FSM in Programming

A trial to polish programming logic

Yao yizhi, Zhang Yunjie

November 26, 2021

UMJI Student Science and Technology Innovation Association

Table of contents

Possible programming scenario

Simulation of daily games

Sometimes(or in some courses) we would simulate different kinds of games including **complex judgement of conditions**, which can cause weird bugs if the **logic behind** is not clearly constructed.....

Figure 1: The action of players are judged by the color of certain cards or the dice.

Simulation of daily games

Then you may get this:

```
if(//some condition) {
    //some implementation
    if(//some condition) {
        //some implementation
    } else {
        //some implementation
        if(//some condition) {
            //some implementation
            if(//some condition) //some implementation
        }
} else if (//some condition){
    //some implementation
} else {
    //some implementation
}
```

which can be annoying when debugging though you know there is some unconsidered cases.

Web Design

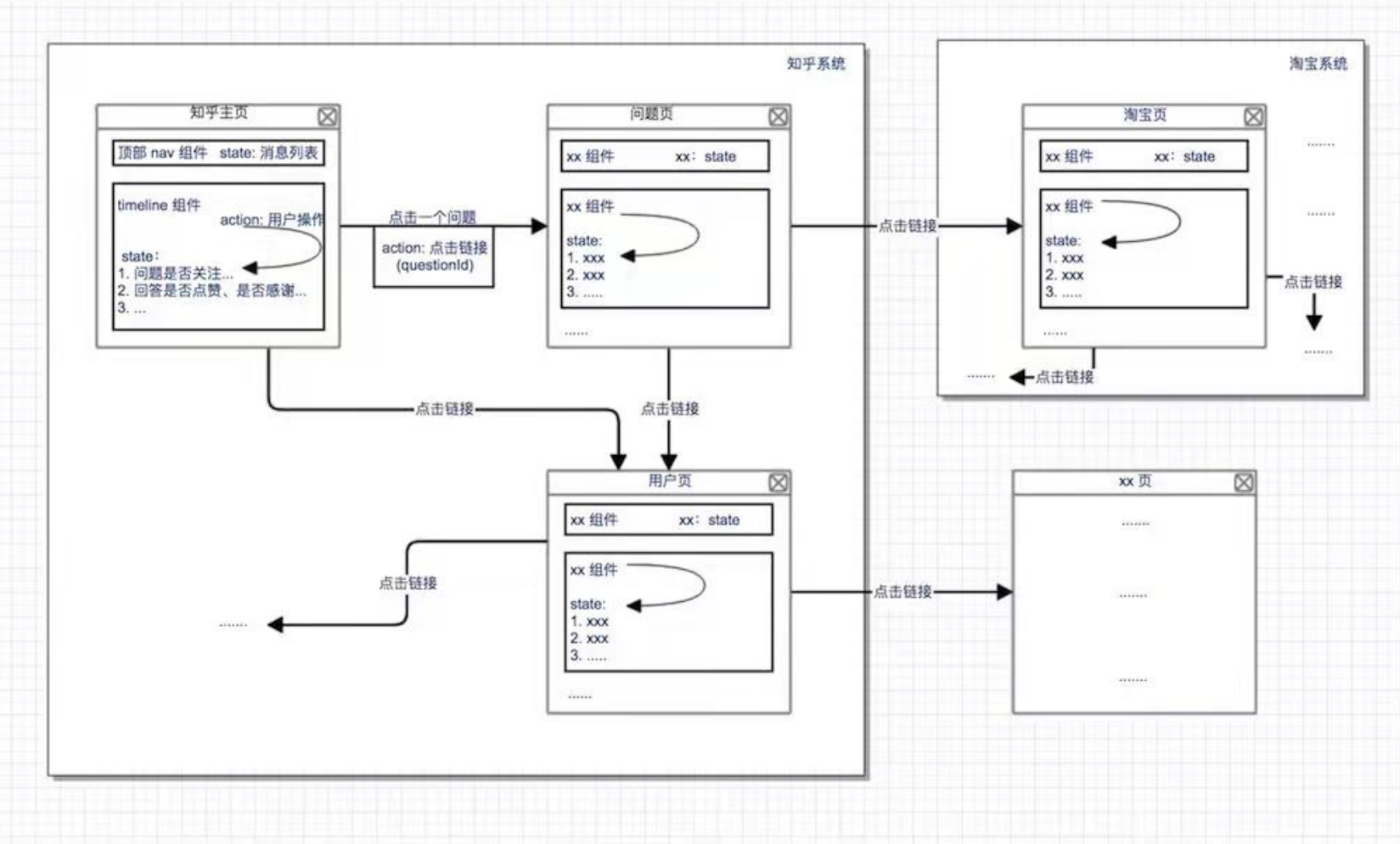


Figure 2: an example of fsm used in web design

From: https://zhuanlan.zhihu.com/p/26524390

Introduction to FSM

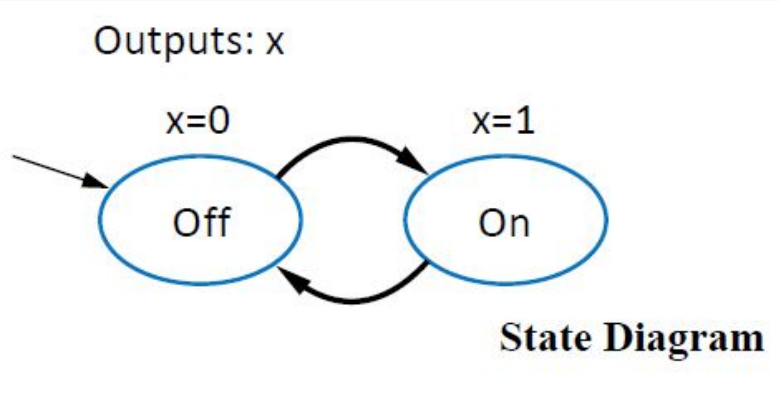
FSM-Finite State Machine

Finite State Machine (FSM) is a method to describe desired behaviour of a sequential event[1]. It's always at a certain state in a finite set of states. It's

A regular FSM consists of:

- Set of states
- Set of inputs and set of outputs
- Initial State
- Set of transitions
- Set of actions

State Diagram



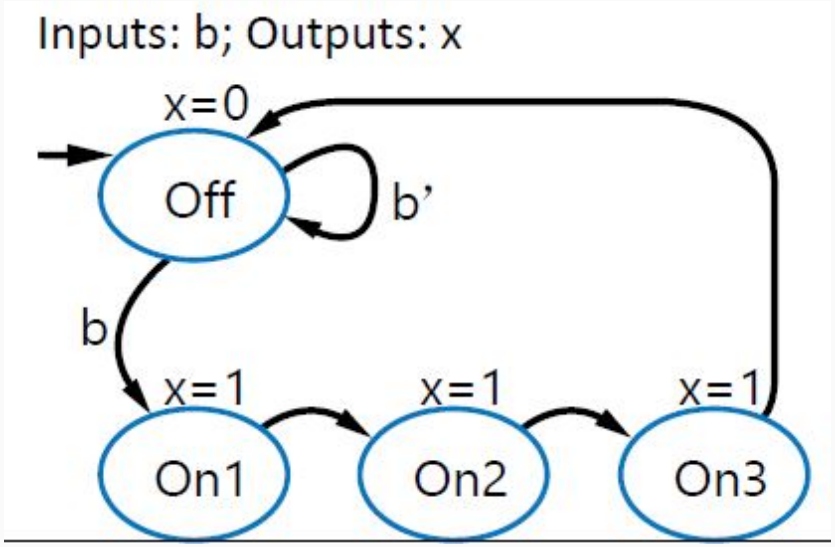


Figure 3: FSM can be represented graphically, known as **state diagram**

Example: House Tour

Here's a simple example of a room tour.

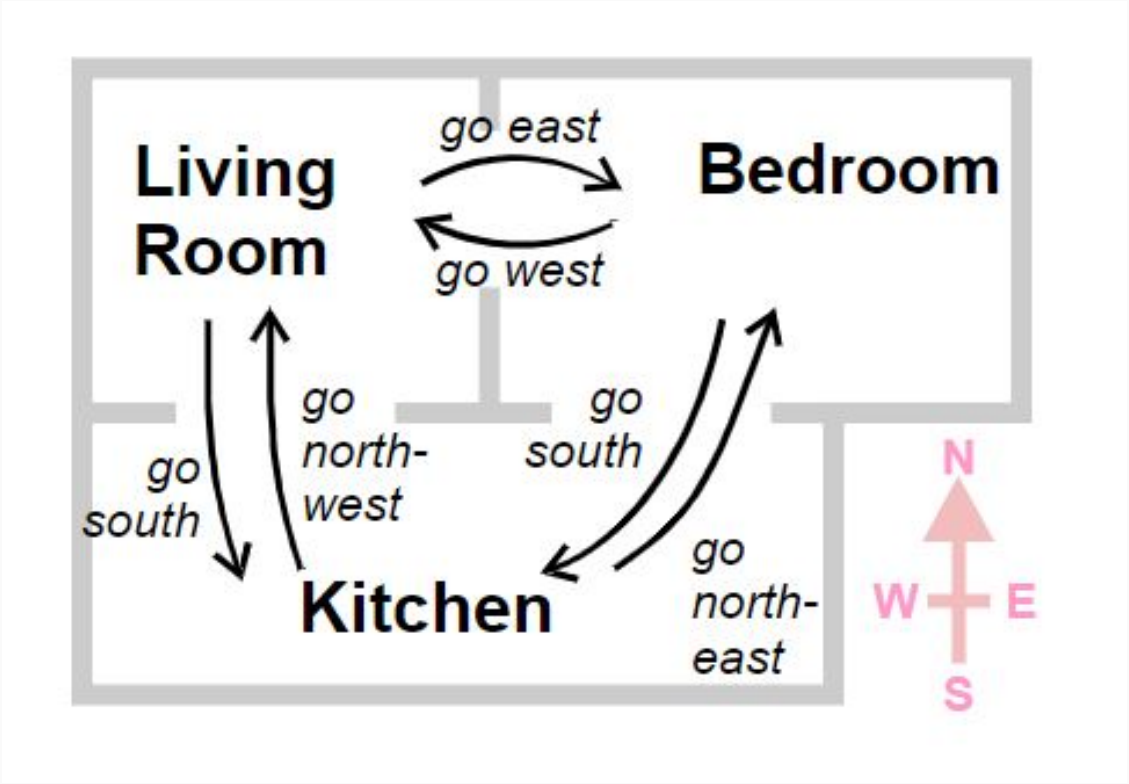


Figure 4: The intuitive diagram of a house.

from: https://www.programmingbasics.org/en/beginner/fsm.html

Example: House Tour

First denote the rooms from 1 to 3.

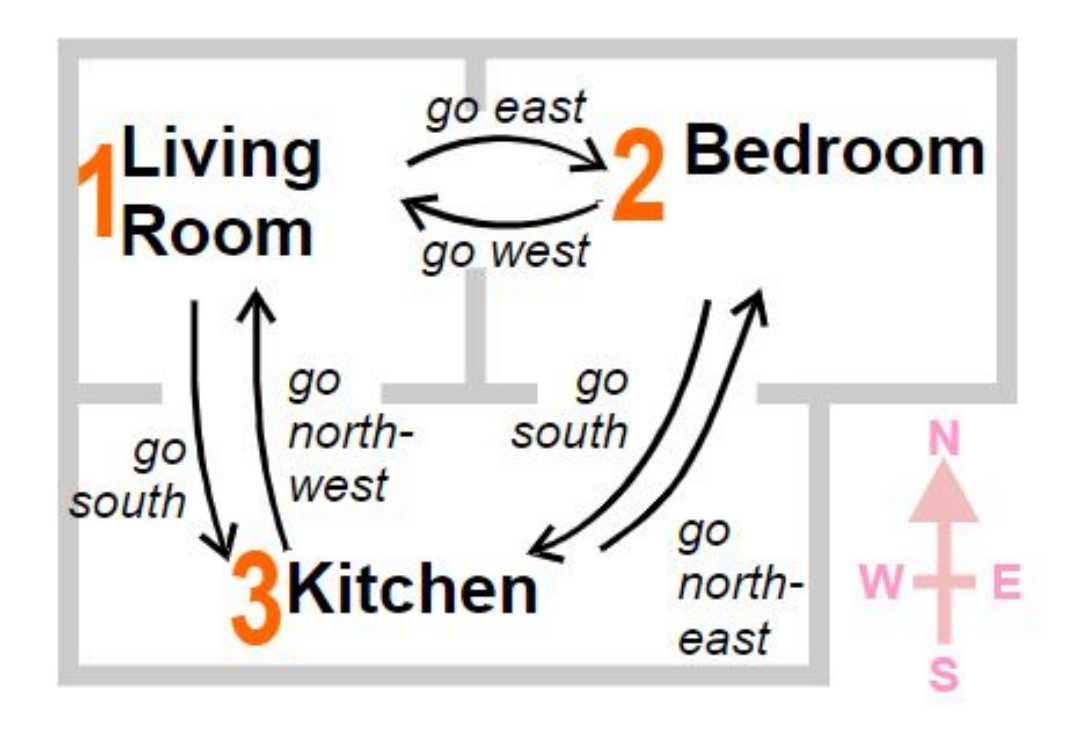


Figure 5: The intuitive diagram of a house.

from: https://www.programmingbasics.org/en/beginner/fsm.html

Change it into code

```
// omit other macro definitions
// balabala
switch (location) {
    case LIVIN_GROOM :
        printf("You are in the living room!\n");
        break;
    case BEDROOM:
        printf("You are in the bedroom!\n");
        break;
    case KITCHEN:
        printf("You are in the kitchen!\n");
        break;
    default:
        printf("You are in a invalid location!\n");
}
// balabala
```

Change it into code

```
// balabala
case LIVING_ROOM :
    printf("You are in the living room!\n");
    int action = getAction();
    switch (action) {
        case GO_EAST :
            location = BEDROOM;
            break;
        case GO_SOUTH :
            location = KITCHEN;
            break;
        default:
            printf("Your action is invalid!\n");
    }
    break;
// balabala
```

Example: Digital Lock

Start with designing a lock with four buttons that can recognize the built-in password.

The door is unlocked($u = 1$) only when the buttons are pressed in the following sequence: start, red, blue, green, red, (represented by input signal: s, r , g, b).

The input signal a equals 1 when some color button is pressed.

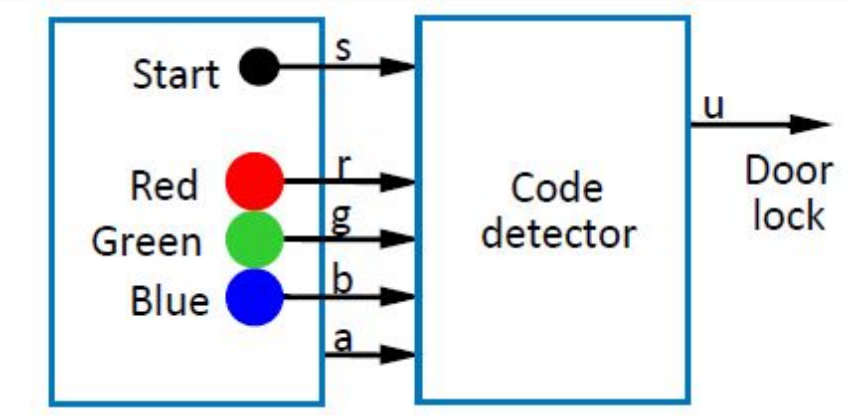


Figure 6: The schematic diagram of a digital lock.

Example: Digital Lock

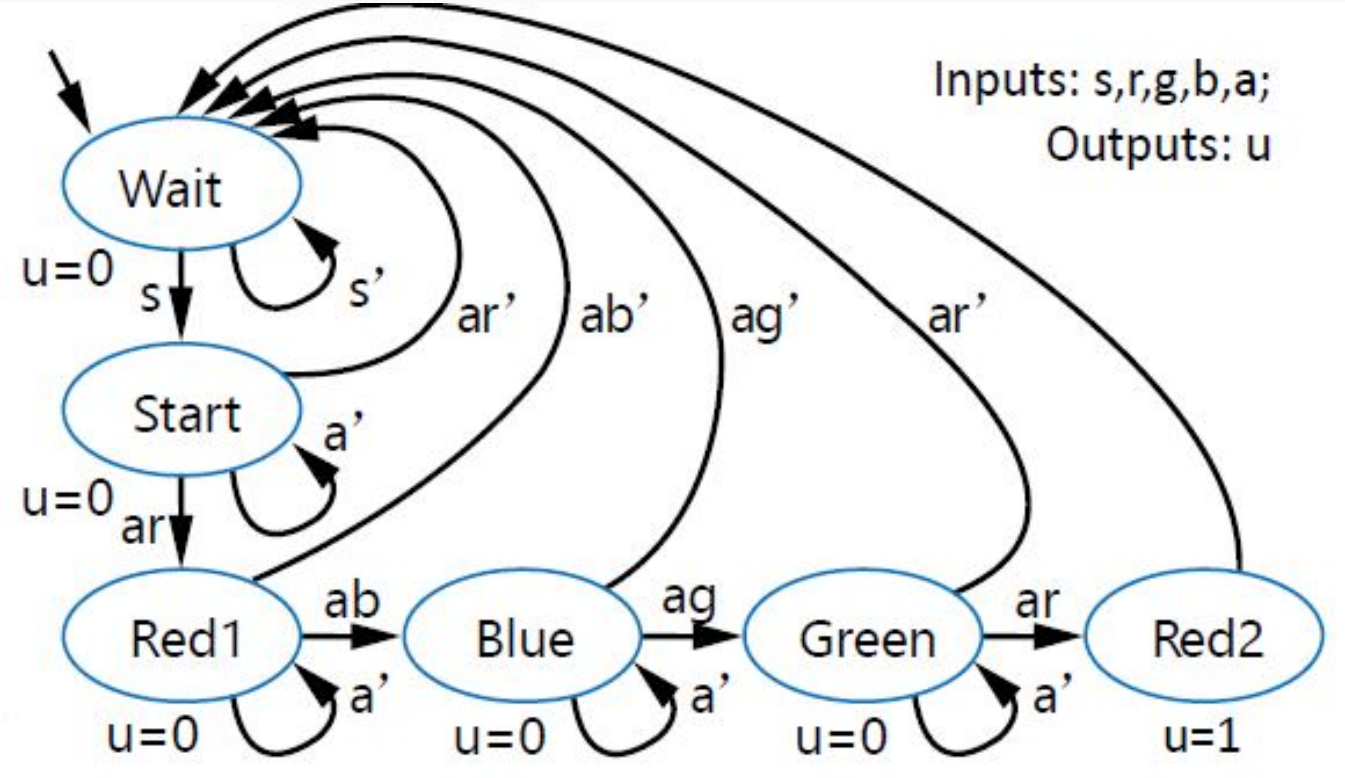


Figure 7: The state diagram of a digital lock.

Can you trick this FSM to open the door, without knowing the code?

Common State Transition Property

Basic Knowledge about Boolean Algebra

- Variables, like a, b, c, x, y, etc. Representing 0 or 1.
- Logic operators: AND($\cdot$), OR(+), NOT(').
- Operator Precedence: Parenthesis() $>$ NOT $>$ AND $>$ OR.

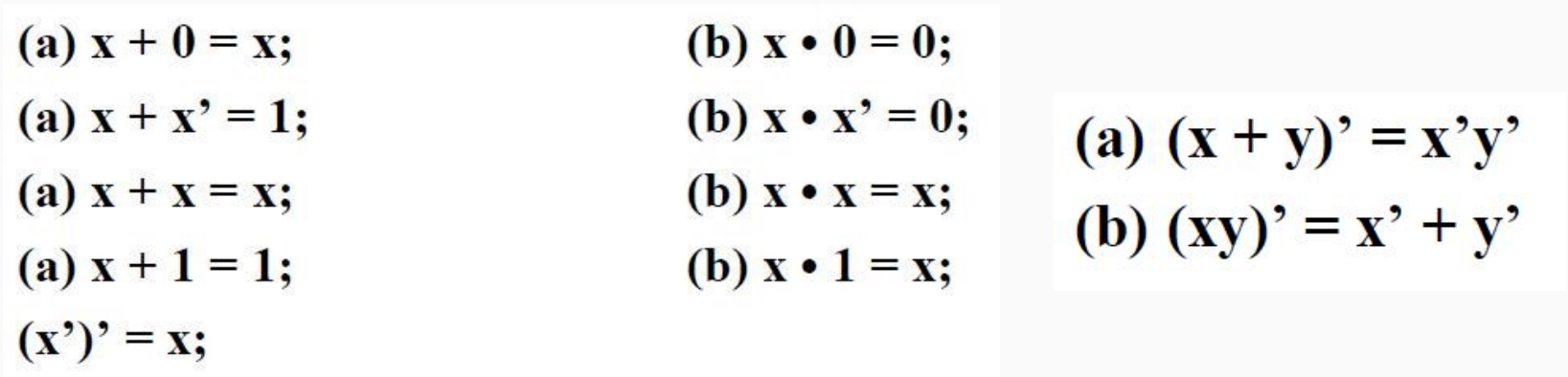
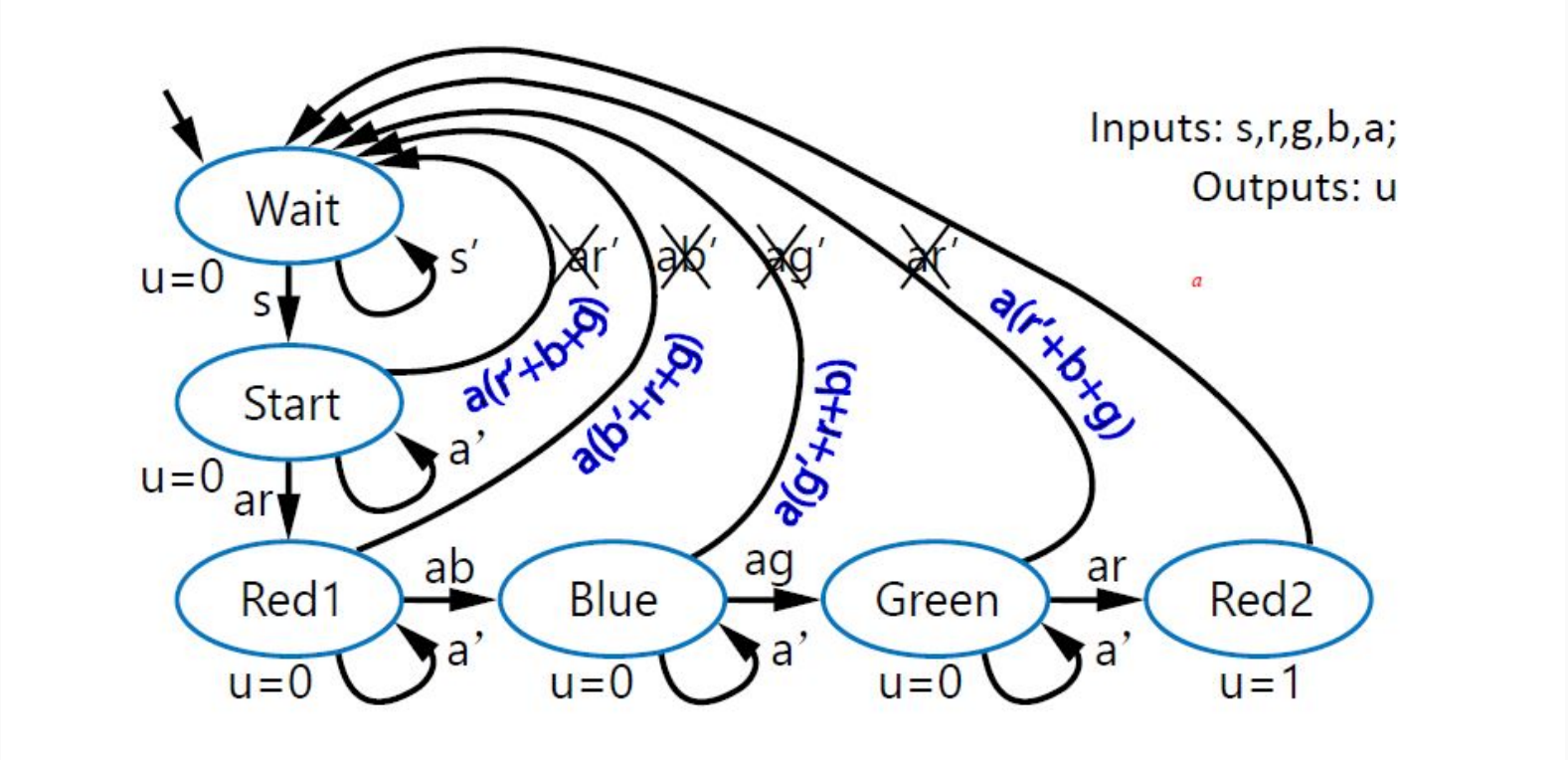


Figure 8: Basic Theorems of Boolean Algebra

This is relatively intuitive. Eg. a b c' d'+a(b'+c+d)+a' = 1

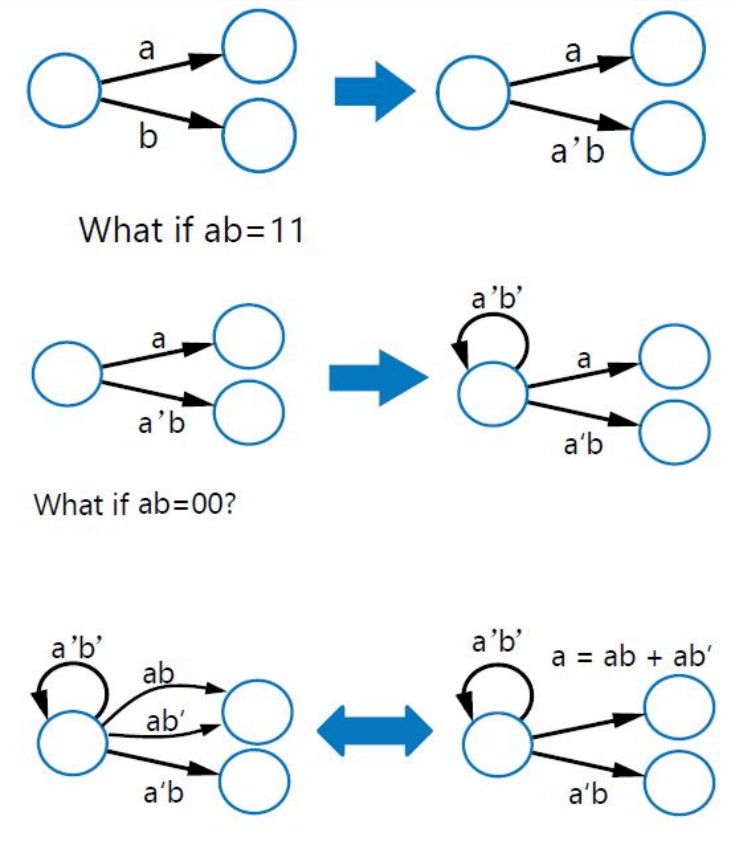
Improved Digital Lock



Refresh the conditions. Now it returns to "Wait" state if wrong button is pressed.

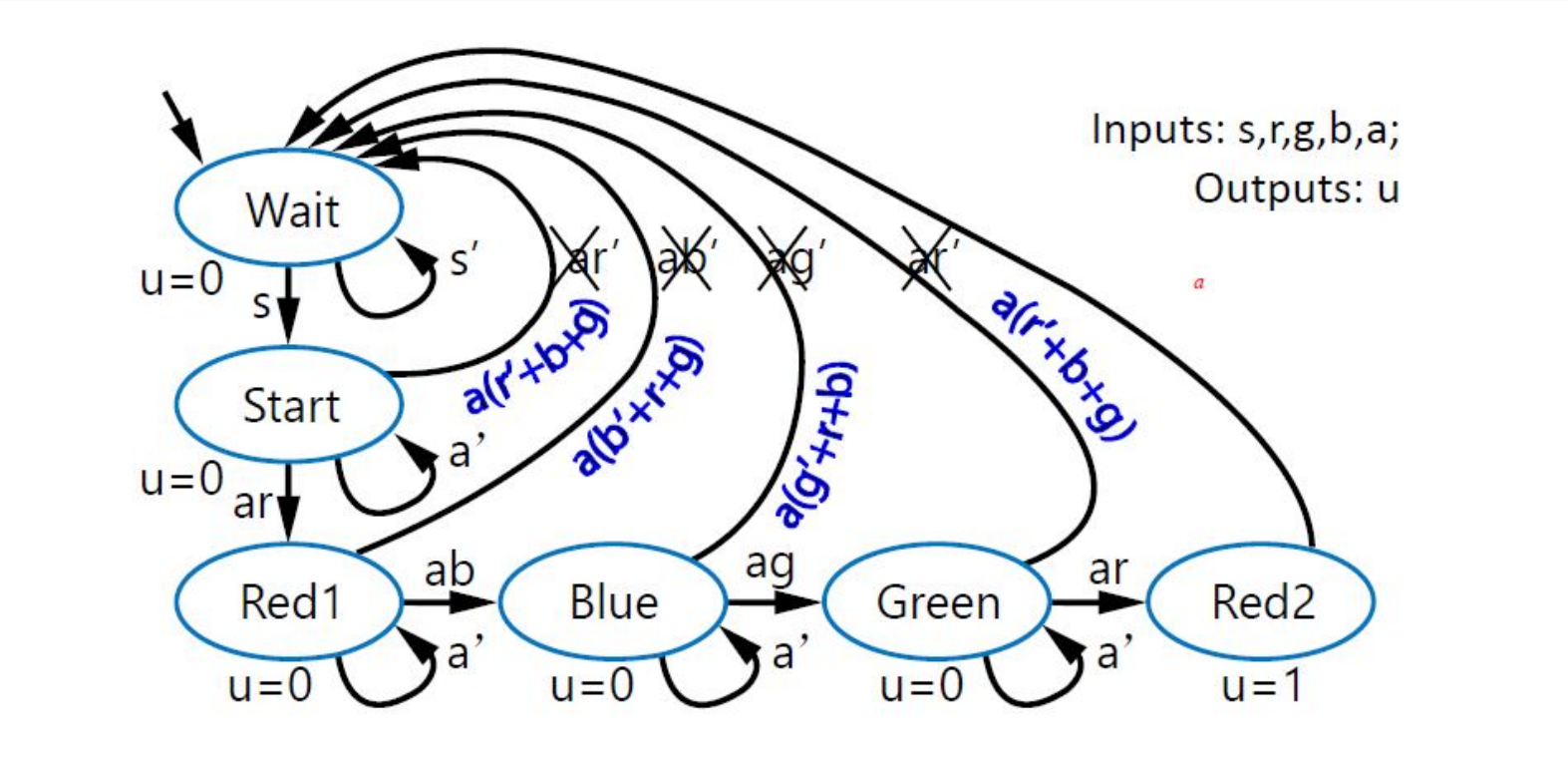
Common State Transition Property

- Among all transitions leaving a state, **only one condition** should be true.
- For any input combination, **one** condition must be true.
- All conditions must be considered when leaving a state. That is to say, their logic sum should be 1.



Improved Digital Lock

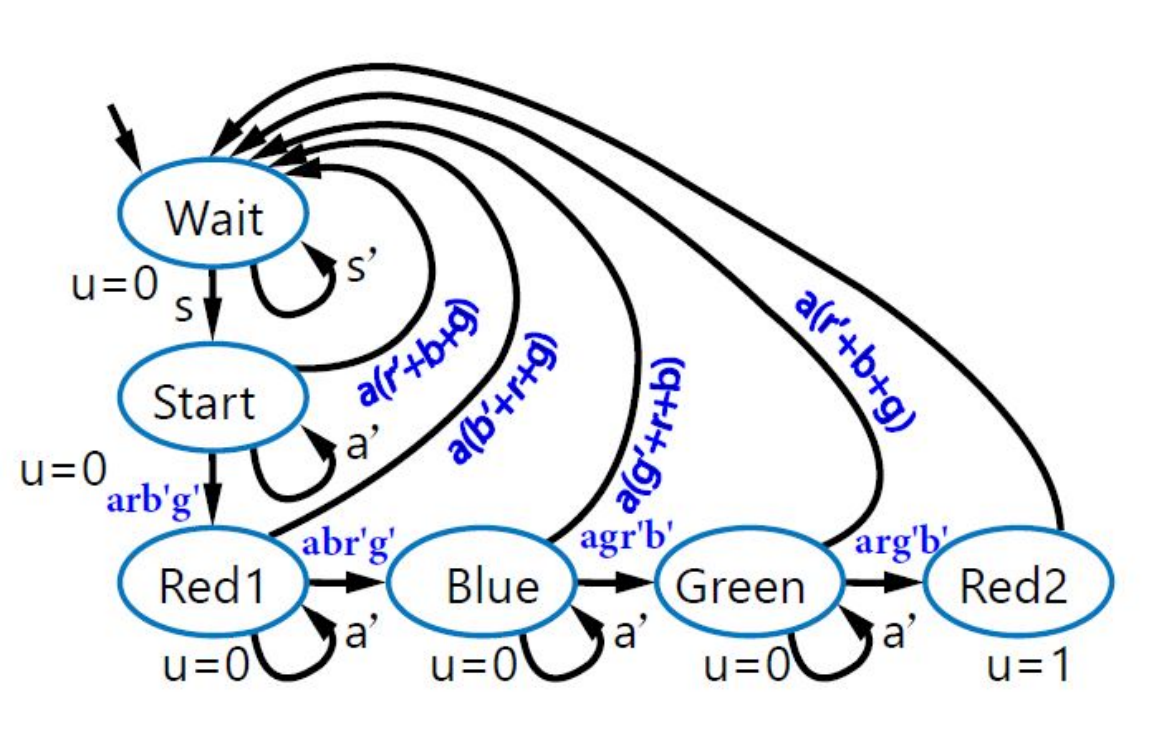
Now go back to the digital lock.



Do the transitions obey the required transition properties?

For example, arbg $= 1111$?

Another Improved Digital Lock



Now all the conditions leaving a state have a logic sum of 1.

Instance for FSM in programming

Hangman Game

Hangman is a classic word game in which you must guess the secret word one letter at a time. Guess **one letter** at a time to reveal the secret word. Each incorrect guess **adds another part** to the hangman.

In our game, we suppose there hangman has **10** lives and the random word chosen is **'apple'**.

Hangman Game

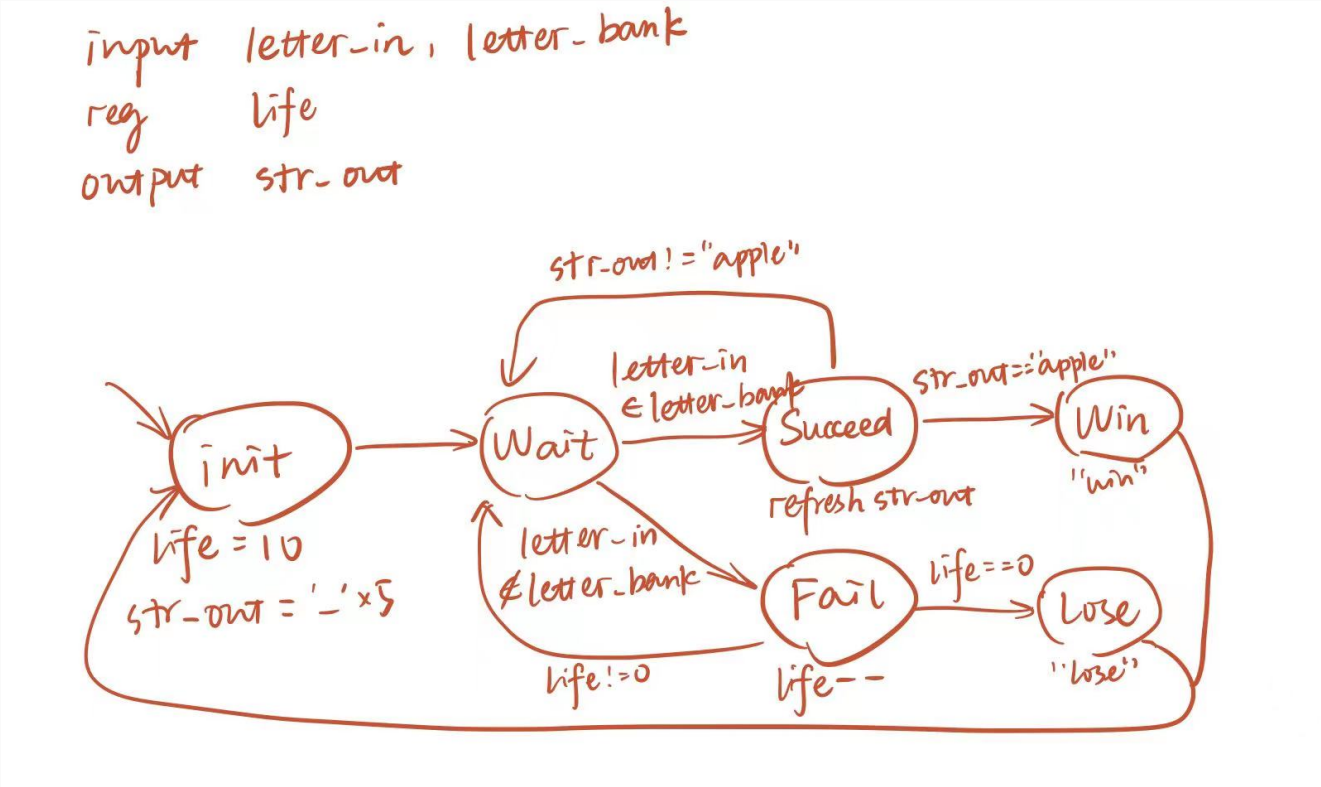
BACKROUND:

1. Random word chosen as 'apple'.
2. Print the letters covered by dashes: "_ _ _ _ _".
3. The letters are stored in letter-bank: $\{a\ p\ l\ e\}$.

Hangman Game

BACKROUND:

1. Random word chosen as 'apple'.
2. Print the letters covered by dashes: "_ _ _ _ _".
3. The letters are stored in letter-bank: $\{a\ p\ l\ e\}$.



Hangman Game

Special items: (shown in item_out)

- $*$ Uncover the left-most undiscovered letter.
- / Kill the hangman and end the game immediately.
- $ Pay to add 5 to the hangman's life, but the total life cannot exceed 10.
- # Mark all undiscovered vowels with #, which can also be discovered later.

The improved version is shown by hand-drawn.

Monopoly

- At least two players.
- Action of players: rolling two dice, making decisions.
- Grids on the map have different features.

Monopoly


SEMINOPOLY
Made in 2007, using Adobe Illustrator and Illustrator on a MacBook Pro.
Board design by Art Gearing
COMMUNITY CHEST
CHANCE
GO
COLLECT $200.00 SALARY AS YOU PASS
THE TOWER
PRICE $400
ROOM AND BOARD
PAY $75.00
ST. JOSEPH CHAPEL
PRICE $350
CHANCE
SISTERS' CAR
PRICE $200
KENWICK LOUNGE
PRICE $320
COMMUNITY CHEST
FOLLOW INSTRUCTIONS ON THE CARD
PRIEST'S LOUNGE
PRICE $300
GLENNON LOUNGE
PRICE $300
GO TO RECTOR'S OFFICE
ST. CHARLES CHAPEL
PRICE $280
STUDENT COMPUTER SERVICES
PRICE $150
MARY MOTHER OF THE WORD CHAPEL
PRICE $260
CONVENT CHAPEL
PRICE $260
WHITE VAN
PRICE $200
PRIEST DINING ROOM
PRICE $240
THE COURTYARD
PRICE $220
CHANCE
THE LOBBY
PRICE $220
FREE PARKING
THE REFECTORY
PRICE $200
THE HEIGHTS
PRICE $160
COMMUNITY CHEST
FOLLOW INSTRUCTIONS ON THE CARD
THE GYM
PRICE $160
SILVER VAN
PRICE $200
THE AUDITORIUM
PRICE $160
THE LIBRARY
PRICE $140
KENRICK LIGHT AND MAGIC
PRICE $150
THE DORM
PRICE $140
JUST VISITING
RECTOR'S OFFICE
WEIGHT ROOM
PRICE $120
LAUNDRY ROOM
PRICE $100
CHANCE
TENNIS COURTS
PRICE $100
RED VAN
PRICE $200
FORMATION FEE
PAY 10% OR $200
BOILER ROOM
PRICE $60
COMMUNITY CHEST
FOLLOW INSTRUCTIONS ON THE CARD
WRECK ROOM
PRICE $60

Monopoly

References

1. Zheng, Gang. "Finite State Machine." *VE270 Lecture Notes*, pp. 8-14. *UMJI Canvas.* https://www.umjicanvas.com/courses/2035/files/folder/LectureNotes. Accessed Nov.14, 2021.
2. Shi, Xiaotian. "RTL Design Worksheet". *RC Slides*, pp. 1-2. *UMJI Canvas.* https://www.umjicanvas.com/courses/2035/files/folder/RCSlides. Accessed Nov.14, 2021.

An Illustration of FSM

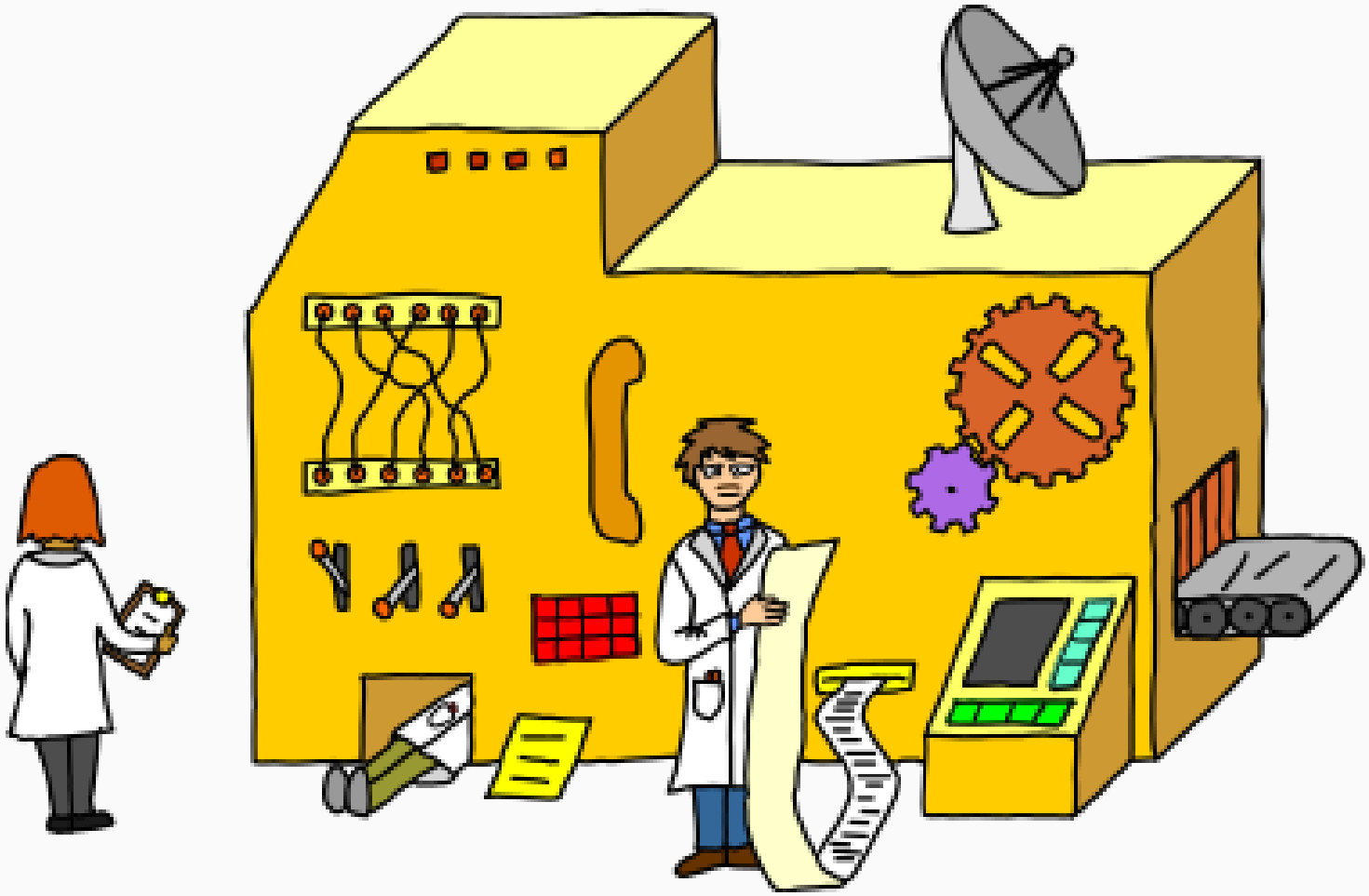


Figure 10: an illustration of FSM

Thanks

We are more than honored to discuss on this topic. (@ ▪ . ▪ @)