Seven-Year Activity to Introduce B-Method to Japanese Technical College Students

FMTea 2026, Tokyo, Japan
May 19 (Tue.), 2026

Takaomi OHNISHI
Yoshihiko NAKAMURA
Ryota YAMAMOTO

National Institute of Technology, Tomakomai College,
Hokkaido, Japan

Contents

Contents

1. Background

"Why we start to teach Formal Methods?"

In 2015, Ohnishi participated in a joint research project with a private company supported by Japanese government.

On the project, Ohnishi was tasked with teaching formal methods at his school.
He has learned about the formal method B-Method from the staff of the company.

We launched the student experiments since 2019, struggling with any constraint.
We set the educational goal for the experiments as giving opportunities to commit a formal method as the software engineer's fundamental knowledge.

1. Background

◆ 4th-grade class of our college (Age of 18-19)
- Whole member (at most 40) of the class (since 2019)
- Mandatory experiments
- 140min./week x 3weeks (“x 2weeks” until 2024)
- No knowledge nor experience about formal method
- Few knowledge nor experience about logical mathematics

◆ Short-time lectures other than our class
- No knowledge nor experience about formal method, logical mathematics

→ Simultaneous conducting both the experiment and the lecture provided the improvement of our methodology.

1. Background

We set the educational goal for the experiments as giving opportunities to commit a formal method as the software engineer's fundamental knowledge.

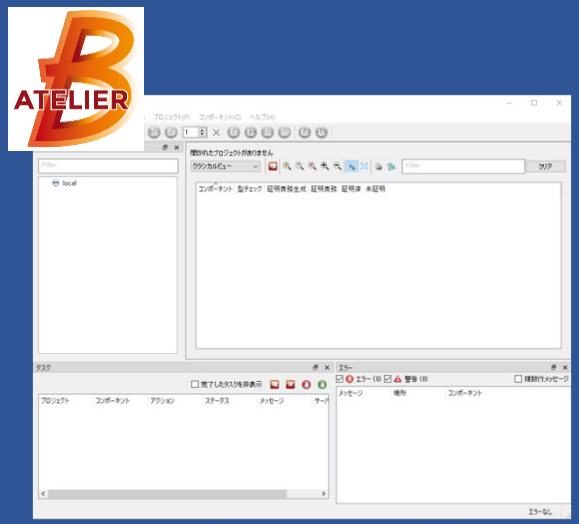
Our experiments include all of :

- Formal specification
- Proof obligation generation
- Model checking

The Levels of formal methods usage

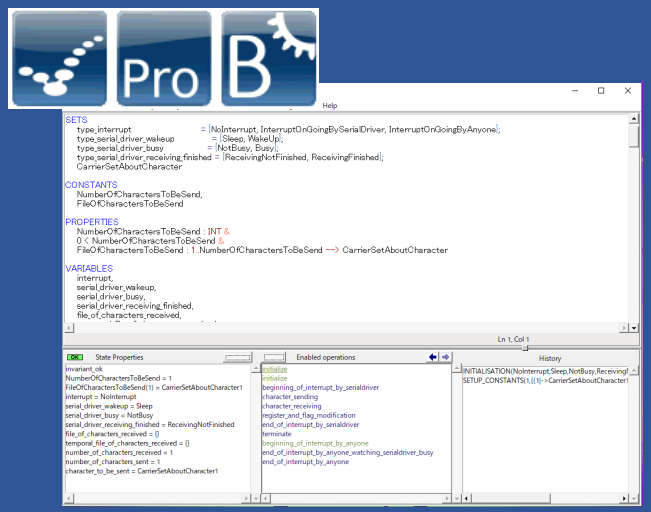
- Level-0: Formal Specification
- Level-1: Formal Development and Formal Verification
- Level-2: Theorem Provers

Hehner, E.C.R.: Computer Science - Formal Methods, https://computingstudy.wordpress.com/formal-methods/



Atelier B Ver 4.5.5
(ClearSy, France)

Formal specification
Proof obligation generating



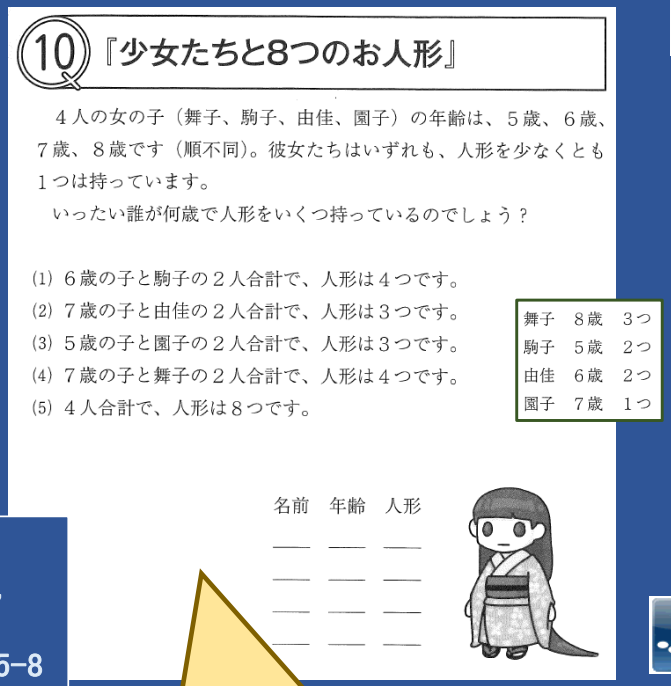
ProB Ver.1.9.3
(Heinrich Heine University Düsseldorf, Germany)

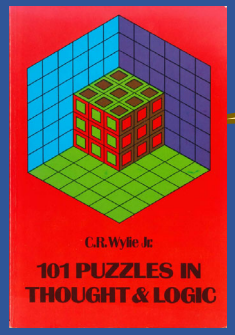
Model checking

◆ Contents

2. Experiment Week 1　　Single-state formal models “Deductive logic puzzles”

Onoda H.:
Logic Puzzle Best 100,
PHP(2015),
ISBN:978-4-569-82545-8



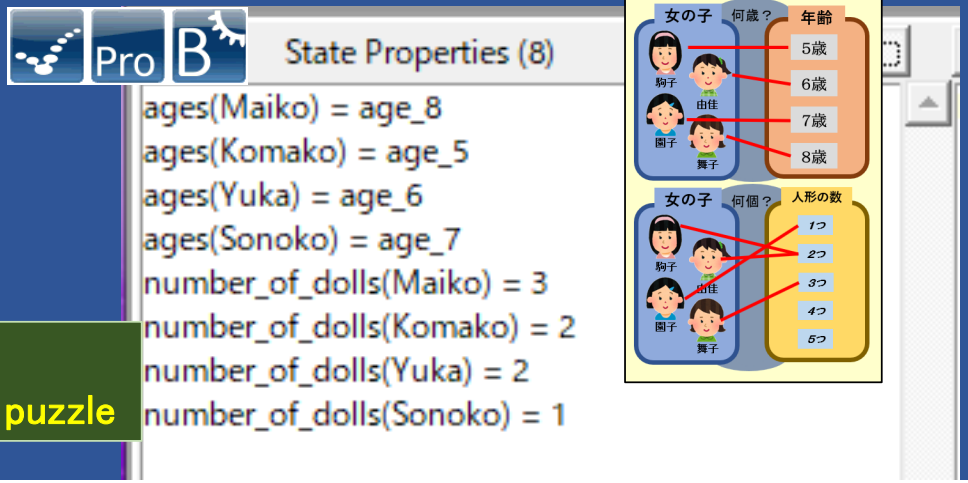


Wylie, C. R..
101 Puzzles in Thought & Logic,
Dover Publication, 1957

Deductive logic puzzle
with tacit knowledges

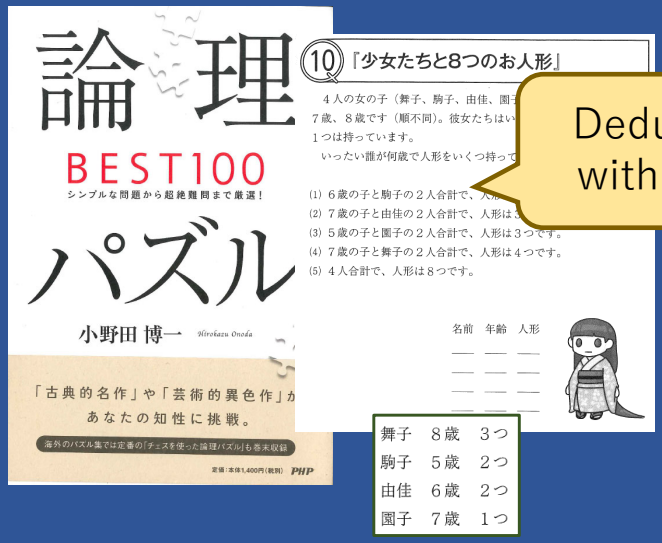
Formal specification

```
MODEL
  NodaPuzzle10

SETS
    GIRL = {Maiko, Komako, Yuka, Sonoko};
    AGE  = {age_5, age_6, age_7, age_8}

CONSTANTS
    ages,
    number_of_dolls

PROPERTIES
    ages             ∈ GIRL ⇸ AGE   ∧
    number_of_dolls ∈ GIRL → 1..5 ∧

    number_of_dolls(ages⁻¹(age_6)) + number_of_dolls(Komako) = 4 ∧
    /*「6歳の子と駒子の2人合計で、人形は4つです。」*/

    《中略。》

//     ∧ (ages             ≠ {Maiko↦age_8, Komako↦age_5, Yuka↦age_6, Sonoko↦age_7} ∨
//          number_of_dolls ≠ {Maiko↦3, Komako↦2, Yuka↦2, Sonoko↦1})
    /* --for verifying and validating no another solution exists-- */

END
```



Obtained answer
of deductive logic puzzle

2. Experiment Week 1



Deductive logic puzzle with tacit knowledges

Onoda H.:
Logic Puzzle Best 100,
PHP(2015),
ISBN:978-4-569-82545-8

Deductive logic puzzle

There are four persons: W, X, Y, and Z,
who are all of different ages: a, b, c, and d, not respectively.
Each of these owns one or more items.
- The person aged b and X together own a total of four items.
- The person aged c and Y together own a total of three items.
- The person aged a and Z together own a total of three items.
- The person aged c and W together own a total of four items.
- Four persons W, X, Y and Z together own a total of eight items.

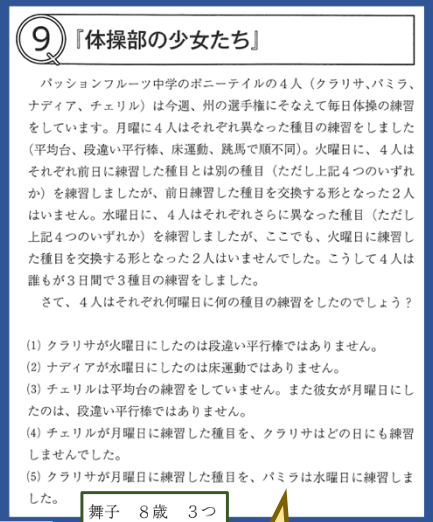
Describe in detail the relationship between the people, their ages, and the number of items they own.
(Answer: W-d-3, X-a-2, Y-b-2 and Z-c-1.)

```
SYSTEM
  PuzzleSample
SETS
  PERSON = {W, X, Y, Z};
  AGE = {a, b, c, d};
CONSTANTS
  ages       : PERSON >->> AGE &
  owns_item : PERSON --> 1..5     &

PROPERTIES
  owns_item(ages~(b)) + owns_item(X) = 4 &
  // The person aged b and X together own a total of four items.
  X|->b /= ages                          &
  // The person aged b is not X, therefore X is not b years old.

  ...omitted...

END
```

A tacit knowledge in the logic puzzle

Formal specification

2. Experiment Week 1



9 『体操部の少女たち』

パッションフルーツ中学のポニーテイルの４人（クラリサ、パミラ、ナディア、チェリル）は今週、州の選手権にそなえて毎日体操の練習をしています。月曜に４人はそれぞれ異なった種目の練習をしました（平均台、段違い平行棒、床運動、跳馬で順不同）。火曜日に、４人はそれぞれ前日に練習した種目とは別の種目（ただし上記４つのいずれか）を練習しましたが、前日練習した種目を交換する形となった２人はいません。水曜日に、４人はそれぞれさらに異なった種目（ただし上記４つのいずれか）を練習しましたが、ここでも、火曜日に練習した種目を交換する形となった２人はいませんでした。こうして４人は誰もが３日間で３種目の練習をしました。

さて、４人はそれぞれ何曜日に何の種目の練習をしたのでしょう？

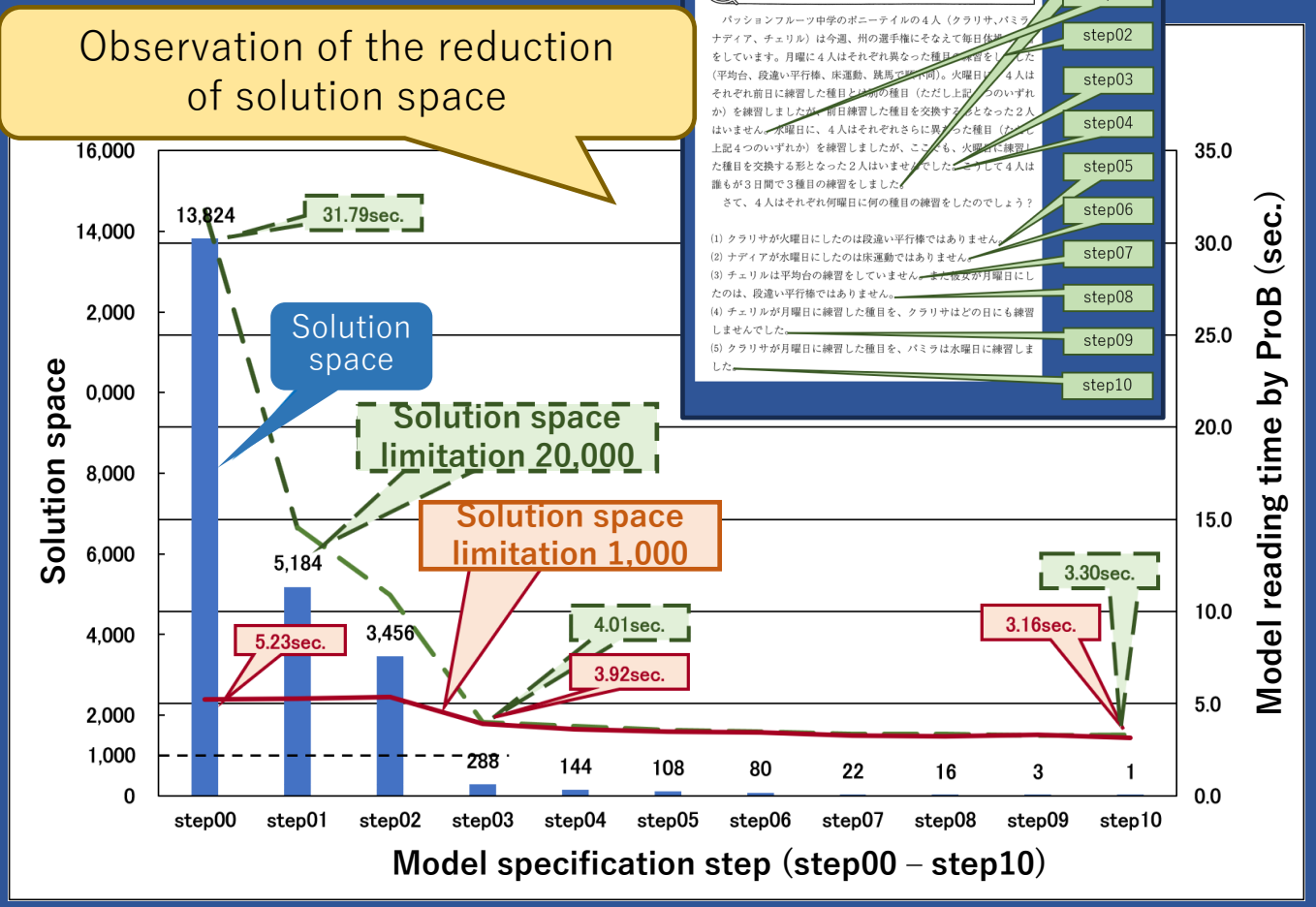
(1) クラリサが火曜日にしたのは段違い平行棒ではありません。
(2) ナディアが水曜日にしたのは床運動ではありません。
(3) チェリルは平均台の練習をしていません。また彼女が月曜日にしたのは、段違い平行棒ではありません。
(4) チェリルが月曜日に練習した種目を、クラリサはどの日にも練習しませんでした。
(5) クラリサが月曜日に練習した種目を、パミラは水曜日に練習しました。

舞子	８歳	３つ
駒子	５歳	２つ
由佳	６歳	２つ
園子	７歳	１つ

Onoda H.:
Logic Puzzle Best 100,
PHP(2015),
ISBN:978-4-569-82545-8

Deductive logic puzzle with tacit knowledges

Observation of the reduction of solution space

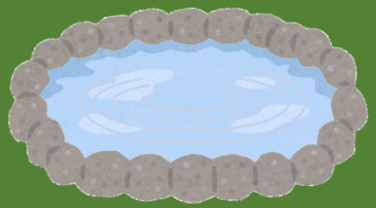


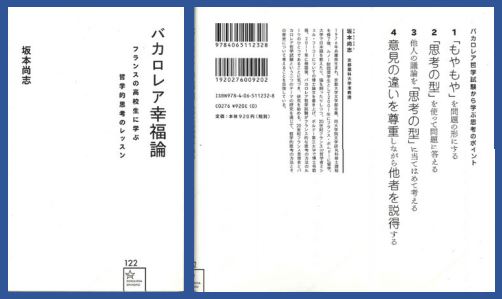
◆ Contents

3. Our idea 1 : Fountain of logic expressions

Source (References)

Aristotle, Descartes, Spinoza, Freud, Kant, Rousseau, Saint-Just, Bentham, Hume, Pascal, Schopenhauer, Nietzsche, …





Sakamoto T.:
The Philosophy of Happiness in the Baccalaureate —Lessons in Philosophical Thinking from French High School Students—,
Seikaisha Shinsho(2018),
ISBN:978-4-06-51132-8

Compared to Japan, France ranks lower in terms of statistics for unemployment, homicide, traffic accidents, and obesity, but it scores higher in terms of well-being. In other words, people there feel happier.

Philosophy Exam Questions

Freedom, Art, Truth, Reason, Morality, History, Technology, Justice, Happiness, Language, …

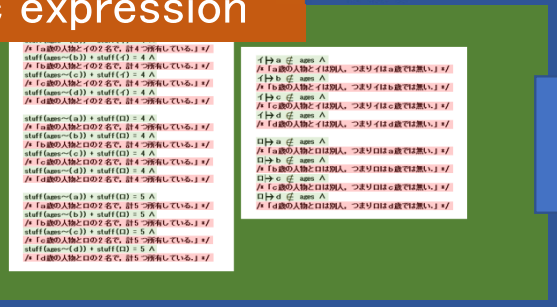
2017's General Baccalaureate Exam
Philosophy (Major of Natural Sciences)
Exam duration: 4 hours
《Dissertation (Philosophy Essay Question)》

1. Is defending one's rights the same as defending one's profits?
2. Is it possible to be free from one's own culture?

This is because, when taking their university entrance exams, French students use "models of thought" in their philosophy exams to develop the ability to think for themselves and overcome challenges.

Serendipitously inspired from European education

Fountain of logic expression

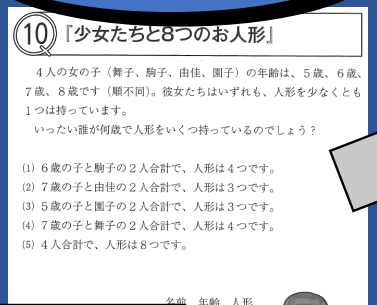


Deductive logic puzzle

Onoda H.:
Logic Puzzle Best 100,
PHP(2015),
ISBN:978-4-569-82545-8

3. Our idea 1 : Fountain of logic expressions

Deductive logic puzzle

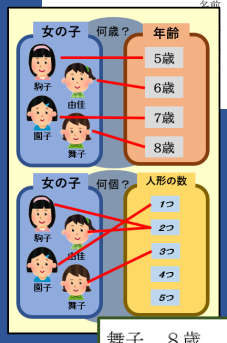


10『少女たちと8つのお人形』

4人の女の子（舞子、駒子、由佳、園子）の年齢は、5歳、6歳、7歳、8歳です（順不同）。彼女たちはいずれも、人形を少なくとも1つは持っています。
いったい誰が何歳で人形をいくつ持っているのでしょう？

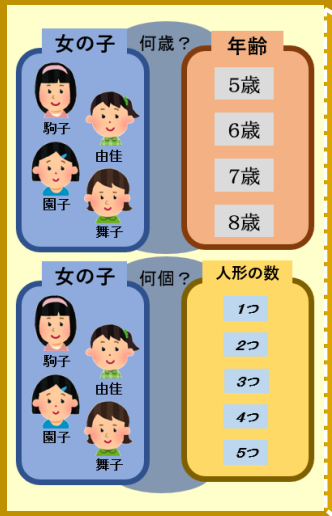
(1) 6歳の子と駒子の2人合計で、人形は4つです。
(2) 7歳の子と由佳の2人合計で、人形は3つです。
(3) 5歳の子と園子の2人合計で、人形は3つです。
(4) 7歳の子と舞子の2人合計で、人形は4つです。
(5) 4人合計で、人形は8つです。

名前　年齢　人形



舞子	8歳	3つ
駒子	5歳	2つ
由佳	6歳	2つ
園子	7歳	1つ

Possible relationships



Constraint

Logical predicate written in B-Method

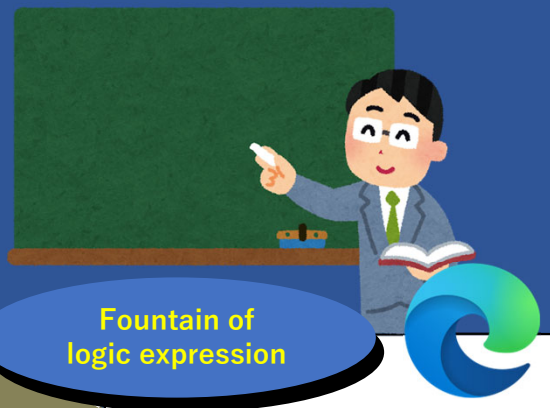
```
& girl_ningyo(girl_nenrei~(nenrei_6sai)) + girl_ningyo(Maiko) = 4
//6歳の子の人形と、舞子の人形は、合計で4つです。
```

Correct

Proposition by natural language (Commented out)

Outer products

Fountain of logic expression

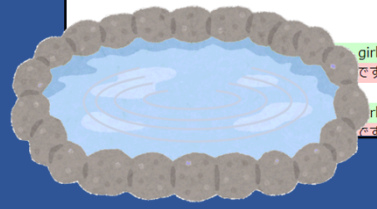


```
& girl_ningyo(girl_nenrei~(nenrei_5sai)) + girl_ningyo(M
//5歳の子の人形と、舞子の人形は、合計で3つです。                    Incorrect

& girl_ningyo(girl_nenrei~(nenrei_5sai)) + girl_ningyo(M
//5歳の子の人形と、舞子の人形は、合計で4つです。                    Incorrect

& girl_ningyo(girl_nenrei~(nenrei_6sai)) + girl_ningyo(M
//6歳の子の人形と、舞子の人形は、合計で3つです。                    Incorrect

& girl_ningyo(girl_nenrei~(nenrei_6sai)) + girl_ningyo(M
//6歳の子の人形と、舞子の人形は、合計で4つです。                    Correct

& girl_ningyo(girl_nenrei~(nenrei_7sai)) + girl_ningyo(M
//7歳の子の人形と、舞子の人形は、合計で3つです。                    Incorrect

& girl_ningyo(girl_nenrei~(nenrei_7sai)) + girl_ningyo(M
//7歳の子の人形と、舞子の人形は、合計で4つです。                    Incorrect

& girl_ningyo(girl_nenrei~(nenrei_8sai)) + girl_ningyo(M
//8歳の子の人形と、舞子の人形は、合計で3つです。                    Incorrect

& girl_ningyo(girl_nenrei~(nenrei_8sai)) + girl_ningyo(M
//8歳の子の人形と、舞子の人形は、合計で4つです。                    Incorrect

& girl_ningyo(girl_nenrei~(nenrei_5sai)) +
//5歳の子の人形と、駒子の人形は、合計で
& girl_ningyo(girl_nenrei~(nenrei_5sai
//5歳の子の人形と、駒子の人形は、合計で
```

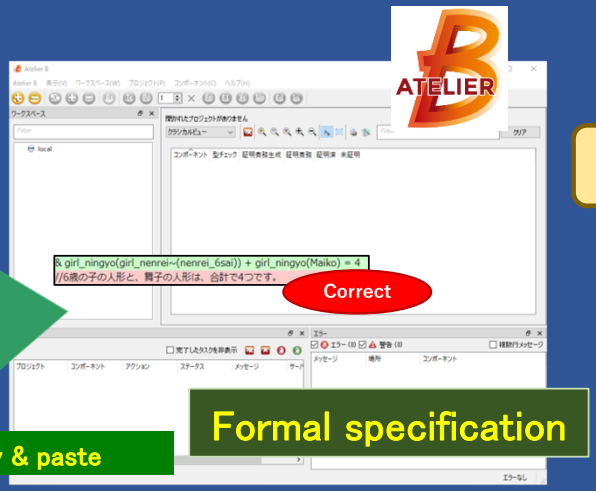
3. Our idea 1 : Fountain of logic expressions

Fountain of logic expression

```
& girl_ningyo(girl_nenrei~(nenrei_5sai)) + girl_ningyo(Maiko) = 3
//5歳の子の人形と、舞子の人形は、合計で3つです。

& girl_ningyo(girl_nenrei~(nenrei_5sai)) + girl_ningyo(Maiko) = 4
//5歳の子の人形と、舞子の人形は、合計で4つです。

& girl_ningyo(girl_nenrei~(nenrei_6sai)) + girl_ningyo(Maiko) = 3
//6歳の子の人形と、舞子の人形は、合計で3つです。

& girl_ningyo(girl_nenrei~(nenrei_6sai)) + girl_ningyo(Maiko) = 4
//6歳の子の人形と、舞子の人形は、合計で4つです。

& girl_ningyo(girl_nenrei~(nenrei_7sai)) + girl_ningyo(Maiko) = 3
//7歳の子の人形と、舞子の人形は、合計で3つです。

& girl_ningyo(girl_nenrei~(nenrei_7sai)) + girl_ningyo(Maiko) = 4
//7歳の子の人形と、舞子の人形は、合計で4つです。

& girl_ningyo(girl_nenrei~(nenrei_8sai)) + girl_ningyo(Maiko) = 3
//8歳の子の人形と、舞子の人形は、合計で3つです。

& girl_ningyo(girl_nenrei~(nenrei_8sai)) + girl_ningyo(Maiko) = 4
//8歳の子の人形と、舞子の人形は、合計で4つです。

girl_ningyo(Komako) = 4
```

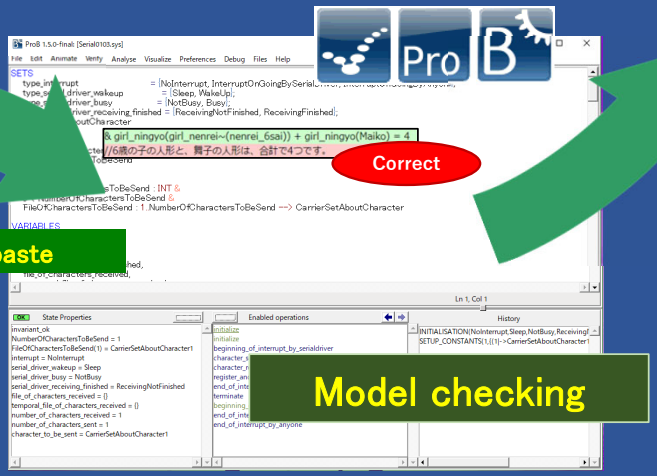


Copy & paste



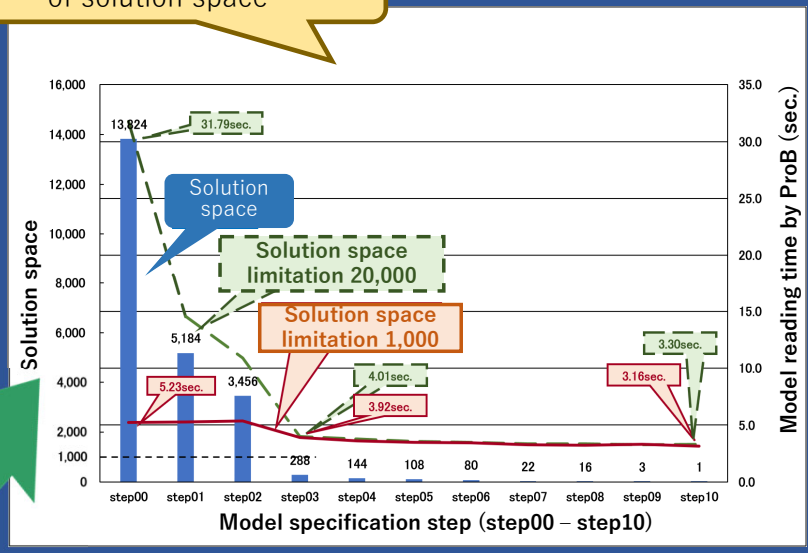
Formal specification

Copy & paste



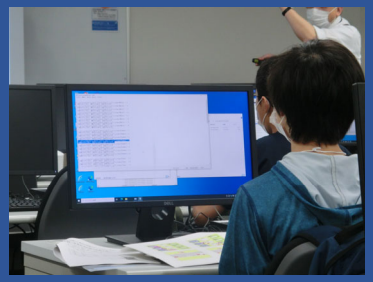
Model checking

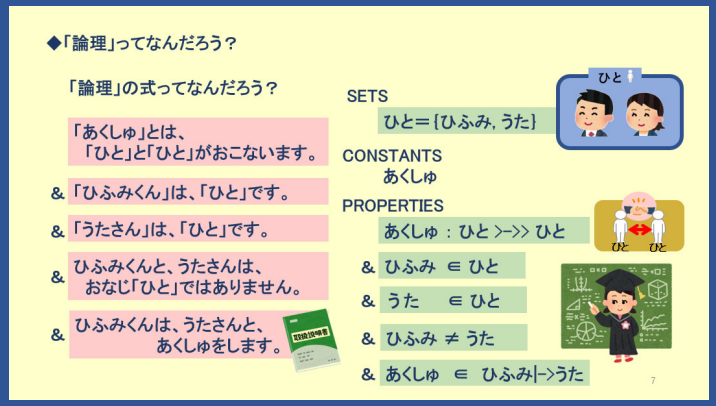
Observation of the reduction of solution space

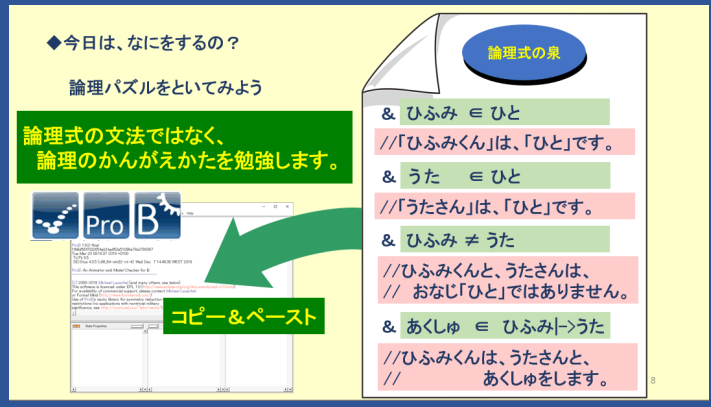


3. Our idea 1 : Fountain of logic expressions

Aim of kid's short-time lecture
"Explicitation of tacit knowledges"







Slides of introduction "What is Logic?"

◆ Contents

4. Experiment Week 2

Our “poor idea” for teach multi-state formal models

問題４《順序論理回路の設計》

問　よしひこ君は飛行機の搭乗中にレポートを書きたいです。表１のように論理変数を割り当てたときに、JKフリップフロップを用いて順序論理回路を設計して、必要な論理式を全て答えよ。

《Mr.Yoshihiko wants to write a report while his boarding the airplane. Design a sequential logic circuit using JK flip-flops. Answer all of essential boolean expressions with the assumption that boolean variables are assigned as Table 1.》

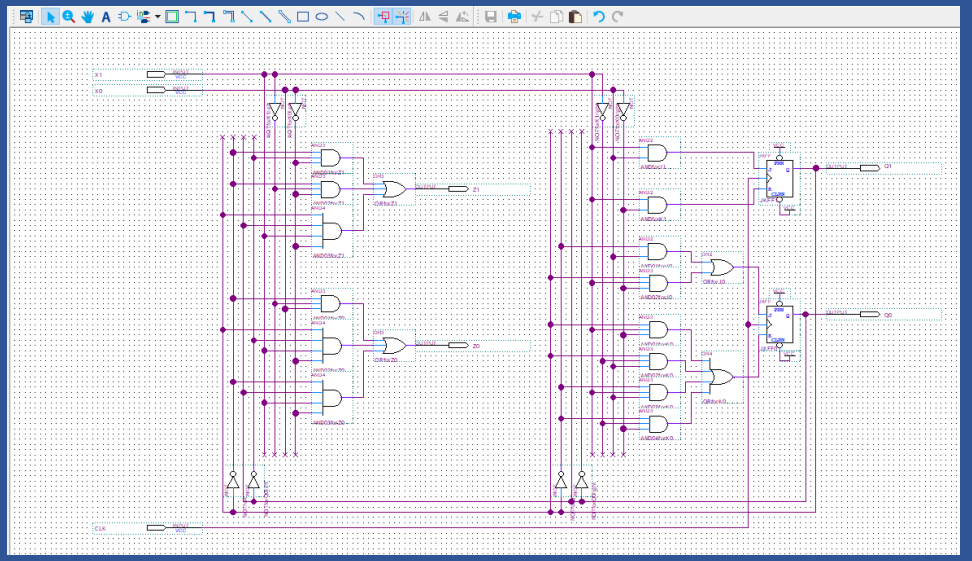
表１　順序論理回路の変数の割り当て

論理変数の名称	論理変数が持つ真理値の意味
状態：Q_1Q_0	00：テーブルは出ていて、紙はテーブルの上にある。《Seat table is pulled down, writing papers are on the table.》 01：テーブルは出ていて、紙はポケットの中にある。《Seat table is pulled down, writing papers are in the seat pocket.》 10：テーブルは片付いていて、紙はポケットの中にある。《Seat table is pushed back, writing papers are in the seat pocket.》 11：テーブルは片付いていて、紙はひざの上にある。《Seat table is pushed back, wringing papers are on his lap.》
入力：X_1X_0	00：もしテーブルが出ているならば、紙をポケットから出す。《If the table is pulled down, take the papers out of the pocket.》 01：紙をポケットの中に片付ける。《Put the papers in the seat pocket.》 10：飛行機が安定しているので、テーブルを出す。《Ensure the airplane is stable, pull the seat table down.》 11：乱気流が来た！客室乗務員がテーブルを片付ける。《Turbulence! A flight attendant pushes back the seat table.》
出力：Z_1Z_0	00：テーブルを引き出すことが期待される。《Hope to pull the seat table down.》 01：レポート用紙を取り出すことが期待される。《Hope to take the papers out of the seat pocket.》 10：無事にレポート執筆が開始できる。《Succeeded to start writing the report.》 11：客室乗務員にドーナツをたくさん注文したくなる。《Want to order lots of doughnuts from a flight attendant.》

Question from the test
of the lecture “Logic Circuit”

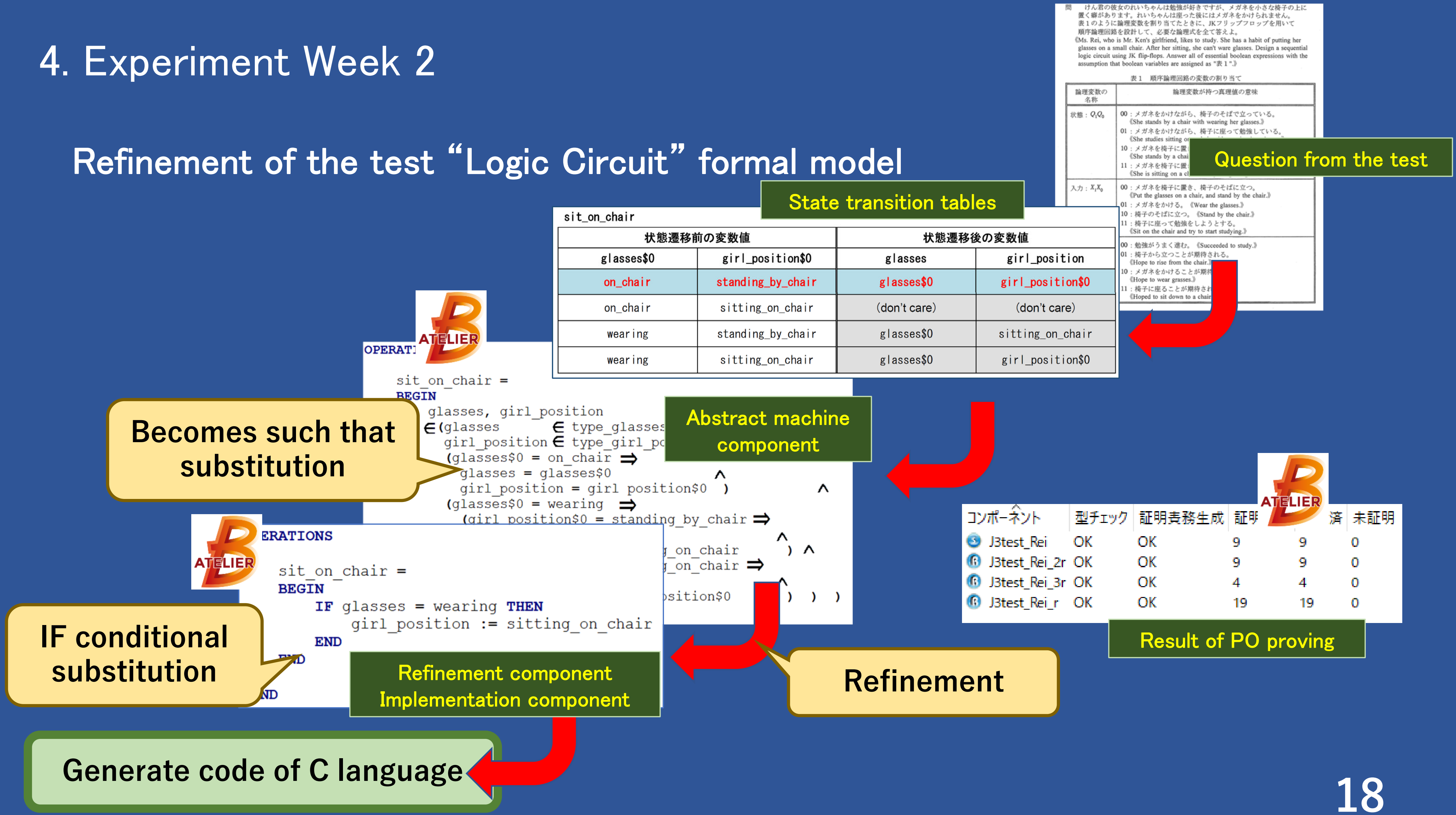
$$
\begin{cases}
J_1 = X_1 \cdot X_0 \\
K_1 = X_1 \cdot \overline{X_0} \\
J_0 = \overline{Q_1} \cdot X_0 + Q_1 \cdot X_1 \cdot \overline{X_0} \\
K_0 = Q_1 \cdot X_1 \cdot \overline{X_0} + Q_1 \cdot \overline{X_1} \cdot X_0 + \overline{Q_1} \cdot X_1 \cdot X_0 + \overline{Q_1} \cdot \overline{X_1} \cdot \overline{X_0} \\
Z_1 = \overline{Q_1} \cdot \overline{Q_0} \cdot \overline{X_0} + \overline{Q_1} \cdot \overline{X_1} \cdot \overline{X_0} + Q_1 \cdot Q_0 \cdot X_1 \cdot \overline{X_0} \\
Z_0 = \overline{Q_1} \cdot \overline{X_1} \cdot X_0 + Q_1 \cdot \overline{Q_0} \cdot X_1 \cdot \overline{X_0} + \overline{Q_1} \cdot Q_0 \cdot X_1 \cdot \overline{X_0}
\end{cases}
$$

Correct answer of the question



4. Experiment Week 2

Refinement of the test “Logic Circuit” formal model



状態遷移前の変数値		状態遷移後の変数値	
glasses$0	girl_position$0	glasses	girl_position
on_chair	standing_by_chair	glasses$0	girl_position$0
on_chair	sitting_on_chair	(don't care)	(don't care)
wearing	standing_by_chair	glasses$0	sitting_on_chair
wearing	sitting_on_chair	glasses$0	girl_position$0

コンポーネント	型チェック	証明責務生成	証明	済	未証明
J3test_Rei	OK	OK	9	9	0
J3test_Rei_2r	OK	OK	9	9	0
J3test_Rei_3r	OK	OK	4	4	0
J3test_Rei_r	OK	OK	19	19	0

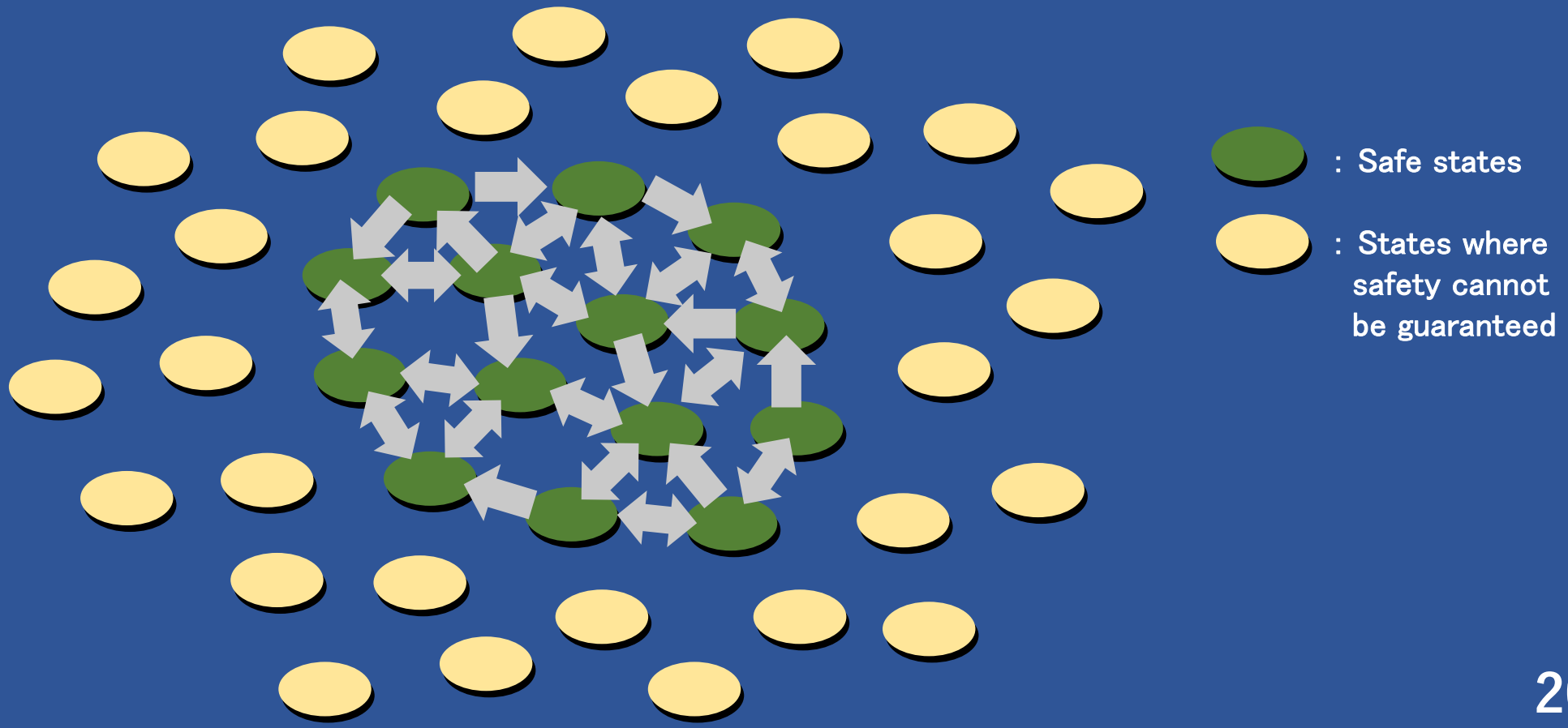
◆ Contents

1. Background
2. Experiment Week 1
3. Our idea 1 : Fountain of logic expressions
4. Experiment Week 2
5. Our idea 2 : Error capable model　Our “better idea” for multi-state formal models
6. Experiment Week 3
7. Students’ questionnaire survey of experiment Week 1-3
8. Our expected direction
9. Summary

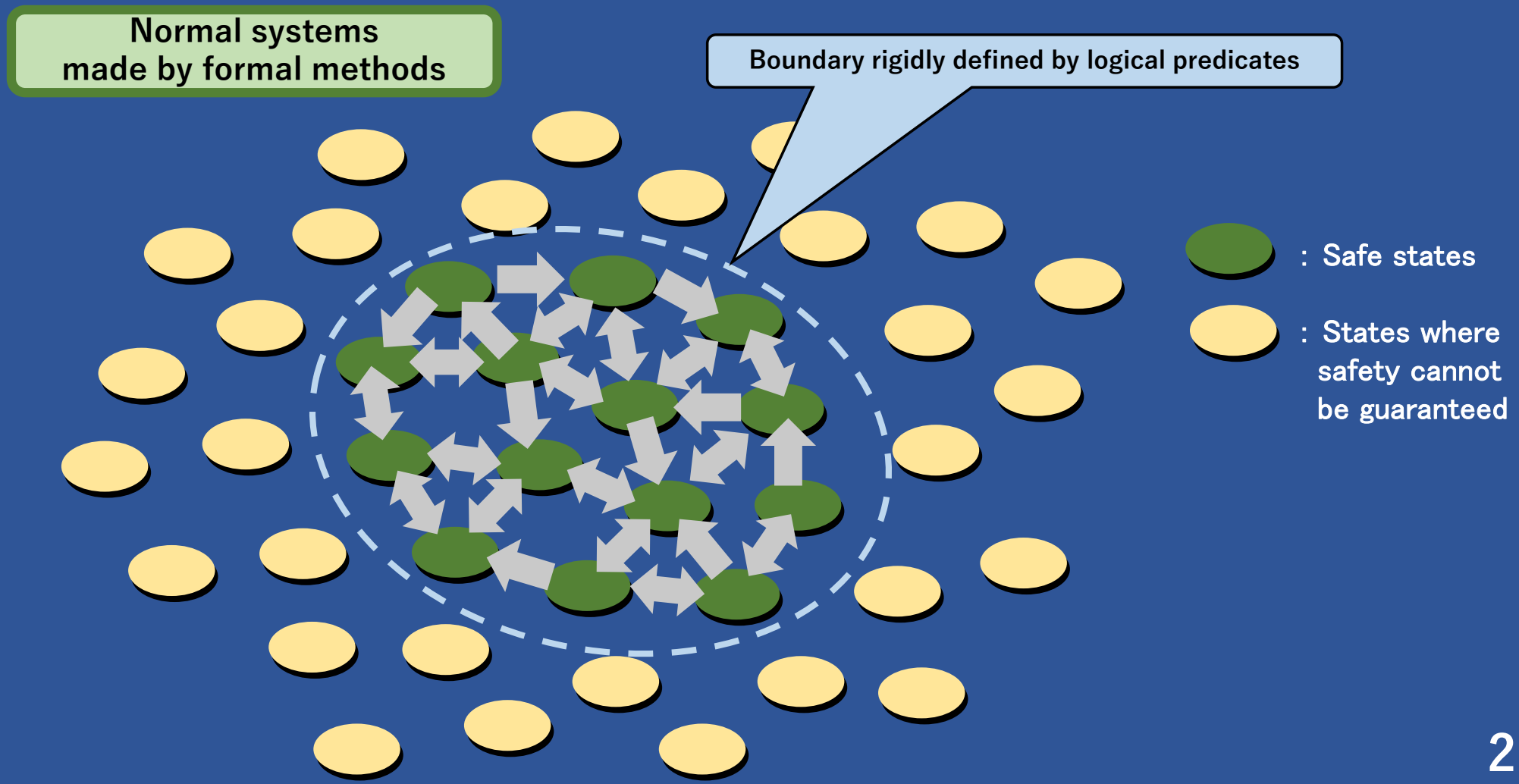
5. Our idea 2 : Error capable model

Normal systems made by formal methods

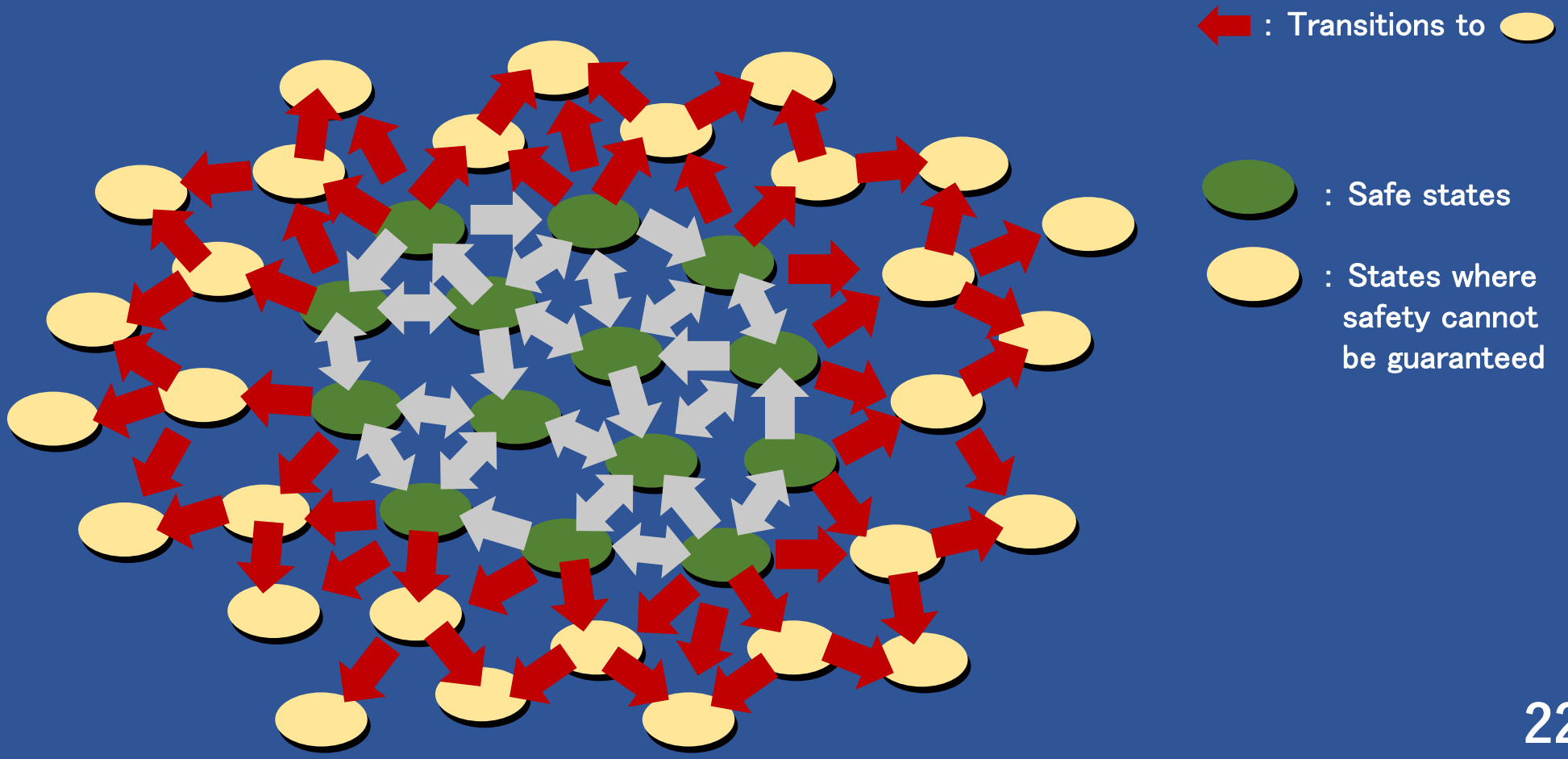
Our “better idea” for multi-state formal models



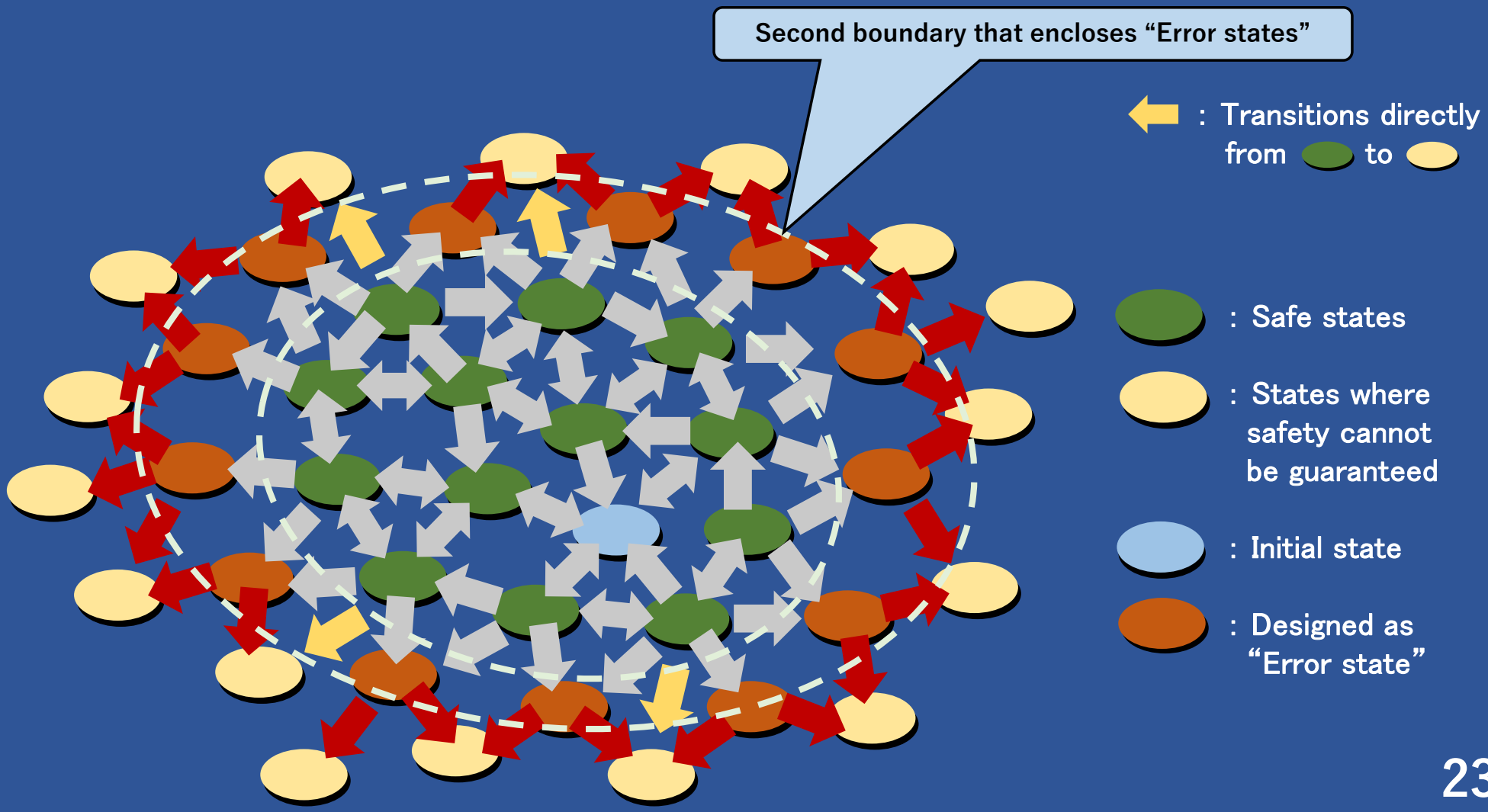
5. Our idea 2 : Error capable model



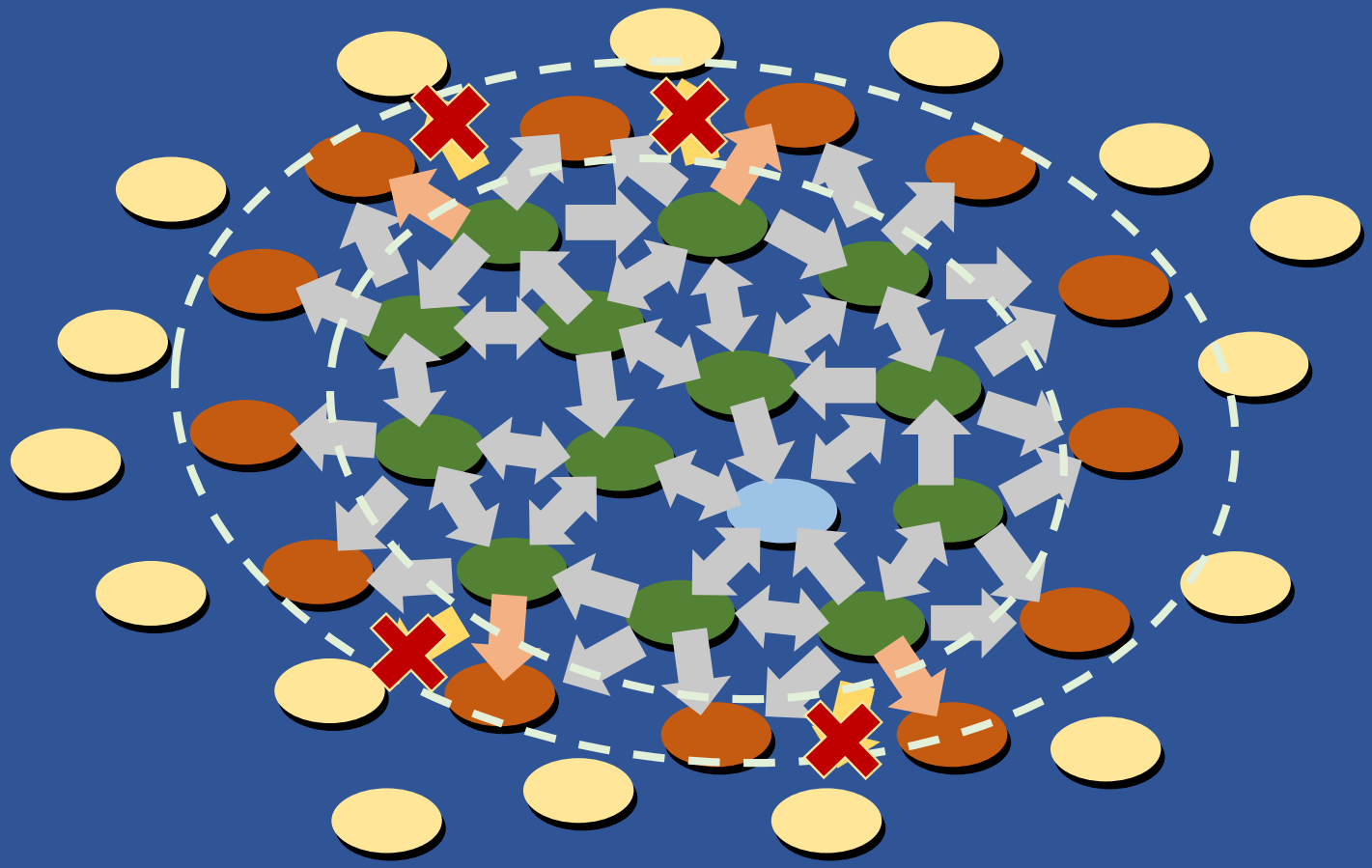
5. Our idea 2 : Error capable model



5. Our idea 2 : Error capable model



5. Our idea 2 : Error capable model



: Suppressed transitions
(from to)
: Redirected transitions
to

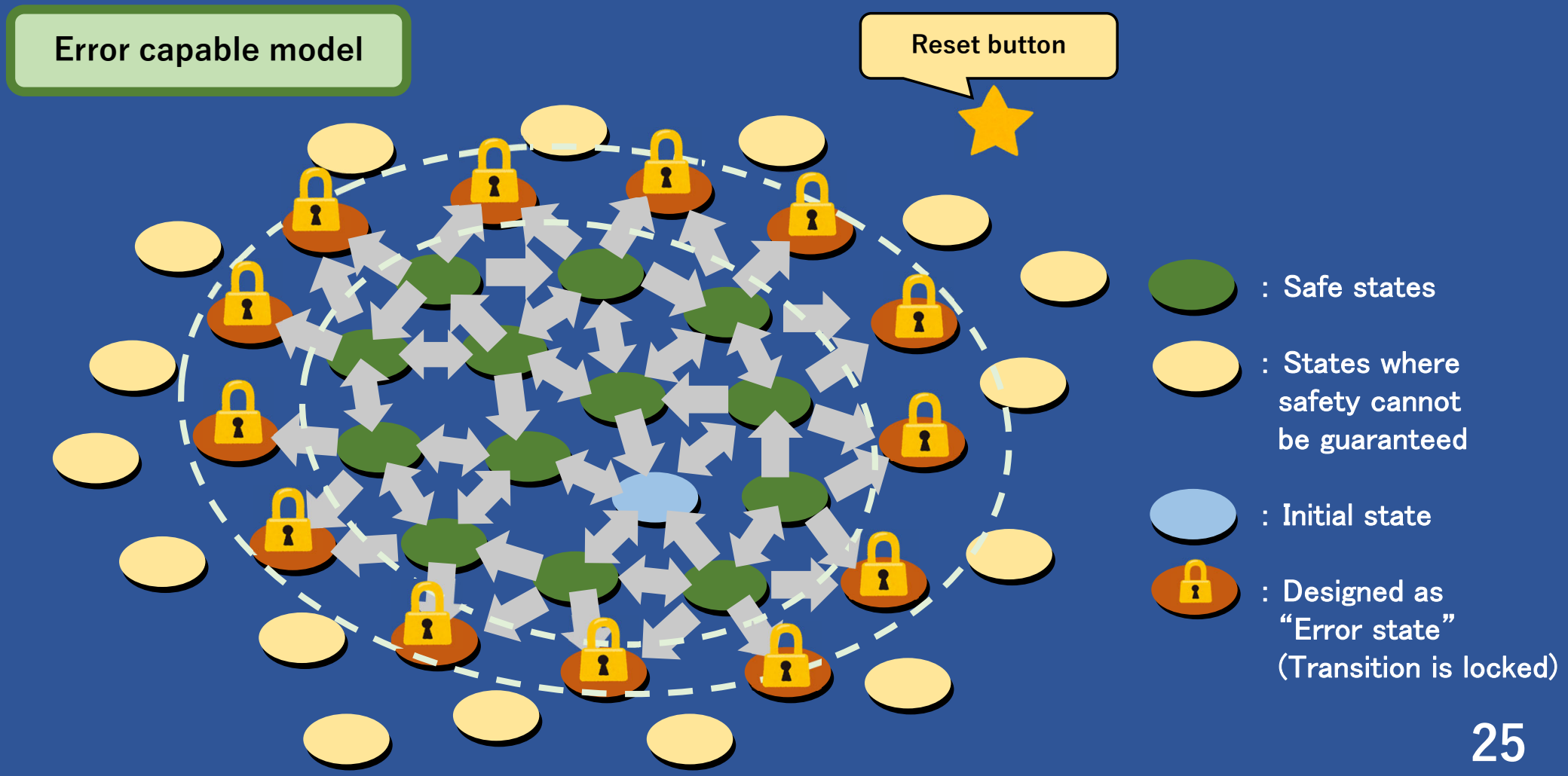
: Safe states

: States where safety cannot be guaranteed

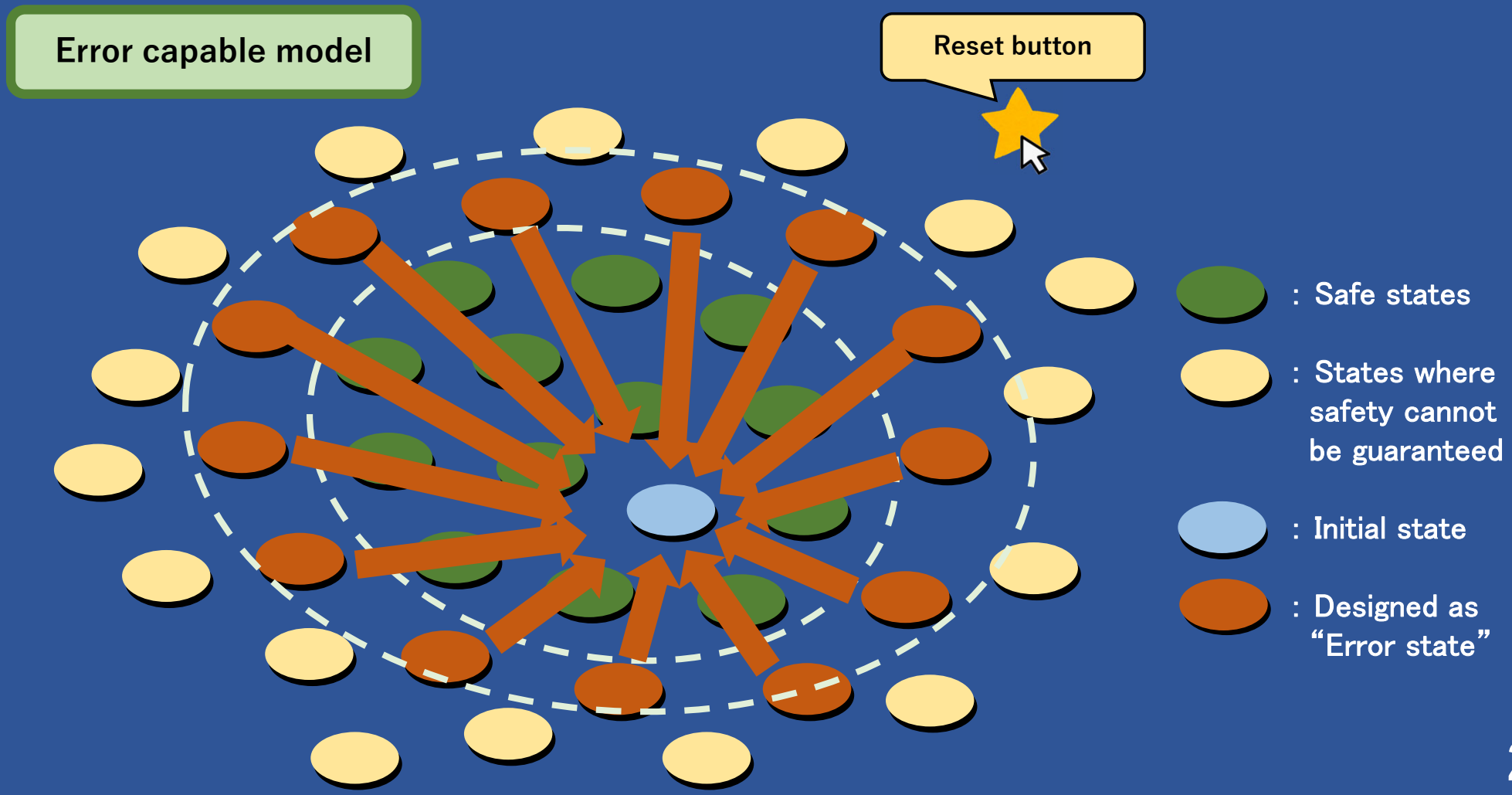
: Initial state

: Designed as “Error state”

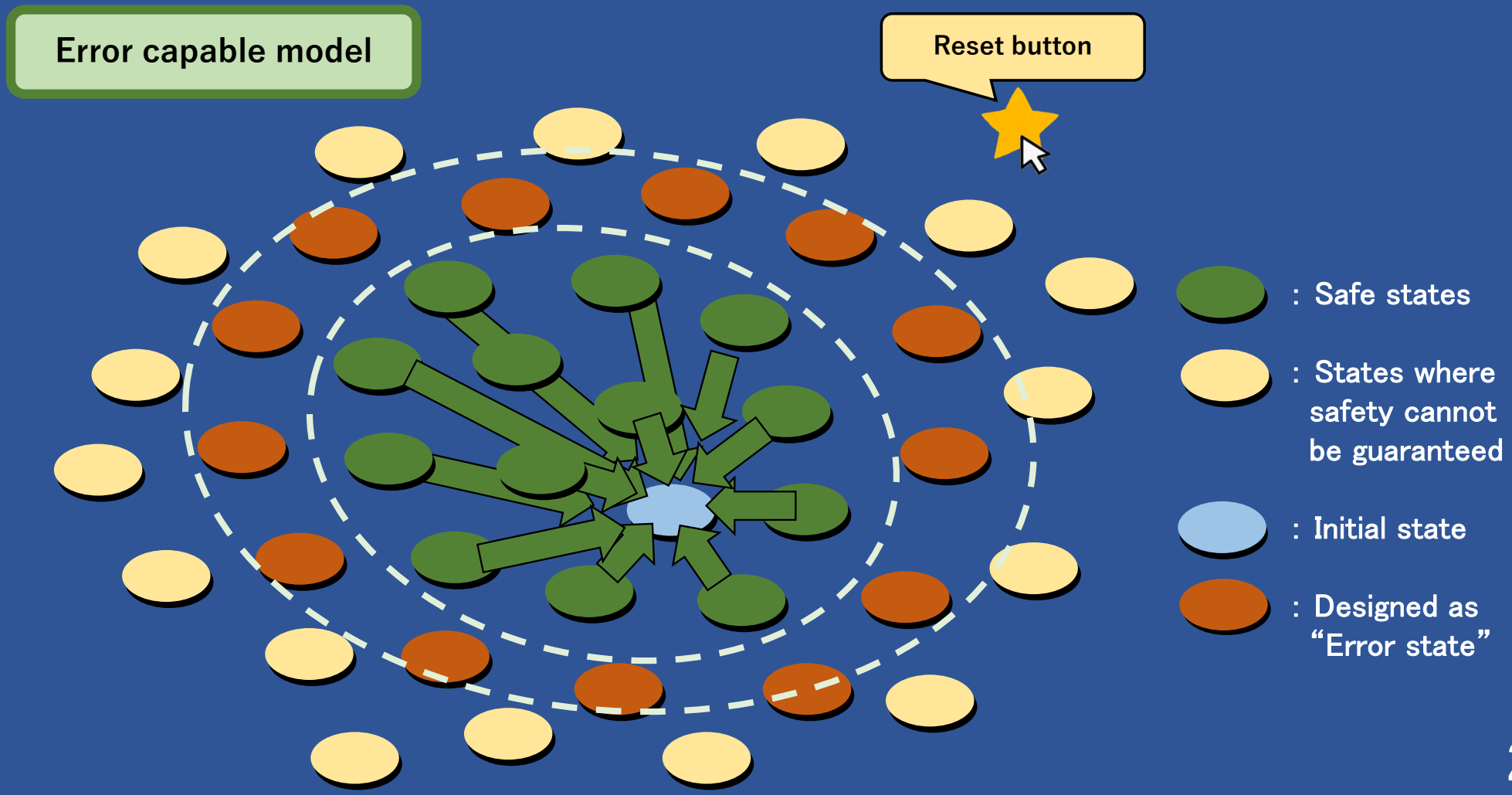
5. Our idea 2 : Error capable model



5. Our idea 2 : Error capable model



5. Our idea 2 : Error capable model

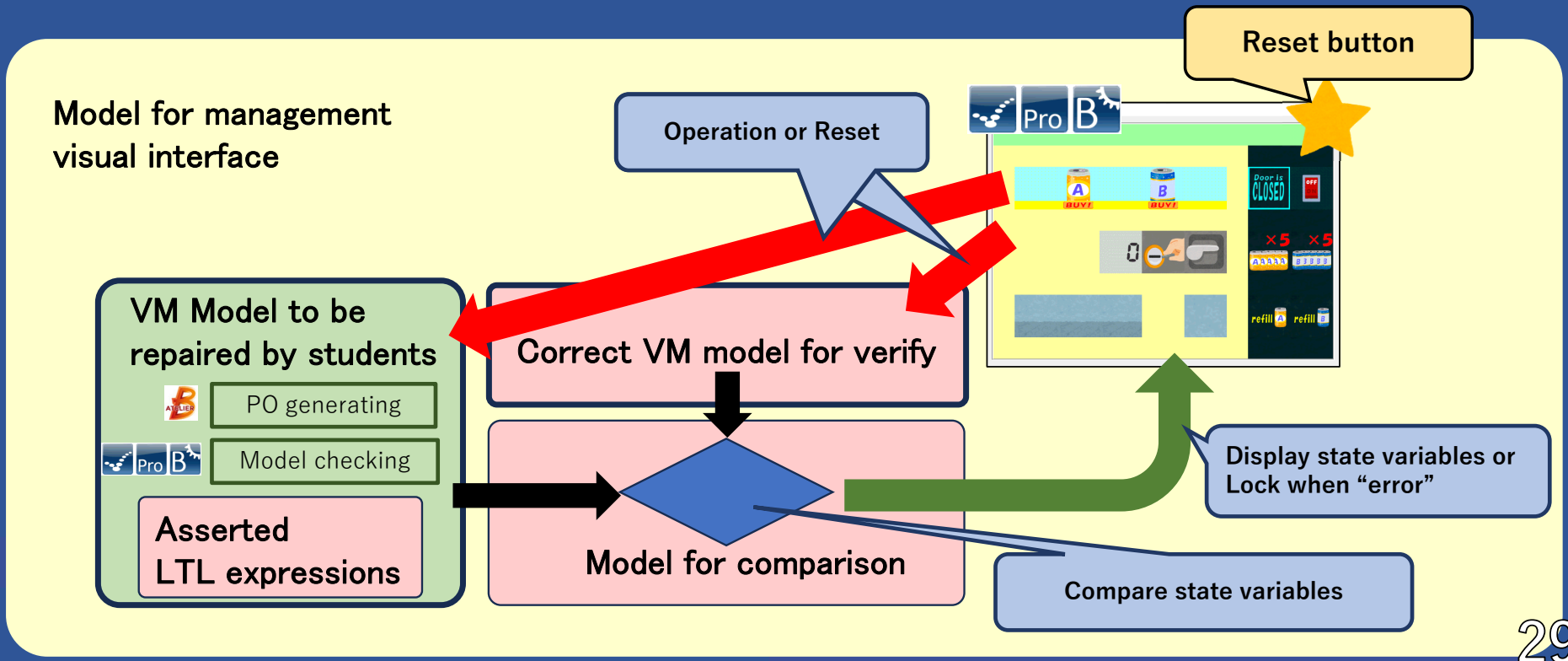


◆ Contents

1. Background
2. Experiment Week 1
3. Our idea 1 : Fountain of logic expressions
4. Experiment Week 2
5. Our idea 2 : Error capable model
6. Experiment Week 3　Applying the concept “error capable model” for VM formal model
7. Students’ questionnaire survey of experiment Week 1-3
8. Our expected direction
9. Summary

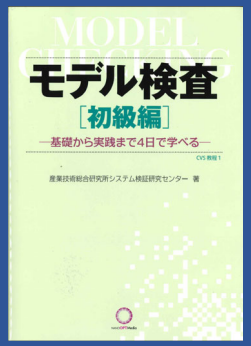
6. Experiment Week 3

Applying the concept “error capable model” for Vending Machine formal model



6. Experiment Week 3

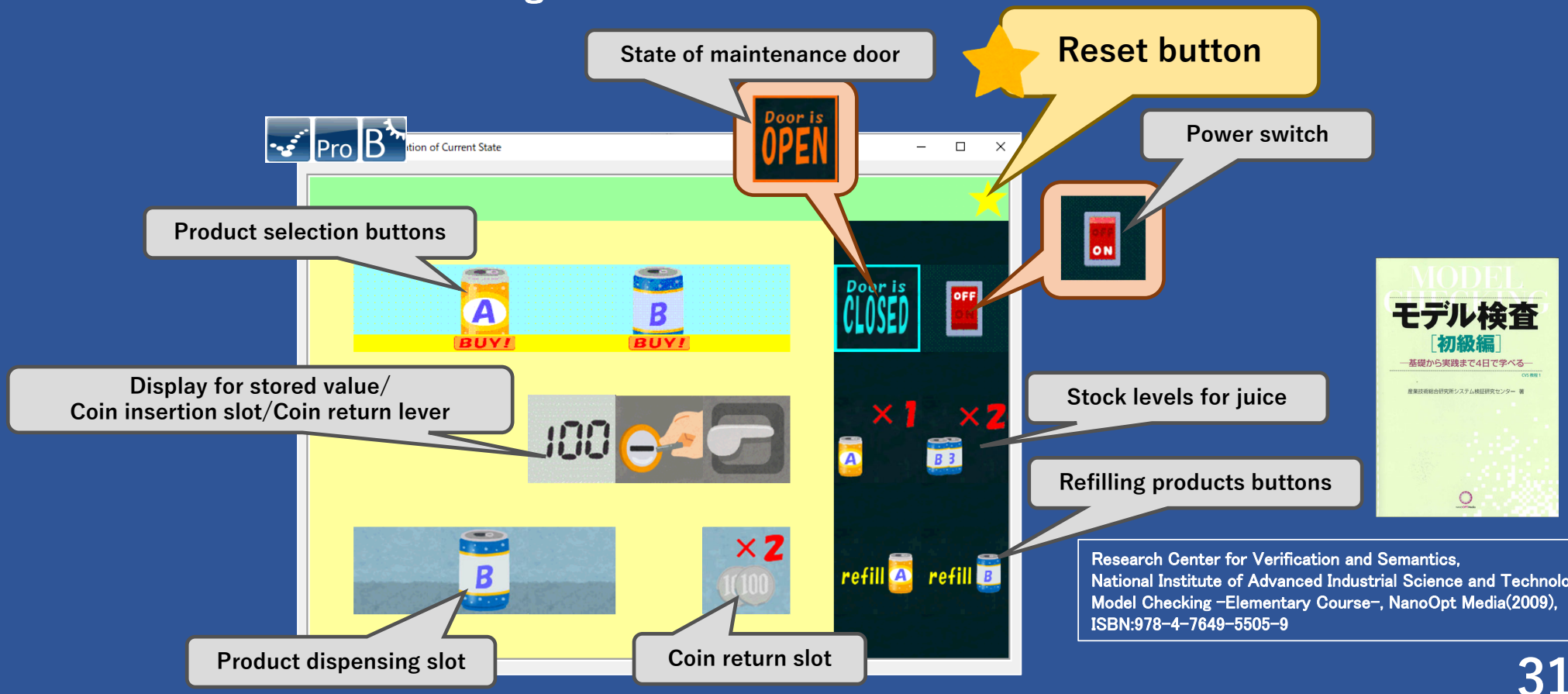
Visual interface of Vending Machine model



Research Center for Verification and Semantics,
National Institute of Advanced Industrial Science and Technology (AIST):
Model Checking –Elementary Course–, NanoOpt Media(2009),
ISBN:978-4-7649-5505-9

6. Experiment Week 3

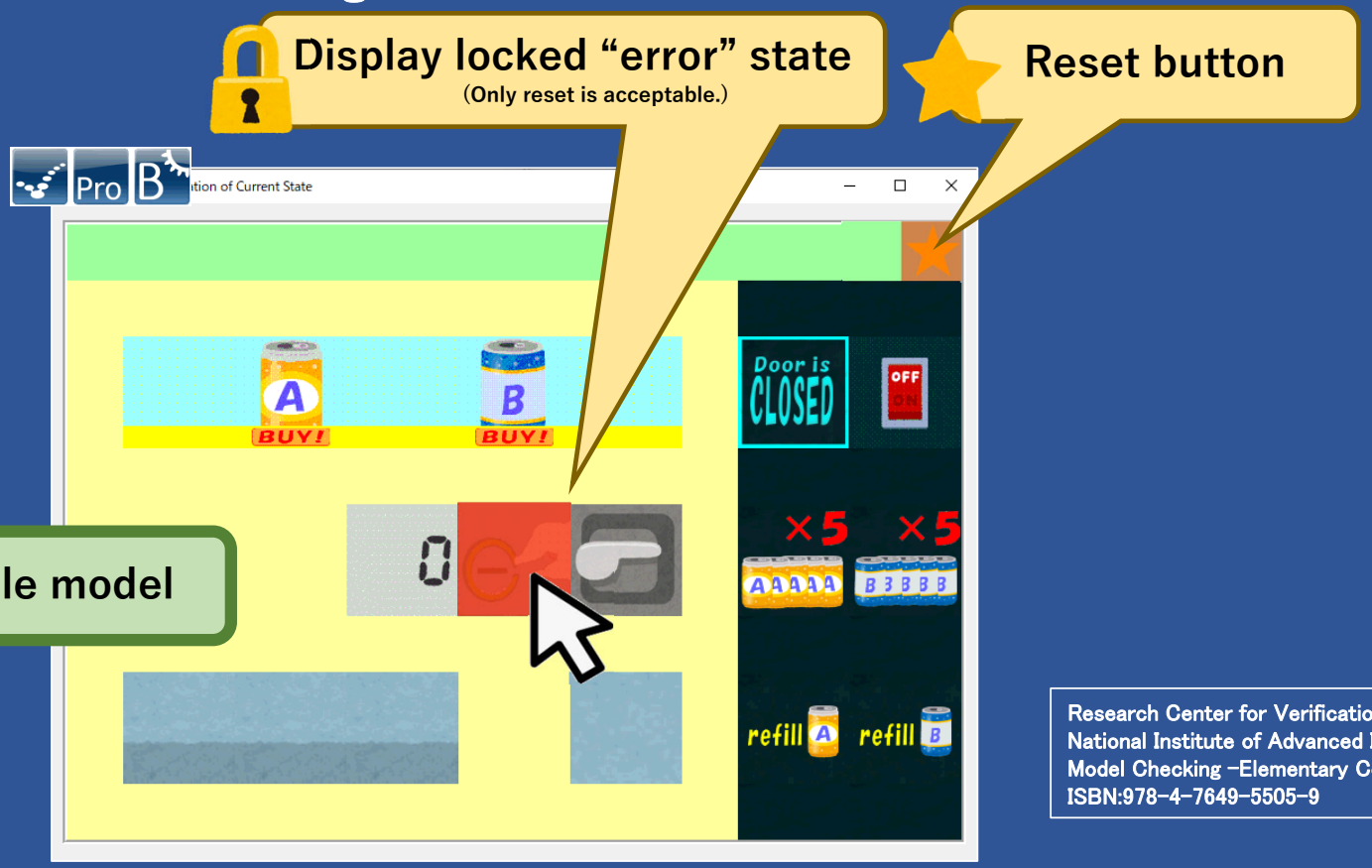
Visual interface of Vending Machine model

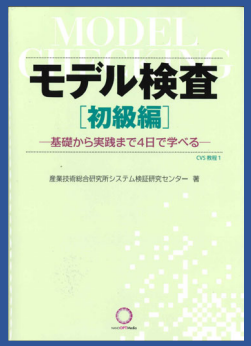


Research Center for Verification and Semantics,
National Institute of Advanced Industrial Science and Technology (AIST):
Model Checking –Elementary Course–, NanoOpt Media(2009),
ISBN:978-4-7649-5505-9

6. Experiment Week 3

Visual interface of Vending Machine model



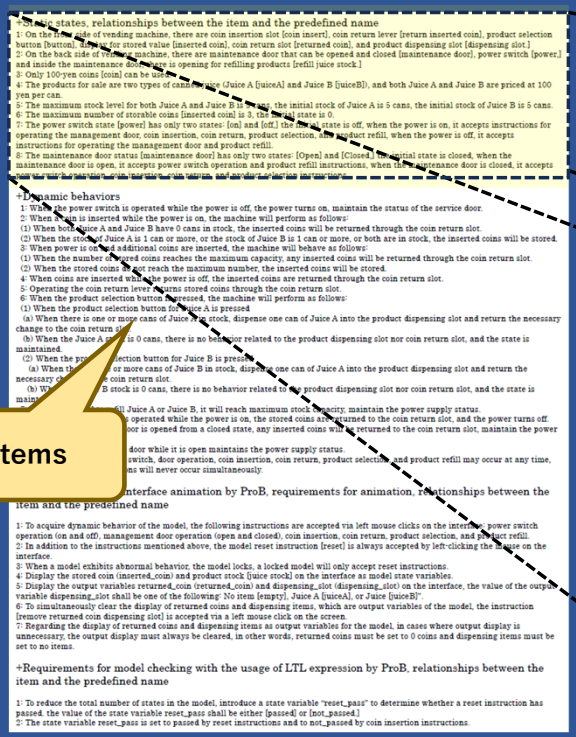


Research Center for Verification and Semantics,
National Institute of Advanced Industrial Science and Technology (AIST):
Model Checking –Elementary Course–, NanoOpt Media(2009),
ISBN:978-4-7649-5505-9

6. Experiment Week 3

Specification of Vending Machine model

- ◆ (8) Static states, relationships between the item and the predefined name
- ◆ (11) Dynamic behaviors
- ◆ (7) Functions of visual interface animation by ProB, requirements for animation, relationships between the item and the predefined name
- ◆ (2) Requirements for model checking with the usage of LTL expression by ProB, relationships between the item and the predefined name



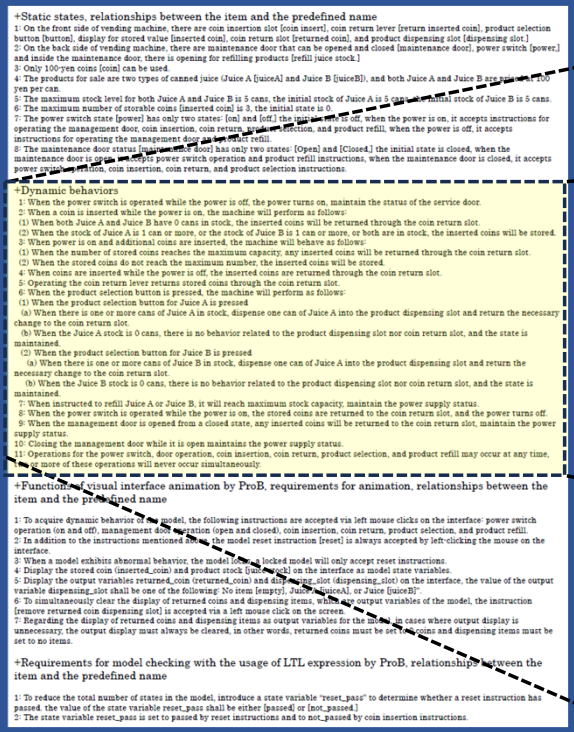
+Static states, relationships between the item and the predefined name
1: On the front side of vending machine, there are coin insertion slot [coin insert], coin return lever [return inserted coin], product selection button [button], display for stored value [inserted coin], coin return slot [returned coin], and product dispensing slot [dispensing slot.]
2: On the back side of vending machine, there are maintenance door that can be opened and closed [maintenance door], power switch [power,] and inside the maintenance door, there is opening for refilling products [refill juice stock.]
3: Only 100-yen coins [coin] can be used.
4: The products for sale are two types of canned juice (Juice A [juiceA] and Juice B [juiceB]), and both Juice A and Juice B are priced at 100 yen per can.
5: The maximum stock level for both Juice A and Juice B is 5 cans, the initial stock of Juice A is 5 cans, the initial stock of Juice B is 5 cans.
6: The maximum number of storable coins [inserted coin] is 3, the initial state is 0.
7: The power switch state [power] has only two states: [on] and [off,] the initial state is off, when the power is on, it accepts instructions for operating the management door, coin insertion, coin return, product selection, and product refill, when the power is off, it accepts instructions for operating the management door and product refill.
8: The maintenance door status [maintenance door] has only two states: [Open] and [Closed,] the initial state is closed, when the maintenance door is open, it accepts power switch operation and product refill instructions, when the maintenance door is closed, it accepts power switch operation, coin insertion, coin return, and product selection instructions.

28 items

6. Experiment Week 3

Specification of Vending Machine model

- (8) Static states, relationships between the item and the predefined name
- **(11) Dynamic behaviors**
- (7) Functions of visual interface animation by ProB, requirements for animation, relationships between the item and the predefined name
- (2) Requirements for model checking with the usage of LTL expression by ProB, relationships between the item and the predefined name



+Static states, relationships between the item and the predefined name
1: On the front side of vending machine, there are coin insertion slot [coin insert], coin return lever [return inserted coin], product selection button [button], display for stored value [inserted coin], coin return slot [returned coin], and product dispensing slot [dispensing slot.]
2: On the back side of vending machine, there are maintenance door that can be opened and closed [maintenance door], power switch [power,] and inside the maintenance door, there is opening for refilling products [refill juice stock.]
3: Only 100-yen coins [coin] can be used.
4: The products for sale are two types of canned juice (Juice A [juiceA] and Juice B [juiceB]), and both Juice A and Juice B are priced at 100 yen per can.
5: The maximum stock level for both Juice A and Juice B is 5 cans, the initial stock of Juice A is 5 cans, and the initial stock of Juice B is 5 cans.
6: The maximum number of storable coins [inserted coin] is 3, the initial state is 0.
7: The power switch state [power] has only two states: [on] and [off,] the initial state is off, when the power is on, it accepts instructions for operating the management door, coin insertion, coin return, product selection, and product refill, when the power is off, it accepts instructions for operating the management door, and product refill.
8: The maintenance door status [maintenance door] has only two states: [Open] and [Closed,] the initial state is closed, when the maintenance door is open, it accepts power switch operation and product refill instructions, when the maintenance door is closed, it accepts power switch operation, coin insertion, coin return, and product selection instructions.

+Dynamic behaviors
1: When the power switch is operated while the power is off, the power turns on, maintain the status of the service door.
2: When a coin is inserted while the power is on, the machine will perform as follows:
(1) When both Juice A and Juice B have 0 cans in stock, the inserted coins will be returned through the coin return slot.
(2) When the stock of Juice A is 1 can or more, or the stock of Juice B is 1 can or more, or both are in stock, the inserted coins will be stored.
3: When power is on and additional coins are inserted, the machine will behave as follows:
(1) When the number of stored coins reaches the maximum capacity, any inserted coins will be returned through the coin return slot.
(2) When the stored coins do not reach the maximum number, the inserted coins will be stored.
4: When coins are inserted while the power is off, the inserted coins are returned through the coin return slot.
5: Operating the coin return lever returns stored coins through the coin return slot.
6: When the product selection button is pressed, the machine will perform as follows:
(1) When the product selection button for Juice A is pressed
(a) When there is one or more cans of Juice A in stock, dispense one can of Juice A into the product dispensing slot and return the necessary change to the coin return slot.
(b) When the Juice A stock is 0 cans, there is no behavior related to the product dispensing slot nor coin return slot, and the state is maintained.
(2) When the product selection button for Juice B is pressed
(a) When there is one or more cans of Juice B in stock, dispense one can of Juice A into the product dispensing slot and return the necessary change to the coin return slot.
(b) When the Juice B stock is 0 cans, there is no behavior related to the product dispensing slot nor coin return slot, and the state is maintained.
7: When instructed to refill Juice A or Juice B, it will reach maximum stock capacity, maintain the power supply status.
8: When the power switch is operated while the power is on, the stored coins are returned to the coin return slot, and the power turns off.
9: When the management door is opened from a closed state, any inserted coins will be returned to the coin return slot, maintain the power supply status.
10: Closing the management door while it is open maintains the power supply status.
11: Operations for the power switch, door operation, coin insertion, coin return, product selection, and product refill may occur at any time, two or more of these operations will never occur simultaneously.

+Functions of visual interface animation by ProB, requirements for animation, relationships between the item and the predefined name

1: To acquire dynamic behaviors of the model, the following instructions are accepted via left mouse clicks on the interface: power switch operation (on and off), management door operation (open and closed), coin insertion, coin return, product selection, and product refill.
2: In addition to the instructions mentioned above, the model reset instruction [reset] is always accepted by left-clicking the mouse on the interface.
3: When a model exhibits abnormal behavior, the model enters a locked model will only accept reset instructions.
4: Display the stored coin (inserted_coin) and product stock [juice_stock] on the interface as model state variables.
5: Display the output variables returned_coin (returned_coin) and dispensing_slot (dispensing_slot) on the interface, the value of the output variable dispensing_slot shall be one of the following: No item [empty], Juice [juiceA], or Juice [juiceB]".
6: To simultaneously clear the display of returned coins and dispensing items, when the output variables of the model, the instruction [remove returned coin dispensing slot] is accepted via a left mouse click on the screen.
7: Regarding the display of returned coins and dispensing items as output variables for the model, in cases where output display is unnecessary, the output display must always be cleared, in other words, returned coins must be set to no coins and dispensing items must be set to no items.

+Requirements for model checking with the usage of LTL expression by ProB, relationships between the item and the predefined name

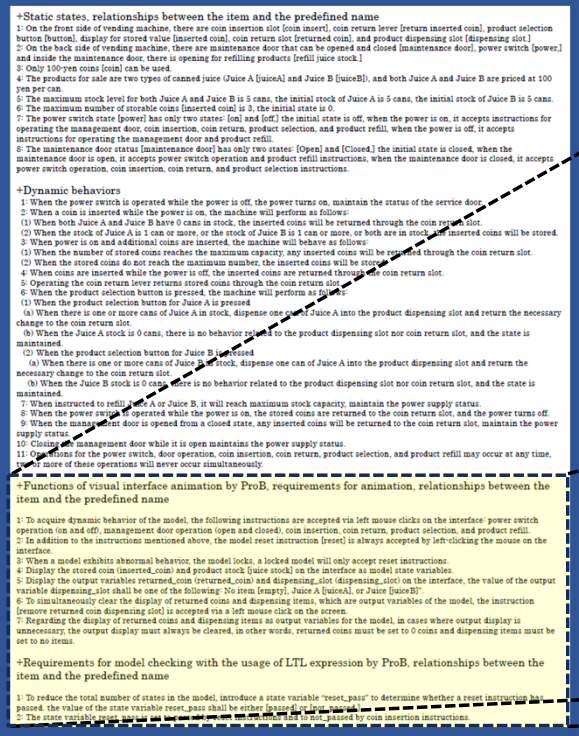
1: To reduce the total number of states in the model, introduce a state variable "reset_pass" to determine whether a reset instruction has passed. the value of the state variable reset_pass shall be either [passed] or [not_passed.]
2: The state variable reset_pass is set to passed by reset instructions and to not_passed by coin insertion instructions.

+Dynamic behaviors
1: When the power switch is operated while the power is off, the power turns on, maintain the status of the service door.
2: When a coin is inserted while the power is on, the machine will perform as follows:
(1) When both Juice A and Juice B have 0 cans in stock, the inserted coins will be returned through the coin return slot.
(2) When the stock of Juice A is 1 can or more, or the stock of Juice B is 1 can or more, or both are in stock, the inserted coins will be stored.
3: When power is on and additional coins are inserted, the machine will behave as follows:
(1) When the number of stored coins reaches the maximum capacity, any inserted coins will be returned through the coin return slot.
(2) When the stored coins do not reach the maximum number, the inserted coins will be stored.
4: When coins are inserted while the power is off, the inserted coins are returned through the coin return slot.
5: Operating the coin return lever returns stored coins through the coin return slot.
6: When the product selection button is pressed, the machine will perform as follows:
(1) When the product selection button for Juice A is pressed
(a) When there is one or more cans of Juice A in stock, dispense one can of Juice A into the product dispensing slot and return the necessary change to the coin return slot.
(b) When the Juice A stock is 0 cans, there is no behavior related to the product dispensing slot nor coin return slot, and the state is maintained.
(2) When the product selection button for Juice B is pressed
(a) When there is one or more cans of Juice B in stock, dispense one can of Juice A into the product dispensing slot and return the necessary change to the coin return slot.
(b) When the Juice B stock is 0 cans, there is no behavior related to the product dispensing slot nor coin return slot, and the state is maintained.
7: When instructed to refill Juice A or Juice B, it will reach maximum stock capacity, maintain the power supply status.
8: When the power switch is operated while the power is on, the stored coins are returned to the coin return slot, and the power turns off.
9: When the management door is opened from a closed state, any inserted coins will be returned to the coin return slot, maintain the power supply status.
10: Closing the management door while it is open maintains the power supply status.
11: Operations for the power switch, door operation, coin insertion, coin return, product selection, and product refill may occur at any time, two or more of these operations will never occur simultaneously.

6. Experiment Week 3

Specification of Vending Machine model

- (8) Static states, relationships between the item and the predefined name
- (11) Dynamic behaviors
- **(7) Functions of visual interface animation by ProB, requirements for animation, relationships between the item and the predefined name**
- **(2) Requirements for model checking with the usage of LTL expression by ProB, relationships between the item and the predefined name**



+Static states, relationships between the item and the predefined name

1: On the front side of vending machine, there are coin insertion slot [coin insert], coin return lever [return inserted coin], product selection button [button], display for stored value [inserted coin], coin return slot [returned coin], and product dispensing slot [dispensing slot.]
2: On the back side of vending machine, there are maintenance door that can be opened and closed [maintenance door], power switch [power,] and inside the maintenance door, there is opening for refilling products [refill juice stock.]
3: Only 100-yen coins [coin] can be used.
4: The products for sale are two types of canned juice (Juice A [juiceA] and Juice B [juiceB]), and both Juice A and Juice B are priced at 100 yen per can.
5: The maximum stock level for both Juice A and Juice B is 5 cans, the initial stock of Juice A is 5 cans, the initial stock of Juice B is 5 cans.
6: The maximum number of storable coins [inserted coin] is 3, the initial state is 0.
7: The power switch state [power] has only two states: [on] and [off,] the initial state is off, when the power is on, it accepts instructions for operating the management door, coin insertion, coin return, product selection, and product refill, when the power is off, it accepts instructions for operating the management door and product refill.
8: The maintenance door status [maintenance door] has only two states: [Open] and [Closed,] the initial state is closed, when the maintenance door is open, it accepts power switch operation and product refill instructions, when the maintenance door is closed, it accepts power switch operation, coin insertion, coin return, and product selection instructions.

+Dynamic behaviors

1: When the power switch is operated while the power is off, the power turns on, maintain the status of the service door.
2: When a coin is inserted while the power is on, the machine will perform as follows:
(1) When both Juice A and Juice B have 0 cans in stock, the inserted coins will be returned through the coin return slot.
(2) When the stock of Juice A is 1 can or more, or the stock of Juice B is 1 can or more, or both are in stock, the inserted coins will be stored.
3: When power is on and additional coins are inserted, the machine will behave as follows:
(1) When the number of stored coins reaches the maximum capacity, any inserted coins will be returned through the coin return slot.
(2) When the stored coins do not reach the maximum number, the inserted coins will be stored.
4: When coins are inserted while the power is off, the inserted coins are returned through the coin return slot.
5: Operating the coin return lever returns stored coins through the coin return slot.
6: When the product selection button is pressed, the machine will perform as follows:
(1) When the product selection button for Juice A is pressed
(a) When there is one or more cans of Juice A in stock, dispense one can of Juice A into the product dispensing slot and return the necessary change to the coin return slot.
(b) When the Juice A stock is 0 cans, there is no behavior related to the product dispensing slot nor coin return slot, and the state is maintained.
(2) When the product selection button for Juice B is pressed
(a) When there is one or more cans of Juice B in stock, dispense one can of Juice A into the product dispensing slot and return the necessary change to the coin return slot.
(b) When the Juice B stock is 0 cans, there is no behavior related to the product dispensing slot nor coin return slot, and the state is maintained.
7: When instructed to refill Juice A or Juice B, it will reach maximum stock capacity, maintain the power supply status.
8: When the power switch is operated while the power is on, the stored coins are returned to the coin return slot, and the power turns off.
9: When the management door is opened from a closed state, any inserted coins will be returned to the coin return slot, maintain the power supply status.
10: Closing the management door while it is open maintains the power supply status.
11: Operations for the power switch, door operation, coin insertion, coin return, product selection, and product refill may occur at any time, two or more of these operations will never occur simultaneously.

+Functions of visual interface animation by ProB, requirements for animation, relationships between the item and the predefined name

1: To acquire dynamic behavior of the model, the following instructions are accepted via left mouse clicks on the interface: power switch operation (on and off), management door operation (open and closed), coin insertion, coin return, product selection, and product refill.
2: In addition to the instructions mentioned above, the model reset instruction [reset] is always accepted by left-clicking the mouse on the interface.
3: When a model exhibits abnormal behavior, the model locks, a locked model will only accept reset instructions.
4: Display the stored coin (inserted_coin) and product stock [juice stock] on the interface as model state variables.
5: Display the output variables returned_coin (returned_coin) and dispensing_slot (dispensing_slot) on the interface, the value of the output variable dispensing_slot shall be one of the following: No item [empty], Juice A [juiceA], or Juice [juiceB]".
6: To simultaneously clear the display of returned coins and dispensing items, which are output variables of the model, the instruction [remove returned coin dispensing slot] is accepted via a left mouse click on the screen.
7: Regarding the display of returned coins and dispensing items as output variables for the model, in cases where output display is unnecessary, the output display must always be cleared, in other words, returned coins must be set to 0 coins and dispensing items must be set to no items.

+Requirements for model checking with the usage of LTL expression by ProB, relationships between the item and the predefined name

1: To reduce the total number of states in the model, introduce a state variable "reset_pass" to determine whether a reset instruction has passed. the value of the state variable reset_pass shall be either [passed] or [not_passed.]
2: The state variable reset_pass is set to passed by reset instructions and to not_passed by coin insertion instructions.

2 additional items to introduce “error capable model”

6. Experiment Week 3

Variables of Vending Machine model

- ◆ (5) State variables of Vending Machine
- ◆ (2) Output variables of Vending Machine
- ◆ (1) State variables for model checking

name of variable	values	meaning	notes
power	on, off	the power supply status	state variable of the model
juiceA_stock	0, 1, 2, 3, 4, 5	Juice A stock	state variable of the model
juiceB_stock	0, 1, 2, 3, 4, 5	Juice B stock	state variable of the model
inserted_coin	0, 1, 2, 3	the number of stored coins	state variable of the model
returned_coin	0, 1, 2, 3	display of returned coins	output variable of the model
dispensing_slot	empty, juiceA, juiceB	display of dispensing items	output variable of the model
maintenance_door	open, closed	the management door status	state variable of the model
reset_pass	passed, not_passed	the state for reset instruction passed	state variable for model checking

1 additional variable to introduce “error capable model”

6. Experiment Week 3

Operations of Vending Machine model

◆ (1) Operation for animation
◆ (11) Operations for Vending Machine

1 additional operation to introduce “error capable model”

name of operation	meaning	notes
reset	reset the model	operation for the animation
power_turn_on	turn on the power	
power_turn_off_return_inserted_coin	turn off the power	
coin_insert	insert a coin	
return_inserted_coin	return the stocked coin	
remove_returned_coin_dispensing_slot	remove the returned coins and dispensing items from the slot	
press_juiceA_button	select juice A	
press_juiceB_button	select juice B	
refill_juiceA_stock	refill juice A to the maximum level	
refill_juiceB_stock	refill juice B to the maximum level	
open_maintenence_door	open the maintenance door	
close_maintenance_door	close the maintenance door	

6. Experiment Week 3

Asserted LTL expressions for Vending Machine model

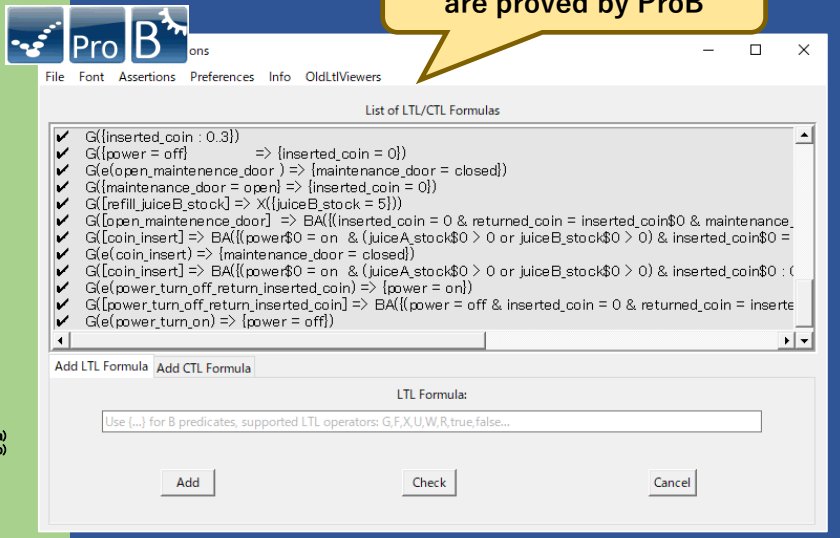
- ◆ (34) Check items made by the authors
- ◆ (13) Check items written in the textbook

Research Center for Verification and Semantics,
National Institute of Advanced Industrial Science and Technology (AIST):
Model Checking –Elementary Course–, NanoOpt Media(2009),
ISBN:978-4-7649-5505-9

47 LTL expressions (some overlap)

```
DEFINITIONS
    ASSERT_LTL1  = "G([power = off]            => [inserted_coin = 0])";
    ASSERT_LTL2  = "G([maintenance_door = open] => [inserted_coin = 0])";
    ASSERT_LTL3  = "G(e(power_turn_on) => [power = off])";
    ASSERT_LTL4  = "G([power_turn_on] => X([power = on]))";
    ASSERT_LTL5  = "G(e(power_turn_off_return_inserted_coin) => [power = on])";
    ASSERT_LTL6  = "G([power_turn_off_return_inserted_coin] => BA({(power = off & inserted_coin = 0 & returned_coin = inserted_coin$0)}))";
    ASSERT_LTL7  = "G(e(coin_insert) => [maintenance_door = closed])";
    ASSERT_LTL8  = "G([coin_insert] => BA({(power$0 = on  & (juiceA_stock$0 > 0 or juiceB_stock$0 > 0) & inserted_coin$0 : 0..2) => (inserted_coin = inserted_coin$0 + 1                )}))";
    ASSERT_LTL9  = "G([coin_insert] => BA({(power$0 = on  & (juiceA_stock$0 > 0 or juiceB_stock$0 > 0) & inserted_coin$0 = 3   ) => (inserted_coin = inserted_coin$0    & returned_coin = 1)}))";
    ASSERT_LTL10 = "G([coin_insert] => BA({(power$0 = on  &  juiceA_stock$0 = 0 &  juiceB_stock$0 = 0                        ) => (inserted_coin = inserted_coin$0    & returned_coin = 1)}))";
    ASSERT_LTL11 = "G([coin_insert] => BA({ power$0 = off                                                                    => (inserted_coin = inserted_coin$0    & returned_coin = 1)}))";
    ASSERT_LTL12 = "G([coin_insert] => BA({(returned_coin$0 = 0) => (inserted_coin$0 + returned_coin$0 + 1 = inserted_coin + returned_coin)}))";
    ASSERT_LTL13 = "G(e(return_inserted_coin)=>[power = on])";
    ASSERT_LTL14 = "G([return_inserted_coin] => BA({(inserted_coin = 0 & returned_coin = inserted_coin$0)}))";
    ASSERT_LTL15 = "G([return_inserted_coin] => BA({(returned_coin$0 = 0) => (inserted_coin$0 + returned_coin$0 = inserted_coin + returned_coin)}))";
    ASSERT_LTL16 = "G(e(remove_returned_coin_dispensing_slot) => [returned_coin /= 0 or dispensing_slot /= empty])";
    ASSERT_LTL17 = "G([remove_returned_coin_dispensing_slot] => X([returned_coin = 0 & dispensing_slot = empty]))";
    ASSERT_LTL18 = "G(e(press_juiceA_button)=>[power = on & juiceA_stock  > 0 & inserted_coin > 0 & maintenance_door = closed])";
    ASSERT_LTL19 = "G([press_juiceA_button] => BA({inserted_coin = 0 & returned_coin = inserted_coin$0 - 1 & juiceA_stock = juiceA_stock$0 - 1 & dispensing_slot = juiceA}))";
    ASSERT_LTL20 = "G([press_juiceA_button] => BA({(returned_coin$0 = 0) => (inserted_coin$0 + returned_coin$0 - 1 = inserted_coin + returned_coin)}))";
    ASSERT_LTL21 = "G(e(press_juiceB_button)=>[power = on & juiceB_stock  > 0 & inserted_coin > 0 & maintenance_door = closed])";
    ASSERT_LTL22 = "G([press_juiceB_button] => BA({inserted_coin = 0 & returned_coin = inserted_coin$0 - 1 & juiceB_stock = juiceB_stock$0 - 1 & dispensing_slot = juiceB}))";
    ASSERT_LTL23 = "G([press_juiceB_button] => BA({(returned_coin$0 = 0) => (inserted_coin$0 + returned_coin$0 - 1 = inserted_coin + returned_coin)}))";
    ASSERT_LTL24 = "G(e(refill_juiceA_stock) => [maintenance_door = open])";
    ASSERT_LTL25 = "G([refill_juiceA_stock] => X([juiceA_stock = 5]))";
    ASSERT_LTL26 = "G(e(refill_juiceB_stock) => [maintenance_door = open])";
    ASSERT_LTL27 = "G([refill_juiceB_stock] => X([juiceB_stock = 5]))";
    ASSERT_LTL28 = "G(e(open_maintenance_door ) => [maintenance_door = closed])";
    ASSERT_LTL29 = "G([open_maintenance_door]  => BA({(inserted_coin = 0 & returned_coin = inserted_coin$0 & maintenance_door = open   )}))";
    ASSERT_LTL30 = "G([open_maintenance_door]  => BA({(returned_coin$0 = 0) => (inserted_coin$0 + returned_coin$0 = inserted_coin + returned_coin)}))";
    ASSERT_LTL31 = "G(e(close_maintenance_door) => [maintenance_door = open  ])";
    ASSERT_LTL32 = "G([close_maintenance_door] => X([                                                maintenance_door = closed  ]))";
    ASSERT_LTL33 = "G([inserted_coin > 0] & [reset_pass = not_passed] => F([returned_coin > 0] or [dispensing_slot /= empty] or [reset_pass = passed]))";
    ASSERT_LTL34 = "([returned_coin = 0]&[dispensing_slot = empty])U[inserted_coin = 0]";

    ASSERT_LTL35 = "G([power_turn_on] => BA({(power$0 = off) => (power = on)}))"; // check01
    ASSERT_LTL36 = "G([coin_insert] => BA({(power$0 = on & juiceA_stock$0 = 0 & juiceB_stock$0 = 0) => (returned_coin = 1)}))"; // ckeck02
    ASSERT_LTL37 = "G([coin_insert] => BA({(power$0 = on & (juiceA_stock$0 > 0 or juiceB_stock$0 > 0)) => inserted_coin > 0}))"; // check03
    ASSERT_LTL38 = "G([coin_insert] => BA({(power$0 = on & (juiceA_stock$0 > 0 or juiceA_stock$0 > 0) & inserted_coin$0 = 3) => (returned_coin = 1)}))"; // check04
    ASSERT_LTL39 = "G([coin_insert] => BA({(inserted_coin$0 = 3) => (returned_coin = 1)}))"; // check05
    ASSERT_LTL40 = "G([press_juiceA_button] => BA({(inserted_coin$0 > 0 & juiceA_stock$0 > 0) => (juiceA_stock = juiceA_stock$0 - 1 & returned_coin = inserted_coin$0 - 1 & dispensing_slot = juiceA)}))"; // check06(a)
    ASSERT_LTL41 = "G([press_juiceB_button] => BA({(inserted_coin$0 > 0 & juiceB_stock$0 > 0) => (juiceB_stock = juiceB_stock$0 - 1 & returned_coin = inserted_coin$0 - 1 & dispensing_slot = juiceB)}))"; // check06(b)
    ASSERT_LTL42 = "G([coin_insert] => BA({(power$0 = off ) => (returned_coin = 1)}))"; // check07
    ASSERT_LTL43 = "G([power_turn_off_return_inserted_coin] => BA({(power$0 = on) => (power = off)}))"; // check08
    ASSERT_LTL44 = "G([power_turn_off_return_inserted_coin] => BA({(inserted_coin$0 > 0) => (power = off & returned_coin = inserted_coin$0)}))"; // check09
    ASSERT_LTL45 = "G([inserted_coin > 0] & [reset_pass = not_passed]=> F([returned_coin > 0] or [dispensing_slot /= empty] or [reset_pass = passed]))"; // check10
    ASSERT_LTL46 = "G([(power = off) => (inserted_coin = 0)]))"; // check11
    ASSERT_LTL47 = "G([inserted_coin : 0..3])"; // check12
```

All LTL expressions are proved by ProB



6. Experiment Week 3

A check item and an LTL expression for Vending Machine model

- ◆ (34) Check items made by the authors
- ◆ (13) Check items written in the textbook

Research Center for Verification and Semantics,
National Institute of Advanced Industrial Science and Technology (AIST):
Model Checking –Elementary Course–, NanoOpt Media(2009),
ISBN:978-4-7649-5505-9

Check item 6: When the product selection button is pressed, the machine will perform as follows:
(1)(a):When the product selection button for Juice A is pressed: When there is one or more cans of Juice A in stock,
dispense one can of Juice A into the product dispensing slot and return the necessary change to the coin return slot.

```
DEFINITIONS
    ASSERT_LTL40 ==
      "G([press_juiceA_button] =>
          BA({(inserted_coin$0 > 0 & juiceA_stock$0 > 0) =>
               (juiceA_stock = juiceA_stock$0 - 1 & returned_coin = inserted_coin$0 - 1 & dispensing_slot = juiceA)})";
```

(Meaning of ASSERT_LTL40):
When the operation "press_juiceA_button" is executed, the preconditions and post conditions written below are always satisfied:
Precondition: "Inserted coin is greater than zero" and "Stock of Juice A is greater than zero."
Post condition: "Stock of Juice A is reduced by one" and
"Number of returned coins is equal to the number of inserted coins minus one" and
"Juice A in present in the product dispensing slot."

6. Experiment Week 3

An LTL expression for Vending Machine model

```
DEFINITIONS
    ASSERT_LTL40 ==
      "G([press_juiceA_button] =>
          BA([(inserted_coin$0 > 0 & juiceA_stock$0 > 0) =>
               (juiceA_stock = juiceA_stock$0 - 1 & returned_coin = inserted_coin$0 - 1 & dispensing_slot = juiceA)])])";
```

(Meaning of ASSERT_LTL40):
When the operation "press_juiceA_button" is executed, the preconditions and post conditions written below are always satisfied:
Precondition: "Inserted coin is greater than zero" and "Stock of Juice A is greater than zero."
Post condition: "Stock of Juice A is reduced by one" and
"Number of returned coins is equal to the number of inserted coins minus one" and
"Juice A in present in the product dispensing slot."

With the usage of Before-After temporal logic operator (BA), we can write LTL expressions with preconditions/post conditions since ProB 1.12.

表１　LTL 式の様相記号

様相記号	語源	意味
G(P)	Globally	モデルの全ての状態において、常に術語 P が成立する。
F(P)	Finally	モデルの状態遷移において、いずれ術語 P が成立する。
X(P)	neXt	モデルの状態遷移において、次の遷移の際に術語 P が成立する。
(P)U(Q)	Until	モデルの状態遷移において、術語 Q が成り立つまでは常に術語 P が成立する。
BA(P)	Before After	モデルの状態遷移において、遷移前-遷移後術語 P が成立する。(ProB のみ) 〔遷移前-遷移後術語Pにおいて、遷移前変数X$0と遷移後変数Xを活用する。〕

表３　ProB における LTL 式の記法

LTL 式の記法	意味・用途
G({P})	全ての状態において、常に術語 P が成立する。 （INVARIANT 節における術語（不変条件）の成立に相当。）
G(e(ope) => {P})	操作 ope において、術語 P は必要な事前条件である。 （操作 ope における PRE-THEN-END による事前条件の術語の一部である。）
G({P} => e(ope))	操作 ope において、術語 P は十分な事前条件である。 （操作 ope における PRE-THEN-END による事前条件の術語の全てである。）
G([ope] => X({P}))	操作 ope において、術語 P は事後条件である。
G([ope] => BA({P})	操作 ope において、遷移前-遷移後術語 P は事後条件である。 〔遷移前-遷移後術語 P において、遷移前変数 X$0 と遷移後変数 X を活用する。〕
G([ope] => BA({P => Q})	操作 ope において、遷移前-遷移後術語「P=>Q」は事前事後条件である。 つまり術語 P は事前条件であり、術語 Q は事後条件である。 〔遷移前-遷移後術語「P=>Q」において、遷移前変数 X$0 と遷移後変数 X を活用する。 術語 P は遷移前変数 X$0 と遷移不変の変数のみで構成されるべきであり、 術語 Q は遷移前変数 X$0、遷移後変数 X、遷移不変の変数で構成される。〕

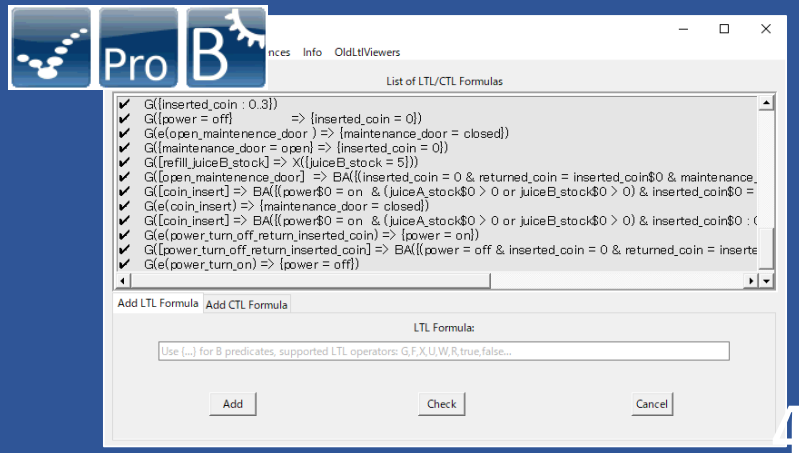
6. Experiment Week 3

Procedure of experiment

- Understand the behavior of correct model
- Be passed the broken model
- Check the behavior of broken model
  - ・PO generating (Atelier B)
  - ・LTL expressions (ProB)
- Repair the model
- Note all the process of repairing

In early model checking, students can observe the state explosion.

コンポーネント	型チェック	証明責務生成	証明責務	証明済	未証明	B0チェック
VM	OK	OK	29	29	0	OK
VM_r	OK	OK	29	29	0	OK
VM_visual	OK	OK	0	0	0	OK
VM_visual_r	OK	OK	797	776	21	OK



6. Experiment Week 3

Procedure of experiment

Understand the behavior of correct model

↓

Be passed the broken model

↓

Check the behavior of broken model
- PO generating (Atelier B)
- LTL expressions (ProB)

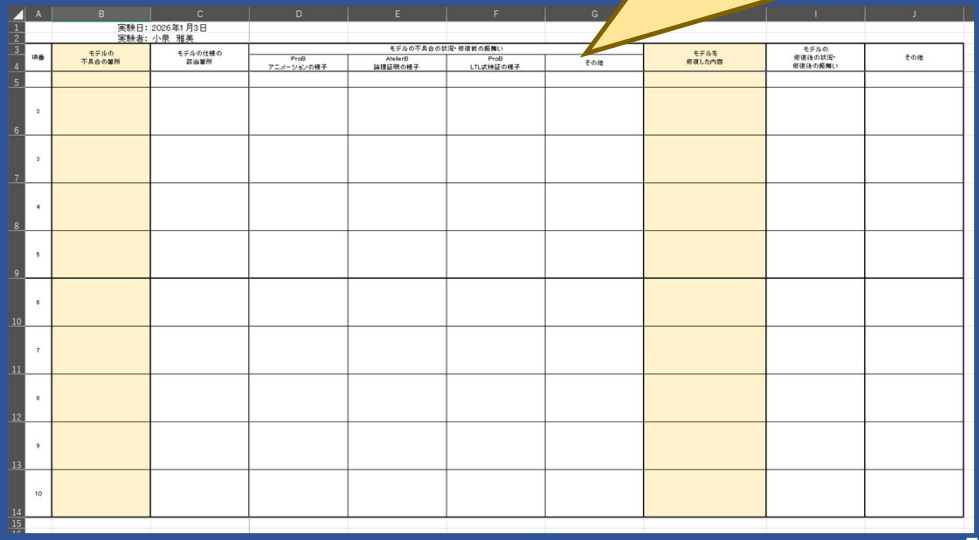
↓

Repair the model

↓

Note all the process of repairing

Note all the process of repairing, using Excel, and report.
→ How to detect defect locations
- ProB animation interface
- Atelier B PO generation
- ProB LTL expression proving



6. Experiment Week 3

Scene of experiment Week 3






6. Experiment Week 3

A sample of student's report

process of repairing VM model

実験日：2025年7月24日
実験者：

項番	モデルの不具合の箇所	モデルの仕様の該当箇所	モデルの不具合の状況・修復前の振舞い				モデルを修復した内容	モデルの修復後の状況・修復後の振舞い	その他
			ProB アニメーションの様子	AtelierB 論理証明の様子	ProB LTL式検証の様子	その他			
1	電源がONの状態で、4枚以上硬貨を投入することができる箇所	6. 貯蔵可能な硬貨の最大枚数は3枚までである。	硬貨を4枚以上投入すると、フリーズする	Operation_coin_insertにおけるPO1にてエラーが発生した。	G([inserted_coin : 0..3])が成立しない。		貯蔵しているコインの枚数における3枚の場合の処理を追加した。すなわち、コインが2枚までの場合と3枚目の時の処理を分割した。	コインを4枚目投入すると、返却口にコインが出るようになった。	
2	OPELATION節における press_juiceA_bottonにおいて、juiceAの在庫が無くなった場合と硬貨を投入せずにjuiceAを購入しようとした場合の処理がない	仕様の『動的な振る舞い』6.(1)(b)および11.に該当する	juiceAの在庫がなくなった場合・硬貨を投入せずにjuiceAを購入しようとした場合、フリーズする	Operation_press_juiceAbuttonにおけるPO1, PO2にてエラーが発生した。			juiceAにおけるstrockを>0とし、inserted_coinを>0とすることで、解決できた	juiceAにおける在庫がなくなった場合の処理及びjuiceAを購入しようとした場合の処理ができるようになった。	
3	OPELATION節における press_juiceB_bottonにおいて、硬貨を投入した際の処理がない	仕様の『動的な振る舞い』6.(2)に該当する	juiceBにおいて、硬貨を投入した際juiceBを選択してもフリーズする	Operation_press_juiceBbuttonにおけるPO2にてエラーが発生した			juiceBにおける貯蔵しているコインの枚数inserted_coinを0にすることで、解決できた	コインを投入し、juiceBを選択しても正しく購入することができた	
4	OPELATION節における open_maintenence_doorにおいて、openになった場合の処理がない	仕様の『動的な振る舞い』9に該当する	管理ドアをopenにしようとしても、フリーズする・ドアを開けても貯蔵しているコインが返却されない	Operation_open_maintenance_doorにおけるPO1にてエラーが発生した			returned_coin, dispensing_slot, maintanance_door, inserted_coinの処理を追加する。	openになった場合の処理が追加され、コインの返却・貯蔵コインを空にする処理が修復された	
5	OPELATION節における power_turn_onのとき、どんな場合でもONの状態にできてしまうこと	仕様の『動的な振る舞い』1に該当する	電源がonの状態に電源ボタンを押下しても再度onの状態となってしまう		G(e(power_turn_on) => [power = off])などが成り立たない		PREにおいて、power = offと明記し、1=1などの処理は削除した。	電源がoffの状態の場合のみ、onの状態に切り替えることができるようになった	
6	OPELATION節における power_turn_off_return_inserted_coinのとき、コイン返却機能などがない	仕様の『動的な振る舞い』8に該当する	電源がonの状態でoffにすると、コインが返却されなく、貯蔵コインの枚数も変化しない		G([power_turn_off_return_inserted_coin] => BA([power = off & inserted_coin = 0 & returned_coin = inserted_coin$0])成り立たない		returned_coin, inserted_coinにおける処理が不足していたため、returned_coinにはinserted_coin, inserted_coinには0を代入した	電源をoffにすると、コインが返却され貯蔵されているコインの枚数は0枚となった	
7	OPELATION節における refill_juiceA_stockのとき、juiceA_stockが3になっている点	仕様の『静的な振る舞い』5に該当する	ジュースAを補充する際、最大の在庫量が3缶までしか補充することができない		G([refill_juiceA_stock] => X([juiceA_stock = 5]))が成り立たない		juiceA_stock := 3という処理を、5に変更した。	ジュースAは最大の在庫量5缶まで補充することができるようになった	
8	OPELATION節における refill_juiceB_stockのとき、PRE条件が1=1となっている点	仕様の『動的な振る舞い』7に該当する	ジュースBにおける補充は、いつどんなタイミングでも行うことができてしまう		G(e(refil_juiceB_stock) => [maintenance_door = open]))が成り立たない		maintenance_door = open のときと、明確に定めるよう修復した	ジュースBの補充処理は、管理ドアが開いている状態のみに限定することができた	

6. Experiment Week 3

process of repairing VM model

実験日：2025年7月17日
実験者：

項番	モデルの不具合の箇所	モデルの仕様の該当箇所	モデルの不具合の状況・修復前の振舞い				モデルを修復した内容	モデルの修復後の状況・修復後の振舞い	その他
			ProB アニメーションの様子	AtelierB 論理証明の様子	ProB LTL式検証の様子	その他			
1	L118からL129 press_juiseA_button	ジュースAの在庫数が0缶の場合、商品取出口・コイン返却口に関する振舞いは無く、状態を維持する。	ストックが0の時に購入ができてしまう	juiceA_stock$1 = juiceA_stock-1 & inserted_coin$1 = 0 & returned_coin$1 = inserted_coin-1 & "Invariant is preserved" => juiceA_stock$1: 0..5			juiceA_stock ＞0 & inserted_coin ＞0 & をL120・L121の間に追加	在庫数が0の場合、購入不可で状態維持	
2	L131からL143 press_juiseB_button	ジュースBの在庫数が0缶の場合、商品取出口・コイン返却口に関する振舞いは無く、状態を維持する。	ストックが0の時に購入ができてしまう	juiceB_stock$1 = juiceB_stock-1 & returned_coin$1 = inserted_coin-1 & "Invariant is preserved" => inserted_coin$1: 0..3			inserted_coin = 0 & をL140・L141の間に追加	在庫数が0の場合、購入不可で状態維持	
3	L82からL96 coin_insert	貯蔵可能なコインの最大枚数は3枚である	4枚目以降を投入したときにエラーが出る	power = on & (0+1<=juiceA_stock or 0+1<=juiceB_stock) & inserted_coin: 0..3 => inserted_coin$1|->returned_coin$1 = inserted_coin+1|->0 & power = on & juiceA_stock = 0 & juiceB_stock = 0 => inserted_coin$1|->returned_coin$1 = inserted_coin|->1 & power = off => inserted_coin$1|->returned_coin$1 = inserted_coin|->1 & "Invariant is preserved" => inserted_coin$1: 0..3			((power=on & (juiceA_stock>0 or juiceB_stock>0) & inserted_coin$0 = 3) => (inserted_coin,returned_coin) = (inserted_coin$0,1)) & をL90・L91の間に追加	3枚以上投入した場合、返却口に1枚出てくる	
4	L163からL172 open_maintenance_door	管理ドアの状態は開・閉の2つのみである	管理ドアをひらけない	inserted_coin$1 = 0 & "Invariant is preserved" => returned_coin$1: 0..3			returned_coin = inserted_coin$0 をL170・L171の間に追加する	管理ドアを開・閉できる	
5	L145からL152 refill_juiceA_stock	ジュースAの補充を指示すると、最大の在庫数になる。	Aのストックが補充できない		✘ G([refill_juiceA_stock] => X([juiceA_stock = 5]))		L149 juiceA_stock := 3 || を juiceA_stock := 5 || に修正する	ジュースAを最大数に補充できるようになった	
6	L154からL161 refill_juiceB_stock	管理ドアが開いている状態において、商品の補充の指示を受け入れる。	管理ドアが開いてないにも関わらず、Bのストックが補充できる		✘ G(e(refill_juiceB_stock) => [maintenance_door = open])		L156 1=1 を maintenance_door = open に修正する	管理ドアが開いた状態でのみ、商品を補充できる	
7	L70からL80 power_turn_off_return_inserted_coin	電源がonの状態において電源スイッチを操作すると、貯蔵しているコインをコイン返却口に返却する。	電源を切っても、コインが返却されない		✘ G([power_turn_off,return_inserted_coin] => BA([(inserted_coin$0 > 0) => (power = off & returned_coin = inserted_coin$0)]))		L77からL78 inserted_coin = 0 & returned_coin = 0 を inserted_coin = 0 & returned_coin = inserted_coin$0 に修正する	電源をoffにしたら、コインが返却された	
8	L61からL68 power_turn_on				✘ G([inserted_coin > 0] & [reset_pass = not_passed]=> F([returned_coin > 0] or [dispensing_slot /= empty] or [reset_pass = passed]))		L63 1=1 を power = off に修正する	コインを投入した後、商品の購入、コインの返却、リセットができる	
9	L61からL68 power_turn_on	電源スイッチの状態はon・offの二つのみである。電源がonの状態において電源スイッチを操作すると、貯蔵しているコインをコイン返却口に返却する。	スイッチのON,OFFを変えられない		✘ G([power_turn_off,return_inserted_coin] => BA([(power = off & inserted_coin = 0 & returned_coin = inserted_coin$0)]))		L63 1=1 を power = off に修正する	電源のon・offを変えれるようになった	

6. Experiment Week 3

First trial experiment Week 3 in 2025

- 8 defects are introduced in the Vending Machine model.
- The difference in the result does not indicate the difference of ability between the proof obligation generation and the model checking.
- Due to the state explosion problems in early model checking by ProB, the teacher has instructed the students to generate proof obligations by Atelier B first.

Student's defect locations detection of Vending Machine model

detected defect locations	N	ave.	Quartiles max.	$Q_{3/4}$	$Q_{2/4}$	$Q_{1/4}$	min.
all of defect locations	34	8.03	10	9	8	8	6
detection by ProB animation interface	34	7.15	10	8	8	7	1
detection by Atelier B proof obligation generation	34	3.38	8	5	4	0	0
detection by ProB LTL expression proving	34	1.68	9	3	1	0	0

Contents

7. Students' questionnaire survey of experiment Week 1–3

Survey questions for the Level of formal method usage

The Levels of formal methods usage

Level–0: Formal Specification
Formal specification may be undertaken
and then a program developed from this informally.

Level–1: Formal Development and Formal Verification
Formal development and formal verification may be used
to produce a program in a more formal manner.

Level–2: Theorem Provers
Theorem provers may be used
to undertake fully formal machine checked proofs.

Hehner, E.C.R.: Computer Science - Formal Methods,
https://computingstudy.wordpress.com/formal-methods/

Level-0	Formal specification may be undertaken and then a program developed from this informally.
[L0-1]	having ability to describe the rigid specification with the usage of mathematical notation.
[L0-2]	having ability to develop the software by referring to the rigid specification edited in mathematical notation.
Level-1	Formal development and formal verification may be used to produce a program in a more formal manner.
[L1-1]	having ability to prove the quality of the software
[L1-2]	having ability to develop the software from the specification by the manner of refinement
Level-2	Theorem provers may be used to undertake fully formal machine-checked proofs.
[L2-1]	having ability to prove the quality of the software with the usage of proof obligation generator
[L2-2]	having ability to verify the quality of the software with the usage of model checker

Araki K.:New Trends in Formal Methods :
Formal Methods : Past, Present and Future
-Toward Practical Applications -,
Magazine of IPSJ, IPSJ, 49(5), 493–498(2008),
https://ipsj.ixsq.nii.ac.jp/records/60933

Refine the Levels of formal method usage,
and make learning goals for experiment

7. Students' questionnaire survey of experiment Week 1–3

Making the survey questionnaires

Level-0	Formal specification may be undertaken and then a program developed from this informally.
[L0-1]	having ability to describe the rigid specification with the usage of mathematical notation.
[L0-2]	having ability to develop the software by referring to the rigid specification edited in mathematical notation.
Level-1	Formal development and formal verification may be used to produce a program in a more formal manner.
[L1-1]	having ability to prove the quality of the software
[L1-2]	having ability to develop the software from the specification by the manner of refinement
Level-2	Theorem provers may be used to undertake fully formal machine-checked proofs.
[L2-1]	having ability to prove the quality of the software with the usage of proof obligation generator
[L2-2]	having ability to verify the quality of the software with the usage of model checker

Learning Goals & Survey Questions	Week 1	Week 2	Week 3
Level-0: Formal Specification			
[L0-1]	(1-A)(1-B) (1-C)(1-D) (1-E)(1-J) (1-K)(1-L)	(2-A)(2-B) (2-L)	(3-A)(3-B) (3-J)
[L0-2]	-	(2-C)(2-O)	(3-C)(3-M)
Level-1: Formal Development and Formal Verification			
[L1-1]	(1-E)(1-F) (1-G)(1-H)	(2-D)(2-E) (2-I)(2-J) (2-N)	(3-D)(3-H) (3-L)
[L1-2]	-	(2-C)	(3-C)
Level-2: Theorem Provers			
[L2-1]	(1-E)(1-M) (1-O)	(2-C)(2-D) (2-E)(2-F) (2-P)(2-Q)	(3-C)(3-D) (3-E)(3-N) (3-O)
[L2-2]	(1-F)(1-G) (1-H)(1-I) (1-M)(1-O)	(2-I)(2-J) (2-K)(2-P) (2-Q)	(3-H)(3-I) (3-N)(3-O)

Make the survey questionnaires with mapping from learning goals

About Atelier B	
	(2-A) through (2-F) .... omitted ....
About ProB	
	(2-G) through (2-K) .... omitted ...
About the whole of B-Method	
(2-L)	having ability to describe the formal spec expressions from natural language
	Do you think you become used to E expressions?
(2-M)	having ability to understand the significa checking's coverage to whole states of sys
	Do you think you were able to unde improve the system's dependability proving and by the usage of checking
(2-N)	having ability to understand the significance contribute to improve the system's dependability
	Do you think you were able t verify the system has no def expressions with checking
(2-O)	having ability to understand the s specification is the proper approa defect-free codes
	Do you think you understand useful to implement no-defect code in contrast to simply creating code that anyways works?
(2-P)	having ability to agree the significance of the process of formal specifications, theorem proving and formal model checking
	Do you think you feel familiar to formal method, B-Method through both example and exercises of the experiment?
(2-Q)	having the opportunity to commit a formal method as the software engineer's fundamental knowledge
	Do you think you were able to have the opportunity to commit a formal method as the software engineer's fundamental knowledge?

Every Weeks

Week 1 Week 2 Week 3

- About Atelier B
- About ProB
- About the whole of B-Method

Result of the survey questionnaires

http://www.asahi-net.or.jp/~ny9t-oons/b-method_english.html

7. Students' questionnaire survey of experiment Week 1–3

Survey question for educational goal

We set the educational goal for the experiments
as giving opportunities to commit a formal method
as the software engineer's fundamental knowledge.

Every week's student's survey questions.

"Do you think you were able to have the opportunity
to commit a formal method as the software engineer's fundamental knowledge?"

Week 1 Week 2 Week 3

week	N	ave.	Quartiles max.	$Q_{3/4}$	$Q_{2/4}$	$Q_{1/4}$	min.	Survey Period
Week 1	202	4.46	5	5	5	4	2	2020-2025
Week 2	202	4.42	5	5	5	4	3	2020-2025
Week 3	36	4.28	5	5	4	4	1	2025

7. Students' questionnaire survey of experiment Week 1–3

Survey question for educational goal

"Why we start to teach Formal Methods?"

On the project, Ohnishi was tasked with teaching formal methods at his school.

We set the educational goal for the experiments as giving opportunities to commit a formal method as the software engineer's fundamental knowledge.

Every week's student's survey questions.

"Do you think you were able to have the opportunity to commit a formal method as the software engineer's fundamental knowledge?"

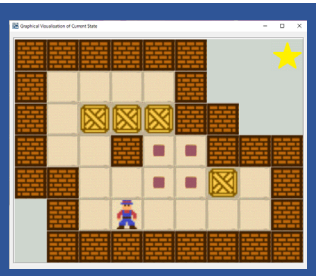
week	N	ave.	Quartiles max.	$Q_{3/4}$	$Q_{2/4}$	$Q_{1/4}$	min.	Survey Period
Week 1	202	4.46	5	5	5	4	2	2020-2025
Week 2	202	4.42	5	5	5	4	3	2020-2025
Week 3	36	4.28	5	5	4	4	1	2025

- It seems that the educational goal is achieved, but there are some abilities to improve the methodology.
- We acknowledge and regret that we have never evaluated student's competence in long term.

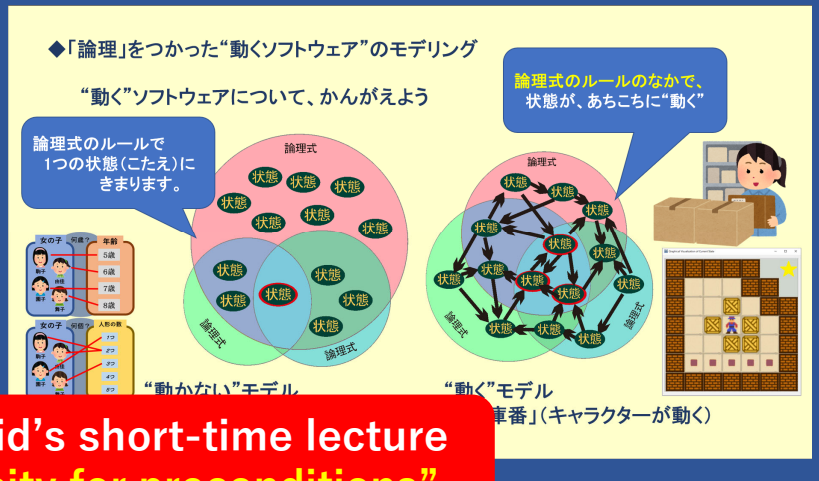
◆ Contents

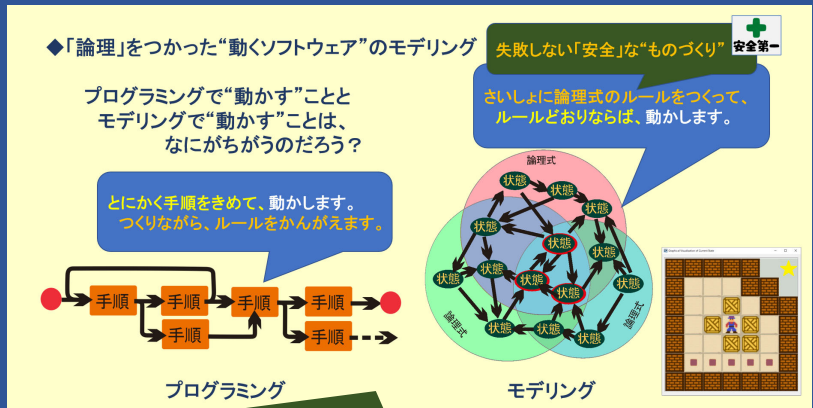
8. Our expected direction

Applying “Fountain of logic expression” to multi-state formal models



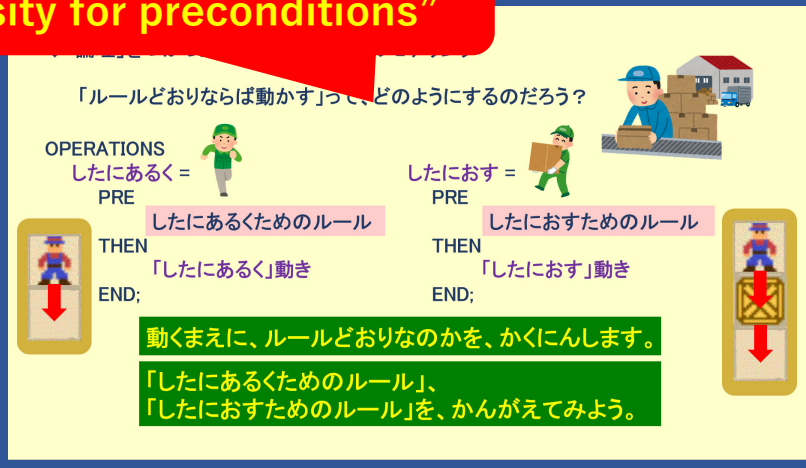
”Sokoban (Warehouse keeper) ”,
https://en.wikipedia.org/wiki/Sokoban

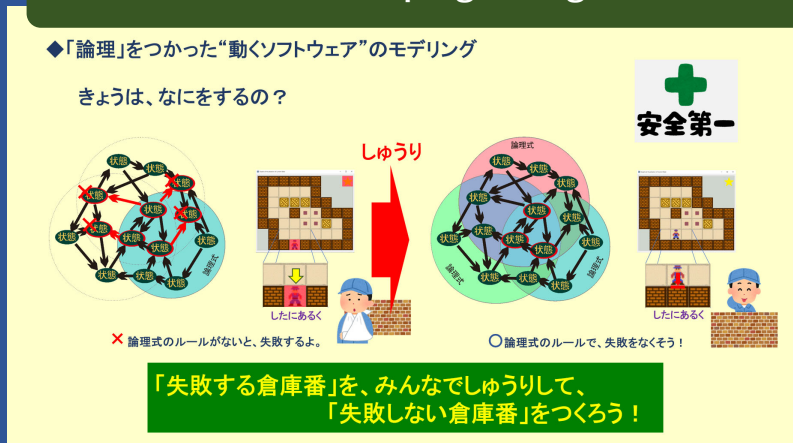




Aim of kid’s short-time lecture
“Necessity for preconditions”

Difference between “programing” and ”modeling”

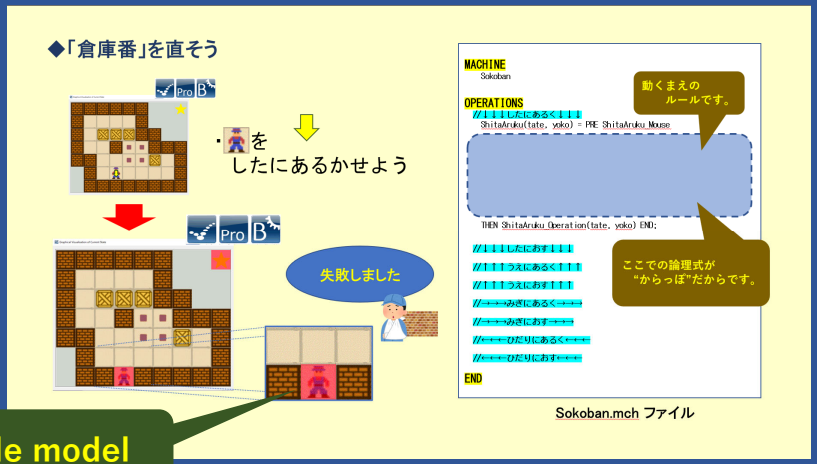




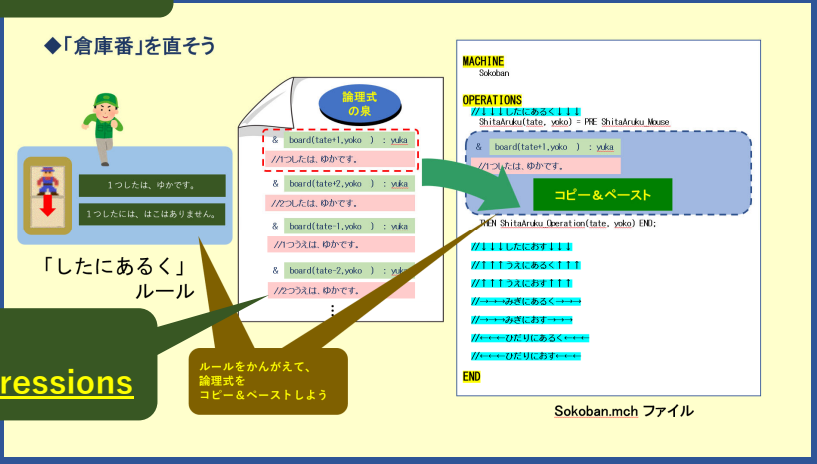
8. Our expected direction

Applying “Fountain of logic expression” to multi-state formal models

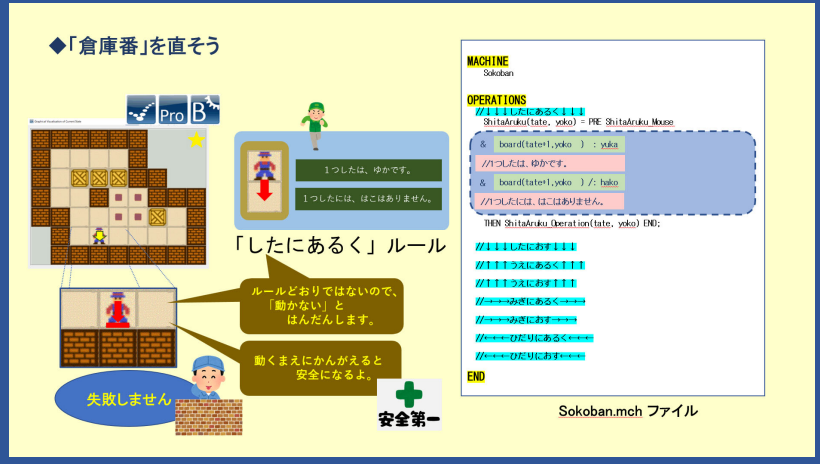
Aim of kid’s short-time lecture
“Necessity for preconditions”



Error capable model



Fountain of Logic Expressions



Contents

9. Summary

(1) Introducing the formal method, B-Method since 2019
- Mandatory 3-week experiments of Japanese technical college
- Short-time lectures other than our class

→ Simultaneous conducting both of two provided the improvement of our methodology.

(2) Experiments of Japanese technical college
- Week 1: Single-state formal models "Deductive logic puzzles"
- Week 2: Multi-state formal models "Test of logic circuit"
- Week 3: Multi-state formal models "Vending Machine" (concept "Error capable model" applied)
- Students' questionnaire survey
  - Introduced from **the level of formal method usage, deriving learning goals, referring our educational goal**

→ It seems that the educational goal is achieved, but there are some abilities to improve the methodology.

→ We acknowledge and regret that we have never evaluated student's competence in long term.

(3) Serendipitous ideas of our activity
- Idea 1: Fountain of logic expressions
- Idea 2: Error capable model

(4) Our expected direction
- Applying "fountain of logic expression" to multi-state formal models