INTERNATIONAL FEDERATION FOR INFORMATION PROCESSING

TECHNICAL COMMITTEE 2 - PROGRAMMING LANGUAGES

Working Group 2.1 - ALGOL

Third Meeting, 16 - 20 March 1964, Tutzing, Germany.

MINUTES (Unconfirmed)

PARTICIPANTS

Officers:

Chairman - W. L. Van der Poel,
Netherlands
Secretary - R. E. Utman, U. S. A.

Members:

<table>
<thead>
<tr>
<th>Name</th>
<th>Country</th>
<th>Tutzing 16-20 March 1964</th>
<th>Delft 10-13 September 1963</th>
<th>Munich 28, 30 August 1962</th>
</tr>
</thead>
<tbody>
<tr>
<td>F. L. Bauer</td>
<td>Germany</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>I. Dahlstrand</td>
<td>Sweden</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>E. W. Dijkstra</td>
<td>Netherlands</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>F. G. Duncan</td>
<td>Netherlands, UK</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>T. Ekman</td>
<td>Sweden</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>J. Erdwinn</td>
<td>U. S. A.</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>R. W. Floyd</td>
<td>U. S. A.</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>A. Grau</td>
<td>U. S. A.</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>J. Green</td>
<td>U. S. A.</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>C. A. Hoare</td>
<td>U. K.</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>P. Z. Ingerman</td>
<td>U. S. A.</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>E. T. Irons</td>
<td>U. S. A.</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>C. Katz</td>
<td>U. S. A.</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>P. Lucas</td>
<td>Austria</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>J. McCarthy</td>
<td>U. S. A.</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>A. Mazurkiewicz</td>
<td>Poland</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>J. N. Merner</td>
<td>U. S. A.</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>S. Moriguti</td>
<td>Japan</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>P. Naur</td>
<td>Denmark</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>M. Pacelli</td>
<td>Italy</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>M. Paul</td>
<td>Germany</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>W. L. van der Poel</td>
<td>Netherlands</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>H. Rutishauser</td>
<td>Switzerland</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>F. Salle</td>
<td>France</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>K. Samelson</td>
<td>Germany</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>G. Seegmüller</td>
<td>Germany</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>R. E. Utman</td>
<td>U. S. A.</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>B. Vauquois</td>
<td>France</td>
<td>R</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Name of Member</td>
<td>Country</td>
<td>T</td>
<td>D</td>
<td>M</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>A. van Wijngaarden</td>
<td>Netherlands</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>J. Wegstein</td>
<td>U. S. A.</td>
<td>R</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>M. Woodger</td>
<td>UK</td>
<td>x</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>A. P. Yershov</td>
<td>USSR</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Category</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Members Present</td>
<td>20</td>
</tr>
<tr>
<td>Members Absent</td>
<td>9</td>
</tr>
<tr>
<td>Total Membership</td>
<td>29</td>
</tr>
</tbody>
</table>

Observers:
- D. W. Barron
- J. Bolliet
- N. Wirth

Notes:
1. Moriguti represented by H. Wada
2. Moriguti represented by K. Inoue
   R Resigned
1. OPENING OF THE MEETING

1.1 Introductions were made, and it was noted that 20 voting members were in attendance. The Chairman announced an invitation from the Mathematical Institute of the Munich Institute of Technology to the opera for Tuesday evening 17 March, with arrangements to be made by Institute staff.

1.2 IFIP ALGOL Bulletin

The Chairman announced the resignation of Mr. Dijkstra as Editor, and sought recommendations for a replacement. The Chair proposed Mr. Duncan, and he accepted with the unanimous approval of the Group. Since no issues of the Bulletin under the name approved at the first WG 2.1 meeting (ALGOL Periodic Bulletin) have been published, the following questions were considered: (1) changing the name to eliminate "Periodic"? (2) the editorial policy to be followed? Without objection the name was changed back to "ALGOL Bulletin"; to resume with issue #16. The Chair emphasized that WG 2.1 would set Bulletin policy, along the objectives defined at the Delft meeting (second meeting of WG 2.1) as listed in the minutes. He proposed that a clear statement of policy should be drafted by a small ad hoc group later at this meeting. Duncan assured that secretarial administrative support will be available from the Netherlands P.T.T., the Mathematical Center and Electrologica. Both Duncan and the Chair emphasized, however, that a Bulletin requires input papers from the membership, and urged such support.

1.3 Mr. Duncan prepared a statement of his "aims and intentions" as A.B. Editor (See Appendix B), and this was read to the group. Editorial responsibility was discussed, and the followed decision unanimously supported:

"A.B. Policy" The ALGOL Bulletin is edited again under the auspice of IFIP/WG 2.1. Editor is Mr. F. Duncan, who intends to put forward his view of the aims of the A.B. in the first new issue (#16) to appear"

2. APPROVAL OF DELFT MINUTES

2.1 After sharp criticism on the tardiness of the Draft Minutes, the Group and Chair decided to postpone their approval to permit proper study.

2.2 The Draft Minutes were subsequently given provisional approval to enable comment during the interim and final approval at the next WG 2.1 meeting.

2.3 A new policy on WG 2.1 minutes was approved, to simplify their preparation and cut-down the size of the official statement of record of WG 2.1 proceedings:
"Minutes Policy - The minutes of future WG 2.1 meetings shall contain, apart from such matters as lists of attendants and so on, only decisions of the Committee. These decisions consist themselves of two actions, viz. (1) The decision proper, and (2) argumentation and voting results if thought necessary. During the meeting the text of such a contribution to the minutes is written, approved and handed over to the Secretary. The minutes, which will now be practically finished at the end of the meeting, are (then) reproduced and mailed by the Secretary as soon as possible to all members of WG 2.1."

This policy and procedure will be put into practice at the next (or 4th) meeting, since it was not approved until the end of the (Tutzing) meeting, too late to be completely effective in the preparation of minutes thereof.

3. APPROVAL OF THE DRAFT AGENDUM

3.1 The chairman called for approval of the content of the Draft Agendum (see Appendix A), pointing out that Item 9 on the ALGOL Bulletin had already been discussed in part. Except for questions of order and scheduling (including a recommendation from Dijkstra to postpone consideration of items 4, 5, 6, 8, 11 and 12, to the afternoon of the last day) the agendwn was accepted without further objection.

3.2 A schedule of discussions (from 0930-1230 and 1400-1730, Monday through Friday) was followed with subjects pursued as follows:

<table>
<thead>
<tr>
<th>DATE</th>
<th>AGENDA ITEM</th>
<th>SUBJECT</th>
</tr>
</thead>
<tbody>
<tr>
<td>16 March</td>
<td>1.</td>
<td>Introductions</td>
</tr>
<tr>
<td></td>
<td>2.</td>
<td>Minutes</td>
</tr>
<tr>
<td>(1000-1730)</td>
<td>3.</td>
<td>Agendum - Schedule</td>
</tr>
<tr>
<td></td>
<td>4.</td>
<td>Membership</td>
</tr>
<tr>
<td></td>
<td>6.</td>
<td>Subset ALGOL</td>
</tr>
<tr>
<td></td>
<td>A.</td>
<td>Miscellaneous Business</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ISO</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ACM Communications</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ALGOL Input-output</td>
</tr>
<tr>
<td>17 March</td>
<td>7.</td>
<td>Future ALGOL</td>
</tr>
<tr>
<td>(0945-1645)</td>
<td></td>
<td>Discussion in general</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;Generalized ALGOL&quot; by Professor V. Wijngaarden.</td>
</tr>
<tr>
<td>18 March</td>
<td>6.</td>
<td>Subset ALGOL</td>
</tr>
<tr>
<td>(0930-1730)</td>
<td></td>
<td>Politics</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Review of Draft Report</td>
</tr>
<tr>
<td></td>
<td>7.</td>
<td>Future ALGOL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;Generalized ALGOL&quot; continued</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Wirtz Report.</td>
</tr>
<tr>
<td>DATE</td>
<td>AGENDA ITEM</td>
<td>SUBJECT</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>19 March</td>
<td>6.</td>
<td>Subset ALGOL</td>
</tr>
<tr>
<td>(0945-1800)</td>
<td></td>
<td>Politics (again)</td>
</tr>
<tr>
<td></td>
<td>A.</td>
<td>Review of Draft Report</td>
</tr>
<tr>
<td></td>
<td>7.</td>
<td>Miscellaneous Business</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ISO and WG 2.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Future ALGOL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Wirtz Report concludes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>V. Wijngaarden report concludes</td>
</tr>
<tr>
<td>20 March</td>
<td>7.</td>
<td>Future ALGOL</td>
</tr>
<tr>
<td>(0945-1645)</td>
<td></td>
<td>Dijkstra report</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Banon report</td>
</tr>
<tr>
<td></td>
<td>5.</td>
<td>General Discussion</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ALGOL Input-Output Report</td>
</tr>
<tr>
<td></td>
<td>6.</td>
<td>Subset ALGOL</td>
</tr>
<tr>
<td></td>
<td>9.</td>
<td>A.B. - Policy</td>
</tr>
<tr>
<td></td>
<td>2.</td>
<td>Minutes - Policy</td>
</tr>
<tr>
<td></td>
<td>11.</td>
<td>Next Meeting</td>
</tr>
<tr>
<td></td>
<td>9.</td>
<td>A.B. - Policy</td>
</tr>
<tr>
<td></td>
<td>5.</td>
<td>ALGOL Input-Output, Approval</td>
</tr>
<tr>
<td></td>
<td>7.</td>
<td>Future ALGOL - superstructure</td>
</tr>
<tr>
<td></td>
<td>6.</td>
<td>Subset ALGOL - Approval</td>
</tr>
<tr>
<td></td>
<td>10.</td>
<td>Press Release</td>
</tr>
<tr>
<td></td>
<td>12.</td>
<td>Adjournment</td>
</tr>
</tbody>
</table>

Woodger observed that WG 2.1 meetings should have objectives and not a time schedule. It was also the general concern to devote this meeting primarily to Future ALGOL, with the necessary minimum of time spent on the Subset and Input-Output.

4. MEMBERSHIP

4.1 Although WG 2.1 membership is the responsibility of IFIP/TC 2, the Chair reported receipt of letters of resignation from Green, Vauquois and Wegstein, which he read. This was considered a great loss to WG 2.1, and the Group expressed for the record its appreciation of the contribution each has made to the ALGOL field. The Chair noted that Bolliet of the University of Grenoble has been recommended by Vauquois to TC 2 as a new WG 2.1 member. Particular note was made of the reasons for Wegsteins resignation as reported in his letter, as they pertain to the WG 2.1 Subset work. From a policy standpoint it was also noted that TC 2 is not obliged to replace WG 2.1 resignees. However, WG 2.1 could recommend anyone to TC 2 and in particular should consider nominees from the list of current and past meeting observers.

4.2 It was reported that Mr. Moriguti would be absent for the third time, and for the second meeting in a row he was delegating his position to a countryman, currently, Mr. Kenzo Inoue. The Chair emphasized as WG 2.1 policy that only in exceptional cases (and travel from Japan is so recognized) would substitution by and for a member be permitted, and that such could not become a permanent arrangement.
5. **ALGOL 60 INPUT-OUTPUT PROCEDURES**

5.0 Summation.
The opening session on this troublesome subject quickly reviewed the primitive but pure ALGOL 60 nature of the Delft preliminary I-O report, and subsequent receipt of proposals prepared by the ACM ALGOL (Knuth) Committee and by the ECMA/TC 5-ALGOL group. Then in summary, WG 2.1 recognized the Delft primitives as insufficient in reality, though meeting the necessary requirement of within ALGOL 60. A narrow majority favored some expansion on Delft, though a persistent minority called for no further WG 2.1 action at all on I-O for ALGOL 60. It was agreed as necessary to at least do a cleaned-up version of I-O at the Delft level with addition of string capability, for presentation to TC 2. Unanswered were questions of WG 2.1's continuing role with respect to I-O for ALGOL 60, or with the ACM and ECMA proposals, although a statement was approved recognizing the ACM Report and its list of references as exemplifying what can be built from the primitive procedures of ALGOL 60, augmented by the I-O primitives in/outsymbol, in/outreal, in/outarray and stringlength. These additional primitives were approved, and a final I-O Report based on the Delft preliminary report with these additional decisions was prepared and approved for submission to IFIP/TC 2 and the Council as an official IFIP Report, and to ISO as an IFIP proposal.

(Monday) 5.1 Proceedings.
The Chairman opened I-O considerations Monday with a quick review of the Delft Report, and announcement of the recent receipt of I-O reports from the ACM and ECMA ALGOL Committees. The agreement of Delft was reiterated on a basic I-O procedure plus some examples as an adequate WG 2.1 Report, and considerations began with the question of adding detailed examples. This immediately provoked a question from a significant minority of WG 2.1 (which persisted throughout the I-O considerations) of dropping further work on I-O for ALGOL 60 altogether in favor of doing a comprehensive job under Future ALGOL.

5.2 Typifying this thinking, Irons proposed that if I-O must be defined for ALGOL 60 it should be done properly in A60 syntax. He preferred, however, that I-O definition should be left as in the Delft report, and then defined completely in Future ALGOL.

5.3 A majority, however, wanted sufficiency now in A60 I-O (i.e., more than the Delft primitives) to enable realization of practical objectives, such as partial comparability with FORTRAN as an international programming language, test capability for published Algorithms, etc.

5.4 As a consequence, several questions confronted the group: (1) should WG 2.1 examine the ACM and ECMA proposals for their adherence to A60?; (2) should WG 2.1 do a sufficient I-O definition for ALGOL 60, or leave such to other activities? In particular, what should WG 2.1 do with its Delft report? With the ACM and ECMA reports? With the ISO request for I-O? With
TC 2 and Council? These questions went mainly unanswered in the first I-O session, although a point was reached where an apparent consensus of WG 2.1 agreed to do more ALGOL 60 I-O work based on a slight expansion of the Delft primitives to include general string handling procedure. The group vociferously rejected endorsement of the ACM proposal until its adherence to A60 could be thoroughly examined, and there was no interest evidenced in conducting such a project in WG 2.1 in real time.

5.5 Regarding the addition of string handling capability, it was voted (10:2:6) to include Naur's proposal of a string length procedure, with suitable explanatory text, to the Delft I-O report. The insymbol and outsymbol of Delft were considered the basic necessities, and after the following inconclusive votes on inreal and outreal --

6:6:7 to include inreal and outreal with the device number, as it stands from Delft;
4:5:8 to include inreal and outreal without the device number; and
6:7:6 to drop inreal and outreal altogether --

it was voted (13:2:4) to include inreal and outreal with the device number of Delft replaced by an abstract channel designation, and with the accompanying explanation that device designation is not necessarily implied. It was also recognized that the minimum I-O capability required is the ability to put out what has been put in, though, to be really useful, it requires ability to input-output strings.

5.6 The Chairman closed Monday's I-O discussion session without taking a vote on the new A60 I-O report content as agreed in parts as above. Instead, in recognition of the general need for such to fit an obvious void, a subcommittee of Duncan, Rutishauser and v. d. Poel was appointed to redraft the Delft report along these lines, with a suitable explanatory preamble as in Naur's new introduction to the subset, for final consideration at a later session after the benefit of further private discussions.

5.7 Thursday. In an effort to finalize an acceptable list of primitives, Rutishauser proposed "inarray-outarray" be considered. This was approved (10:3:4). This led to consideration of "outstring" and string "length" being replaced by "outsymbol" - i.e., semi-primitives replaced by primitives. Duncan suggested need for a procedure "outlabel", also a primitive. It was voted (7:4:6) to include "length" rather than "outstring". Therefore, it was concluded that WG 2.1 accepts an I-O report that includes basically the preliminary I-O Report of Delft plus "length" plus "inarray-outarray", plus
examples of the use of those primitive procedures to build-up more detailed or specialized I-O procedures that anyone may want.

5.8 It was then decided to describe these decisions and give recognition to other ALGOL I-O developments and proposals by inclusion in the final WG 2.1 I-O Report of a statement (attributed to van Wijngaarden and Rutishauser) such as follows:

"Some procedures of input-output are considered as primitives. Among these are in/out-symbol, in/out-real, in/out-array and string length. Of course, there are many others; for example, outlabel, etc. However, apart from these primitive procedures one meets in practice a fuller set of I-O procedures. The language ALGOL 60 is so flexible that different schemes of I-O procedures can be defined in it, largely by means of the primitives above. Therefore, WG 2.1 does not propose a specific set of I-O procedures, but wants to draw attention to the 1964 proposal of the ACM ALGOL Committee (Knuth, Chairman), which contains in addition an extensive list of references."

This was accepted and the Chairman named an ad hoc editing committee of Duncan, Rutishauser, and v. d. Poel and van Wijngaarden to draft a final I-O Report, for approval on the last day, based on these decisions.

5.9 Friday. The Chair read the ad hoc committees' draft I-O report aloud, and minor corrections were noted. No significant changes were proposed. It was then approved (11:1:4) for submission to TC 2 and the IFIP Council as a proposed IFIP Report, with subsequent submission to ISO as a proposed standard of ALGOL I-O. The Chair noted that it was agreed this revised I-O Report could be sent immediately to ISO/TC 97/SC 5 as is as an IFIP Working Paper, with the statement that it yet requires Council approval to become an official IFIP Report.

6. IFIP SUBSET ALGOL 60 REPORT

6.0 Summation.
The Delft '63 Preliminary Subset Report was distributed in September 1963 to the Delft meeting attenders, for comment and revision at this meeting. Comments received by mail were considered first, and individual changes to the subset were adopted as appropriate, including an almost complete rewrite of the Introduction. A drafting committee was appointed to prepare a final subset report for review and approval on the last meeting day. Late comments of Yershov were considered by the ad hoc drafting committee, and annotated on the final draft distributed just prior to the adjournment. In the last hours the basic "van Wijngaarden Draft" was accepted provisionally (9:5:5) with the understanding it would be final edited, printed, and distributed to WG 2.1 for study and editorial comment within 2 weeks of receipt. The approved final printed report will then be submitted to TC 2 and the IFIP Council in May, 1964, for their approval as an IFIP Report, and to ISO as a proposed standard. It is to be entitled "Subset ALGOL 60 (IFIP)".

6.1 Proceedings.
In particular, the following comments on the Delft Subset Report (1963) were first considered, with the resultant actions as noted:

(1) Rutishauser letters of 25 September 1963 and 14 February 64:

4.1.3.2 - Voted 11:0:5 to change in part to read "this actual parameter being an identifier or string, otherwise the name replacement is undefined."
It was noted for the record that the intent was to put in strings for name parameters, but not constants. The Chair instructed the drafting committee to add a suitable explanation column statement consistent with this intent.

(2) van Wijngaarden letter of 30 September 1963

Introduction - Agreed without objection to strike the words in the third paragraph "...the more precisely defined...". (It was later agreed to reunite almost completely the entire Introduction).


4.7.3.2. - His point (1-a) was covered above.
5.4.4. - His point (1-b) resulted in a vote of 10:1:6 to strike the last sentence entirely, with the note for the record that this again permits Dijkstra's ambiguity function.

The Chairman declared he would answer Bumgarner's "perplexities" in a separate letter.


5.4.4. - It was voted 10:0:7 to add the following sentence to the explanation column -

"A function designator must be such that the execution of its procedure body does not give rise to any change of the values of variables which are nonlocal to the procedure body."

It was also agreed that the Introduction would contain a warning that explanatory (right column) remarks are for exposition only, and have no formal status.

(5) Woodger Comments

5.0 Agreed without vote to change to read "Delete first two sentences ..."

2.1 - Voted 9:0:8 to (2) add a stroke in front of the A, making it /A B C..., to (2) change the first sentence of 2.1 to read "this alphabet may arbitrarily be restricted."

3.3.4 - Decided to leave as is, except to change the "of" to "or" in the second sentence.

3.5.1. - Voted 6:0:11 to change the word "sentences" to "formulae", and to change the explanation to -- "In the subset only unconditional and unparenthesized designational expressions are provided for."

4.3.5 - Voted 7:0:8 to change left column only to read simply - "4.3.5 Delete!"

(6) Naur Comments

Introduction - Criticized the Introduction in general, and proposed new wording with acceptance in general by the group. The Chair asked the drafting committee to revise it to at least include the warning on unofficiality of the Explanations, and to include the last paragraph of the old Introduction.

The title was reaffirmed as "SUBSET ALGOL 60 (IFIP)", and it was acknowledged that it is a proper subset of ECMALGOL, and in turn of the IFIP Revised ALGOL 60.

(7) Merner Comment.

5.4.5. - His proposed change, to read "Complete specification..." was rejected by a vote of 1:3:8, leaving 5.4.5. as it is.
In closing this discussion, the Chair asked Naur, Woodger (and himself) to act as the editorial committee to draft the final subset report for presentation in writing on the last day of the meeting.

6.2 Wednesday. On the political side, van Wijngaarden, from his Councilmanic standpoint, asked that WG 2.1 be polled on 3 questions regarding the implications of the proposed subset and its preamble:

1. Does WG 2.1 want an IFIP authorized subset?
2. If so, should it be the ECMALGOL?
3. Or should it be the WG 2.1 proposal?

The Chairman reported that the Council at Gola recognized the importance of obtaining TC 2 and WG 2.1 guidance on the controversial subset questions of need for and implications of an IFIP ALGOL Subset.

6.3 Several objections were immediately raised to asking such questions now after two years of WG 2.1 subset work. It was emphasized that a priority subset project was prominent in WG 2.1's initial Program of Work, as approved by TC 2 and the Council in 1962. The TC 2 Oslo meeting minutes were read in this regard, specifically directing WG 2.1 to give priority to such work. However, it was also pointed out that while WG 2.1 has continuously voted in large majorities for a particular set of technical characteristics of a subset ALGOL, and also for the totality of its specification, the Group has never formally indicated its willingness (or not) to support the purposes of a subset ALGOL. Therefore, to the extent such non-technical expression from WG 2.1 would be useful guidance to the TC 2 and Council, it was decided to conduct such a vote. It was duly noted in the record that Samelson objected to such a vote on a procedural basis, in that he felt the time to have answered these questions was at WG 2.1's first meeting at Munich in 1962. The vote results were as follows:

1. WG 2.1 favors an IFIP ALGOL Subset, 11:3:6.
2. WG 2.1 rejects ECMALGOL as the IFIP Subset, 6:10:4.
3. WG 2.1 favors its own proposal as the IFIP Subset, 10:8:2.

It was also recorded that this vote count included only the 20 WG 2.1 members present, and excluded 9 absentee active members.

6.4 In explaining potential meaning of these votes the Chair recognized that the subset issue has been controversial in WG 2.1 since its beginning (August 1962), and such voting results indicate quantitatively the degree of this situation, for whatever value such has to TC 2 and the IFIP Council.

6.5 Continuing with preparation of the subset draft report, Naur distributed a proposed alternative statement to the Delft Introduction for consideration. It was decided to drop Naur's last paragraph reporting the controversial nature of WG 2.1's support to the proposed subset, and also to drop the list of WG 2.1 membership from the Delft statement on the basis that many members have never attended any meetings or have otherwise been participant in WG 2.1 subset deliberations. It was also agreed (6:3:6) to drop all reference to ECMALGOL in the introduction, though it was recognized for the record that "ECMA had kindly made its subset available to WG 2.1 "at the beginning as a basis for IFIP SUBSET Considerations. With these changes, the ad hoc subset editorial committee was enjoined to draft a new introduction to accompany its revision of the draft subset specifications.

6.6 Friday. A late arrival letter, dated 12 March 1964, from Academician A. A. Dorodnicyn of the U.SSR., enclosing extensive comments and recommendations
of Yershov was introduced at the last minute. This raised the question of whether to consider the tardy comments or not? After rather excited discussion it was reluctantly decided that since matters of substance were raised, as well as of an editorial and typographical nature, the subset ad hoc committee (Naur, Bauer, v. Wijngaarden) should attempt to incorporate such changes as they deemed appropriate into their final "van Wijngaarden" draft report, to be voted on at meetings (and days) end.

6.7 In conclusion, the final, or "van Wijngaarden", draft was distributed just prior to adjournment, and the unincorporated Yershov changes proposed by the ad hoc committee were then read to enable annotation of the document.

6.8 The Chairman then moved to vote on the acceptability of the "van Wijngaarden" draft report, with changes as understood, before it is final edited, printed and circulated. This action was authorized by vote of 9:6:3. However, Woodger, Dijkstra (for himself, and for the now absent Naur) and van Wijngaarden vigorously protested a final vote until a typeset document incorporating all understood changes had been distributed for several weeks of study and debugging. The Chairman, therefore, decided to take a provisional vote now, to be followed by a final edit of the "van Wijngaarden draft", printing, distribution for review and editorial criticism within 7 weeks, enabling subsequent submission to the TC 2 and Council meetings of May 1964 in Prague. This distribution is to include a letter asking WG 2.1 Tutzing participants if they think the final report is completely in accord in content, wording and punctuation as agreed provisionally at Tutzing, with any resultant comments to be received within 2 weeks of receipt.

6.9 On this basis of understanding and conditions, it was then voted 9:5:5 to accept provisionally the subset document as proposed by van Wijngaarden. (A remark of Rutishauser was noted in conclusion - that having an official IFIP ALGOL Subset was for him a matter of success or failure, since he is unable to "push full ALGOL on my Engineers." It is much too confusing to them without a subset. He felt that much of the problem WG 2.1 has encountered in creating this subset stems from just this problem with full ALGOL 60).
7. FUTURE ALGOL (FA)

7.1 General Explanatory Discussion (Tuesday, 0945-1240)
(This is presented in greater detail to provide background for future more definitive discussions).

(Tuesday, A.M.) The Chairman reviewed the Delft minutes pertaining to FA, and several corrections were noted. Professor v. Wijngaarden's Rome 1962 paper "Generalized ALGOL" was distributed. The ALCOR letter of 26 February 1964 was read into the record (See Appendix C), which raised the first discussion question:

Q1 - Should WG 2.1 (1) concentrate on a quick early solution of not such a modern ALGOL and turn out an A65 incorporating many of our ideas? or -
(2) delay our work and make a big stride forward with an A70?

This became the subject of a round of comments, which went essentially as follows:

Bauer - We can do both. We must keep our eyes on what should go into A70 as we work on A65.

Woodger - Suggests content of A65 be discussed via the ALGOL Bulletin for maximum consideration. We don't want to rush with it.

Dijkstra - Wants a more drastic improvement aimed at A70. More rapid improvement will result in fighting on small details with loss of much time.

Naur - Thinks an A65 would be too early. Wants a long post-deliberation period after AX or AY is finalized to remove flaws.

Chairman - Suggests in future discussions calling the immediate or short range ALGOL objective "AX" (previously spoken of as A65), and the longer range language "AY" (previously called A70). This was agreed.

Rutishauser - Supports Naur's proposal of a cooling off period. Parallel development of 2 languages would have to go out to 2 groups, making annual reports to WG 2.1, so that the whole Group only concerns itself with general lines. Development work can only be done in small groups.

Chairman - WG 2.1 in its entirety is responsible for ALGOL development. Can't countenance small groups which would just fight.

Paul - Supports the ALCOR letter, and the proposal of Rutishauser. We first need a good working AX, with multiple precision, etc., for the users. Since AX would be for the users it should take preference over AY.

Samelson - We can't work on two languages simultaneously. We can try to improve A60, or go for a new language AX = improved A60. Then AY would be a completely new job, starting from scratch and forgetting what had been done in A60. Calls for a straw vote. (not held).

Chairman - Could be that AX and AY will be identical.

v. Wijngaarden - Not so pessimistic on simultaneous development of AX and AY. Strings, symbols, etc., on top of A60 could be AX, without in any way restricting A60. Could simultaneously go on with development of AY.

Bauer - We are now talking about 2 vague concepts, AX and AY. It would help to know what v. Wijngaarden thinks on the basis of his Rome paper on "Generalized ALGOL" regarding how quickly such could be completed when modified as suitable?

v. Wijngaarden - Within a year.

Bauer - In doing so, wouldn't it be possible to include such concepts as are suggested by ALCOR, Floyd, Yershov, and thus fulfill these practical requirements on the basis of a new language?
Wirth - It is not so much on the matter of timing that we differ between AX and AY. Rather it is that WG 2.1 is split between the "ALGOL Extenders" (Yershov, etc.) and the "ALGOL Generalizers". They are mutually exclusive in their interests, and this condition supports Rutishauser's proposal of working in 2 groups.

Chairman - Recognizes "ALGOL Generalists" and "ALGOL Extensionalists" in WG 2.1.

Irons - I am personally interested in both.

v. Wijngaarden - Proposes we do not split the church.

Rutishauser - Does not want to see extension go too far, and could see generalization entrusted to another group without having to participate closely himself.

Lucas - (Missed his statement).

Chairman - We must start on AX without the burden of having to base it on A60 entirely.

Bauer - Agree that we needn't keep to A60 if it proves a burden.

Rutishauser - But we must not create a drastic revision.

Bauer - There is so much patchwork in A60, which should be eliminated.

Naur - I would be very worried over the risk involved in a WG 2.1 split-up. On the other hand, we must compromise, and the "Extensionalists" must assure the result is workable, and the "Fundamentalists" or "Generalists" must assure the result is well defined, etc.

Dahlstrand - Agree WG 2.1 must not split-up. If we do, the ALGOL force could become weakened to the point of succumbing to other forces. The weaker part of the ALGOL Group will drop out over a long development period.

Bauer - Agrees it is not necessary to split. But would personally refuse to be called either an "Extensionalists" or "Generalist". Asks for a straw vote to identify "Extensionalists", and "Fundamentalists", and others interested in doing good practical work, etc.

Chairman - Can't vote. The reason for "Generalists" is to enable extension. Really isn't a conflict, so we don't want to stress it.

Dijkstra - Thinks there is a marked difference. Convinced that if have a good algorithm in a language, hope it can be used to extend your machinery. Concerned with how next language will be influenced by current or future machinery. On other hand, can show how existent machines are unable to support such languages at all as, for example, Iverson's. On other hand, can design a language without regard to machine capability, but it would then be impractical. Can't decide to what extent we should have our language desires influenced by practical machinery capability. We could let the language set the machinery design goals, or vice-versa. We must have the form, structure, etc., of a language influenced by the practicality of machinery, but due to their inadequacies we must realize this as detracting from the desirable in language.

Naur - Languages have influenced machine design. The features of machines that we should take into account are those that inimitable or basic. Avoiding unnecessary operations is always a good thing. We should try to derive from our knowledge of machines those features that are basic, and reflect such in our languages - for example, hierarchies of storage, which we will always have. This pyramid of storage is inherent of machines, and it should be reflected in our languages.

Hoare - Languages should come first, and machines designed from them.

Dijkstra - You can't separate one from the other. Can only decide to what extent you should try, and when, or on which features.
Chairman - We also have the other basic question of the Delft meeting of Formality vs Practicality? or defining the next language more rigorously vs defining it along the same lines as the last ALGOL?

Duncan - Can't think that anything so widely used as A60 can be said to be impractical.

Naur - Agrees this is an extremely important point. We should start at an early point with the description exercise and tools in mind and practiced. We must learn and use and practice with these description tools early in AX or AY development, to assure that we can use them confidently and familiarly.

v. Wijngaarden - We must keep the description technique simple or we will end in describing by example. For international purposes it is essential, however, that the description be solid and rigorous. Next time we must avoid all bickering over words, semantics, etc.

Irons - Yes. We should describe the next ALGOL in something very basic and symbolic or primitive to avoid ambiguity and misinterpretation.

Dijkstra - Hopes the description of the next ALGOL will be pragmatic and complete to such an extent and with such rigor that it is clear what each sequence of basic characters means. This is a question to all of WG 2.1 - Do we think that definition with this rigor, i.e., not leaving a borderline between definitely wrong or uniquely right, is possible or not?

Chairman - Question - would you give any input string meaning, or allow undefined?

Dijkstra - No. Would like a complete definition whereby every possible text would enable a definite reaction. "Undefined" would not be permitted in the next ALGOL.

Example (on board):

```
procedure doubtful (b,p): boolean b, procedure p,
begin real x, y, ---
  if b then p(x) else p (x, y)
end
```

Including two calls - doubtful (true, from 1)
  doubtful (false, from 2)

The question is, do we hope that this hope is reasonable or not? That this borderline of doubtfulness is empty or not?

Chairman - This is, of course, our hope.

Bauer - Agrees we should try. However, do not know what is meant by a "pragmatic" approach. Also do not understand what v. Wijngaardens "Generalized ALGOL" approach is, i.e., a language defined in or by itself? Perhaps he could explain this? How do you know that common understanding is ever achieved? You have to stop somewhere in saying what your basic building blocks or primitives are.

Wirth - Regarding Dijkstra's gap of undefinedness, maybe we could agree that the language is undefined on the basis of not understanding the tool of definition.

Chairman - As in v. Wijngaarden's paper, there is no bottom to descriptive purity or rigor. There is only a question of how far down we should go for the degree of rigor we desire.

Bauer - There are two ways we can do this:
(1) Use a meta-language such as BNF, or
(2) Shift the problem to the meta-level, i.e., describe a language in that language.

There are always elements which must be commonly understood to enable common understanding of such as basic arithmetic operations.
Somehow we must exclude the effect of international misunderstanding, as mathematics excludes such, i.e., with symbols and rules.

**v. Wijngaarden** - If our language is to define such notions as "plus", then it must do so in the language itself.

**Hoare** - We seem to be making excursions into meta-mathematics, and if so we could probably be aided by meta-mathematicians, who have avoided going into meta-meta-mathematics. They have methods of describing, such as "well-formed formulae" and a "proof" in very clear language. BNF uses this idea of defining a "well-formed formula". A "computation" is a sequence of "well-formed formulae".

**Bauer** - Consider the Turing machine as a defining language. We have to find more practical means of defining a language than a Turing machine, or symbolic logic, etc.

**Woodger** - Regardless of how we define the language, we must do it.

**Wirth** - We used to describe algorithms in the tedious detail of a Turing machine. Now we seem to be going backward to this level of tedium again. Block structure, etc., are features far above such primitives as "assignment" and represent the level that we should concentrate on.

**Dijkstra** - With respect to the completeness of definition, and the reaction, hopefully unique, there is another way - and it is a question of both a theoretical and practical nature.

(Coffee Break - 25 minutes)

**Rutishauser** - Concerning AY, I have the feeling that nothing short of an axiomatic approach will help us - meaning start with Axiom #1, and then #2, and so on, with the hope that the axioms are complete and consistent.

**Woodger** - This is not appropriate. An axiom sits there and does nothing, whereas an algorithm describes action.

**Naur** - In A60 we made the real operations vague. This may be the way to do it. In Numerical Analysis we try to get useful results out of undefined operations, and by some pure luck we succeed. It is important in programming languages not to expect complete and unique understanding. We mustn't force this complete understanding. We want to keep this property that given a program that when run on one machine, gives an 8, and on another, gives a 7. We have the extremes of symbol manipulation, which can only lead to a correct or wrong result, and of our intent, which should not be so bare, so scant. We should permit a range of interpretation within a range of accuracy of required result.

**Hoare** - Agree completely. Perhaps paradoxically, the characteristics of our language are formed more by the areas which are purposefully left vague or untouched.

**Naur** - (To Bauer) In principle, don't you agree that we can't produce a formal definition?

**Irons** - As an approach let's adopt some basic level of language, say elemental elements, and proceed to describe formally the more complex features of the language in terms of these. The point is that the simple language would be well understood by most people. It may be difficult to describe the more complex procedures by or in such simple elements; for example, description in terms of a push-down list etc.

**Dijkstra** - This is perhaps a good approach, but we would have to use great care in describing complexity in such simple terms to avoid mystification.

**Chairman** - All WG 2.1 has to provide is a basic report, leaving it to others to develop tutorial courses on it.
Bauer - And such explanations should be ready sooner than in the case of A60.

Naur - No. Not such haste. I circulated an ALGOL course 4 months (Nov. 1960) after the A60 Report was out.

Chairman - But most courses (McCracken, etc.) appeared 2-4 years later.

Irons - Proposes a formal definition with a narrative explanation of what formalism means, with no legality associated with the explanation.

v. Wijngaarden - Will we give as a language the language in which we will describe the language, and call it AX?

Dijkstra - The hope is that we will do both. Hope it is possible to devise the general nature of a couple of basic building blocks or primitives, and then say how to build from these primitives the elements we want. We can start by formalizing the general building blocks, from which we can then come to the construction of such sub-blocks as addition, multiplication, procedure call, etc. And then hope that this doesn't remain just an intellectual exercise, in that it can be implemented efficiently, i.e., building blocks which can be implemented with existing or future machine features. I have two fears - i.e., I fear two ways in which our efforts might deviate: (1) we might stick completely to problems of defining a language as an intellectual exercise, and (2) we might forget to pay sufficient attention to the subject matter of the processors we intend to describe.

v. Wijngaarden - Don't we look to AY as the meta-language by which we can describe a class of problems as people may produce?

Paul - We must keep our feet on the practical ground of current machine usage, but also seek theoretical languages as design objectives. WG 2.1 must, however, release as soon as possible a better means for publication and distribution of programs for computers of common features.

Bauer - We should aim at an open language, but with a nucleus of standard building elements, which is such that we can use just the nucleus for simple problems, and yet also use the extended nucleus for more complex or unorthodox problems with unusual machine structure. The nucleus should be oriented to machines, and pedagogically, and programs should be competitive to hand tailored programs.

Naur - Agree somewhat. We should take this into the open, and face the world of machine reality. Languages should not stop those with machines with unique features from using them; and on the other hand, should not force commonness of machine features. Propose a common description feature to enable algorithms for specialized machines, and other algorithms for machines without special features.

Wirth - A very fundamental nucleus would not be very useful, and must have extensions.

v. Wijngaarden - Definition of a language is first a definition of utterly small and elementary machinery, able to swallow any machine and any language - which are ultimately one and the same, i.e. one unique thing. The question is - what should we offer the public now? Should we offer the definition of the completely open language, that can swallow anything, and grow and live, but which is practically unimplementable? Should we sell this concept, and then fill it with a particular filling, resulting in a closed language, to which we can offer meta-statements for modification, and result in the original language? I propose that we define the concept, and then
offer the public a closed language first (calling it the AX or X language) which can be implemented, and then offer the developed concept as a language later, as the AY or Y language.

Woodger - Redenclation? How far should this go? How to implement such? Do not know!

Samelson - We must (not try) to shut people out from the machines they are working with. A60 does not permit users to use machine features. We should try to have a language which people can open up, and add definitions to. It must have the ability to create its own procedures, but not its own notation.

Wirth - We should publish a report on this very basic language, and at the same time a report on what is practical and realizable in it. This basic language would be the direction for years.

Chairman - In summary, it is happy to see so much concensus about WG 2.1. We started in the direction of a split, but now have a consensus. All agree in general that the next language must be pretty much formalized and more rigorously defined, i.e., with less ambiguities, etc. It is agreed that we should have a theoretical objective, and at the same time an immediate realizable nucleus or filling of practical nature.

Chairman - To proceed from this basis of agreement, it is proposed that we next collect what we have to build from, i.e., have a round of exposition, not discussion. We will then see how we can construct from this store of capabilities, and what we can begin to develop at later meetings.

7.2 (Tuesday P.M.) Professor van Wijngaarden - next presented a more detailed exposition of his Rome 1962 paper on "Generalized ALGOL" (1425-1645). He was unable to complete (Wednesday A.M.) his presentation until the next day (1025-1105), and no conclusions were reached or attempted.

7.3 Dr. Wirth - next presented some ideas representing further development from his cleaned up first conception - "A Generalization of ALGOL" (Communications of the ACM, Volume 6, Number 9, of September 1963). These ideas were originally programmed in NELLAC, but are currently described in Iverson's notation. He also recommended study of Leavonworth's Princeton Conference paper "FORTRAN IV as a Syntax Language" (See January 1964 ACM Communications). At the conclusion of his presentation (1140-1240, 1630-1715), the Chairman asked that specifications of his compiler be sent to the Secretary for distribution to WG 2.1, and Wirth replied that such would be done when the specifications are completed in a couple of weeks.

7.4 (Thursday A.M.) Professor Dijkstra - was next invited to describe his work of 6 months prior, which unfortunately was not available yet as a paper. The following abstract was offered, however:

A simple mechanism modelling some features of ALGOL 60.

The purpose of this note is to describe an abstract machine which has been designed primarily for explanatory purposes. As such a machine responds in a unique way to any given input character sequence one may hope that such a machine may serve a useful purpose as a tool for language definition. This does not pretend to provide an adequate tool for the definition of an arbitrary programming language. The presentation offers a clarification of some of the mechanisms essential for the understanding of ALGOL 60. These are evaluation of an
expression, assignment to a variable, the block structure, the possible recursive call of a procedure and the parameter mechanism. It does not deal with labels and arrays.

At the conclusion of Dijkstra's presentation (0945-1200), the Chairman reported his decision to have the Dijkstra, Wirth and v. Wijngaarden papers (and Subset ALGOL 60, and I-O Reports) retyped in the Netherlands and distributed to WG 2.1 as soon thereafter as possible.

7.5 Dr. Barron - was next invited to present his informal remarks indicating very briefly a method of formally defining the semantics of a programming language, which is being developed to describe CPL. The method is based on the work of P. J. Landin (Computer Journal, January 1964), and reference should be made to this paper for further details. It was the thought that this method could be used to describe the structure of any program in ALGOL or CPL. (1210-1240).

7.6 After this round of expository contributions from v. Wijngaarden, Wirth, Dijkstra and Barron, the Group turned to consideration of an intended line of action on these various concepts. Bauer suggested that the most important Future ALGOL subjects - (1) meta-language, and (2) features to be included - should be carefully separated in discussions. He proposed that priority attention be given to discussion of the meta-language needed to enable formulation of WG 2.1 ideas. It was agreed to devote a remaining session to this subject, and a final session to features of Future ALGOL; i.e., preprocessor features, language features, etc.

7.7 (Thursday, P.M.) Meta-ALGOL Discussion (1440-1540).
(Again reverting to abstracted individual remarks to ensure greater accuracy).

Chairman - Calls for general discussion of the four expository papers, and of the direction WG 2.1 should go on meta-ALGOL.

Bauer - Thinks v. Wijngaarden's proposal (Generalized ALGOL) is the most complete, and is excellent. It lacks some details necessary to knowing languages could be produced, but basically it would seem that we could use it to produce almost any language WG 2.1 could want to. Dijkstra's concept was close to that of UNCOL of 6-7 years ago in principle, but much better. Very few concepts are required, without reference to a computer.

Paul - Reflecting a previous luncheon discussion, it is hoped that the priority question of AX and AY has not been dropped yet. The lectures heard fall into two groups:

(1) v. Wijngaarden and Dijkstra - Define a machine as a formal system to handle very large classes of formal systems as yet undefined. They are aimed at AY.
(2) Wirth and Barton - were more in the sense of A58 and A60, and much closer to what we conceive of as AX, i.e., refined and extended A60.

Wirth - It is not surprising that three of the proposals - Wirth, v. Wijngaarden, Dijkstra - do not differ much. The question is now how to put v. Wijngaarden's scheme into reality? The Wirth proposal was an attempt to do this.

Hoare - Noticed a common feature of all 4 talks. They concentrated on describing elegant procedures for such as procedure calls, etc., but neglected a feature we must recommend - i.e., array structure. It is not obvious that it is easy to describe an array structure in any of these 4 schemes.
v. Wijngaarden - It is quite easy in my concept, and in Wirth's. How to embed it in a particular implementation is another question.

Irons - Feel strongly that we must produce a language that is readily usable, and for which "good" compilers can be made. Otherwise we will just have a good paper. There must be ability in the language to describe itself. The scheme of v. Wijngaarden is of this kind. However, it is difficult to envision a way of implementing his scheme such that good compilers can associate with it. Suggests WG 2.1 strive for a less exotic means of describing a language in a language that can describe itself and extensions. It is more valuable to have even a restricted self-defining capability in a language than greater generality.

Naur - In reaction to the several presentations, particularly v. Wijngaarden's, they are very interesting but it is not obvious what they imply. v. Wijngaarden's, for example, is presumably a tool WG 2.1 will work with, and if so it is essential that we all have an intuitive direct understanding of it as a tool. Otherwise we can't use it. A58 was first described one way. Then Backus on his own tried to describe it otherwise, which he presented as a paper at Paris. Others tried to use BNF to describe ALGOL, and it worked out to be a better way. Therefore, v. Wijngaarden or others should try to sell us on the use of such schemes rather than expect us to just pick them up and try to use them, or to adopt any one meta-language. I specifically want to see some part of ALGOL used as an example, and described in the proposed meta-language.

Dijkstra - Once we know what the object is to be described, it may result in a description tool designed for just that subject and very little else. If you wish to describe a certain mechanism, you can in all probability create a machine that does this in a highly unambiguous way. For example, COBOL is highly inappropriate for other purposes.

v. Wijngaarden - Naur's concern that an author should sell his proposal is understandable. I wouldn't want to put too much stress on the importance of a test description of A60, however. Rather it should be our intention to develop a description language based not on some past language description requirements, but on future wants. Do we want a language with possibilities of meta-description? It then becomes a question of how far one wants to go in meta-ing. It is proposed that there be a little programming power of a meta-nature in AX. How far is a very general language practical in giving guidance to compiler construction?

Example - Associative memory developments will make scanning processes better in the future.

Bauer - It would be most helpful to WG 2.1 if v. Wijngaarden could provide some extensive part of A60 described in his language. It would be the best thing v. Wijngaarden could do. I favor the attitude expressed by Paul and Naur, i.e., we need a good description language to realize an AX.

Rutishauser - Proposes the description be of both A60 and RA60 in v. Wijngaarden's language.

Wirth - Cautions that v. Wijngaarden's paper describes more than is understood at first glance.

Chairman - I have formulated a few questions which we must have a clear picture:

Q 1 - Do we all agree that the Future ALGOL must be defined formally and as strictly as possible in a meta-language?
(Yes - unanimously)

Q 2 - To what limit do we go in fixing the substructure, and making available meta-features? Does v. Wijngaarden's scheme really have to be implemented to the bottom? Do we really want all possibilities? (No answers).

Q 3 - How much should be prepared and made standard in the superstructure of a Future ALGOL, although users could change some features? Should we standardize the superstructure? Do we see sharply that there will be a necessity to have a big superstructure? (No answers).

Bauer - Proposes WG 2.1 describe AX by AY, and in that way introduce AY to the public to enable work with it to begin.

Dijkstra - Urges that we see to it that in the ways we can influence (or not) compiler design that we ensure that we can compile at full speed when such is (not) a matter of importance.

Naur - Trying to understand. Would it be possible to stop the use of AX in going to AY? Or is there a simple way of stopping AX from becoming AY by extension of AX?

Chairman - Expresses belief that we have reached a good understanding on the nature of AX.

Dijkstra - Should we make a big point of drawing a clear line between translation and execution? My proposal makes little difference in reading a program in and in its execution, which can be a drawback.

Paul - Users still want fast running programs. The more general your language the more interpretation required. The more specific your program the more generative and fast running your program.

Woodger - If you have a language which can concatenate strings you have just one more thing that denies a program from surveying what it must do.

Bauer - If we can guarantee that we are not forced by a language to repeat certain investigations on and on, we do not care if we can't draw a clear line between interpretation and execution.

Irons - Agrees with Bauer. We must not create a situation which makes it impossible to detect ways of doing something most efficiently. (General agreement here).

Duncan - I am concerned with the transition stage between translation and execution, and we haven't heard any discussion of what can be done therein. If it is true that there is a distinction between translation and execution, then it is not necessarily clear that translation is over once interpretation or execution begins. Translation is something done once, but the interpretation begins and you can now see what must be done it may be appropriate that more one-time translation action occur, i.e., the two should probably go on alternatively.

Seegmüller - Procedure declaration being generated at run time? We must think carefully about such a requirement, which is really asking for compilation at run time.

Paul - This is for AY.

Wirth - Thinks two stages are useful for understanding and pedagogy rather than for efficiency. It is more useful to be able to explain in steps of translation and execution.

Rutishauser - We must assume that in 10 years machines will be arbitrarily silly.

(Conclusion of this discussion)
7.8 (Friday) Features in Future ALGOL (1510-1525)

Chairman - Calls for a brief general discussion on what features should be included in the superstructure of FA?

Hoare - (On the blackboard) -

```
select (expression); (expression list)
else (expression)
a: = select j: a-l, a, a+l
```

Chairman - (on Hoare's departure) Hoare promises to submit a write-up on the above for the ALGOL Bulletin.

(Concluding arguments on the Subset ALGOL 60 Report prevented continuance of this discussion, which therefore ended considerations of Future ALGOL at this meeting; except for the decision to hold a two-day next WG 2.1 meeting around the September 1964 Vienna Conference with one day to be devoted to Future ALGOL features, and one day to meta-ALGOL.)
8. MISCELLANEOUS BUSINESS

8.1 Subset to ISO - The Chairman reported receipt by the IFIP President of a letter dated 19 December 1963 from the ISO/TC97/SC5 requesting release by IFIP of the WG 2.1 Input-output and Subset ALGOL Preliminary (Delft) Reports as working papers for May 1965 consideration by ISO. IFIP President Auerbach's letter of response dated 4 January 64 was also read granting such request only on the subset, and not on the Delft I-O Report. It was noted for the record that such grant should in no way imply IFIP/WG 2.1 favor or support of the purposes and objectives of ISO/TC97/SC5 and its work.

8.2 ACM
The Chair reported recent changes in the editorship and policy of the CACM Algorithm Section, and read his letters of 24 December 1963 and 11 February 1964 to George Forsythe (as new section Editor) suggesting changes to the ground rules for contributions as published in the March 1964 issue to more accurately reflect their objectives of compliance with ALGOL 60. No further action was taken.

9. NEXT WG 2.1 MEETING

9.1 Princeton May 1965 - Irons and Utman invited a WG 2.1 meeting to Princeton, New Jersey, USA, for 17 - 21 May 1965, just prior to the IFIP Congress 1965 in New York. This was unanimously agreed.

9.2 Vienna September 1964 - Since this Princeton meeting would be over a year away, and WG 2.1 needing to progress Future ALGOL development by a consideration of features (missed at this meeting) and meta-ALGOL, and since the IFIP Council has been critical of WG 2.1 lack of activity, there was rather reluctant (6:6:6) approval to consider a minimal interim meeting around the IFIP/TC2 Working Conference Vienna 1964 on "Formal Language Description Languages", certainly an appropriate and related subject area. To assure all WG 2.1 members are also invited to the Working Conference the following action was approved:

"WG 2.1 shall ask Dr. Zemanek whether he will be so kind as to invite those members of WG 2.1 who are not yet invited as members of the Working Conference Vienna as guests to this conference. The reason for this decision is that WG 2.1 realizes that many of its members are present in Vienna anyhow and should like to have a short meeting of its members without causing too much trouble to them."

With this understanding, the following plan was approved (11:1:6):

"Under the assumption that Dr. Zemanek does invite as guest the members of WG 2.1 in Vienna (as above), WG 2.1 shall organize a short meeting in Vienna on 14 and 19 September 1964, with one main point of agenda, i.e., Future ALGOL."

10. PRESS RELEASE
The following text was approved without objections as a formal press release on this meeting:
1. **ALGOL BULLETIN**

The ALGOL BULLETIN will resume under the auspices of IFIP/WG 2.1 and editorship of Mr. Frazer Duncan. Policy will be outlined in the initial issues. Contributions and inquiries should be mailed to:

Mr. Frazer Duncan  
ALGOL Bulletin Editor  
Lubeckstraat 71  
The Hague, Holland

2. **SUBSET ALGOL 60**

IFIP/WG 2.1-ALGOL, at its meeting at Tutzing, Germany, 16-20 March 64 completed specification of SUBSET ALGOL 60 (IFIP). It will be proposed for adoption by IFIP as the official proper subset of the IFIP ALGOL language under the name of SUBSET ALGOL 60 (IFIP).

3. **INPUT-OUTPUT**

A "Report on ALGOL Input-Output Procedures" was also completed for proposed adoption as an IFIP Report. A basic set of input-output procedures is defined. It is further recognized that in practice a fuller set of procedures is needed. The flexibility of ALGOL 60, however, enables definition of different schemes by means of the basic procedures, as is exemplified in the report.

4. **FUTURE ALGOL** - Groundwork for future ALGOL begun at Delft in 1962 was augmented by a further exchange of ideas. Concepts will now be proposed and developed through the medium of the ALGOL BULLETIN, for consideration at future WG 2.1 meetings.

5. **ADJOURNMENT** (at 1645 Friday 20 March 1964)

At the conclusion of the last and stormy session on Subset ALGOL 60, the Chairman thanked Professor Bauer and his Institute as the hosts for this meeting, and adjourned the meeting until the next at Vienna in September 1964.

Respectfully submitted,

R. E. Utman, Secretary  
IFIP/WG 2.1 - ALGOL

Appendices:  
(A) Draft Agenda for Tutzing meeting.  
(B) ALGOL Bulletin policy of Duncan, dated 20 March 1964  
(C) ALCOR Letter of 20 FEB. 1964 on Future ALGOL.
DRAFT AGENDUM (Proposed)

1. Chairman introductory remarks and report.
2. Approval of Delft 1963 meeting minutes.
3. Approval of draft agenda.
4. Membership and administrative business.
5. Review and approval of ALGOL 60 Input-Output Procedures Report.
6. Review and approval of IFIP Subset ALGOL 60 Report.
8. Reports to IFIP/TC2 and Council meetings of May 1964 in Prague.
11. Next meeting plans.
12. Closing remarks and adjournment.

R. E. Utman
WG 2.1 Secretary
F. Duncan's statement of 20 March 1964:

New ALGOL Bulletin

1. Aims and Intentions

(a) To enable those interested in ALGOL to be informed of news, opinions, and activities in the field as quickly as possible.
(b) To provide a forum for discussion on ALGOL development between meetings of the IFIP WG 2.1.
(c) To publish informally papers of concern to the ALGOL world.
(d) To provide information to readers in such a way that it can be easily assimilated, and any items of more than transient interest detached and filed.

2. Form

2.1 The Bulletin to appear in sections, each section (and in the articles section, each article) to begin on a new sheet of paper.

Sections as follows:
0. Contents list.
A. News, official statements, etc.
B. Correspondence.
C. References, etc., to new publications.
D. Articles and papers.

This list may be extended as need arises. In the contents list will be shown the number of sheets in each section (which may, of course, be sometimes zero), and the titles of major contributions. With this form it will be possible to discard all except items of quasi-permanent value, and to file the latter as appropriate.

2.2 Official statements from bodies such as IFIP, the ALCOR-group, ECMA, ASA, ISO, etc., will be printed verbatim. Other items of news (new compilers, changes of address, forthcoming events) will be edited and the editor will be responsible for their accuracy.

2.3 Correspondence is encouraged, and letters will, as far as possible, be reproduced fully (except for any purely personal paragraphs). The editor reserves the right to correct any mis-statement of fact by means of a footnote, though not interfering with the correspondent's text.
2.4 The first issue (#16) of the now A.B. will give references to books, papers, etc., on ALGOL topics published since the beginning of this year. Subsequent issues will cover the period since the last issue. Items for inclusion are welcomed, and omissions should be notified. The intention is to ensure that all workers in the field are informed of their colleagues' publications as soon as possible.

2.5 Under "articles and papers" it is intended to publish particularly contributions towards the development of the next ALGOL. In the past inspiration has come in the period immediately after a meeting of the working group, and has faded gradually until a week or two before the next meeting. It is hoped to "strike while the iron is still hot". Other subjects of interest are accounts of input-output systems for particular groups new techniques with the existing language, new compiling methods, and so on. In general, papers will be published as soon as possible. Again, the editor will use footnotes to draw attention to errors of fact.

3. Frequency of Publication

It is felt that it is better to publish at fixed intervals rather than only when there is a large quantity of material on hand. In this way publication of important material is not delayed, and readers can be sure that they are up to date. The minimum issue is a contents page, giving title, number and section headings, each followed by a new. A reasonable interval between issues is 2-3 months.

4. Editorial Policy

The editor's intention is to represent views of correspondents and contributions as accurately as possible, and his attention should be drawn to his failings in this respect. He can, of course, enter discussions in his private capacity; his contributions here will be clearly marked, and will be quite separate from any editorial matter.

5. Language

Since the working language of the ALGOL-Group is English, contributions are requested to be in English wherever possible. However, contributions in other languages will be reproduced if (a) their authors need to express themselves in non-English, and (b) they are submitted in typescript.

F. G. Duncan

Tutzing, March 20, 1964
Munich February 26, 1964

Dear Mr. Utman,

The ALCOR group had a meeting on 12/13 February, 1964 in Munich and decided to bring the following request to the IFIP Working Group 2.1:

Irrespective of the possibility of future developments of ALGOL the ALCOR group proposes to the IFIP Working Group 2.1 for the time being to aim at a moderate and early solution on the basis of ALGOL 60 providing some of the most necessary improvements and corrections.

Particularly improvements should be made on the following subjects:

1) The type declaration STRING should be added, and the IFIP working group should give suitable string operations as standard operations.

2) The type declarations COMPLEX and INTEGER COMPLEX should be added.
3) It should be considered how to extend ALCOL by suitable matrix operations.

4) It should be considered how to introduce lists of quantities in ALGOL with suitable list operations.

5) The IFIP working group 2.1 should introduce a better name-value concept similar to the CPL proposal.

6) It is desirable to have a suitable notation in order to express and to handle bound variables as such.

7) It should be possible to handle multiple precision in ALGOL.

8) One should think of introducing a simpler concept of the for-statement for the average common user of ALGOL in addition to the general concept of the for-statement.

Yours sincerely,

(Dr. Hans Langmaack)

Secretary of the ALCOR group
ISO/TC 97/SC 5 RESOLUTIONS

ISO/TC 97/SC 5 having heard the reports of the ad hoc working groups and WG A-Survey which were presented at its meeting of 28 May 1964 in New York approves these reports and adopts the following resolutions:

1. TC 97/SC 5 resolves that the secretariat shall use document ISO/TC 97/SC 5(N.Y.'64-9)78 "criteria for standardization of a programming language" as a guide for any possible standards that the secretariat may be asked to draft.

2. TC 97/SC 5 resolves that the secretariat shall transmit document "ISO/TC 97/SC 5(N.Y.'64-9)78" to the members for comments, and requests that these comments be returned in time for the next meeting of SC 5.

3. TC 97/SC 5 resolves that at the next meeting of SC 5 a similar ad hoc working group shall be formed to consider any comments that have been received as a result of the previous resolution, and such a group shall prepare a final document for SC 5 action.

4. TC 97/SC 5 resolves that its secretariat will cooperate as requested with the secretariat of TC 97/SC 8 in establishing liaison and in exchanging appropriate documents on the subject of programming languages for numerically controlled machine tools, and further resolves to accept the offer of the USA member body to carry on this liaison and to formulate a scope and program of work for TC 97/SC 5 in this matter.

5. TC 97/SC 5 resolves that documents 97/SC 5 N 45, 64 and 65 be transmitted to the CODASYL COBOL Committee.

6. TC 97/SC 5 resolves that ISO/TC 12-Quantities, Symbols, Units, Conversion Factors and Conversion Tables be informed of conflicting usage of the decimal point and comma to indicate the radix point in numbers, and recommends that the ISO Draft Proposal document ISO/TC 12(Secretariat-177)500 be altered to include both the decimal point and the comma in English and programming languages.

7. TC 97/SC 5 resolves that an ISO Draft Proposal on ALGOL shall be prepared based on document 97/WG-E(IFIP-1)89 as the definition.

8. TC 97/SC 5 resolves that in the Draft Proposal a unique subset of ALGOL shall be recognized.

*This document is the IFIP ALGOL 60 Report. Verbatim.
9. TC 97/SC 5 resolves that that subset shall be based on the one defined in document 97/SC 5(IFIP-5)52.1 in conjunction with document 97/WG E(IFIP-1)89.

10. TC 97/SC 5 resolves that the Draft Proposal shall include two levels of I/O procedures:

a. One elementary level based on document 97/SC 5(IFIP-6)53.1.

b. One level which takes into account documents 97/SC 5(IFIP-6)53.1 and 97/SC 5(USA-9)40.1.

Additional procedure to those specified in document 97/SC 5 N 53.1 will be necessary at the elementary level, in order that the higher level may be expressed in terms of the elementary level.

11. TC 97/SC 5 resolves that document 97/SC 5 N 14, as modified to include integer division and string quotes, shall become part of an appendix to the draft proposal on ALGOL.

12. TC 97/SC 5 resolves that the first draft proposal for ALGOL shall provide, in an appendix a transliteration table from the ALGOL reference characters to the 6 and 7 bit code for information interchange (ISO/TC 97/SC 2(N.Y.'64-13)118, May 1964) and at least one representation of these reference characters in media shall also be provided as an example in an appendix.

This procedure shall also be followed as a general principle for other programming languages.

13. TC 97/SC 5 resolves that the following title and scope shall be provided for the First Draft Proposal:

TITLE: ISO Draft Proposal on the algorithmic language ALGOL.

SCOPE: Standardization of a language and a subset thereof suitable for expressing a large class of numerical processes in a form sufficiently concise for direct automatic translation into the language of programmed automatic computers.

* This document (52.1) is the IFIP subset ALGOL 60 Report.
* This document (53.1) is the IFIP ALGOL I-O Report.
* This document (40.1) is the USA ACM ALGOL I-O Report.
1 June 1964
R. E. Utman

IFIP/WG 2.1 - ALGOL

Membership Mailing List

Members

Bauer, F. L., Prof. Dr.
Technische Hochschule
Arcistrasse 21
Muenchen 2, Germany

Bollet, L.
Institute De Mathematiques Appliques
Universite de Grenoble
B. P. No. 7
St. Martin D'Heres, Isere, France

Dahlstrand, I.
Facit Electronics AB
Box 13072
Goteborg 13, Sweden

Dijkstra, E. W., Prof.
Dept. Of Mathematics
Technological University
Insulindelaan 2
Eindhoven, Netherlands

Duncan, F. G.
N. V. Electrologica
Post Bus 207
The Hague, Netherlands

Ekman, T.
Numerical Analysis
Tekniska Hogskolan i Lund
Solvegatan 14
Lund, Sweden

Floyd, R. W.
Computer Associates, Inc.
Lakeside Office Park
Wakefield, Massachusetts 01880
U. S. A.

Garwick, Jan V.
Norwegian Defense Research
Establishment
P. O. 25
Kjeller, Norway

Grau, A.
Oak Ridge National Laboratory
Box X
Oak Ridge, Tennessee
U. S. A.

Hoare, C. A. R.
Scientific Computing Division
Elliot Brothers (London) Ltd.
Elstree Way
Borehamwood (Hertfordshire)
England

Ingerman, P. Z.
R. D. 32, Box 7
Millersville, Maryland 21108
U. S. A.

Irons, E. T.
Institute for Defense Analysis
100 Prospect Avenue
Princeton, New Jersey
U. S. A.

Katz, Charley
G. E. Computer Dept.
13430 N. Black Canyon Highway
Phoenix, Arizona U. S. A.

Lucas, P.
IBM Laboratory Vienna
Wien 1
Parkring 10, Austria

Mazurkiewicz, Antoni
Instytut Maszyn Matematycznych
Koszykowa 79
Warszawa 1, Poland

McCarthy, J.
Computation Center
Stanford University
Palo Alto, California
U. S. A.
Merner, J. N.
Product Planning
Burroughs Corporation
460 North Sierra Madre Villa
Pasadena, California
U. S. A.

Moriguti, S. Dr. Prof.
Faculty of Engineering
University of Tokyo
Bunkyo-ku
Tokyo, Japan

Naur, F., Dr.
A/S Regnegentralen
Smallegade 2B
Copenhagen F. Denmark

Pacelli, M.
Laboratorio Ricerche Elettroniche
Olivetti Corporation
Borgolombardo
Milano, Italy

Paul, M.
Technische Hochschule
Arcistrasse 21
Muenchen 2, Germany

Poel, W. L. v. d. Dr. Prof.
Laan van Meerdervoort 1715
The Hague, Netherlands

Randall, B.
A. P. D. Computing Office
The English Electric Co. Ltd.
Cambridge Rd.
Wheatstone
North Leicester, England

Rutishauser, H. Dr. Prof.
Swiss Federal Institute of Technology
Zurich, Switzerland

Salle, F.
Compagnie des Machines Bull
94 Avenue Gambetta
Paris 20°, France

Seegmüller, Gerhard
Kommission für Elektronisches Rechnen
der Bayerischen Akademie der
Wissenschaften
Richard Wagner Strasse, 18
Muenchen, Germany

Utman, R. E.
J. C. Penney Co.
Box 200
Princeton, New Jersey
U. S. A.

van Wijngaarden, A. Dr.
Mathematisch Centrum
2 Boerhaavestraat 49
Amsterdam (O), Netherlands

Wirth, Niklaus Dr.
Computer Science Division
Stanford University
Palo Alto, California
U. S. A.

Woodger, M.
National Physical Laboratory
Teddington (Middlesex)
England

Yershov, A. P.
Vychislitelnyi Zentr
Novosibirsk 72
SSSR
Extra Mailing List (7)

Mr. I. L. Auerbach
Auerbach Corporation
1634 Arch Street
Philadelphia 3, Pennsylvania
U. S. A.

Dr. A. P. Speiser
IBM Research Laboratory
Ruschlikon
Zurich, Switzerland

Mr. G. C. Toottill
European Space Technology Centre
Mijnbouwplein 11
Delft
Holland

Prof. Dr. A. van Wijngaarden
Mathematisch Centrum
2 Boerhaavestraat 49
Amsterdam (O), Netherlands

Dr. Heinz Zemanek
IBM Laboratory Vienna
Parkring 10
Wien 1, Austria

Prof. R. A. Buckingham
Computing Center
University of London
London, England

Miss C. M. Popplewell
Computing Machine Laboratory
The University of Manchester
Manchester 13, England
RECURSIVE DEFINITION OF SYNTAX AND SEMANTICS

A. van Wijngaarden
Mathematisch Centrum Amsterdam
RECURSIVE DEFINITION OF SYNTAX AND SEMANTICS

In a former paper with the name "Generalized ALGOL" /1/ a mechanism is described which interprets a text, called program, and delivers another text, called the value of the text so far read. Knowledge of the working of this machine enables the writer of the program not only to describe the process he wants to describe, but also the language he is using. It acts therefore both as language as well as metalanguage. In /1/ more emphasis was laid on showing how a language of the ALGOL-type, but much more general could be defined in this way, using only few definitions. We shall now investigate more in detail some metalinguistical properties, without being interested in the quality of the language to be described.

The machine which interprets the text was considered to consist out of two parts, a preprocessor and a processor. The preprocessor was not formalized and actually varies from case to case. The processor was formalized to a high degree and does not vary.

Let us first turn our attention to the preprocessor. This rewrites the text into an equivalent one in a more restricted language. Indeed, a language may contain many pseudoconcepts, viz. those expressable in other concepts of the language. It is therefore advantageous to split the definition of the language correspondingly into two parts.

By way of example, we take ALGOL 60 and see which are pseudoconcepts in this case.

The first is the comment, which has no semantical meaning at all. Hence, in any occurrence outside strings one may delete certain sequences of basic symbols completely.

Similarly, outside strings the sequence ") (letter string) : ( may be replaced by ' , , '. Or again, outside strings the basic symbol 'array' if not preceded by ' (local or own type)' may be replaced by ' real array '. This might perhaps not seem a simplification, but it
simplifies the description of the language.

Our next victim is the for statement which can be rewritten in its defining sequence of statements. Next comes the function designator and the corresponding type procedure. Replace the type procedure by a non type procedure with one more formal parameter and replace the assignment to the procedure identifier by an assignment to that formal parameter. Replace all primaries in expressions in an assignment statement by auxiliary variables, let the statement be preceded by a sequence of statements assigning the required values to these auxiliary variables and let it be followed by one or more statements assigning the value of the left part auxiliary variable to the actual left part variable or variables. This last precaution is in view of formal parameters. Inside a procedure body a formal parameter is either passed on, or one requires its value, or one wants to assign to it or execute a goto statement leading to it. Replace the actual parameters in a procedure call which also may contain expressions by the identifiers of procedures with two parameters, which depending upon the value of the second parameter assign the value of the first one to the actual parameter (output) or the value of the actual parameter to the first one (input) or execute a goto statement leading to the first one, and so on. In the procedure body the formal parameters are then replaced by the corresponding calls.

Conditional expressions which are not actual parameters can be removed by text splitting, e.g. \( a := \text{if} \ b \ \text{then} \ c \ \text{else} \ d \) is replaced by \( \text{if} \ b \ \text{then} \ a := c \ \text{else} \ a := d \). If they occur in actual parameters, the corresponding formal parameters of which are called by name, this would not work, but this case has been taken care of already by the preceding measures.

By such an intricate but still lexicographical process one does not only eliminate the function designator but actually defines what it means. As a matter of fact the ALGOL 60 Report /2/ contradicts itself on this point.
If one has performed the mentioned reductions of the text, it has a much simpler appearance. Procedure calls no longer appear in expressions and actual parameters, and conditional expressions no longer exist. This enables us to do away with switches, labels and goto statements. The switch declaration is replaced by a corresponding procedure declaration containing goto statements and a goto statement referring to a switch element is replaced by the corresponding procedure call. For instance, switch 5 : = 51, 52, 53 is replaced by

procedure S (n) ; value n ; integer n ; if n = 1 then goto S1 else if n = 2 then goto S2 else go to S3

and correspondingly, 'go to S[i]' is replaced by 'S(i)'. Actually the fact that the switch list is replaced in this way by a statement is the reason why the difficulty of function designators and conditional expressions in a switch declaration was deliberately overlooked above.

Next remove all multiple labels, renaming references to the removed ones. This obviously reduces the number of labels possibly but the following steps do not seem promising. Provide each procedure declaration with an extra formal parameter, specified label, and insert at the end of its body a goto statement leading to that formal parameter. Correspondingly, label the statement following a procedure statement, if not labelled already, and provide that label as the corresponding extra actual parameter. Also, label each statement following a goto statement, if not labelled already, and complete a conditional goto statement which is an if statement by 'else go to L', where L stands for the label immediately following the statement. Label each block, if not labelled already except the outermost one and label the first statement of each block if not labelled already. If a label, which is not the first inside a block is not preceded by a go to statement, then insert a goto statement leading to that label. Enclose each sequence of statements between two successive labels in the brackets begin end if not already enclosed that way. The structure of the program that is now obtained is remarkable. It consists after the insertions corresponding to the performed procedure calls of a sequence of elements, viz. compound statements and blocks, not containing go to statements, except one at each end linking it to another element. It is now completely harmless to insert at the end of each
block an unlabelled go to statement leading to the first statement of that block, since this statement will never be executed. So far, we have only increased the number of labels and go to statements. But now we can perform the following operations:

i) write before each label procedure,

ii) replace the colon following it by a semicolon,

iii) strike each go to.

The program is then again syntactically correct, contains no labels and go to statements and defines exactly the same sequence of operations as before.

This sketch may suffice to show the power of preprocessing. ALGOL 60, reduced in this way, is seen to contain only a few concepts, like

i) some arithmetical and Boolean operations,

ii) assignment,

iii) the procedure without or with parameters, call by value and call by name

iv) locality, own concept.

In /1/ it is shown that with some minor modifications of the language, the concepts under ii) and iii) can be identified and expressed in some simple rules of preprocessing and processing. The concepts under iv) need more care, but concepts like those under i) are simply dealt with by the processor as we shall show.

First, however, we want to be somewhat more specific about the precise role of the preprocessor than in /1/. For a language like ALGOL 60 in which a program is a fixed text it suffices to separate preprocessor and processor completely in so far that the processor can process the preprocessed text without needing the assistance of the preprocessor anymore. However, for the description of languages that enable the generation of pieces of program by the program itself this does not hold. In this case the preprocessor must
continuously stand by to preprocess new pieces of text that have been generated. If the preprocessing can be defined as idempotent then any text, generated or not can always be preprocessed before being processed. If this is not the case, the preprocessed text must be distinguished from not preprocessed text. One might visualize the unpreprocessed text as written in black ink, whereas the preprocessor turns out red text.

We turn now our attention to the processor which by evaluating the preprocessed text produces its value \( \overline{V} \), a dynamically varying text. This text \( \overline{V} \) on the other hand is scanned by the processor recursively for two reasons. Either it wants to determine the value of a piece of text, or it wants to ascertain whether a truth in \( \overline{V} \) is applicable to the question it is concerned with. Apart from some loose remarks concerning locality and so on, \( \overline{V} \) consists of a sequence of truths, separated by commas. Some of them are of a syntactic nature, like

\[
a \text{ in } \langle \text{letter} \rangle,
\]

which states that \( a \) is one of the values that the metalinguistic variable \( \langle \text{letter} \rangle \) may take.

Others are of a semantic nature. They contain the equality sign \( = \) and they may contain the metaoperator \( \text{value} \) which operates on the immediately following metaprimary. Any sequence of symbols for that matter, can be turned into a metaprimary by enclosing it in the metabrackets \( \{ \} \). Examples are :

\[
\text{value} \langle \text{name 1} \rangle = \langle \text{name 2} \rangle,
\]

\[
10 - \langle \text{digit 2} \rangle = 9 \cdot \text{value} \{ \langle \text{digit 2} \rangle - 1 \},
\]

\[
2 + 1 = 3
\]

In the evaluationscan the processor is interested in the semantic truths but in order to know whether one applies, it has to undertake an applicability scan. Both scans are performed backwards, i.e. the truths in \( \overline{V} \) are examined one by one in order, to start with the last one contained in \( \overline{V} \). The applicability scan may assert that a truth is
applicable. This means that the quantity, \( \langle \text{name 1} \rangle \) say, whose value must be determined, or that value itself is identical with the left hand side of the truth after permissible substitutions. The evaluation scan then applies this fact by applying the same substitutions to the right hand side. If the truth is then of the form

\[
\text{value } \langle \text{name 1} \rangle = \langle \text{name 2} \rangle ,
\]

then the required value is simply \( \langle \text{name 2} \rangle \) and the evaluationscan is ended. If the truth is of the form

\[
\langle \text{name 1} \rangle = \langle \text{name 2} \rangle
\]

then the required value is \( \langle \text{name 2} \rangle \) and a new evaluationscan is started to find this value. Since \( \langle \text{name 2} \rangle \) may contain itself the operator value, this may be a complex affair, evaluation necessarily starting from the inside and further from left to right.

In order to find out whether or not a substitution in accordance with the truths in \( \mathcal{T} \) will make the left hand side of a truth identical with an entity under consideration, the applicability scan applies a systematic parsing process to the left hand side until it has success. Which parsing process is used is not relevant here, but it must be defined in order to guarantee unambiguous interpretation. If, for instance, the left hand side of the truth contains \( p \) primaries and the entity under consideration \( g \) primaries, \( g \geq p \), then these \( g \) primaries can be parsed into

\[ g_1, g_2, \ldots, g_p \]

sequences of primaries, \( g_1 + g_2 + \ldots + g_p = g \). The sequence \( g_1, g_2, \ldots, g_p \) can be considered as a number in the base \( g \). Then a simple parsing scheme is to investigate the parsings in increasing magnitude.

The applicability scan then compares each primary under consideration with the corresponding primary in the truth under consideration to see whether they are identical or, if the latter contains a metalinguistic variable, can be made identical by permissible substitution what generates another independent applicability scan.
If all parsings have been tried without success, the next truth is investigated. If \( V \) is exhausted in this way, this is defined to mean non-applicability. The applicability scan, therefore, always yields its answer in a finite number of steps. The evaluation scan also yields a definite answer, since at the bottom of \( V \) we suppose to find

\[
\text{value} \quad \langle \text{name } 1 \rangle = \langle \text{name } 1 \rangle,
\]

which always applies if nothing else has done so.

The descriptive power of our metalanguage can be increased considerably by assuming that the machine understands the logical operators \( \neg \) (not) and \( \rightarrow \) (implies). As an example, we define a row, say, as a sequence of letters, non of which are equal, by

\[
\begin{align*}
\{ \langle \text{letter } 1 \rangle \, \{ \langle \text{row } 1 \rangle \} & \rightarrow \{ \langle \text{letter } 1 \rangle \, \{ \langle \text{row } 1 \rangle \, \langle \text{letter } \rangle \} \}, \\
\langle \text{letter } 1 \rangle \, \{ \langle \text{row } \rangle \, \langle \text{letter } \rangle \} & \rightarrow \{ \langle \text{row } \rangle \, \langle \text{letter } \rangle \, \langle \text{letter } \rangle \}, \\
\langle \text{row } \rangle \, \langle \text{letter } \rangle \, \langle \text{letter } \rangle & \rightarrow \{ \langle \text{row } \rangle \, \langle \text{letter } \rangle \, \langle \text{letter } \rangle \}, \\
\{ \langle \text{letter } 1 \rangle \, \{ \langle \text{row } 1 \rangle \} & \rightarrow \{ \langle \text{row } 1 \rangle \, \langle \text{letter } \rangle \, \langle \text{letter } \rangle \}, \\
\langle \text{letter } \rangle \, \{ \langle \text{row } \rangle \} & \rightarrow \{ \langle \text{row } \rangle \, \langle \text{letter } \rangle \, \langle \text{letter } \rangle \}.
\end{align*}
\]

where in passing an auxiliary operator \( \exists ! \) is defined.

As a more complicated example, we give a partial description of decimal arithmetic, viz. the addition and subtraction of two integers. Since in a completely formalized description of a language the sequences of letters chosen to represent metalinguistic variables may be chosen arbitrarily, we abbreviate 'digit' to 'di', 'unsigned integer' to 'ui' and so on, in order to save space. The definition is very slightly redundant to increase efficiency and cleaner output:
0 in <di>, 1 in <di>, 2 in <di>, 3 in <di>, 4 in <di>,
5 in <di>, 6 in <di>, 7 in <di>, 8 in <di>, 9 in <di>,

<d1> in <ui>, <ui> <di> in <ui>,
+ in <pm>, - in <pm>, <pm> in <in>, <ui> in <in>,
0 in <xe>, <xe> 0 in <xe>, <xe> in <ui>,

+ <ui1> <pm1> <ui2> = <ui1> <pm1> <ui2>,
- <ui1> + <ui2> = <ui2> - <ui1>,
- <ui1> - <ui2> = - Value {<ui1> + <ui2>},
<ui1> + - <ui2> = <ui1> - <ui2>,

<u1> <di1> <pm1> <ui2> <di2> = Value {<ui1> <pm1> <ui2> 0}
+ Value {<di1> <pm1> <di2>},
<ui1> <di1> <pm1> <di2> = <ui1> 0 + Value {<di1> <pm1> <di2>},
<di1> <pm1> <ui2> <di2> = <pm1> <ui2> 0 + Value {<di1> <pm1> <di2>},

<ui1> 0 + <di2> = <ui1> <di2>,
<di1> + <ui2> 0 = <ui2> <di1>,

<u1> 0 - <di2> = Value {<ui1> - 1 + 0 + Value {<ui1> - 1 - <di2>}},
0 - <di2> = 9 - Value {<di2> - 1},

<di1> <pm1> <di2> = Value {<di1> <pm1> 1} <pm1> Value {<di2> - 1},

<ui1> <pm1> <xe> = <in1> 0 <xe> + <ui1> = <ui1> 0 <xe> - <ui1> = - <ui1>,
<xe> <pm1> <xe> = 0,

0+1=1, 1+1=2, 2+1=3, 3+1=4, 4+1=5,
5+1=6, 6+1=7, 7+1=8, 8+1=9, 9+1=10,

{<di1> + 1 = <di2>} => {<di2> - 1 = <di1>}. 

References

/1/ VAN WIJNGAARDEN, A. : Generalized ALGOL ;
Symbolic languages in data processing.

ALGOL 6X SUBSET

J. McCarthy

ALGOL 6X represents a major advance on ALGOL 60. For example, it provides features for quantities of mixed precision, hexagonal arrays, and functions whose number and domain of definition of variables depends on the phases of the moon. None of these features are included in FORTRAN VII even though FORTRAN VII was finalized a year after the ALGOL 6X meeting in January 1967, held in Tangiers. The practicality of ALGOL 6X has been demonstrated by the University of Wester Tasmania's pioneering ALGOL 6X dual compiler-interpreter known as (CI)² operating on the IBM 1620X including 37 tape units and cryogenic associative memory (it includes almost all of ALGOL 6X except for the controversial application of l'Hopital's rule in the symbolic differentiation of Boolean procedures).

IFIP/TC2 has praised IFIP/WG 2.1 for its excellent reports leading to the discovery that ALGOL 6X almost falls into the category of a Ginsburg modification of a Chomsky 4.34 language. Entire confidence has been expressed that ALGOL 6X will meet with wide acceptance.

However, since Burroughs sold its computer division to Digitek, the ALGOL situation in America has become complicated on account of Digitek's commitments to FORTRAN 6.3. The Joint User's Group under the Chairmanship of Howard Aiken, the new head of Sperry Rand's Univac Division, has resolved that programming languages should be user oriented.

Therefore, IFIP/TC2 meeting at the IFIP building in Geneva (formerly the League of Nations), has asked IFIP/WG 2.1 to prepare a subset of ALGOL 6X that can be programmed in less than 7 man years.

J. McCarthy
Technical Assistant for Protocol
IBM Corporate Marketing V.P.
UN Headquarters, New York, USA 1001.