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

1 Algorithmic Refinement

This lab session is devoted to the specification and implementation of standalone operations. For each exercise below,

1. Specify the desired operation in a B abstract machine of the following form:

```
MACHINE
Mach
OPERATIONS
res <-- oper (xx) =
...
END
```

2. Write a B implementation for this abstract machine, and prove, with the interactive prover, the proof obligations generated by Atelier B. To simplify the development, *ignore all B0-related error messages*.

1.1 Divisibility

We consider the following B abstract machine:

```
MACHINE

Divisibility

DEFINITIONS

Divides (d, n) == (#kk . (kk : NATURAL & (n) = kk*(d)))

OPERATIONS

res <-- divides (dd, nn) =

PRE

dd : NAT &

nn : NAT

THEN

res := bool(Divides(dd, nn))

END

END
```

Exercise 1 Implement the machine Divisibility using the modulo (mod) operator.

1.2 Square Root

Exercise 2 Specify and implement an operation returning the square root $\lfloor \sqrt{x} \rfloor$ of a given natural number $x \in NAT$.

1.3 Array Maximum

Exercise 3 Specify and implement an operation returning the maximum of a given array $a \in 1..n \rightarrow INT$, where $n \in NAT$ is a (constant) parameter such that $n \ge 1$.

1.4 Divisibility Revisited

Exercise 4 Implement the operation divides without using modulo (mod) or division (/).