

Über „Über den Plankalkül“ von Konrad Zuse*

Günter Hotz, Saarbrücken

Konrad Zuse schwebte mit dem Plankalkül mehr vor als eine Programmiersprache. Er wollte einen Plankalkül entwickeln, indem sich Algorithmen eindeutig beschreiben ließen und der es gestatten sollte, die Rechenpläne selbst zu Objekten von Berechnungen zu machen. Als Vorbild sah Zuse den Aussagenkalkül und die Prädikatenlogik an. Die Rechenpläne des Plankalküls sind Programme in unserer heutigen Terminologie.

Die Programme des Plankalküls enthalten drei Klassen von Variablen, die „Eingangsvariablen V“, die „Zwischenvariablen Z“ und die „Resultatvariablen R“. Verschiedene Variablen werden durch Indizes unterschieden. Die Variablen haben alle eine Struktur. Index und Struktur der Variablen werden stets mitgeführt. Die Programme können andere Programme als Unterprogramme enthalten (wobei rekursive Prozeduren nicht vorgesehen waren). Die Abgrenzung der Unterprogramme erfolgt durch Klammerpaare. Zur Bezeichnung der Unterprogramme dient der Buchstabe P mit einem Index, der das Unterpro-

gramm eindeutig bestimmt. Durch den „Randauszug“ wird die Struktur der Eingangs- und Ausgangsvariablen des Programms festgelegt. Die Variablennamen und die Unterprogrammnamen sind lokal. Die Werteübergabe an die Unterprogramme ist der „call by value“. Die Identifizierung der Variablen erfolgt durch die Stellung der Variablen im Aufruf und der Deklaration des Unterprogramms. Die Programme enthalten bedingte Anweisungen: Ist die Bedingung nicht erfüllt, wird zur nächsten Anweisung übergegangen. Es gibt verschiedene Varianten von Wiederholungsanweisungen.

Die eigentliche Überraschung für den heutigen Leser stellen die vorgesehenen Datenstrukturen dar. Der Basistyp und die Basisstruktur, auf der alles aufbaut, ist das Bit, das heißt die Variablen vom Typ *boolean*, wie wir heute sagen. Ist n eine natürliche Zahl, dann ist $S1 \cdot n = n \times S0$ die Bezeichnung für Variablen, deren Variationsbereich die n -stelligen binären Folgen sind. Eine Variable mit der Struktur $\square \times S0$ variiert über die Menge aller binären Folgen. Ist A eine Menge, dann entspricht die Operation $\square \times A$ der in der Theorie der formalen Sprachen gebräuch-

lichen Notation A^* . Induktiv: Ist σ eine bereits definierte Variablenstruktur, dann kann man aus dieser die Strukturen $\tau = n \times \sigma$ und $\tau = \square \times \sigma$ bilden. Das sind Folgen der Länge n bzw. beliebiger Länge der Struktur σ . Hinzu kommt weiter die Operation, die aus Strukturen σ und τ als neue Struktur das Paar (σ, τ) bildet. Die beiden erst genannten Möglichkeiten erlauben es, binäre Arrays beliebiger Dimensionen zu definieren. Die Bildung des Paares, die als nicht assoziative Operation gedacht ist, erlaubt es, beliebige binäre Bäume zu definieren. Betrachtet man alle Operationen zusammen, so sieht man, dass sich im Plankalkül sehr allgemeine Listenstrukturen definieren lassen. Der Plankalkül enthält einige Standardstrukturen, für die auch feste Bezeichnungen vorgesehen sind. Hierzu gehören – wieder in unserer Terminologie – die Typen *integer* und *real*. Diese können auch genauer durch Angabe der Stellenzahlen, die für den Exponenten und die Mantisse vorgesehen sind, spezifiziert werden.

Konrad Zuse erprobte seinen Kalkül an nicht trivialen Beispielen. Darunter befanden sich numerische Programme, Sortierprogramme, Programme, die Pro-

* Aus dem Beitrag „Konrad Zuse: Forschung und Entwicklung“ von Günter Hotz, erschienen im Informatik-Spektrum 3, 41–47 (1980).

bleme der Graphentheorie betreffen, Schachprogramme und Programme nicht numerischen Rechnens. Es handelte sich, wie Donald E. Knuth und Luis Trabb Pardo in einem 1976 erschienenen technischen Bericht bemerkten, um die weitaus kompliziertesten Algorithmen, die zu dieser Zeit jemals programmiert worden waren.

Nach all diesem fragt man sich, woran es lag, dass der Plankalkül auf die Entwicklung der Programmiersprachen keinen Einfluss hatte, dass er von den Vätern der ersten Programmiersprachen nicht zur Kenntnis genommen wurde.

Neben der „zweidimensionalen“ Notation der Programme war es wohl äußerlich das Fehlen des vom Maschinenprogrammieren her gewohnten *goto*, das des *if-then-else* und der rekursiven Programme, was den Plankalkül in Misskredit brachte, bei den wenigen Leuten, denen er bekannt war – der Plankalkül wurde erst recht spät publiziert; Zuse hatte keine Zeit, den Kalkül zu „verkaufen“ vielmehr musste er eine Firma gründen und versuchen, seine Erfindungen durch Patente zu schützen –. Tiefer liegt eine andere Ursache. Gerade seine reichhaltigen

Datenstrukturen, die uns den Plankalkül so interessant macht, mussten eine technische Realisierung dieser Sprache damals wenig aussichtsreich erscheinen lassen. Man hatte doch schon genug mit den *syntaktischen* Schwierigkeiten zu tun, die die freiere Verwendung von Namen mit sich brachte. Die Zielsetzung Zuses und die der Väter von ALGOL oder FORTRAN waren sehr verschieden. Für Zuse war die endgültige Maschinenstruktur noch nicht gefunden. Diese hatte sich den Aufgaben anzupassen. Also musste der Weiterentwicklung der Maschinen die Entwicklung eines Plankalküls vorausgehen, der es erlaubte, die einschlägigen Probleme zu beschreiben. Die Väter von FORTRAN beispielsweise wollten eine Programmierhilfe für ihre vorhandenen Maschinen. Diese Hilfe brauchten sie morgen schon. Die übersetzten Programme mussten effizient sein.

Zuse war 1945 mit seinem Plankalkül wohl zu früh. Er scheiterte an den technischen Möglichkeiten, so wie es Babbage 100 Jahre früher mit seinen Plänen für die Entwicklung einer programmgesteuerten Rechenmaschinen ergangen ist.



Prof. Dr. Dr. h.c. mult. Günter Hotz

studierte Mathematik und Physik in Frankfurt und Göttingen. Nach der Promotion in Göttingen arbeitete Günter Hotz drei Jahre in der Industrie. Er wirkte bei Telefunken bei der Rechnerentwicklung mit, der Postscheckautomatisierung und dem ersten Flugsicherungssystem. Er habilitierte in Saarbrücken und lehrte in Tübingen und Saarbrücken. Günter Hotz ist Leibniz-Preisträger der Deutschen Forschungsgemeinschaft, Träger der Konrad-Zuse-Medaille der Gesellschaft für Informatik sowie des Saarländischen Verdienstordens. Ehrendoktoren wurden Günter Hotz verliehen von der Universität Frankfurt, der Universität Darmstadt, der Universität Tiflis und der Universität Paderborn. Günter Hotz ist Mitbegründer und erster Vorsitzender der Gesellschaft für Informatik e.V. Seit 2002 ist Günter Hotz Ehrenmitglied der GI.

Adresse: Universität des Saarlandes, Fachbereich Informatik, Postfach 151150, 66041 Saarbrücken, Deutschland, E-Mail: in11gh@rz.uni-sb.de