Conservation of Data

Ian Mackie

Conservation of energy, conservation of momentum, conservation of mass, and numerous others from physics, chemistry and biology provide us with a number of fundamental laws of nature.

In 1961, Rolf Landauer [1] discovered that energy is needed to erase data. This lead to much work on reversible computation (and quantum computation) to avoid erasing data, and similar arguments have been made for avoiding copying data also.

These observations and works have motivated our investigation into an approach to computation where data is conserved: programs do not erase or copy data (i.e., the program inputs are not copied or erased).

We review some of the work on reversible and quantum computation to avoid erasing data, and put forward other approaches such as linearity and directly conserving data by defining new models. We suggest an approach where syntax and semantics are developed hand-in-hand so that data conservation is also reflected in the syntax.

The work is parametrised on what we take as the data that we conserve: bits, memory locations, data structures, programs, etc., and also which kind of computational model we wish to use. We briefly review the appropriateness of term rewriting systems, lambda calculus, interaction nets, chemical abstract machine, and most importantly, geometry of interaction, as well as suggesting some directions for new models.

To justify that the general ideas can work, we give some examples of models of computation that conserve data, and show several programming examples (sorting algorithms, Dijkstra's partition algorithm). In these examples the syntax and semantics both reflect the use of data in the program, and thus we obtain a visually appealing syntax for in-place algorithms.

We conclude the talk with some suggestions of adapting existing models of computation and developing new models and syntax. Developing a programming language syntax at the same time as understanding the problem semantically seems fruitful, and programming directly with the computational model avoids the potential distinction as to what is conserved. Much of the effort is currently being applied in these areas, where we envisage applications to topics such as in-place algorithm design, and compiler technology for optimising these algorithms.

References

[1] R. Landauer. Irreversibility and heat generation in the computing process. *IBM Journal of Research and Development*, 5(3), 1961.