Exercise 1: Abstract Data Types

MASTER SOLUTION

1.) Balances of accounts are never negative

There are three functions that yield BANK_ACCOUNT: “new_account”, “deposit” and “withdraw”. For each of these functions, we have to prove that the balance of its return value is never negative:

balance (new_account (o)) ≥ 0

This can be easily derived from the axiom “balance (new_account (o)) = 0”.

balance (deposit (a, v)) ≥ 0

Using the axiom “balance (deposit (a, v)) = balance (a) + v”, we have to show that “balance (a) + v ≥ 0”. From the precondition of deposit, we know that “v ≥ 0”. Using the structural assumption, we know that “balance (a) ≥ 0”. “(v ≥ 0) ∧ (balance (a) ≥ 0) ⇒ (balance (a) + v ≥ 0)” always holds.

balance (withdraw (a, v)) ≥ 0

Using the axiom “balance (withdraw (a, v)) = balance (a) - v”, we have to show that “balance (a) - v ≥ 0”. From the precondition of withdraw, we know that “balance (a) ≥ v”. “(balance (a) ≥ v) ⇒ (balance (a) - v ≥ 0)” always holds.

2.) Sufficient completeness

The following axioms are missing:

- owner (deposit (a, v)) = owner (a)  (axiom 5)
- owner (withdraw (a, v)) = owner (a)  (axiom 6)

Otherwise it is not possible to reduce the term “owner (deposit (a, v))” any further.

The two queries available in the ADT are “balance” and “owner”. We can prove sufficient completeness by showing that each term that uses balance and owner can be either reduced to a shorter term using BANK_ACCOUNT (for all commands) or to a term not using BANK_ACCOUNT at all (for all creators):

- balance (new_account (o)) = 0  (using axiom 1)
- balance (deposit (a, v)) = balance (a) + v  (using axiom 3)
- balance (withdraw (a, v)) = balance (a) - v  (using axiom 4)
- owner (new_account (o)) = o  (using axiom 2)
- owner (deposit (a, v)) = owner (a)  (using axiom 5)
- owner (withdraw (a, v)) = owner (a)  (using axiom 6)
3.) Transfer money function

TYPES
   BANK_ACCOUNT_PAIR

FUNCTIONS
   transfer: BANK_ACCOUNT × BANK_ACCOUNT × INTEGER ↦ BANK_ACCOUNT_PAIR
   source: BANK_ACCOUNT_PAIR → BANK_ACCOUNT
   target: BANK_ACCOUNT_PAIR → BANK_ACCOUNT

PRECONDITIONS (with \( v \in \text{INTEGER}, a_1, a_2 \in \text{BANK_ACCOUNT} \))
   transfer \((a_1, a_2, v)\) require balance \((a_1) \geq v\) and \(v \geq 0\) and \(a_1 \neq a_2\)

AXIOMS
   source (transfer \((a_1, a_2, v)\)) = withdraw \((a_1, v)\)
   target (transfer \((a_1, a_2, v)\)) = deposit \((a_2, v)\)