Tony Hoare

From: George Necula [necula@eecs.berkeley.edu]

Sent: 30 April 2003 17:29

To: Tony Hoare; J.C.P.Woodcock; zcc@ios.ac.cn

Cc: Greg Morrisett (jgm@cs.cornell.edu)

Subject: RE:

Hi,

I apologize for the late reply. Please do not take it as a sign of lack of interest. I am extremely pleased to see that some momentum is gathering in this direction. I am also of the impression that technology has matured enough to warrant a new thrust in formal methods.

There is one research direction that should definitely be touched on in such a report: translation validation. The idea is that one can design a compiler such that it is possible to argue, on a compilation-by-compilation basis rather than by verifying the compiler itself, that the semantic meaning of the program is preserved through compilation and optimization. Amir Pnueli started this area and since then it has a small following. There is one workshop (Compiler Optimization Meets Compiler Verification - COCV) where some of the most recent papers in this area have appeared. In addition to the proceedings of the workshop I list below a few papers in this direction.

George.

```
@InProceedings{Pnueli98,
 author      = "Amir Pnueli and M. Siegel and Eli Singerman",
 title       = "Translation Validation",
 booktitle   = "Tools and Algorithms for Construction and Analysis of
               Systems, 4th International Conference, {TACAS} '98",
 pages       = "151--166",
 year        = 1998,
 editor      = "Bernhard Steffen",
 volume      = "LNCS 1384",
 publisher   = "Springer"
}
@InProceedings{Verifix,
 author  = "Wolfgang Goerigk",
 title   = "Towards Rigorous Compiler Implementation Verification",
 editor  = "Rudolf Berghammer",
 year    = 1997,
 booktitle = "Proc. of the 1997 Workshop on Programming Languages and
          Fundamentals of Programming",
 pages = "118--126",
}
@InProceedings{NeculaPLDI00,
 author =      "George C. Necula",
 title =       "Translation Validation for an Optimizing Compiler",
 booktitle =    "ACM SIGPLAN '00 Conference on Programming Language
            Design and Implementation (PDLI)",
 address =     "Vancouver, BC, Canada",
 month =       "18--21 " # jun,
 year =        "2000",
 pages =       "83--94",
 publisher =   "ACM SIGPLAN",
 abstract =    "We describe a translation validation infrastructure
            for the GNU C compiler. During the compilation the
            infrastructure compares the intermediate form of the
            program before and after each compiler pass and
            verifies the preservation of semantics. We discuss a
            general framework that the optimizer can use to
          communicate to the validator what transformations were
          performed. Our implementation however does not rely on
          help from the optimizer and it is quite successful by
          using instead a few heuristics to detect the
          transformations that take place. The main message of
          the paper is that a practical translation validation
          infrastructure, able to check the correctness of many
          of the transformations performed by a realistic
          compiler, can be implemented with about the effort
          typically required to implement one compiler pass. We
          demonstrate this in the context of the GNU C compiler
          for a number of its optimizations while compiling
          realistic programs such as the compiler itself or the
          Linux kernel. We believe that the price of such an
          infrastructure is small, considering the qualitative
          increase in the ability to isolate compilation errors
          during compiler testing and maintenance. (25
          References).",
}

@inproceedings{Cimatti97,
  title =  "A Provably Correct Embedded Verifier for the
           Certification of Safety Critical Software",
  author =  "Alessandro Cimatti and others",
  booktitle = "Computer Aided Verification. 9th International
           Conference. Proceedings",
  publisher =  "Springer-Verlag",
  place =  "Haifa",
  year =  1997,
  month =  jun,
  pages =  "202--213"
}
```

-----Original Message-----
From: Tony Hoare [mailto:thoare@microsoft.com]
Sent: Thursday, April 17, 2003 9:01 AM
To: J.C.P.Woodcock; necula@cs.berkeley.edu; zcc@ios.ac.cn
Subject:

Dear Colleague,

Towards the Verifying Compiler

We define a Verifying Compiler as one that gives high assurance of
the correctness of the programs that it compiles.
Correctness includes any desired observable property of the
program, including safety, security, liveness, dependability,
responsiveness, and performance. The desired properties are specified
by redundant annotations associated with the program, including type
declarations, abstractions, assertions, and descriptions of dynamic
program behaviour. Consistency of the program with its specification is
established automatically by a combination of program analysis, type
inference, model checking, decision procedures, proof search, test case
generation, and any other method that justifies increased trust in the
quality of the program.

1. We are proposing to write a survey article with the above title. It
would summarise and assess the current state of the art in all the
computing technologies that might be relevant to the implementation of
this old challenge in computing science.

We conjecture that the state of the art in these areas is now sufficiently advanced that Computer Scientists could begin to talk about and even begin to fulfil the challenge of constructing a useful Verifying Compiler, especially if it was a collaborative venture. Of course, another main reason for renewed hope in meeting the challenge is the amazing advancement in hardware power and capacity since the last attempts were made, as early as the early seventies.

2. We would be very grateful for your help in ensuring that we are aware of the best of all current theoretical and practical work going on in this field, and especially in your own specialities. Could you please send us a brief list of references to any important work that you regard well, and which might be relevant to our survey paper and to a project for constructing the Compiler? It will be a pleasure to read anything that you recommend.

If you are more deeply interested, you may care to open the attachment. It is a preliminary list of about forty references that we have assembled so far. The other attachment is a brief note describing why the references are believed to be relevant. It may help to stimulate your own thinking, or at least avoid duplicating what we already have. A more detailed account of prospects for a Verifying Compiler is given in reference 46. We could send you an updated draft of it if you like.

3. If you would like to help us even more substantially, we would very much welcome a summary of your own assessment of the relevant state of the art in any of the research areas that specially interests you. Anything from a couple of sentences to a complete section of the paper. Put a star against any of the references that you would be prepared to summarise in this way. We could then share out the work among a larger team of scientists, including those with real specialist knowledge in each of the component technologies.

Any contribution of this kind would be acknowledged by including you among the list of authors of the paper, or in any other way you like. In fact, even if you are willing just to comment on future drafts of the paper, and support its general message, it would be a privilege to have you among its authors.

Yours sincerely,

Tony Hoare and Greg Morrisett