OLEG GOLUBITSKY, Ontario Research Centre for Computer Algebra, London, Ontario, Canada, N6A 5B7

Implementation of Arithmetics in Aldor

Aldor is a programming language designed for implementing symbolic computation algorithms. On the one hand, it allows to encode mathematical structures and algorithms in a natural way, exhibiting their essence and hiding various lower-level programming issues. On the other hand, the Aldor compiler allows to use machine resources much more efficiently than computer algebra systems.

The efficiency of implementation of most symbolic methods significantly depends on the efficiency of basic arithmetic operations. The latter include computations with fixed size integers, residue classes modulo an integer, arbitrary precision integers, single and double floating point numbers, and arbitrary precision floating point numbers.

We will discuss a new implementation of arithmetic types in Aldor, which combines and extends two existing Aldor libraries, AxlLib and LibAldor. In this implementation, we refine the hierarchy of arithmetic types in an attempt to reflect structural relationships between them more naturally. We also solve the issue of compatibility with multiple platforms, on which the machine arithmetic types may have different sizes. In particular, we develop a general framework for doubling the size of a given machine integer type or restricting it to an integer of smaller size (including a new algorithm for dividing double integers). We show how the Aldor optimizer translates this generic object-oriented code into highly efficient machine code by in-lining small subroutines and unfolding records.

This work is part of a larger project on unifying existing Aldor libraries and providing one efficient general-purpose library for implementing computer algebra algorithms.

This is joint work with Stephen Watt.