Bank Account Example

First we must define some basic global data types.

Basic data types: [INT, STRING].

Then we can define the schema "Person". It will have two state variables (data): name and address.

\[
\begin{align*}
\text{Person} & \\
\text{name} & : \text{STRING} \\
\text{address} & : \text{STRING}
\end{align*}
\]

No constraints are needed here.

Our second schema is "Account". State variables (data): owner, balance.

\[
\begin{align*}
\text{Account} & \\
\text{owner} & : \text{Person} \\
\text{balance} & : \text{INT} \\
\text{balance} & \geq 0
\end{align*}
\]

We must define the constraint that balance cannot be negative.

The last state schema that we will define here is "Bank".

\[
\begin{align*}
\text{Bank} & \\
\text{ownership} & : \text{Person} \leftrightarrow \text{Account}
\end{align*}
\]

The relation "ownership" defines a set of pairs (person, account). We allow one person to have multiple bank accounts.

Now we have to define schemas for two operations: withdraw and deposit.

\[
\begin{align*}
\text{Withdraw} & \\
\Delta \text{Account} & \\
\text{amount}? & : \text{NAT} \\
\text{person}? & : \text{Person} \\
\text{person}? & = \text{owner} \\
\text{balance}' & = \text{balance} - \text{amount}
\end{align*}
\]

Here, the condition \( \text{person}? = \text{owner} \) represents a precondition for the operation, and the expression \( \text{balance}' = \text{balance} - \text{amount} \) captures its effect.

The operation "deposit" can be defined in a similar way.

Concrete values (constants) may be defined using a schema like this:

\[
\begin{align*}
\text{JoeDoe} & \\
\text{Person} & \\
\text{name} & = \text{Joe Doe} \\
\text{address} & = \text{New York}
\end{align*}
\]
We include the schema "Person", effectively reusing all the declared state variables.

Instance of the "Bank" schema:

\[
\begin{align*}
&\text{GoldmanSachs} \\
&\text{Bank} \\
&\text{ownership} = \{(\text{JoeDoe}, \text{AccountJD})\}
\end{align*}
\]

We assume that "AccountJD" is an existing constant of the schema "Account".