Exercise 1: Abstract Data Types

MASTER SOLUTION

1.) Accounts are never negative

There are three functions that yield BANK_ACCOUNT: “new_account”, “deposit” and “withdraw”. For each one of the three functions, we have to prove that balance to 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*) + $v \geq 0$”. From the precondition of deposit, we know that “$v \geq 0$”. Using the structural assumption, we know that “balance (*a*) ≥ 0”. “($v \geq 0$) $\wedge$ (balance (*a*) ≥ 0) $\Rightarrow$ (balance (*a*) + $v \geq 0$)” always holds.

balance (withdraw (*a, v*)) ≥ 0

Using the axiom “balance (withdraw (*a*, *v*)) = balance (*a*) - *v*”, we have to show that “balance (*a*) - $v \geq 0$”. From the precondition of withdraw, we know that “balance (*a*) $\geq v$”. “(balance (*a*) $\geq v$) $\Rightarrow$ (balance (*a*) - $v \geq 0$)” 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 (see slides) 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 5)

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 $v \in$ INTEGER, $a1, a2 \in$ BANK_ACCOUNT)

transfer ($a1$, $a2$, v) **requires** balance ($a1$) $\geq v$ **and** $v \geq 0$ **and** $a1 \neq a2$

AXIOMS

source (transfer ($a1$, $a2$, v)) = withdraw ($a1$, v)
target (transfer ($a1$, $a2$, v)) = deposit ($a2$, v)