The WG, and also its ALGOL 68 Support Subcommittee and its Transput Task Force, met in Summit, New Jersey, at the beginning of April.

On the Algol 68 front, the major event was the acceptance of the Implementation Model of the ALGOL 68 Transput (see AB44.1.1 in this issue). This is intended to make it easier for all implementors of the language to incorporate a correct, compatible and efficient transput system into their compilers. Hans van Vliet worked for one and a half years in modeling and remodeling the transput section of the Revised Report in such a way that it can now be implemented efficiently, still preserving virtually all of the external specifications of the Revised Report. Moreover, his description has the virtues of a good textbook for implementors: it is clear, understandable and precise. The model is based on a buffer concept which provides the proper interface with arbitrary operating systems as they exist nowadays. In recognition of his great achievement, the Model is now always informally referred to as the "Hansput".

Another document accepted at the meeting was the Revised HC Test Set, by D.Grune (see AB44.1.2 in this issue). Originally, this Test Set was prepared as an acceptance test for the CDC ALGOL 68 compiler. It has now been extensively revised, with many new programs added. It is now up to all users who are contemplating purchasing ALGOL 68 compilers to insist that the Test Set be run and an agreed standard for conformance obtained. Only in this way will we gradually force implementors to move towards correct implementations of the language.

Finally, two more Commentaries on the Revised Report were released (see AB44.3.1 in this issue). Now, at last, things seem to have become fairly quiet so far as the appearance of fresh bugs in the Report is concerned.

In spite of all this ALGOL 68 activity, the main part of the meeting was taken up with papers on programming methodology - part of the continuing search for that elusive language "ABSTRACTO. It was decided that the next step should be to prepare an agreed set of example problems against which each proposed "ABSTRACTO language should be tested. In fact, what we need is lots of concrete ABSTRACTO examples to look at.

At this point, I must make my usual plea for material for the next issue. This issue contains two papers which are more or less in the category of "algorithms", such as I asked for in AB41. More material of the same sort would be particularly welcome.

This report has been written by J.C. van Vliet on the request of the Task Force on Transput, which was set up by the Subcommittee on ALGOL 68 Support of IFIP WG2.1. It aims at a precise description of the transput of ALGOL 68, conforming with section 10.3 of the Revised Report. Whereas section 10.3 of the Revised Report describes the intention of transput, the emphasis in this report is on implementability.

A variety of ALGOL 68 implementations exist or are near completion. They all support some kind of transput, although they all differ slightly from each other and from the Revised Report. This diversity renders the transfer of programs from one implementation to the other very difficult, if not virtually impossible.
The existence of so many different transput systems may to some extent be due to the fact that the definition as given in the Revised Report does not really facilitate implementation of the transput. Each implementor again has to struggle his way through the transput section and locate the problems with the particular operating system.

The approach taken is similar to the one in the Revised Report: the transput is described in pseudo-ALGOL 68. The pseudo-ALGOL 68 part can be considered as a language extension which is reasonably implementable. The primitives underlying the model are not defined in ALGOL 68. Instead, their semantics are given in some kind of formalized English, resembling the way in which the semantics of the Revised Report are defined. One advantage of a description in pseudo-ALGOL 68 is that it can largely be tested mechanically. It has been the intention that the ALGOL 68 text, after suitable substitution of the pseudo comments, could be compiled, thereby automatically creating part of the runtime environment.

This report was accepted at the recent meeting of WG2.1 held in Summit, New Jersey, and the WG resolved to ask its parent committee, IFIP TC2, to authorize the following statement for release with it.

This implementation model of the ALGOL 68 transput has been reviewed by IFIP Working Group 2.1. It has been scrutinized to ensure that it correctly interprets the transput as defined in section 10.3 of the Revised Report. This model is recommended as the basis for actual implementations of the transput.

Copies of the Report can be obtained from the Mathematisch Centrum, 2e Boerhaavestraat 49, 1109 AL Amsterdam at a price of HF1 27 (plus postage). Its full title is

Mathematical Centre Tracts No. 111
ALGOL 68 Transput Part II - An Implementation Model.
by J.C. van Vliet.

(Part I, to be published later, will contain some of the background to and motivations for the model. The two parts taken together will constitute Hans van Vliet’s doctoral thesis.) The text of the model is also available in machine-readable form.

AB44.1.2 The Revised MC Test Set.

The Revised MC Test Set comprises 190 ALGOL 68 programs, in part correct ones, in part intentionally incorrect ones. They are designed to explore the full range of ALGOL 68 language features, and include many attempts to trip the compiler up or to uncover incorrect short-cuts.

Many of the programs are pathological and should not be considered as representative of ALGOL 68 programming style. With this in mind, almost all the programs are worth while reading, some as puzzles, some for the good programming features they contain, some for their not widely known programming techniques and a few for their good style.

The test set is not complete, firstly because such a product is never complete: there is no exhaustive testing and one cannot cater for every contingency. Secondly, a few aspects of the language are under-represented (e.g. SHORT and LONG values and bulk I/O). However, if a compiler processes the test set well and also works well on the daily stream of average programs, it may be regarded as a very good compiler. Certainly, all implementors should be encouraged to use it and, especially, to report in their accompanying documentation how it behaved.

The Test Set was accepted at the recent meeting of WG2.1 held in Summit, New Jersey, and the WG resolved to ask its parent committee, IFIP TC2, to
authorize the following statement for release with it.

This ALGOL 68 Test Set has been reviewed by IFIP Working Group 2.1, which wishes to recommend it as a valuable means of testing implementations of ALGOL 68.

Copies of the Test Set will shortly be available from the Mathematisch Centrum, 2e Boerhaavestraat 49, 1109 AL Amsterdam. It will also be available in machine-readable form on most reasonable formats of magnetic tape.

AB44.1.3 TORRIX.

At its meeting in Jablonna, Poland, in August 1978, IFIP WG2.1 authorized the release of the following statement.

The library package TORRIX comprising definitions for handling vectors and matrices in ALGOL 68, as published in the Mathematical Centre Tracts series, has been scrutinized to ensure that:

a) It strictly conforms to the definition of ALGOL 68.

b) It is consistent with the philosophy and orthogonal framework of that language.

c) It addresses a significant application area in a comprehensive and appropriate manner.

In releasing this statement the intention is to encourage the incorporation of this library package in library preludes of ALGOL 68 implementations.

TORRIX is published as Mathematical Centre Tracts No. 86. Volume 1 is currently available. Volume 2 will be available later in the year. See AB42.1.5 for further information.

AB44.1.4 The Revised Report in German.

The Revised Report on the Algorithmic Language ALGOL 68 has now been translated into German by Prof. Immo O. Kerner of the Paedagogische Hochschule Dresden. It is published by Akademie-Verlag, 108 Berlin, Leipziger Str. 3-4 (list number 202, 100/401/78), under the title "Revidierter Bericht über die Algorithmische Sprache ALGOL 68".

Although the text is in German, all hyper-rules and paranotions are given in English, as is the standard-prelude (except for comments).

AB44.1.5 An Axiomatic Semantic Definition of ALGOL 68.

This doctoral thesis, by Richard Schwarz, is obtainable from the Computer Science Department, School of Engineering and Applied Science, University of California, Los Angeles, CA 90024, so long as stocks last (after that, it should be obtainable in microfiche from NTIS, Springfield, Virginia 22151).

The report gives a formal axiomatic definition of a major subset of ALGOL 68. The definition, roughly the same length as the axiomatic definition of EUCLID, handles many features generally considered to be serious impediments to program verification. The small set of very general rules governing the semantics of ALGOL 68 leads to a very clean axiomatic definition, defining an extraordinarily expressive language.

It should be required reading for anyone who still believes that side
parameters and unrestricted aliasing of names are absolute bars to program verification. They are not. All necessary axioms are given here and, because of the orthogonal structure of the language, the axiomatic definition is surprisingly short.

AB44.1.6. Other reports available.

The following reports are available from the Mathematisch Centrum, 2e Boerhaavestraat 49, 1109 AL Amsterdam.


(a tool for use with the CDC ALGOL 68 compiler permitting, especially, ALGOL 68 procedures to be passed as parameters to FORTRAN subroutines).

A Modules and Separate Compilation Facility for ALGOL 68 by C.H. Lindsey and H.J. Boom (Report No. IW 105/78 - HFl 6 plus postage)

(as already published in AB43.3.2).