Wertetabellen

Ein einfaches Mittel um die Funktionsweise von Algorithmen, aber auch Methoden oder kleine Programme nachzuvollziehen sind Wertetabellen. Dabei wird der Algorithmus schrittweise durchlaufen und jede Änderung an einem Wert, der im Algorithmus verwendet wird in einer Tabelle protokolliert. Diese Tabelle enthält einmal den Namen des Wertes und dessen verschiedene Zustände. Dabei ist es unerheblich, ob die Werte in Spalten oder Zeilen aufgeschrieben werden. Im weiteren Verlauf wird mit einer spaltenweisen Darstellung gearbeitet.

Lineare Algorithmen

Bei der Darstellung unterscheidet man aber, ob der zu untersuchende Algorithmus einen linearen Ablauf hat oder er eine Wiederholung bzw. einen rekursiven Aufruf enthält. Beim linearen Ablauf wird der Algorithmus, je nach Schreibweise, Zeile für Zeile bzw. Satz für Satz durchgearbeitet. Jede Zeile bzw. jeder Satz erhält dann eine eigene Zeile in der Tabelle. Wird bei dem einem Wert eine Änderung vorgenommen, so wird dieses in der entsprechenden Zeile angegeben, auch wenn der Zustand des Wertes dem vorherigen entspricht. Werden im Algorithmus durch eine Bedingung bestimmte Zeilen nicht beachtet, dann werden diese auch nicht in der Tabelle aufgeführt. Diese Sachen sind auch im folgenden Beispiel dargestellt.

```
1 a <- 4
2 b <- -7
3 c <- a * b
4 wenn c < 0:
5     c <- c * -1
6     d <- 'ja'
7 sonst:
8     d <- 'nein'
9 a <- a * a
```

Zeile	a	b	c	d
1	4			
2		-7		
3			-28	
5			28	
6				ja
9	16			

Abbildung 1: Linearer Algorithmus

Iterative und rekursive Algorithmen

Bei iterativen Algorithmen (von lat. iterare = wiederholen) oder bei rekursiver Algorithmus wird aus Platzgründen nur für jede Wiederholung bzw. jeden rekursiven Aufruf eine eigene Zeile genommen. Dazu kommen ggf. jeweils eine Zeile für die Element vor oder nach der Wiederholung. Auch hier gilt, dass bei einem Wert nur dann etwas eingetragen

wird, wenn der Zustand neu gesetzt wird. Kommt es dazu, dass ein Wert in einer einzelnen Wiederholung mehrfach seinen Zustand ändert, so wird in der entsprechenden Zeile der erste Zustand so durchgestrichen, dass er noch erkennbar ist und der neue Zustand daneben geschrieben. Dieses Vorgehen kann man am Beispiel mit einer Wiederholung nachvollziehen:

```
1  a <- 5
2  b <- 6
3  e <- 0
4  z <- 0
5  i <- 0
6  solange i < b:
7      e <- e + a
8      wenn e > 8:
9          z <- z + 1
10         e <- e - 8
11     i <- i + 1
```

Wiederh.	a	b	e	z	i
vorher	5	6	0	0	0
1			5		1
2			~~10~~ 2	1	2
3			7		3
4			~~12~~ 4	2	4
5			~~9~~ 1	3	5
6			6		6

Abbildung 2: Algorithmus mit Wiederholung