Cooperation of Automated and Interactive Theorem Provers

Bernd, Ingo Dahn

Humboldt-University, Institute of Pure Mathematics
Ziegelstr. 13a, D-10099 Berlin
e-mail : dahn@mathematik.hu-berlin.de

QED is too big a task to be achieved by a single group. Therefore, it will demand the cooperation of different groups which have developed up to now their specific systems for their specific purposes. QED has to use the skills and achievements of these groups. Therefore, the introductory talk discussed possible ways of cooperation of existing and forthcoming theorem provers. There are several technical and social problems to overcome for such a cooperation. These were examined based on the experience of the ILF system within the research program *Schwerpunktprogramm Deduktion* of the Deutsche Forschungsgemeinschaft. Problems of parallelism in deductions have not been touched.

It is argued, that cooperation of provers gives stronger and more flexible systems. Within a cooperation the value of a prover will judged according to its ability to augment other provers.

We can distinguish theoretically the following four types of cooperation.

1 Monolithic

In this approach there is a single system combining the best properties of the subsystems. These subsystems cooperate by procedure calls. There seemed to be a consensus among the workshop participants that this approach is unrealistic.

2 Cooperation in Proof Production

In this approach, provers must be able to incorporate periodically new results produced by other systems during their work. They must output intermediate results that can be useful for others. Recently, within the German *Schwerpunktprogramm Deduktion*, Jörg Denzinger (`denzinge@informatik.uni-kl.de`) has argued that the adaptation of existing theorem provers to such a type of cooperation is a promising task. However, this requires a rather restrictive description of what should be communicated between the systems. Moreover, for existing theorem provers it is often impossible to incorporate more than literals during their work into the basic data structures.

3 Server for Proof Production

In this approach a general purpose - often interactive - system distributes subtasks to its subsystems. These try to solve subtasks without further cooperation. All implemented cooperating systems which are known to the author use this type of cooperation. The effect of this kind of cooperation depends on the distribution of subtasks. This is facilitated, if the specific strengths of the integrated provers are known. Therefore, cooperative provers which take specific properties of a theory or domain into account are best suited here. It is also possible, to use prover competition within the system to determine the suitability of the provers for a given task.

4 Cooperation in Proof Exchange

In this approach, provers supply lemmas that can be loaded by other provers before starting a proof search. Here, the provers can run completely independent at different times and at different locations, connected by a network that gives access to a library of theorems. This is likely to be the most important form of cooperation within the QED project. It requires new tools for proof administration and exchange. It is desirable, to support the combination of proofs from different systems.

Experience in ILF

ILF is an interactive theorem prover which integrates the automated theorem provers Otter, Discount, Setheo and KoMeT. Its development is supported by the Deutsche Forschungsgemeinschaft. In ILF the user can communicate with the automated provers through a uniform graphical user interface. The integrated systems remain completely independent and no change in their code is required. Problems and effects of this cooperation within ILF and of the necessary cooperation with the authors of these systems have been discussed.

ILFs ProofPad

The ProofPad is a configuration of ILF where the user enters lines of a proof and the automated provers in the background try to prove, that each line is a logical consequence of the preceding lines. Therefore, the user does not have to know theorem provers or logical calculi. Experiments in ILF in the field of lattice ordered groups show, that by the use of automated provers it suffices, that the user enters about 10% of the proof interactively to have it formally verified, see a sample manuscript:

```
anonymous ftp from:     info.mathematik.hu-berlin.de ,
                                        (141.20.54.8)
password:               (enter own identification)
in file:                /pub/ilf/busulini.dvi .
```

The Schwerpunktprogramm Deduktion

An important criterion for the projects within the *Schwerpunktprogramm Deduktion* is their ability to cooperate. Moreover the program has special funding for cooperation. All this turned out to be very helpful for the work of an integrating project like ILF. It should also give hints for establishing a cooperation of QED projects. Within ILF, tools have been developed that take demands of other groups in the Schwerpunktprogramm into account. The TreeViewer - a tool for the visualisation and manipulation of directed acyclic graphs - is a separate program that can also be used as part of the graphical user interface of other theorem provers. It can be obtained from:

```
anonymous ftp from:   info.mathematik.hu-berlin.de ,
                                        (141.20.54.8)
password:             (enter own identification)
in file:              /pub/ilf/tview.tar.Z  .
```

The natural language proof presentation facilities of ILF have been developed to be largely independent of a specific calculus. They are made available to other groups by the automated ILF mail server that takes proofs and returns LaTeX sources. You can get more information on the ILF server by sending a mail with text help to `ilf-serv@mathematik.hu-berlin.de`.

Other Problems

All provers currently integrated into ILF prove theorems in fragments of first order logic. The cooperation could be extended provers for other logics if it is known whether provability in one of these logics implies provability in the other logics in use.