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

This lab session is the continuation of the previous one. You may either use your vending machine development, or the solution available from the web page.

1 Separate Module for Product Management

Since the set of products is specified as a deferred set in the abstract machine, some machinery is needed to obtain values of this abstract set. Here, products shall be numbered, and a new operation shall be provided to get the product corresponding to a given number. This warrants the introduction of a new module to manage products.

```
MACHINE Product

SETS

PRODUCT

CONSTANTS

price, product_count

PROPERTIES

price : PRODUCT --> NAT1 &

product_count : NAT1 &

product_count = card(PRODUCT)

OPERATIONS

pp <-- product (nn) =

PRE nn : 1..product_count THEN

pp :: PRODUCT

END

END
```

Exercise 1 Add to your Atelier B project the above machine Product, and write an implementation for it. Use the Product module in the vending machine module, via SEES clauses.

2 Command-Line User Interface and Execution

It is now time to write a Main module, providing a user interface for the vending machine. For compatibility with the translator, this module should have a single operation, for instance:

```
MACHINE Main
OPERATIONS
do_main = BEGIN skip END
END
```

Some operations in the vending machine have a pre-condition involving the credit variable. The user interface has to make sure that pre-conditions are satisfied before calling operations. To this end, the variable credit of the vending machine is made visible as follows.

Exercise 2 Declare the variable credit of the abstract machine specifying the vending machine in the CONCRETE_VARIABLES clause. Adjust the refinements.

Exercise 3 Write an implementation for the Main module. Translate¹ your project into C, compile the resulting code, and execute it.

 $^{^1 \}rm Use$ the home-grown C4B front-end <code>~gsutre/c4b/translate.sh</code>.