Zahlenreihen mit Computern lösen

In gebräuchlichen Intelligenztests, wie auch im Mensa-Aufnahmetest, kommen häufig Zahlenreihen vor, die man vervollständigen muss. Beispielsweise 3, 5, 8, 12 - wie lautet die nächste Zahl? Wenn es ein Zeichen von Intelligenz ist, wenn Menschen diese Aufgaben lösen können, stellt sich die Frage, ob es sich um eine spezifisch menschliche Eigenschaft handelt. Oder können auch Computer Zahlenreihen lösen? Die Antwort ist: ja, sie können es, wenn sie entsprechend programmiert worden sind. Im folgenden Artikel möchte ich zeigen, wie es geht.

Beginnen wir mit ganz kurzen Zahlenreihen, wo nur zwei Zahlen vorgegeben sind. 1, 2 - wie lautet die nächste in der Reihe? Es gibt zwei Möglichkeiten: 3 oder 4. Vorerst möchten wir uns auf additive Zahlenreihen beschränken; in diesem Fall wäre die 3 richtig. Wenn die Zahlenreihe allgemein a, b lautet, wie kommt man dann auf c? Man rechnet c = b + (b - a) = 2b - a. Das wäre eine allgemeine Lösungformel für additive Zahlenreihen, die nur aus zwei Zahlen bestehen. Wenn wir nun drei Zahlen haben, a, b, c, wie berechnen wir dann die vierte Zahl d? Beispiel: 1, 2, 3. Nun, in diesem Fall gibt es zwei Möglichkeiten: d = c + (c - b) = 2c - b oder d = c + (b - a). Beide führen zum selben Ergebnis. Das gilt aber nur, weil die Differenz c - b gleich b - a ist. Was wäre, wenn sie verschieden wären? Beispiel: 1, 2, 4. Da wir davon ausgehen, dass es sich um eine additive Zahlenreihe handelt, müsste das Ergebnis 7 lauten (2 - 1 = 1, 4 - 2 = 2, also müssen wir 4 + 3 = 7 rechnen). Symbolisch haben wir also gerechnet: d = c + (c - b) + ((c - b) - (b - a)) = 3(c - b) + a. Dies ist die allgemeine Formel für additive Zahlenreihen, die aus drei Zahlen bestehen. Damit können wir jede Zahlenreihe dieser Art lösen.

Für Zahlenreihen, die aus vier oder mehr Zahlen bestehen, lassen sich analoge Formeln berechnen. Um das Ganze etwas anschaulicher zu gestalten, legen wir folgende Notation fest: b‘ = b - a; c‘ = c - b; usw., weiters: c“ = c‘ - b‘, d“ = d‘ - c‘ usw. Bei einer Zahlenreihe, die aus vier Zahlen besteht, ergibt sich dann als Lösungsformel: e = d + d‘ + d“ + d“‘.

Wenn man einen Computer programmieren will, damit er in der Lage ist, additive Zahlenreihen selbsttätig zu lösen, gibt es zwei Möglichkeiten: Entweder man berechnet möglichst einfache Formeln für Zahlenreihen aus zwei, drei, vier,... Zahlen. Dann genügt es, in diese Formeln einzusetzen. Das macht die Berechnung sehr effizient. Oder man programmiert den Computer so, damit er in der Lage ist, Zahlenreihen zu lösen, die aus einer beliebigen Anzahl von Zahlen bestehen. In diesem Fall speichert man die letzte Zahl (z. B. f) ab, berechnet dann b‘ bis f‘ und addiert f‘ zum Ergebnis, berechnet weiters b“ bis f“ usw. Da man die Zahlen der aktuellen Generation überschreiben kann (sobald man f zwischengespeichert hat, kann man f mit f‘ überschreiben, danach e mit e‘ usw.), entsteht kein zusätzlicher Speicherbedarf, doch dauert die Berechnung eben länger. Die Anzahl der Subtraktionen, die durchgeführt werden müssen, beträgt (wenn n die Anzahl der vorgegebenen Zahlen ist) n(n - 1)/2, in der Sprache der Informatiker also $O(n^2)$. Das ist nicht gerade effizient, aber dafür erlaubt dieser Ansatz, mit beliebig langen Zahlenreihen zu arbeiten.

Dieser zweite Ansatz ist aber auch hilfreich, wenn wir unser Programm ergänzen wollen, dass es auch mit Zahlenreihen zurecht kommt, die multiplikative Elemente enthalten. Wenn man 1, 2, 4 als multiplikative Zahlenreihe behandelt, ist die Lösung nicht 7, sondern 8. Darauf kommt man leicht, indem man genauso wie vorhin verfährt, aber nicht Differenzen, sondern Quotienten berechnet und das Ergebnis nicht addiert, sondern multipliziert. Rein multiplikative Zahlenreihen zu lösen, ist also auch kein Problem. Wie löst man aber nun 1, 3, 7 oder 1, 4, 10? Zunächst sollte man den multiplikativen Ansatz versuchen. Da aber 7 nicht ohne Rest durch 3 teilbar ist (und 10 nicht durch 4), muss man additiv vorgehen. Es ergeben sich die Differenzen 2, 4 bzw. 3, 6. Jetzt können wir multiplikativ vorgehen. Es ergibt sich also als Lösungformel für die erste Reihe d = 7 + 4 * 2 = 15 und für die zweite Reihe d = 10 + 6 * 2 = 22. Indem wir bei jedem Schritt zuerst den multiplikativen Ansatz versuchen und, wenn dieser nicht geht, additiv vorgehen, können wir also solche gemischten Zahlenreihen lösen. Selbstverständlich gilt aber Punktrechnung vor Strichrechnung. Wir können nicht, wie bei rein additiven bzw. rein multiplikativen Zahlenreihen, einfach die Differenz c‘ zu c hinzuaddieren, sondern wir müssen die Differenz c‘ zuerst mit dem Quotienten c“ multiplizieren. Das kann man realisieren, indem man mit einem primitiven Stack arbeitet. Man speichert in der Variablen r1 den Wert von c zwischen, dann in r2 den Wert der Differenz c‘. Würde im nächsten Schritt wieder der additive Ansatz erfolgen, könnte man den Wert von r2 zu r1 dazuaddieren und c“ in r2 speichern. Da aber nun multiplikativ vorgegangen wird, muss r2 mit c“ multipliziert werden und erst im nächsten Schritt r2 zu r1 addiert werden.

Man sieht also, dass manches von dem, was der Mensch für spezifisch menschliche Talente hielt, auch von Computern bewerkstelligt werden kann - aber es bedarf immer noch eines klugen Menschen, der in der Lage ist, den Computer entsprechend zu programmieren.

Claus D. Volko, cdvolko@gmail.com