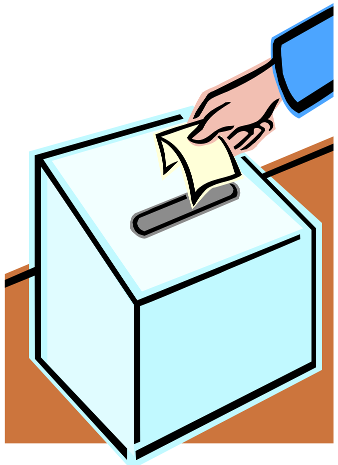
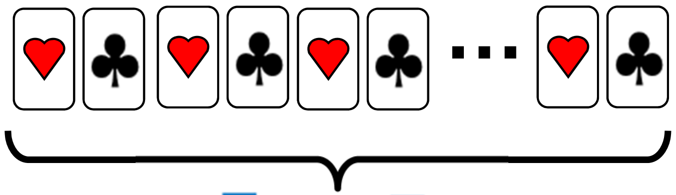
UCNC 2013 (Jul. 4, 2013, Milano)





$2\lceil \log n \rceil + 6$

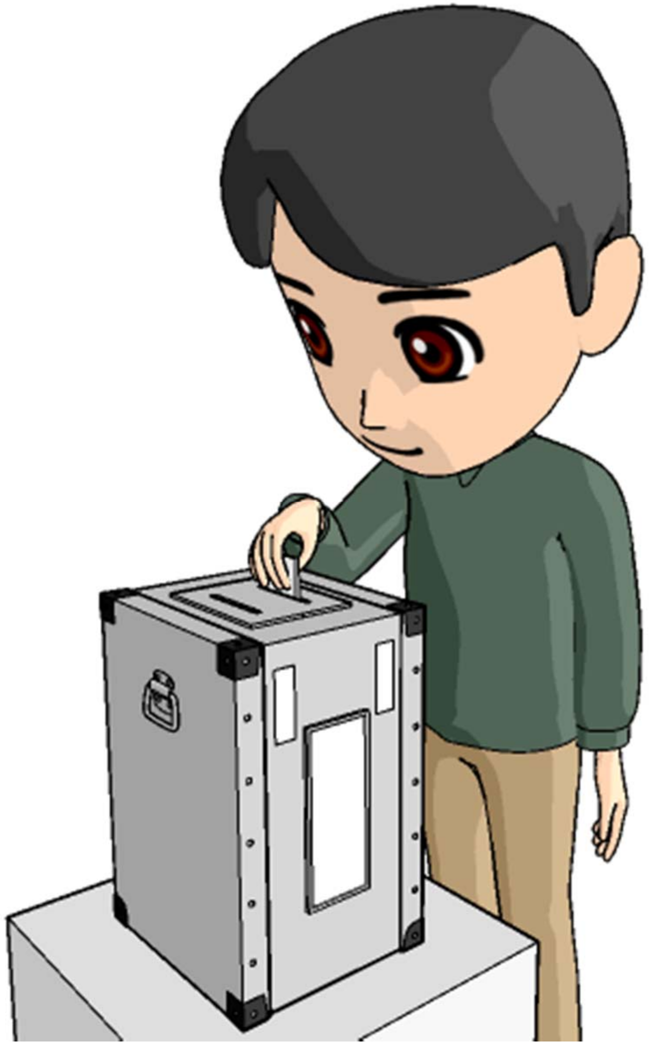
n voters

Voting with a Logarithmic Number of Cards

Takaaki Mizuki, Isaac Kobina Asiedu, Hideaki Sone

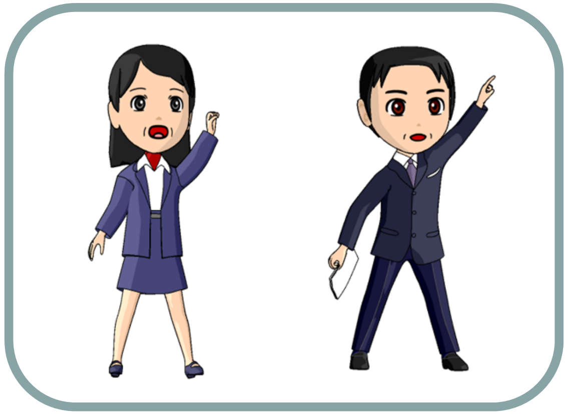
Tohoku University

Abstract

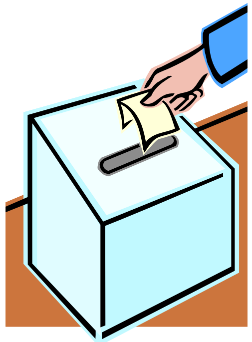


Abstract

- ✓ There are 2 candidates and n voters.



2 candidates

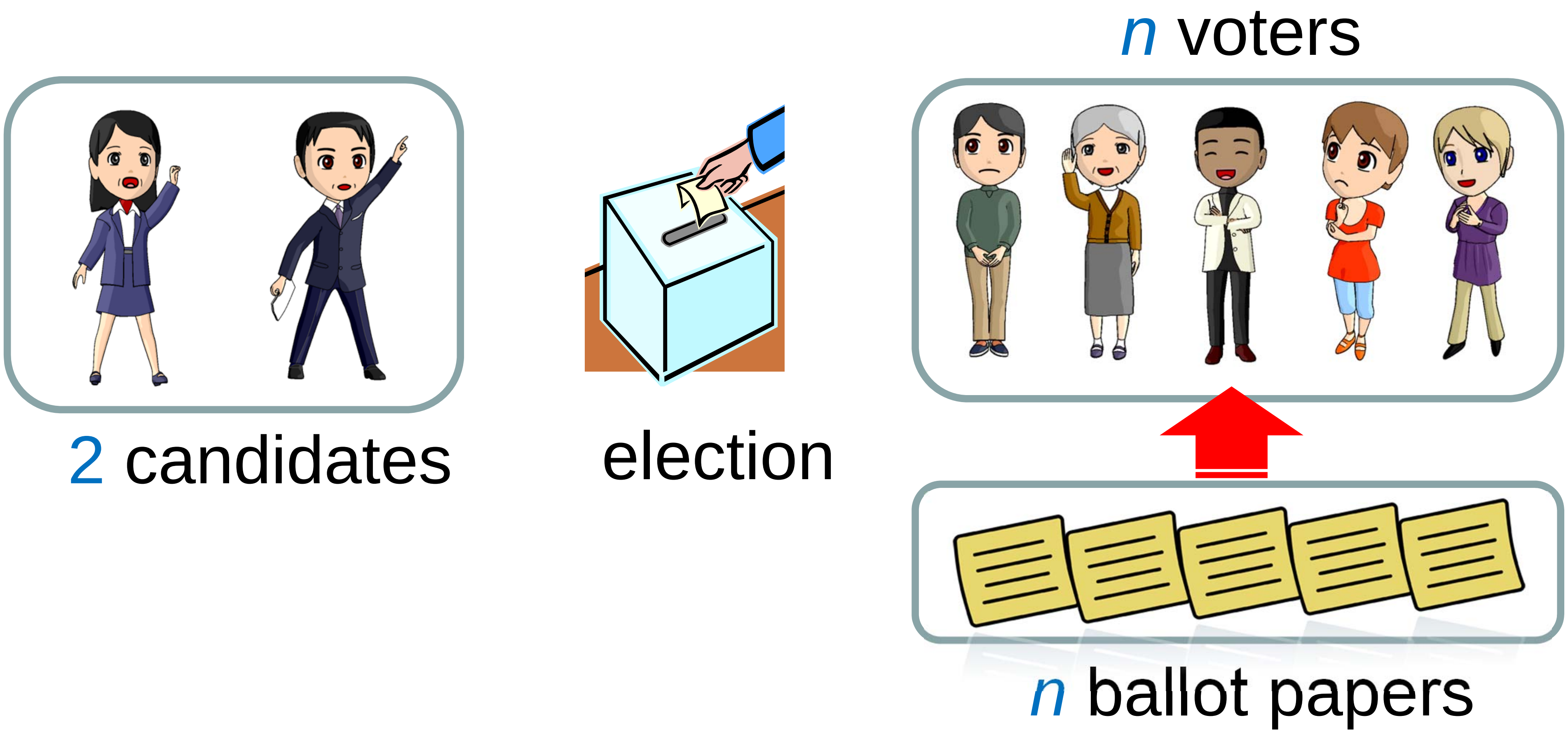


election

n voters

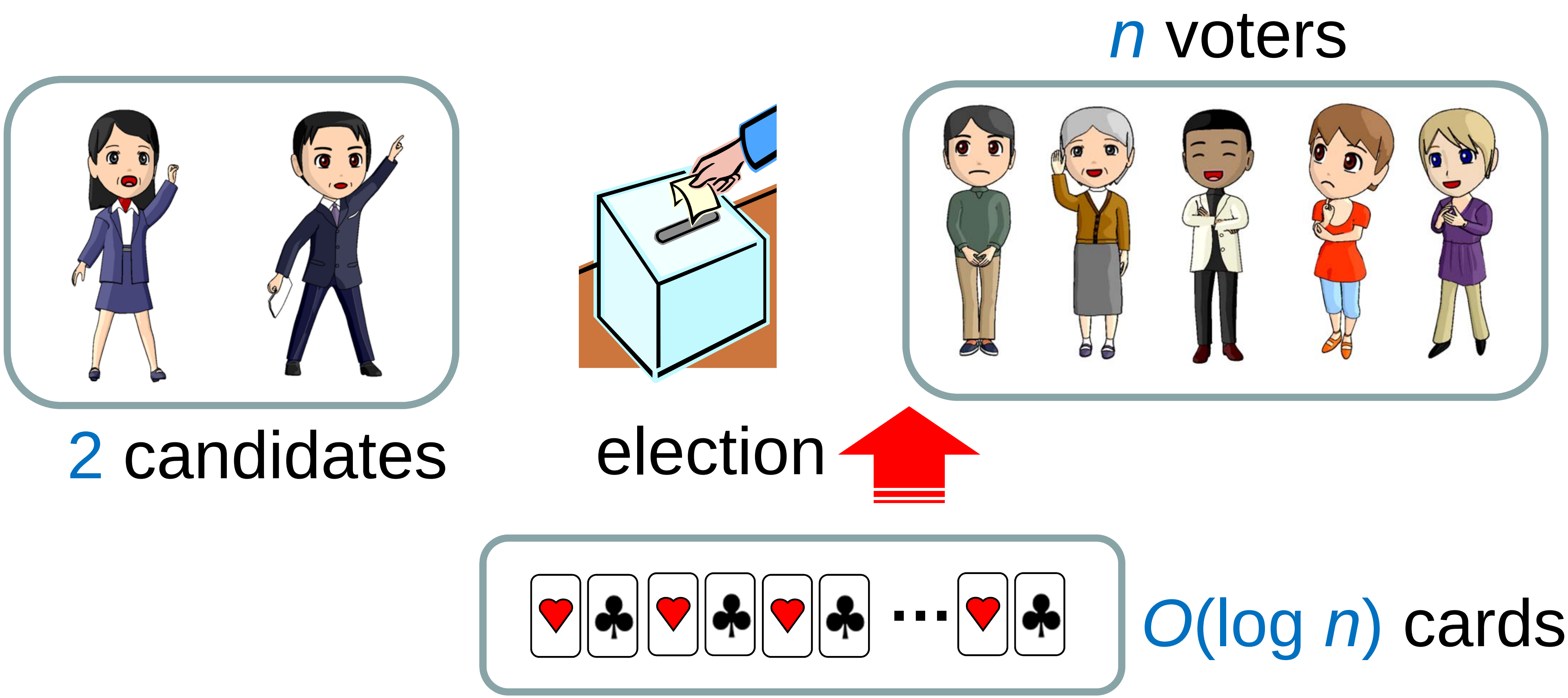
Abstract

- ✓ There are 2 candidates and n voters.
- ✓ Usually, n ballot papers are required.



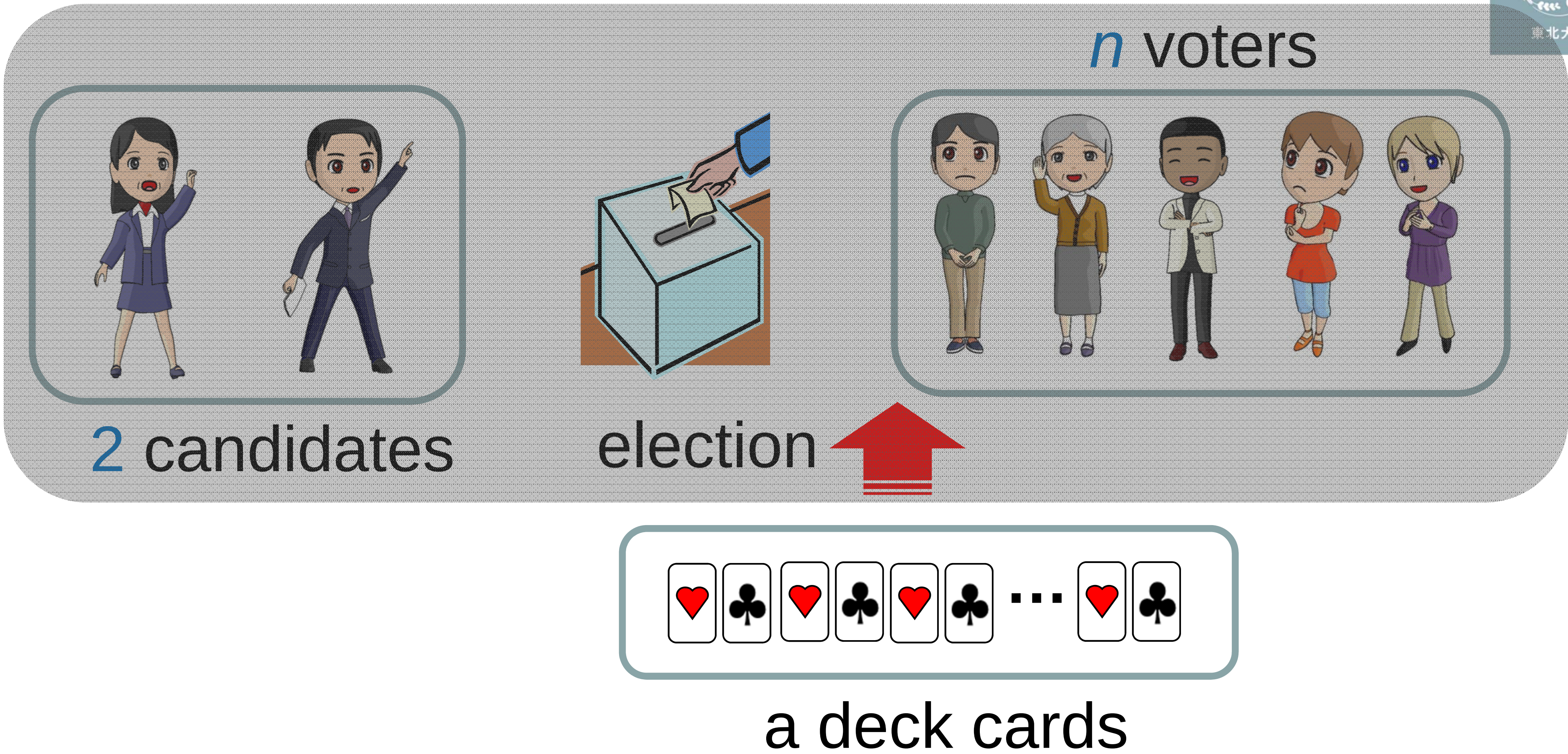
Abstract

- ✓ There are 2 candidates and n voters.
- ✓ Usually, n ballot papers are required.
- ✓ We show $O(\log n)$ cards conduct an election.

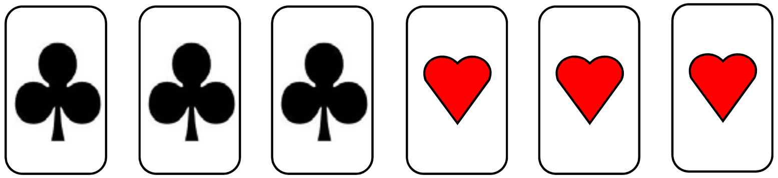


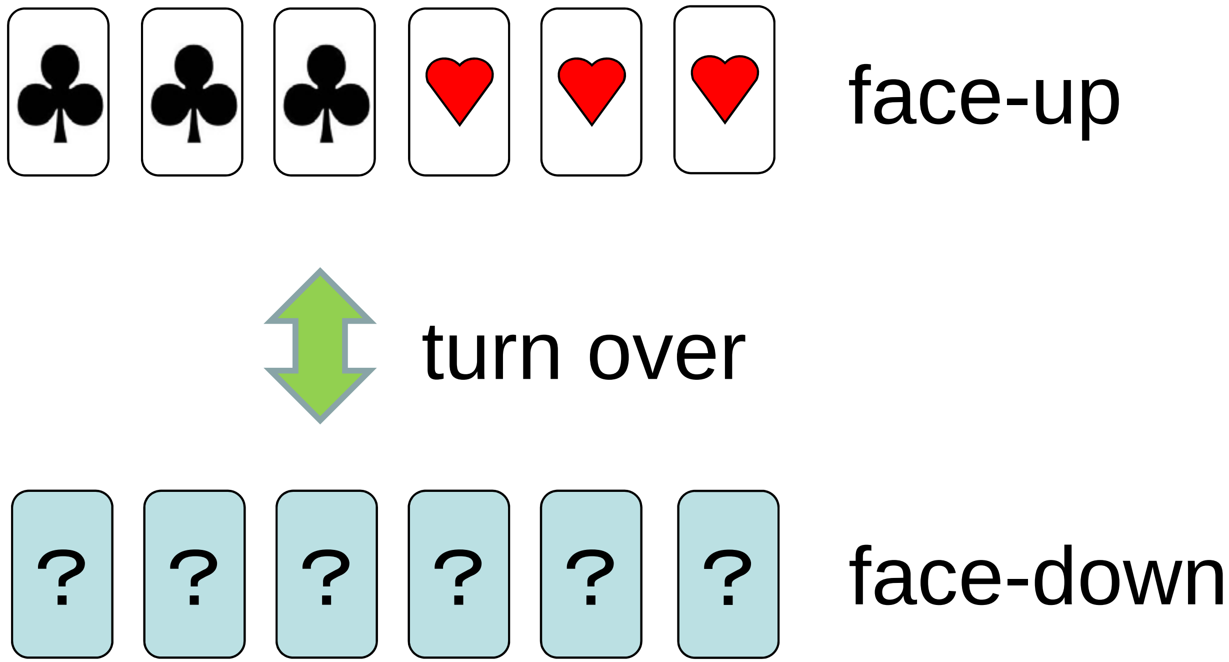
Contents

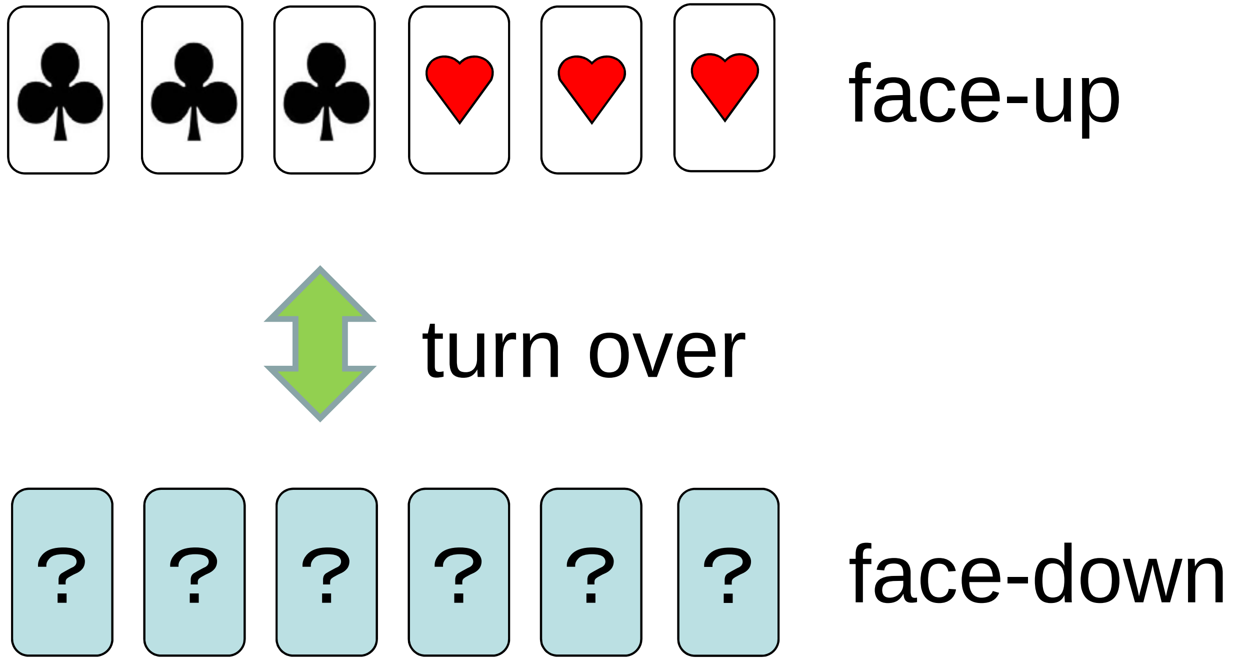
Contents



In this paper, we use a deck of ***cards***.

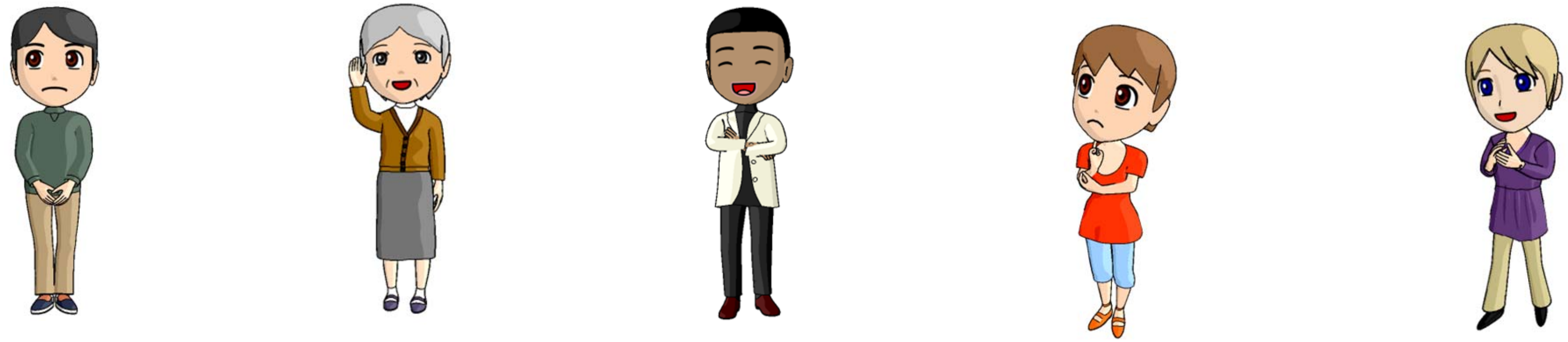



face-up
turn over
face-down



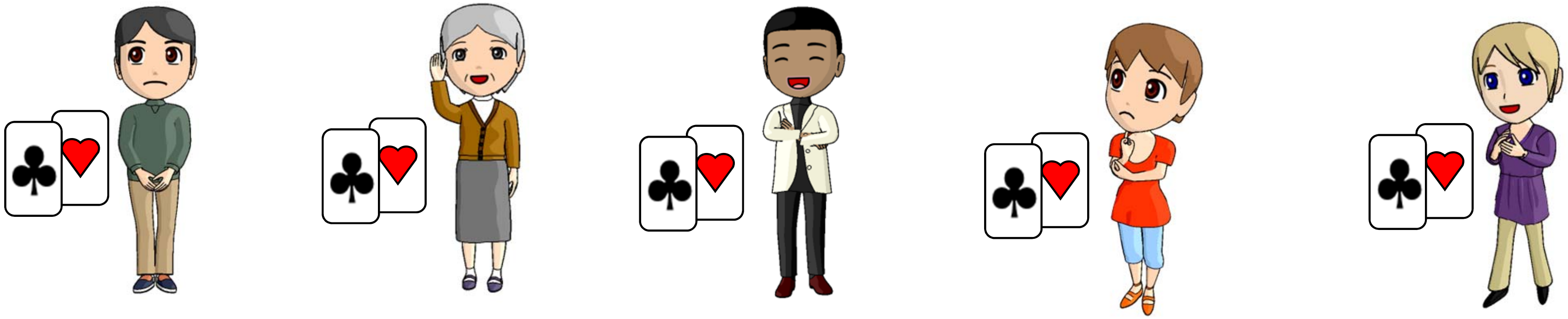
How to implement voting?
The simplest way is as follows.

1. Distribute two cards of different suits to each voter.



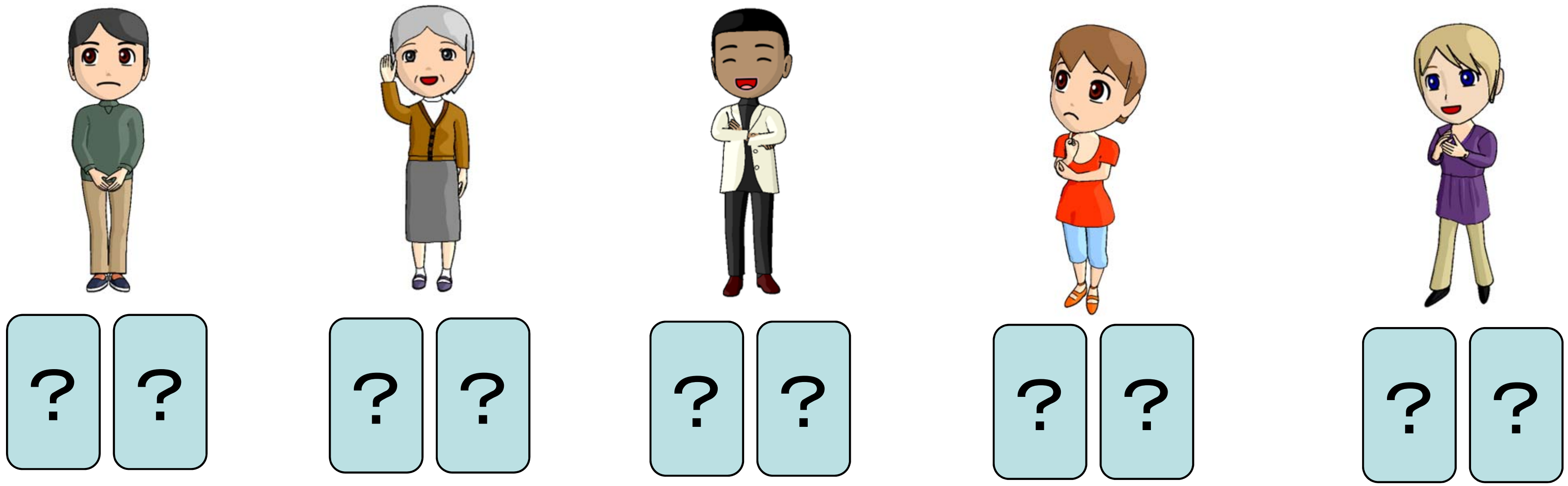
n voters

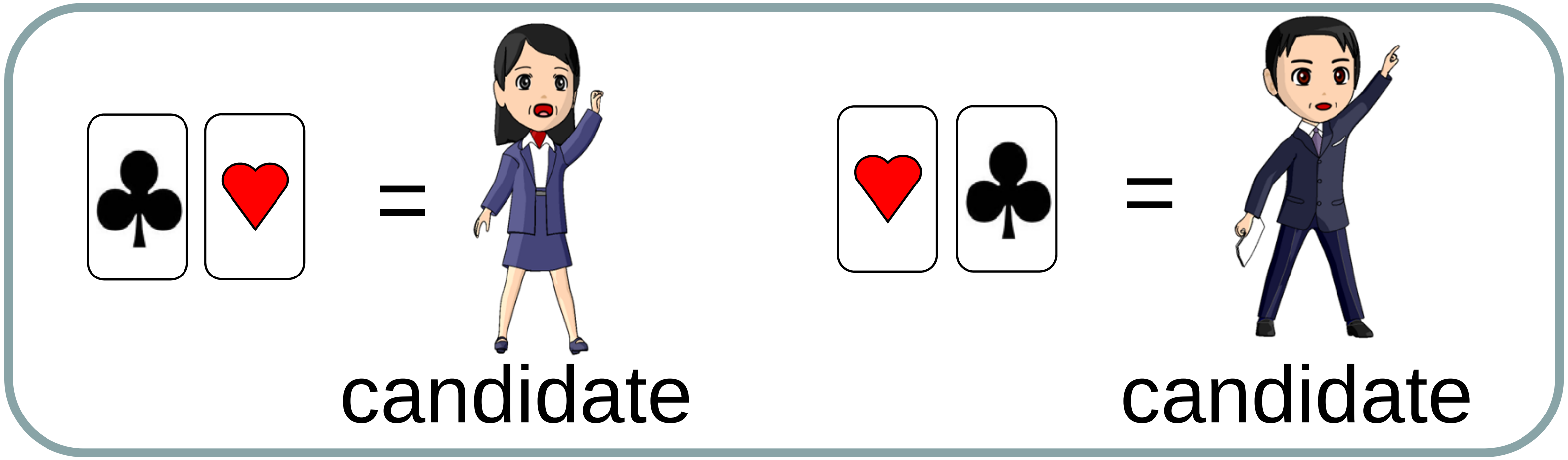
1. Distribute two cards of different suits to each voter.



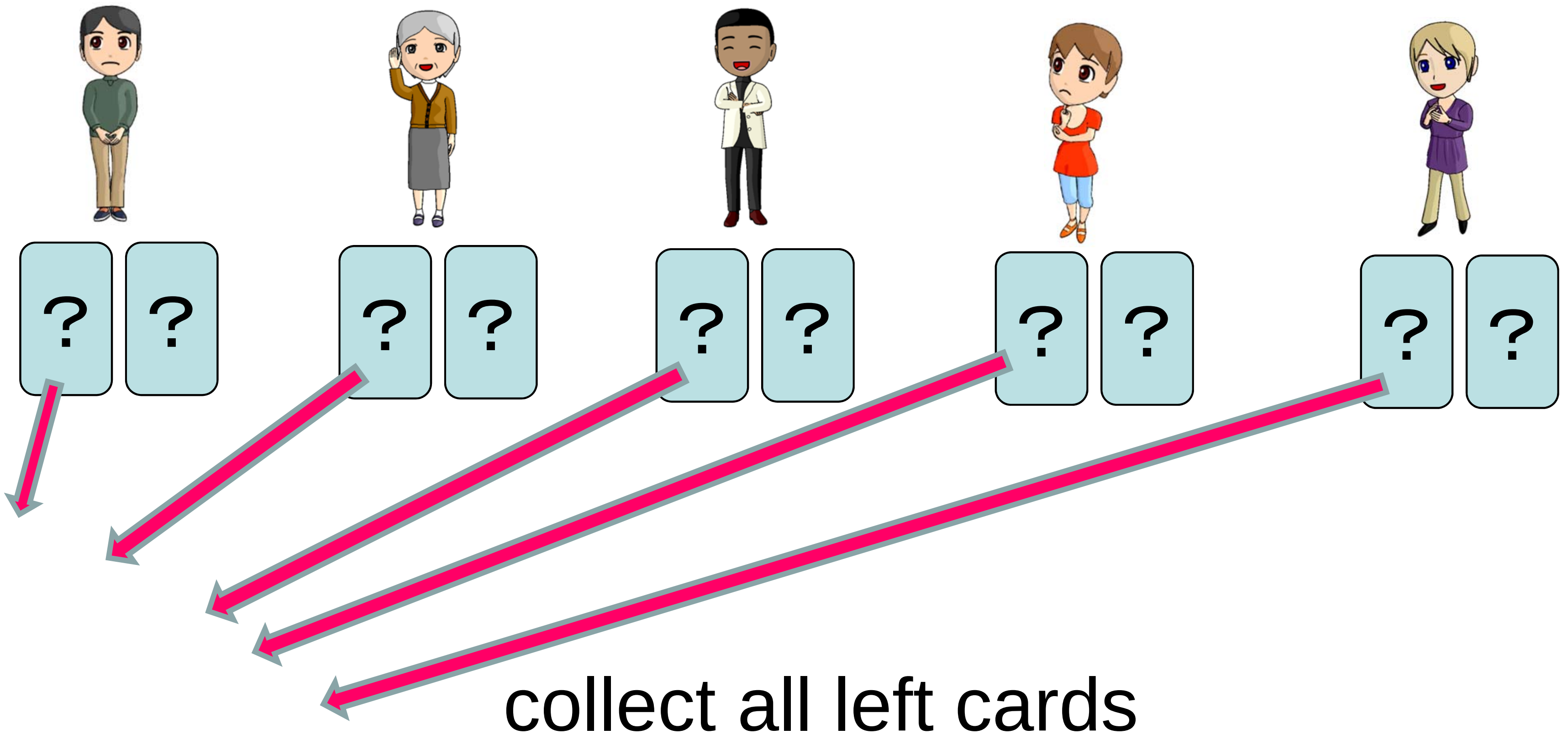
n voters

1. Distribute two cards of different suits to each voter.
2. Each voter privately commits his/her ballot according to the encoding.

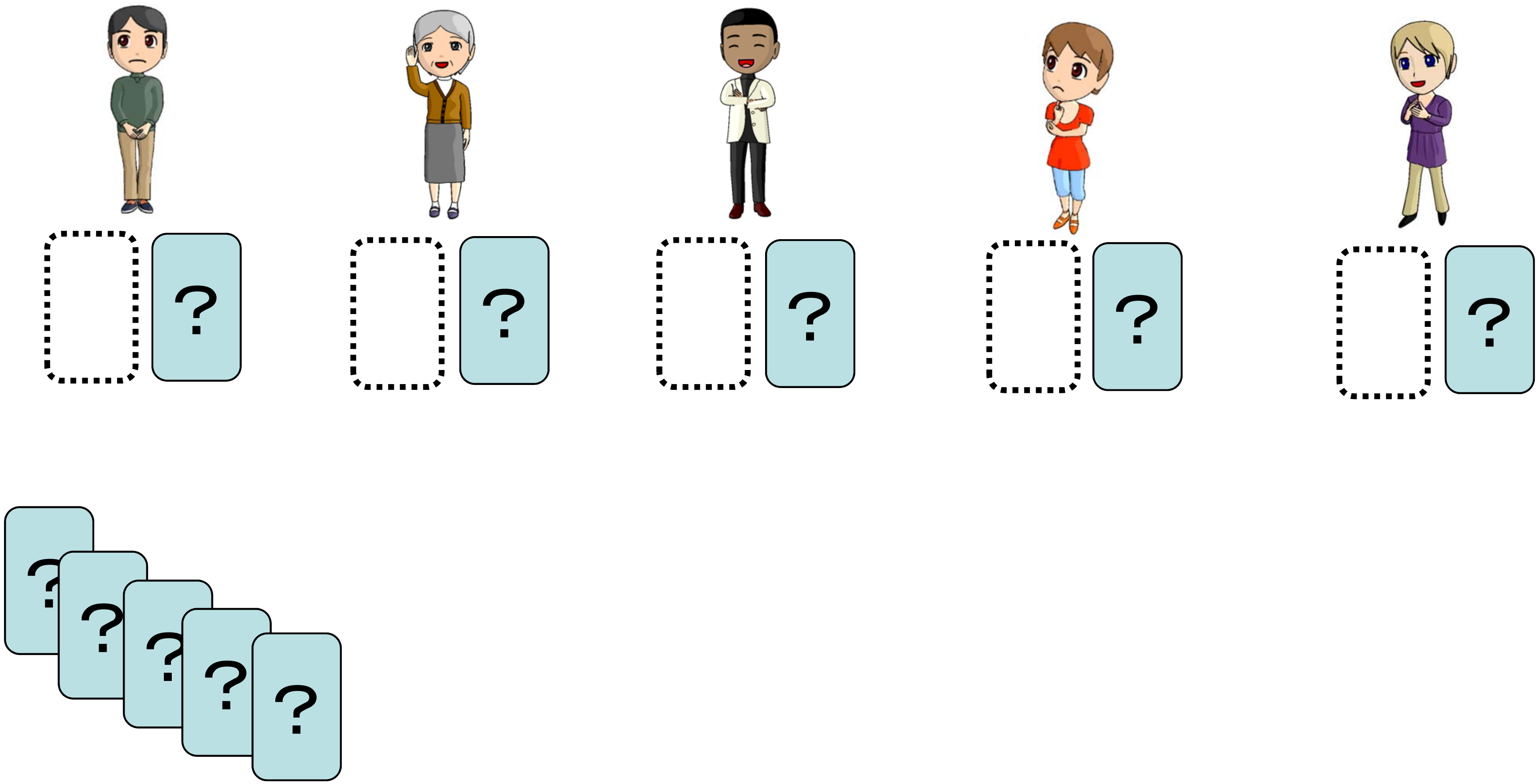




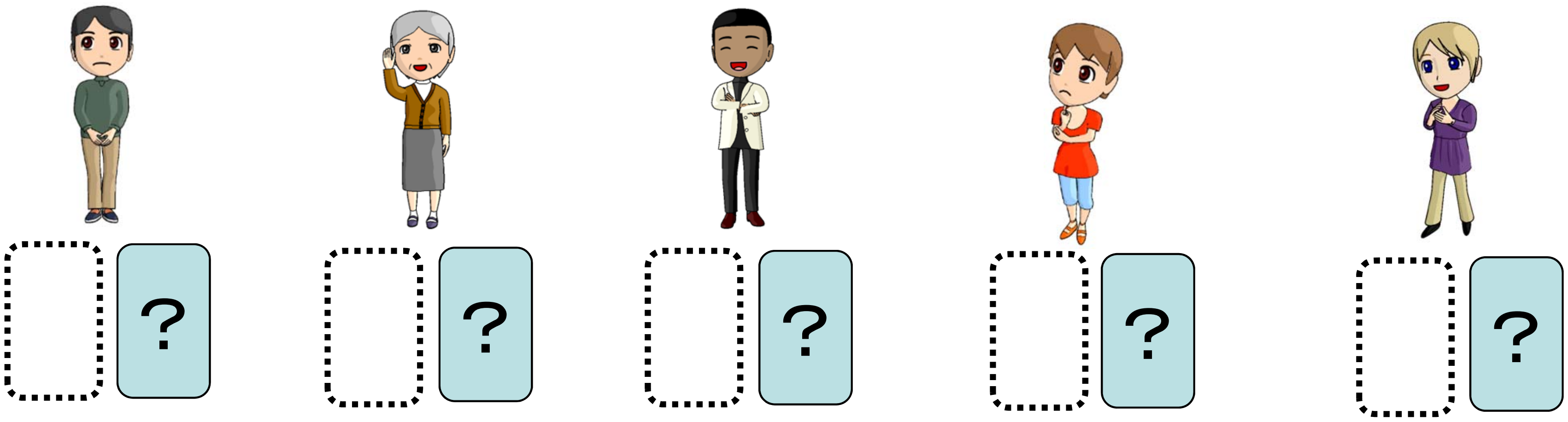
1. Distribute two cards of different suits to each voter.
2. Each voter privately commits his/her ballot.
3. Shuffle all left cards and reveal them.

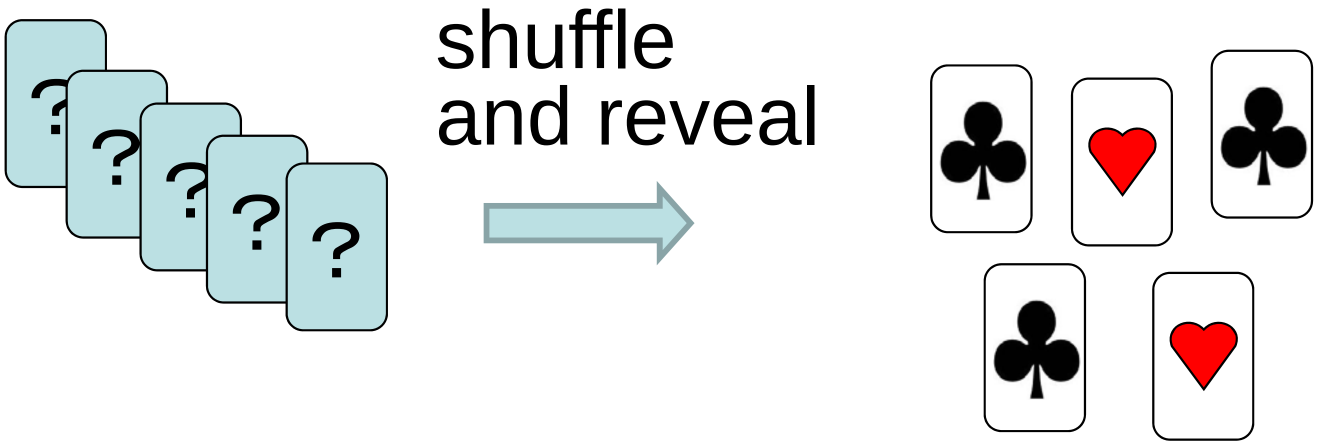


1. Distribute two cards of different suits to each voter.
2. Each voter privately commits his/her.
3. Shuffle all left cards and reveal them.

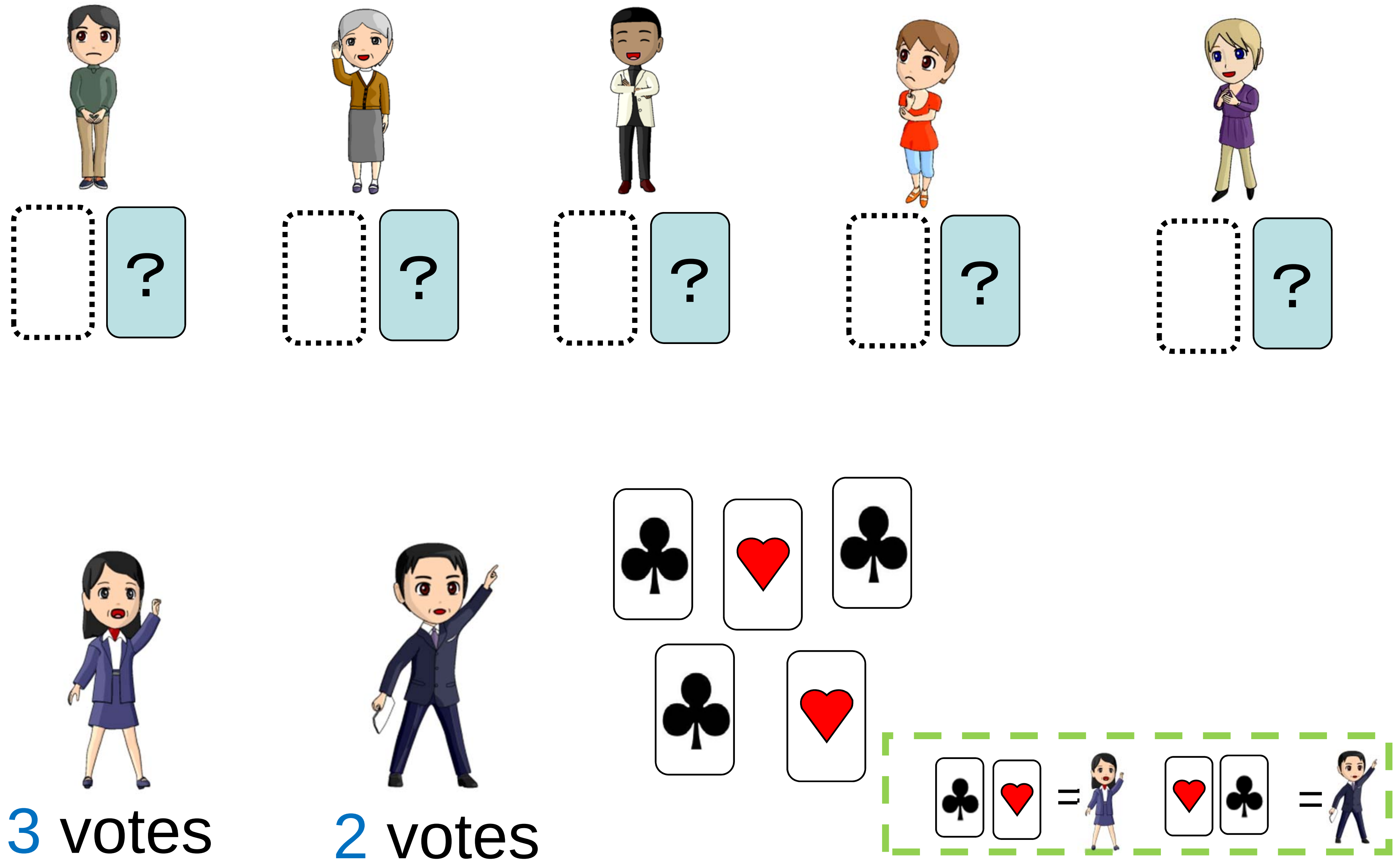


1. Distribute two cards of different suits to each voter.
2. Each voter privately commits his/her.
3. Shuffle all left cards and reveal them.

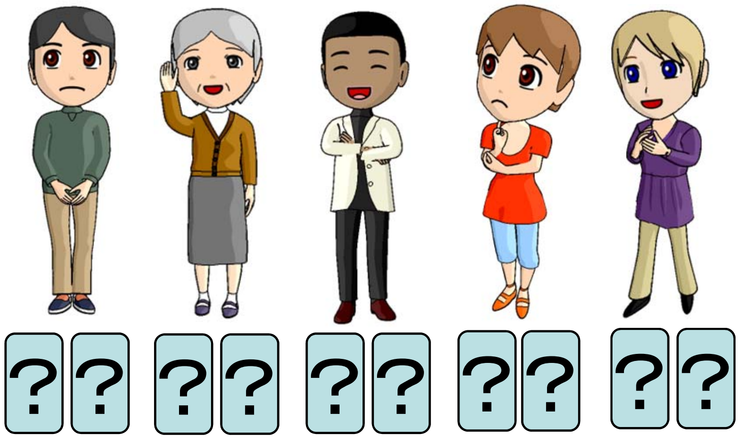




1. Distribute two cards of different suits to each voter.
2. Each voter privately commits his/her.
3. Shuffle all left cards and reveal them.



n voters



$2n$ cards

✓ Voting can be naively done using $2n$ cards.

n voters

- ✓ Voting can be naively done using $2n$ cards.
- ✓ This paper shows that, by ***applying card-based cryptographic protocols***, $O(\log n)$ cards can also conduct voting.

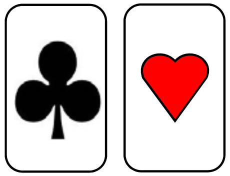
Notations and the history

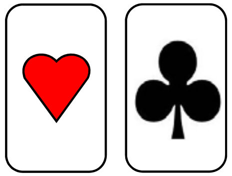
Card-based protocols provide secure computation.

Notations and the history

Card-based protocols provide secure computation.

To deal with Boolean values, this encoding is used:

 = 0

 = 1

Notations and the history

To deal with Boolean values, this encoding is used:

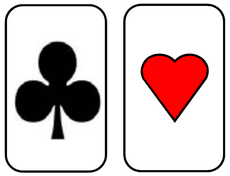
♣ ♥ = 0   ♥ ♣ = 1

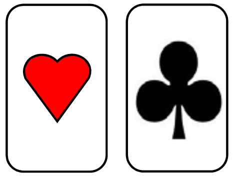
commitment

A ***commitment*** to a bit $x \in \{0,1\}$ is a pair ? ? of two face-down cards holding the value of x.

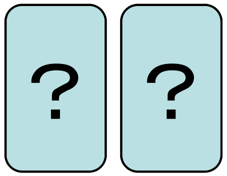
Notations and the history

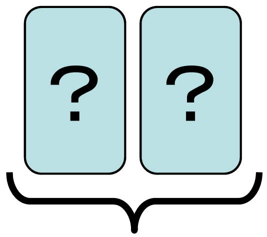
To deal with Boolean values, this encoding is used:





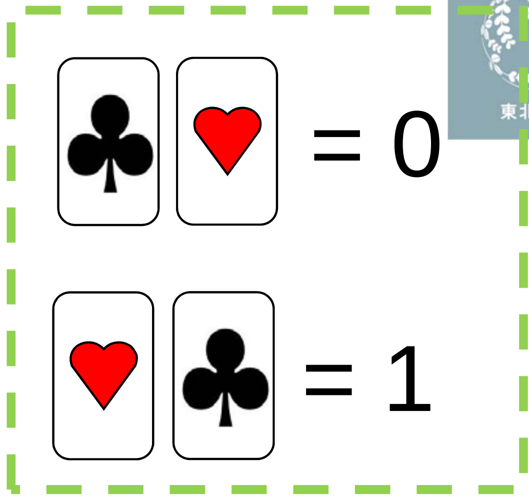
commitment

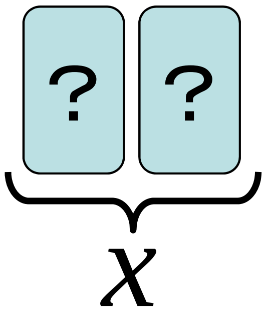
A ***commitment*** to a bit $x \in \{0,1\}$ is a pair  of two face-down cards holding the value of x.



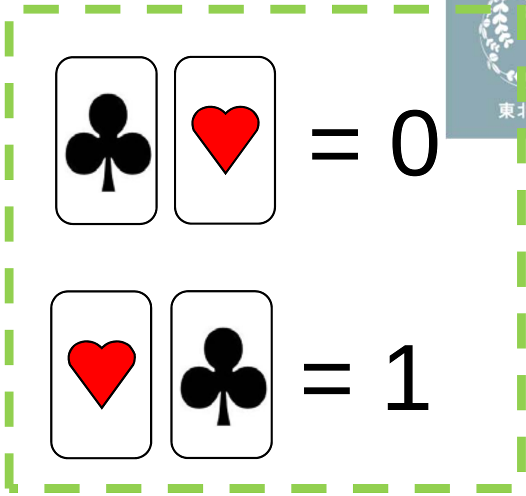
Notations and the history

Example of secure computation



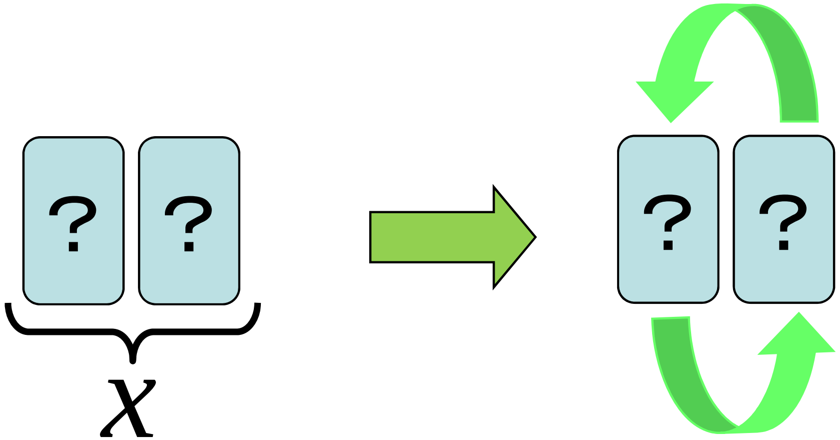


Notations and the history



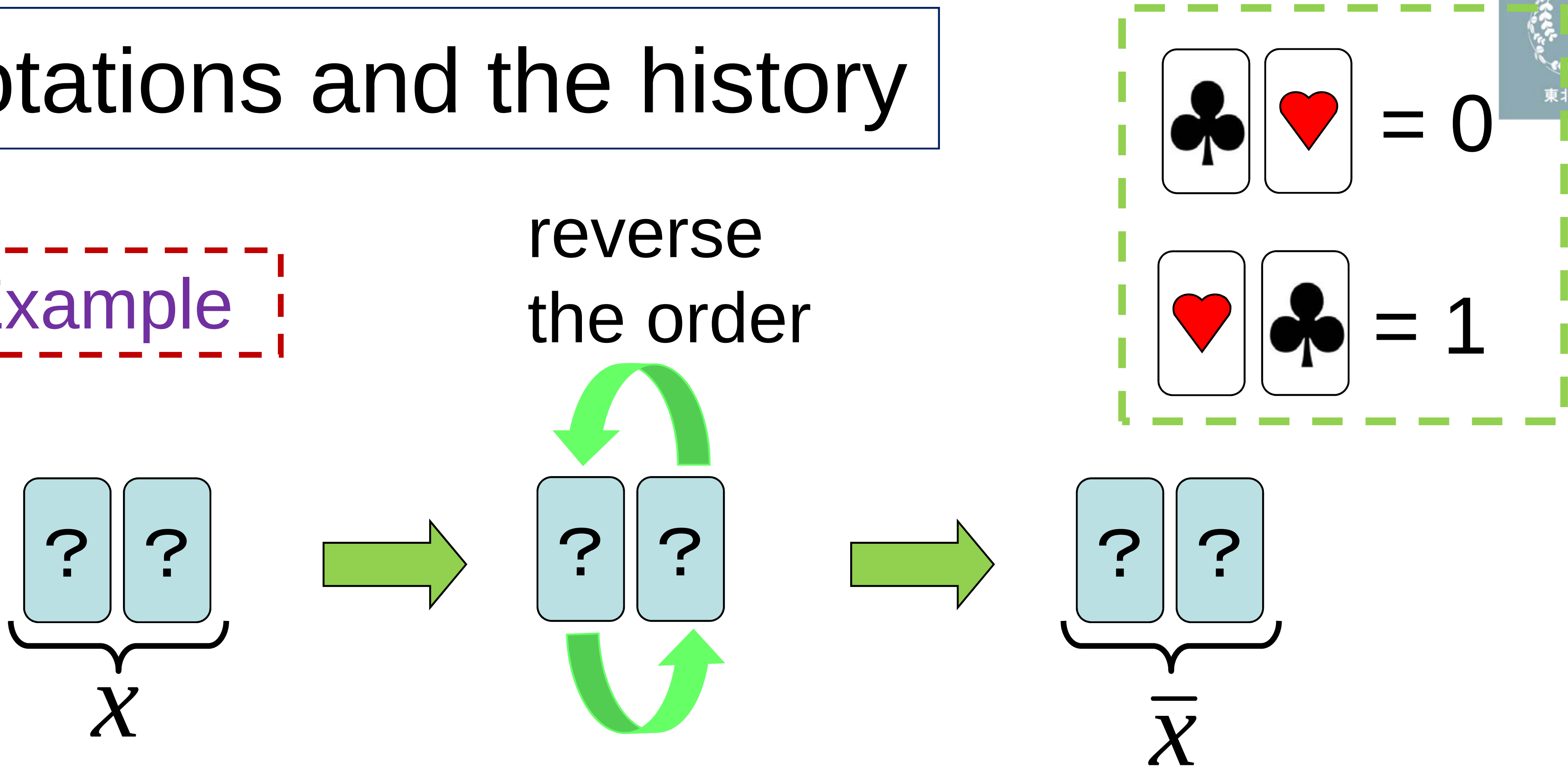
Example

reverse
the order

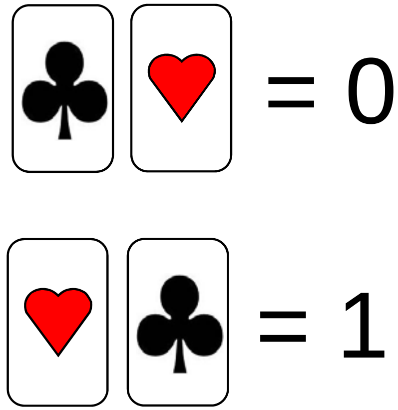


Notations and the history

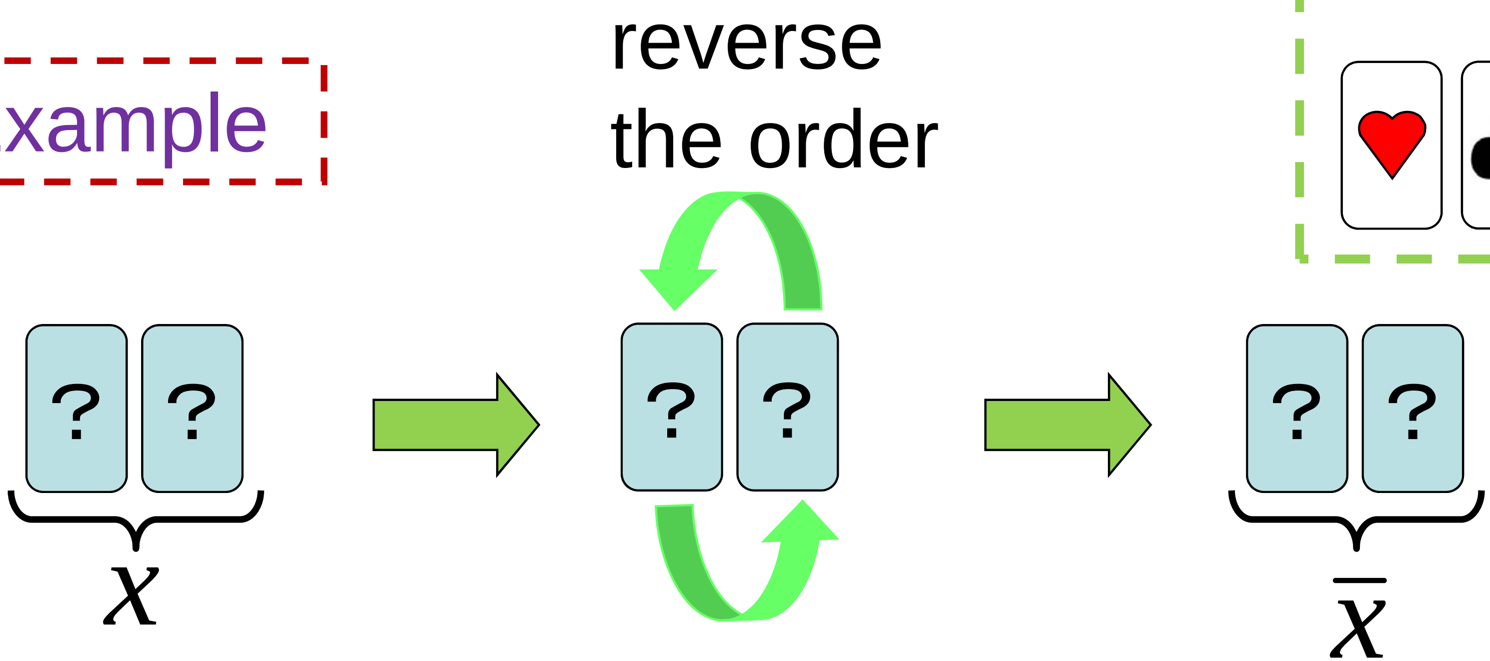
Example



Notations and the history

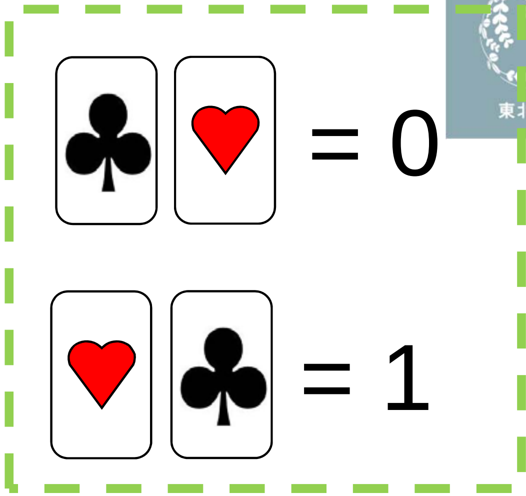


Example

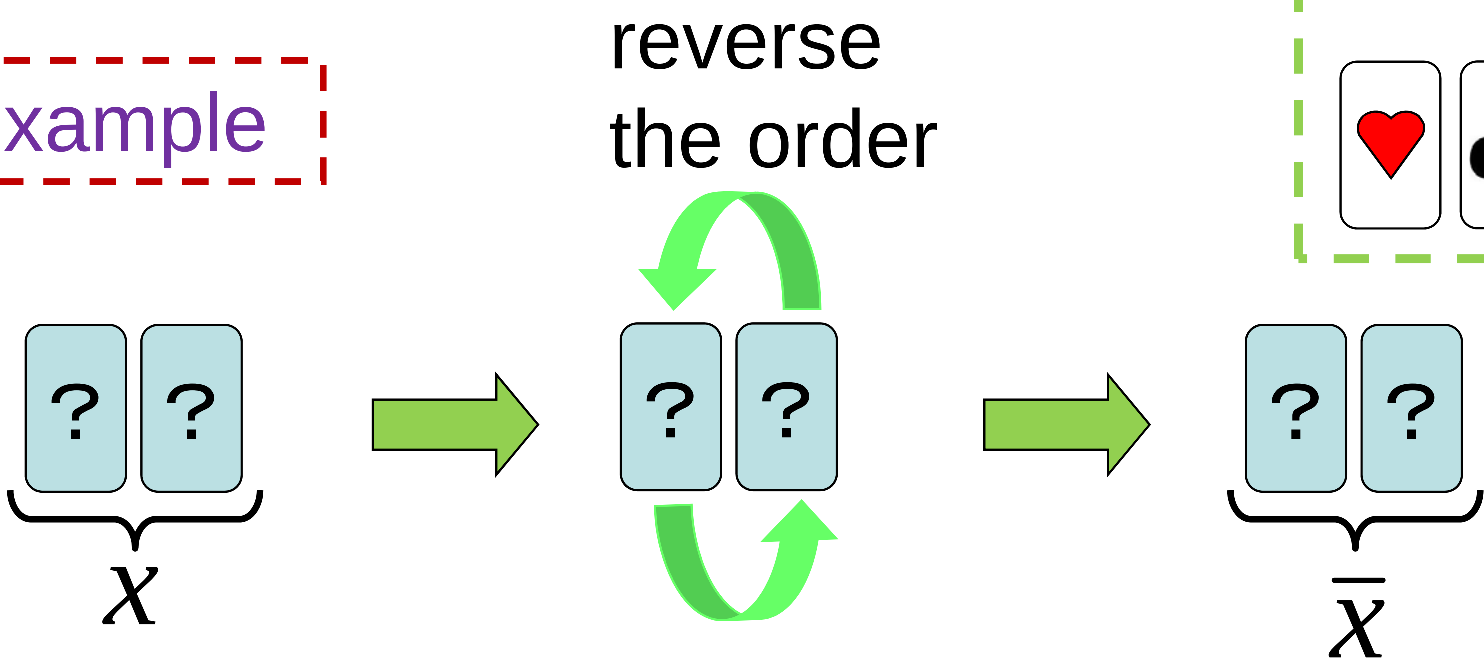


With keeping the value of x secret, we can get a commitment to the negation $\overline{x}$ of x .

Notations and the history



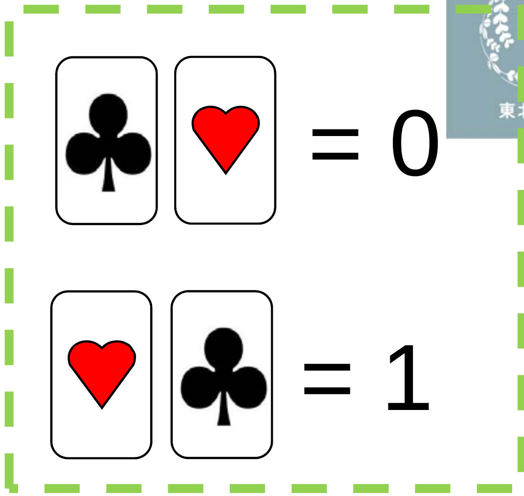
Example



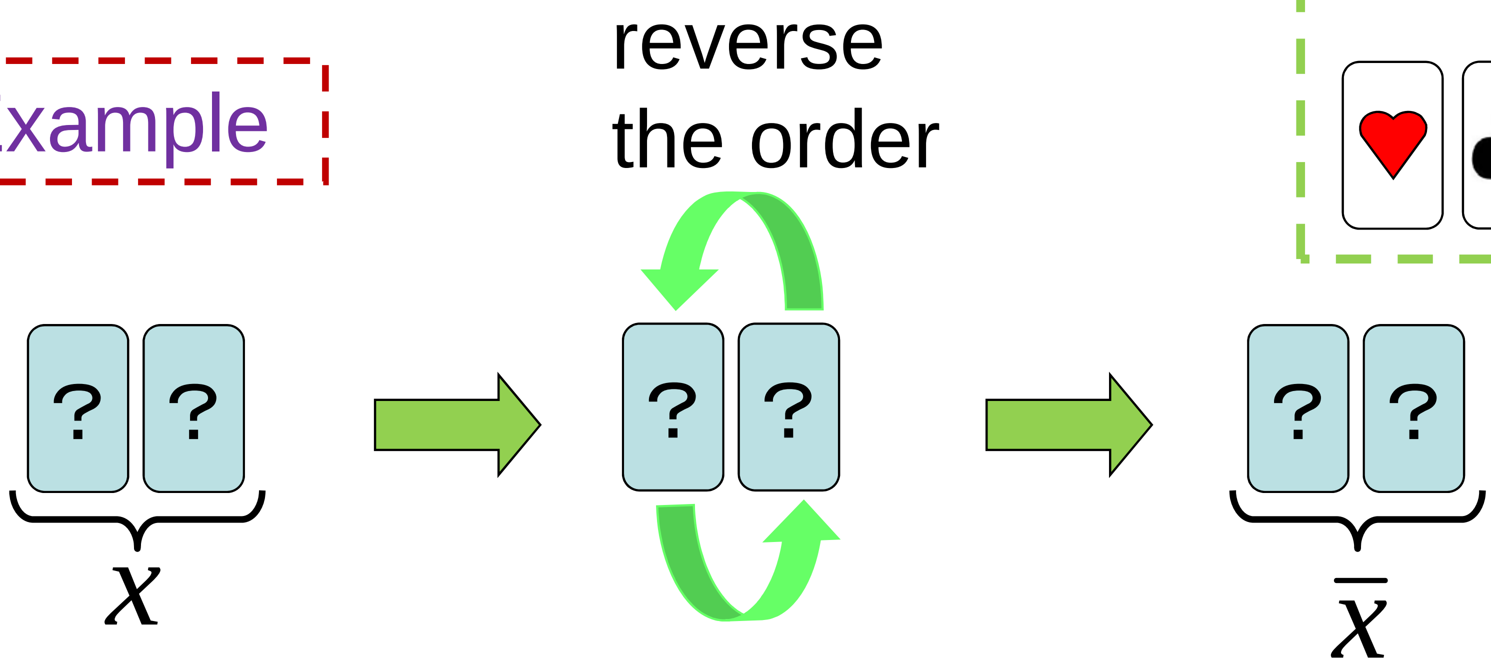
With keeping the value of x secret, we can get a commitment to the negation $\bar{x}$ of x .

- Secure **NOT** operation is trivial.

Notations and the history



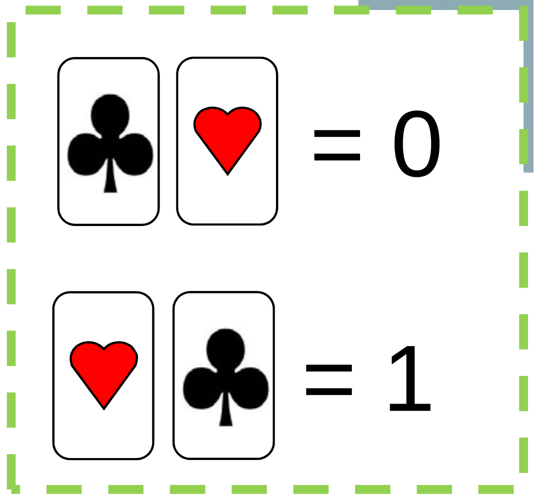
Example



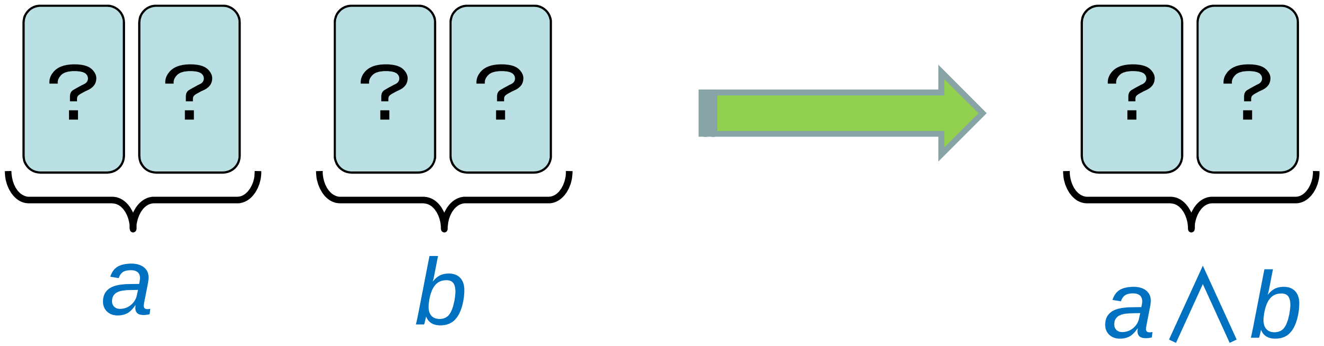
With keeping the value of x secret, we can get a commitment to the negation $\overline{x}$ of x .

- Secure **NOT** operation is trivial.
- How about secure **AND** operation?

Notations and the history



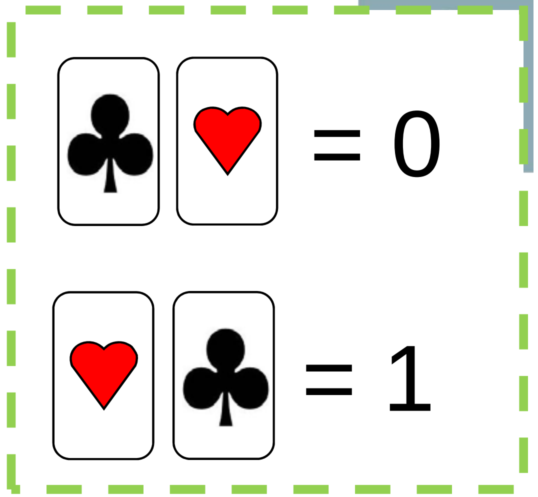
- How about secure **AND** operation?

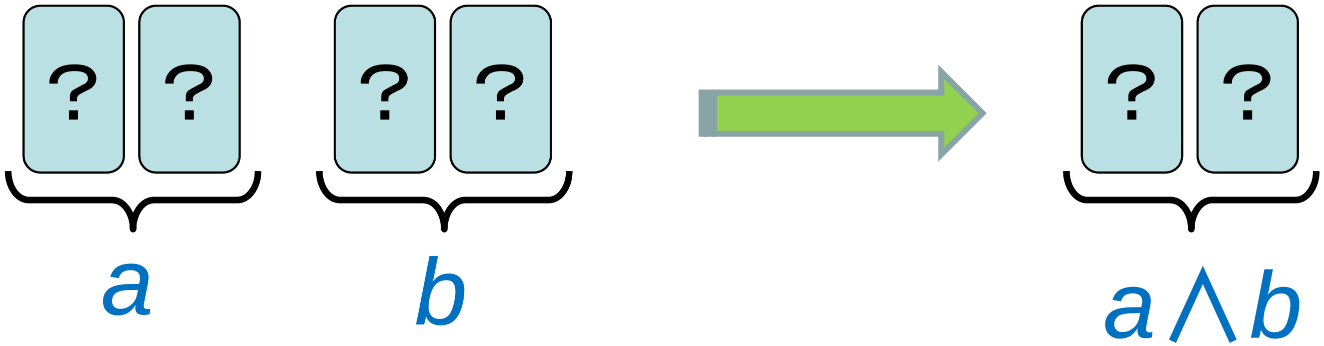


With keeping the values of a and b secret,
we want to get a commitment to $a \wedge b$.

Notations and the history

➢ How about secure **AND** operation?

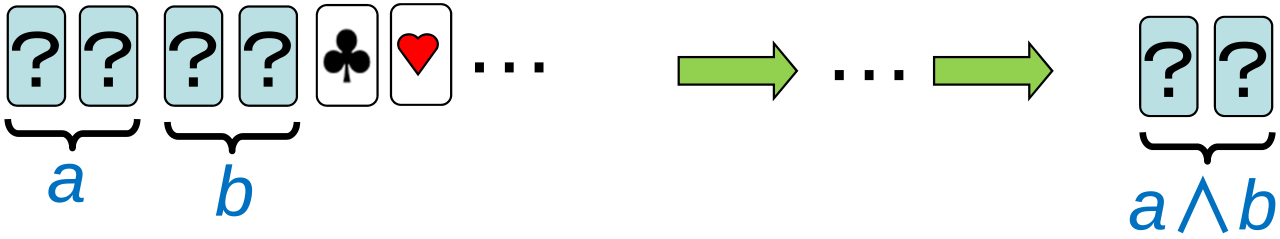




With keeping the values of a and b secret, we want to get a commitment to $a \wedge b$.

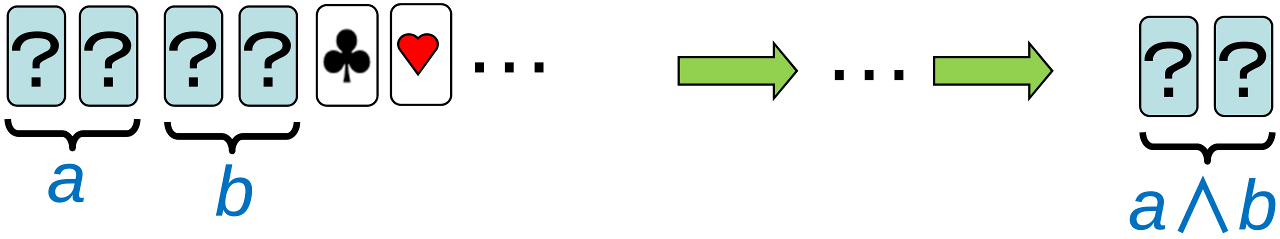
There have been such four protocols in the literatures.

History of Secure AND protocols



AND	required cards	avg. # of trials
Crepeau-Kilian [CRYPTO ’93]	10 ♣♣♣♥♥♥♠♠♦♦	6
Niemi-Renvall [TCS, 1998]	12 ♣♣♣♣♣♣♥♥♥♥♥♥	2.5
Stiglic [TCS, 2001]	8 ♣♣♣♣♥♥♥♥	2
Mizuki-Sone [FAW 2009]	6 ♣♣♣♥♥♥	1

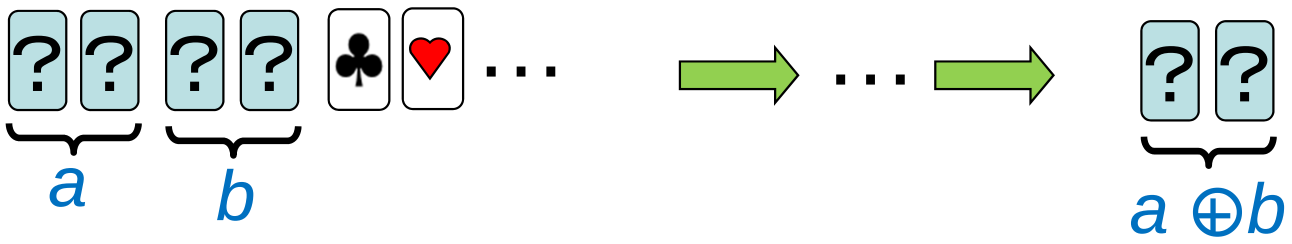
History of Secure AND protocols



AND	required cards	avg. # of trials
Crepeau-Kilian [CRYPTO '93]	10	6
Niemi-Renvall [TCS, 1998]	12	[illegible]
Stiglic [TCS, 2001]	8	2
Mizuki-Sone [FAW 2009]	6	1

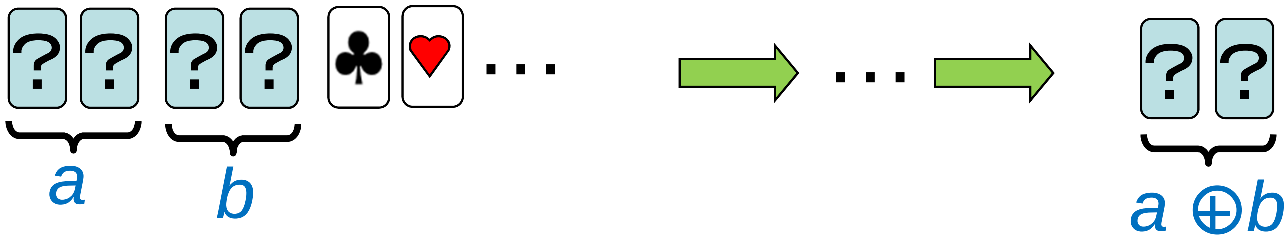
Will be introduced in Section 2.2

History of Secure XOR protocols



XOR	# of required cards	# of types	avg. # of trials
Crepeau-Kilian [CRYPTO '93]	14	4	6
Mizuki, et. al [AJoC, 2006]	10	2	2
Mizuki-Sone [FAW 2009]	4	2	1

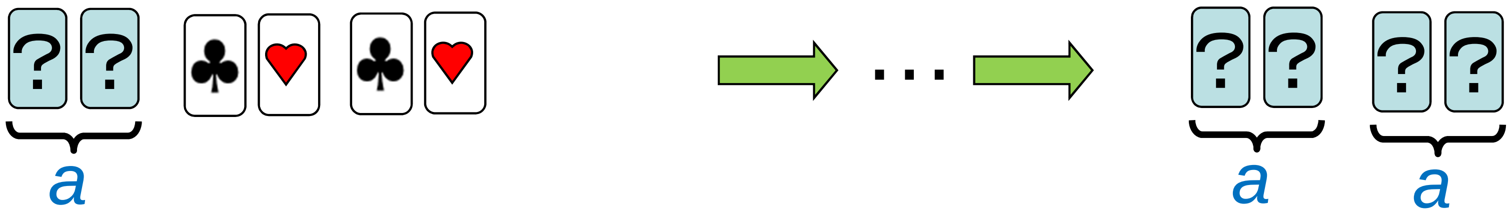
History of Secure XOR protocols



XOR	# of required cards		avg. #
Crepeau-Kilian [CRYPTO '93]	14		
Mizuki, et. al [AJoC, 2006]	10	2	2
Mizuki-Sone [FAW 2009]	4	2	1

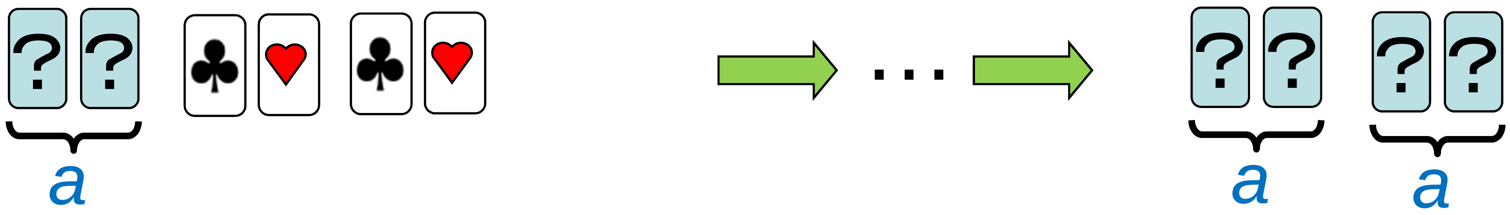
Will be introduced in Section 2.3

Existing COPY protocols



Make identical copies of a commitment.

Existing COPY protocols

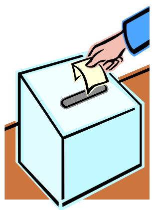


Make identical copies of a commitment.

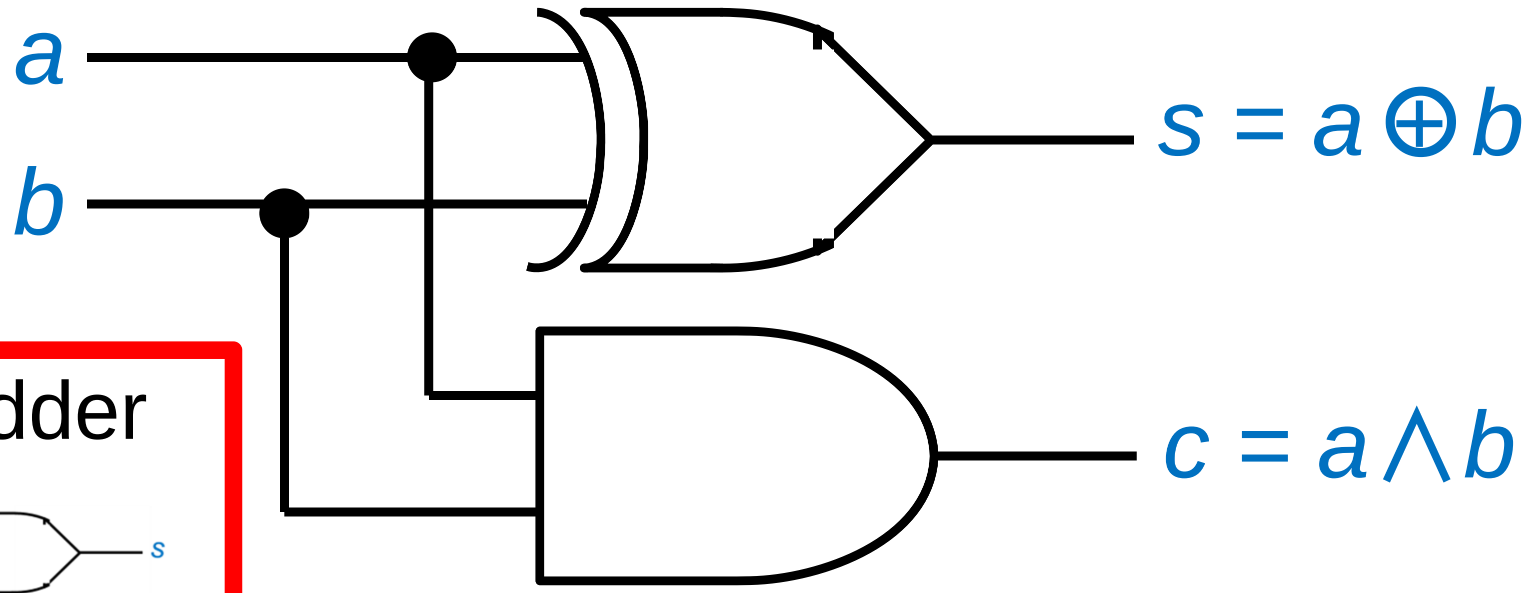
I'll introduce the best existing COPY protocol in Section 2.4

Outline of our results

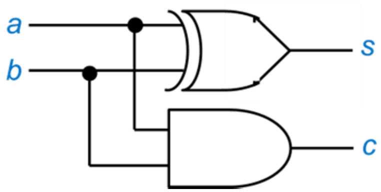
Voting



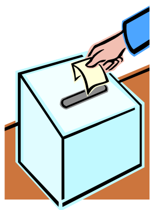
Outline of our results



Half adder

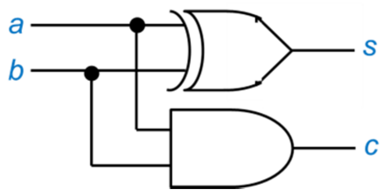
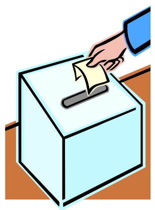


Voting



Outline of our results

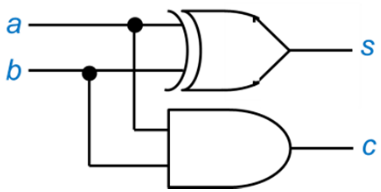
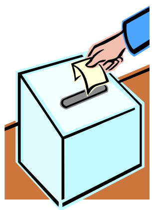
Using existing AND/XOR/COPY protocols

Half adder 	10
Voting 	

[# of cards]

Outline of our results

Using existing AND/XOR/COPY protocols

Half adder 	10
Voting 	$2\lceil \log n \rceil + 8$

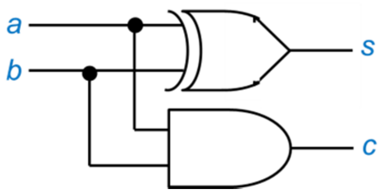
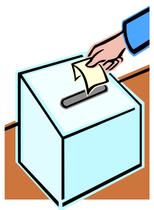
[# of cards]

Applying a half adder

Outline of our results

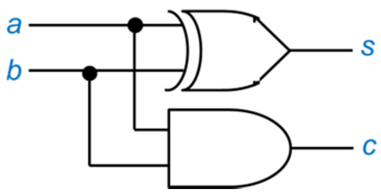
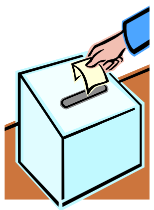
Using existing AND/XOR/COPY protocols

Devising a tailor-made half adder

Half adder 	10	8
Voting 	$2\lceil \log n \rceil + 8$	

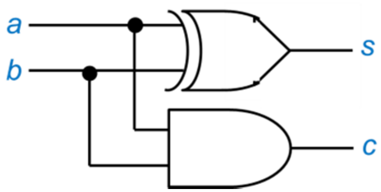
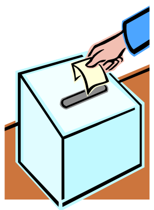
[# of cards]

Outline of our results

	Using existing AND/XOR/COPY protocols	Devising a tailor-made half adder
Half adder 	10	8
Voting 	$2\lceil \log n \rceil + 8$	$2\lceil \log n \rceil + 6$

[# of cards]

Outline of our results

	Using existing AND/XOR/COPY protocols	Devising a tailor-made half adder
Half adder 	10	8
Voting 	$2\lceil \log n \rceil + 8$	$2\lceil \log n \rceil + 6$

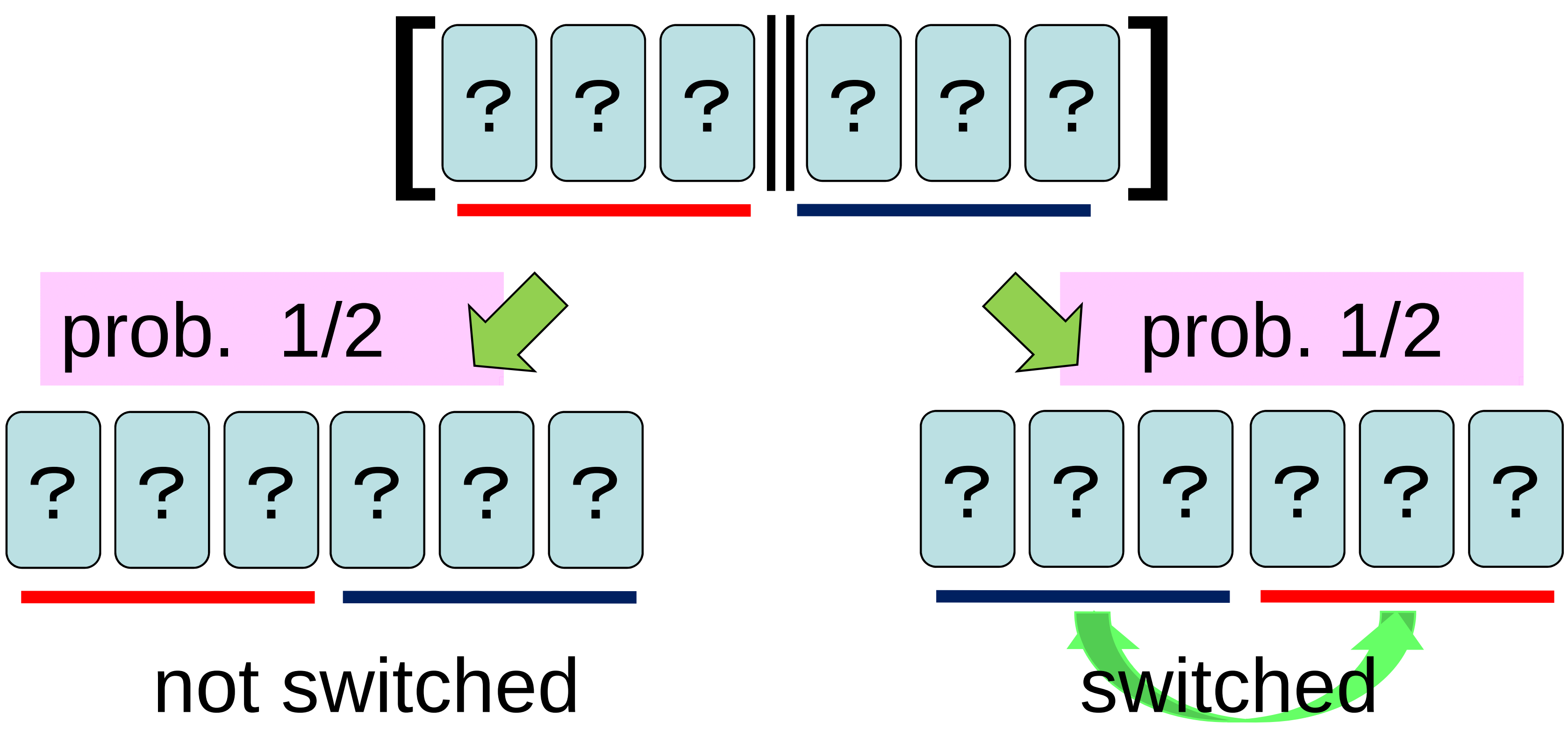
[# of cards]

Contents

2.1 Random Bisection Cuts

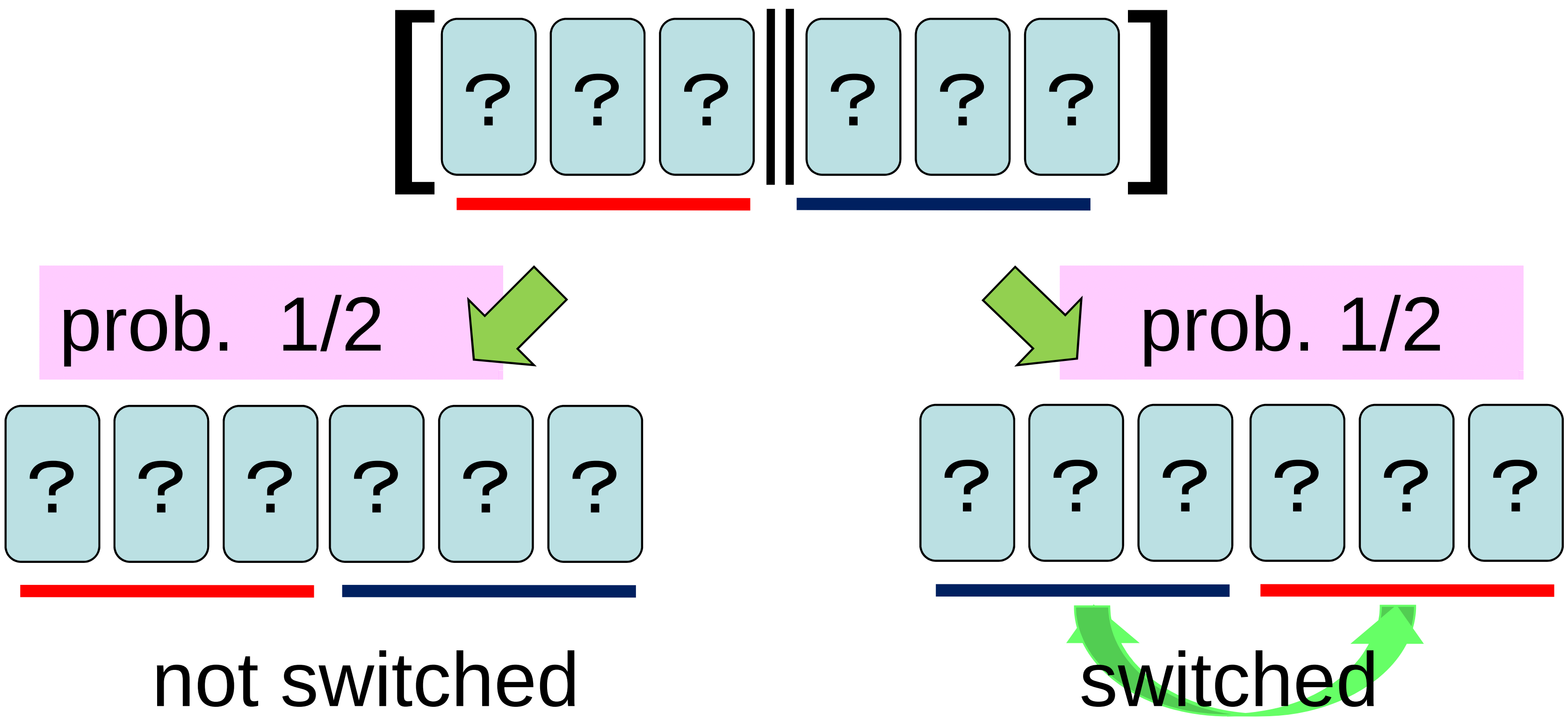
a *random bisection cut*

Bisect a given deck of cards, and then randomly switch the resulting two portions:



a *random bisection cut*

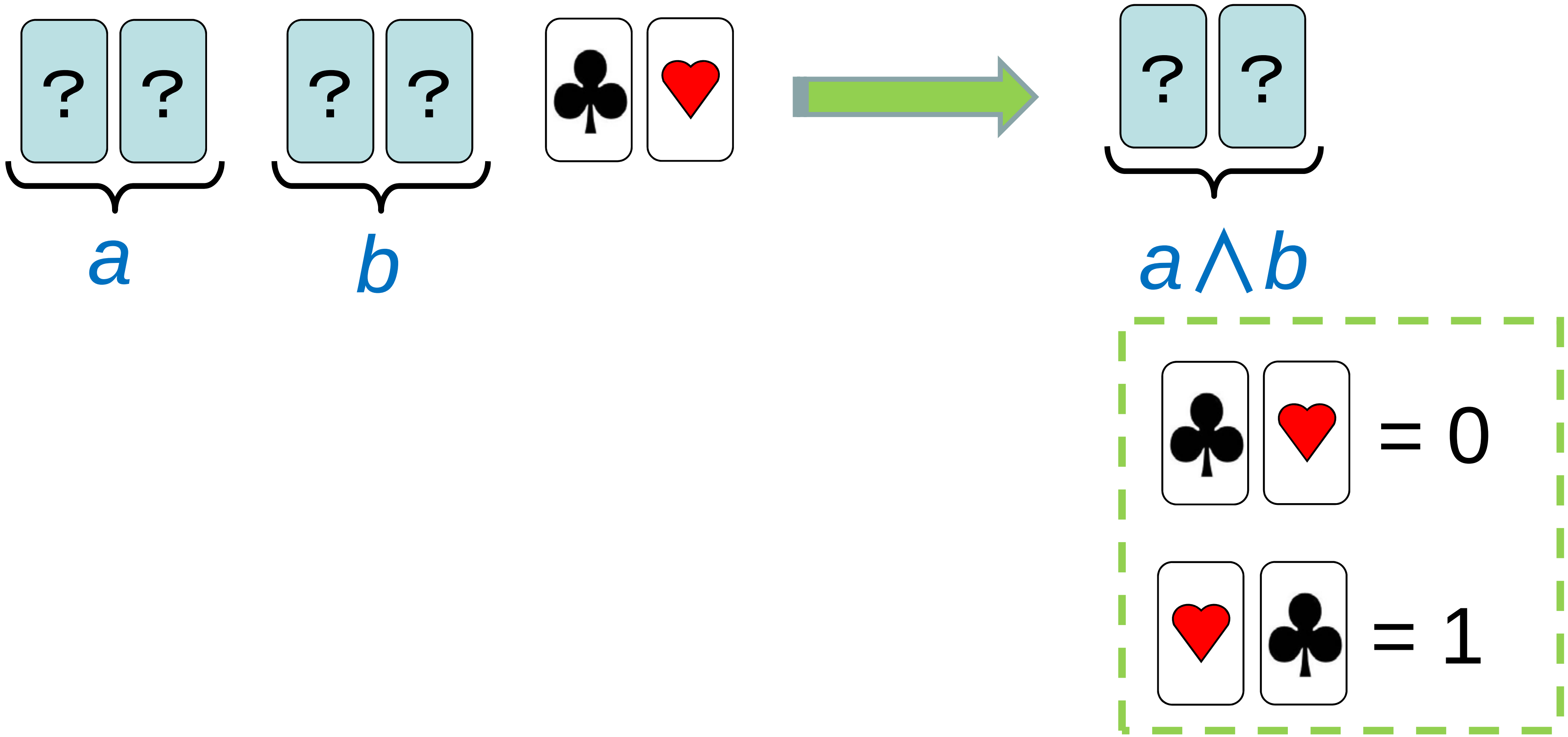
Bisect a given deck of cards, and then randomly switch the resulting two portions:



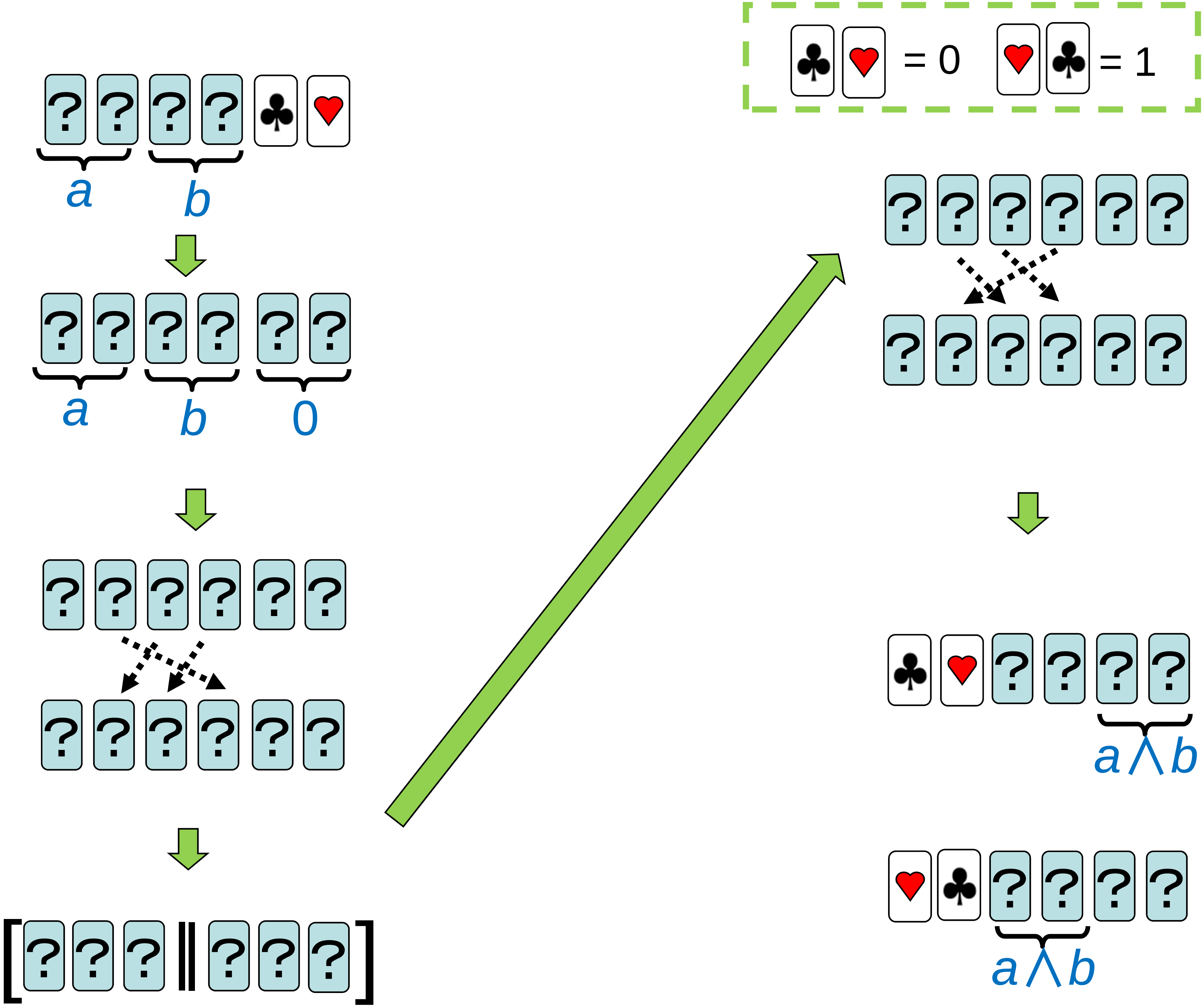
easy-to-implement card shuffling operation

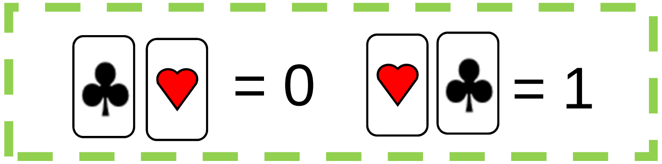
2.2 Six-Card AND Protocol

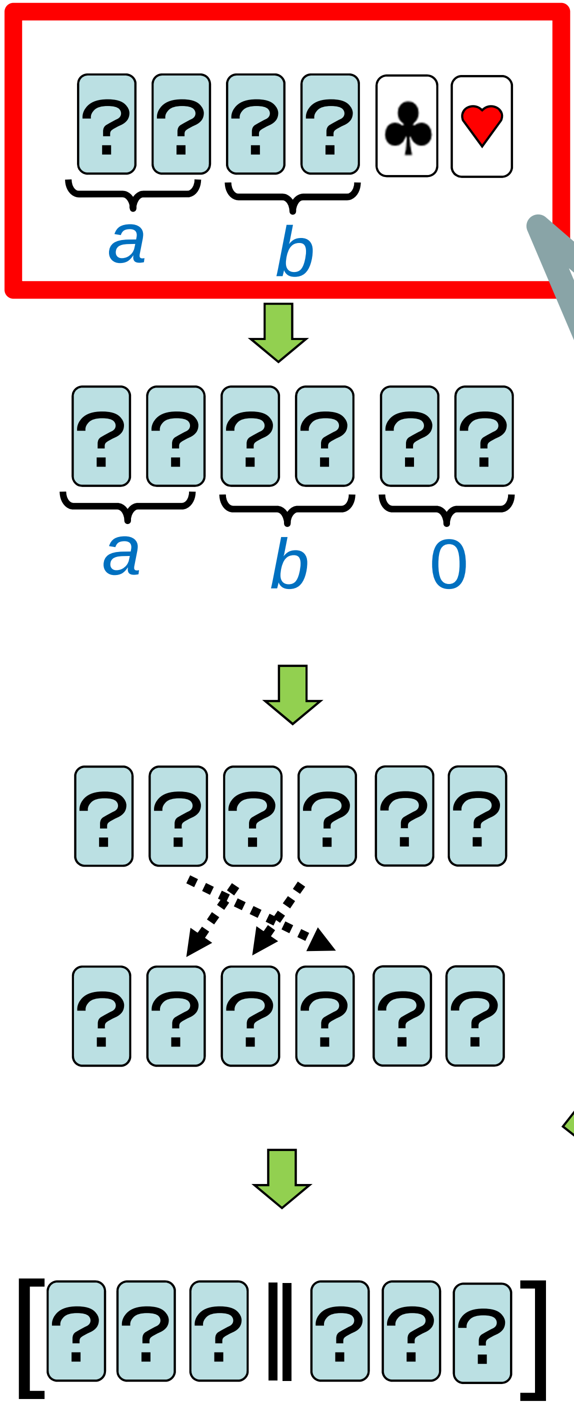
Secure AND can be done with 6 cards [6].



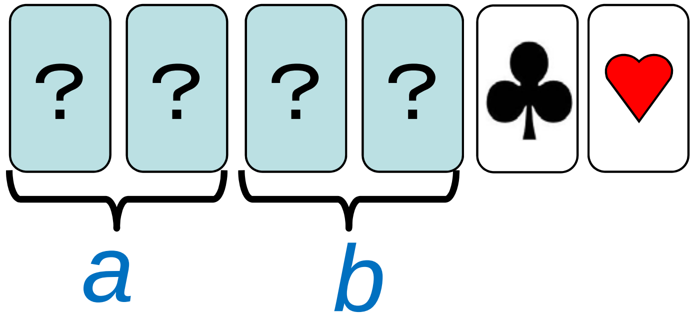
[6] T. Mizuki and H. Sone, Six-Card Secure AND and Four-Card Secure XOR, FAW 2009, LNCS 5598, pp. 358–369, 2009.

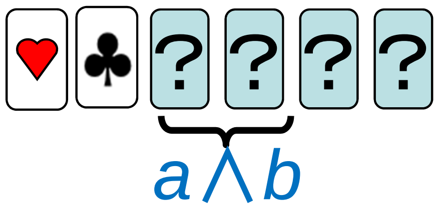

= 0
= 1
a
b
a
b
0
a∧b
a∧b

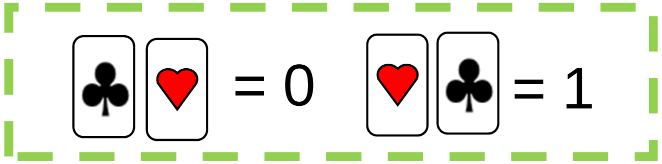


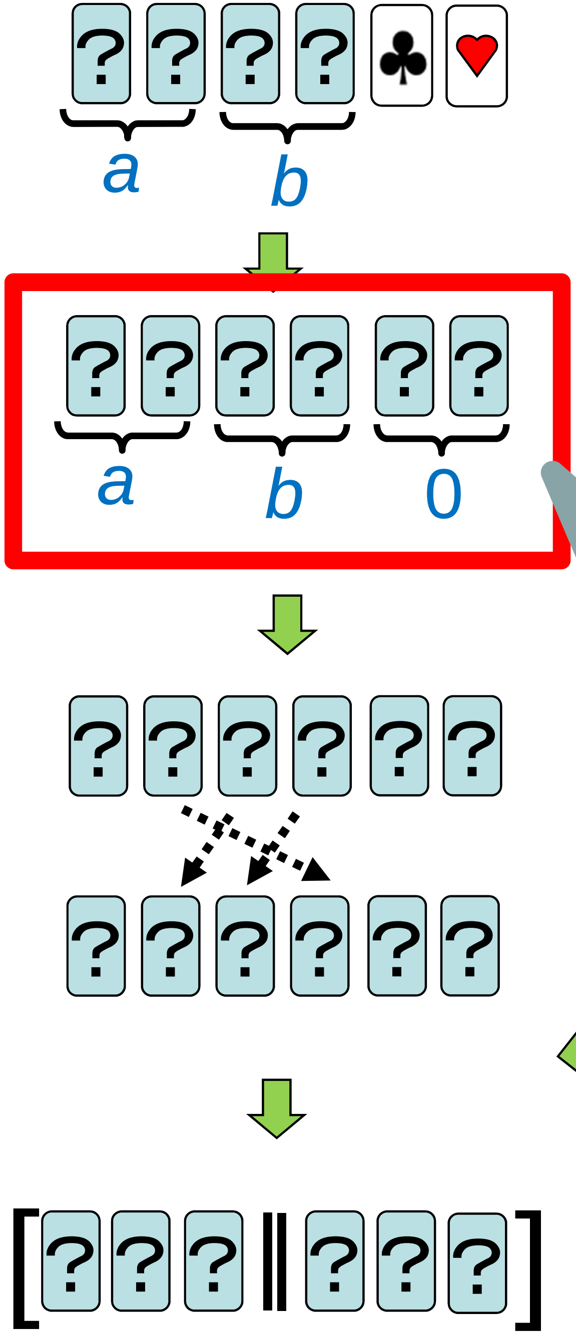


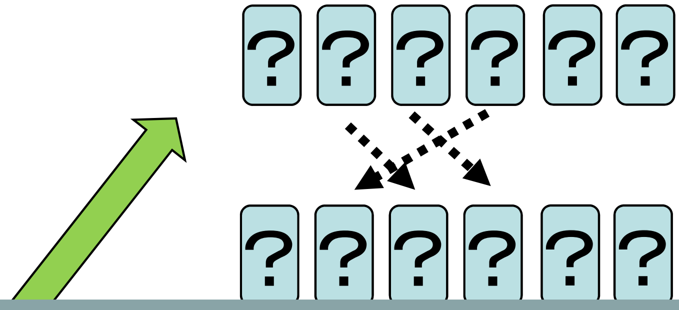
Arrange 2 commitments and 2 additional cards:



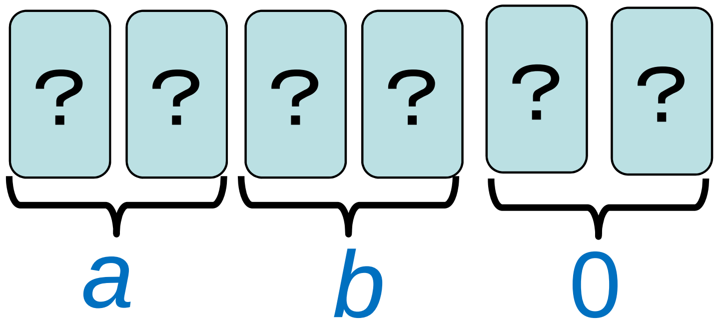




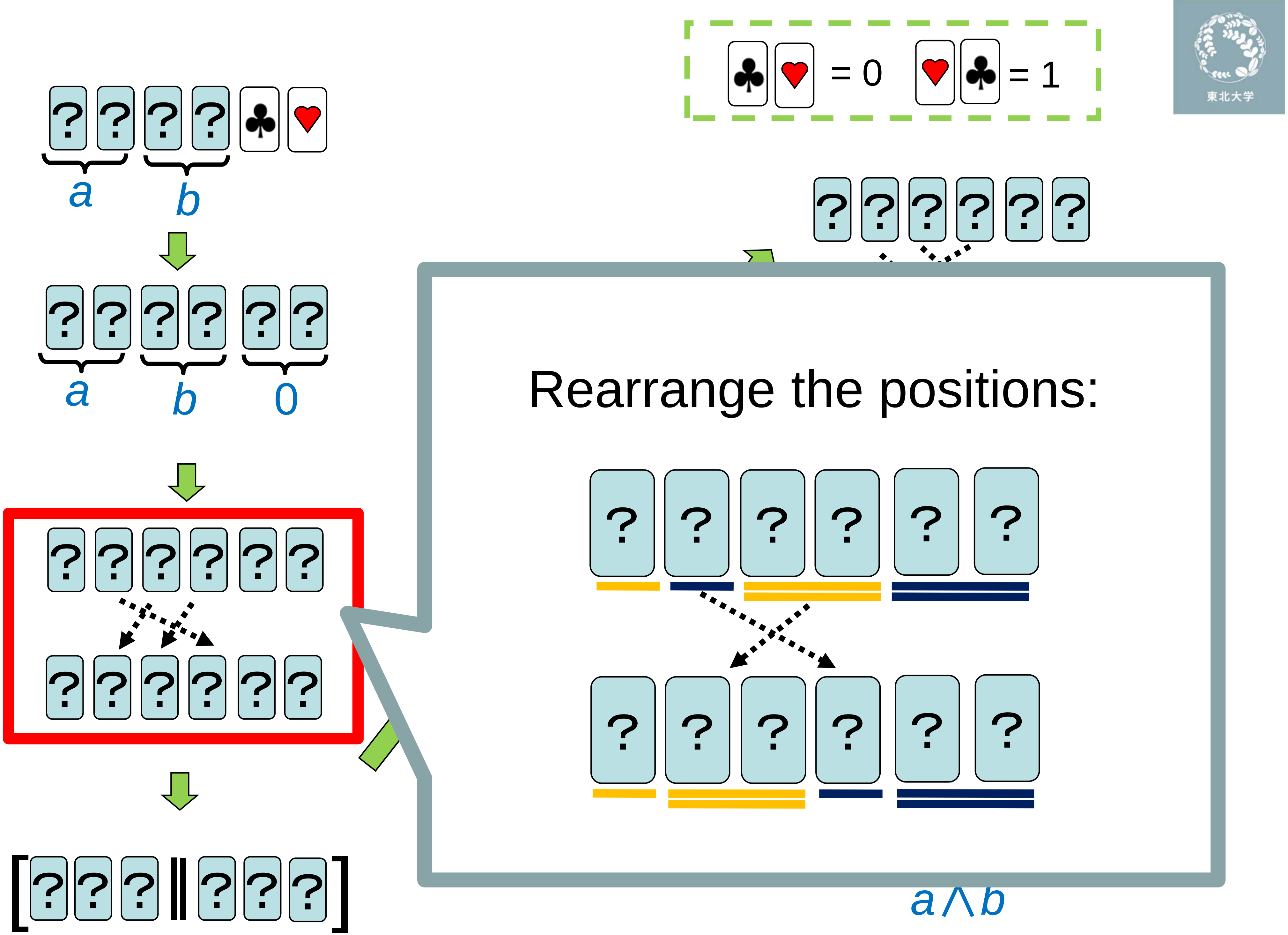




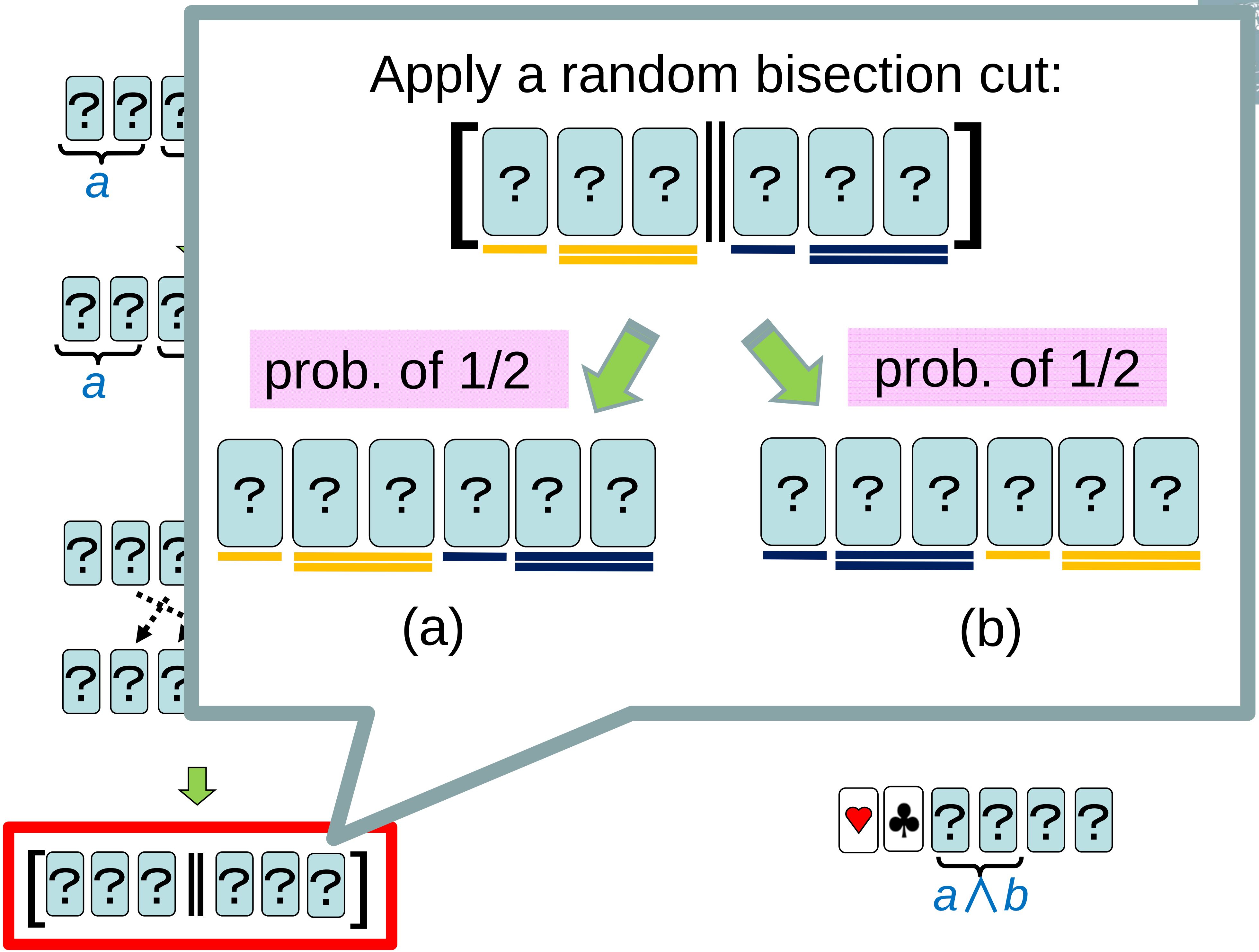
Turn over the rightmost two cards:

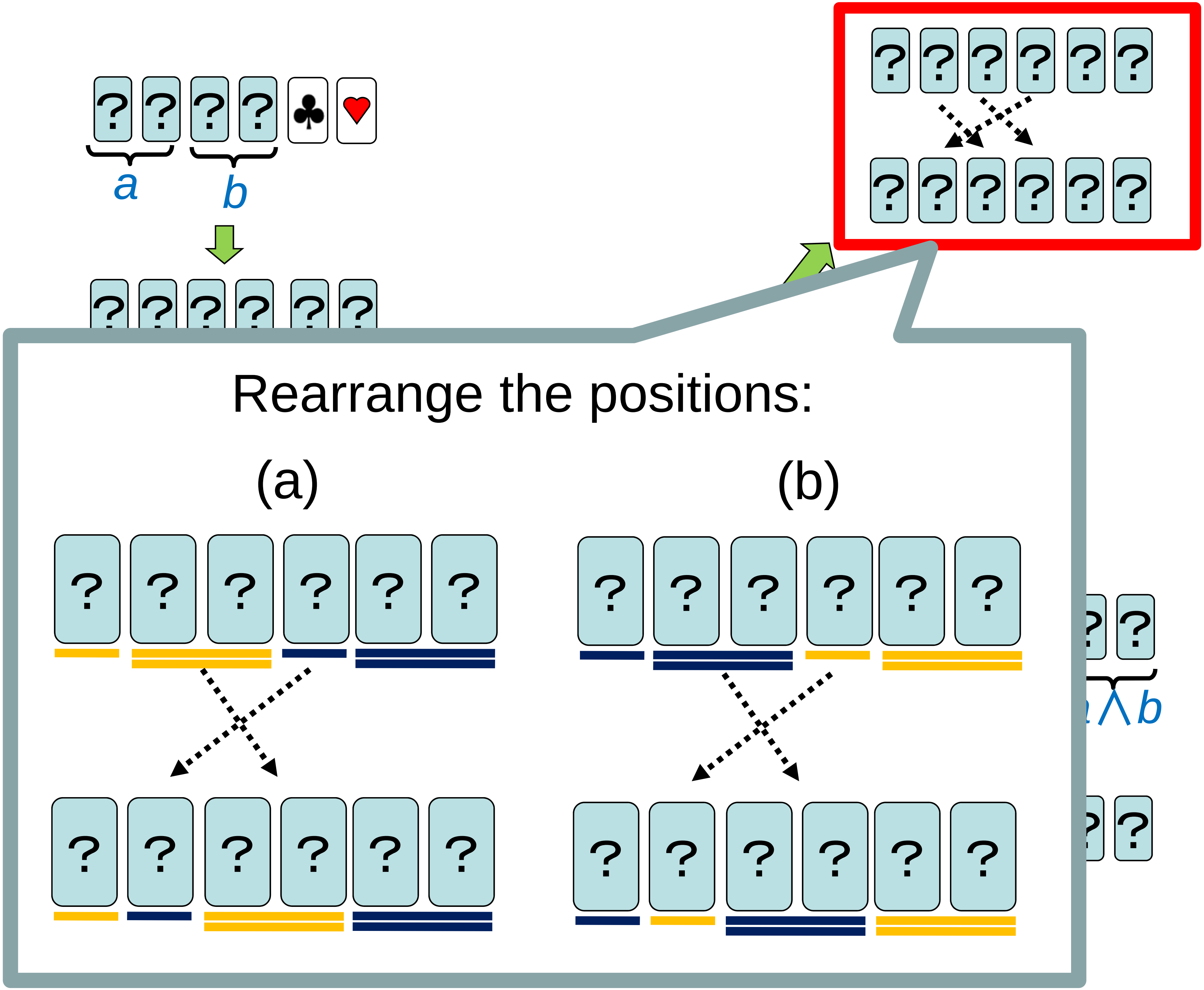


They become a commitment to 0.

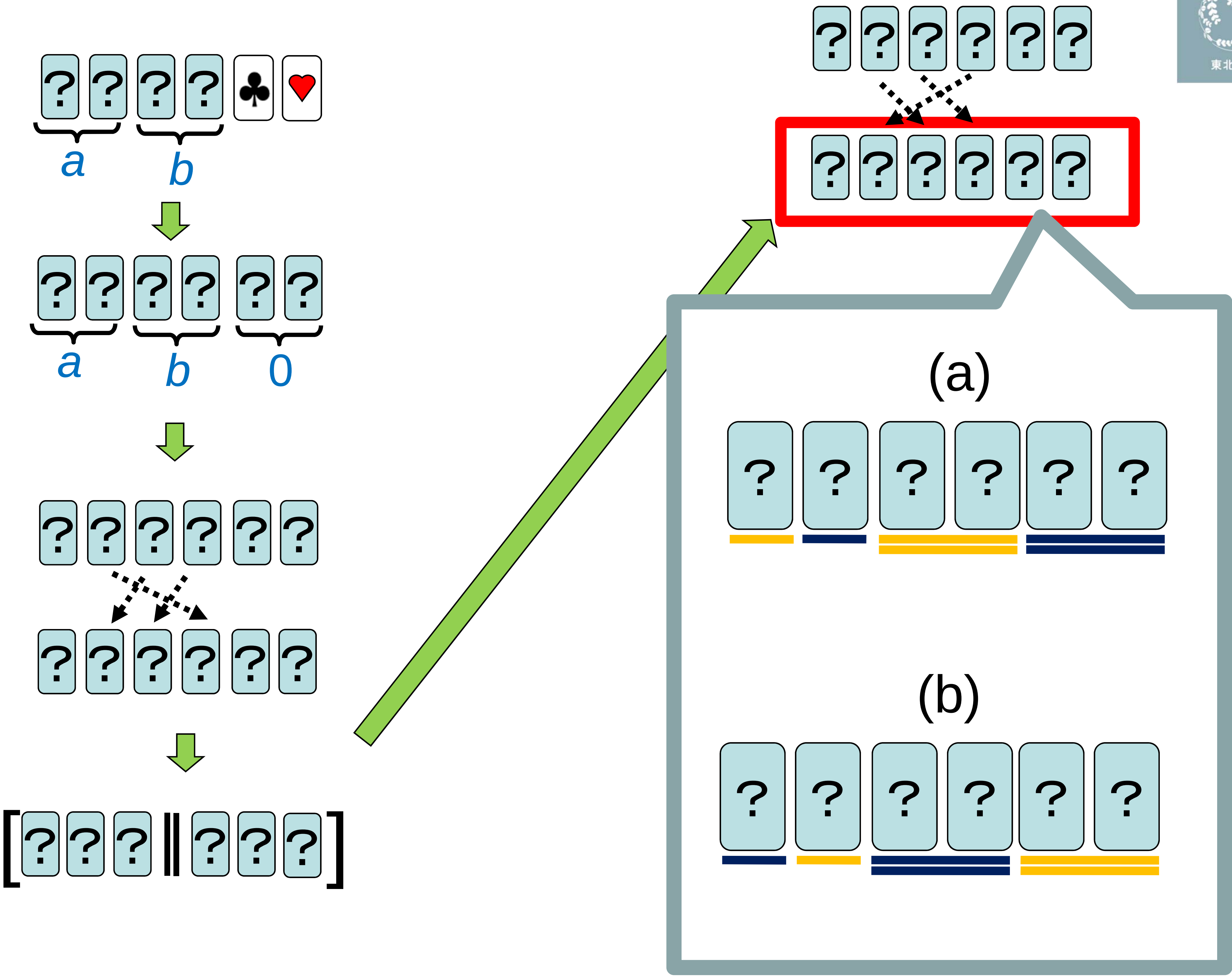


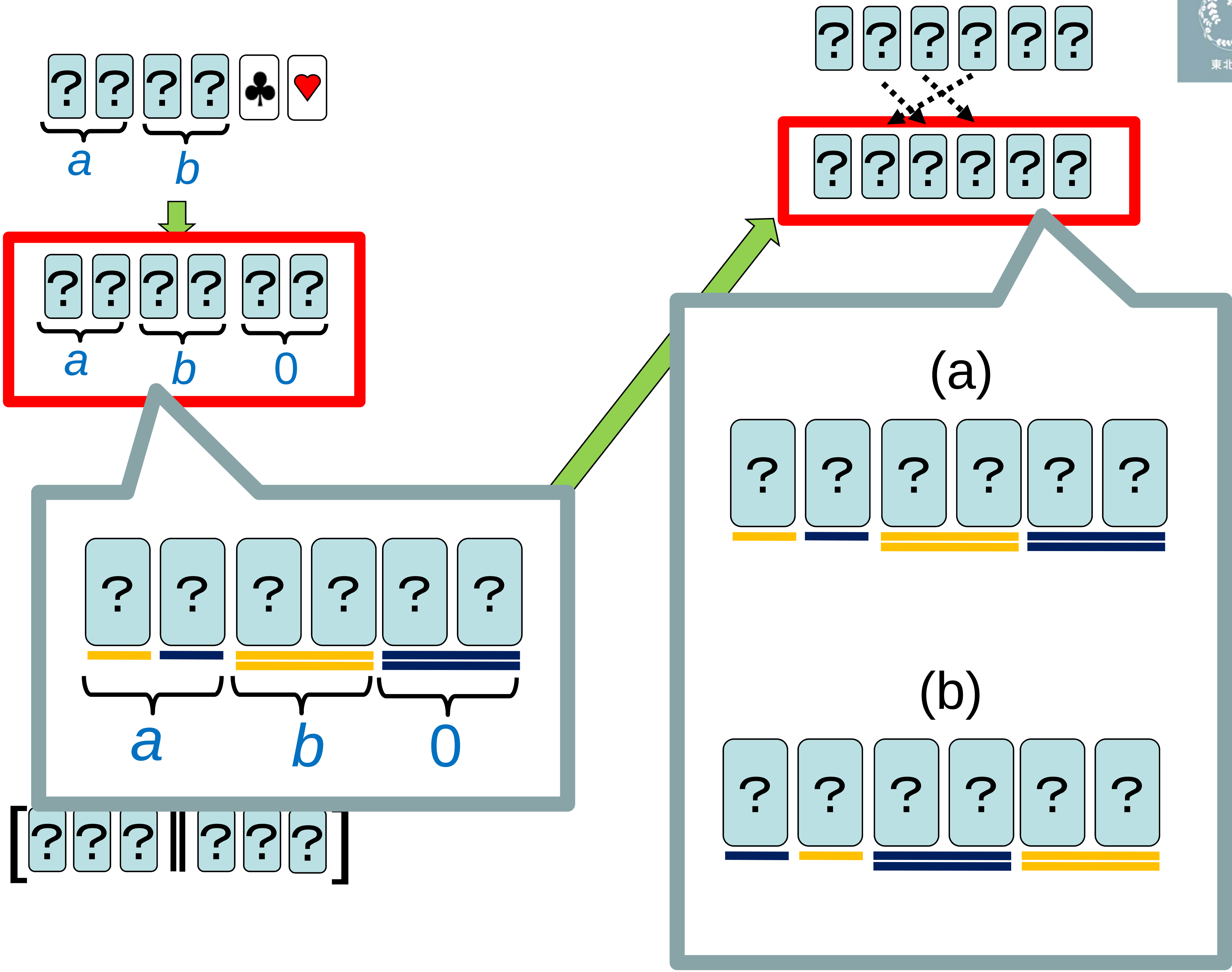
= 0
= 1
東北大学
a
b
a
b
0
Rearrange the positions:
a ∧ b

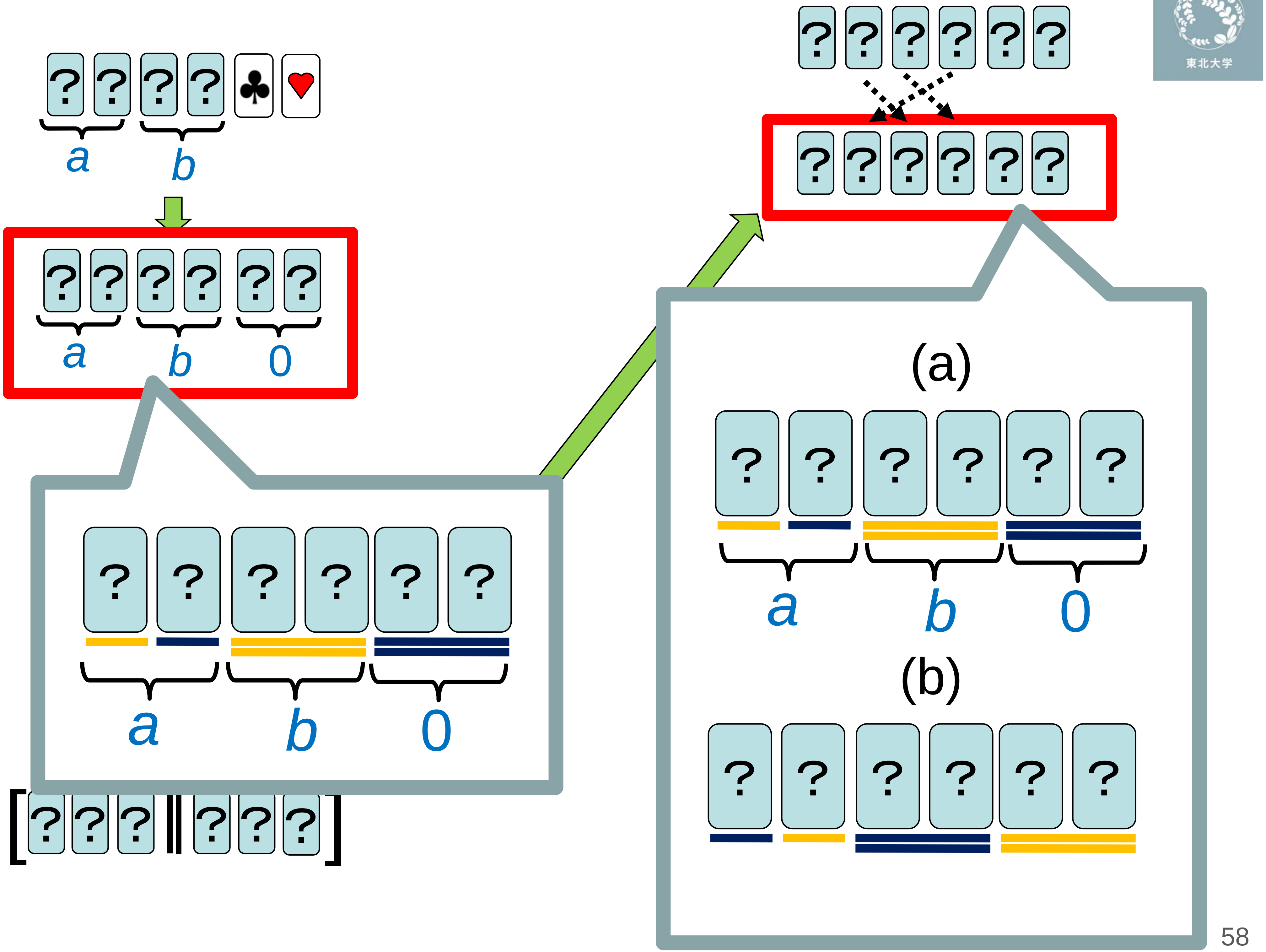

Apply a random bisection cut:
prob. of 1/2
prob. of 1/2
(a)
(b)
a
a
a∧b

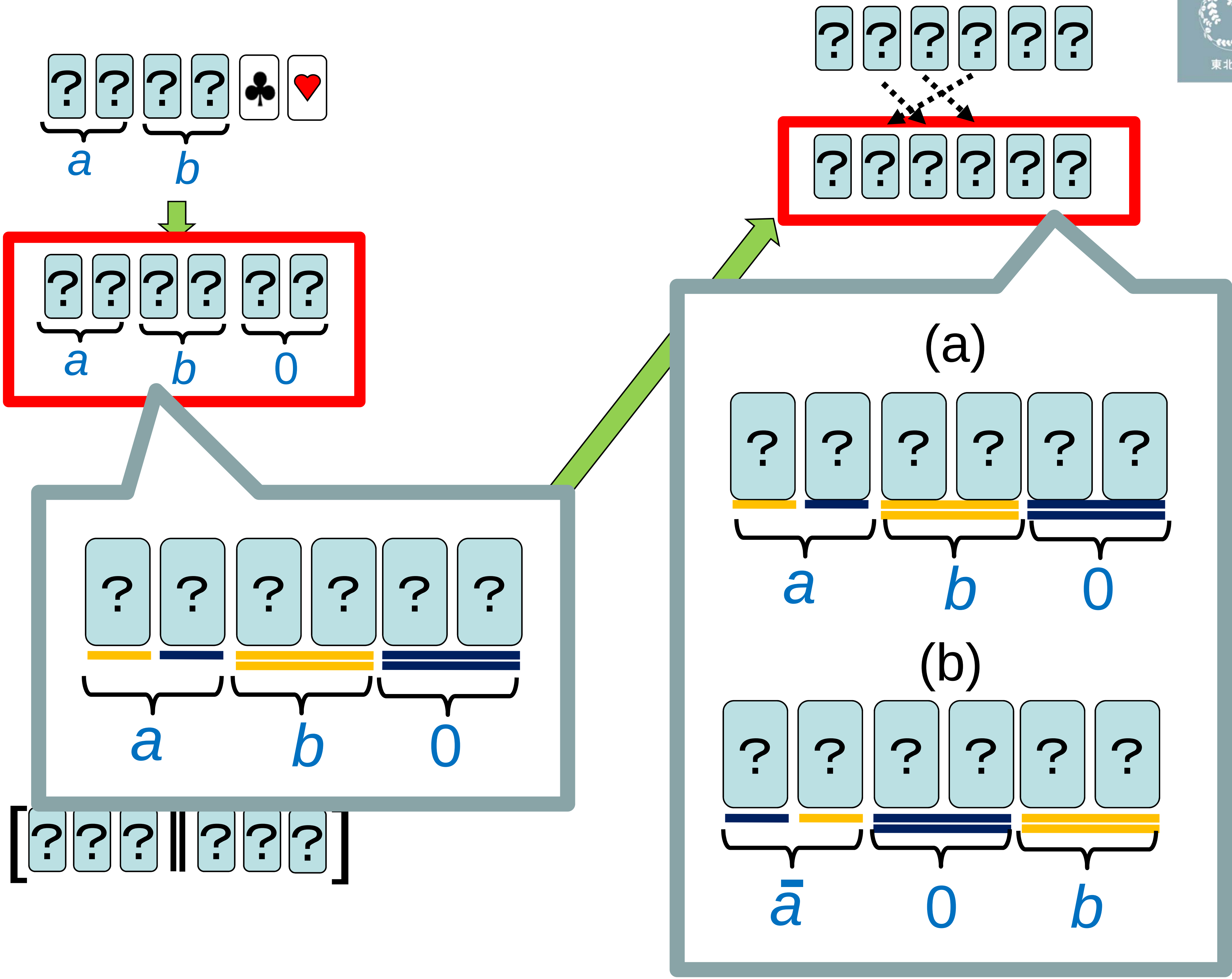


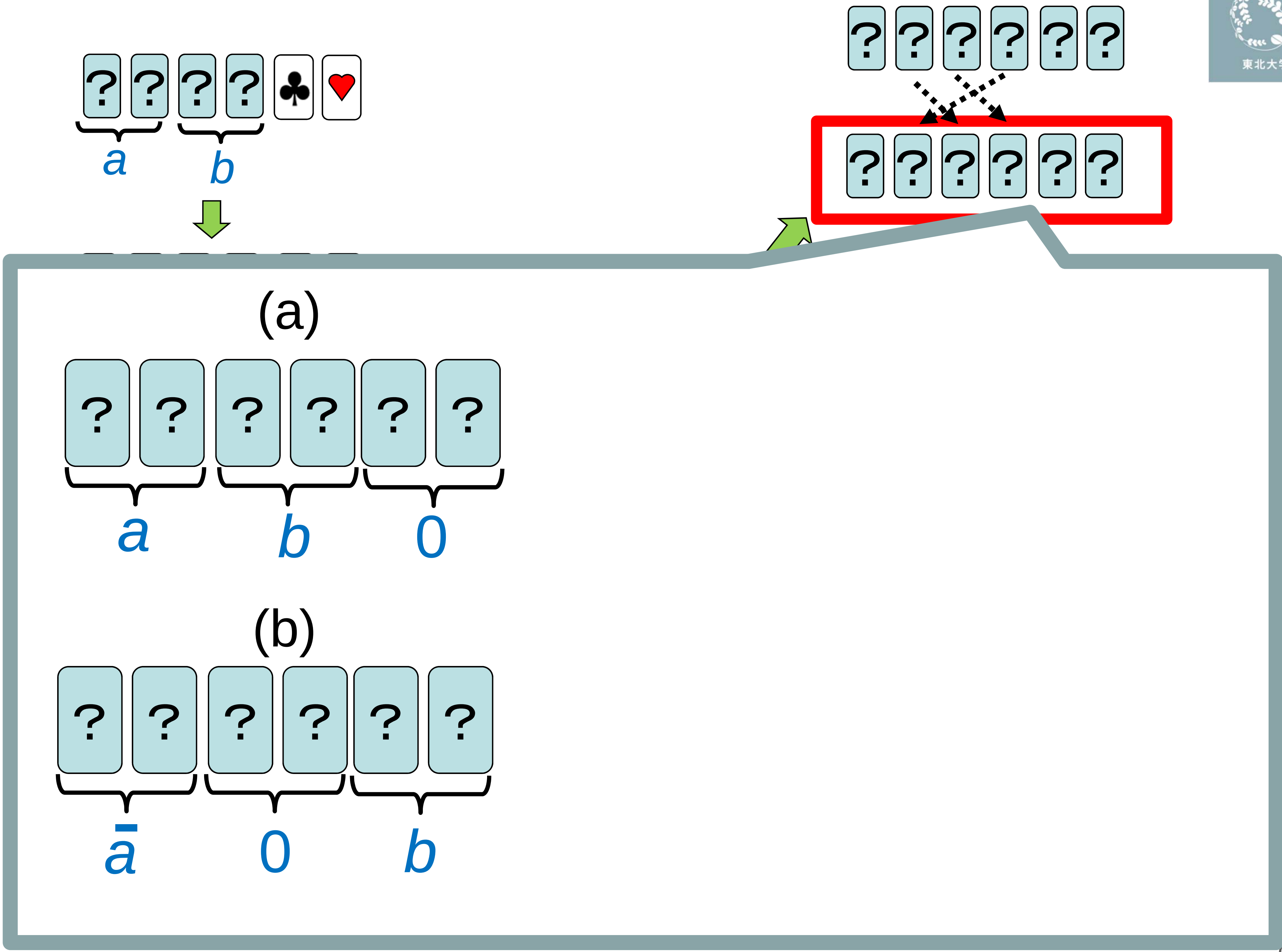
a
b
Rearrange the positions:
(a)
(b)
a∧b

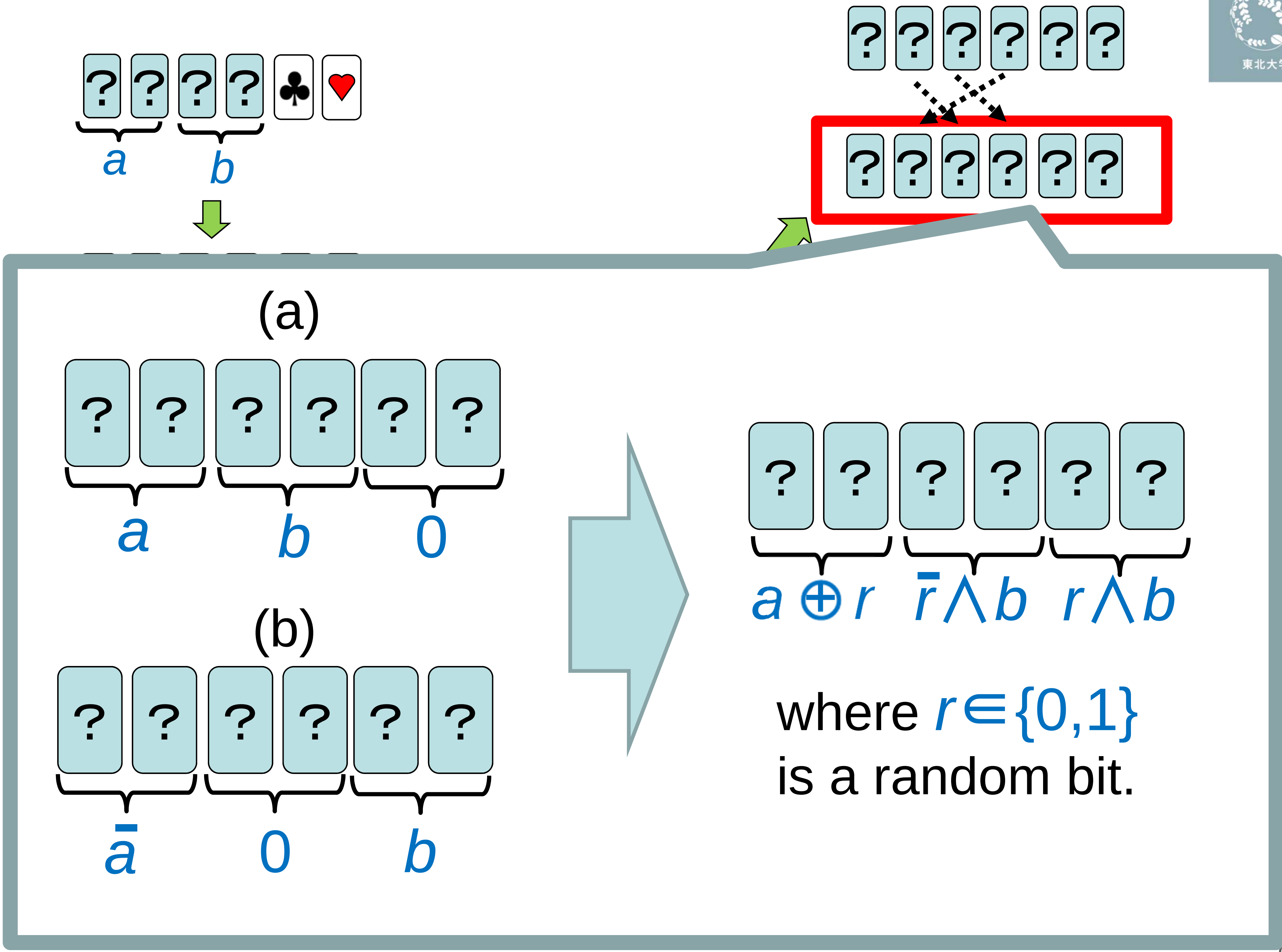

a
b
a
b
0
(a)
(b)

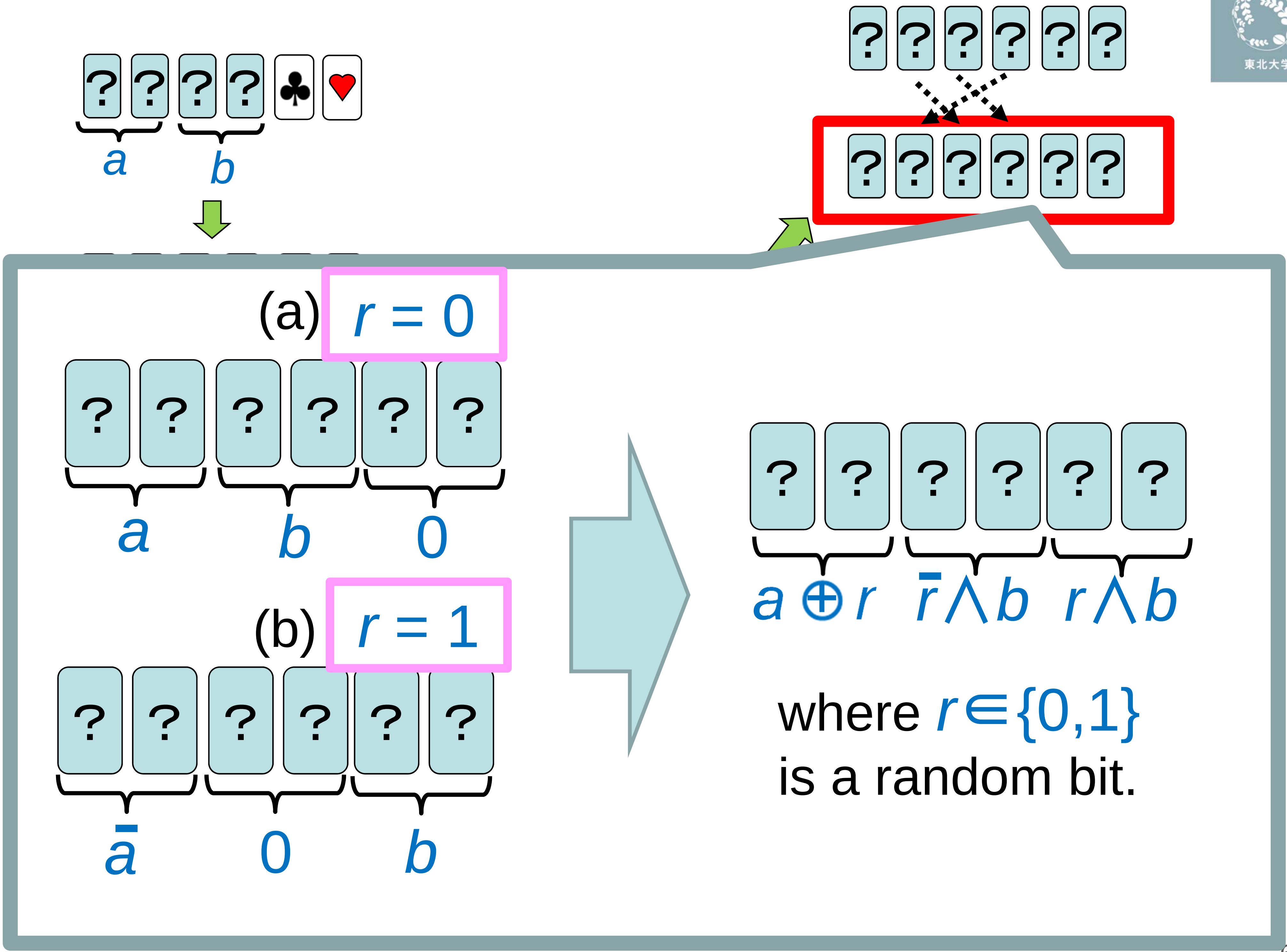

a
b
a
b
0
a
b
0
(a)
(b)

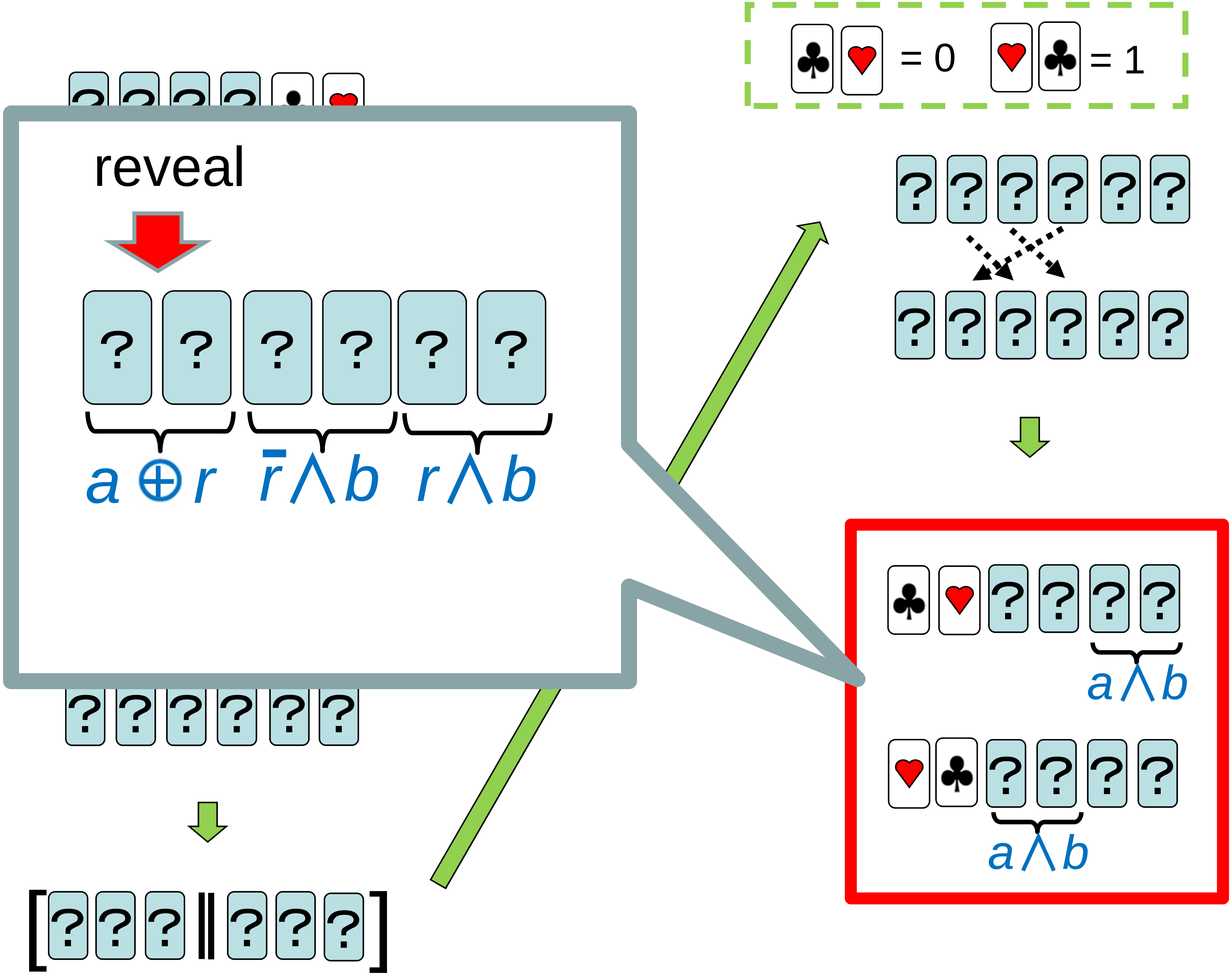

a
b
a
b
0
a
b
0
(a)
a
b
0
(b)
東北大学


a
b
a
b
0
a
b
0
(a)
a
b
0
(b)
ā
0
b

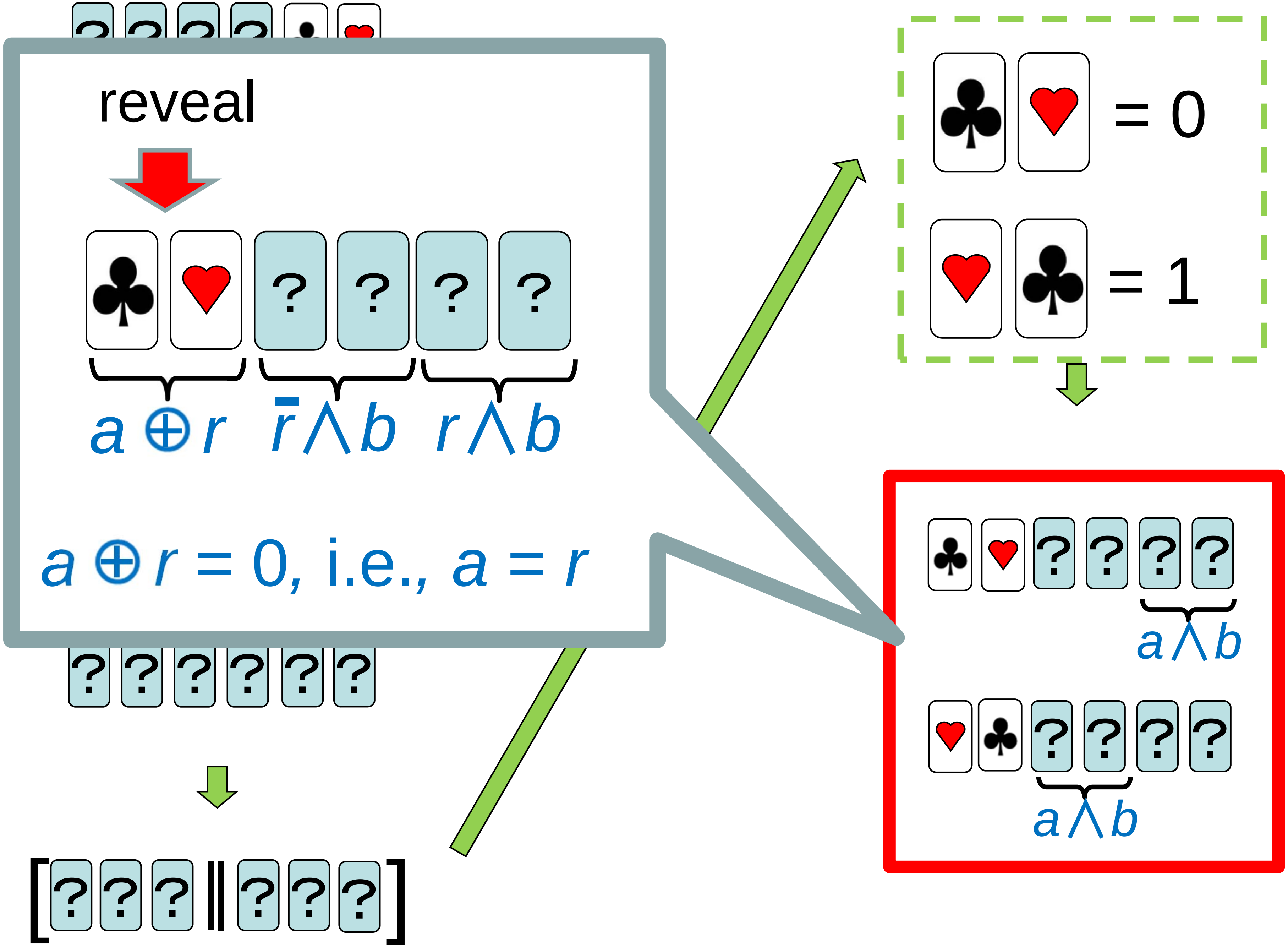

a
b
(a)
a
b
0
(b)
$\bar{a}$
0
b

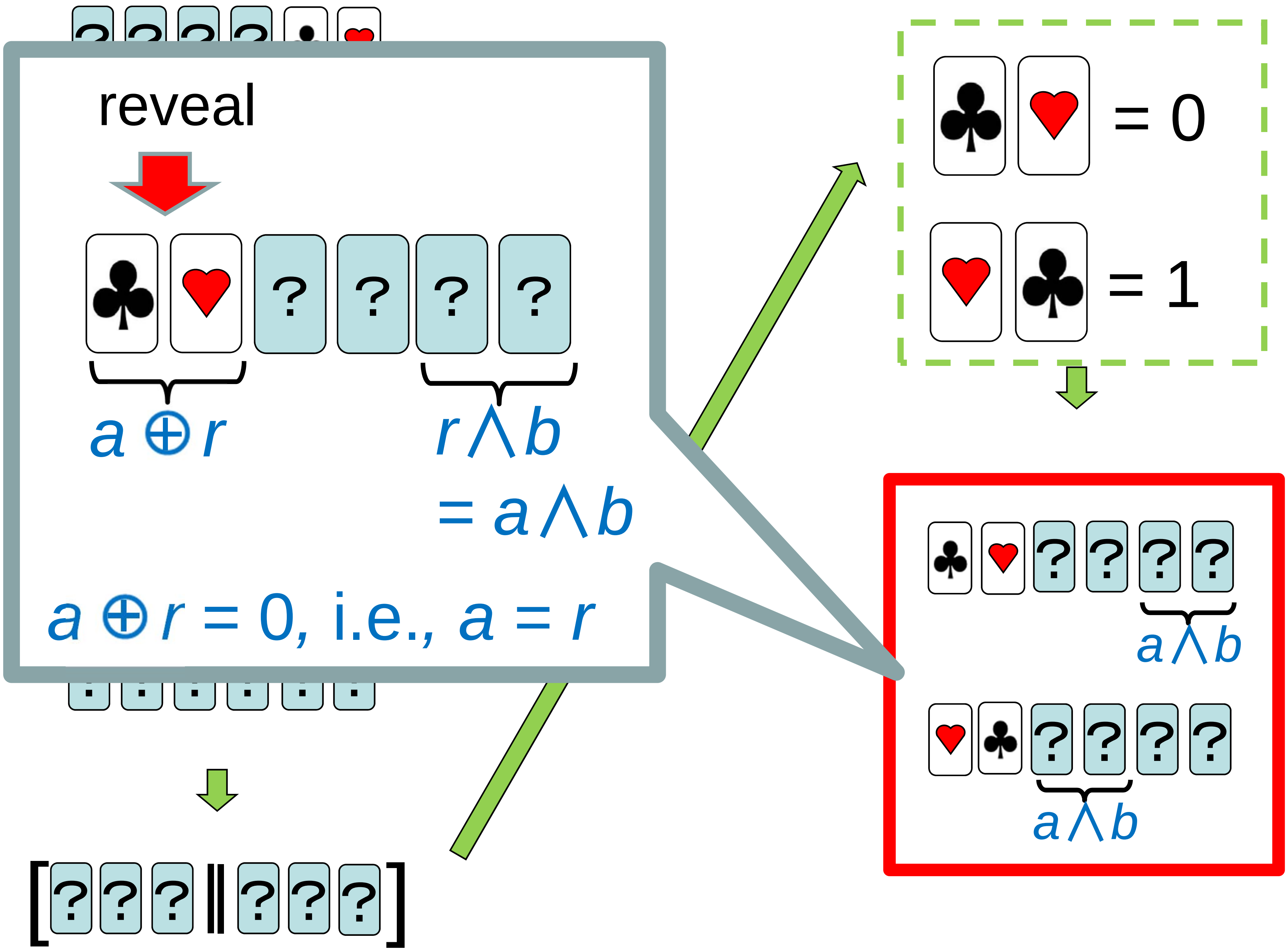

a
b
(a)
a
b
0
(b)
ā
0
b
a ⊕ r
r̄ ∧ b
r ∧ b
where r ∈ {0,1}
is a random bit.


a
b
(a) $r = 0$
a b 0
(b) $r = 1$
$\bar{a}$ 0 b
$a \oplus r$ $\bar{r} \land b$ $r \land b$
where $r \in \{0,1\}$
is a random bit.

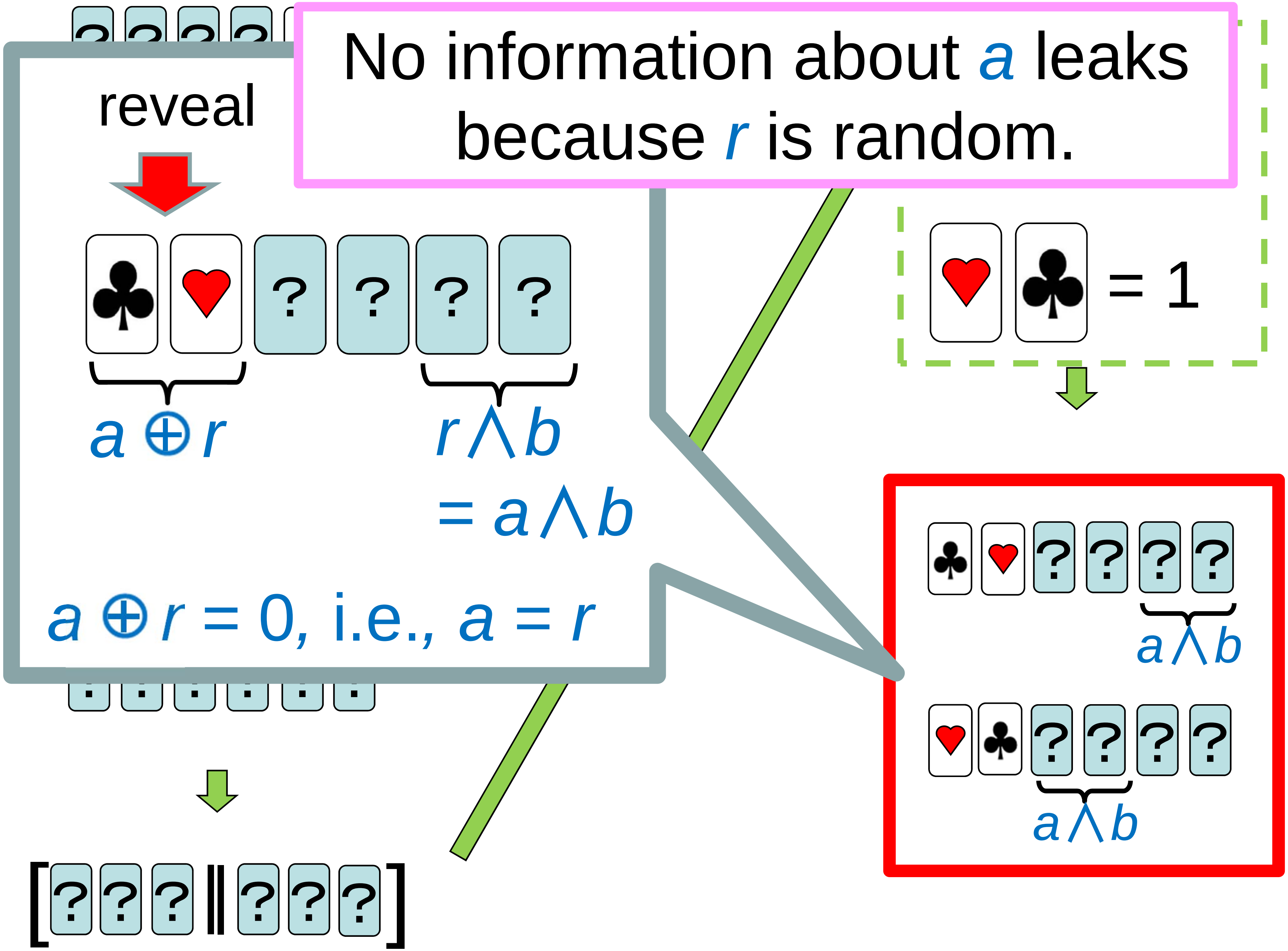


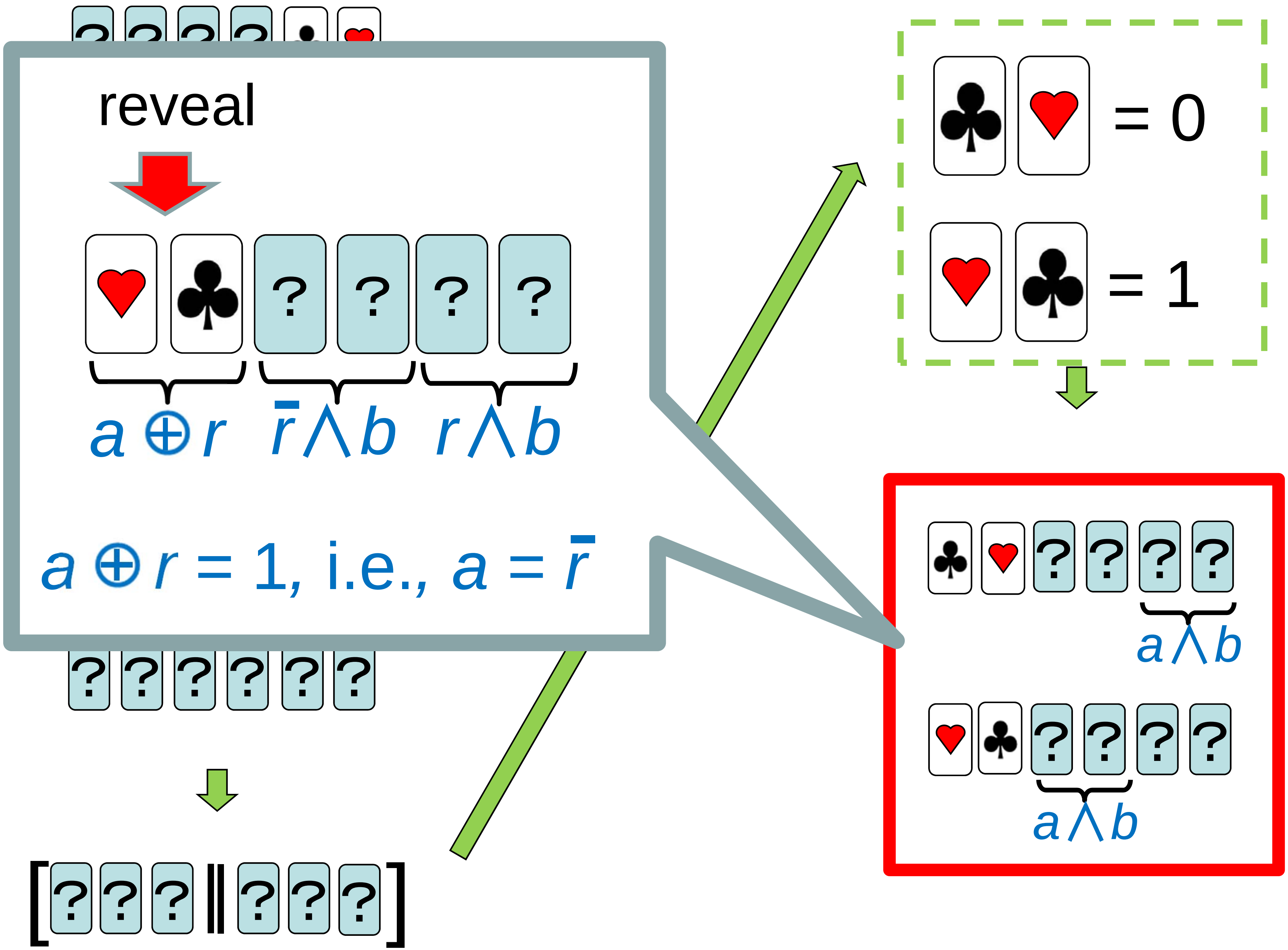
= 0
= 1
reveal
a ⊕ r
r̄ ∧ b
r ∧ b
a ∧ b
a ∧ b


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

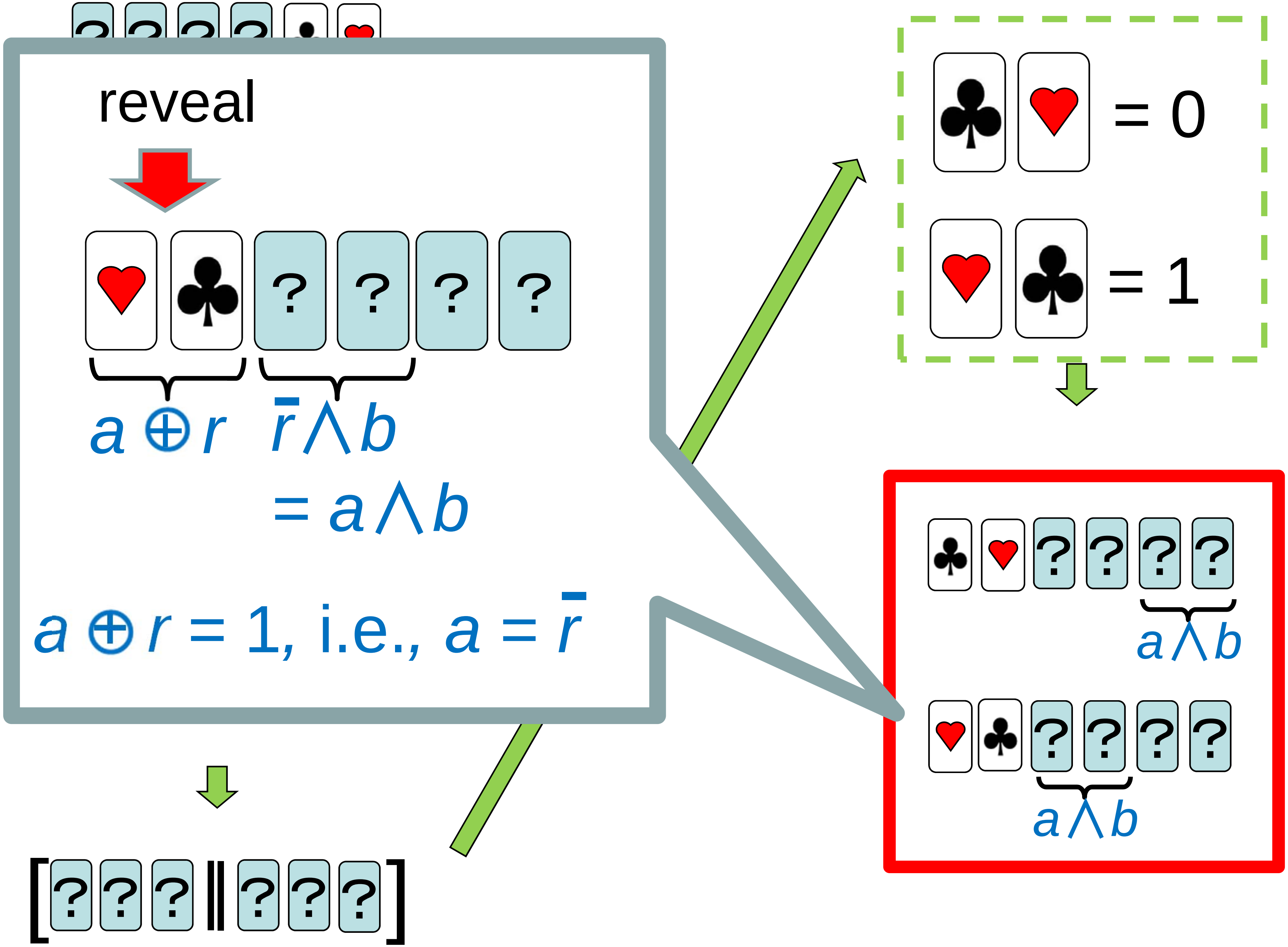


reveal
?
?
?
?
$a \oplus r$
$r \wedge b$
$= a \wedge b$
$a \oplus r = 0$, i.e., $a = r$
[??? ‖ ???]
= 0
= 1
????
$a \wedge b$
????
$a \wedge b$

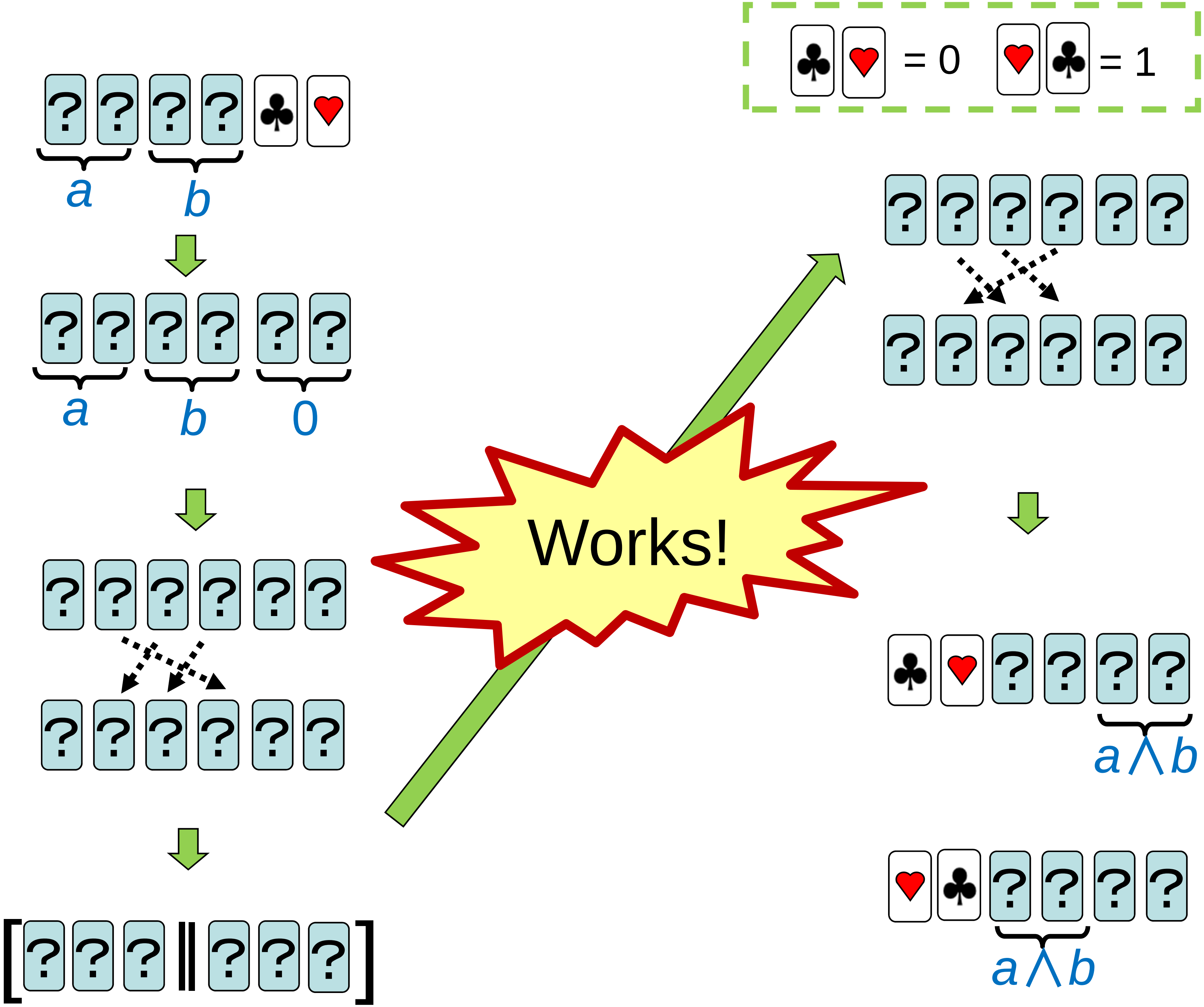

No information about a leaks because r is random.
reveal
a ⊕ r
r ∧ b
= a ∧ b
a ⊕ r = 0, i.e., a = r
= 1
a ∧ b
a ∧ b



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

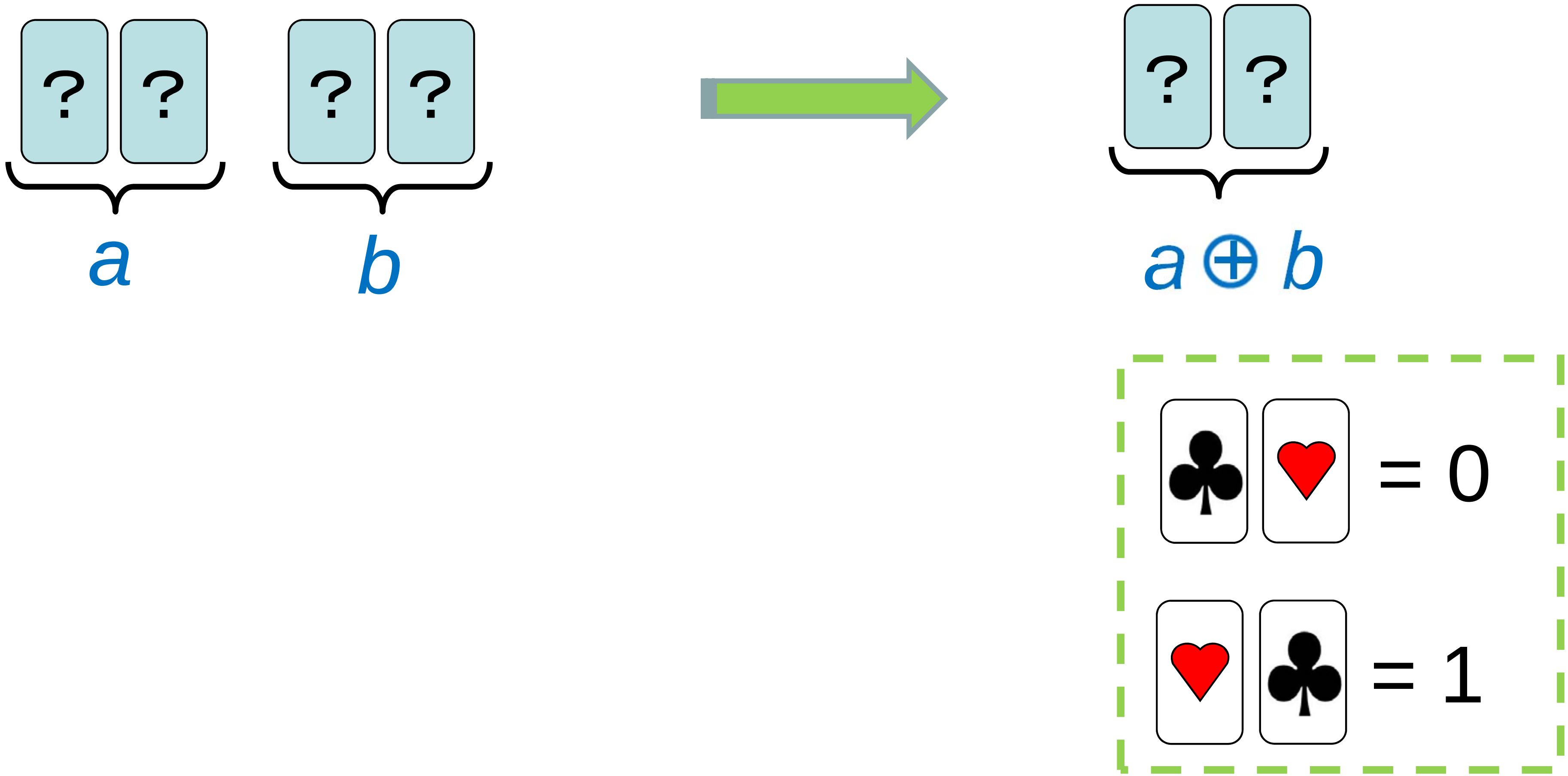


reveal
$a \oplus r$
$\bar{r} \wedge b$
$= a \wedge b$
$a \oplus r = 1$, i.e., $a = \bar{r}$
= 0
= 1
$a \wedge b$
$a \wedge b$

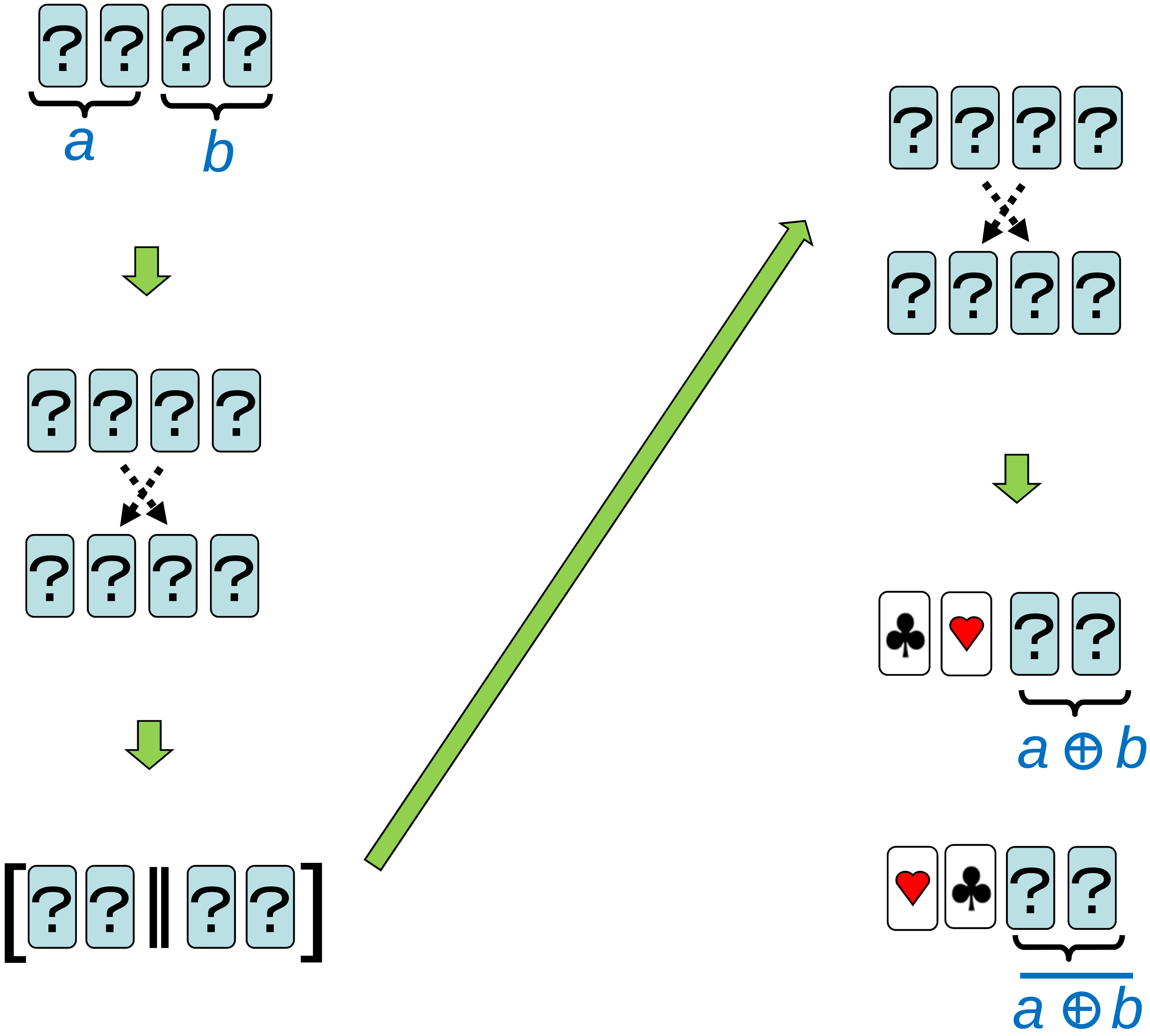

= 0
= 1
a
b
a
b
0
Works!
a∧b
a∧b

2.3 Four-Card XOR Protocol

Secure XOR can be done with 4 cards [6].



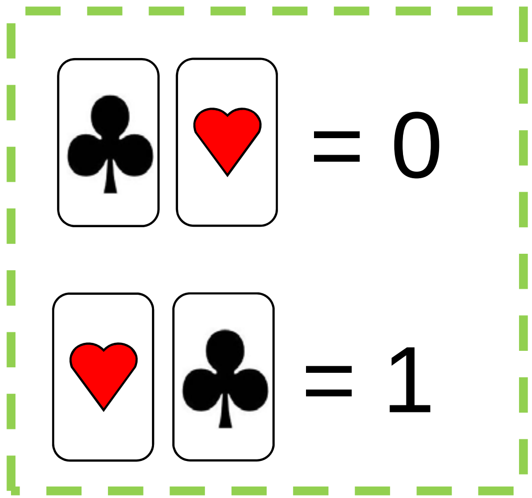
[6] T. Mizuki and H. Sone, Six-Card Secure AND and Four-Card Secure XOR, FAW 2009, LNCS 5598, pp. 358–369, 2009.



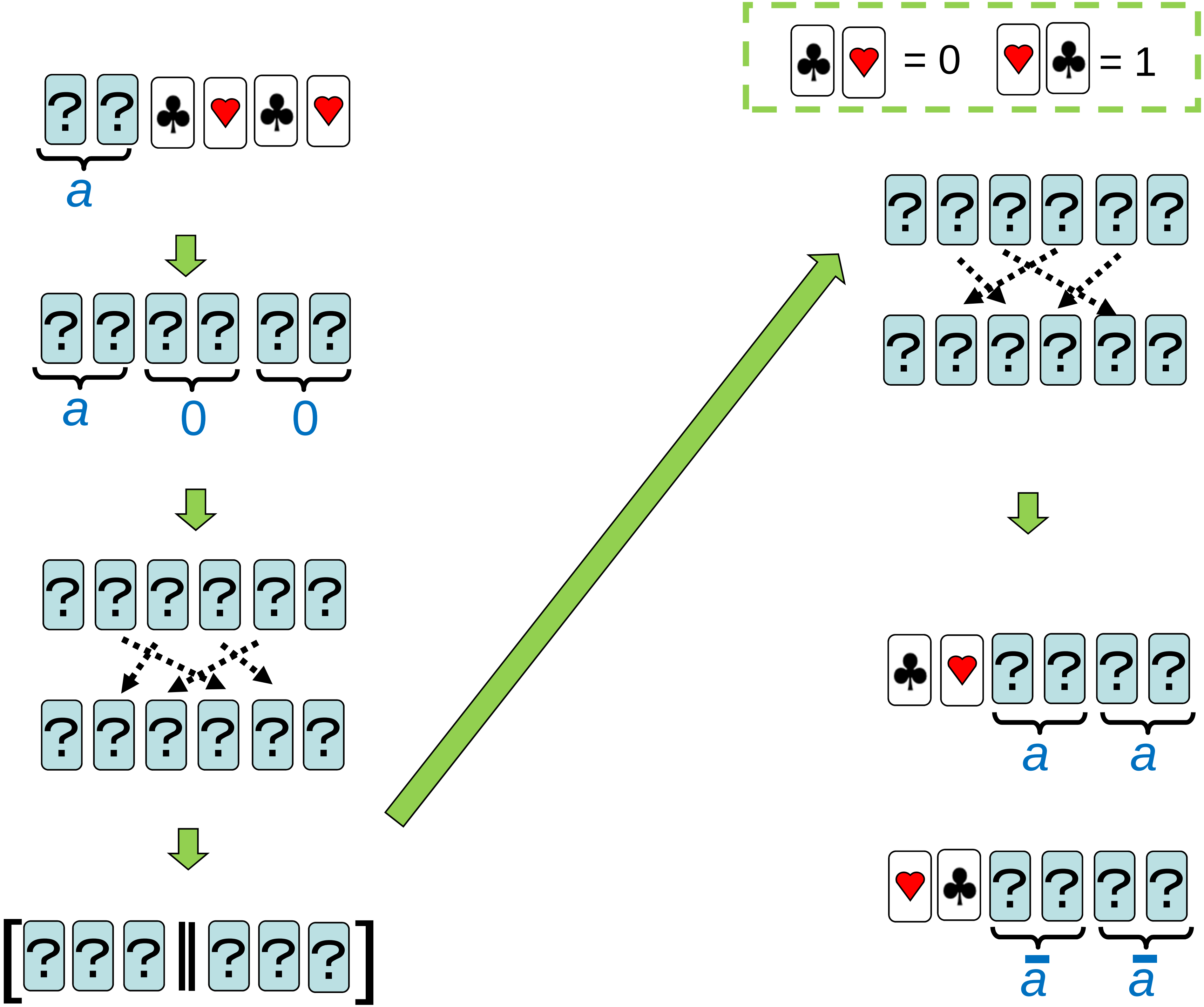
a
b
a ⊕ b
$\overline{a \oplus b}$

2.4 Copy Protocol with a Random Bisection Cut

Making a copy can be done with 4 additional cards [6].



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

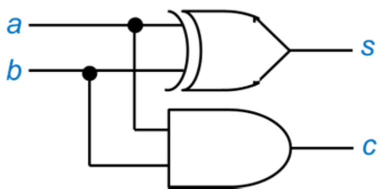
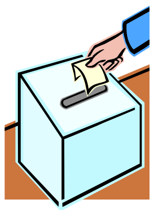

= 0
= 1
a
a
0
0
a
a
ā
ā

Contents

Outline of our results

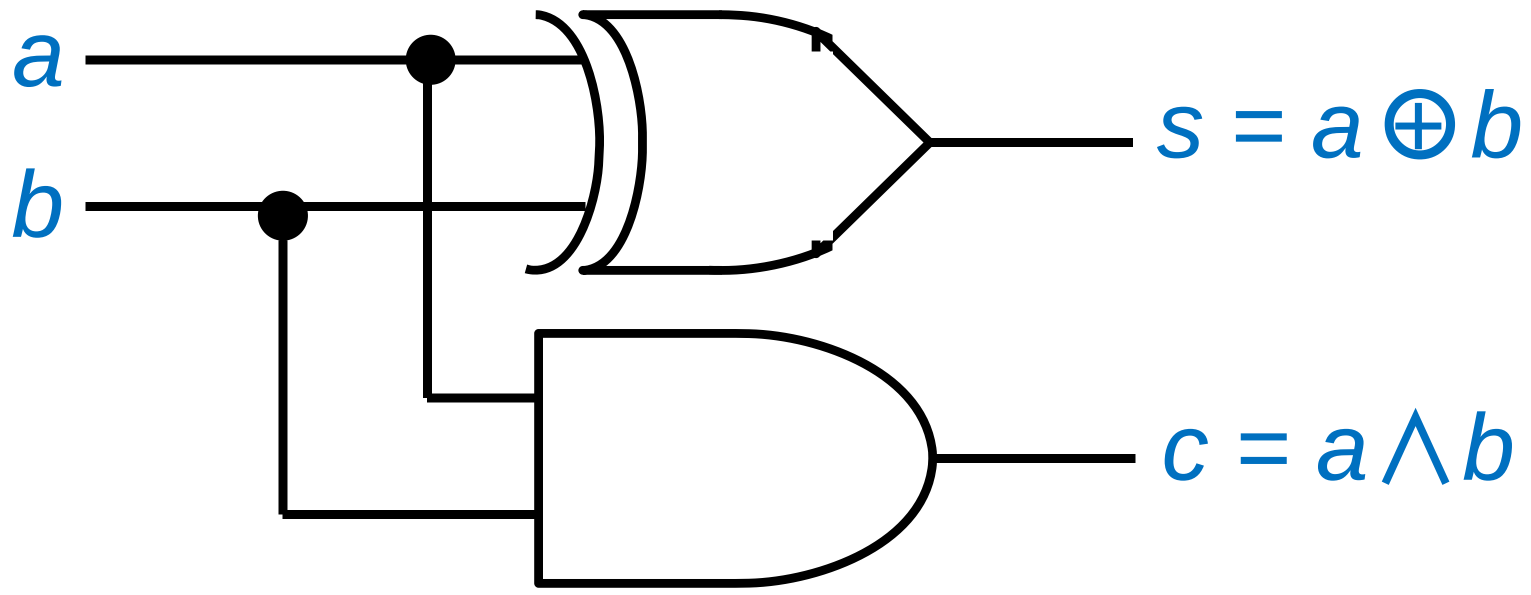
Using existing AND/XOR/COPY protocols

Devising a tailor-made half adder

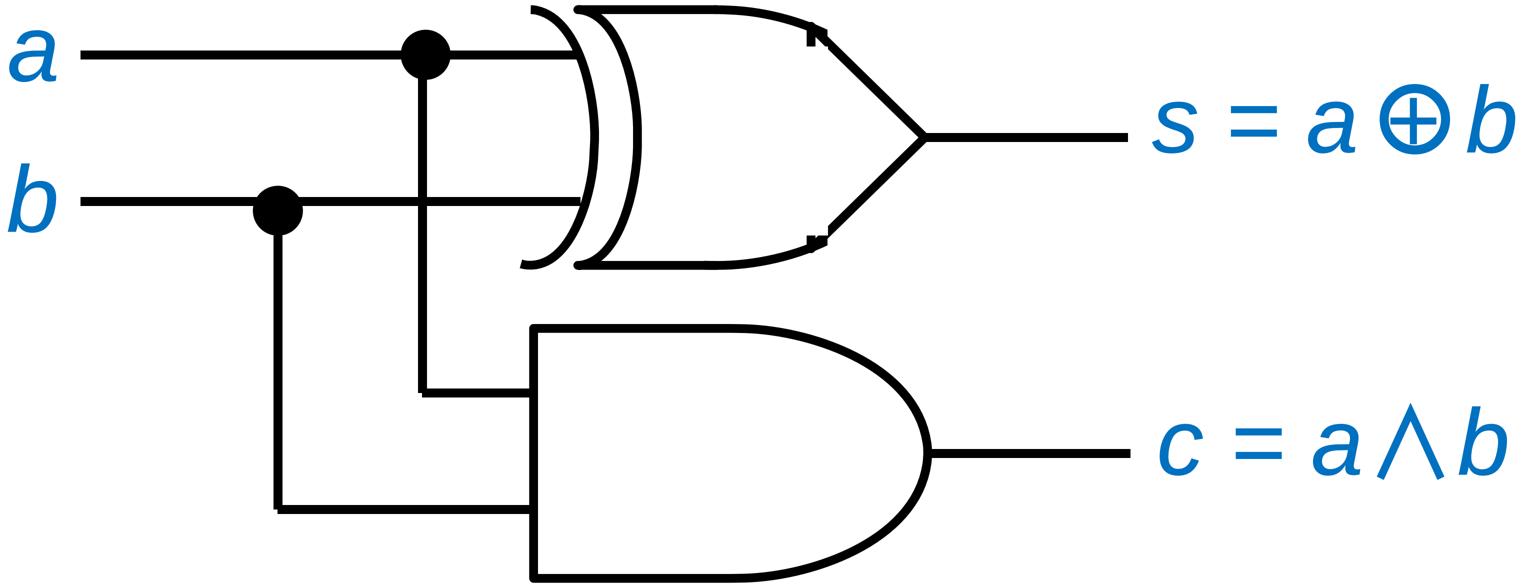
Half adder 	10	8
Voting 	$2\lceil \log n \rceil + 8$	$2\lceil \log n \rceil + 6$

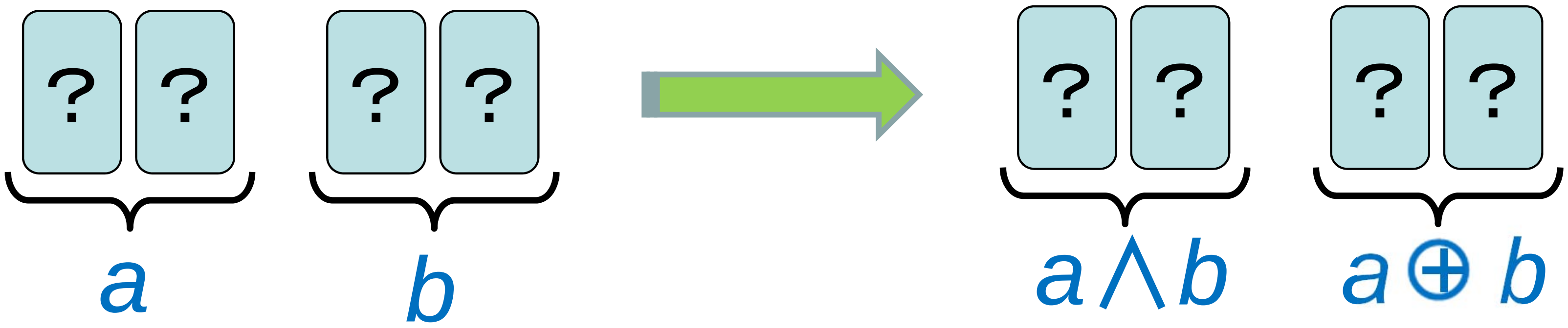
[# of cards]

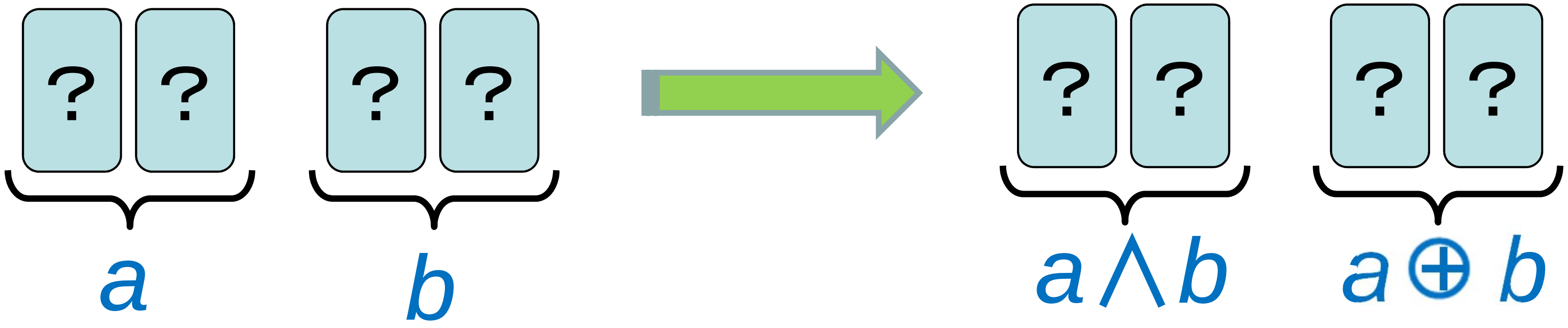
Half adder



Half adder



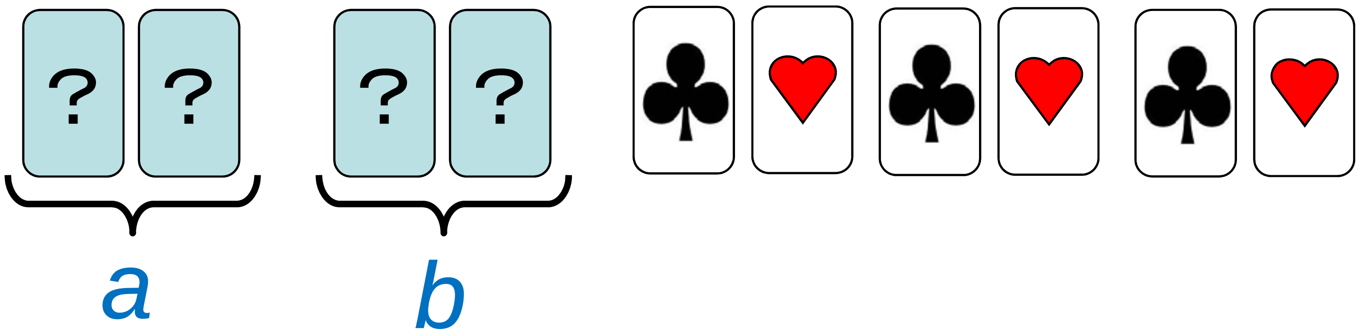




Remember that

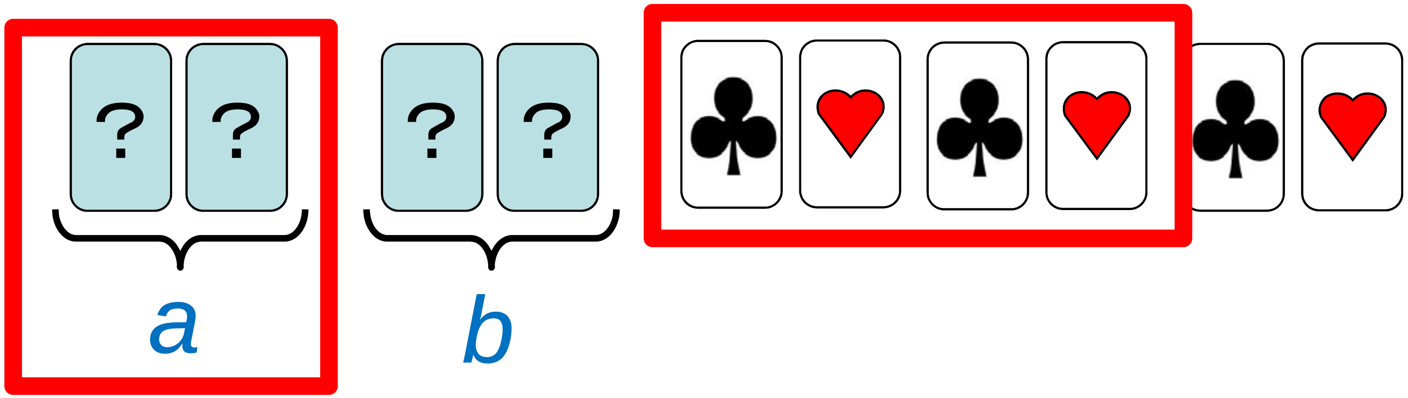
- ✓ AND can be done with 6 cards;
- ✓ XOR can be done with 4 cards;
- ✓ COPY can be done with 4 additional cards.

- ✓ AND can be done with 6 cards;
- ✓ XOR can be done with 4 cards;
- ✓ COPY can be done with 4 additional cards.

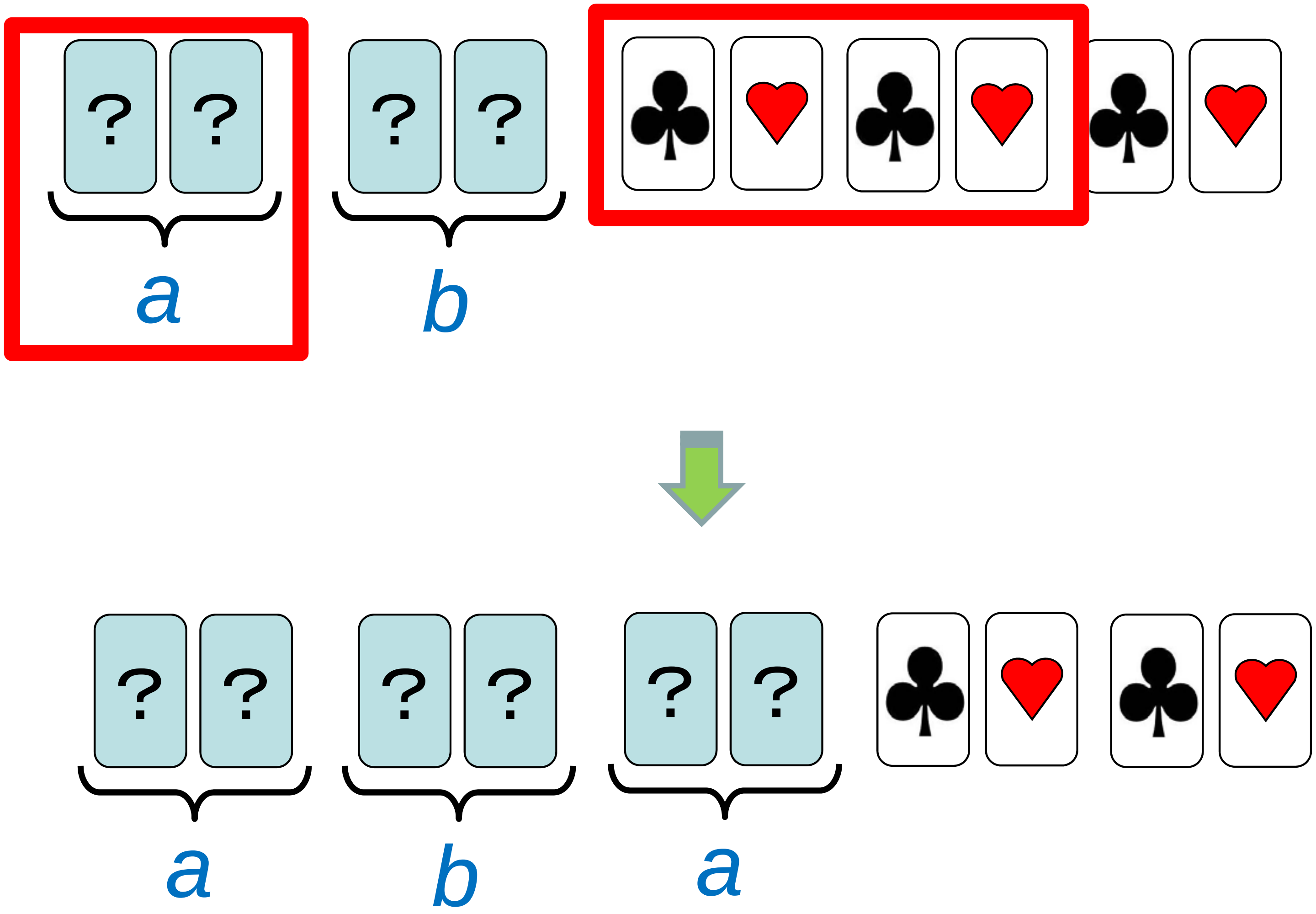


10 cards

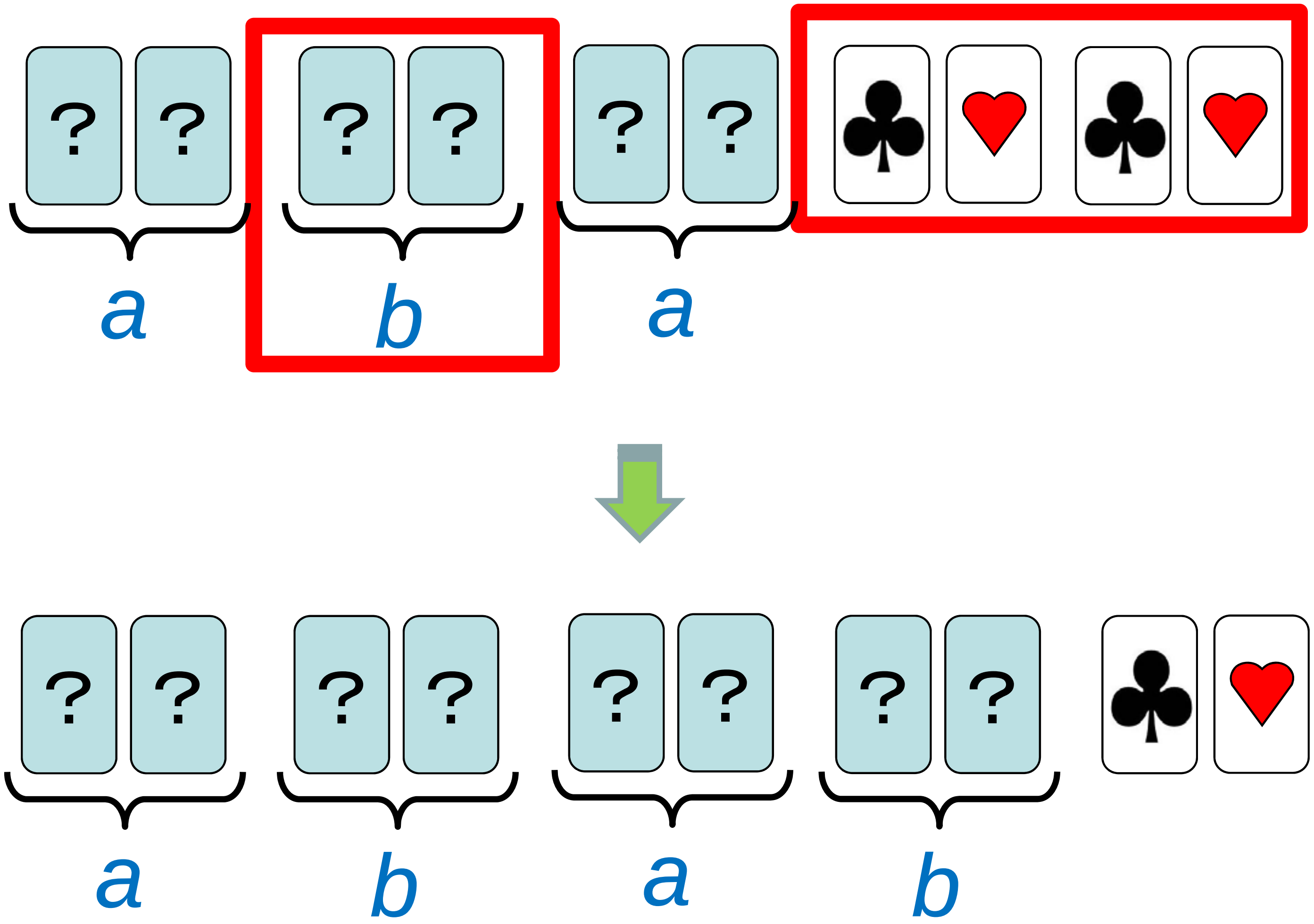
- ✓ AND can be done with 6 cards;
- ✓ XOR can be done with 4 cards;
- ✓ COPY can be done with 4 additional cards.



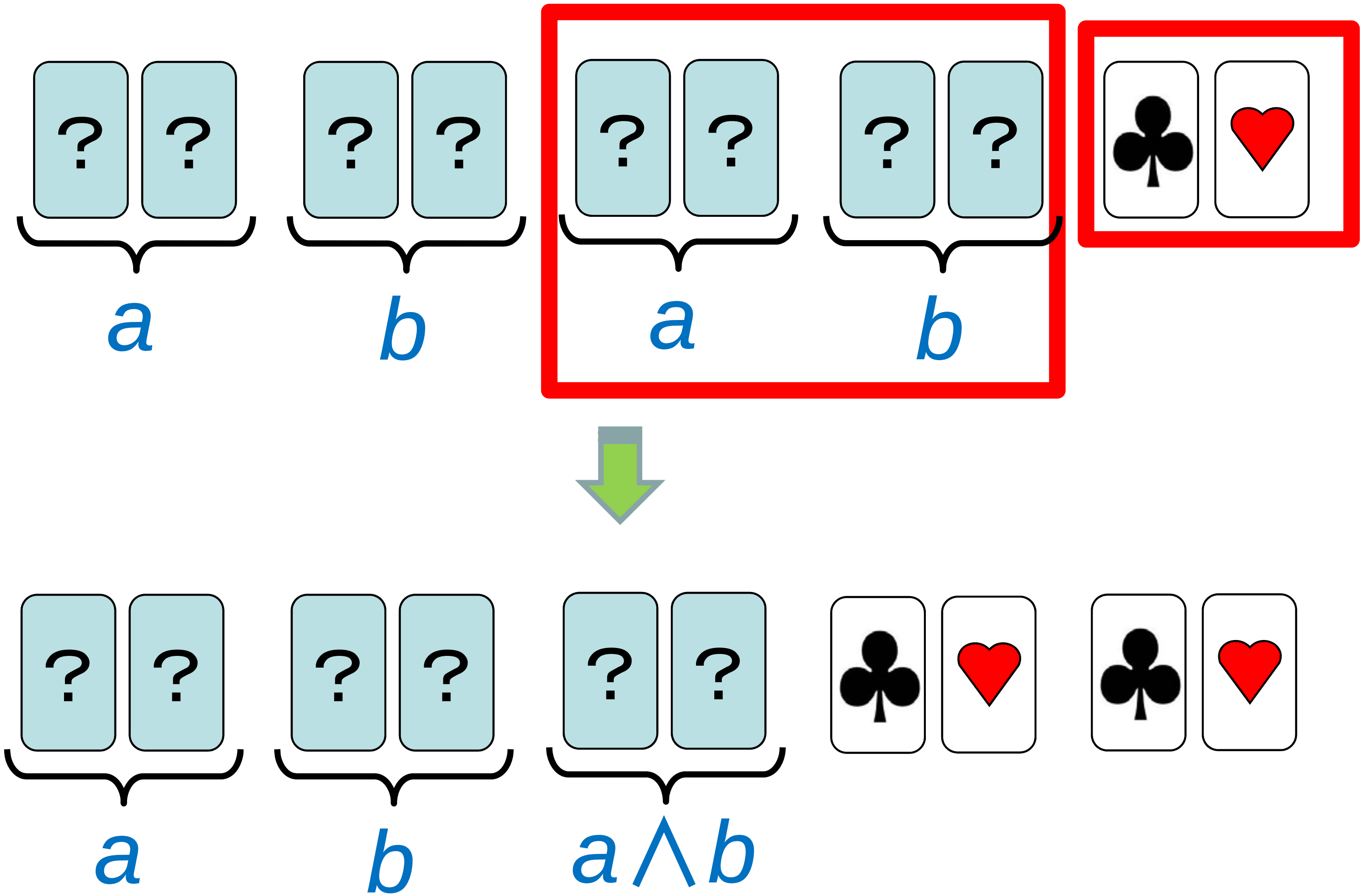
- ✓ AND can be done with 6 cards;
- ✓ XOR can be done with 4 cards;
- ✓ COPY can be done with 4 additional cards.



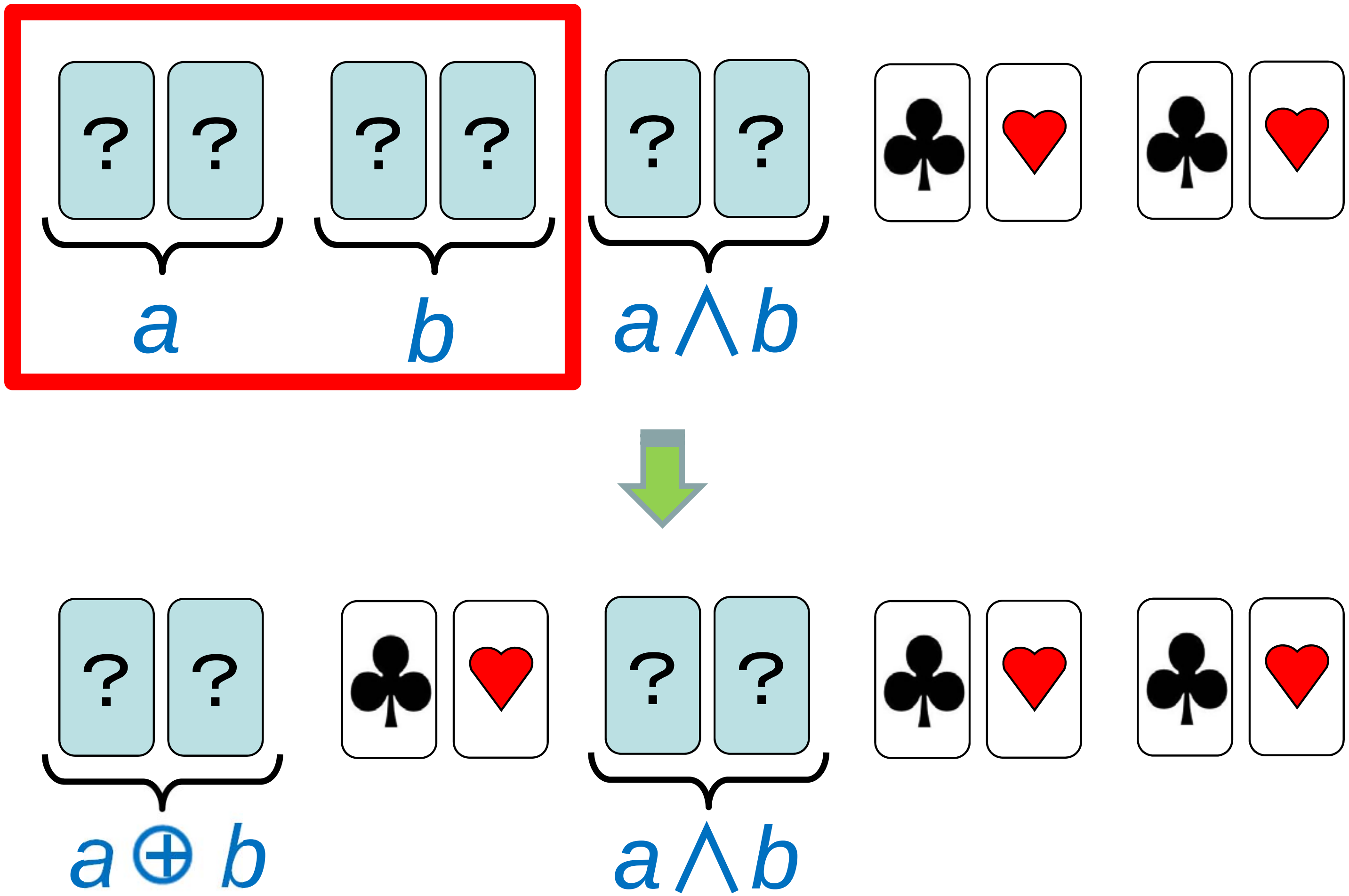
- ✓ AND can be done with 6 cards;
- ✓ XOR can be done with 4 cards;
- ✓ COPY can be done with 4 additional cards.



- ✓ AND can be done with 6 cards;
- ✓ XOR can be done with 4 cards;
- ✓ COPY can be done with 4 additional cards.



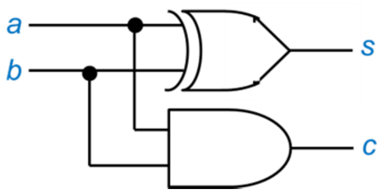
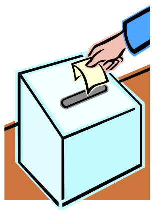
- ✓ AND can be done with 6 cards;
- ✓ XOR can be done with 4 cards;
- ✓ COPY can be done with 4 additional cards.



Outline of our results

Using existing AND/XOR/COPY protocols

Devising a tailor-made half adder

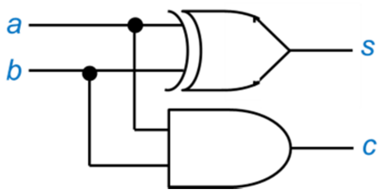
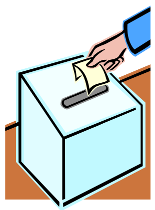
Half adder 	10	8
Voting 	$2\lceil \log n \rceil + 8$	$2\lceil \log n \rceil + 6$

[# of cards]

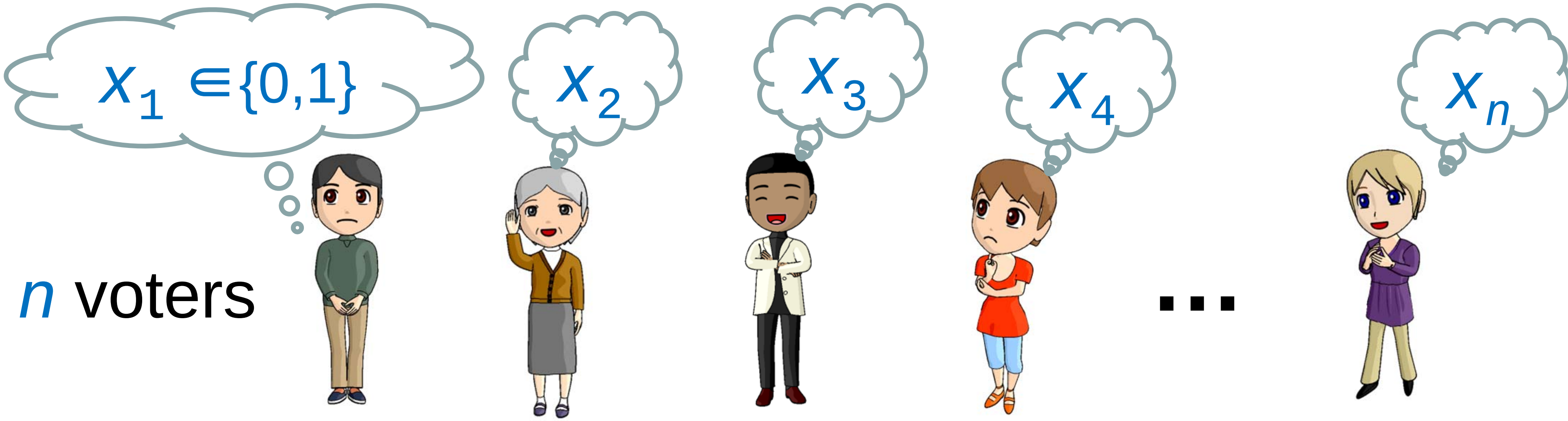
Outline of our results

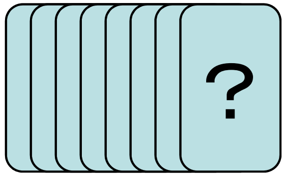
Using existing AND/XOR/COPY protocols

Devising a tailor-made half adder

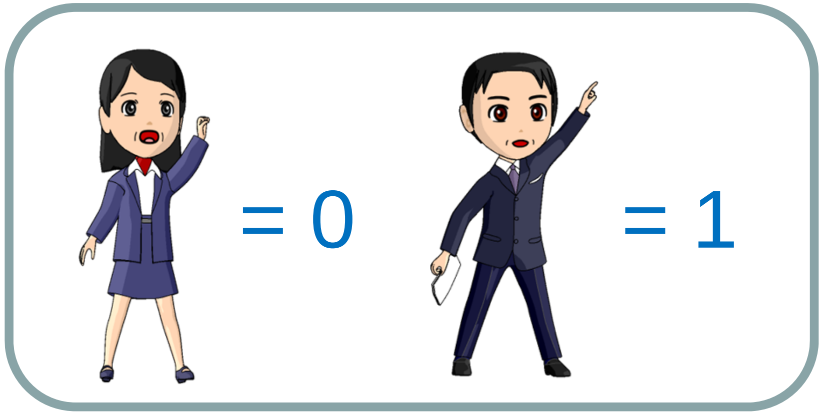
Half adder 	10 ✓	8
Voting 	$2\lceil \log n \rceil + 8$	$2\lceil \log n \rceil + 6$

[# of cards]

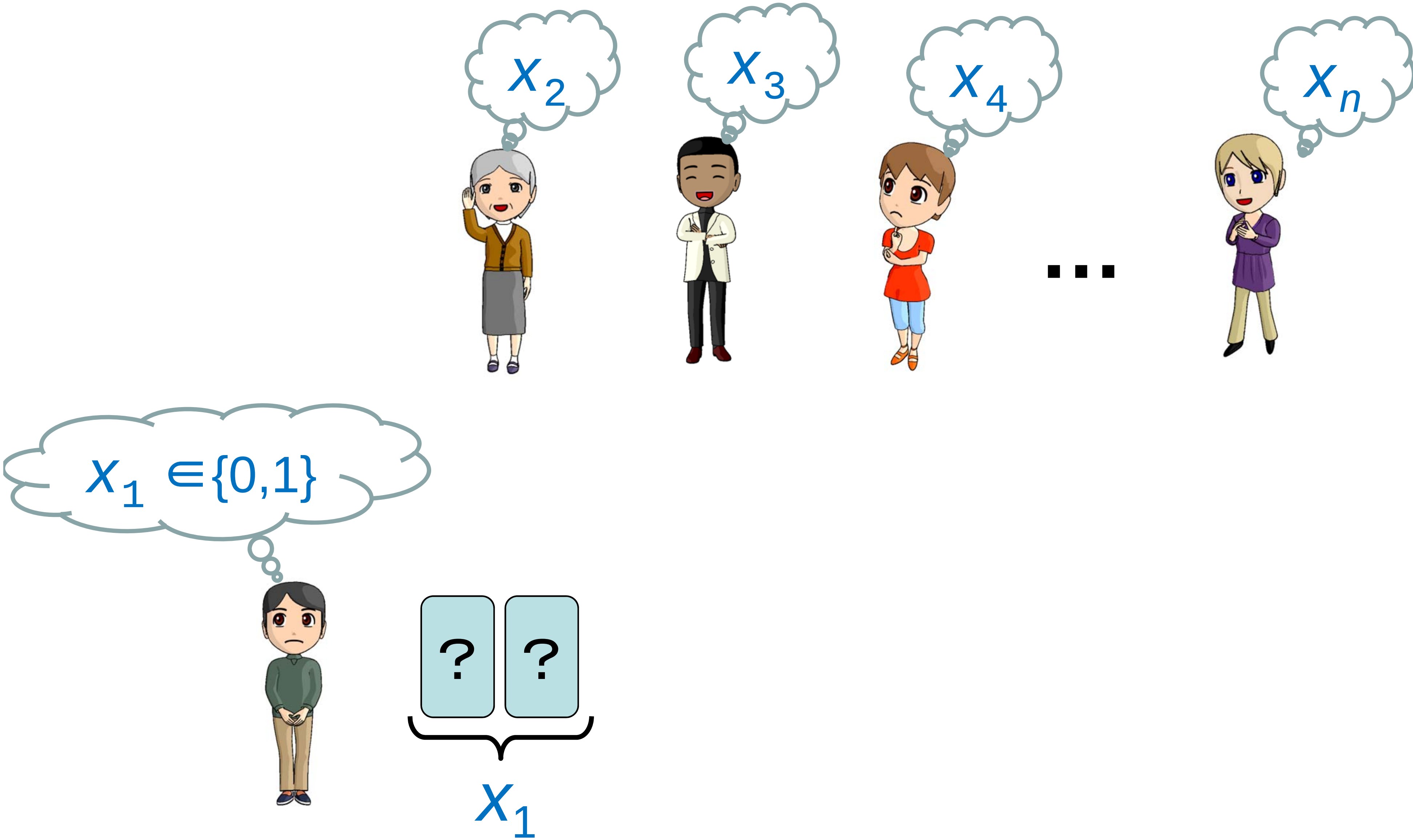


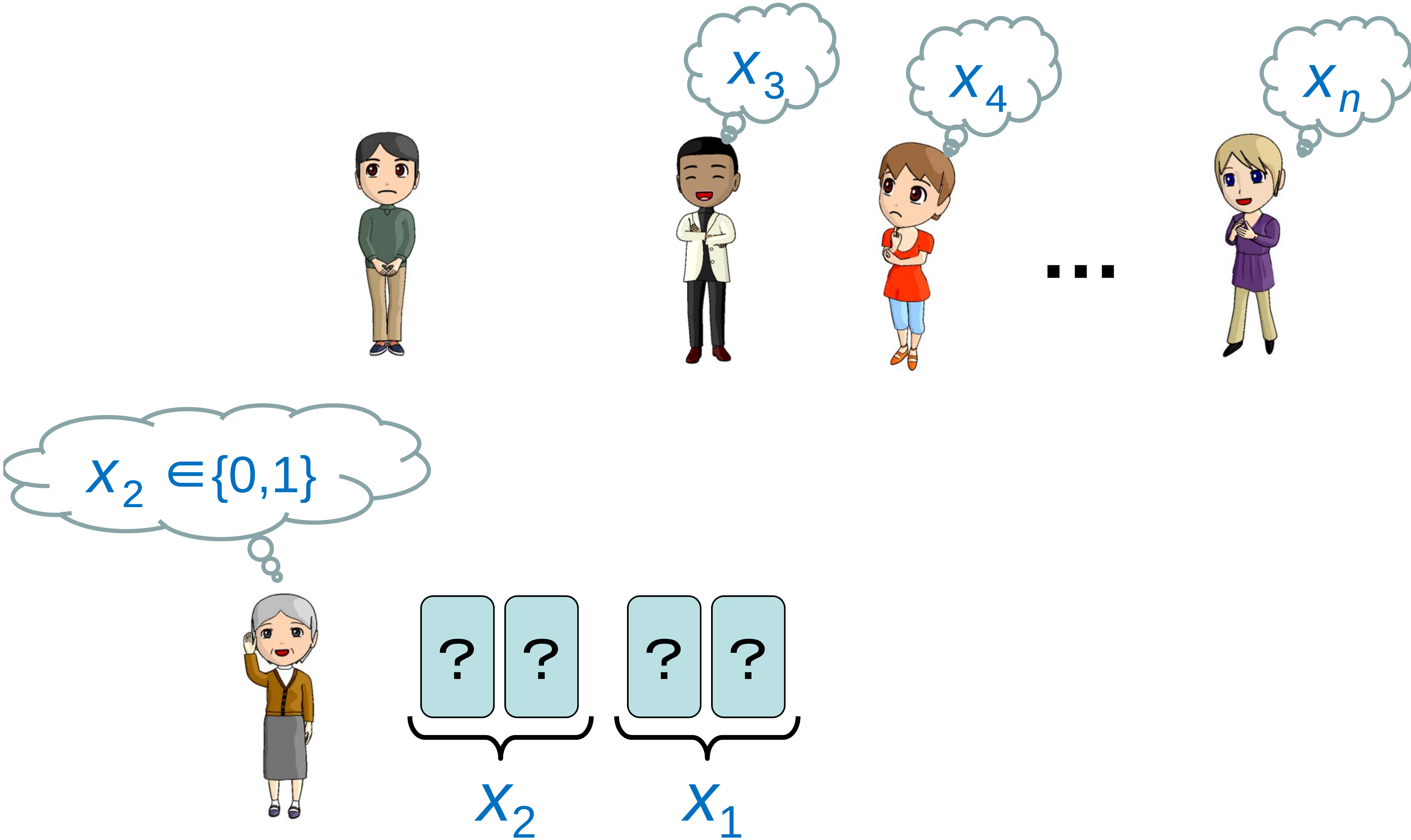


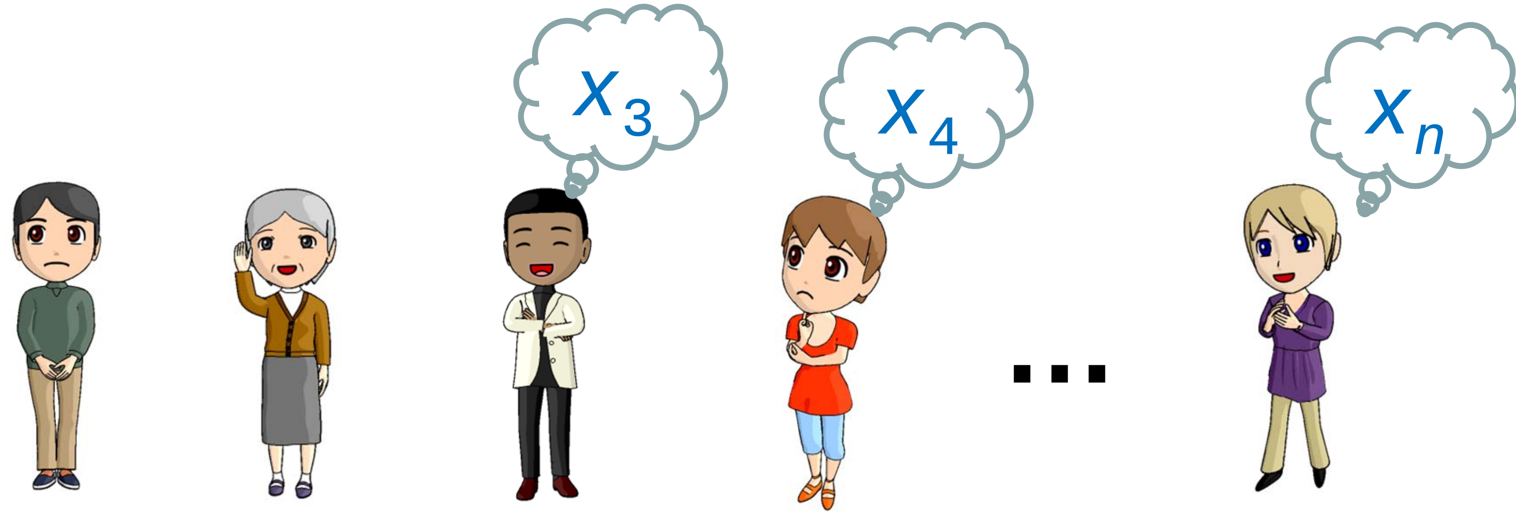
$2\lceil \log n \rceil + 8$ cards

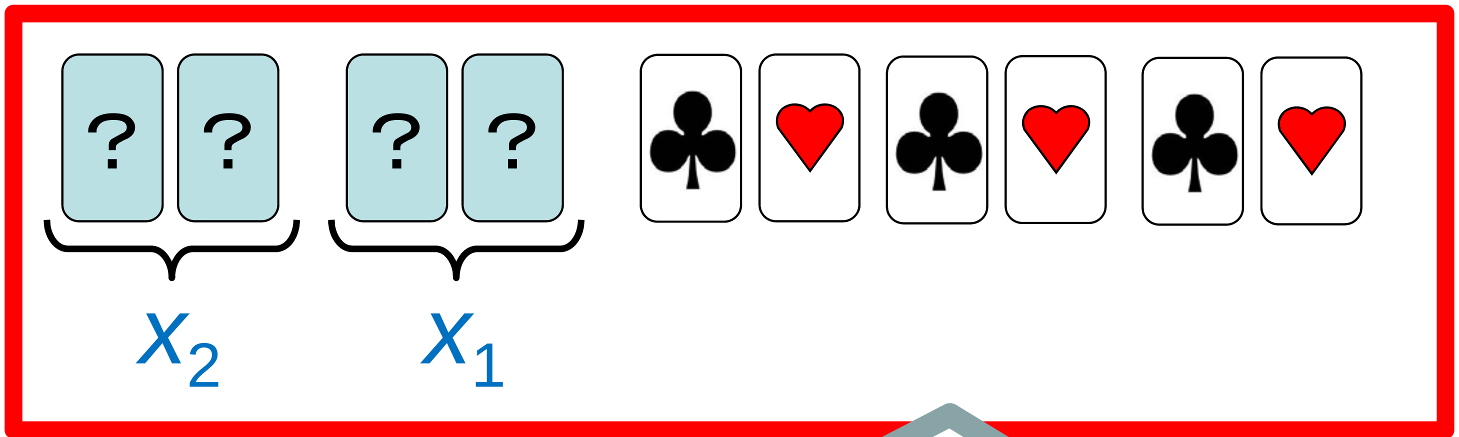


encoding for candidates

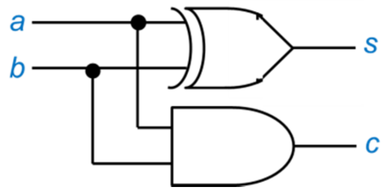

x_2
x_3
x_4
…
x_n
$x_1 \in \{0,1\}$
?
?
x_1

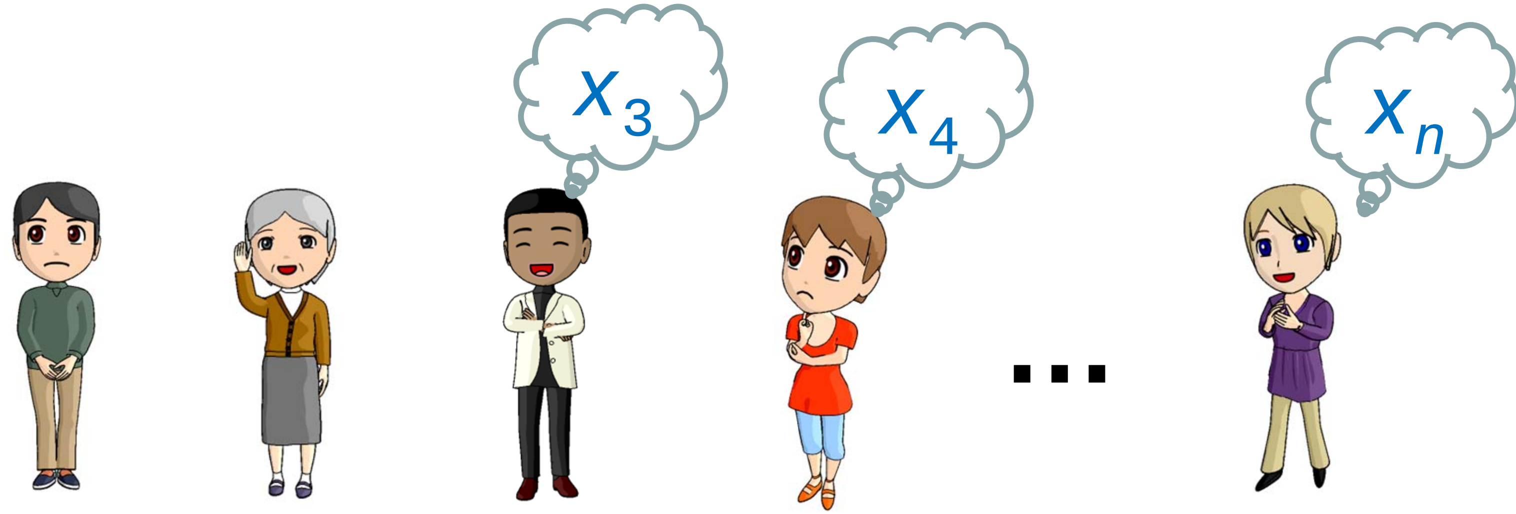

x_3
x_4
x_n
…
$x_2 \in \{0,1\}$
?
?
?
?
x_2
x_1

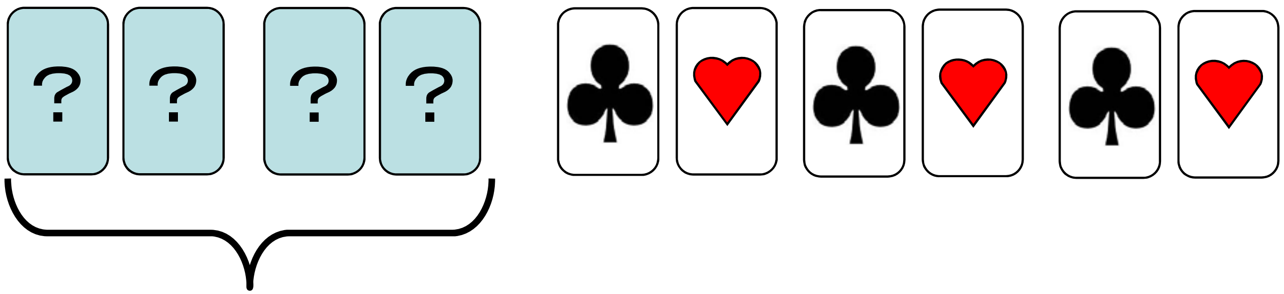




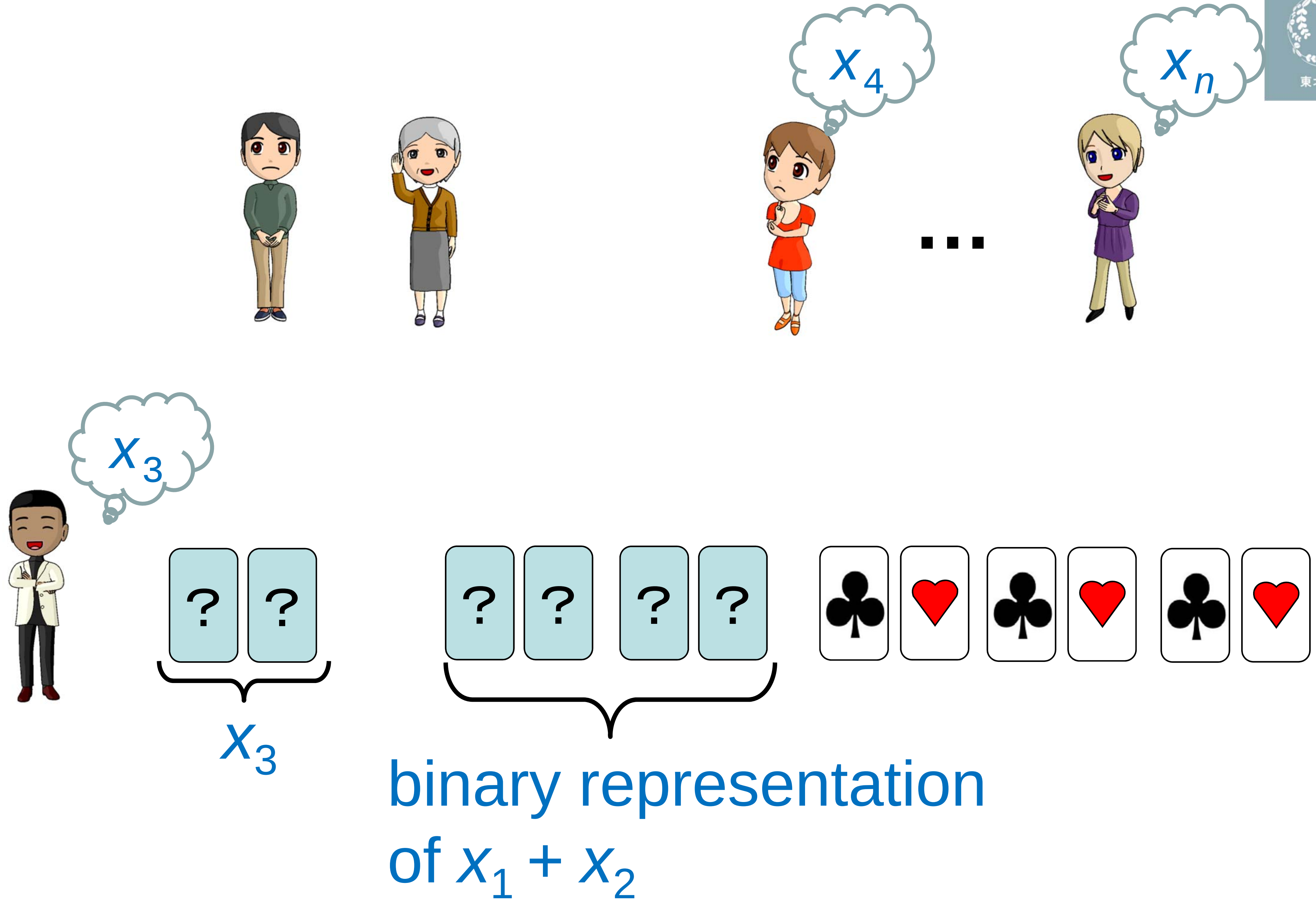
Apply a half adder

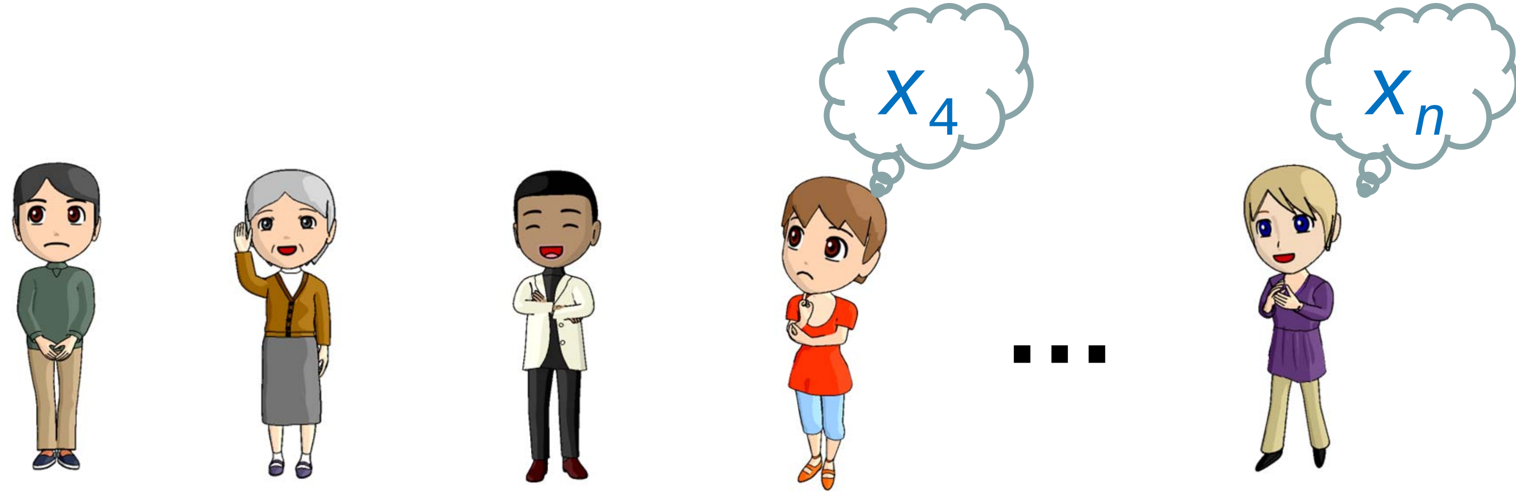


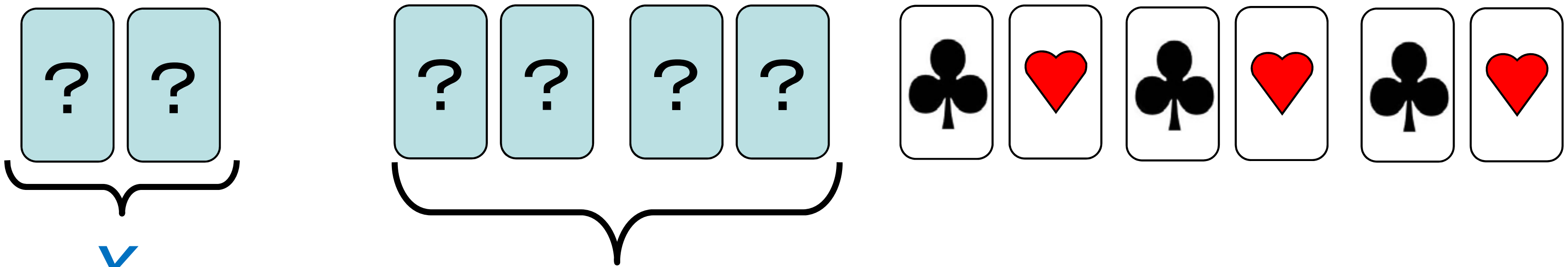




binary representation of $x_1 + x_2$


x_4
x_n
x_3
x_3
binary representation of $x_1 + x_2$

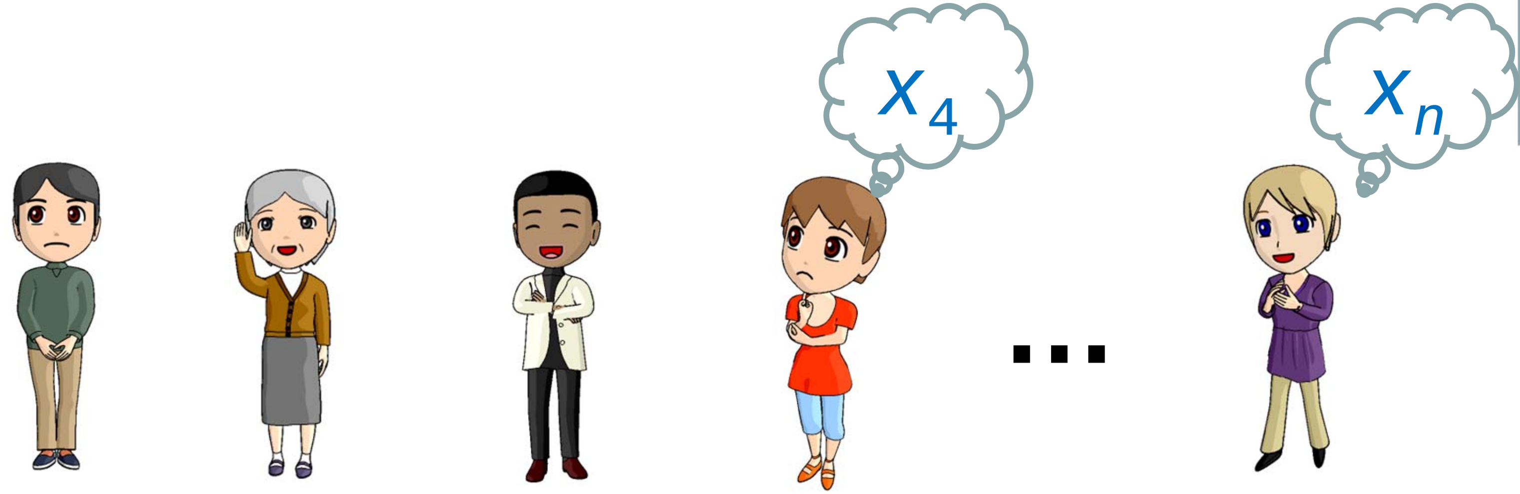


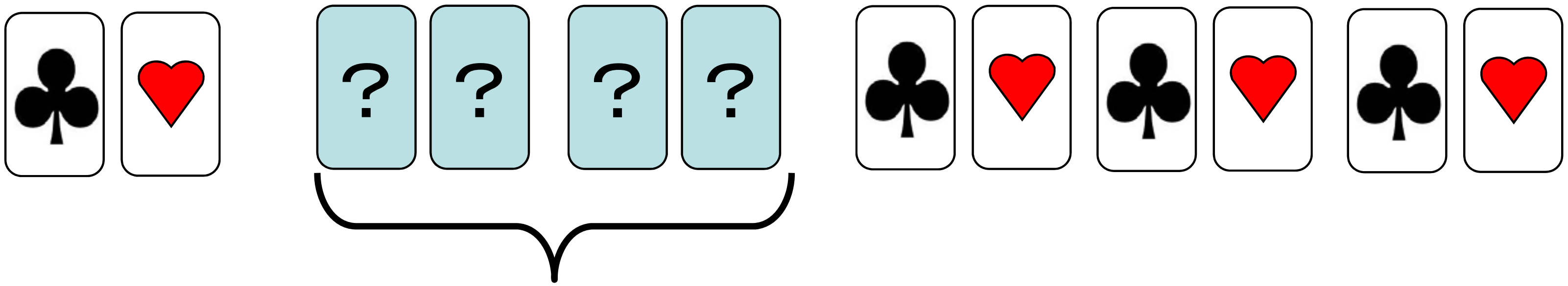


x_3

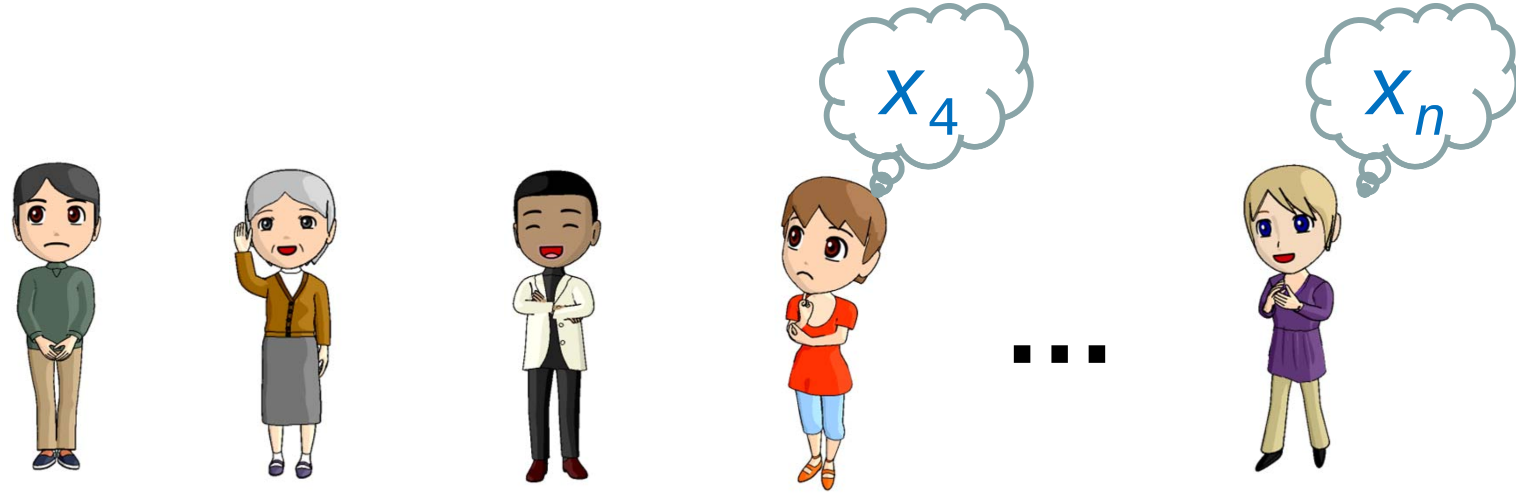
binary representation of $x_1 + x_2$

Apply a half adder (and XOR)





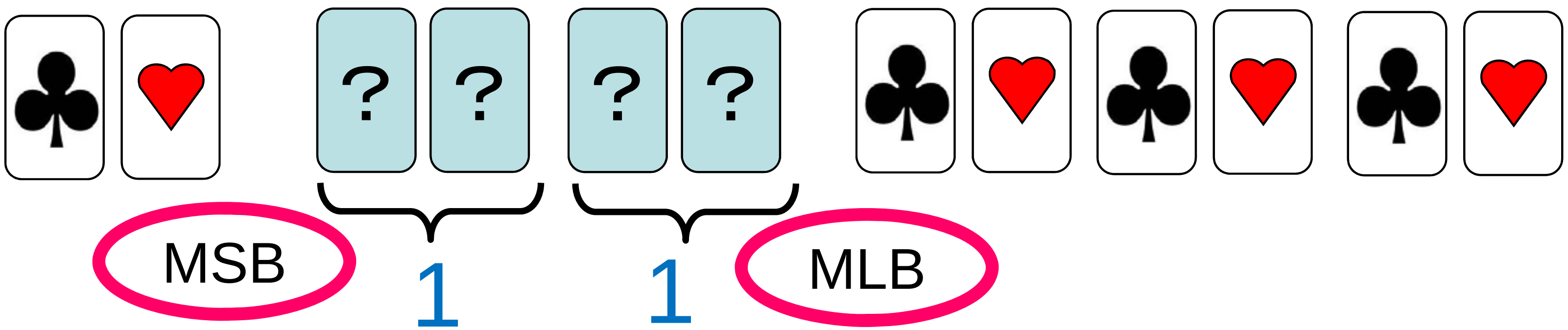
binary representation
of $x_1 + x_2 + x_3$



For example,

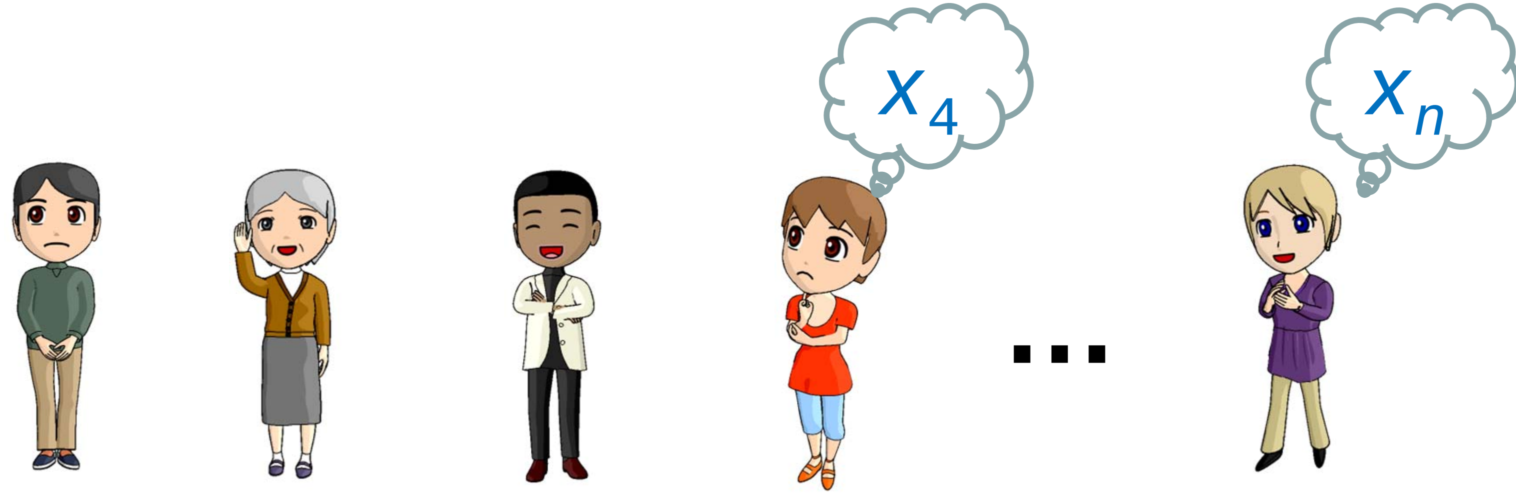
if $x_1 = x_2 = x_3 = 1$,

then

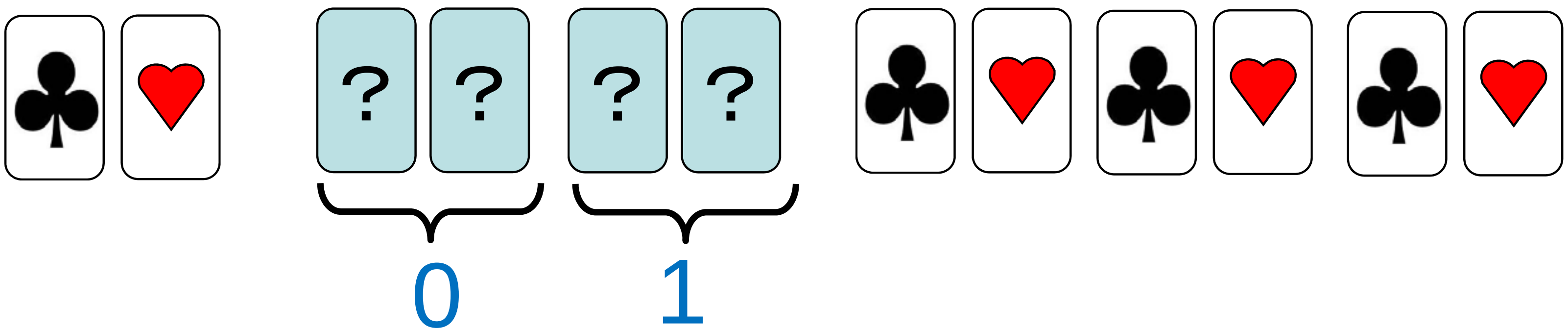


binary representation

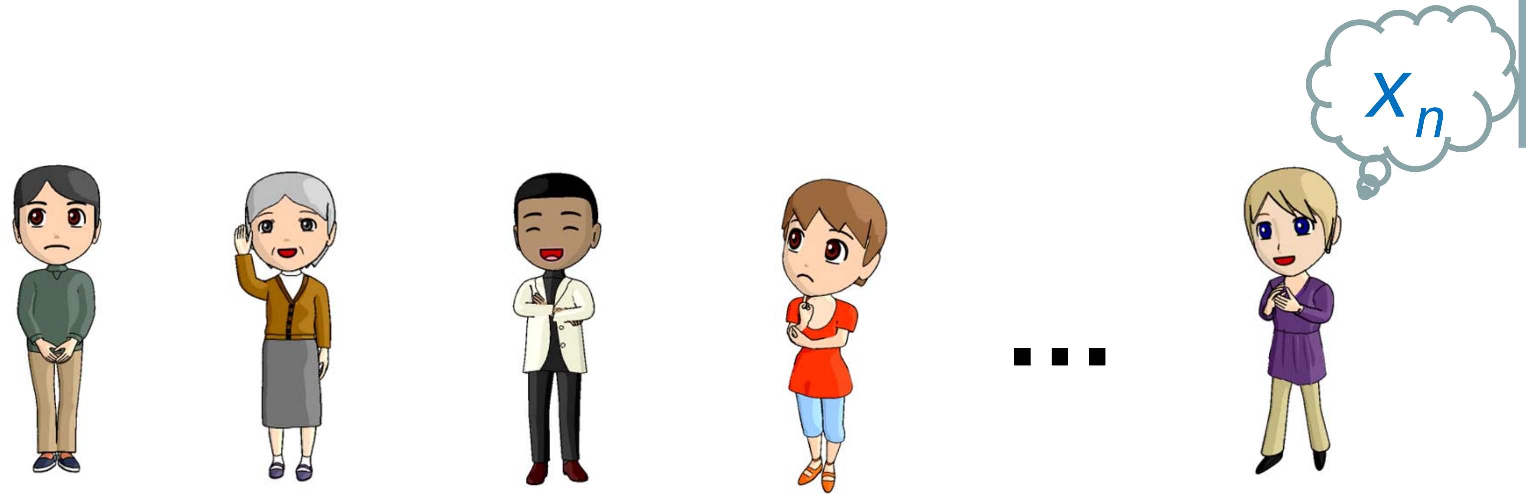
of $x_1 + x_2 + x_3$



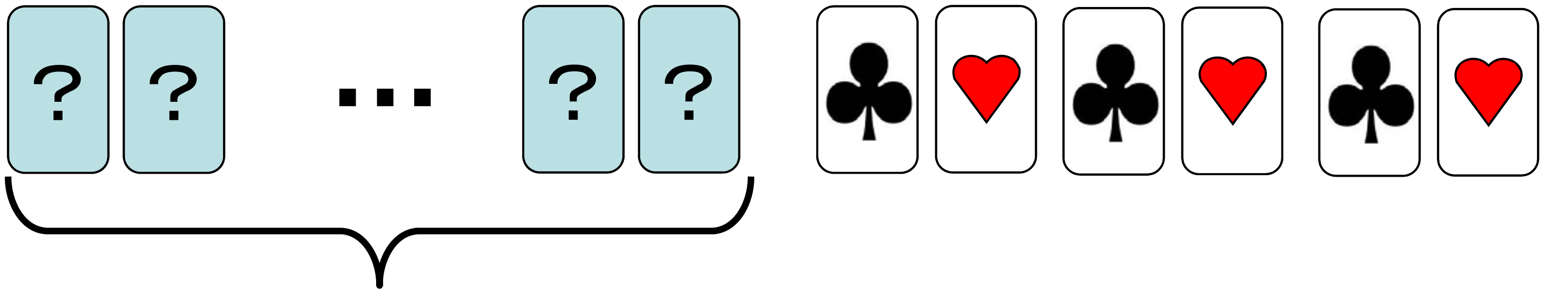
For example,
if $x_1 = 1$ and $x_2 = x_3 = 0$,
then



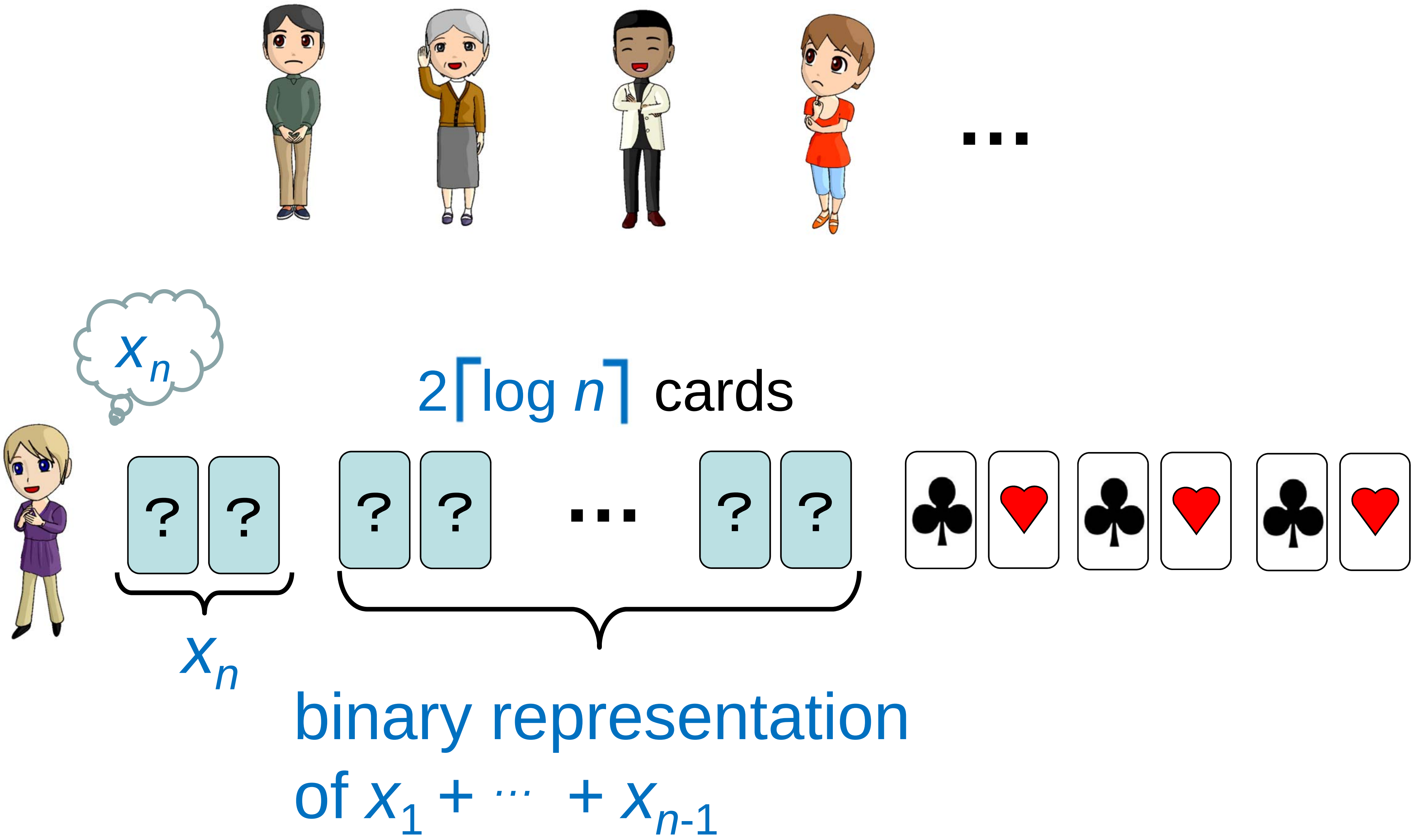
binary representation
of $x_1 + x_2 + x_3$

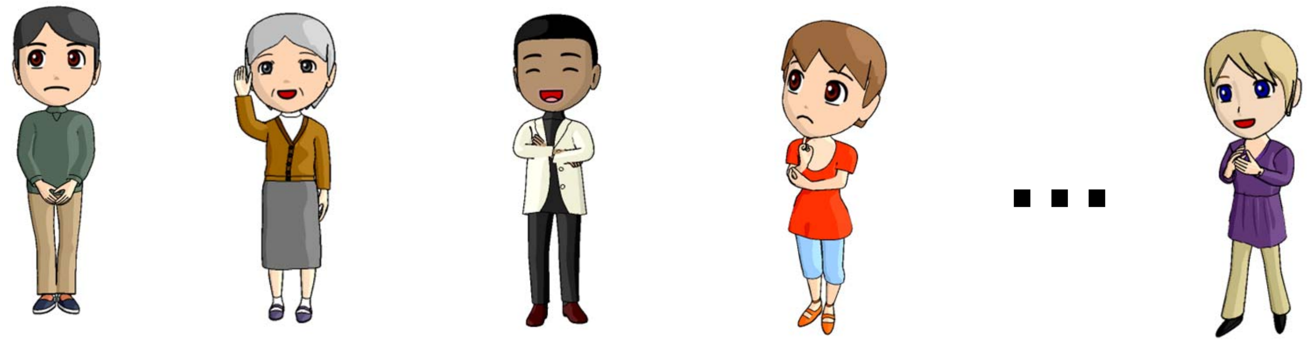


$2\lceil \log n \rceil$ cards

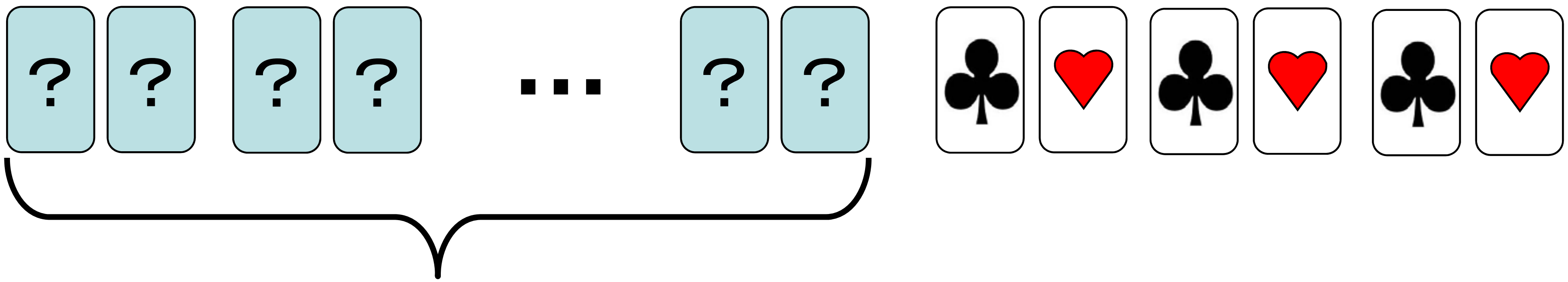


binary representation of $x_1 + \cdots + x_{n-1}$


x_n
$2\lceil \log n \rceil$ cards
?
?
?
?
?
?
x_n
binary representation
of $x_1 + \cdots + x_{n-1}$



$2\lceil \log n \rceil + 2$ (or $2\lceil \log n \rceil$) cards

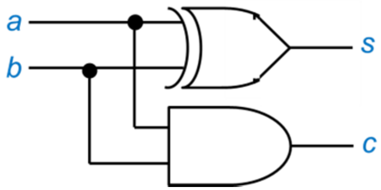
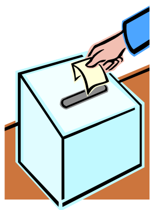


binary representation of $x_1 + \cdots + x_n$

Outline of our results

Using existing AND/XOR/COPY protocols

Devising a tailor-made half adder

Half adder 	10	8
Voting 	$2\lceil \log n \rceil + 8$	$2\lceil \log n \rceil + 6$

[# of cards]

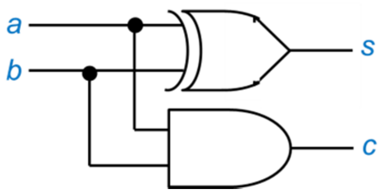
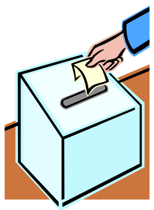
Contents

1. Introduction
2. Known Protocols
3. Voting with a Logarithmic Number of Cards
4. New Adder Protocols
5. Conclusion

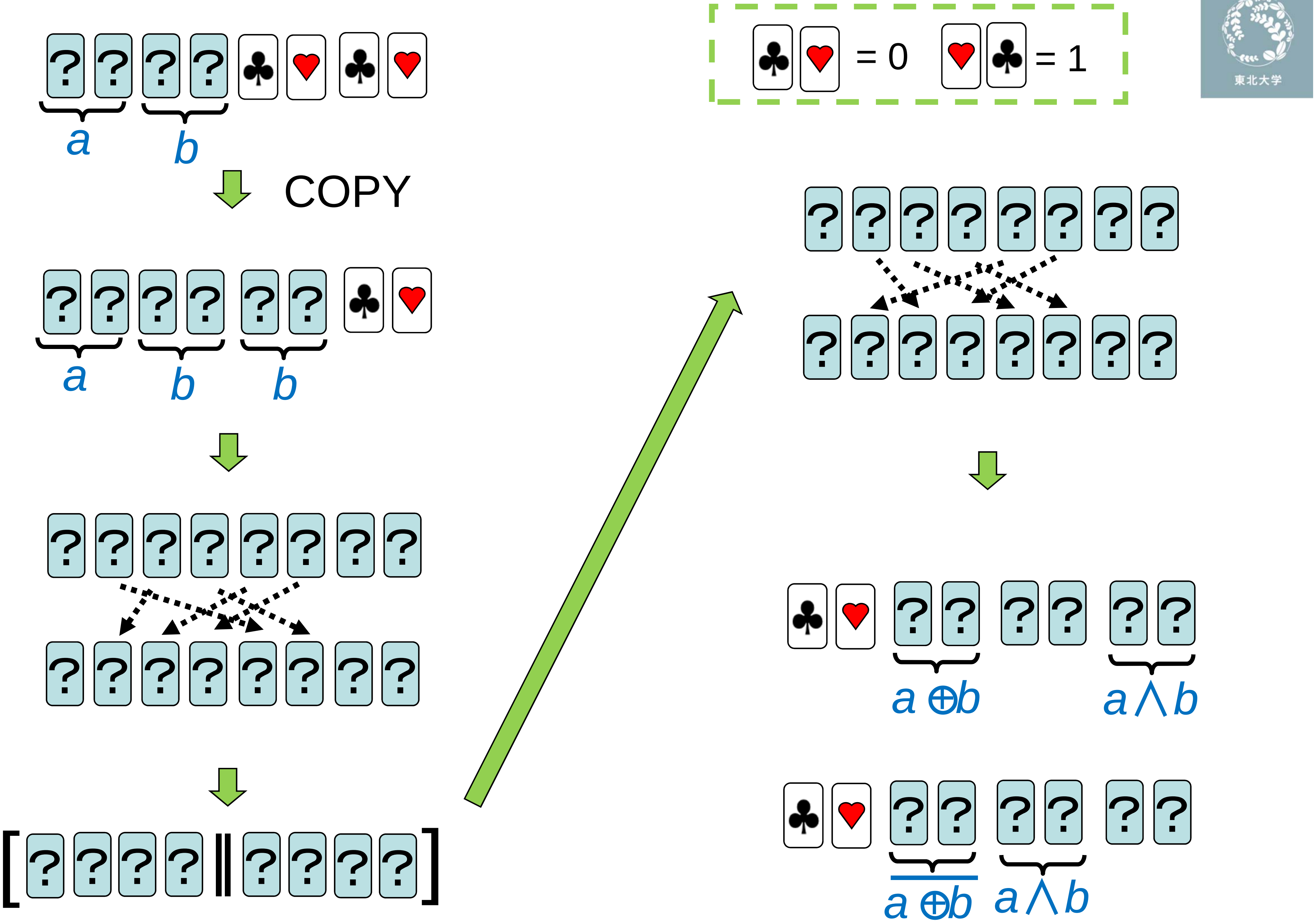
Outline of our results

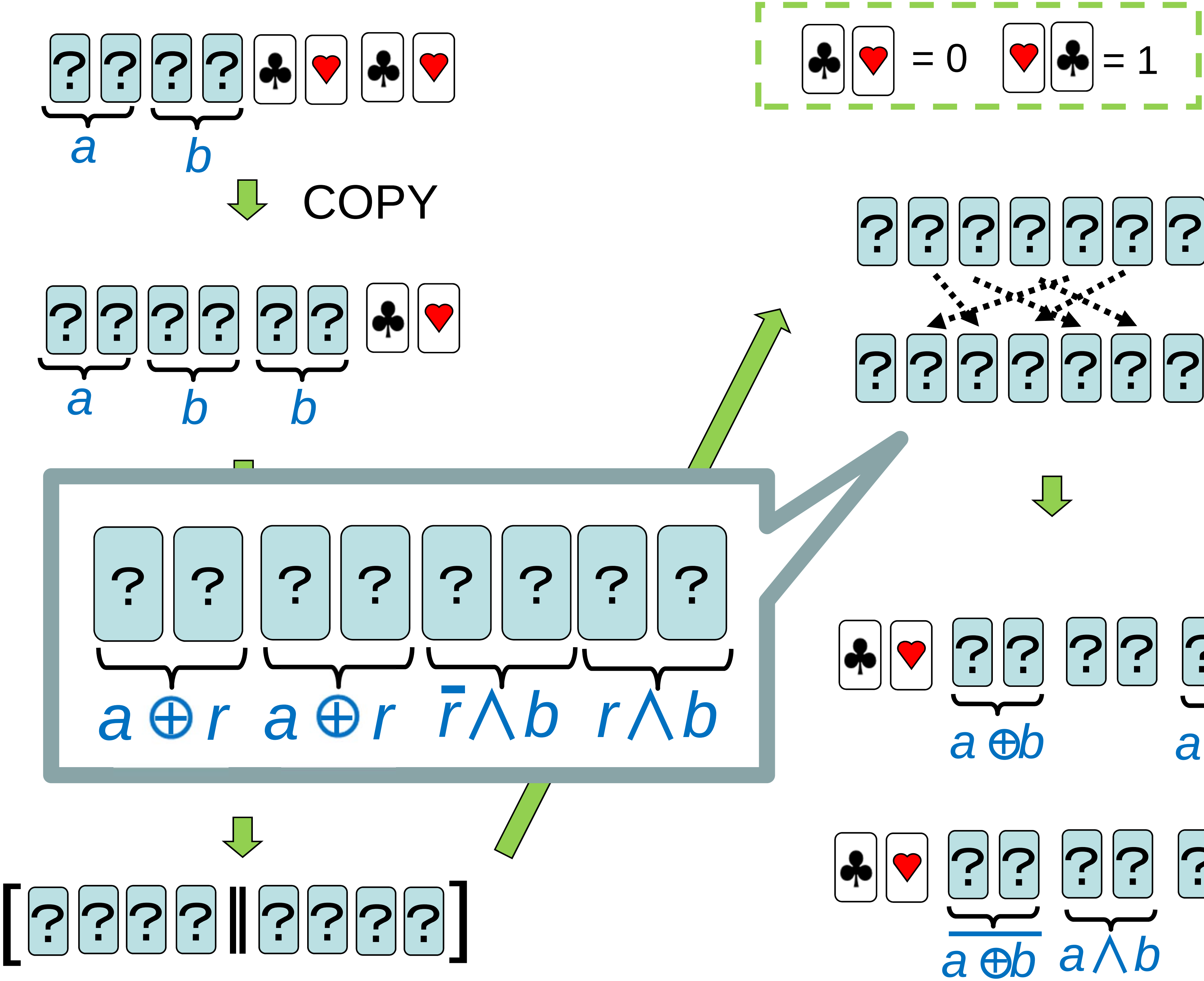
Using existing AND/XOR/COPY protocols

Devising a tailor-made half adder

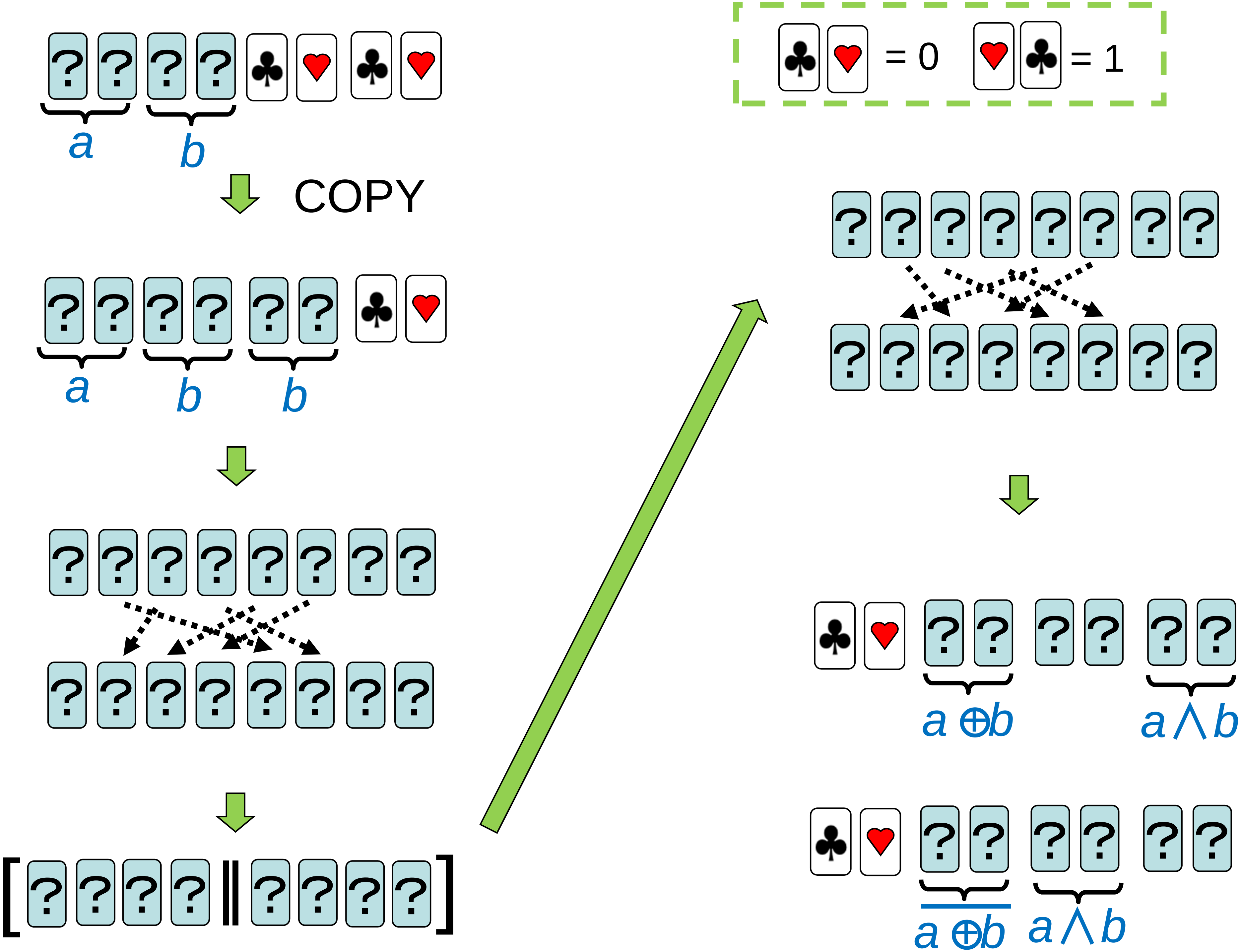
Half adder 	10	8
Voting 	$2\lceil \log n \rceil + 8$	$2\lceil \log n \rceil + 6$

[# of cards]


= 0
= 1
a
b
COPY
a
b
b
a ⊕b
a∧b
a ⊕b
a∧b
東北大学



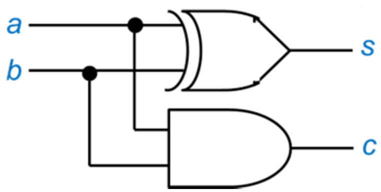
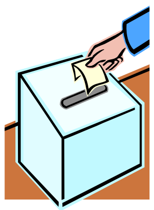
= 0
= 1
a
b
COPY
a
b
b
a ⊕ r
a ⊕ r
r̄ ∧ b
r ∧ b
a ⊕ b
a ∧ b
$\overline{a \oplus b}$
a ∧ b


= 0
= 1
a
b
COPY
a
b
b
a ⊕ b
a ∧ b
a ⊕ b
a ∧ b

Outline of our results

Using existing AND/XOR/COPY protocols

Devising a tailor-made half adder

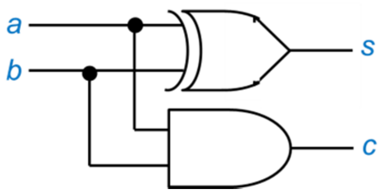
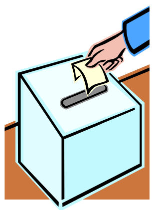
Half adder 	10	8
Voting 	$2\lceil \log n \rceil + 8$	$2\lceil \log n \rceil + 6$

[# of cards]

Outline of our results

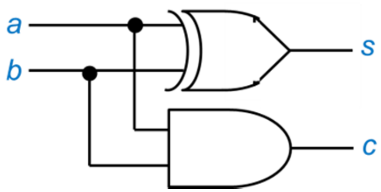
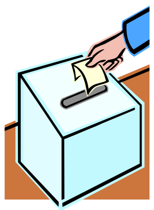
Using existing AND/XOR/COPY protocols

Devising a tailor-made half adder

Half adder 	10	8
Voting 	$2\lceil \log n \rceil + 8$	$2\lceil \log n \rceil + 6$

[# of cards]

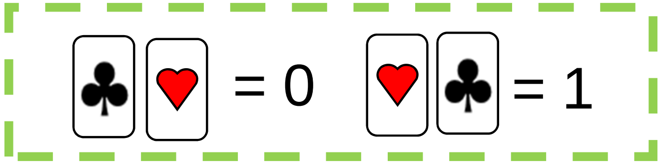
Outline of our results

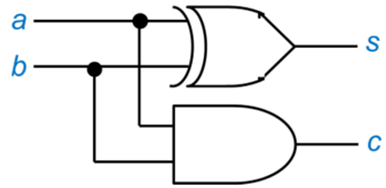
	Using existing AND/XOR/COPY protocols	Devising a tailor-made half adder
Half adder 	10	8
Voting 	$2\lceil \log n \rceil + 8$	$2\lceil \log n \rceil + 6$

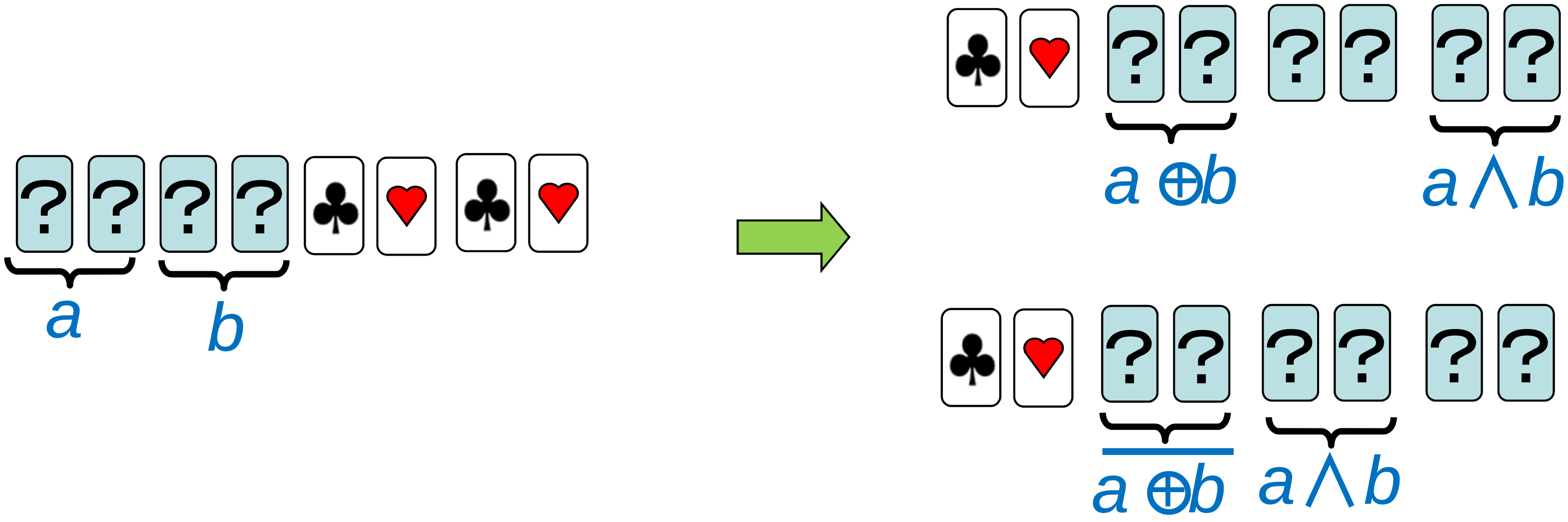
[# of cards]

Contents

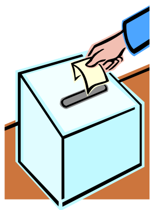
We gave a 8-card secure half adder protocol.

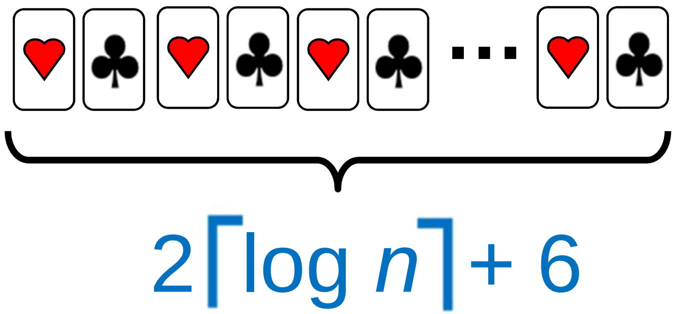






It enables us to conduct voting with $2\lceil \log n \rceil + 6$ cards.





n voters

I hope card-based protocols would help you with

- intuitive explanation of crypto. to non-specialists
- education in classroom.

That's all.
Thank you for your attention.

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