Register - Maschinen

1. Zur allgemeinen Konzeption

Die Konzeption der Register-Maschine geht auf Turing (19..), Post (19..), Wang (19..), Moore (19..), Kalmár (19..), Minsky (19..) u.a. zurück. In der reinen Form dient sie der Klärung der Begriffe Berechenbarkeit und Entscheidbarkeit. In der angewandten Form hat die Einführung von Registerstrukturen neben den schon traditionellen Rechen- und Speicherwerken praktische Bedeutung bekommen.*) Hier soll im wesentlichen nur die reine Form behandelt werden.

2. Konventionen

Eine Registermaschine besteht hier aus:

2.1 Einer endlichen Folge von Registern, A, B, C, ..., X, ... , deren Inhalt durch bestimmte Operationen verändert wird, und

2.2 die auf das Vorliegen bestimmter Eigenschaften befragt werden können, ferner

2.3 einem Flußdiagramm, aus Elementarmaschinen, welches festlegt, was wann geschieht, also: dem Programm.

Register können sein (z.B.)

2.5 Zähler, auf welche die Operationen s(+1, "successor") und p(-1, "predecessor") einwirken; welche die Frage ?=0≠? beantworten. Die Operation pX soll an die Voraussetzung ⟨X⟩≠0 gebunden sein.

---

*) Ich erinnere mich noch an einen Programmierkurs in den 50er Jahren, in welchem das Rechenwerk für jede Änderung eines Iterations-Index geräumt werden mußte.

2.6 Bänder mit einer diskreten Folge von Feldern, von welchen jeweils eines ansteht. Auf diese wirken Schreib-Operationen d(1 drucken), c(0 drucken, "clear") oder Schiebe-Operationen r(zum rechten) bzw. l(zum linken Nachbarfeld). Gefragt werden kann jeweils (nur) nach dem Inhalt [X] des anstehenden Feldes.

2.7 Keller, oder offene Schieberegister, wirken ähnlich wie Bänder, mit den Unterschieden:
Es steht ständig das obere (Keller) bzw. erste Feld von links (Schieberegister) an; die Operation r kommt nur in den Verbindungen rd bzw. rc vor; sinngemäß d bzw. c nur in den Verbindungen rd bzw. rc. Da jedes reale Schieberegister endlich ist, muß gef. definiert werden, was (0 oder 1) bei l vom Ende (das ja nicht "ansteht") hereingeholt ("gezogen") wird .

2.8 Zyklische Schieberegister. Hier ist auch "eitel" r bzw. l möglich, Es genügt anscheinend, d, c nur in der Form rd, rc einzuführen, d.h.: Einschieben (serielle Eingabe) von 1 bzw. 0 in das anstehende Feld.

2.9 Multiple Bänder bzw. Schieberegister, die "im Gleichschritt" laufen, können ein Alphabet, größer als $\{0,1\}$, realisieren *) (k für 2^k Zeichen).

Elementarmaschinen sind entweder

2.10 Operationsmaschinen (z.B. wirkt sA als +1 auf Zähler A), die eine Folge-(El.-) Maschine bestimmen, oder

---

*) Lochbänder, die je anstehendes Feld 5 bzw. 8 Bit anzeigten, waren - vor rd. 30 Jahren - sogar in praktischem Gebrauch. Sie konnten allerdings nur "vorwärts" gelesen werden.

2.11 Frage-Maschinen, die k Register simultan prüfen und eine von $\leq 2^k$ Folge-Maschinen bestimmen. Für eine Baukasten-Lösung scheint k=1 vorzuziehen; die verfügbare Hardware spricht für $k \leq 4$.

2.12 Ende der Rechnung kann dadurch angezeigt sein, daß der ablesende Beobachter als Folge-Maschine - z.B. akustisch - aufgerufen wird. Für theoretische Lösungen auch: Eine Frage-Maschine, die sich selbst als Folge-Maschine aufruft (s dynam. Stop-Problem: muß von außen erkennbar sein.)

2.13 Eine erweiterte Elementarmaschine (Frage) beschreibt nach einem Prüf-Prozeß je Register höchstens eine Operation, bevor eine weitere solche Maschine aufgerufen wird.

3. Beispiele

3.1 Addition mit Zählern A,B und Summe →A.

a→A	B ≠	pB	sA
b→B	0	halt ; lies A	

Variante mit 2.13

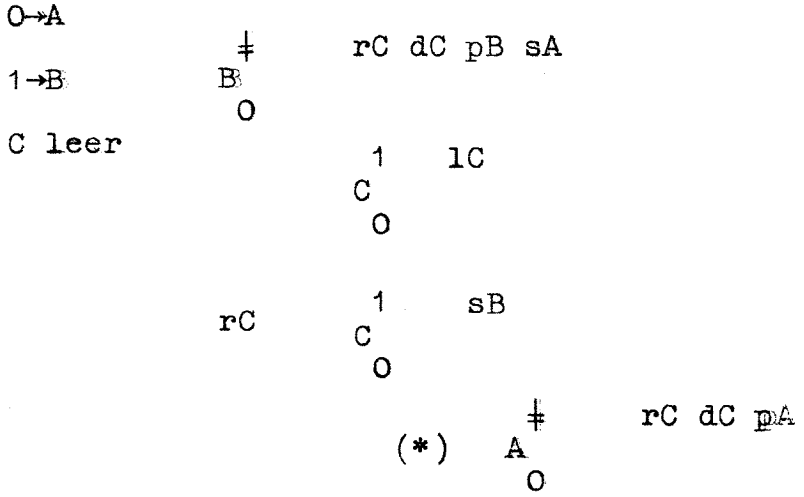
a→A	B ≠	pB	sA
b→B	0	halt	lies A

Im Hinblick auf eine Programmalgebra, die auch bedingte Sprünge linear codiert (s.) steht " —" bzw. lineare Reihung für das Fortschreiten nach rechts.

3.2 Spezielle Fibonacci-Folge

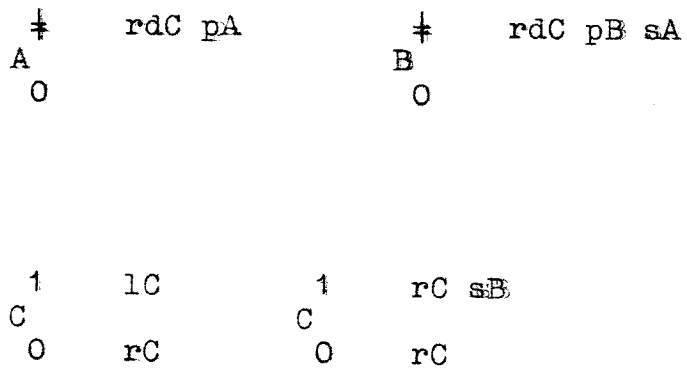
$a_0=0,\ a_1=1,\ a_{n+2}=a_n+a_{n+1}$

Zähler A, B ; Band C, darauf erscheint Resultat "längen-codiert".

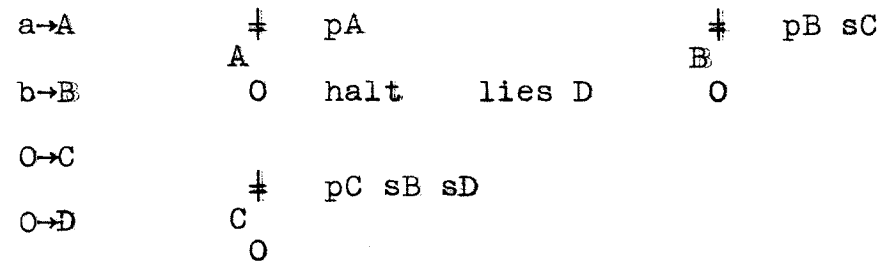


Die "Hauptschleife" beginnt bei (*) und ersetzt (a_n, a_{n+1}) durch (a_{n+1}, a_{n+2}) in (A, B), wobei C als Zwischenspeicher dient. Mit rC dC als einer Operation rdC bietet sich eine Variante mit erweiterten Elementarmaschinen (2.13) an.

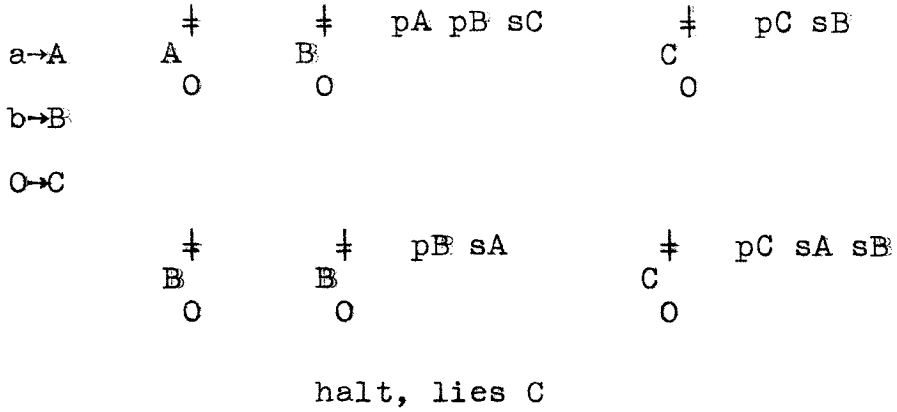
0→A, 1→B, C leer



3.3 Multiplikation, mit Zählern A, B, C, D; Produkt →D
(unmittelbar mit erweiterten Elementarmaschinen)



3.4 Größter gemeinsamer Teiler, mit Zählern A, B, C; Wert →C
(a, b≠0 wird vorausgesetzt)



Die Hauptschleife ersetzt jeweils das Paar (a,b) durch (b, Rest)

3.5 Nach Minsky (19) kann jede berechenbare 2-stellige Funktion mit ≤ 4 Zählern berechnet werden (allgemein: n-stellig mit n+2 Zählern). Die dem Beweis zugrunde liegende Methode ist schon für die Potenz wesentlich aufwendiger als etwa die Einführung eines zusätzlichen Zählers im Anschluß an 3.3.

4. Universelle Maschinen

Eine Maschine ist universell, wenn sie so programmiert ist, daß sie beliebige spezielle Flußdiagramme, die codiert in einem Register stehenm lesen und ausführen kann (im Sinne von 3.5, ggf. für alle Funktionen fester Stellenzahl).

4.1 Die Methode, alle (z.B.) 1-stelligen berechenbaren Funktionen zu einer 2-stelligen (partiellen) *) Funktion dadurch zusammenzufassen, daß man eine arithmetische Codierung (Gödelisierung) des jeweiligen Flußdiagramms als zusätzliches Argument einführt, führt auf recht aufwendige Leseprogramme **) (Turing, vgl. Hermes [ ], S.). Moore (19) hat vorgeschlagen, Programme für das Rechnen nach Turing auf einem Band unter Verwendung von zwei weiteren Bändern zu codieren.

4.2 Die durch Lochbänder nahegelegte, auch von Wang (19) untersuchte Möglichkeit, ohne die Operation c auszukommen, ist zwar aufwendig in Bezug auf Zeit und Band, erlaubt aber sehr einfache Leseprogramme. Begnügt man sich mit dem dynamischen Stop (2.12), so reichen acht erweiterte Elementarmaschinen (2.13) aus zur Aufstellung eines Leseprogramms für den folgenden Code

---

*) Jede Codierung bezieht auf eine beweisbar unvermeidliche Art auch unbrauchbare Flußdiagramme ein.

**) Modell eines Alptraums: ein solches Programm schreiben und den Ablauf beobachten zu müssen.

Rechenband A, Programmband B, Zähler C

$\{Op_1, Op_2, Op_3\} = \{rA, lA, dA\}$

Code auf B	für	B kann als zyklisches Schieberegister
1	Op_1	vorgestellt werden (besser: Lochband),
01	Op_2	auf dem die Befehle fortlaufend stehen.
001	Op_3	
000 0^n1		bei [A]=1: geh n Befehle zurück; sonst: lies den nächsten Befehl. (Also z.B. 0001 meint: lies den selben Befehl noch mal)

In den folgenden erweiterten Elementarmaschinen steht die mittler Zeile für Operationen, die sowohl bei der ersten, als auch bei der zweiten Beobachtung auszuführen sind.

1 Op_1	1 Op_2	1 Op_3	1
B rB	B rB	B rB	A
0	0	0	0

1	1 lB	≠ pC	1
B rB	B sC	C	B lB
0	0 rB	0 1B	0

Zu Beginn der Rechnung stehen auf A , in geeigneter Codierung, die Argumente, das Feld dahinter "steht an", B steht auf dem Anfang des 1. Befehls, und ⟨C⟩=0.

4.3 Variante

Mit dem Code auf B: 11 für lA, 10 für rA, 01 für dA, 000^n1 für Rücksprung um n Einser auf B bei [A]=1; läßt sich sogar ein Flußdiagramm aus zwei (erweiterten) Elementarmaschinen angeben, welche A, B, C im Sinne von 2.11 simultan lesen und im Sinne von 2.13 erweitert sind.