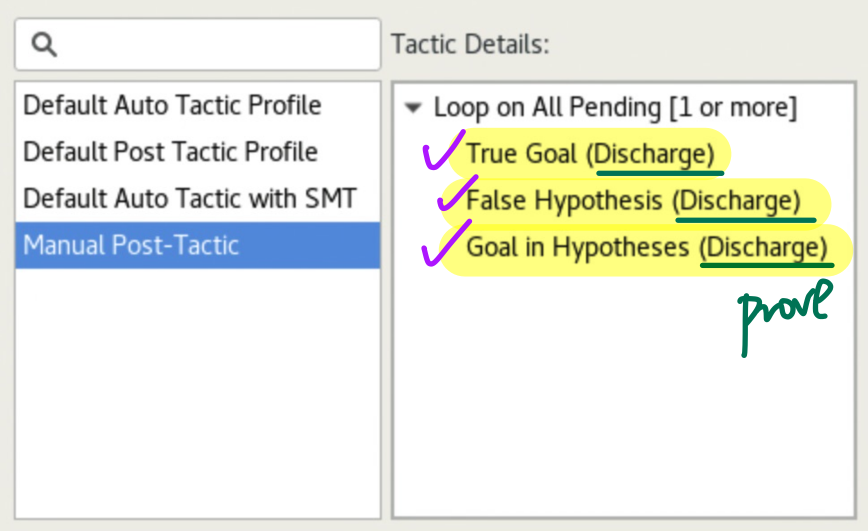
Tutorial

Introduction to the Rodin Platform for Formal Specifications

Post-Tactic Configurations



Tactic Details:

- Default Auto Tactic Profile
- Default Post Tactic Profile
- Default Auto Tactic with SMT
- Manual Post-Tactic

▾ Loop on All Pending [1 or more]

✓ True Goal (Discharge)

✓ False Hypothesis (Discharge)

✓ Goal in Hypotheses (Discharge)

prove

$$\boxed{P} \Rightarrow \boxed{Q}$$

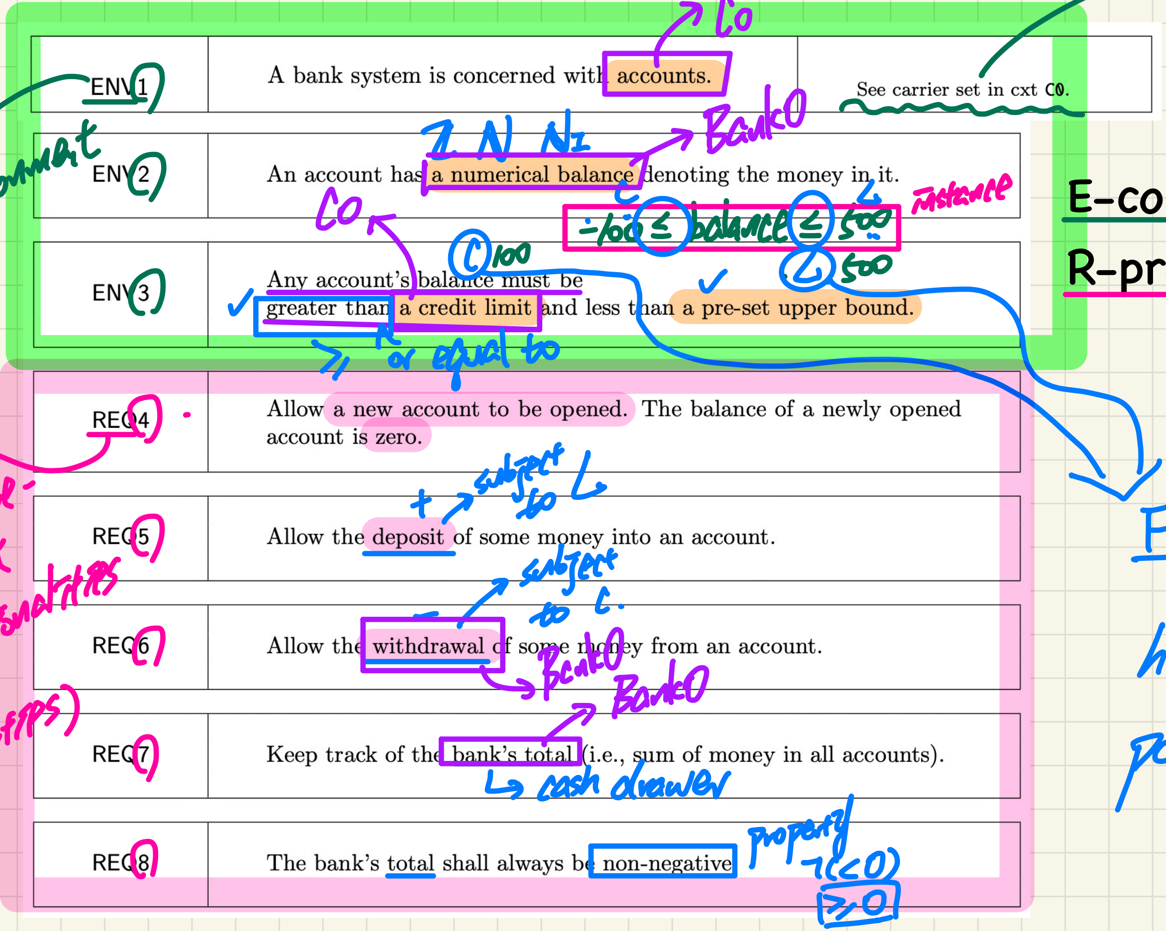
hypothesis goal

$\_\_ \Rightarrow True$ ✓

$False \Rightarrow \_\_$ ✓

$H \wedge Q \Rightarrow H$ ✓

Bank System: Requirements Document



ENV1	A bank system is concerned with accounts.	See carrier set in cxt C0.
ENV2	An account has a numerical balance denoting the money in it.	
ENV3	Any account's balance must be greater than a credit limit and less than a pre-set upper bound.	
REQ4	Allow a new account to be opened. The balance of a newly opened account is zero.	
REQ5	Allow the deposit of some money into an account.	
REQ6	Allow the withdrawal of some money from an account.	
REQ7	Keep track of the bank's total (i.e., sum of money in all accounts).	
REQ8	The bank's total shall always be non-negative	

Environment

Requirement (functionalities + properties)

C0

$\mathbb{Z}$ $\mathbb{N}$ $\mathbb{N}_1$ → Bank0

C

C0

C 100

$-100 \le balance \le 500$ instance

$<$ 500

$\geq$ or equal to

+ subject to L

- subject to C.

Bank0

Bank0

cash drawer

property $\neg(<0)$ $\boxed{\geq 0}$

tracing the E- or R-des. in the model.

E-constraints: Axioms

assumed without proof

R-properties: Theorems

to prove using axioms and/or theorems.

FM

prove properties holding on all possible combinations of C and L.

balance "b"

(~~acc3~~, 46)

$\{ (acc1, 230), (acc2, 460), (acc3, -23) \}$

$b \in ACCOUNT \nrightarrow \mathbb{Z}$

$\nrightarrow$

ACCOUNT

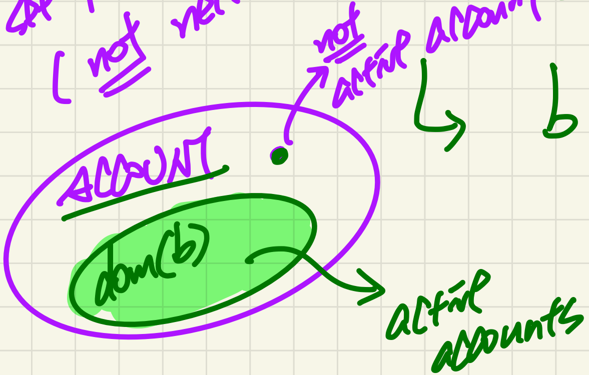
↳ set of all accounts

[not necessarily all are active in the bank)

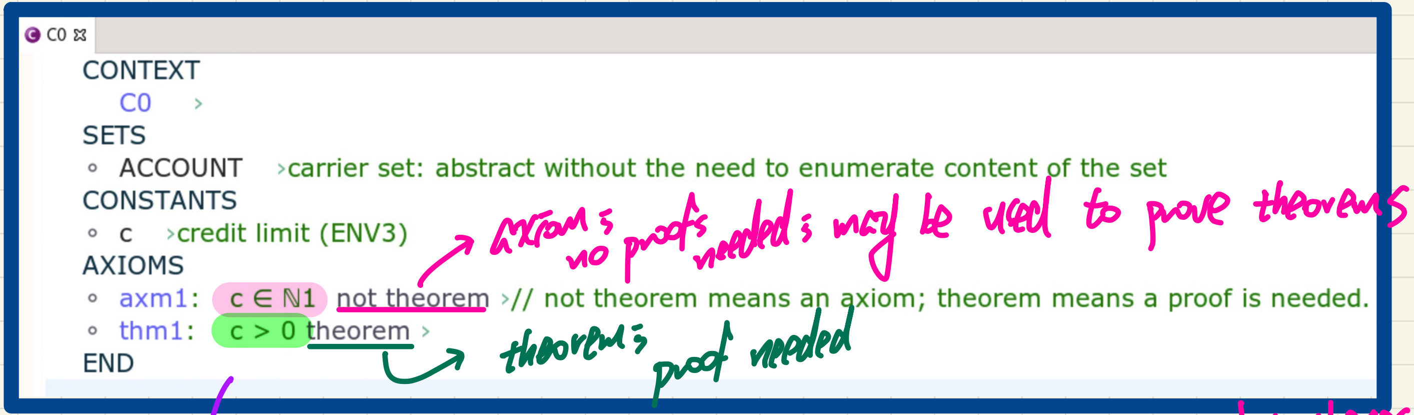
↳ b should not be a relation

↳ b should be a function

↳ $\nrightarrow$ or $\rightarrow$? → for justification, see Lecture W2.



Axiom vs. Theorem



C0

```
CONTEXT
    C0    ›
SETS
∘   ACCOUNT    ›carrier set: abstract without the need to enumerate content of the set
CONSTANTS
∘   c    ›credit limit (ENV3)
AXIOMS
∘   axm1:   c ∈ ℕ1  not theorem ›// not theorem means an axiom; theorem means a proof is needed.
∘   thm1:   c > 0 theorem ›
END
```

axiom: no proofs needed; may be used to prove theorems

theorem: proof needed

$$c \in \underset{\text{positive number}}{\mathbb{N}_1} \Rightarrow c > 0$$

$$\{x \mid x \in \mathbb{Z} \land x > 0\}$$

hypotheses

$$axm1 \land axm2 \ldots \land axm_n$$

$$\Rightarrow \boxed{thm1}$$

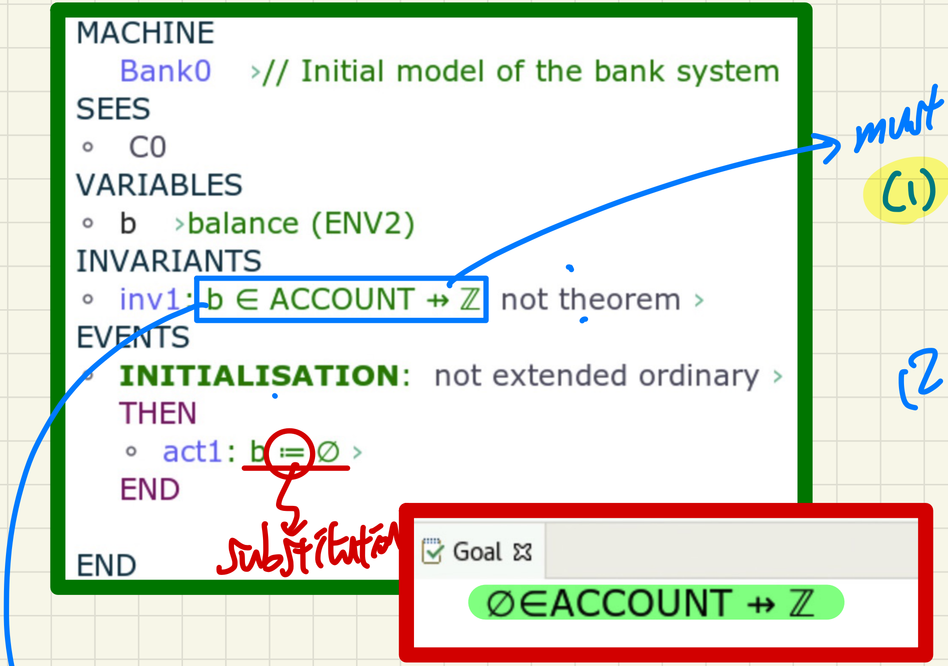
goal

Event Action

$v := \text{value}$

≈ variable assignment.

Proof Obligation: INITIALIZATION/inv1/INV



```
MACHINE
    Bank0   ›// Initial model of the bank system
SEES
∘   C0
VARIABLES
∘  b   ›balance (ENV2)
INVARIANTS
∘  inv1 : b ∈ ACCOUNT ⇸ ℤ   not theorem ›
EVENTS
∘  INITIALISATION:  not extended ordinary ›
   THEN
   ∘  act1 : b ≔ ∅ ›
   END
END
```

substitution

Goal

∅∈ACCOUNT ⇸ ℤ

must be:
(1) initialized/established by INITIALIZATION
(2) maintained by other events

$\cancel{b} \in \text{ACCOUNT} \nrightarrow \mathbb{Z}$

$\emptyset$

should hold to establish inv.

Event-B

modelling

events

↳ guards

deadlock: all events disabled

- ↳ True → event enabled to invoke
- ↳ False → event disabled

Java

programming

methods

↳ preconditions (exception)

- ↳ True → method exec.
- ↳ False → exceptions thrown.

$b \in Account \nrightarrow \mathbb{Z}$ partial func.

↳ relation

$\{(acc1, 240), (acc2, \overset{-33}{\cancel{-23}}), (acc3, 46)\}$

$dom(b) = \{acc1, acc2, acc3\}$

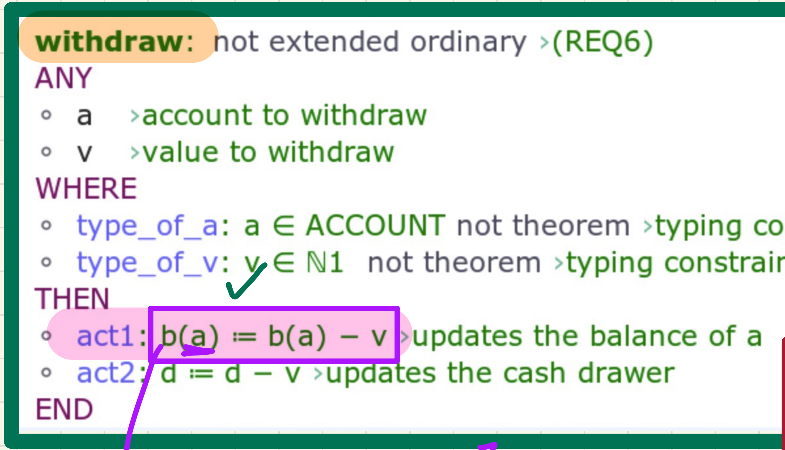
$\boxed{\text{withdraw } \underline{\$10} \text{ from } acc2}$

$$b \mathbin{\lhd\!\!\!+} \{(\underline{acc2}, \underline{b(acc2) - \underline{\underline{10}}})\}$$

Rodin : $\underline{b(acc2)} := \boxed{b(acc2) - 10}$

Proof Obligation: withdraw/act1/WD

→ well-definedness
↳ preconditions of math op. being satisfied.



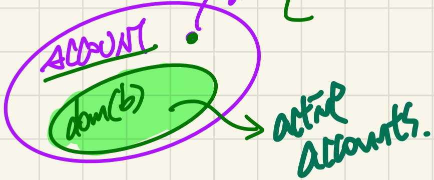
```
withdraw: not extended ordinary ›(REQ6)
ANY
  ∘ a   ›account to withdraw
  ∘ v   ›value to withdraw
WHERE
  ∘ type_of_a: a ∈ ACCOUNT not theorem ›typing constraint of event parameter a
  ∘ type_of_v: v ∈ ℕ1  not theorem ›typing constraint of event parameter v
THEN
  ∘ act1: b(a) ≔ b(a) − v ›updates the balance of a
  ∘ act2: d ≔ d − v ›updates the cash drawer
END
```

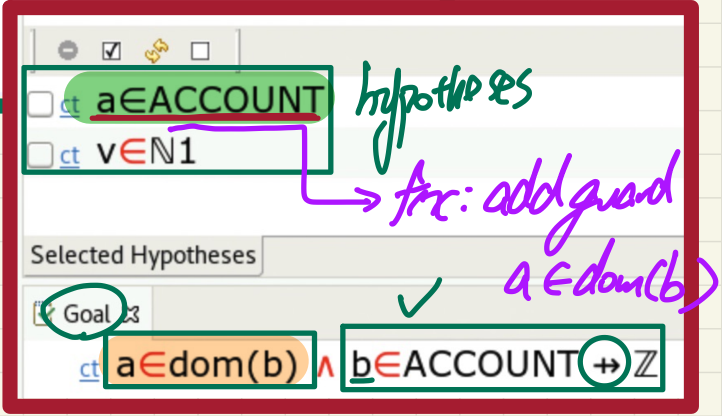
✓

$$b \Leftarrow\!\!\!\!\!\lhd \{(a, \boxed{b(a)} - v)\}$$

↳ function application
↳ precondition:
$a \in dom(b)$

not active account





□ ct a∈ACCOUNT
□ ct v∈ℕ1

Goal: ct a∈dom(b) ∧ b∈ACCOUNT ⇸ ℤ

hypotheses

fix: add guard $a \in dom(b)$

✓

$a \in ACCOUNT \land v \in \mathbb{N}1$
$\Rightarrow a \in dom(b)$

(ENV3)

e.g. 200

$b(a) \geq -200$

All accounts' balance values $\geq$ credit limit $(\geq \underline{\underline{-C}})$

$$\boxed{\forall a .\ a \in dom(b) \Rightarrow b(\underline{\underline{a}}) \geq -C}$$

Proof Obligation: withdraw/inv3

$$b(a) := b(a) - v$$

$$\equiv$$

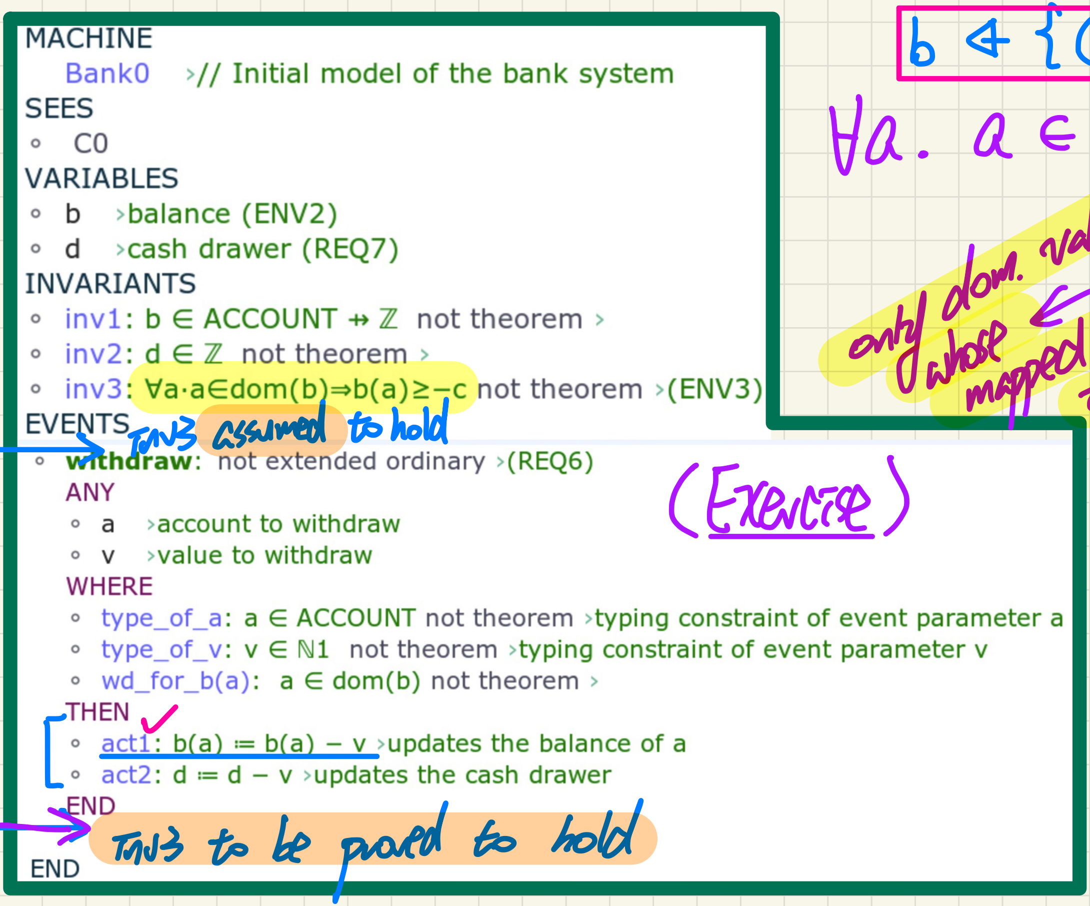
$$\boxed{b \mathbin{<\!\!+} \{(a, b(a) - v)\}}$$

$$\forall a.\ a \in dom(\cancel{b}) \Rightarrow b(a) \geq -c$$

$$b \mathbin{<\!\!+} \{(a, b(a) - v)\}$$

only dom. value whose mapped value in var. got changed

inv3 (assumed to hold)



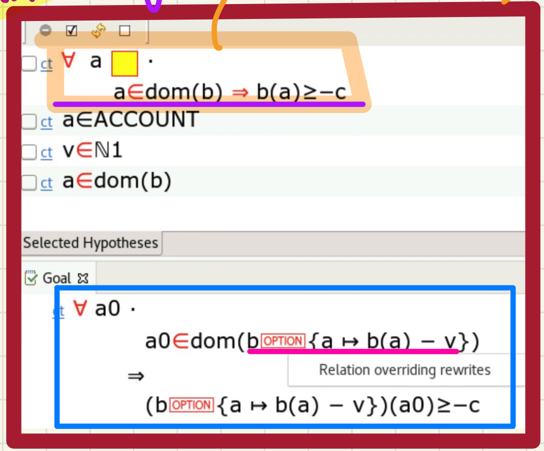
```
MACHINE
    Bank0   ›// Initial model of the bank system
SEES
 ◦   C0
VARIABLES
 ◦  b    ›balance (ENV2)
 ◦  d    ›cash drawer (REQ7)
INVARIANTS
 ◦  inv1: b ∈ ACCOUNT ⇸ ℤ  not theorem ›
 ◦  inv2: d ∈ ℤ  not theorem ›
 ◦  inv3: ∀a·a∈dom(b)⇒b(a)≥−c not theorem ›(ENV3)
EVENTS
 ◦  withdraw:  not extended ordinary ›(REQ6)
    ANY
     ◦  a    ›account to withdraw
     ◦  v    ›value to withdraw
    WHERE
     ◦  type_of_a: a ∈ ACCOUNT not theorem ›typing constraint of event parameter a
     ◦  type_of_v: v ∈ ℕ1  not theorem ›typing constraint of event parameter v
     ◦  wd_for_b(a):  a ∈ dom(b) not theorem ›
    THEN
     ◦  act1: b(a) ≔ b(a) − v ›updates the balance of a
     ◦  act2: d ≔ d − v ›updates the cash drawer
    END
END
```

inv3 assumed to hold

✓

inv3 to be proved to hold

(Exercise)



ct ∀ a □ ·
a∈dom(b) ⇒ b(a)≥−c

ct a∈ACCOUNT

ct v∈ℕ1

ct a∈dom(b)

∀ a0 ·
a0∈dom(b OPTION {a ↦ b(a) − v})
⇒   Relation overriding rewrites
(b OPTION {a ↦ b(a) − v})(a0)≥−c