

<http://www.labri.fr/~sutre/Teaching/B>

## 1 Specification of a Vending Machine

The first part of this lab session is devoted to the specification of a vending machine. For our purposes, a vending machine is, informally, an automatic dispenser that gives products in exchange for payment. For simplicity, we shall assume that payment is performed by inserting a number of “credits”. The vending machine shall have two modes of operation: *on* and *off*.

**Exercise 1** *Using your experience of vending machines, and your imagination, give an informal specification of the vending machine.*

**Exercise 2** *Write, within Atelier B, an abstract machine specifying the vending machine. Availability of products dispensed by the vending machine should be modeled qualitatively, as a subset of the set of all products.*

**Exercise 3** *Add in your machine the requirement that the stored number of inserted credits is zero when the vending machine is off.*

## 2 Refinement of the Vending Machine

The specification was, purposely, imprecise on the availability of products. This led to non-determinism in some operations of the vending machine. The first refinement will consist in replacing the qualitative stock of products by a quantitative one. This shall render the operation **dispense** fully deterministic.

**Exercise 4** *Refine the abstract machine of the previous exercise. Availability of products shall be refined by a partial function from the set of all products to the set of strictly positive integers.*

In the previous refinement, a partial function is used to model the stock of products. To bring the module closer to an implementation, this partial function will be replaced by a total function.

**Exercise 5** *Refine the previous refinement. The stock of products shall be refined by a total function from the set of all products to the set of nonnegative integers.*

## 3 Implementation of the Vending Machine

The refinement written in the previous exercise should be very close to an implementation. Recall that an implementation should give *values* to deferred sets and concrete constants. Moreover, abstract variables and abstract constants are forbidden in an implementation.

**Exercise 6** *Write an implementation refining the refinement of Exercise 5. The implementation must pass B0 checking.*