Computing and Software 734
Formalized Mathematics

William M. Farmer
McMaster University, Winter 2014

Exercise 3

20 pts.

Due 25 February 2014

Assigned: 7 February 2014
Revised: 7 February 2014

Part 1. Using your chosen proof assistant, create a theory of an abstract monoid called something like Monoid. The language should have a type called mon, a constant e of type mon, and a constant mul of type

$$\mathsf{mon} \to (\mathsf{mon} \to \mathsf{mon}).$$

The axioms should say that mul is associative and that e is a left and right identity element with respect to mul. If possible, make mul an infix operator.

Part 2. Define a constant prod of type

$$\mathbb{Z} \to (\mathbb{Z} \to ((\mathbb{Z} \to \mathsf{mon}) \to \mathsf{mon}))$$

that denotes the iterated operation for mul, usually written as $\prod_{i=m}^{n} f(i)$, where m and n are integers and f is function from integers to monoid elements. ($\mathbb{Z}$ is the type of integers.) Prove the following lemmas:

1. $\forall\, m, n : \mathbb{Z}, f : \mathbb{Z} \to \mathsf{mon}\ .$
   $m < n \Rightarrow \mathsf{prod}(m)(n)(f) = f(m)\ \mathsf{mul}\ \mathsf{prod}(m+1)(n)(f).$
2. $\forall\, m, n : \mathbb{Z}, f : \mathbb{Z} \to \mathsf{mon}\ .\ m = n \Rightarrow \mathsf{prod}(m)(n)(f) = f(m).$
3. $\forall\, m, n : \mathbb{Z}, f : \mathbb{Z} \to \mathsf{mon}\ .\ m > n \Rightarrow \mathsf{prod}(m)(n)(f) = \mathsf{e}.$

Part 3. Create an extension of this theory called something like ComMonoid that includes an axiom that says mul is commutative. In this theory prove the following lemma:

$$\forall\, m, n : \mathbb{Z}, f, g : \mathbb{Z} \to \mathsf{mon}\ .$$
$$\mathsf{prod}(m)(n)(f)\ \mathsf{mul}\ \mathsf{prod}(m)(n)(g) = \mathsf{prod}(m, n, \lambda i : \mathbb{Z}\ .\ f(i)\ \mathsf{mul}\ g(i)).$$

Send the instructor the files you produced with comments and instructions on how they can be loaded and checked.