Dr. E. Schmid
Rechenzentrum
der TH Stuttgart

Das Rechnen mit dem Algol-Uebersetzer ER 560-6

Dieser Umdruck enthaelt die ueber das ALGOL-Manual der Alcor-Gruppe von R. Baumann hinausgehenden Anregungen und Vorschriften fuer Algol-Programme beim Elektronischen Rechner ER 56.0.

Mit Verwendung der Beschreibung des Algol-Compilers fuer den ER 56 von H. Wettstein, Karlsruhe, zusammengestellt von E. Schmid, Stuttgart.

Juni 1967

Seite fehlt im Original

0. Verlauf einer ALGOL-Rechnung

Der Elektronenrechner ER 56.0 kann - wie jede elektronische Rechenanlage - nur relativ einfache Operationen ausfuehren. Die Anweisung zur Ausfuehrung einer solchen Operation heisst Befehl. Kompliziertere Operationen oder ganze Rechenalgorithmen muessen aus einzelnen Befehlen zusammengesetzt werden.

Daher kann der ER 56.0 ein in ALGOL formuliertes Programm zunaechst nicht ausfuehren. Er muss sich vielmehr mit Hilfe eines (bereits im Interncode des ER 56.0 erstellten) Uebersetzungsprogrammes (ALGOL-Uebersetzer, auch Compiler benannt) aus den in der Programmiersprache ALGOL in Form eines Rechenprogrammes niedergelegten Rechenanweisungen(''Quellenprogramm'') eine Folge von Befehlen bilden (''Objektprogramm''), die auf dem ER 56.0 den Ablauf der gewuenschten Rechnungen bewirkt.

Damit entsteht folgender Ablauf:

1. Schreiben des Programmes in ALGOL und Ablochen (am Fernschreiber)
2. Uebersetzen des ALGOL-Programmes in die Maschinensprache (Interncode) des ER 56.0 (Aus dem ''Quellenprogramm'' wird mit Hilfe des ALGOL-Uebersetzers das ''Objektprogramm'' gebildet)
3. Rechnen mit dem Objektprogramm.

Soll ein Programm spaeter wieder verwendet werden, dann kann man sich das durch die Uebersetzung entstandene Objektprogramm ausgeben lassen (''Stanzen'') und es spaeter wieder in den Rechner eingeben und damit rechnen. Dies ist besonders bei langen Programmen von Vorteil, weil das Eingeben schneller geht als das Neuuebersetzen (siehe 1.3.).

1.1.3. Bereitstellen des Compilers zum Uebersetzen

a) Versuche, ob sich der Compiler aus dem Trommelspeicher bereitstellen laesst.

Vorgehen: Verfahre nach 1.1.2.; dann

Drehschalterstellung 3000 057, F, START
" 5000 059, F, START
" 5000 991, F, START
D, START

Wenn die Maschine nach ca. 8 sec mit Stopp 9000 919 haelt ist der Compiler uebersetzungsbereit. Sonst verfahre nach 1.1.3.b). [2]

b) Einlesen des Compilers vom Lochstreifen: Stelle die Grundstellung der Maschine gemaess 1.1.2. her. Dann lege den Lochstreifen des Compilers ein (Eingabeschalter auf ((↓)).
Drehschalter 5000 067, F, START; Fotoleser liest ca. 1 m Streifen und haelt dann an.
Drehschalter 5000 991, F, START
D, START

Damit wird der Compiler vollends eingelesen und in den Trommelspeicher abgespeichert; (Bloecke 300...599) und gleichzeitig zum Uebersetzen bereitgestellt. Laeuft der Lochstreifen durch, verklemmt er sich oder haelt der Leser an, bevor der Streifen ordnungsgemaess eingelesen ist (START blinkt und evtl. leuchtet eine der Lampen Stoerung), dann druecke sofort STOP, Entsperrt, Grundstellung. Weiter siehe 1.1.4. 1).

1.1.4. Hinweise bei Stoerungen

Das wichtigste Anzeichen dafuer, dass in der Maschine etwas nicht ordnungsgemaess ablaeuft, ist das Auftreten des Blinkens der Taste START; waehrend des Einlesens eines Lochstreifens und waehrend Ausgaben (ueber Stanzer oder Fernschreiber) ist es jedoch zulaessig.

Grundregel: Haelt die Maschine infolge einer Unregelmaessigkeit (z.B. Aufleuchten einer Lampen Stoerung oder ist der Lochstreifen im Leser durchgelaufen oder verklemmt), dann druecke man grundsaetzlich sofort STOP, dann Entsperr. und dann Grundstellung.

---

[2] Nach dem Rechnen mit ALGOL-Programmen oder nach dem Uebersetzen laesst sich der Compiler meist auf folgende (schnellere) Art bereitstellen:
Drehschalter 5000 991, Taste n/n+1 druecken;
Wenn jetzt die 14-stellige Anzeige die Zahl
3004057 9998959
zeigt, druecke F, Start
D, Start
Wenn die Maschine nach ca. 8 sec. mit Stopp 9000 919 haelt, ist der Compiler uebersetzungsbereit, sonst verfahre nach 1.1.3.a).

Einige Beispiele:

1) Stoerung beim Einlesen des Compilers

   a) Fotoleser stoppt, START blinkt, Lampe E leuchtet auf.

      Grund: Lochstreifen nicht in Ordnung (beschaedigt), Loecher schlecht; Transportloch ausgerissen oder verstopft), oder seitenverkehrt eingelegt.
      Abhilfe: Druecke sofort STOP, Entsperr., Grundstellung. Bringe den Streifen in Ordnung bzw. lege ihn richtig ein und versuche gemaess 1.1.3. b) noch-einmal.

   b) Streifen laeuft durch (seitenverkehrt eingelegt oder Eingabeschalter nicht auf ($\frac{5}{2}$)).
      Druecke sofort STOP, Entsperr., Grundstellung, behebe die Stoerungsursache und versuche gemaess 1.1.3. b) nocheinmal.

2) Haeufige Fehler beim Uebersetzen oder Rechnen:

   a) Eingabeschalter steht nicht auf α .
   b) Lochstreifen liegt seitenverkehrt im Leser.
   c) 10 x Zi fehlt am Programmende.
   d) Lochstreifen war schlecht aufgespult und verklemmt beim Einlaufen.
   e) Beim Anlaufen des Fotolesers kein Streifen eingelegt.

   Druecke sofort STOP, Entsperrt, Grundstellung; beginne neu gemaess 1.1.3.a) bzw. 1.4.2.c).

3) Auftreten von Stoerungen KW, RW, T/B, KS

   a) waehrend des Uebersetzens:
      verfahre nach 1.1.2. und beginne neu gemaess 1.1.3.
   b) waehrend des Rechnens:
      verfahre nach 1.1.2. und versuche neu zu starten gemaess 1.4.2.c.

4) Beim Einlesen der Daten laeuft der Streifen durch:

   Es sind zu wenig Daten auf dem Streifen, oder die anzuhaengenden 5 Kommata fehlen, oder der Streifen liegt seitenverkehrt im Leser, oder 2 Daten sind durch Punkt statt Komma getrennt.
   Druecke STOP, Entsperr., Grundstellung; verfahre nach 1.4.2.c).

1.2. Der Arbeitsgang ''Uebersetzen''

1.2.1. Eingabeschalter auf α, Ausgabeschalter auf α_{St} oder α_{FS}

Der Compiler muss gemaess 1.1.3. bereitgestellt sein. Der mit dem Fernschreiber geschriebene Algol-Programmstreifen (Quellenprogramm) [3], der mit 5 ✠ und 10 Ziffernumschaltzeichen schliesst (es duerfen somit nirgends im Programm 10 oder mehr Zi-Zeichen aufeinanderfolgen), wird in den Fotoleser eingelegt. Nach Druecken der Starttaste wird dann das Programm (in Gruppen von 100 Rechner-Woertern) eingelesen und uebersetzt. Die Uebersetzung endet mit

Stopp 9022 919

Das uebersetzte Programm (Objektprogramm) steht ab Speicherzelle 0000.
(Feststellung der Zahl seiner Befehle s. 1.2.4.)

1.2.2. Eingabe von Bibliotheks-Prozeduren

a) Bei Bibliotheks-Prozeduren mit Algoltext ist es empfehlenswert sowohl Prozedurkopf wie -Rumpf in den Lochstreifen einzublenden; eine vorherige Uebersetzung von Algol-Prozeduren mit dem Alcor ist nicht moeglich.

(Man kann das Einkopieren der Bibliotheksprozedur in den Lochstreifen auf folgende Art umgehen: Statt der Prozedur blendet man ca. 1 m Zwischenraumzeichen ein. Wird beim Uebersetzen die Stelle der Prozedur erreicht, dann drueckt man die Taste STOP wenn die ersten Zwr-Zeichen des 1 m-Stueckes in den Fotoleser eingelaufen sind. Er liest dann noch ein wenig, haelt aber innerhalb des 1 m-Stueckes. Dann legt man die Prozedur, an deren Ende ein 1 m langes Stueck mit Zwr angehaengt ist, in den Leser und drueckt START. Wenn die ersten Zwr-Zeichen am Ende der Prozedur einlaufen, drueckt man STOP. Der Leser haelt innerhalb des 1 m-Stueckes und man legt das ALGOL-Programm wieder in den Leser (und zwar so, dass noch Zwr-Zeichen des 1 m-Stueckes gelesen werden und drueckt START.)

b) Code-Prozeduren, also Prozeduren, die im Interncode des ER 56 geschrieben sind, (mit einer bestimmten Parameteruebergabe) [4], werden im Algolprogramm mit dem ueblichen Kopf vereinbart, an dessen Ende als Prozedurrumpf das Wort 'Code' geschrieben wird. Sind solche Prozeduren vereinbart, so tritt im Laufe des Uebersetzungsvorgangs der

Stopp 9011 919, 9012 919... auf;

---

[3] Es empfiehlt sich, den FS-Streifen in mehrere Abschnitte aufzuteilen, die durch mehrere Leerstellen von einander getrennt sind (natuerlich muessen sie aneinander haengen), und immer mit ''Zi'' bzw. ''Bu'' anfangen. Man muss dann bei der Fehlerbeseitigung nur den betr. Abschnitt neu schreiben und zwischenkleben. Zusammensetzen mitten im Streifen fuehrt leicht zu Stoerungen.

[4] Eine Beschreibung der Vorschriften fuer eine Code-Prozedur wird gesondert erscheinen.

dann muss nacheinander der Hauptteil (body) der ersten, zweiten... Codeprozedur in den Fotoleser eingelegt und die Starttaste (weiterhin bei ''Dauer'', Eingabeschalter auf $\vee$) gedrueckt werden.

Wird eine falsche Codeprozedur in den Fotoleser eingelegt, oder steht der Eingabeschalter auf α , so laeuft der Rechner wieder auf den gleichen Stopp bis der richtige Streifen (richtig mit $\vee$) eingelesen ist. War der Eingabeschalter versehentlich auf ($\frac{5}{2}$), dann kann (Eingabe-) Stoerung auftreten. Dann druecke STOP, Entsperr., Grundstellung; lege die Codeprozedur neu ein und stelle den Eingabeschalter auf und starte; ggf. wiederholen. Gelingt die Eingabe trotzdem nicht, dann verfahre nach 1.1.2. und 1.1.3. und beginne die Uebersetzung nocheinmal. Sind alle Code-Prozeduren eingegeben, so geht die Maschine auf

Stopp 9022 919

Das Algolprogramm ist jetzt uebersetzt, falls keine Fehlerangabe ausgestanzt wurde (vgl. 1.2.5.) und steht, relativ adressiert, ab Kernspeicheradresse 0000.

1.2.3. Kapazitaetsueberschreitung

Ausser auf Stopp 9022 919 kann der Rechner auch auf

Stopp 9066 919

laufen; in diesem Fall liegt eine Kapazitaetsueberschreitung vor, es kann nicht weiter uebersetzt werden.

Eine Kapazitaetsueberschreitung (Fehler 60 oder Stopp 9066 919) kann auch davon herruehren, dass am Schluss das Programms ein 'end' oder die 5 Maltheserkreuze fehlen. Dann koennen Fehler fuer Zeilen ausgeschrieben werden, die es im Quellenprogramm gar nicht gibt.

1.2.4. Laenge des uebersetzten Programms (Objektprogramm)

Man schaue sich nach dem Stopp 9022 919 der Uebersetzung den Inhalt der Speicherzelle 0002 an, er ist x 981. Subtrahiert man von x (= vierstellige Zahl) die Zahl 5, so hat man die Adresse des Omega-Doppelwortes am Ende des Programms. (Es besteht aus hoechstens 5000 Befehlen. Die restlichen Kernspeicher (bis 5979) stehen fuer Variable und Felder zur Verfuegung.).

1.2.5. Fehler im Programm

Werden waehrend der Uebersetzung Fehler im Programm entdeckt, so wird noch vor Stopp 9022 919 einer der beiden Saetze ausgegeben:

Fehler <Fehlernummer> in Zeile <Zeilennummer>
Nicht definiert <Name einer Marke>

Ein Verzeichnis der Fehler mit ihren Nummern s. Anhang 1. Alle Zeilen werden gezaehlt, in denen etwas steht, auch wenn es nur ein comment ist; es kommt jedoch vor, dass die Zeilenzahl um 1 weitergeschritten ist, ehe der Fehler erkannt wird. Nach einer Fehlersituation wird die Fehlersuche bis zum naechsten ; |end|else|begin ausgesetzt. Trotzdem wird der groesste Teil der syntaktischen Fehler schon waehrend eines Durchlaufs gefunden.

1.3. Der Arbeitsgang ''Programm stanzen''

Drueckt man nach Stopp 9022 919 der Uebersetzung bei ausgeschaltetem ''Schaltersprung'' und Ausgabeschalter ($\frac{5}{2}$) die Starttaste, so wird das uebersetzte Programm im (2 aus 5) Code bis zum Omega-Doppelwort ausgestanzt.

Wichtig: Das erzeugte Programm wird auch ausgestanzt, wenn dem Stopp 9022 919 eine Fehlermeldung vorausging und es also fehlerhaft ist.

Dieses Programm enthaelt sowohl ein Einlese u. Druckprogramm als auch saemtliche benoetigten Standardfunktionen. Feststellung seiner Laenge: siehe 1.2.4. Es ist relativ adressiert und kann von einer beliebigen Speicherzelle mit geradzahliger Adresse an (Eingabeschalter ($\frac{5}{2}$)) eingelesen werden (mit erneuten Einlesebefehlen an den Teilspeicherenden) und braucht den Algolcompiler nicht mehr. Der Rechner benoetigt zum Ausstanzen eines Kernspeichers (1000 Zellen) ca. 1 Minute (zum spaeteren Einlesen 20 sec). Der Rechner haelt dann an mit

Stopp 9033 919.

(Will man das Programm nicht ausstanzen lassen, so muss man spaetestens nach dem Stopp 9022 919 der Uebersetzung, besser (damit man es nicht vergisst) schon von Anfang an, die Taste ''Schaltersprung'' druecken (vgl. Fussnote zu 1.1.1.) Will man eine Befehlsliste des erzeugten (Objekt-) Programmes haben, dann kann man sie durch Anwendung des Bibliotheksprogrammes BA II erstellen.

1.4. Der Arbeitsgang ''Rechnen''

1.4.1. Schreibweise der Daten

Die Zahlen werden nach Algol-Vorschrift geschrieben (mit Dezimalpunkt und tiefgestellter 10 (sofern noetig), ohne vorheriges Malzeichen bei vorhandenem Exponentialteil. Zwischen den einzelnen Zahlen muss ein Trennungszeichen stehen; solche sind

a) 3 oder mehr unmittelbar aufeinanderfolgende Zeichen ''Zwischenraum'' (mittlere breite Taste des Fernschreibers)
b) das Zeichenpaar ''Wagenruecklauf - Zeilentransport''
c) das Trennungszeichen Komma oder Semikolon.

Die Reihenfolge der Daten ist durch die read-Anweisung im Programm festgelegt. An das Ende des Datenstreifens setze man unbedingt Trennungszeichen, am besten Kommas, und zwar mindestens fuenf.
Obwohl in der Mehrzahl der Faelle ohne Dezimalpunkt oder Zehnerexponent eingegebene, d.h. wie 'integer' geschriebene, aber als 'real' vereinbarte Zahlen vom Rechner in 'real'-Zahlen umgewandelt werden (s.a. 2.1.3.), ist dies bei Prozeduren und bedingten arithmetischen Ausdruecken nicht immer der Fall. Zur Vermeidung von Unannehmlichkeiten wird deshalb dringend geraten, 'real'-Zahlen auch als solche zu schreiben (z.B. 5.0, 3 ₁₀ 2; die Schreibweise 5. ist nach Algol nicht erlaubt.) Die Eingabe von Werten fuer 'boolean'-Ausdruecke kann mit 'true' oder 'false' oder aber mit +1 oder -1 erfolgen. (Ausgabe: immer + oder -1).

1.4.2. Durchfuehrung der Rechnung

Eingabeschalter: α , Ausgabeschalter: α_{SF} oder α_{FS}

Hat man den Datenstreifen in den Fotoleser eingelegt, so kann die Durchfuehrung der Rechnung auf zweierlei Weise eingeleitet werden:

a) Steht der Rechner (nach Uebersetzen des Programms) auf Stopp 9022 919 oder (nach Ausstanzen des Objektprogramms wegen nicht gedrueckter Schaltersprungtaste) auf Stopp 9033 919, so ist zum Beginnen der Rechnung nur die Starttaste zu druecken. [5] Der Rechner liest dann, (falls notwendig), den Datenstreifen ein, rechnet und gibt die Ergebnisse aus. Das normale Programmende wird angezeigt durch

Stopp 9099 919

b) Steht der Rechner (nach vorausgegangenem, ganz zu Ende gekommenem normalen Programmablauf) auf Stopp 9099 919 dann bewirkt Druecken von START den Beginn einer neuen Rechnung mit demselben Programm.

c) Ist weder 1.4.2.a) noch 1.4.2.b) der Fall, dann muss das (Objekt-) Programm ab seiner Anfangsadresse n (n = vierstellige Zahl) gestartet werden:

Drehschalter n 991, F, START
D, START

Im allgemeinen ist n = 0000, da der Compiler das erzeugte Objektprogramm ab 0000 in den KS schreibt. n hat nur einen anderen Wert, wenn man das Programm gemaess 1.3. ausgestanzt und ab n statt ab 0000 eingelesen hat.

d) Stoerungen waehrend der Rechnung: siehe 1.1.4.2) bis 4)

---

[5] War dem, Stopp 9022 919 eine Fehlermeldung vorausgegangen, so geht der Rechner jetzt beim Starten auf Stopp 9044 919 und bleibt bei erneutem Starten auf diesem Stopp als Hinweis auf das fehlerhafte Programm.

1.4.3. Fehler waehrend der Rechnung

Gewisse Fehler koennen sich auch waehrend des Rechnens durch Stoppen des Rechners bemerkbar machen, so z.B. negatives Argument fuer ln(x), x 'power' n und sqrt(x), gewisse Kapazitaetsueberschreitungen u.a. Bleibt der Rechner stehen, so leuchtet die rote Lampe in der Taste Stopp auf. Gleichzeitig leuchtet eine der 9 weissen Anzeigelampen auf und zeigt den Grund des Stoppens an:

Exp.-Stop oder Ueberlauf-Stop (oder beides)

Stopp, weil im Lauf der Rechnung der Zahlenbereich des ER 56 ueberschritten wurde.

Div-Stop (meist mit Exp.-Stop)

Es wurde eine Zahl durch 0 dividiert und somit der Zahlenbereich des ER 56 ueberschritten.

Schalter Stop — Es wurde die Prozedur Stop(a) mit a > 100 bei gedrueckter Taste SchStop im Lauf des Programmes erreicht (Reg. zeigt dann 80 b 919; b = rechte zwei Stellen von a).

Befehls-Stop — Es gibt 2 Moeglichkeiten:
a) Es wurde die Prozedur Stop (a) mit a < 100 im Lauf des Programmes erreicht (Reg. zeigt dann (90a 919)
b) Es trat einer der in Anlage 2 aufgefuehrten numerierten Stopps auf. (Reg. zeigt die Stopp-Nr. an).

Nach solchen Stopps zeigt <I5>, an welcher Stelle des Quellenprogrammes der Stopp auftrat: Die Nr. der Zeile liegt zwischen (<I5>)...(<I5>+4).
Es gibt noch folgenden Fehler waehrend der Rechnung, der keinen Halt der Maschine bewirkt: Das Programm befindet sich in einer Schleife, die entweder keinen Ausgang hat oder bei der die Bedingung zum Verlassen der Schleife nie erfuellt werden kann (z.B. Forderung ''x = 0'' und x = 0 wird nicht erreicht; Abhilfe: etwa ''abs(x) < $_{10}$ -8'' anstelle von ''x = 0'').

1.4.4. Rechnungen mit dem gleichen Programm aber mit anderen Datensaetzen

Wenn die Verarbeitung eines Datensatzes mit Stopp 9099 919 beendet ist, so wird die Verarbeitung des naechsten allein durch erneutes Druecken von D, START erreicht (vgl. 1.4.2.). Der Datensatz kann, nur durch ein Trennungszeichen (z.B. ein Komma) getrennt, an den vorhergehenden direkt angeschlossen sein.

Es werden dann noch soviele Grundsymbole vom naechsten Datensatz hereingeholt, bis die letzte Doppelzelle gefuellt ist. Beim Einlesen dieses naechsten Datensatzes ist dann sein Anfang bereits im Rechner. Wenn eine Rechnung nicht zu Ende (d.h. zum Stopp 9099 919) gefuehrt wird, so muss man den naechsten Datensatz mit (n+2) 991; F, START; D, START starten. (n = Anfangsadresse, vgl. 1.4.2.c). (Der allererste Datensatz verlangt jedoch immer Start mit n 991).

Man kann aber auch die Datensaetze voneinander durch 5 Kommata trennen. Dann kann man jeden Datensatz mit n 991 starten.

1.5. Verwendung der ausgestanzten Ergebnisse

Die ausgestanzten Zahlen koennen (ohne Umcodierung) als Daten fuer weitere Rechnungen wieder eingegeben werden.

Werden von einem Programm groessere Datenmengen erzeugt, die man sofort wieder als Eingangsdaten fuer andere Programme verwendet, dann empfiehlt es sich, sie im $(\frac{5}{1})$-Code ausstanzen zu lassen (Ausgabedrehschalter auf $(\frac{5}{2})$ waehrend der Rechnung) und in diesem Code beim Lauf des Programmes, das sie als Eingangsdaten verwendet, wieder einzulesen (Eingabedrehschalter $(\frac{5}{2})$).
Grund: im $(\frac{5}{2})$-Code werden Stanz- und Lesefehler erkannt.

2. Numerische Charakteristik und Kapazitaetsbeschraenkung

2.1. Numerische Charakteristik

2.1.1. Maschineninterne Zahldarstellung

a) Die Zahlen vom Typ 'real' entsprechen der maschineninternen Gleitpunktdarstellung mit 11 Mantissenstellen. Bei Ein- und Ausgabe werden hoechstens 10 Stellen uebermittelt.

Vz	r	r	r	r	r	r	r	r	r	r	r	c	h

Vz = 1 ist +, Vz = 2 ist -,
zweistellige Charakteristik: ch = Exponent +50

b) Zahlen vom Typ 'integer' belegen im Rechner ebenfalls eine Doppelzelle:

Vz	g	g	g	g	g	g	g	g	g	g	0	0	0

Die letzten 3 Ziffern werden nicht ausgenutzt; es koennen also als 'integer' ganze Zahlen mit bis 10 Stellen erscheinen. Auf Indexpositionen werden nur die 4 letzten Stellen g einer ganzen Zahl ausgewertet. Beim Ausgeben von 'integers' werden vorlaufende Nullen unterdrueckt.

c) Boolesche Zahlen werden wie 'integers' behandelt.

2.1.2. Bereichsgrenzen der maschineninternen Zahldarstellung

Typ 'real'	$10^{-50} \leq r < 10^{49}$ oder 0
Typ 'integer'	$i < 10^{10}$
Typ 'boolean'	true, false oder +1, -1 in der Eingabe +1, -1 in der Ausgabe

2.1.3. Typ-Umwandlung

Massgebend ist die Konvention:

''Die Typenvereinbarung im Programm hat Vorrang''.

Der Uebersetzer unterscheidet streng zwischen Groessen vom Typ 'real' einerseits und Groessen vom Typ 'integer' und 'boolean' andererseits:
Die ersten haben Dezimalpunkt und/oder eine tiefgestellte Zehn, die zweiten nicht.
Im allgemeinen wird eine notwendige Typ-Umwandlung im Programm vorgenommen, doch wird dadurch Programm und Rechenzeit verlaengert. Es gibt aber auch Faelle, bei denen eine falsche Typ-Angabe zu vollstaendig falschen Ergebnissen fuehrt, so

A. wenn korrespondierende Groessen in Prozeduranweisung und Prozedurvereinbarung nicht vom gleichen Typ sind,

B. wenn beide Teile eines bedingten arithmetischen Ausdrucks nicht vom selben Typ sind.

Eine 'integer'-Zahl bleibt 'integer' auch wenn sie im Laufe der Rechnung den Wert 10^{10} erreicht oder uebersteigt: Die Ziffern vom Stellenwert 10^{10} und hoeher werden unterschlagen, man erhaelt also ein vollstaendig falsches Resulatat. In diesem Fall ist es also unbedingt noetig, diese Variable als 'real' zu vereinbaren. Allgemein wird es sich lohnen, alle 'real'-Zahlen mit Dezimalpunkt oder Zehnerpotenz zu schreiben. Die Wandlung vom Typ 'real' in den Typ 'integer' geschieht so:

'integer'-Groesse: = entier ('real'-Groesse + 0,5)

2.2. Beschraenkungen fuer das Algol-Programm durch den Uebersetzer ALCOR ER 560

2.2.1. Beschraenktes Element

		Hoechstzahl
Gleichzeitig definierte	Variable	ca. 150
	Felder	25
unterschiedliche Zahlen		98
''	UP-Aufrufe	50
2 * Prozeduren + Parameter		50
externe und interne Marken		116
Schachtelung		50

2.2.2. Interne Marken werden benoetigt von	Anzahl
Laufanweisungen, Alternativen	2
Bedingungen, Verteiler, Prozeduren, Parameter mit Namenslauf	1

2.2.3. Bei der Schachtelung zaehlt	
'begin' und Ergibtzeichen	zweifach
'for', 'if', oeffnende eckige oder runde Klammer, Komma	einfach

2.3. Grenzen fuer die Speicherung

2.3.1. Speicherbedarf fuer den Compiler

Der Compiler benoetigt die Kernspeicher 3000...5999; es koennen jedoch vom Objektprogramm die Kernspeicher bis 4999 ueberschrieben werden.
Auf der Trommel sind vom Compiler die Bloecke 300...599 belegt, waehrend der Uebersetzung ausserdem die Bloecke 250...299 fuer Vormerkungen; von Block 249 an abwaerts wird das erzeugte Objektprogramm zwischengespeichert.

2.3.2. Zulaessige Laenge des Quellenprogramms

Hierueber kann keine Aussage gemacht werden, schon weil die relative Zahl der Zwischenraeume von Programm zu Programm stark schwankt.

2.3.3. Das erzeugte Objektprogramm

Siehe 1.2.4. und 1.3. und Kap. 6.

2.3.4. Plaetze fuer Daten

Ab Objektprogrammende +5 (s. 1.2.4.) bis 5979; dann fuer Felder ab y bis 5979, wobei y die vierstellige Adresse +9 ist, die in Zelle n+8 steht. (n s. 1.4.2.c))
(Jede Zahl nimmt 2 Zellen ein.)

3. Anpassung des Algolprogramms an den Rechner

3.1. Verwendung von Trommel und Baendern als Datenspeicher kann nur ueber Code-Prozeduren geschehen; die Trommel wird aber weitgehend vom Compiler in Anspruch genommen (s. 2.3.1.).

3.2. Das erzeugte Programm wird schneller, wenn man

a) zwischen step und until einzelne (positive oder negative) Zahlen einsetzt,
b) keine Ausdruecke vor 'until' und 'do' gebraucht,
c) in Ausdruecken keine Groessen verschiedenen Typs benuetzt,
d) keine Parameter mit Namensaufruf verwendet.

4. Besondere sprachliche Einschraenkungen fuer ER 56.0

1. Formale Parameter, die Prozeduren sind (die also in der Parameterliste einer anderen Prozedur stehen), muessen vor ihrer Benuetzung vereinbart sein. Notfalls kann dies in einer Pseudo-Vereinbarung geschehen, in dem der Prozedurrumpf leer ist:

```
'procedure's(1,i,k);
'real'h,i,k;
'begin''end'
```

2. Standard-Funktionen koennen nicht als Parameter benutzt werden.

3. Formale Parameter von Prozedurkoerpern koennen nicht in die Parameterliste der read- und print-Prozedur uebernommen werden.

4. In Prozedurkoerpern kann die Laufvariable einer 'for'-Anweisung nicht formaler Parameter sein.

5. In der Parameterliste einer Prozeduranweisung darf kein Prozeduraufruf vorkommen (Verbot u.a. von rekursiven Prozeduren, also Pr(Pr(x)), wenn Pr eine Algolprozedur ist).

6. Einander entsprechende formale und aktuelle Parameter einer Prozedur muessen vom gleichen Typ sein und so vereinbart bzw. spezifiziert sein.

7. In einem step-until-Element darf kein Prozeduraufruf erfolgen.

8. In einer type- und print-Anweisung darf keine Laufanweisung stehen (aber eine bedingte Anweisung).

9. Zu vermeiden ist read (i, feld[i]).

10. Potenzen duerfen keine negative Basis oder die Basis 0 haben; wohl aber den Exponenten 0. Fuer (-a) 'power' n schreibe man

```
x: = 1; 'for' i: = 1 'step'1'until'n'do'; x: = -a * x.
```

11. Ein ganzzahliger Exponent einer Potenz muss kleiner als 10 000 sein.

5. Fest vereinbarte Prozeduren

5.1. Die Einleseprozedur READ (a,b...)

Mit dieser Anweisung wird jeder in der Klammer aufgefuehrten Groesse eine Zahl zugeordnet in natuerlicher Reihenfolge. Der Zahlenstreifen soll zum Abschluss mindestens 5 Kommas haben. Die Zahlen koennen beliebige Algolform und beliebigen Typ haben. Kommentar und Klartext werden ueberlesen. Im uebrigen beherzige man das in Abschnitt 1.4.1. Gesagte.

5.2. FORMAT (...,.)

Die Formatanweisung dient zur Steuerung des Druckbilds. Es gibt sie in 3 Formen

a) FORMAT (sg)
b) FORMAT (sg, zw)
c) FORMAT (sg, zw, sn)

Die Klammerparameter muessen ganze Zahlen, ganzzahlige Variable oder ganzzahlige Ausdruecke sein. Die Formatanweisung (FA) gilt fuer alle im Rechengang folgenden Ausgabeanweisungen bis zu einer neuen FA oder Programmende. Sonst kann sie im Anweisungsteil an beliebigen Stelle stehen. Innerhalb einer Printanweisung kann nur eine FA gelten.

Es bedeuten:

sg: Die gewuenschte Gesamtstellenzahl (bei Gleitpunkt jedoch nur die der Mantisse).

zw: Die Anzahl der Zwischenraeume vor der Zahl (s. aber 5.4.b).

sn: Die Anzahl der Stellen nach dem Dezimalpunkt. (Es ist sn < sg notwendig). Damit kann eine Umwandlung von Gleitpunktzahlen in Festpunktzahlen gesteuert werden.
Bei sn = 0 oder sg - sn < Exponent werden die Zahlen in Gleitpunkt-Darstellung ausgegeben.

Soll nur sg veraendert werden, so genuegt Form a; wenn zw oder zw und sg, dann Form b. Enthaelt ein Programm keine Formatanweisung, dann wird mit

FORMAT (10, 2,0) ausgegeben.

5.3. Die Ausgabeanweisung PRINT (a, b...)

Die Anweisung print bewirkt nacheinander:
Wagenruecklauf, Zeilenvorschub, Ausdrucken der Werte a,b...
in dem zuletzt angegebenen Format.
Die Zahlen sind folgendermassen zusammengesetzt:

Gleitpunktzahlen:	zw Stellen	Zwischenraeume
	1 Stelle	- oder Zwischenraum
	1 Stelle	Dezimalpunkt
	sg Stellen	Mantisse (gerundet)
	1 Stelle	tiefgestellte Zehn
	1 Stelle	Exponentenvorzeichen
	2 Stellen	Exponent
Festpunktzahlen:	zw Stellen	Zwischenraeume
	1 Stelle	- oder Zwischenraum
	sg-sn Stellen	Ziffern vor dem Punkt; wenn weniger Ziffern vorhanden (vorlaufende Nullen), dann statt jeder vorlaufenden Null ein Zwischenraum.
Ganze Zahlen:	zw Stellen	Zwischenraeume
	1 Stelle	- oder Zwischenraeume
	sg Stellen	Ziffern der ganzen Zahl. (Vorlaufende Nullen wurden nicht gebucht, aber der Platz frei gehalten).

Bei Festkommazahlen und bei ganzen Zahlen findet Nullenunterdrueckung und Vorzeichenverschiebung statt. Die Anzahl der moeglichen Parameter in der Klammer, also der Zahlen je Zeile ist von der Formatangabe abhaengig. Beim Fernschreiber fasst die Zeile 68 Stellen. Damit kann man die zulaessige Anzahl von Parametern berechnen. Hat man zuviel, so wird der Ueberschuss am Zeilenende ineinander gedruckt. Bei FORMAT (10,2,0) gehen 4 Gleitpunktzahlen nicht auf eine Zeile, wohl aber bei FORMAT (10, 1, 0).

Man beachte 4.9.: In einer print-Liste darf keine Laufanweisung stehen, wohl aber eine bedingte Anweisung.

PRINT ohne Parameterliste bewirkt lediglich Wagenruecklauf - Zeilenvorschub.

5.4. Die Ausgabeanweisung TYPE (a,b...)

Diese Anweisung hat gegenueber PRINT folgende Unterschiede

a) kein Wagenruecklauf - Zeilenvorschub,
b) keine Zwischenraeume zwischen den Zahlen (unabhaengig von der Vereinbarung im FORMAT),
c) bei ganzen Zahlen fallen unterdrueckte Nullen und positive Vorzeichen ganz weg.

5.5. Die Ausgabeanweisung WRITE (''<beliebiger Text>'')

Mit dieser Anweisung kann Klartext ausgedruckt werden; man kann also Ueberschriften, naehere Bezeichnungen der Rechenergebnisse, Angaben ueber Nichtexistenz von Loesungen usw. zwischen die Zahlenergebnisse setzen. Dabei ist zu beachten, dass der Text an der Stelle einsetzt, an der die letzte Druckanweisung aufgehoert hat. Es wird sich daher meistens empfehlen, nach den Anfuehrungszeichen (doppelter Apostroph) mit Wagenruecklauf - Zeilenvorschub zu beginnen, damit der nun folgende Text vorne in der Zeile beginnt. Hoert nun dieser z.B. mit a ='); auf, so setzt die Ausgabeanweisung TYPE(a) den Wert von a hinter das Gleichheitszeichen. Bei PRINT(a) kommt dagegen a auf die neue Zeile.

5.6. Die Stoppanweisung STOP(a)

a muss eine ganze Zahl, eine ganzzahlige Variable oder ein ganzzahliger Ausdruck sein.

Mit a < 100 unbedingter Stopp (Man verwende fuer a nach Moeglichkeit nicht die Stoppnummern von Anlage 2. (s. 1.4.3.)).

Mit a > 100 Schalterstopp.

Die beiden letzten Ziffern der Groesse a werden angezeigt (s. 1.4.3.). Mit Druecken der Starttaste bei Dauer wird die Rechnung fortgesetzt. Eine Ausdruck-Anweisung hinter einer Stoppanweisung druckt den Ausdruckspeicher 5980-5999 nicht aus, wenn dieser nicht vollstaendig aufgefuellt ist. Der Rest wird erst gedruckt durch den Sprung ins Ausdruck-Programm, der unmittelbar vor dem Befehlsstopp 9099 919 stattfindet. Nach Stopps der Maschine infolge STOP(a) ist so oft zu starten, bis das normale Programmende (Stopp 9099 919) erreicht ist.

6. Das Objektprogramm im Interncode des ER 56.0

Wie bereits erwaehnt, ist das ausgestanzte Programm relativ programmiert und kann ab einer geraden Adresse n eingelesen werden; es wird dann ab dieser Adresse gestartet. Es enthaelt saemtliche benoetigten Standardfunktionen und -prozeduren, sowie das Einlese- und das Druckprogramm; daher kann es verwendet werden, ohne dass der Compiler in der Rechenanlage steht. Aus diesem Grund ergibt ein kurzes Quellenprogramm schon ein beachtlich grosses Objektprogramm.

Kuerzere Algol-Programme werden deshalb besser nicht ausgestanzt, sondern jedesmal neu uebersetzt. Bei Wiederholung der Rechnung mit neuen Daten beachte man die Ausfuehrung ueber die Eingabe, Abschnitt 1.4.4.; steht der Rechner auf Stopp 9099 919, so ist Weiterstarten mit ''Dauer'' in jedem Fall richtig; bei anderen Stopps kann man die weitere Rechnung mit Start bei n oder n+2 versuchen (vgl. 1.4.2. und 1.4.4.). Die Adresse des Doppel-Omega-Wortes am Ende des Objektprogrammes erhaelt man durch Subtrahieren von 5 von der Adresse x des im Kernspeicher n+2 stehenden Wortes x 981 (vgl. 1.2.4.).

Das Objektprogramm besteht aus mehreren Abschnitten:

1) Der Vorblock beginnt mit Loeschen der Programmerker 2 und 3, welches die Anfangsbedingungen fuer die Einlese- bzw. Druckprozedur herstellt. Dann werden die Indexregister 1...5 geladen. Indexregister I 1, I 2, I 3 behalten ihren Wert waehrend der ganzen Rechnung bei.

   I 1 zeigt den Anfang des Variablenblocks (s. 12))

   In der Speicherzelle mit der Nummer <I 1>-8 (die identisch ist mit der Adresse in Zelle 0002 abzueglich 5) steht das Doppelomegawort des Programms.

   I 2 ist die Adresse des Anfangs des Zahlenblocks (s. 7))
   I 3 weist auf den ersten Befehl des Hauptverteilers (s. 9)
   I 4 gibt den Beginn des Blocks der Felder an (s. 13))
   I 5 ergibt die Zuordnung zwischen Quellenprogramm und eigentlichem Programm.

2) In dem sich an den Vorblock anschliessenden Programmteil wird nach je 5 Zeilen des Fernschreibprotokolls (also des Quellenprogramms) ein Befehl n 591 gebracht. Es befinden sich also waehrend des Rechengangs im Indexregister 5 laufend die Zahlen 5, 10, 15, ...; man kann so aus dem Inhalt von I 5 auf die ungefaehre Lage eines Fehlers im Quellenprogramm schliessen oder das ausgedruckte Objektprogramm dem Algol-Quellenprogramm zuordnen (da ja beide Programme ungefaehr gleichen zeitlichen Ablauf haben). Entsprechend ihrer Lage im Quellenprogramm werden die Aufrufe vorgenommen und die Operationen verknuepft; die Algolprozeduren werden hier zerlegt und uebersetzt, auch die Koepfe der Code-Prozeduren stehen hier, waehrend die (Ruempfe der) festen Prozeduren (wie Standardfunktionen, Lese-, Druck-, Formatprozeduren), sowie die Ruempfe der Codeprozeduren erst nach dem Zahlenblock und damit hinter dem Stoppbefehl des Programms stehen. Diese Programmteile werden nach Bedarf ueber einen Befehl U 310 (U = 0,1,2,...) [6] angesprungen, der zum Hauptverteiler fuehrt (er steht ab I 3 kurz vor dem Omega-Doppelwort am Programmende).

---

[6] Bei einigen internen Programmen (z.B. Zahlenumwandlung, Pruefen von Feldvereinbarungen) heisst der Sprung m 2 10 mit m ≥ 9946, s. Abschnitt 6) des Objektprogrammes.

Von dort aus geht es mit einem Sprung V993 (damit die Rueecksprungadresse ins Hauptprogramm nicht verloren geht) in das betreffende Unterprogramm.

Muessen (mit einem Unterprogramm) Zahlenwerte eingelesen oder ausgegeben werden, so steht nach dem Sprungbefehl n 3 10 der sog. Versorgungsblock. Er enthaelt die indizierten programminternen Adressen saemtlicher diesbezueglicher Zahlenwerte. Es steht also in den 4 ersten Stellen die relative Adresse bezueglich des in der 5. Stelle angegebenen Indexregisters; an 6. und 7. Stelle steht bei real-Zahlen 00, bei Zahlen vom Typ integer 11. Abgeschlossen wird ein Versorgungsblock durch 9999 000.

3) Nach dem letzten Sprung, der im allgemeinen ein Sprung in das Ausgabeprogramm sein duerfte, folgt der Rechenstopp 9099 919.

4) Unmittelbar danach kommt der Sprungbefehl, der I 9 - fuer einen neuen Rechengang - auf die Adresse 0002 setzt, also unter Umgehung der Bedingungen fuer den Erst-Anfang PM 2 und 3 ''aus'', die in Speicherzelle 0 und 1 sitzen.

5) Hat dieser Sprungbefehl eine geradzahlige Adresse, so wird der Leerbefehl 0000 000 hinzugegeben.

6) Es folgt ein programminterner Block von

   16 Befehlen, wenn im Programm keine Felder vorkommen (besorgt Umwandlung vom integer- zum real-Typ und umgekehrt)

   44 Befehlen, wenn nur eindimensionale Felder vorkommen (worin die obigen 16 Befehle enthalten sind; zu diesen kommt z.B. die Pruefung hinzu,ob die zweite Grenze in der Feldvereinbarung groesser ist als die erste oder ob fuer die Felder genuegend Platz vorhanden ist.)

   54 Befehlen, wenn mindestens ein mehrdimensionales Feld vorkommt. Darin sind alle 44 Befehle von oben enthalten.

Anschliessend steht der

7) Zahlenblock (Anfangs-Adresse in I 2). Er enthaelt die integer-Zahlen 0 und 109, sowie saemtliche expliziten Zahlen, seien sie vom real- oder integer-Typ, die im Quellenprogramm vorkommen, aber jede nur einmal; daran anschliessend

8) der Unterprogrammblock mit Leseprogramm, Standardfuntionen, Code-Prozeduren in der Reihenfolge der Vereinbarung bzw. des Aufrufs.

9) der Hauptverteiler (mit v 993 Befehlen) veranlasst die Spruenge nach den Unterprogrammen (im weiteren Sinn). Seine Adresse steht in I 3. Im Bedarfsfall kommt nach den Sprungbefehlen ein Nullbefehl und darauf

10) das Doppelomegawort als Ende des Programms.

Weitere Speicherplaetze werden zur Unterbringung der Daten und Ergebnisse benoetigt. Im Falle, dass bei der Uebersetzung die Schaltersprungtaste auf ''ein'' stand, werden die anschliessenden Speicher fuer 6 Hilfszellen und fuer die Variablen benuetzt, stand sie auf ''aus'', so werden die Speicher ab 2000 genommen, sofern das Doppelomegawort im nullten oder 1. Kernspeicher steht, sonst wie bei Schaltersprung ''ein''. Zuerst kommen also

11) 6 Hilfszellen. Anschliessend der

12) Block der Variablen: Seine Anfangsadresse steht in I 1. Die Variablen stehen (in Doppelzellen) in der gleichen Reihenfolge wie in den Vereinbarungen. Sie sind nur in den Zeiten zugaenglich, in denen sie definiert sind. Bei dieser Abzaehlung werden nicht mehr definierte Variable nicht mitgezaehlt. Anschliessend einige Hilfszellen.

13) Vor der ersten Feldvereinbarung steht in I 4 die Anfangsadresse des Blocks der Felder, spaeter gibt I 4 den Anfang des freien Teils dieses Blocks an.

14) Fuer die zur Ausgabe kommenden Ergebnisse dienen die Zellen 5980 bis 5999 als Pufferspeicher.

Anlage 1

Bedeutung der ausgegebenen Fehlernummern (waehrend der Uebersetzung)

Fehler Nr.	Bedeutung
1	Syntaktischer Fehler
2	Syntaktischer Fehler in ARRAY-Vereinbarung
3	Blockstrukturfehler
4	Nichtuebereinstimmung einer Dimension
5	Fehler im arithmetischen Ausdruck
6	Nichtdefinierte Variable
7	Nichtdefiniertes Feld
8	Ungueltiges Wortsymbol
9	Ungueltiger Anweisungsname
10	Zahlenfehler
11	Fehler in Prozedurvereinbarung
12	Prozedurfehler
13	Nichtvereinbarte Prozedur
14	Syntaktischer Fehler in Typenvereinbarung
15	Syntaktischer Fehler in Verteilervereinbarung
30	Nicht vorhandenes Standardunterprogramm
51	Zu viele Variable
52	Zu viele Felder
53	Zu viele Zahlen
54	Zu viele Unterprogrammaufrufe
55	Zu viele Prozeduren
56	Zu viele Marken
57	Zu viele Spruenge
58	Zu grosse syntaktische oder operative Schachtelung
60	Zu langes Programm.

Fehlerangaben fuer Zeilen, die nicht im Programm existieren, deuten auf ein fehlendes 'end'.

Bedeutung der Stopps waehrend der Uebersetzung

Es koennen nur Befehlsstopps auftreten, d.h. ausser der roten Lampe in der Taste STOP leuchtet die weisse Lampe ''Befehlsstop''. Die Stopp-Nr. wird bei gedrueckter Taste ''Reg.'' an der 7-stelligen Anzeige angezeigt.

9000 919 — Compiler steht zum Uebersetzen bereit

9011 919, 9012 91,... — Codeprozedur muss in den Fotoleser eingelegt werden

9022 919 — Stopp nach dem Uebersetzen

9033 919 — Stopp nach Ausstanzen des Objektprogrammes

9066 919 — Programm zu lang (Kapazitaetsueberschreitung)

andere: — (treten beim Rechnen auf; s. Anlage 2)

Anlage 2

Bedeutung der Befehlsstopps waehrend der Rechnung

Es sind hier nur die Befehlsstopps aufgefuehrt, die waehrend der Rechnung auftreten koennen (ausser der roten Lampe in der Taste STOP brennt auch noch die weisse Lampe Befehlsstop; bei gedrueckter Taste ''Reg.'' wird die Stopp-Nr. an der 7-stelligen Anzeige angezeigt).

Leuchtet nach einem Stopp der Maschine waehrend der Rechnung nicht die Lampe Befehlsstopp'' auf, sondern eine andere aus dem Anzeigenfeld fuer Stopp-Ursachen, dann liegt kein Befehlsstopp vor. Dann siehe 1.4.3.

Stopp Nr.	Bedeutung
9044 919	Stopp nach versuchtem Start der Rechnung weil beim Uebersetzen Fehler erkannt wurden
9093 919	zu grosse Formatangabe.
9094 919	Negativer Radikand beim Quadratwurzel-Unterprogramm
9095 919	negatives Argument zum ln oder ln(0) (dieser Stopp tritt auch bei negativer Basis b einer Potenz b 'power' c auf)
9096 919	Obere Dimensionsgrenze eines Feldes kleiner als die untere
9097 919	Ueberschreiten der Kernspeicherkapazitaet
9098 919	Ueberschreiten eines Verteilerbereichs
9099 919	normales Programmende (Druck auf die Starttaste leitet neuen Rechenvorgang ein)
andere:	(Koennen durch STOP(a) mit a < 100 entstehen, s. 1.4.3. und 5.6.4)

Anlage 3

Stand des Befehlsfolgezaehlers (Befehlsadressenregister) I 9 bei Stopps waehrend der Uebersetzung

Reg	<I 9>
9000 919	5021
9011 919	5369
9022 919	5540
9033 919	5552
9044 919	5557
9066 919	5297