Grand Challenge – Exploiting Division by Zero

Dr James A.D.W. Anderson

Mathematics has a long and complex history, interweaving formal developments with practical applications. But one theme that emerges is a continuing effort to remove exceptions from mathematics so as to develop total systems that always work. In our computerised society, where machines impact every aspect of our lives, total systems of computation are more important than ever before.

In the past few millennia, mankind has worked out how to make many arithmetical operations total. For example, addition, subtraction and multiplication can all be applied to any numbers with the result being a number, but division is partial. Real arithmetic, the theoretical basis of analogue and digital computers, does not allow division by zero.

Division by zero has a long history. In the 8th century, Indian scholars thought they could divide by zero – they were nearly right, but their methods treated infinity inconsistently. In the 17th century, European scholars developed calculus, which comes arbitrarily close to dividing by zero, but does not divide exactly by zero. In the 20th century, a plethora of methods for dividing exactly by zero were developed: methods based on logic, sets, topology, and even extensions of arithmetic. It is now known that while the axioms of real arithmetic forbid division by zero, the algorithms allow it, confirming the views of the ancient Indian scholars, and opening up the possibility of developing new mathematical techniques, new kinds of computers, and new kinds of software that deliver more reliable results at speeds that are fundamentally unachievable by today's partial computers.

The Grand Challenge of Exploiting Division by Zero devolves into a number of sub-challenges, each of which may succeed in its own terms, and all of which build toward a fuller understanding of total systems of computation. Just a few of these challenges are listed below.

Exploiting Division by Zero to Double the Range of Floating-Point Numbers

IEEE Floating-point arithmetic uses a large class of Not-a-Number (NaN) objects to encode some of the infinitely many exceptions which occur in real arithmetic as a result of dividing by zero. Replacing IEEE arithmetic with transreal arithmetic removes all such exceptions and replaces all of the NaN objects with a single number, nullity, thereby freeing up enough binary states in the floating-point representation to double the arithmetical range of real numbers encoded by the floating-point bits. A practical challenge is to cash in this theory by developing such floating-point units. They will improve many kinds of scientific, engineering, medical, and audio visual computations, impacting all of the Computing Grand Challenges identified so far.

Exploiting Division by Zero to Remove all Logical Run-Time Errors

Any total arithmetic is free from arithmetical exceptions. In theory, any digital computation can be Gödelised so that it is turned into a total arithmetical program. Such a program will run without any exceptions, though it might still fail on physical faults. In practice, Gödelised languages are too difficult to use, but it is feasible to design Turing complete processors that implement total functions of their binary states. One way to do this is to establish a bijection from the computational states to the states of a total arithmetic. Such processors are maximally efficient in their use of binary states and execute with no logical exceptions. Physical faults can be detected by comparing program runs in a parallel computer so that there are no reportable exceptions at all – the reporting of a mismatch between program runs being a normal part of the operation of the parallel computer. Such a computer may be programmed using subsets of conventional languages, with the benefit that syntactic checking in the compiler guarantees that there are no semantic faults other than physical faults. It is also feasible to develop specialist languages for such processors. The absence of run-time errors is important in safety critical systems and is necessary for a certain class of computers that execute faster than von Neumann machines.

Exploiting Division by Zero to Execute Programs in Less Than a Clock Tick

If a computation can be implemented as a sequence of in-line instructions with no exceptional states then it can be executed in a pipeline. It is feasible to construct a computer with a pipeline length of a few million instructions that can pipeline across branches and non-recursive subroutine calls and returns. Such a machine can execute many copies of a small, in-line, program in each processor clock tick, and can be made recursive by cycling data from the program's outputs to its inputs. This allows many important scientific and commercial applications to be executed many times per processor clock tick. And it does so using a compiler model, pipelining, that is very well understood and is much simpler than conventional approaches to parallel computation. As a rather pedantic point, neither the von Neumann nor Turing machines, which specify execution of a single instruction in unit time, can match this speed of execution. Sub-clock-tick program execution will impact many Computing Grand Challenges.

Exploiting Division by Zero in Mathematical Physics

Total arithmetics preserve some information at singularities. There are many totalisations of real arithmetic, but there are, currently, no totalisations of complex arithmetic that preserve a lot of information at singularities. If such an arithmetic can be developed then the equations of mathematical physics will apply automatically at singularities and might yield useful information at these points. Such an arithmetic would simplify mathematical physics by removing the need for infinitely many physical censors that stop the universe from adopting singular configurations.

Exploiting Division by Zero in the Public Understanding of Science

The general public responds vociferously to discussion of division by zero. This makes it easy to engage on aspects of mathematics and computing. Transreal arithmetic, in particular, is so simple that it has been taught to 12 year old children. It could be made a part of the secondary school syllabuses for Mathematics and for Information Communication Technology. The Grand Challenge of Exploiting Division by Zero will impact on all of the Computing Grand Challenges and promises to be accessible to scientists and the public of many nations.