

ADT Exercise Solution

Prove by structural induction

Base case:

$$\text{balance}(\text{new_account}(o)) \geq 0$$

Step case:

Assume $\text{balance}(e) \geq 0$ where e is of n nested levels

Prove that $\text{balance}(e') \geq 0$ where expression e' is of $n+1$ nested levels

Exercise 1: Prove for all bank account b : $\text{balance}(b) > 0$

Prove:

Base case: $\text{balance}(\text{new_account}(o)) \geq 0$ axiom 1

Step case:

e' can be of either form: $\text{deposit}(e, v)$ or $\text{withdraw}(e, v)$

Case 1: e' is $\text{deposit}(e, v)$

$\text{balance}(\text{deposit}(e, v)) = \text{balance}(e) + v$	axiom3
$\text{balance}(e) \geq 0$	induction hypothesis
$v \geq 0$	precondition of deposit
$\text{balance}(e) + v \geq 0$	

Case 2: e' is $\text{withdraw}(e, v)$

$\text{balance}(e') = \text{balance}(\text{withdraw}(e, v))$	axiom 4
$\text{balance}(e) \geq 0$	induction hypothesis
$v \geq 0$ and $\text{balance}(e) \geq v$	precondition of withdraw
$\text{balance}(e) - v \geq 0$	

QED

What is completeness anyway?

For mathematician, a theory is complete if its axioms and rules of inference are powerful enough to prove the truth or falsity of any formula that can be expressed in the language of the theory.

In First Order Logic (FOL):

- Only expressible (also called well-formed) formulae need to be considered, $a \wedge b$, but not $a \wedge \wedge b$.
- Value of a formula can be decided: True or False

Transpose math concepts into ADT specification

In math

- Only well-formed formulae need to be considered
- Value of a formula can be decided: True or False



In ADT

- Only well-formed expressions need to be considered
- Value of an expression can be decided.

Let's play with our transposition a bit

Functions

deposit: $BANK_ACCOUNT \times INTEGER \mapsto BANK_ACCOUNT$

withdraw: $BANK_ACCOUNT \times INTEGER \mapsto BANK_ACCOUNT$

Preconditions

deposit (a, v) requires $v \geq 0$

withdraw (a, v) requires $balance(a) \geq v$ and $v \geq 0$

Do we need to consider?

deposit (a, 1)

withdraw (deposit (a, 100), 50)

deposit (a, -200)

Correct ADT expression

Let $f(x_1, \dots, x_n)$ be a **well-formed** expression involving one or more functions on a certain ADT. This expression is correct if and only if all the x_i are (recursively) correct, and their values **satisfy the precondition** of f , if any.

Definition: sufficient completeness

An ADT specification for a type **T** is **sufficiently complete** if and only if the axioms of the theory make it possible to solve the following problems for any well-formed expression **e**:

S1 Determine whether **e** is **correct**.

S2 If **e** is a query expression and has been shown to be correct under S1, express **e**'s value under a form **not involving** any value of type **T**.

Exercise 2

Using existing axioms, can we decide the value of:

owner (deposit (a , v))

owner (withdraw (a , v))

So we add the following:

owner (deposit (a , v)) = owner (a) (axiom 5)

owner (withdraw (a , v)) = owner (a) (axiom 6)

Prove sufficient completeness

We need to show that we can:

- S1 Determine correctness of well-formed expression e .
- S2 Decide the value of e , if e is correct query.

Prove sufficient completeness by structural induction

Base case: `new_account` is correct

`balance (new_account (o))` is correct and value is `0`

`owner (new_account (o))` is correct and value is `o`

Step case: assume expression `e` with `n` nested level is correct, prove that we can decide whether `e'` with `n+1` nested level:

- `balance (e')` is correct
- `owner(e')` is correct
- `deposit (e', v)` is correct
- `withdraw (e', v)` is correct

And when `e'` is correct and if `e'` is an expression, show we can decide its value.

Prove sufficient completeness – Base case

Base case: to prove

- `new_account` is correct

Because `new_account (o)` is the shortest expression possible and it has no precondition, so it is correct.

Prove sufficient completeness – Base case

Base case: to prove

- `balance (new_account (o))` is correct and value is 0

Prove:

`balance (new_account (o))` is correct because `new_account(o)` is correct and `balance` has no precondition.

Value of `balance(new_account(o))` is 0 axiom1

Prove sufficient completeness – Base case

Base case: to prove

- $\text{owner}(\text{new_account}(o))$ is correct and value is o

Prove:

$\text{owner}(\text{new_account}(o))$ is correct because $\text{new_account}(o)$ is correct and owner has no precondition.

Value of $\text{owner}(\text{new_account}(o))$ is o axiom2

Prove sufficient completeness – Step case

Assume expression e with n nested levels is correct and if expression, value can be decided.

Prove that correctness of $\text{balance}(e')$ can be decided, and show its value, where e' is with $n+1$ nested levels.

Prove:

e' can be of either form: $\text{deposit}(e, v)$ or $\text{withdraw}(e, v)$

Case 1: e' is $\text{deposit}(e, v)$

$\text{balance}(e')$ is $\text{balance}(\text{deposit}(e, v))$, it is correct iff $v \geq 0$.

When e' is correct, its value is:

$$\text{balance}(\text{deposit}(e, v)) = \text{balance}(e) + v \quad \text{axiom3}$$

Case 2: e' is $\text{withdraw}(e, v)$

$\text{balance}(e')$ is $\text{balance}(\text{withdraw}(e, v))$,

it is correct iff $v \geq 0$ and $\text{balance}(e) \geq v$.

When e' is correct, its value is:

$$\text{balance}(\text{withdraw}(e, v)) = \text{balance}(e) - v \quad \text{axiom 4}$$

Prove sufficient completeness – Step case

Assume expression e with n nested levels is correct and if expression, value can be decided.

Prove that correctness of $\text{owner}(e')$ can be decided, and show its value, where e' is with $n+1$ nested levels.

Prove:

e' can be of either form: $\text{deposit}(e, v)$ or $\text{withdraw}(e, v)$

Case 1: e' is $\text{deposit}(e, v)$

$\text{owner}(e')$ is $\text{owner}(\text{deposit}(e, v))$, it is correct when e is correct and $v \geq 0$.

When e' is correct, its value is:

$$\text{owner}(\text{deposit}(e, v)) = \text{owner}(e) \quad \text{axiom5}$$

Case 2: e' is $\text{withdraw}(e, v)$

$\text{owner}(e')$ is $\text{owner}(\text{withdraw}(e, v))$,

it is correct when e is correct and $v \geq 0$ and $\text{balance}(e) \geq v$.

When e' is correct, its value is:

$$\text{owner}(\text{withdraw}(e, v)) = \text{owner}(e) \quad \text{axiom 6}$$

Prove sufficient completeness – Step case

Assume expression e with n nested levels is correct

Prove that correctness of e' with $n+1$ nested level can be decided, where e' is $\text{deposit}(e, v)$

Prove:

$\text{deposit}(e, v)$ is correct iff e is correct and $v \geq 0$

Prove sufficient completeness – Step case

Assume expression e with n nested levels is correct

Prove that correctness of e' with $n+1$ nested level can be decided, where e' is $\text{withdraw}(e, v)$

Prove:

$\text{withdraw}(e, v)$ is correct iff e is correct and $v \geq 0$ and $\text{balance}(e) \geq v$

QED

Exercise 3: Transfer money

TYPES

ACCOUNT_PAIR

FUNCTIONS

transfer: ACCOUNT \times ACCOUNT \times INTEGER \mapsto ACCOUNT_PAIR

source: ACCOUNT_PAIR \rightarrow ACCOUNT

target: ACCOUNT_PAIR \rightarrow ACCOUNT

PRECONDITIONS (with $v \in \text{INTEGER}$, $a1, a2 \in \text{ACCOUNT}$)

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

AXIOMS

source (transfer ($a1, a2, v$)) = $a1$

target (transfer ($a1, a2, v$)) = $a2$

Transfer money

TYPES

ACCOUNT_PAIR

FUNCTIONS

transfer: ACCOUNT \times ACCOUNT \times INTEGER \mapsto ACCOUNT_PAIR

source: ACCOUNT_PAIR \rightarrow ACCOUNT

target: ACCOUNT_PAIR \rightarrow ACCOUNT

PRECONDITIONS (with $v \in \text{INTEGER}$, $a1, a2 \in \text{ACCOUNT}$)

transfer ($a1, a2, v$) require 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 ($a2, v$)