



UvA-DARE (Digital Academic Repository)

New Insights from Old Programs

The Structure of The First ALGOL 60 System

van den Hove d'Ertsenryck, G.M.C.J.T.G.

Publication date

2019

Document Version

Other version

License

Other

[Link to publication](#)

Citation for published version (APA):

van den Hove d'Ertsenryck, G. M. C. J. T. G. (2019). *New Insights from Old Programs: The Structure of The First ALGOL 60 System*. [Thesis, externally prepared, Universiteit van Amsterdam].

General rights

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

ADD, operation, 187, 200.
add idents from pre scan list, subroutine, 148.
add label to pre scan list, macro, 142.
add procedure to pre scan list, macro, 142.
add switch to pre scan list, macro, 142.
add to future refs if ident not yet encountered,
 subroutine, 148.
add to pre scan list, subroutine, 142.
address mask, constant, 146.
address of RFP, variable, 197.
address type, constant, 196.
after subscript, variable, 179.
 AND, operation, 187, 202.
append, subroutine, 150.
append address, subroutine, 156.
append data, macro, 150.
append enter block if necessary, subroutine, 157.
append future, macro, 150.
append leave block, subroutine, 158.
append load address, subroutine, 174.
append load non formal address, subroutine, 174.
append load value, subroutine, 182.
append op, macro, 150.
append procedure call, subroutine, 177.
at begin of expression, variable, 167.
augment pre scan list, subroutine, 141.
block number, variable, 148.
block number mask, constant, 147.
block number position, constant, 147.
bn, variable, 198.
boolean, type, 139.
both integers or convert, subroutine, 199.
build storage function of dynamic array, subroutine,
 205.
build storage function of static array, subroutine, 165.
call explicit procedure, subroutine, 211.
 CEP, operation, 157, 175, 177, 211.
 CEPR, operation, 177, 212.
 CFP, operation, 177, 217.
 CFPR, operation, 177, 217.
consts, variable, 145.
consts append, subroutine, 149.
consts index, variable, 145.
consts ref type, constant, 146.
copy storage function of value array, subroutine, 206.
cur delim, variable, 140.
cur descr, variable, 146.
cur descr index, variable, 147.
cur tok, variable, 140.
cur tok is formal, variable, 147.
cur tok is integer or real, variable, 140.
cur tok is label or switch, variable, 147.
cur tok is procedure, variable, 147.
cur tok is real, variable, 140.
cur tok is set, variable, 140.
current block index, variable, 142.
display, variable, 198.
 DIV, operation, 187, 201.
do generate for local and value arrays, variable, 156.
dynamic arrays counter, variable, 167.
dynamic arrays dimension, variable, 167.
dynamic to static address, subroutine, 202.
end of execution, variable, 198.
end of preliminary scan, variable, 144.
end of translation, variable, 193.
 EQ, operation, 187, 202.
equal, subroutine, 201.
 EQV, operation, 187, 202.
execution types, type, 196.
false, constant, 139.
fill link and create parameter descriptions, subroutine,
 210.
fix future ref, subroutine, 146.
for CEP future ref, variable, 174.
for FOR0 future ref, variable, 175.
for statement future ref, variable, 175.
 FOR0, operation, 176, 219.
 FOR1, operation, 175, 222.
 FOR2, operation, 176, 220.
 FOR3, operation, 175, 220.
 FOR4, operation, 176, 220.
 FOR5, operation, 175, 221.
 FOR6, operation, 175, 221.
 FOR7, operation, 176, 222.
 FOR8, operation, 176, 223.
fp, variable, 198.
future ref type, constant, 146.
future refs, variable, 145.
future refs index, variable, 145.
generate at end of elementary expression, subroutine,
 186.
generate at end of expression, subroutine, 191.
generate at end of for statement, subroutine, 177.
generate at end of statement, subroutine, 192.
generate for local and value arrays, subroutine, 156.
generate for simple local variables, subroutine, 155.
generate operations at operator with priority,
 subroutine, 187.
 GEQ, operation, 187, 202.
get address, macro, 146.
get array size, subroutine, 205.
get block number, macro, 147.
get parameter, macro, 198.
get parameter type, macro, 178.
get ref type, macro, 146.
get type, macro, 147.
greater, subroutine, 201.
grow, subroutine, 139.
 GTA, operation, 184, 218.
has not been encountered bit, constant, 147.
ident stack, variable, 145.
ident stack top, variable, 145.
 IDI, operation, 187, 201.
if integer convert to real, subroutine, 199.
if real convert to integer, subroutine, 199.
 IMP, operation, 187, 202.
in actual parameter list, variable, 179.
in expression, variable, 167.
in for clause, variable, 174.
in integer variable declaration, variable, 166.
in procedure heading, variable, 143.
in subscript, variable, 179.
in switch declaration, variable, 162.
 IND, operation, 183, 208.
integer, type, 139.
integer addition would overflow, subroutine, 200.
integer multiplication would overflow, subroutine, 200.
integer type, constant, 196.
is by value or array bit, constant, 147.
is dynamic ident bit, constant, 147.
is dynamic parameter, constant, 178.
is formal ident bit, constant, 147.
is formal parameter, constant, 178.
is integer, constant, 147.
is label or switch ident bit, constant, 147.
is local array bit, constant, 147.
is procedure ident bit, constant, 147.
is procedure with address result, constant, 147.
is procedure with non address result, constant, 147.
is real, constant, 147.
is static parameter, constant, 178.
 ISF, operation, 168, 205.
 IVA, operation, 156, 206.
 JMPF, operation, 190, 217.
 JMPS, operation, 163, 217.
 JMPU, operation, 157, 160, 162, 174, 175, 176, 177,
 179, 184, 191, 217.
 LAP, operation, 157, 206.
 LEQ, operation, 187, 202.
 LES, operation, 187, 202.
link size, constant, 157.
load, subroutine, 197.
local simple variables counter, variable, 155.
local variables index, variable, 156.

lookup and add to consts, subroutine, 149.
lookup ident, subroutine, 147.
main, subroutine, 224.
main stack, variable, 140.
main stack top, variable, 140.
max value array dimensions, constant, 156.
mem, variable, 196.
MOR, operation, 187, 202.
MUL, operation, 187, 201.
NEG, operation, 187, 199.
NEQ, operation, 187, 202.
NON, operation, 187, 202.
non ref type, constant, 146.
object program, variable, 145.
object program index, variable, 145.
OR, operation, 187, 202.
panic, subroutine, 147.
parameter number, variable, 198.
parameter type mask, constant, 178.
pc, variable, 198.
peek, macro, 141.
peek delim, macro, 150.
peek priority, macro, 187.
pop, macro, 141.
pre scan list, variable, 140.
pre scan list end, variable, 140.
preliminary scan, subroutine, 145.
preliminary scan process begin, subroutine, 143.
preliminary scan process block begin, subroutine, 142.
preliminary scan process declaration, subroutine, 143.
preliminary scan process end or semicolon, subroutine, 144.
priority mask, constant, 150.
priority position, constant, 150.
process array, subroutine, 171.
process array reference, subroutine, 183.
process assign, subroutine, 194.
process begin, subroutine, 192.
process begin of dynamic array variables, subroutine, 167.
process begin of procedure statement with parameters, subroutine, 179.
process colon, subroutine, 194.
process comma, subroutine, 194.
process comparison, subroutine, 188.
process delimiter, subroutine, 195.
process do, subroutine, 176.
process dynamic array variables colon or comma, subroutine, 167.
process dynamic simple variables, subroutine, 169.
process else, subroutine, 191.
process end of dynamic array variables, subroutine, 168.
process end of procedure statement with parameters, subroutine, 182.
process end or semicolon, subroutine, 193.
process for, subroutine, 174.
process for assign, subroutine, 175.
process for comma, subroutine, 176.
process formal parameter list, subroutine, 158.
process go to, subroutine, 172.
process if, subroutine, 190.
process integer or Boolean, subroutine, 170.
process label colon, subroutine, 171.
process label or switch ident, subroutine, 184.
process left bracket, subroutine, 183.
process left parenthesis, subroutine, 194.
process logical operation, subroutine, 188.
process own, subroutine, 171.
process parameter comma, subroutine, 181.
process plus or minus, subroutine, 188.
process procedure, subroutine, 160.
process procedure assign, subroutine, 189.
process procedure or label or switch ident declaration, subroutine, 149.
process procedure statement without parameters, subroutine, 177.
process real, subroutine, 170.
process right bracket, subroutine, 185.
process right parenthesis, subroutine, 195.
process static array variables, subroutine, 166.
process static simple variables, subroutine, 168.
process step, subroutine, 175.
process switch, subroutine, 162.
process switch assign, subroutine, 162.
process switch comma, subroutine, 163.
process switch reference, subroutine, 184.
process switch semicolon, subroutine, 163.
process then, subroutine, 190.
process times or divide, subroutine, 188.
process to the power, subroutine, 187.
process until, subroutine, 175.
process value and specification lists, subroutine, 159.
process variable assign, subroutine, 188.
process while, subroutine, 175.
prune, macro, 141.
push, subroutine, 140.
push cur delim with priority, macro, 150.
push on ident stack, subroutine, 146.
put block number, macro, 147.
put ref type, macro, 146.
read toks until next delimiter, subroutine, 140.
real, type, 139.
real type, constant, 196.
ref type mask, constant, 146.
ref type position, constant, 146.
REP, operation, 158, 213.
RFP, operation, 197, 215.
RIP, operation, 181, 214.
RSF, operation, 168, 205.
RSV, operation, 155, 204.
run, subroutine, 223.
RVA, operation, 156, 206.
SCC, operation, 157, 160, 213.
set, macro, 141.
short circuit, subroutine, 213.
si, variable, 198.
sign, subroutine, 221.
skip read toks, variable, 140.
sp, variable, 198.
SSI, operation, 184, 218.
ST, operation, 159, 187, 203.
STA, operation, 187, 204.
STAP, operation, 187, 212.
START, operation, 196, 198.
static variable ref type, constant, 146.
static variables end address, variable, 197.
static variables index, variable, 166.
STOP, operation, 193, 198.
store procedure result, subroutine, 212.
STP, operation, 187, 212.
SUB, operation, 187, 200.
take address, subroutine, 202.
take formal, subroutine, 216.
take result, subroutine, 203.
TER, operation, 182, 186, 203.
TFA, operation, 174, 217.
TFR, operation, 159, 182, 184, 217.
TIAD, operation, 159, 174, 203.
TIAS, operation, 174, 203.
TRAD, operation, 159, 174, 203.
translate, subroutine, 196.
translation types, type, 146.
TRAS, operation, 174, 203.
true, constant, 139.
TTP, operation, 187, 201.
type mask, constant, 147.
type position, constant, 147.
typed word, type, 139.
unspecified type, constant, 196.
update display, subroutine, 213.
VAP, operation, 157, 207.
word, type, 139.
wp, variable, 198.