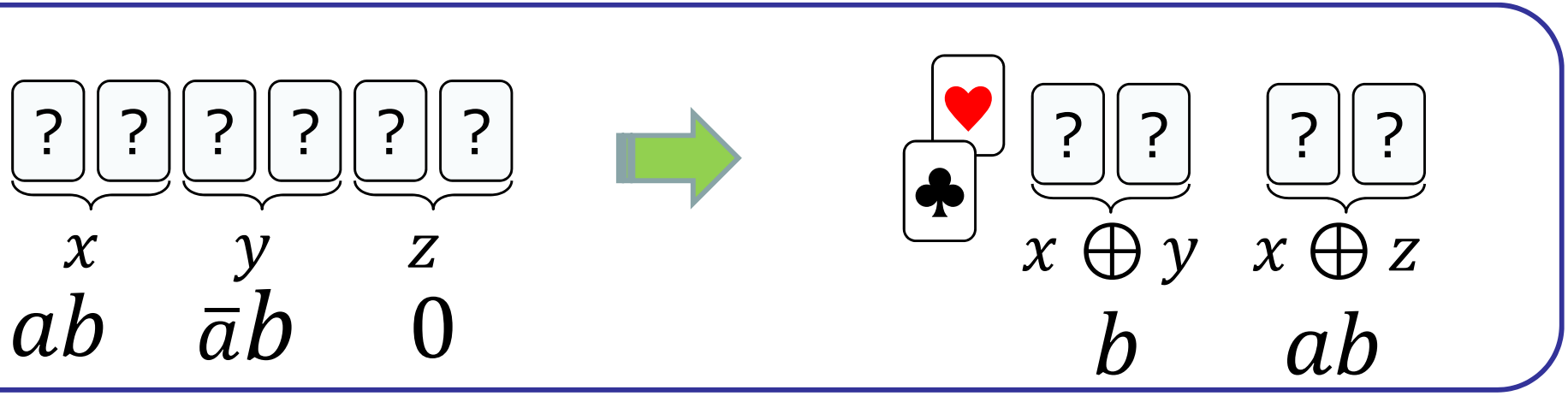
to be transformed into b

Idea:

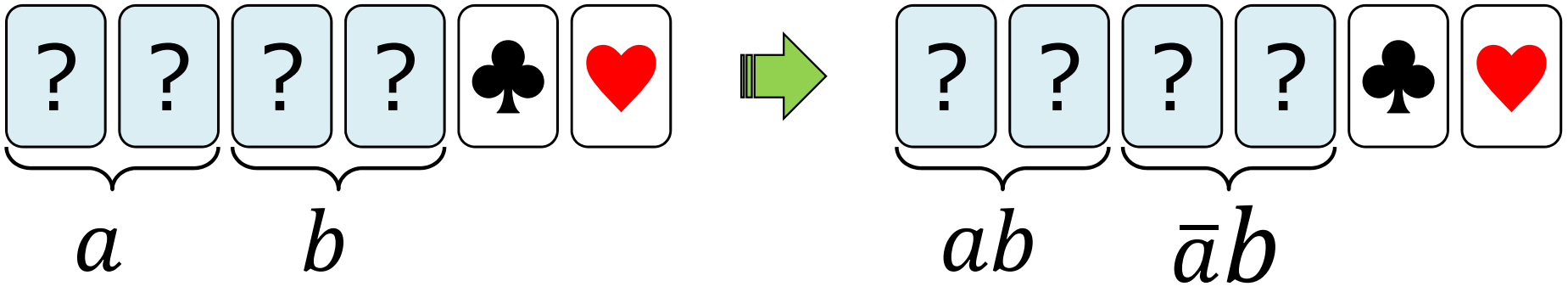
(1) Revisit the existing AND protocol



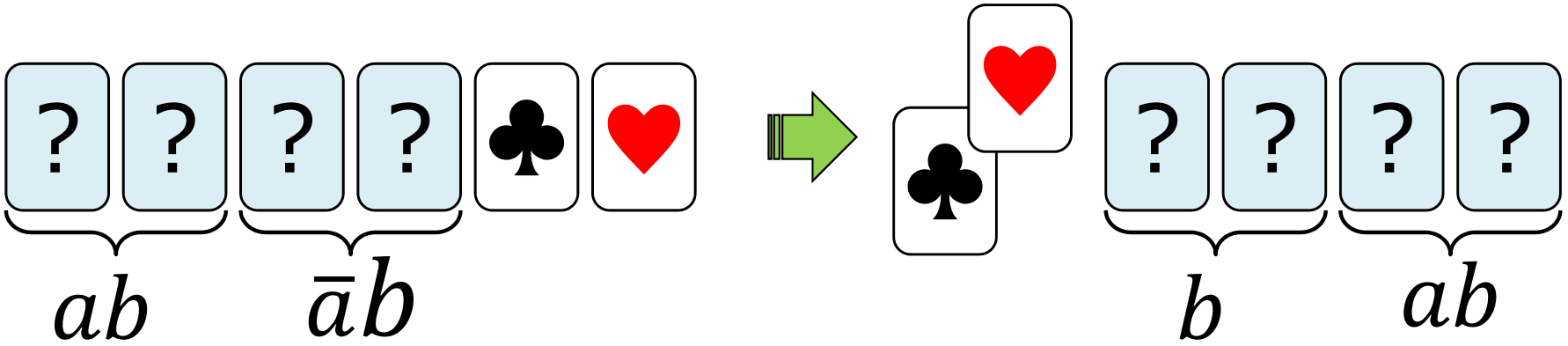
(2) Identity: $ab \oplus \bar{a}b = b$



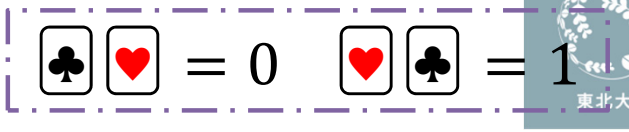
(1)

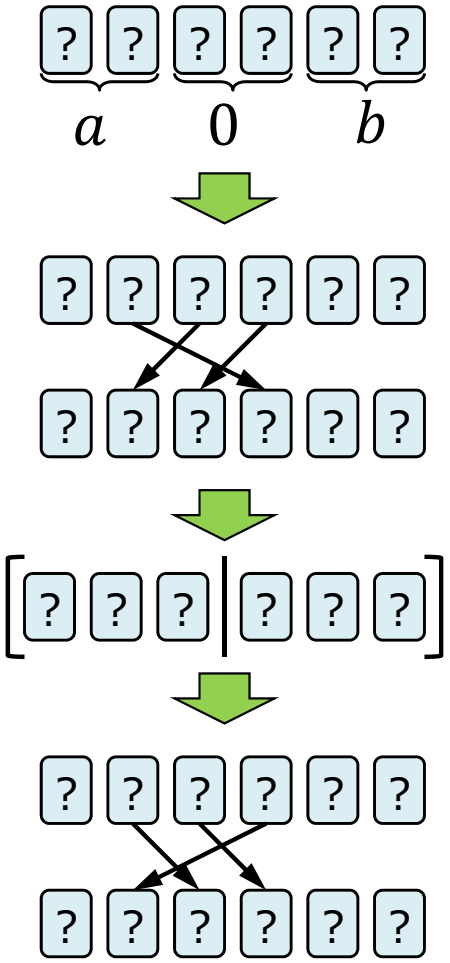


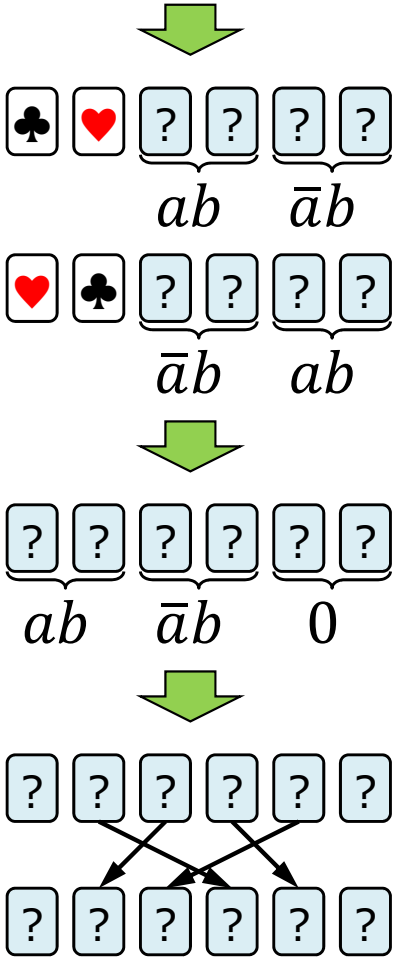
(2) $ab \oplus \bar{a}b = b$

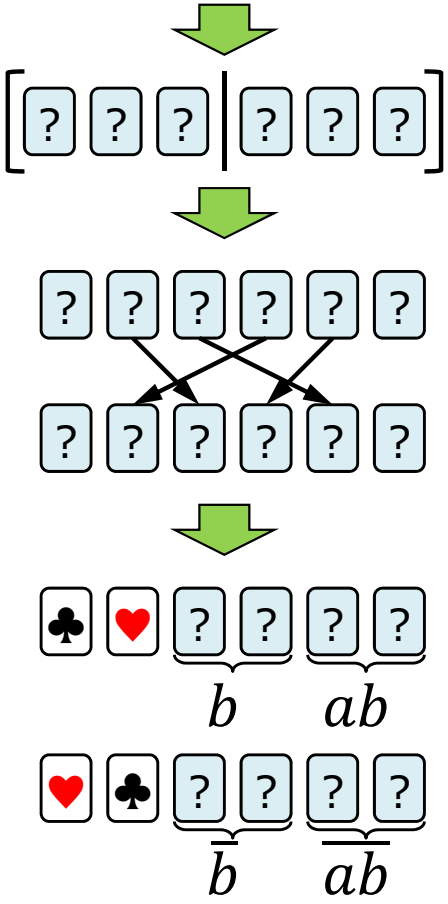


Full description of improved AND

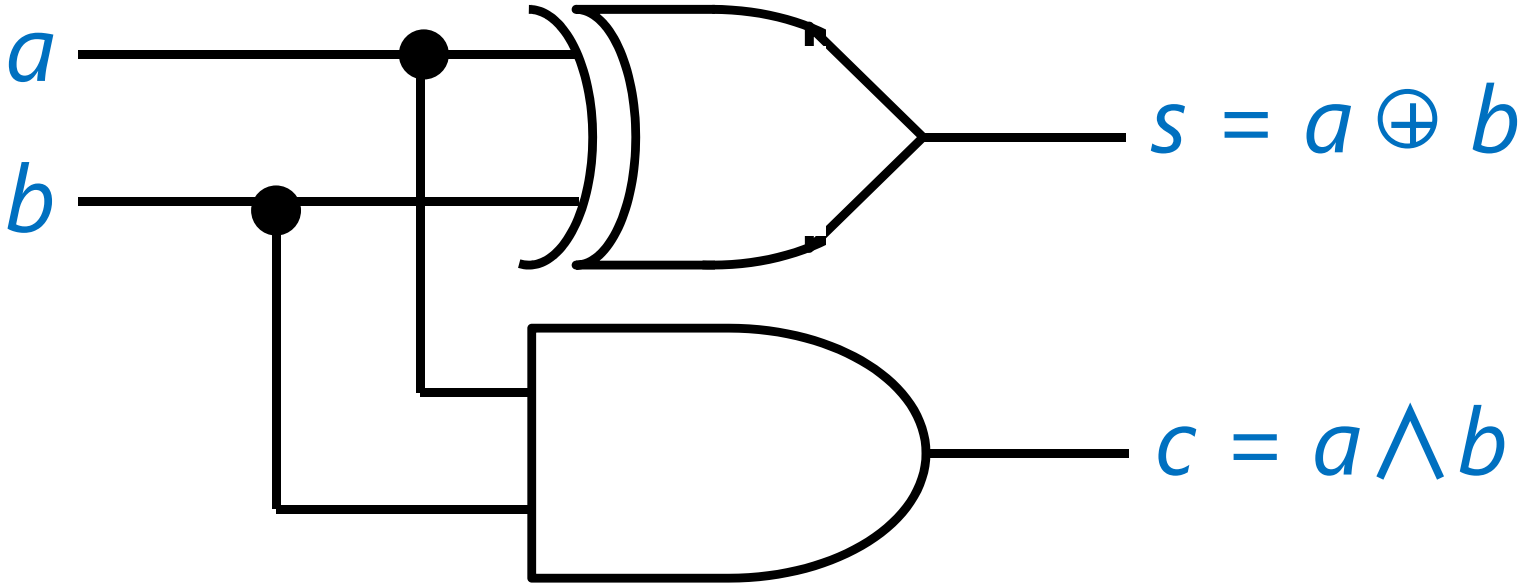








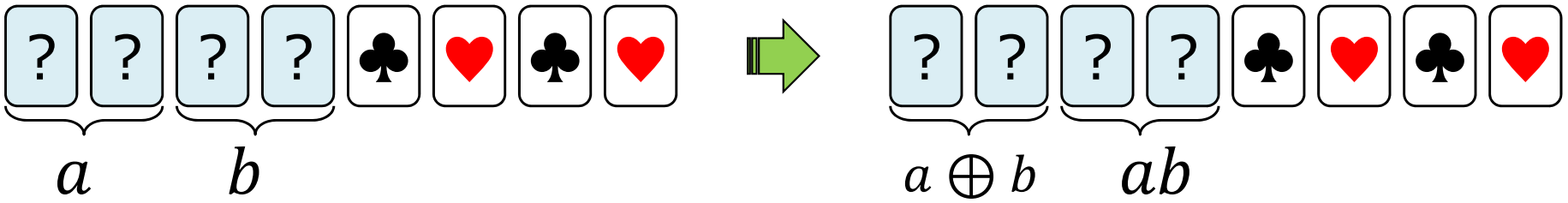
2.2 Improved Half-Adder Protocol



[6] T. Mizuki, I.K. Asiedu, and H. Sone, Voting with a Logarithmic Number of Cards, UCNC 2013, LNCS 7956, pp.162-173, 2013.

2.2 Improved Half-Adder Protocol

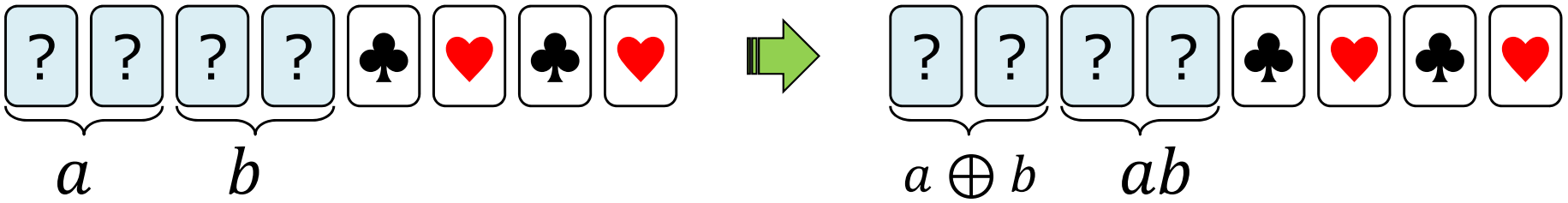
The existing half-adder protocol [6]



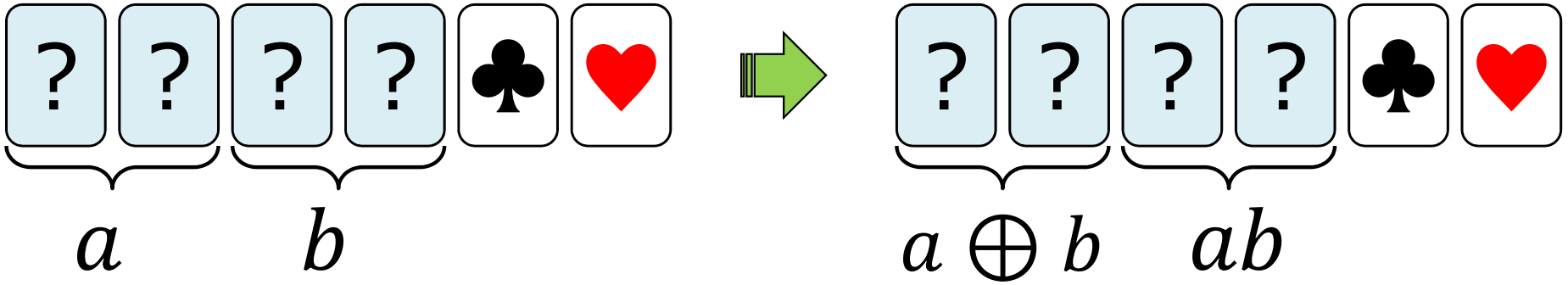
[6] T. Mizuki, I.K. Asiedu, and H. Sone, Voting with a Logarithmic Number of Cards, UCNC 2013, LNCS 7956, pp.162-173, 2013.

2.2 Improved Half-Adder Protocol

The existing half-adder protocol [6]

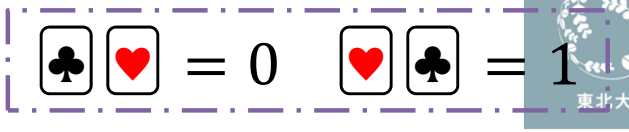


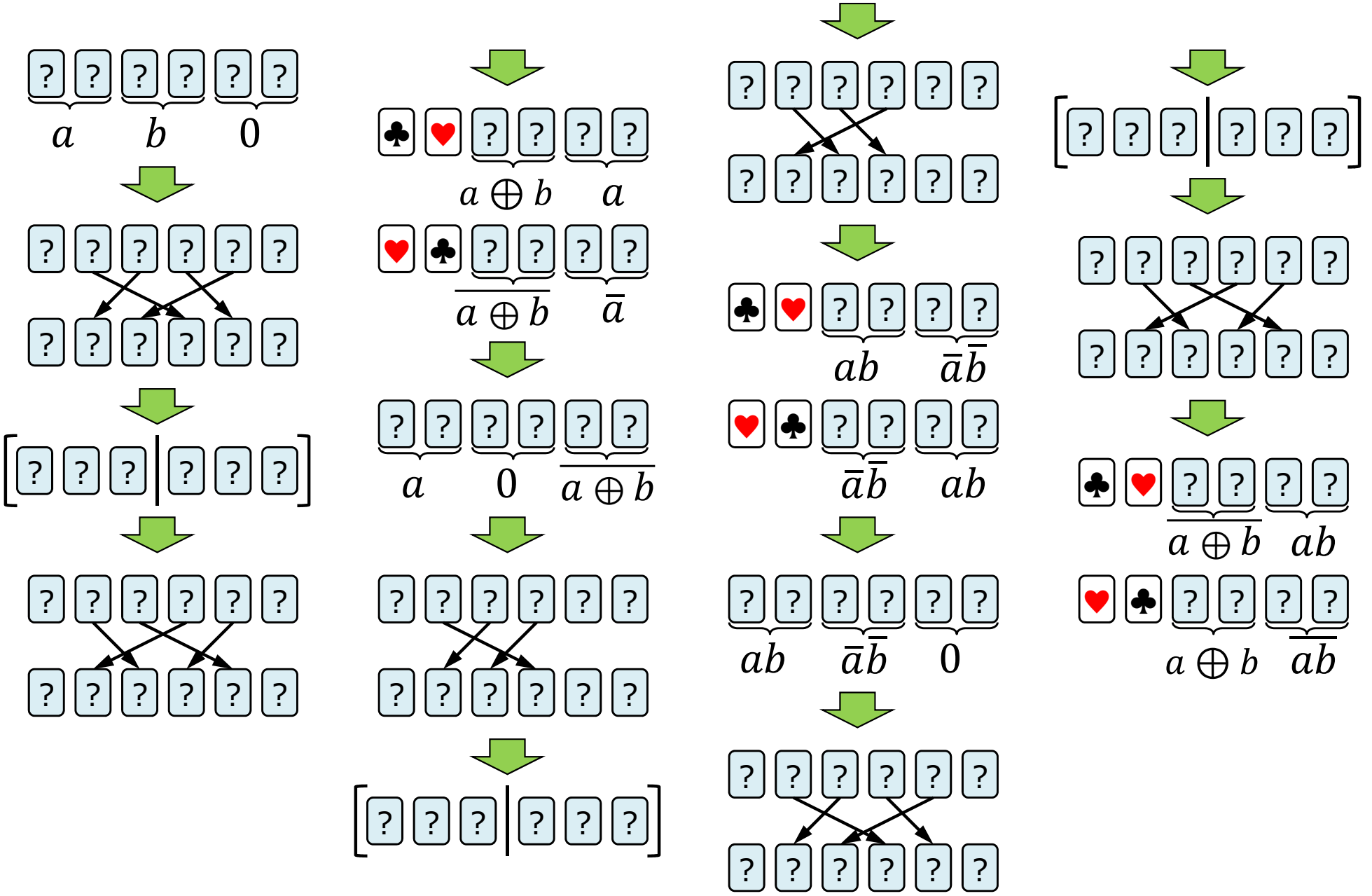
Our improved half-adder protocol



[6] T. Mizuki, I.K. Asiedu, and H. Sone, Voting with a Logarithmic Number of Cards, UCNC 2013, LNCS 7956, pp.162-173, 2013.

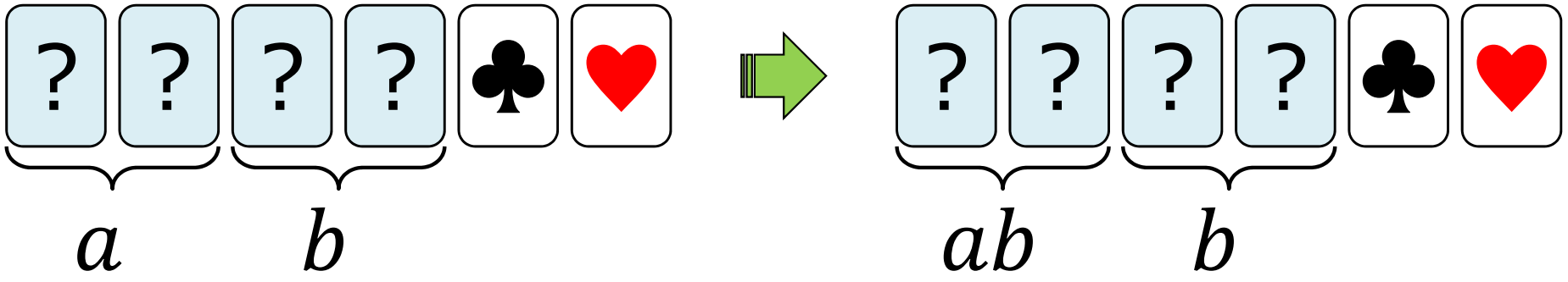
Full description of improved HA





Building Blocks

Improved AND:



Improved half-adder:

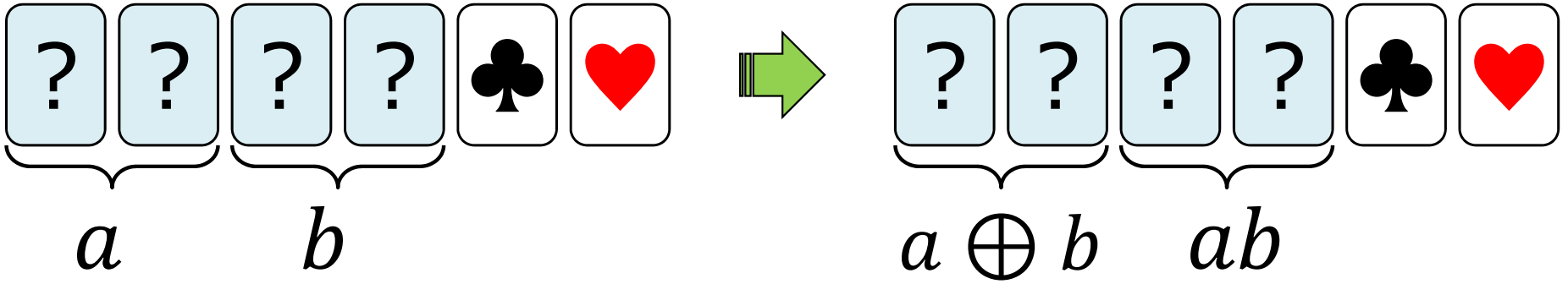


Table of Contents

AND-XOR expression: Shannon expansion

- Any n-variable function $f(x_1, x_2, \ldots, x_n)$ can be expressed as the Shannon expansion

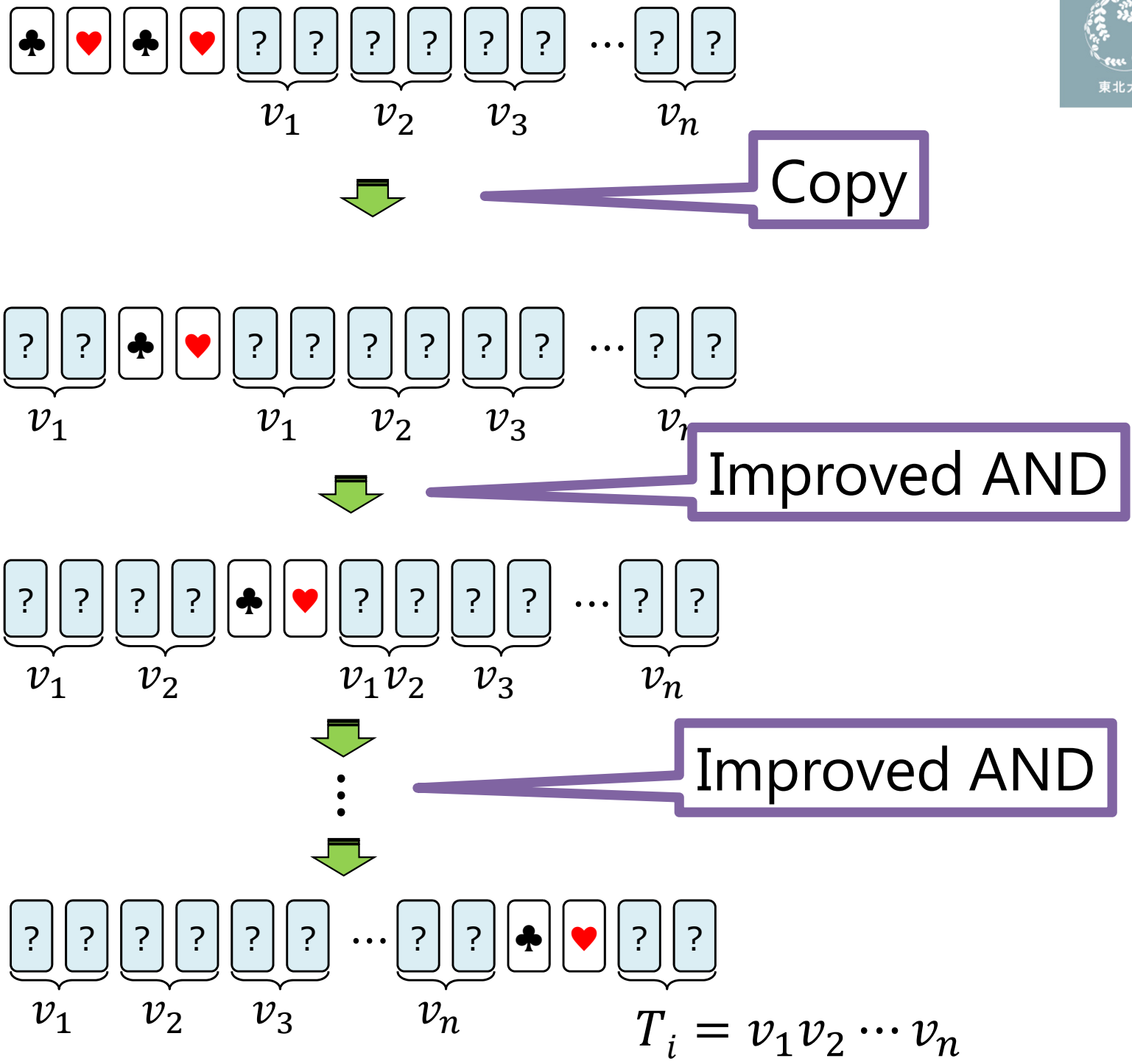
$$
\begin{aligned}
f(x_1, x_2, \ldots, x_n) = \; & \bar{x}_1 \bar{x}_2 \cdots \bar{x}_n f(0,0, \ldots, 0) \\
& \oplus x_1 \bar{x}_2 \cdots \bar{x}_n f(1,0, \ldots, 0) \\
& \oplus \bar{x}_1 x_2 \cdots \bar{x}_n f(0,1, \ldots, 0) \\
& \oplus x_1 x_2 \cdots \bar{x}_n f(1,1, \ldots, 0) \\
& \vdots \\
& \oplus x_1 x_2 \cdots x_n f(1,1, \ldots, 1)
\end{aligned}
$$

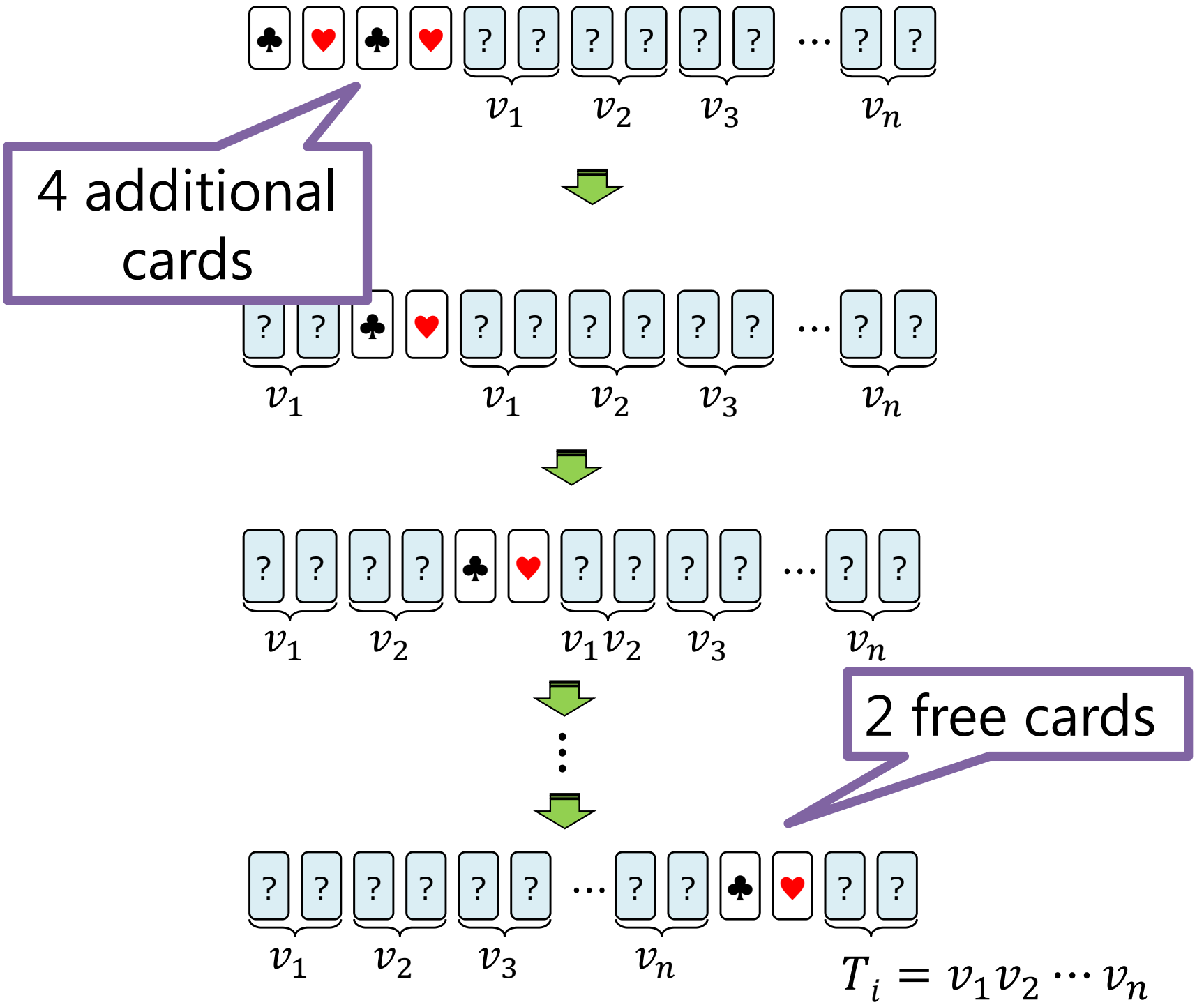
where a product term can be deleted if $f(*,*,\ldots,*)=0$.

$$f(x_1, x_2, \ldots, x_n) = T_1 \oplus T_2 \oplus \cdots \oplus T_i \oplus \cdots$$

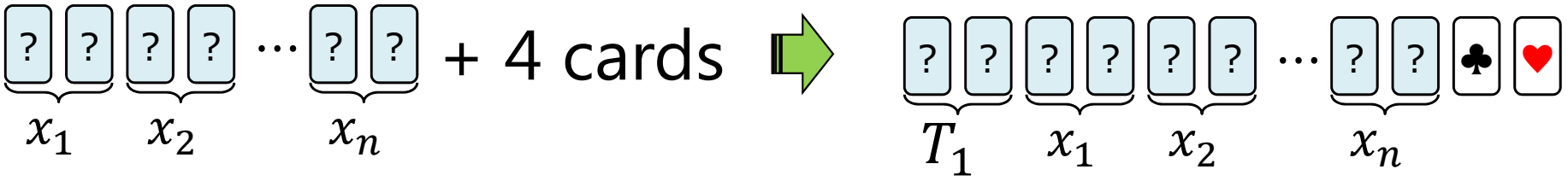
product term

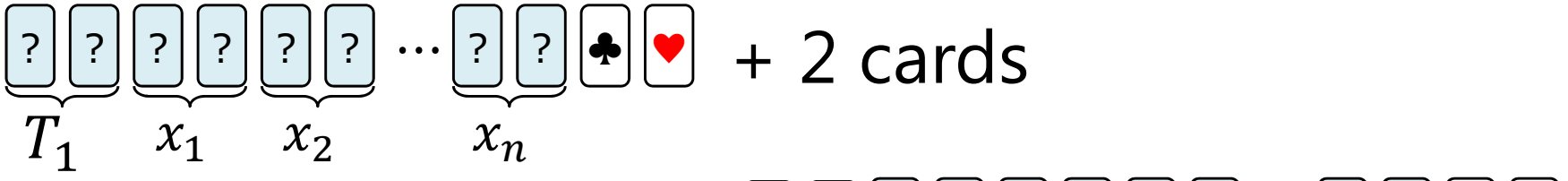
A product term $T_i = v_1\, v_2 \cdots v_n$ can be manipulated with 4 additional cards without losing original commitments, as follows:

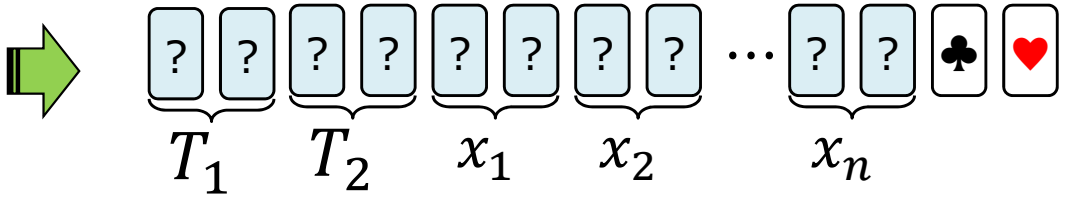

v_1 v_2 v_3 v_n
Copy
v_1 v_1 v_2 v_3 v_n
Improved AND
v_1 v_2 v_1v_2 v_3 v_n
Improved AND
v_1 v_2 v_3 v_n
$T_i = v_1v_2 \cdots v_n$

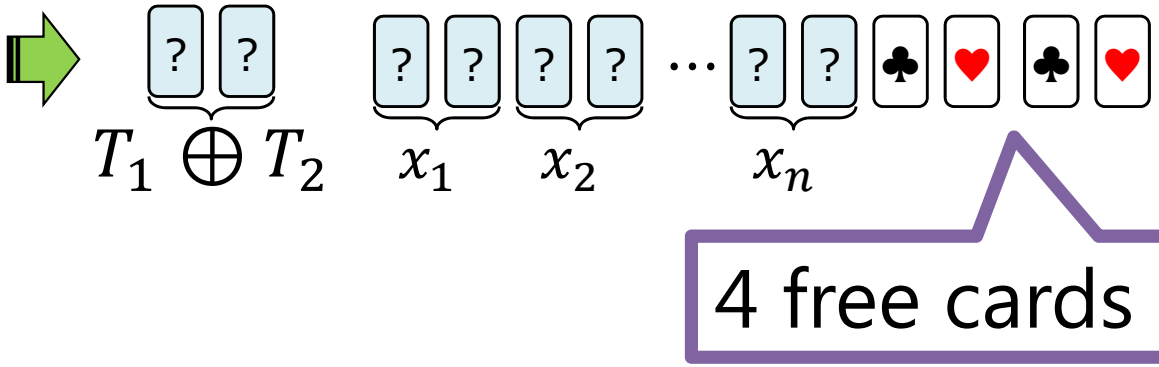

4 additional cards
2 free cards
v_1 v_2 v_3 v_n
v_1 v_1 v_2 v_3 v_n
v_1 v_2 v_1v_2 v_3 v_n
v_1 v_2 v_3 v_n
$T_i = v_1v_2 \cdots v_n$

$$f(x_1, x_2, \ldots, x_n) = T_1 \oplus T_2 \oplus \cdots \oplus T_i \oplus \cdots$$

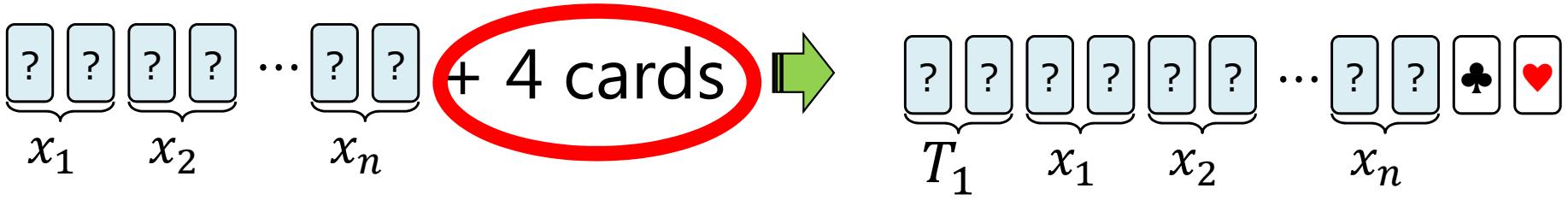


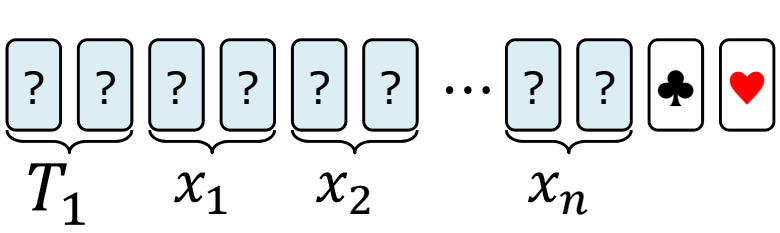


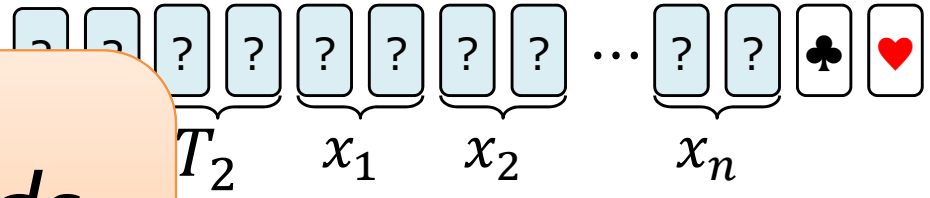




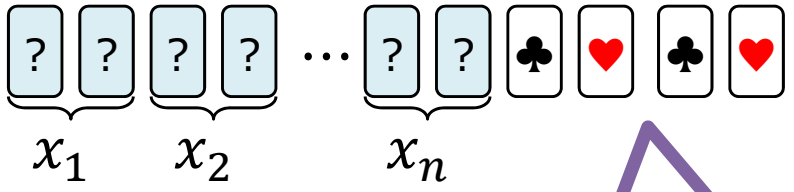
$$f(x_1, x_2, \ldots, x_n) = T_1 \oplus T_2 \oplus \cdots \oplus T_i \oplus \cdots$$







6 additional cards are sufficient



4 free cards

Theorem 6.

- Let f be an n-variable function.

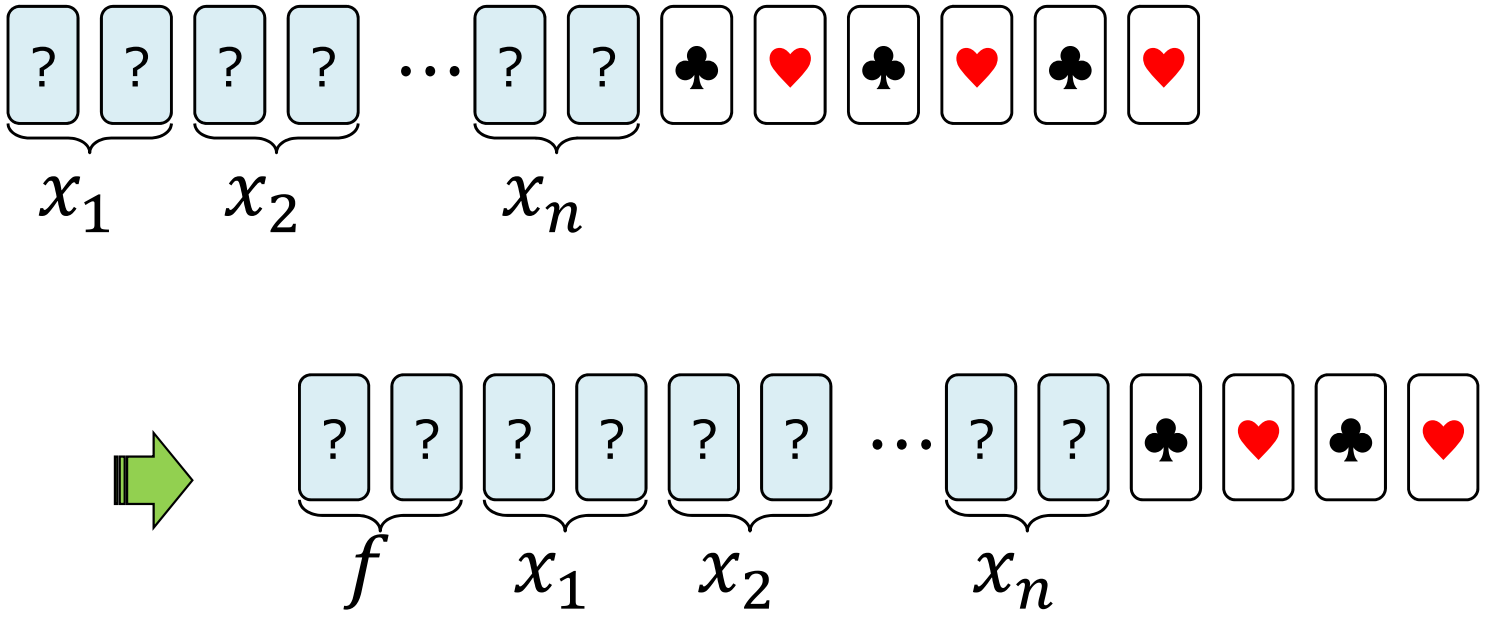
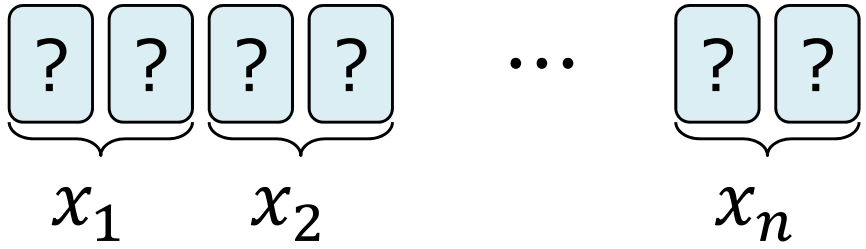


Table of Contents

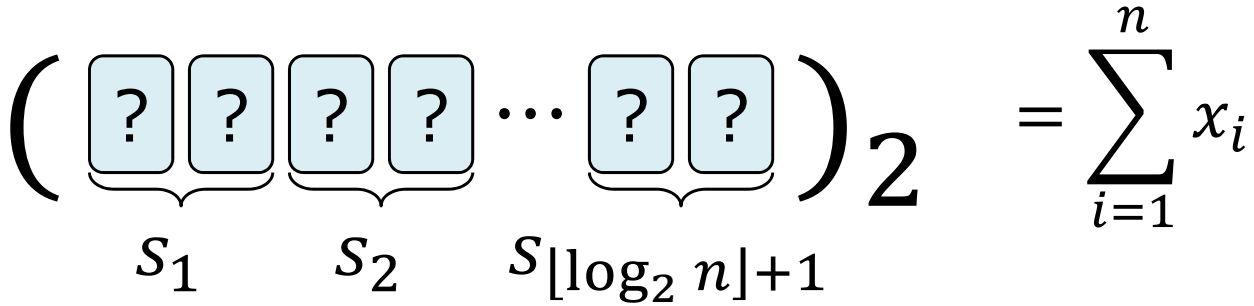
- Let $f(x_1, x_2, \ldots, x_n)$ be a symmetric funtion.
- The value of f depends on only the number of variables that take 1., namely $\sum x_i$.

- Let $f(x_1, x_2, \ldots, x_n)$ be a symmetric funtion.
- The value of f depends on only the number of variables that take 1., namely $\sum x_i$.

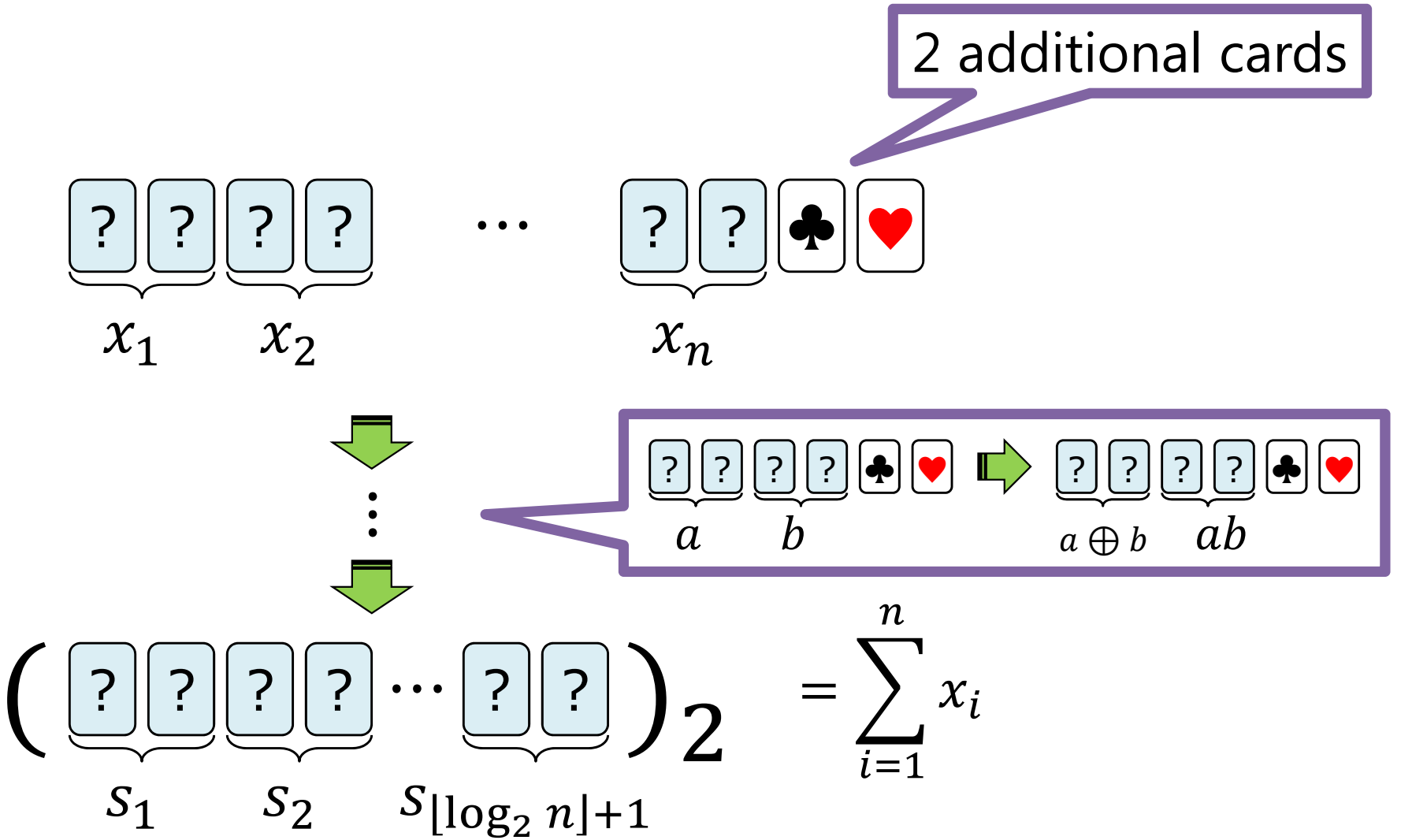
input commitments



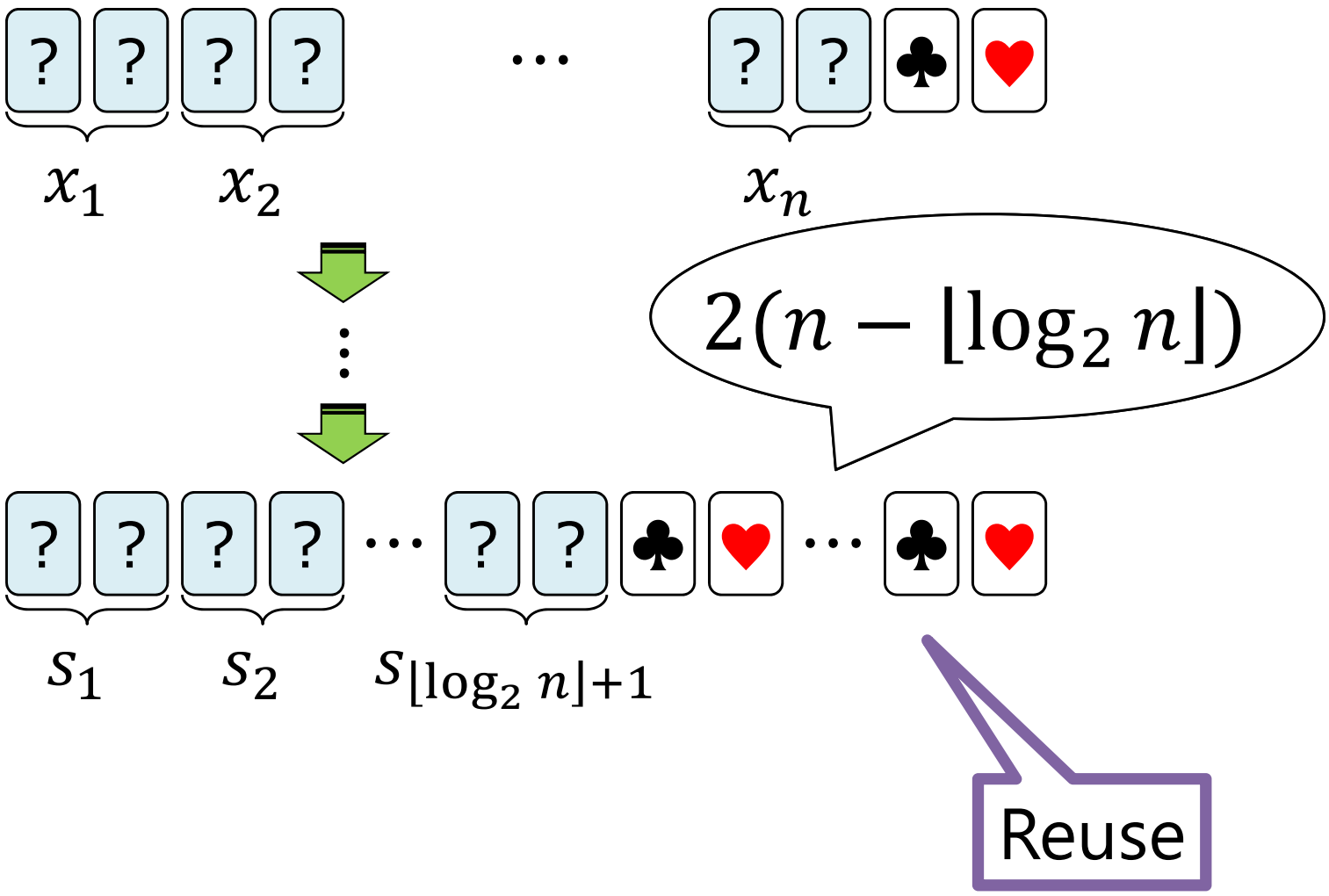
binary representation of $\sum x_i$



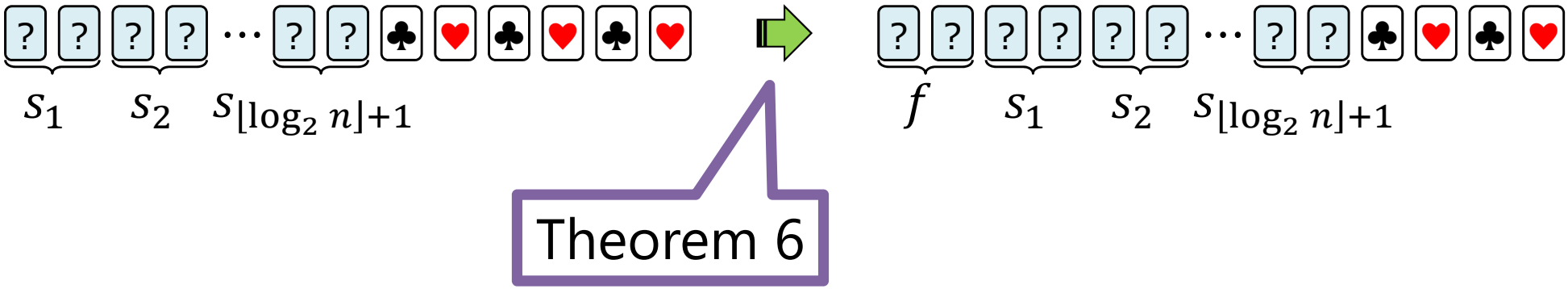
This can be done by the improved half-adder protocol with 2 additional cards.



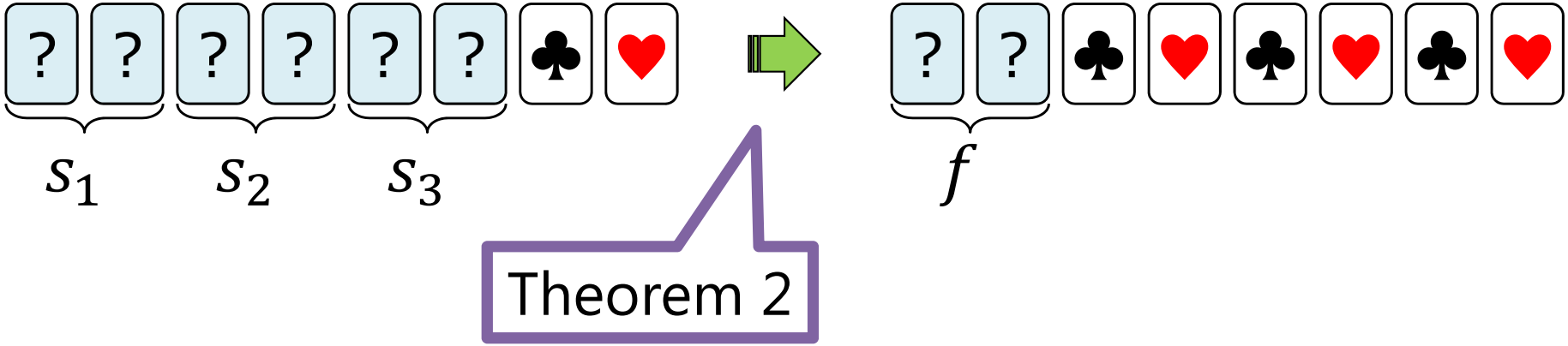
Note that some free cards will arise.



- If $n \geq 5$, then there are at least 6 free cards.



- If $n = 4$, then $\sum x_i$ is a 3-bit sequence.



Theorem 8.

- Let $n \geq 4$ and let f be an n-variable symmetric function.

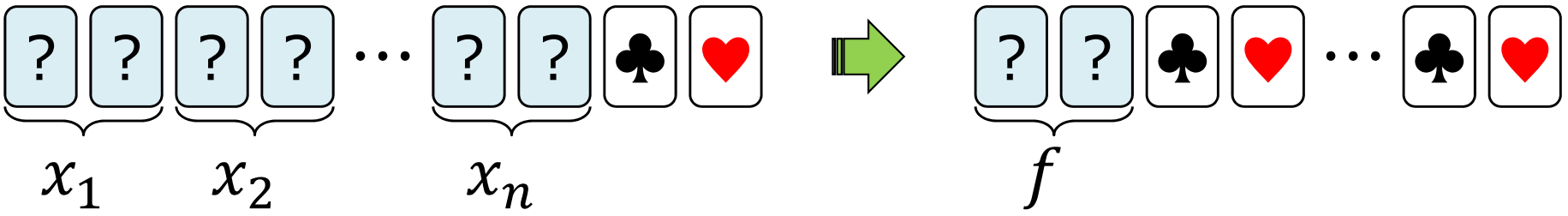


Table of Contents

Conclusion

- 6 additional cards are sufficient for any function
- 2 additional cards suffice for any symmetric one

A (real) deck of cards available to the first several people; please contact the speaker.