# Vending Machine Mdel Checking

진정하 / 김우빈
MBC Lab
2015. 4. 27.

# Vending Machine

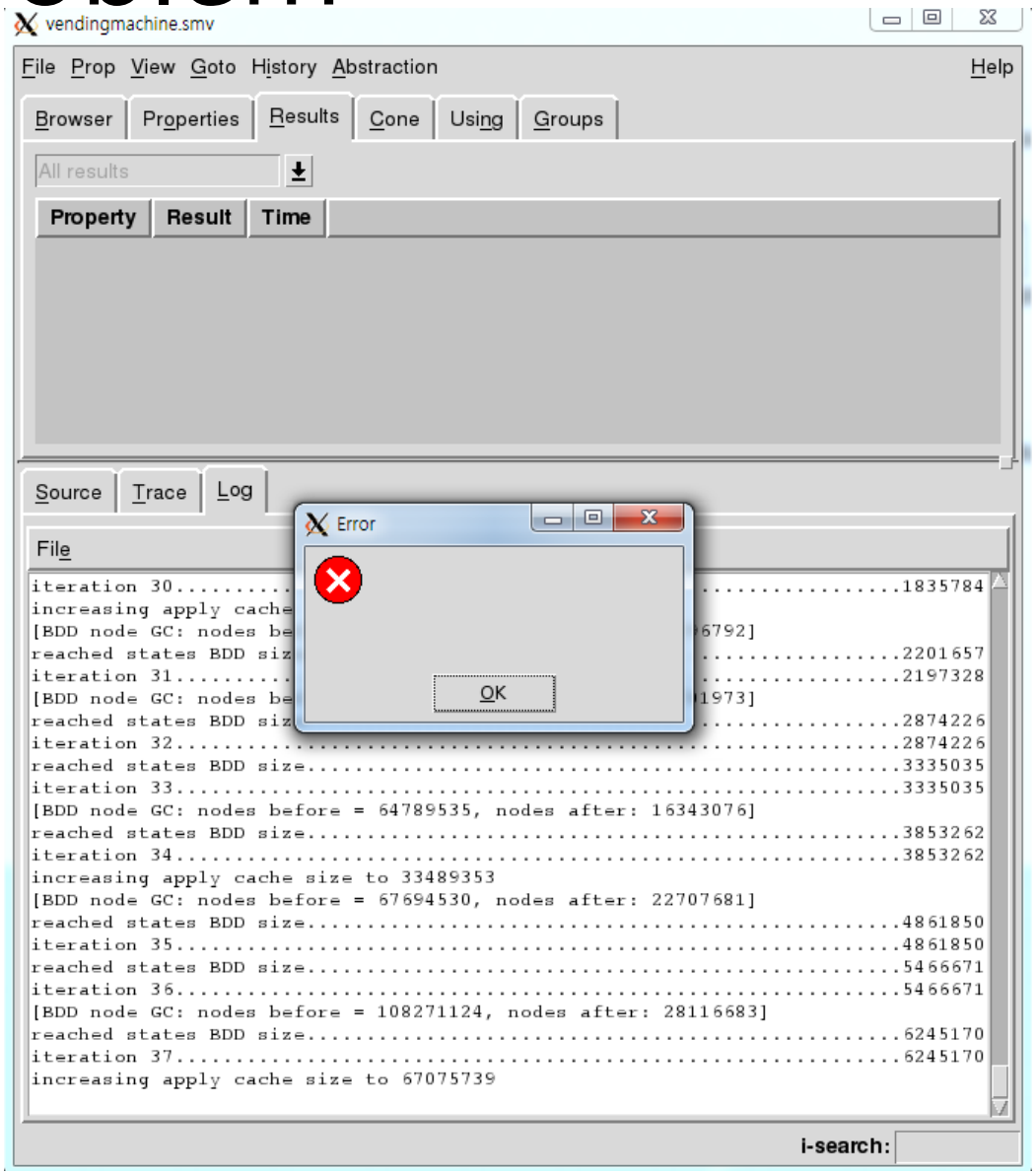
- System
  - sell product
    - can drink : 500원 X 2ea , 600원, 700원, 1,000원
    - coffee : 200원 X 2ea, 300원 X 3ea
- Model Checkers
  - SMV

# parameter setting

- Input
  - coin : 50원, 100원, 500원, 1,000원(지폐)
  - button : can_button 5ea, coffee_button 5ea, refund 1ea
- Output
  - can_button_lamp 5ea, coffee_button_lamp 5ea, coffee_lamp 3ea
  - money_display

# Problem

- 검증시 많은 노드와 BDD
- 메모리 오버로 검증이 불가



# parameter setting

- Input
  - coin : 50원, 100원, 500원, 1,000원(지폐)
  - button : can_button 5ea, coffee_button 5ea, refund 1ea
- Output
  - can_button_lamp 5ea, coffee_button_lamp 5ea, coffee_lamp 3ea
  - money_display

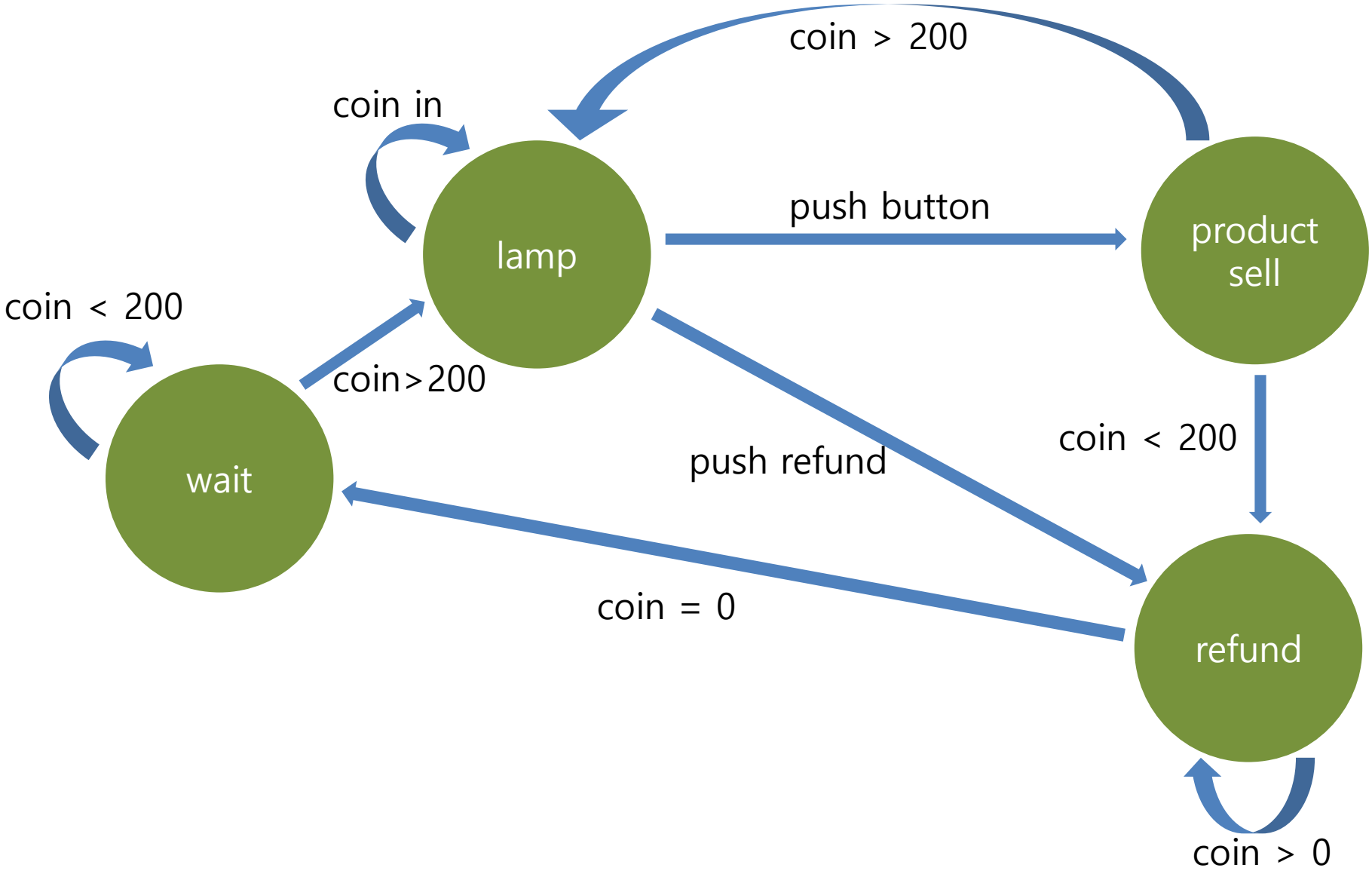
# constraint-1

- coin max : 5,000원
  - 동전의 최소 단위 : 50원
  - 지폐사용 : 1,000원
- button
  - 동시에 여러개의 상품 버튼을 누를 때 동작하지 않는다(단, 커피와 캔음료는 별개로 동작)
  - 상품버튼을 누르면서 반환버튼을 누를시 상품이 출하되고 남은 잔액에 대해 refund 된다
  - 버튼에 램프가 들어와 있지 않는 경우 눌러도 아무런 동작을 하지 않는다

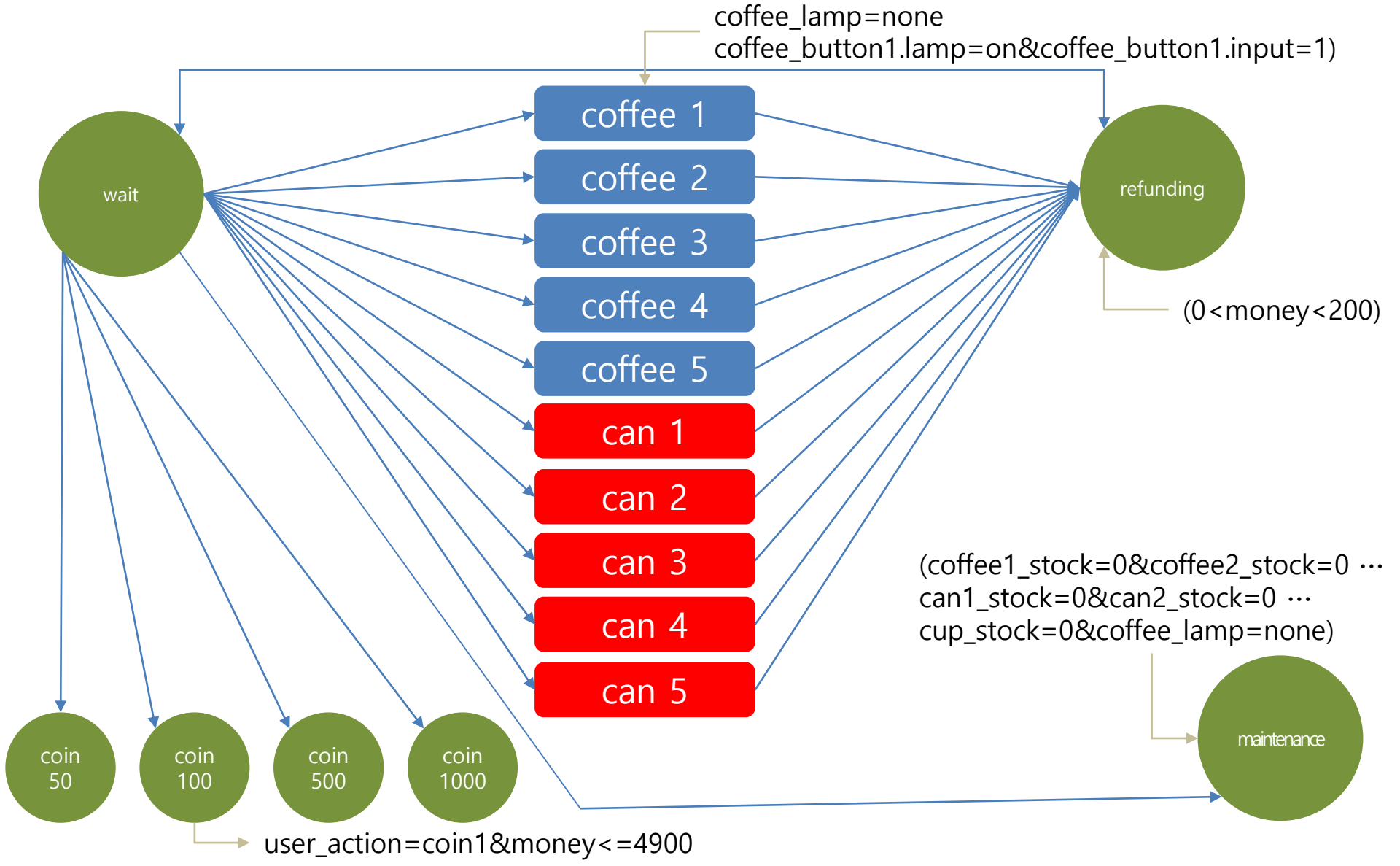
# constraint-2

- refund
  - 자판기에 입금된 돈에서 작은 단위의 돈부터 사용
  - 다수의 지폐를 투입후 반환 요청시 지폐는 한 장만 반환하고 남은 금액은 동전으로 반환
  - 잔액이 400원 있는 경우 100원 추가후 반환버튼 누를시 100원 5개로 반환

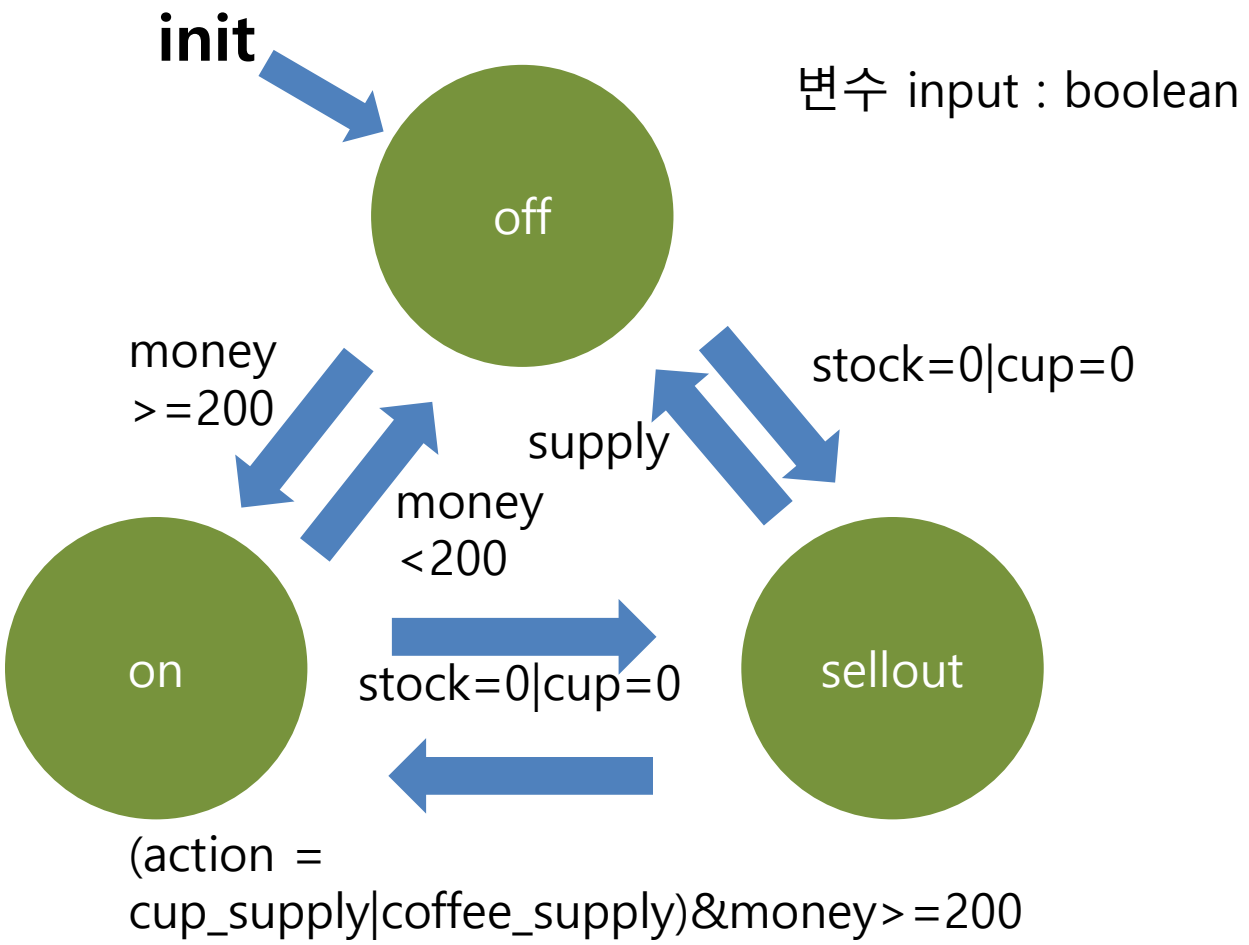
# Automata


coin > 200
coin in
lamp
push button
product sell
coin < 200
coin>200
wait
push refund
coin < 200
coin = 0
refund
coin > 0

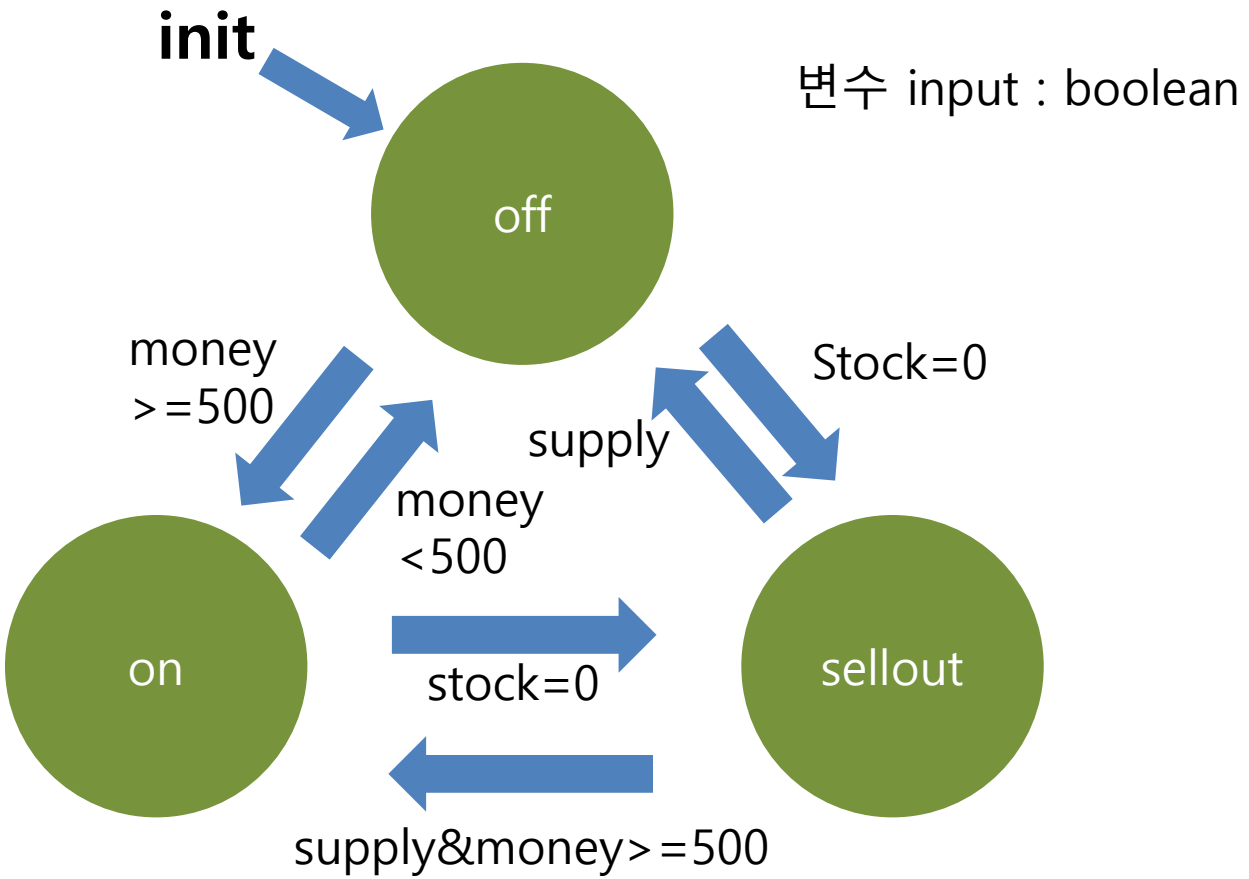
# Automata – main module



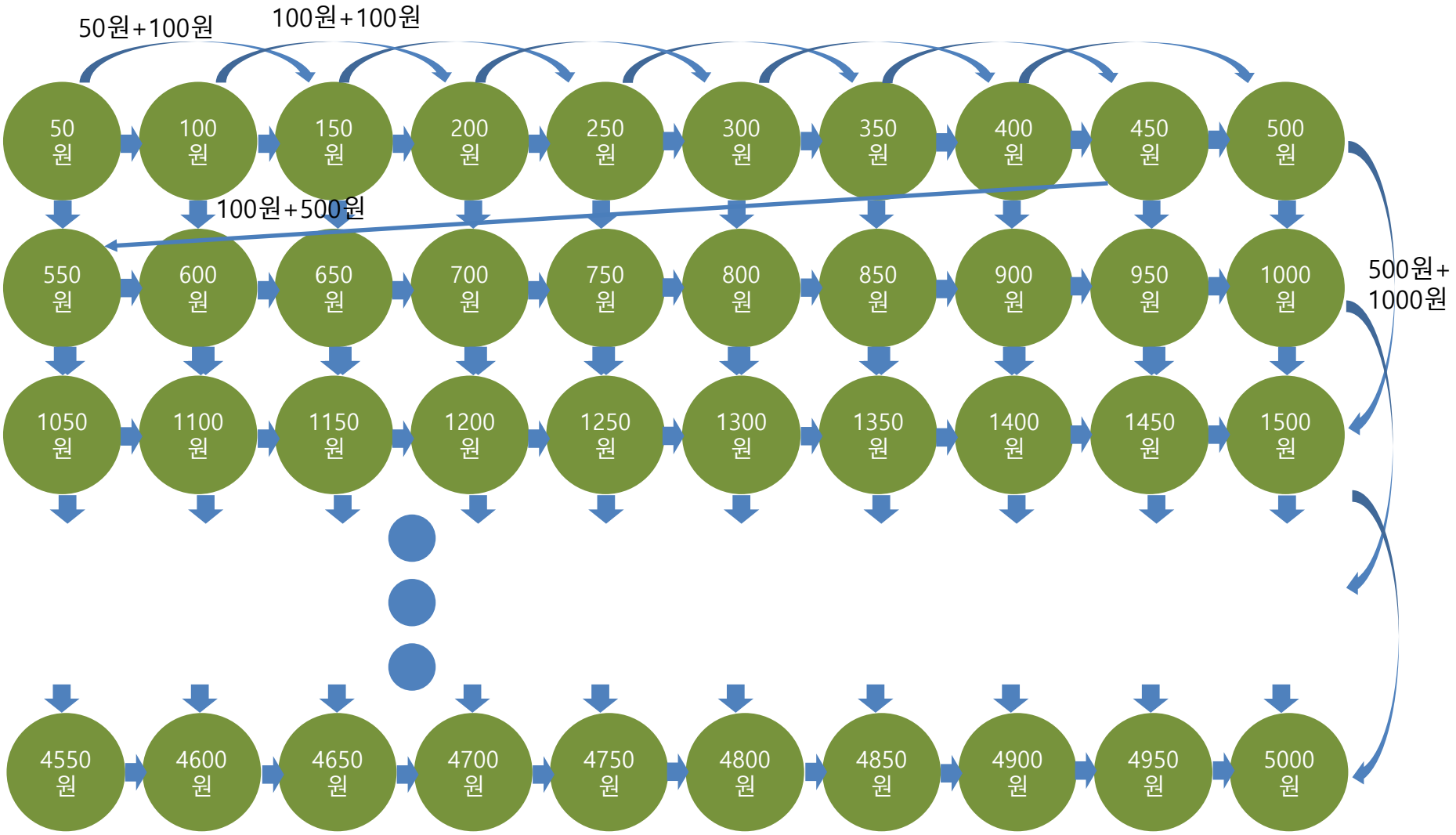
# Automata - coffee button module



# Automata - can button module



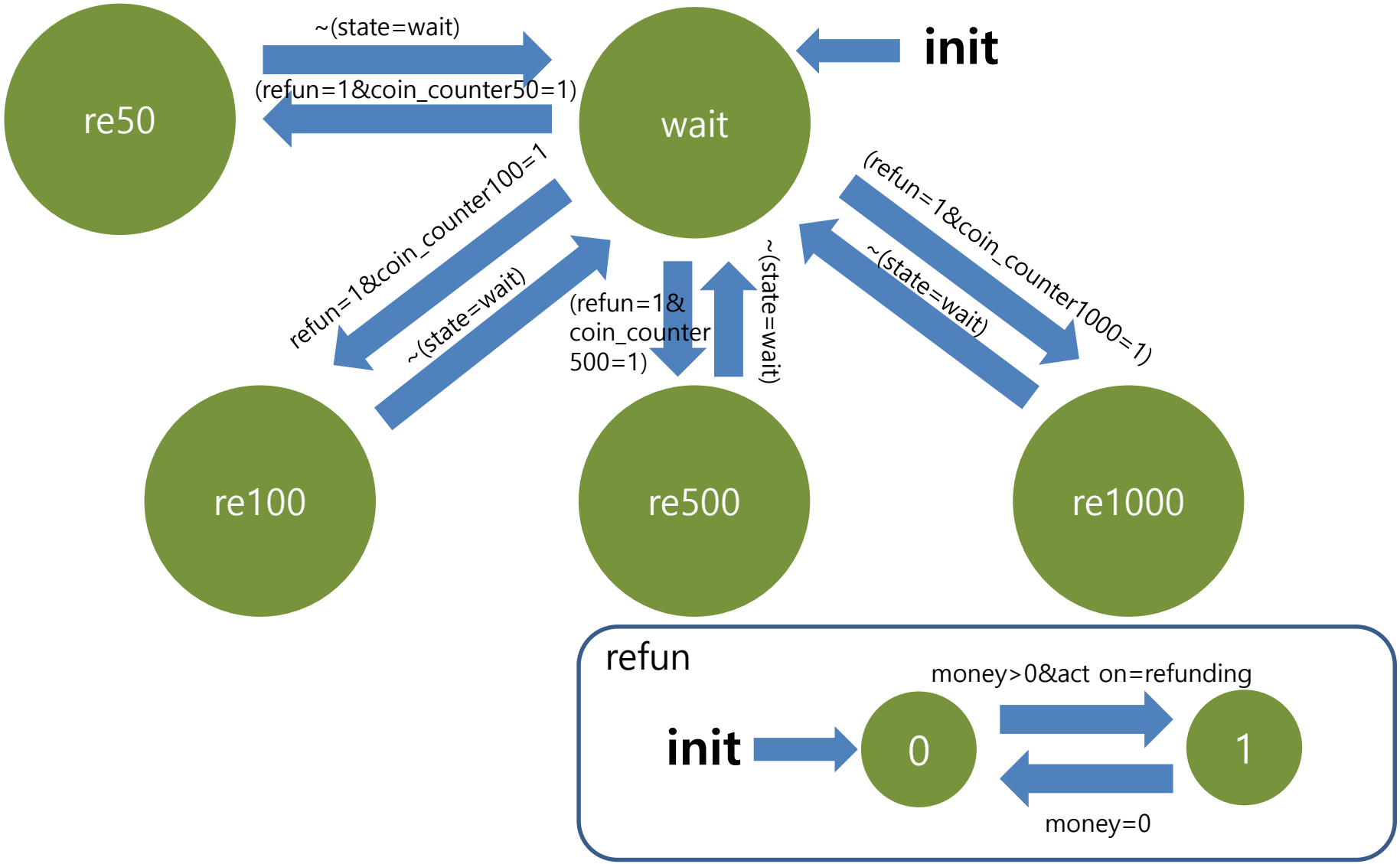
# Automata – money display state



# Automata – money display state

| | |
|---|---|
| 50원 | 50원 |
| 100원 | 50원*2, 100원 |
| 150원 | 50원*3, 50원+100원 |
| 200원 | 50원*4, 50원*2+100원, 100원*2 |
| 250원 | 50원*5, 50원*3+100원, 100원*2+50원 |
| 300원 | 50원*6, 50원*4+100원, 100원*2+50원*2, 100원*3 |
| 350원 | 50원*7, 50원*5+100원, 100원*2+50원*3, 100원*3+50원 |
| 400원 | 50원*8, 50원*6+100원, 100원*2+50원*4, 100원*3+50원*2, 100원*4 |
| 450원 | 50원*9, 50원*7+100원, 50원*5+100원*2, 50원*3+100원*3, 50원+100원*4 |
| 500원 | 50원*10, 50원*8+100원, 50원*6+100원*2, 50원*4+100원*3, 50원*2+100원*4, 100원*5, 500원 |

# Automata - refund module



# module 구현 -1

- coffee 버튼 모듈

```
MODULE button1(money,stock,cup)
VAR
  lamp : {on, off, sellout};
ASSIGN
  init(lamp) := off;
  next(lamp) :=
  case
    (cup > 0 & stock > 0 & money < 200) : off;
        (cup > 0 & stock > 0 & money >= 200) : on;
        (stock = 0) : sellout;
        (cup = 0) : sellout;
        1: lamp;
  esac;
```

- can 버튼 모듈

```
MODULE button3(money,stock)
VAR
  lamp : {on, off, sellout};
ASSIGN
  init(lamp) := off;
  next(lamp) :=
  case
    (stock > 0 & money < 500) : off;
        (stock > 0 & money >= 500) : on;
        (stock = 0) : sellout;
        1: lamp;
  esac;
```

- refund 버튼 모듈

```
MODULE refund(re,c100,c500,c1000)
VAR
  state : {wait, re100, re500, re1000};
ASSIGN
  init(state) := wait;
  next(state) :=
  case
        (re = 1 & c1000 = 1) : re1000;
        (re = 1 & c500 > 0) : re500;
        (re = 1 & c100 > 0) : re100;
        1 : wait;
  esac;
```

# module 구현 -2

- main 모듈

```
MODULE main
  VAR
    money : {0,100,200,300,400,500,600,700,800,900,1000,
                        1100,1200,1300,1400,1500,1600,1700,1800,1900,2000,
                        2100,2200,2300,2400,2500,2600,2700,2800,2900,3000,
                        3100,3200,3300,3400,3500,3600,3700,3800,3900,4000,
                        4100,4200,4300,4400,4500,4600,4700,4800,4900,5000};
        coin100_count : 0..50;
        coin500_count : 0..10;
        coin1000_count : 0..1;
        thousand : boolean;
        refun : boolean;
        R : refund(refun,coin100_count,coin500_count,thousand);
        action : {wait,refunding,
                        coin100,coin500,coin1000,
                        coffee1,coffee3,
                        supply_coffee1,supply_coffee3,supply_cup,
                        can1,can3,can4,can5,
                        supply_can1,supply_can3,supply_can4,supply_can5};
        user_action : {coin1,coin5,coin10,co1,co3,ca1,ca3,ca4,ca5,wait,ref};
        coffee_lamp : {none,first,second,third};
        coffee1_stock : 0..5;
```

# module 구현 -3

- main 모듈

```
        coffee1_stock : 0..5;
--      coffee2_stock : 0..5;
        coffee3_stock : 0..5;
--      coffee4_stock : 0..5;
--      coffee5_stock : 0..5;
        cup_stock : 0..10;
        can1_stock : 0..5;
--      can2_stock : 0..5;
        can3_stock : 0..5;
        can4_stock : 0..5;
        can5_stock : 0..5;
        coffee_button1 : button1(money,coffee1_stock,cup_stock);
--      coffee_button2 : button1(money,coffee2_stock,cup_stock);
        coffee_button3 : button2(money,coffee3_stock,cup_stock);
--      coffee_button4 : button2(money,coffee4_stock,cup_stock);
--      coffee_button5 : button2(money,coffee5_stock,cup_stock);
        can_button1 : button3(money,can1_stock);
--      can_button2 : button3(money,can2_stock);
        can_button3 : button4(money,can3_stock);
        can_button4 : button5(money,can4_stock);
        can_button5 : button6(money,can5_stock);
```

# vending machine property-1

- SPEC AG ((money < 500) -> AX (can_button1 = off))
- SPEC AG ((coin1000_count = 1 & refun = 1) -> AX(R.state = re1000))
- SPEC (AG ((money>200) ->(AX (coffee_button1.lamp=on))))

# vending machine property-2

- SPEC (AG ((money>200) ->(AX (coffee_button1.lamp=on))))
  – 앞선 property의 검증을 위해 소스 코드 수정

- SPEC AG ((user_action = co1) -> AF(coffee_lamp = first))

- SPEC AG ((action = coffee1 & coffee1_stock > 0 & money>=500) -> AX(coffee_lamp = first))

# SMV에서의 검증-1

- SPEC AG ((money.state > 400) -> AX (coffee_button1.lamp = on | coffee_button3.lamp = on))
  - 금액이 충분할 때 커피 버튼에 불이 들어오는가
  - 결과는 false

# SMV에서의 검증-1

- 재고가 없어서 sellout이 발생

Source | Trace | Log

File Edit Run View

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R.refun | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R.state | wait | wait | wait | wait | wait | wait | wait | wait | wait | wait | wait | wait |
| action | wait | wait | coin1000 | wait | coffee1 | wait | wait | coin1000 | wait | coffee1 | wait | can1 |
| can1_stock | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| can2_stock | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| can3_stock | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| can4_stock | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| can5_stock | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| can_button1.input | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | |
| coffee1_stock | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| coffee2_stock | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| coffee3_stock | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| coffee4_stock | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| coffee5_stock | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| coffee_button1.input | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | |
| coffee_button1.lamp | off | off | off | off | on | on | on | on | on | on | on | sellout |
| coffee_button2.input | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| coffee_button3.input | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| coffee_button3.lamp | off | off | off | off | on | on | on | on | on | on | on | sellout |
| coffee_button4.input | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| coffee_button5.input | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| coffee_lamp | none | none | none | none | none | first | second | third | none | none | first | second |
| coin.coin1000_count | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| coin.coin100_count | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 3 | 3 | 3 | 1 | 1 |
| coin.coin500_count | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| cup_stock | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| money.state | 0 | 0 | 0 | 1000 | 1000 | 800 | 800 | 800 | 1800 | 1800 | 1600 | 1600 |
| user_action | wait | coin10 | coin1 | coin1 | coin1 | wait | coin10 | coin1 | coin1 | coin1 | coin1 | coin1 |
| | | | | | | | | | | | | |

# SMV에서의 검증-2

- SPEC AG ((money.state < 500) -> AX (can_button1.lamp = off))
  - 돈이 부족하면 램프가 off 상태이다
  - 결과는 false

# SMV에서의 검증-2

- sellout이 발생하는 경우 존재

Source | Trace | Log

File Edit Run View

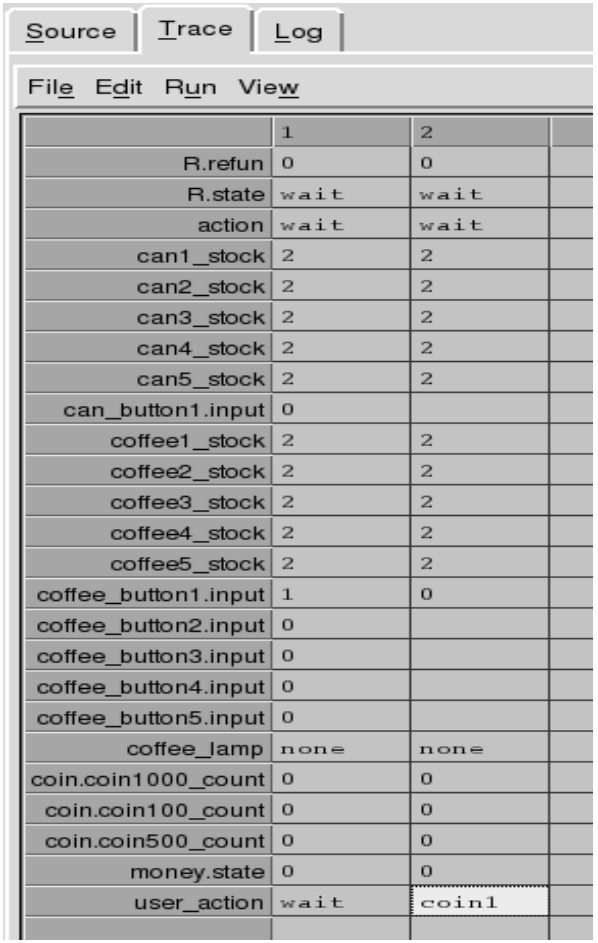
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| R.refun | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R.state | wait | wait | wait | wait | wait | wait | wait | wait | wait |
| action | wait | wait | coin1000 | wait | can1 | wait | can1 | wait | coin100 |
| can1_stock | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 0 | 0 |
| can2_stock | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| can3_stock | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| can4_stock | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| can5_stock | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| can_button1.input | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | |
| can_button1.lamp | off | off | off | off | on | on | on | on | sellout |
| coffee1_stock | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| coffee2_stock | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| coffee3_stock | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| coffee4_stock | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| coffee5_stock | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| coffee_button1.input | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| coffee_button2.input | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| coffee_button3.input | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| coffee_button4.input | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| coffee_button5.input | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| coffee_lamp | none | none | none | none | none | none | none | none | none |
| coin.coin1000_count | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| coin.coin100_count | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| coin.coin500_count | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| money.state | 0 | 0 | 0 | 1000 | 1000 | 500 | 500 | 0 | 0 |
| user_action | wait | coin10 | coin1 | coin1 | coin1 | coin1 | coin1 | coin1 | coin1 |

# SMV에서의 검증-3

- SPEC AG ((coffee_button1.input = 1) -> AX(action = coffee1))
  - 커피 버튼을 누르면 커피가 나온다
  - 결과는 false

# SMV에서의 검증-3

- money가 0이어서 제품이 나오지 않음




| | 1 | 2 |
|---|---|---|
| R.refun | 0 | 0 |
| R.state | wait | wait |
| action | wait | wait |
| can1_stock | 2 | 2 |
| can2_stock | 2 | 2 |
| can3_stock | 2 | 2 |
| can4_stock | 2 | 2 |
| can5_stock | 2 | 2 |
| can_button1.input | 0 | |
| coffee1_stock | 2 | 2 |
| coffee2_stock | 2 | 2 |
| coffee3_stock | 2 | 2 |
| coffee4_stock | 2 | 2 |
| coffee5_stock | 2 | 2 |
| coffee_button1.input | 1 | 0 |
| coffee_button2.input | 0 | |
| coffee_button3.input | 0 | |
| coffee_button4.input | 0 | |
| coffee_button5.input | 0 | |
| coffee_lamp | none | none |
| coin.coin1000_count | 0 | 0 |
| coin.coin100_count | 0 | 0 |
| coin.coin500_count | 0 | 0 |
| money.state | 0 | 0 |
| user_action | wait | coin1 |
| | | |

# SMV에서의 검증-4

- SPEC AG ((coffee_button1.input = 1 & coffee1_stock > 0 & cup_stock > 0 & money.state = 300) -> AX(action = coffee1))
  - 조건(돈, 재고)이 충분할 때 커피가 나오는지
  - 결과는 false

# SMV에서의 검증-4

- 커피가 나오는 도중 다른 커피를 뽑을 수 없음

Source | Trace | Log

File Edit Run View

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| R.refun | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R.state | wait | wait | wait | wait | wait | wait | wait |
| action | wait | wait | coin500 | wait | coffee2 | wait | coin100 |
| can1_stock | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| can2_stock | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| can3_stock | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| can4_stock | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| can5_stock | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| can_button1.input | 0 | 0 | 0 | 0 | 0 | 0 | |
| coffee1_stock | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| coffee2_stock | 2 | 2 | 2 | 2 | 2 | 1 | 1 |
| coffee3_stock | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| coffee4_stock | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| coffee5_stock | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| coffee_button1.input | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| coffee_button2.input | 0 | 0 | 0 | 1 | 0 | 0 | |
| coffee_button3.input | 0 | 0 | 0 | 0 | 0 | 0 | |
| coffee_button4.input | 0 | 0 | 0 | 0 | 0 | 0 | |
| coffee_button5.input | 0 | 0 | 0 | 0 | 0 | 0 | |
| coffee_lamp | none | none | none | none | none | first | second |
| coin.coin1000_count | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| coin.coin100_count | 0 | 0 | 0 | 0 | 0 | 3 | 3 |
| coin.coin500_count | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| cup_stock | 2 | 2 | 2 | 2 | 2 | 1 | 1 |
| money.state | 0 | 0 | 0 | 500 | 500 | 300 | 300 |
| user_action | wait | coin5 | coin1 | coin1 | coin1 | coin1 | coin1 |

# SMV에서의 검증-5

- SPEC AG (((coin.coin1000_count = 1 | coin.coin500_count > 0 | coin.coin100_count > 0) & action = refunding) -> AX(R.refun = 1))
  - 돈이 있을 때 refund 버튼을 누르면 자판기가 refunding 상태가 된다
  - 결과는 true

# SMV에서의 검증-6

- SPEC AG ((coin.coin1000_count = 0 & coin.coin500_count = 0 & coin.coin100_count = 3 & action = coffee1) -> AX(action = refunding))
  - 제품을 뽑고 잔액이 200원 미만인 경우 refunding 상태가 된다
  - 결과는 true

# SMV에서의 검증-7

- EF (money.state>5000)
  - money가 5000원을 초과하는 경우가 있는가
  - 결과는 false

# SMV에서의 검증-8

- EF (user_action = ref & coffee_lamp = second)
  - 커피가 나오는 도중에 반환이 가능한가
  - 결과는 true

# SMV에서의 검증-8

- SPEC AG (coin.coin1000_count = 1 & coin.coin500_count = 0 & action = coin1000) -> AX(coin.coin500_count = 2)
  - 1000원 지폐 2장을 넣었을 경우 1000원 1장 500원 2개인 상태가 된다
  - 결과는 true

# SMV에서의 검증-8

- SPEC EF (~(coffee_lamp = none) & action = can1)
  - 커피가 나오는 도중 캔을 뽑을 수 있다
  - 결과는 true