Maschine M

Code	Instruktion	Semantik
0	move-dd a_1 a_2	$Z(a_2) := Z(a_1); PC := PC + 3$
1	move-id a_1 a_2	$Z(a_2) := Z(Z(a_1)); PC := PC + 3$
2	move-di a_1 a_2	$Z(Z(a_2)) := Z(a_1); PC := PC + 3$
3	move-ii a_1 a_2	$Z(Z(a_2)) := Z(Z(a_1)); PC := PC + 3$
4	store n a	$Z(a) := n; PC := PC + 3$
5	jump-d a	$PC := a$
6	jump-i a	$PC := Z(a)$
7	branch=0? a_1 a_2	$PC := \begin{cases} a_2 & \text{falls } Z(a_1) = 0 \\ PC + 3 & \text{sonst} \end{cases}$
8	branch<0? a_1 a_2	$PC := \begin{cases} a_2 & \text{falls } Z(a_1) < 0 \\ PC + 3 & \text{sonst} \end{cases}$
9	add a_1 a_2 a_3	$Z(a_3) := Z(a_1) + Z(a_2); PC := PC + 4$
10	sub a_1 a_2 a_3	$Z(a_3) := Z(a_1) - Z(a_2); PC := PC + 4$
11	mul a_1 a_2 a_3	$Z(a_3) := Z(a_1) * Z(a_2); PC := PC + 4$
12	halt	Hält die Maschine an.

Adressen: $a \in$ Adr $:= \{0, \ldots,$ max-adr$\}$

Darstellbare Zahlen: $n \in$ Num $:= \{$min-int$, \ldots,$ max-int$\}$

Adr $\subseteq$ Num und $\{0, 1, \ldots, 12\} \subseteq$ Num

Speicherzustand: Z : Adr $\longrightarrow$ Num

Befehlszähler: $PC \in$ Adr

Maschinenzustand: (Z, PC)

Fehler: Maschine hält bei Fehler an mit (Z, PC) wie vor Fehler

Mögliche Fehler: Adreßfehler, Arithmetiküberlauf, Instruktionsfehler

Transitionen: $(Z, PC) \rightarrow (Z', PC')$ durch Ausführung der an $Z(PC)$ beginnenden Instruktion