Higher order programming using *Mathematica* in teaching programming

Hiroshi Ohtsuka
Graduate School of Mathematics, Kyushu University
ohtsuka@math.kyushu-u.ac.jp

Abstract:

In teaching and examination of programming for graduates in mathematics course, one of the important features for programming languages is the ability to treat functions as "first–order" objects. *Mathematica* as well as functional programming languages provides this feature as higher order functions, which take functions as arguments or return functions as results. `D` and `Integrate` in *Mathematica* intrinsic mathematical functions are simple examples of higher order functions. Other but different type of examples are `Map` and `Apply`, which are used in many places of higher order functional programming.

This paper is separated in two parts both of which are focussed on higher order programming in *Mathematica*. In the first part, we shall use examples to present advantages of this feature compared to procedural programming languages such as *C*. These examples given with results of questionnaires over graduates are some numerical algorithms and their analysis used in our examination for them.

In the second part, we turn our attention to the relation between this feature and other functional features, type structure and evaluation strategy, in *Mathematica*. As programs (functions) of *Mathematica* work without explicit typing and *Mathematica* evaluates them with applicative order by default, we sometimes must be absorbed in their types and how they evaluated. We can analyze these properties for *Mathematica* programs using abstract interpretation and meta–programming in *Mathematica*.