

AB47.3 Working Papers.

AB47.3.1

Errata to "A Modules and Separate Compilation Facility for ALGOL 68".

The following errata were authorized by the ALGOL 68 Sub-Committee of IFIP WG2.1 on May 12 1981. They are to be applied to:

"A Modules and Separate Compilation Facility for ALGOL 68" by C.H.Lindsey and H.J.Boom,

as published in the ALGOL Bulletin AB43.3.2 and also as Mathematisch Centrum Report IW 105/78.

Errors in Formal Definition.

4.9.1.b+1 # DECSEY invoked => DECSEY INKSEY invoked #

{4.9.1.b+3 is ambiguous; the rule may succeed with any TAU1 such that TAU1 TALLY = TAU.}

1.2.3.R # TAU :: MU. => TAU :: MUum. #

3.6.1.h+3,h+5 # {h,i} => {h,i,-} #

Errors in Implementation Methods.

2.5+9 # HOLE => NEST #

{In section 1.1, the grammar (although formally correct) should be brought into line with the corresponding grammar in the Formal Definition}.

1.1+1:1.1+17 # ??? =>

compilation input:

```
prelude packet &
  imposed module interface option &
  joined module interface {for definition modules, if any,
    accessed by this one};
definition module packet &
  imposed module interface option &
  joined module interface {for definition modules, if any,
    accessed by this one} &
  hole interface;
particular program &
  joined module interface {for definition modules, if any,
    accessed by the particular program};
stuffing packet &
  hole interface &
  joined module interface {for definition modules, if any,
    accessed by the stuffing}.
```

source packet:

```
prelude packet {a module-declaration within the standard environment};
definition module packet {a module-declaration within
  a specified hole};
particular program {a stuffing within the standard environment};
stuffing packet {a stuffing within a specified hole}.
```

AB47.3.2 Errata to "ALGOL 68 Transput. part II: An Implementation Model.

The following errata were authorized by the ALGOL 68 Sub-Committee of IFIP WG2.1 on May 12 1981. They are to be applied to:

"ALGOL 68 Transput, Part II: An Implementation Model" by J.C.van Vliet (see AB44.1.1),

as published as Mathematical Centre Tracts 111 and also in J.C.van Vliet's doctoral thesis.

- {Typing error}
p47+7: # intercative => interactive #
- {Improper handling of the logical file end in the primitives 'do newline' and 'do newpage' at page overflow}
p47c)+11, p48d)+30:
At the end of a call of 'do newline' or 'do newpage', the buffer is newly initialized if one is currently writing to the book. The logical end as recorded in the book then still is at the end of the previous line (because of the preceding call of 'write buffer'). In general, this is no problem, since the 'write back'-flag is raised by 'init buffer' if one is writing, so the buffer will eventually be written back, also resulting in the proper updating of the logical file end information. However, the 'write back'-flag is not (and should not be) raised if a page overflow is detected by 'init buffer'. In that case, an immediately following call of a routine like 'close', 'set' or 'reset' will not properly set the logical end (since no call of 'write buffer' results). Therefore, the following change should be made in both routines:

```
# (init buffer OF cover)(f) =>
  (init buffer OF cover)(f);
  IF status OF cover SUGGESTS page end AND
    status OF cover SUGGESTS lfe in current line
  THEN set logical pos(f)
  FI
#
```
- {Improper handling of the logical end in 'do newpage' at file overflow}
This error is similar to the one reported above, and so is the remedy:
p48d)+27:

```
# THEN status =>
  THEN set logical pos(f);
  status
#
```
- {Interchanging lines in 'open'}
p69+12/13:
Since a call of 'set write mood' may lead to a call of 'init buffer', the buffer primitives ought to be made available first. Therefore, these two lines should be interchanged.
- {Improper handling of the physical file end in 'associate'}
p70+25, p72+19:
It must be recorded that for associated files the physical file is also ended if the logical file is (on reading). (Otherwise a bounds error may occur after a change to write mood.) Therefore: