Isabelle/HOL Exercises
Projects

BIGNAT – Specification and Verification

Hardware platforms have a limit on the largest number they can represent. This is normally fixed by the bit lengths of registers and ALUs used. In order to be able to perform calculations that require arbitrarily large numbers, the provided arithmetic operations need to be extended in order for them to work on an abstract data type representing numbers of arbitrary size.

In this exercise we will build and verify an implementation for BIGNAT, an abstract data type representing natural numbers of arbitrary size.

Representation

A BIGNAT is represented as a list of natural numbers in a range supported by the target machine. In our case, this will be all natural numbers in the range [0, BASE-1]. (Note: natural numbers in Isabelle are of arbitrary size.)

```
types
  bigNat = "nat list"
```

Define a function *valid* that takes a value for BASE, and checks if the given BIGNAT is valid.

```
consts valid :: "nat ⇒ bigNat ⇒ bool"
```

Define a function *val* that takes a BIGNAT and its corresponding BASE, and returns the natural number represented by it.

```
consts val :: "nat ⇒ bigNat ⇒ nat"
```

Addition

Define a function *add* that adds two BIGNATs with the same value for BASE. Make sure that your algorithm preserves the validity of the BIGNAT representation.

```
consts add :: "nat ⇒ bigNat ⇒ bigNat ⇒ bigNat"
```

Using *val*, verify formally that your function *add* computes the sum of two BIGNATs correctly.

Using `valid`, verify formally that your function `add` preserves the validity of the BIGNAT representation.

Multiplication

Define a function `mult` that multiplies two BIGNATs with the same value for BASE. You may use `add`, but not so often as to make the solution trivial. Make sure that your algorithm preserves the validity of the BIGNAT representation.

```
consts mult :: "nat ⇒ bigNat ⇒ bigNat ⇒ bigNat"
```

Using `val`, verify formally that your function `mult` computes the product of two BIGNATs correctly.

Using `valid`, verify formally that your function `mult` preserves the validity of the BIGNAT representation.