model

CUSTOMER +

account

ACCOUNT

customer_with_name, ...

BANK

zero, one, total: VALUE
count: INTEGER
customers: LIST[CUSTOMERS]

new(name1: STRING)

name1 ∉ customers

[?] sum_of_all_balances = old sum_of_all_balances
and count = old count +1
and total = old total
[!] and name1.name = name
and name1.balance = zero
and customers_unchanged_other_than name1

deposit(a_name:STRING; a_value: VALUE)
a_name ∈ customers
and a_value > 0

[?] count = old count
and total = old total + a_value
a_name.balance > old a_name.balance
[!] and customers_unchanged_other_than a_name
and sum_of_all_balances = old sum_of_all_balances +
a_value

withdraw(a_name:STRING; a_value: VALUE)

a_name ∈ customers
and a_value > 0
[?] and a_value ≤ a_name.balance

count = old count
and total = old total - a_value
[!] a_name.balance < old a_name.balance
and customers_unchanged_other_than a_name
and sum_of_all_balances = old sum_of_all_balances -
a_value

transfer(name1: STRING; name2:STRING; a_value: VALUE)

name1 ≠ name2
[?] name1 ∈ customers
name2 ∈ customers

[!] invariant

zero= 0 and 1 = 1
customers = customers.count