The Verifying Compiler: A Grand Challenge for Computing Research

C.A.R. Hoare

Microsoft Research Ltd.,
7 JJ Thomson Ave,
Cambridge CB3 0FB, UK
thoare@microsoft.com

Abstract. I propose a set of criteria which distinguish a grand challenge in science or engineering from the many other kinds of short-term or long-term research problems that engage the interest of scientists and engineers.

The primary purpose of the formulation and promulgation of a grand challenge is to contribute to the advancement of some branch of science or engineering. A grand challenge represents a commitment by a significant section of the research community to work together towards a common goal, agreed to be valuable and achievable by a team effort within a predicted timescale. The challenge is formulated by the researchers themselves as a focus for the research that they wish to pursue in any case, and which they believe can be pursued more effectively by advance planning and co-ordination. Unlike other common kinds of research initiative, a grand challenge should not be triggered by hope of short-term economic, commercial, medical, military or social benefits; and its initiation should not wait for political promotion or for prior allocation of special funding. The goals of the challenge should be purely scientific goals of the advancement of skill and of knowledge. It should appeal not only to the curiosity of scientists and to the ambition of engineers; ideally it should appeal also to the imagination of the general public; thereby it may enlarge the general understanding and appreciation of science, and attract new entrants to a rewarding career in scientific research.

As an example drawn from Computer Science, I revive an old challenge: the construction and application of a verifying compiler that guarantees correctness of a program before running it. A verifying compiler uses automated mathematical and logical reasoning methods to check the correctness of the programs that it compiles. The criterion of correctness is specified by types, assertions, and other redundant annotations that are associated with the code of the program, often inferred automatically, and increasingly often supplied by the original programmer. The compiler will work in combination with other program development and testing tools, to achieve any desired degree of confidence in the structural soundness of the system and the total correctness of its more critical components. The only limit to its use will be set by an evaluation of the cost and benefits of accurate and complete formalization of the criterion of correctness for the software.

H. Kosch, L. Böszörményi, H. Hellwagner (Eds.): Euro-Par 2003, LNCS 2790, p. 1, 2003.