Aufgabe: Rechenrätsel

Bei dieser Aufgabe soll für eine gegebene Anzahl n von Operatoren (also $n+1$ Ziffern) eine Folge $a_0, a_1, a_2, \dots, a_n$, $a_i \in \{0,1,2,\dots,9\}$ und eine positive ganze Zahl b gefunden werden, sodass das Rätsel

$$a_0 \circ_1 a_1 \circ_2 \cdots \circ_n a_n = b$$

genau eine Lösung hat, wobei jedes $\circ_i$ durch ein Element der Menge $O := \{+,-,\cdot,:\}$ ersetzt werden soll.

2.1 Lösungsidee

Da b groß werden kann und es viele Möglichkeiten gibt b auszuwählen, bietet es sich an, a_0 bis a_n zu fixieren und für $b = 1,2,3,\dots$ zu prüfen, ob das so entstandene Rätsel genau eine Lösung besitzt oder nicht. Für eine fixe Folge der a_i können wir dazu die Multimenge M der erreichbaren Zahlen b betrachten, in der jedes b so oft vorkommt, wie es mögliche Belegungen der $\circ_i$ mit Elementen aus O gibt. Haben wir M gefunden, können wir mit linearer Suche prüfen, welche Zahlen dort genau einmal vorkommen. Die Folge der a_i soll dabei vorher schon zufällig (mit bestimmten Einschränkungen, siehe unten) bestimmt werden.

Dass dies funktioniert, kann veranschaulicht werden, wenn man sich beim Beispielrätsel

$$4 \circ 3 \circ 2 \circ 6 \circ 3 \circ 9 \circ 7 \circ 8 \circ 2 \circ 9 \circ 4 \circ 4 \circ 6 \circ 4 \circ 4 \circ 5$$

die Liste der kleinsten eindeutig darstellbaren positiven ganzen Zahlen anschaut:

$$4792, 5118, 5136, 5138, 5142, 5152, 5156, 5163, 5167, 5168, 5194, 5200, 5213, 5215$$

Schaut man sich andere Rätsel an, ergeben sich ähnliche Listen. Dies liegt daran, dass die kleinen Zahlen sich fast immer auf mehrere Arten darstellen lassen. Aber ab 15 Operatoren gibt es zum Beispiel ca. 4000 bis 8000 viele Lösungen für b, die auch schnell größer werden. Wünschenswert wären kleine Ergebnisse, damit das Rätsel möglichst interessant wird und einfache Lösungswege über Blöcke von Multiplikationen, um überhaupt die richtige Größenordnung zu erreichen, ausgeschlossen werden. Somit werden wir die kleinste mögliche Lösung für b wählen.

„Uninteressante“ und „Interessante“ Rätsel

In der Aufgabenstellung ist die Bedingung genannt, dass die Rätsel „interessant und unterschiedlich“ sein sollen. Dies schließt etwa aus, bewusst $b = \prod_{i=0}^{n} a_i$ zu wählen. Für $n \geq 2$ und $a_i \geq 2$ lässt sich nämlich mit einem Größenargument beweisen, dass $\circ_1 = \circ_2 = \cdots = \circ_n = \cdot$ dann die einzige Lösung wäre.

Es gibt auch Rätsel, bei denen es aus einem *zahlentheoretischen* Grund, also etwa wegen Teilbarkeits- oder Restklassenbetrachtungen, genau eine Lösung gibt. Ein einfaches Beispiel hierfür ist $9 \circ_1 9 \circ_2 3 \circ_3 1 = 83$. Eine Größenbetrachtung schließt $\circ_1 = :$ aus, durch eine Betrachtung modulo 3 sieht man ein, dass $\circ_3$ durch $-$ ersetzt werden soll, mit modulo 9, dass $\circ_2$ durch $+$ ersetzt werden soll. Dies ergibt die einzig mögliche Lösung $9 \cdot 9 + 3 - 1 = 83$.

Durch eine Kombination dieser beiden Ansätze können sehr effizient sehr große Rätsel produziert werden, für die die Lösung allerdings ebenfalls schnell in linearer oder höchstens quadratischer Zeit durch sukzessive Größen- und Modulobetrachtungen gefunden werden kann.

Stattdessen wollen wir Rätsel erzeugen, für deren Lösung es eher Brute-Force-ähnlicher Ansätze bedarf. Dafür gehen wir davon aus, dass solche Rätsel am ehesten erstellt werden können, indem das Programm, das die Rätsel erstellt, ebenfalls nur Brute-Force-ähnliche Ansätze verwendet, auch wenn dabei durchaus Optimierungen möglich sind (die wir auch umsetzen werden).

Grundlegender Ansatz

Wir können einfach jede mögliche Belegung der $\circ_i$ testen und jeweils den Wert von b berechnen, den man dadurch erhält. Insgesamt wird so die Multimenge M aller möglichen Rechenergebnisse berechnet. Dabei müssen alle $\mathcal{O}(4^n)$ Belegungen ausprobiert und jedes Mal das Ergebnis berechnet werden, was $\mathcal{O}(n)$ Zeit in Anspruch nimmt. Werden die möglichen Ergebnisse in einer Multimenge verwaltet, kann diese bis zu 4^n Elemente haben. Das Einfügen einer Zahl in diese Multimenge benötigt Zeit $\mathcal{O}(\log(4^n)) = \mathcal{O}(n)$. Mit jeder Belegung muss also ein Ergebnis berechnet und einmal in die Multimenge eingefügt werden, woraus sich eine Laufzeit von $\mathcal{O}(4^n)\,(\mathcal{O}(n)+\mathcal{O}(n)) = \mathcal{O}(4^n \cdot n)$ ergibt.

Verbesserter Ansatz mit Dynamischer Programmierung

Das Hauptproblem beim grundlegenden Ansatz ist, dass oft der gleiche Wert für b gefunden wird. Es ist egal, ob man eine Zahl b mit 2 oder 2000 verschiedenen Belegungen der $\circ_i$ erhält. Kann man zum Beispiel mit den ersten k ($k < n$) Operatoren $\circ_1$ bis $\circ_k$ ein Zwischenergebnis schon auf zwei verschiedene Arten darstellen, muss mit einem dritten gleichen Zwischenergebnis nicht weiter gerechnet werden. Es ist nur die Information wichtig, dass man mit diesem Zwischenergebnis keine eindeutige Lösung erhält. Anders ausgedrückt: es muss nicht mit jeder Belegung von $\circ_1$ bis $\circ_k$, die zum gleichen Zwischenergebnis b_k führt, separat weiter gerechnet werden, da die gleichen Rechnungen ausgeführt werden.

Weiterhin kommt es vor, dass die gleichen Zwischenergebnisse immer wieder berechnet werden. Zum Beispiel das Produkt zweier benachbarter Zahlen.

Um diese beiden Probleme zu beheben, wird ein verbesserter Ansatz mit dynamischer Programmierung eingeführt.

Die Berechnung von Zwischenergebnissen soll in der gleichen Reihenfolge wie die tatsächliche Berechnung von b erfolgen, also Punkt vor Strich. Sei also zunächst $P_{i,j}$ die Multimenge der positiven ganzen Zahlen b, die

$$a_{i-1} \circ_i a_i \circ_{i+1} \cdots \circ_j a_j$$

als Wert annehmen kann, wenn $\circ_i$ bis $\circ_j$ durch $\cdot$ oder $:$ ersetzt werden. Offenbar gilt $P_{i+1,i} = \{a_i\}$ für $i = 0, 1, \ldots, n$. Ferner kann $P_{i,j}$ aus $P_{i,j-1}$ mithilfe der Rekursion

$$P_{i,j} = \left\{xa_j \,|\, x \in P_{i,j-1}\right\} \uplus \left(\left\{\frac{x}{a_j} \,|\, x \in P_{i,j-1}\right\} \cap \mathbb{Z}\right)$$

für $1 \leq i \leq j \leq n$ berechnet werden.[8] Ganz analog lässt sich S_i, die Multimenge der möglichen Summen/Differenzen von Produkten berechnen. Formal sei S_i die Multimenge der ganzen Zahlen b, die

$$a_0 \circ_1 a_1 \circ_2 \cdots \circ_i a_i$$

als Wert annehmen kann, wenn $\circ_1$ bis $\circ_i$ jeweils durch ein beliebiges Element von O ersetzt werden. Die Berechnung von S_j aus S_0 bis S_{j-1} kann also als Vereinigung der Verknüpfungen von jeweils S_{i-1} mit $P_{i,j}$ erfolgen, wobei i die Werte 1 bis j annimmt. In Formeln ausgedrückt wird also $S_0 = \{a_0\}$ und dann

$$S_j = P_{0,j} \uplus \left(\biguplus_{i=1}^{j} \left(\{s+p \,|\, s \in S_{i-1},\, p \in P_{i,j}\} \uplus \{s-p \,|\, s \in S_{i-1},\, p \in P_{i,j}\} \right) \right)$$

für $j = 1, 2, 3, \ldots$ berechnet. So erhält man zuletzt die gesuchte Multimenge S_n, in der nach dem kleinsten positiven Element gesucht werden kann, das dort genau einmal vorkommt. Am kleinen Beispiel aus der Aufgabenstellung

$$4 \circ 4 \circ 3 = 13$$

kann dies so verdeutlicht werden:

j	0	1	2
$P_{0,j}$	$\{4\}$	$\{1,16\}$	$\{3,48\}$
$P_{1,j}$		$\{4\}$	$\{12\}$
$P_{2,j}$			$\{3\}$
S_j	$\{4\}$	$\{0,1,8,16\}$	$\{-8,-3,-2,3,3,4,5,11,13,16,19,48\}$

Man sieht, dass 13 zum Beispiel eindeutig darstellbar ist.

Das Problem, dass Zwischenergebnisse mehrfach berechnet werden, kann mit diesem Ansatz auch überwunden werden. Identische Elemente in S_i bzw. $P_{i,j}$ erzeugen immer identische Ergebnisse für S_n. Also kann jede Zahl, die mehr als zweimal in $P_{i,j}$ bzw. S_i vorkommt, so weit gestrichen werden, bis sie dort nur zweimal vorhanden ist, bzw. nur eingefügt werden, wenn dieses Element bisher nur höchstens einmal enthalten ist.

Mit diesem Ansatz wird auch im schlimmsten Fall keine schlechtere Laufzeit als mit dem grundlegenden Ansatz erzielt. Die Multimenge S_k hat höchstens 4^k Elemente, S_0 bis S_n haben also insgesamt höchstens

$$\sum_{i=0}^{n} 4^i = \frac{4^{n+1}-1}{3} \in \mathcal{O}(4^n)$$

Elemente, mit $\mathcal{O}(n)$ Zeit für das Einfügen haben wir hier eine Laufzeit von höchstens $\mathcal{O}(4^n \cdot n)$ und somit eine Gesamtlaufzeit von $\mathcal{O}(2^n \cdot n^3 + 4^n \cdot n) = \mathcal{O}(4^n \cdot n)$. Anders gesagt: Es wird nie mehr als beim grundlegenden Ansatz berechnet, aber es kann bedeutsame Einsparungen geben, da im Vergleich zu den (Zwischen-)Ergebnissen sehr kleine Ziffern und viele Dopplungen gestrichen und Berechnungen gespart werden können.

[8] Wir benutzen $\uplus$ als Symbol für die Vereinigung von Multimengen.

Wahl der a_i

Bisher haben wir nichts über die Wahl der Werte der a_i gesagt und diese beliebig gewählt, laut Aufgabenstellung aus der Menge $\{0,1,2,\dots,9\}$. Um das Rätsel jedoch *interessant* zu halten, wird auf die Ziffern 0 und 1 verzichtet. Durch 0 darf man überhaupt nicht teilen und bei diesen Ziffern entstehen schnell mehrere Lösungen für viele kleine Ergebnisse. So ist es etwa egal, ob 0 addiert oder subtrahiert, 1 multipliziert oder dividiert und ob ein Produkt oder ein Quotient mit 0 multipliziert wird.

Es ist auch möglich, ein wenig Rechenzeit zu sparen, indem bevorzugt kleine a_i oder aufeinanderfolgende Zahlen möglichst teilerfremd gewählt werden. Hier werden die a_i unabhängig voneinander mit folgenden empirisch bestimmten Wahrscheinlichkeiten gewählt:

Wert	2	3	4	5	6	7	8	9
Wahrscheinlichkeit	27,0%	21,6%	16,2%	10,8%	8,1%	6,5%	5,4%	4,3%

Im Vergleich zur gleichverteilten Wahl läuft so das Programm bei ca. 20 Operatoren ungefähr doppelt so schnell und liefert wesentlich kleinere Rätselergebnisse. Nachteil wiederum ist, dass die mit höherer Wahrscheinlichkeit gewählten Ziffern entsprechend häufig in den Rätseln auftauchen, wodurch diese als weniger „interessant“ angesehen werden können.

2.2 Beispiele

Zum Beispiel werden diese Rätsel gefunden:

- 3 Operatoren: $2 \circ 3 \circ 2 \circ 6 = 6$, gefunden in 0,00 s
- 5 Operatoren: $4 \circ 5 \circ 4 \circ 7 \circ 2 \circ 5 = 11$, gefunden in 0,00 s
- 8 Operatoren: $4 \circ 3 \circ 2 \circ 4 \circ 3 \circ 3 \circ 2 \circ 3 \circ 3 = 110$, gefunden in 0,00 s
- 10 Operatoren: $4 \circ 9 \circ 3 \circ 7 \circ 7 \circ 3 \circ 2 \circ 3 \circ 2 \circ 3 \circ 8 = 353$, gefunden in 0,00 s
- 12 Operatoren: $2 \circ 3 \circ 2 \circ 9 \circ 8 \circ 5 \circ 9 \circ 7 \circ 2 \circ 7 \circ 8 \circ 3 \circ 8 = 1521$, gefunden in 0,02 s
- 15 Operatoren: $6 \circ 5 \circ 4 \circ 7 \circ 6 \circ 9 \circ 3 \circ 3 \circ 3 \circ 3 \circ 7 \circ 8 \circ 2 \circ 3 \circ 6 \circ 3 = 6295$, gefunden in 0,19 s
- 18 Operatoren: $4 \circ 3 \circ 4 \circ 2 \circ 8 \circ 3 \circ 6 \circ 3 \circ 4 \circ 3 \circ 8 \circ 3 \circ 7 \circ 7 \circ 2 \circ 3 \circ 2 \circ 7 \circ 4 = 17464$, gefunden in 1,17 s
- 20 Operatoren: $2 \circ 7 \circ 6 \circ 9 \circ 2 \circ 3 \circ 2 \circ 2 \circ 2 \circ 3 \circ 2 \circ 3 \circ 8 \circ 3 \circ 6 \circ 3 \circ 2 \circ 7 \circ 4 \circ 4 \circ 2 = 19330$, gefunden in 1,67 s

2.3 Bewertungskriterien

Die aufgabenspezifischen Bewertungskriterien werden hier erläutert.

1. Lösungsweg

(1) *Problem adäquat modelliert*: Es sind alle internen Darstellungen von Rätseln akzeptabel, die das Generieren nicht unnötig verkomplizieren. Die Verwendung von Strings ist eher keine gute Idee.

(2) *Laufzeit des Verfahrens in Ordnung*: Die asymptotische Laufzeit des Programms, sofern analytisch charakterisierbar, sollte die des grundlegenden Ansatzes keinesfalls überschreiten. Verbesserungen (oder auch alternative Verfahren) sollten ermöglichen, dass auch für 15 oder sogar mehr Operanden in wenigen Sekunden eine Lösung gefunden wird. Verfahren mit besonders kurzen Laufzeiten können mit Pluspunkten belohnt werden, sofern sie nicht erheblich leichter zu lösende Rätsel generieren als aufwändigere Verfahren. Verfahren mit polynomiellen Laufzeiten, die zu leichten Rätseln führen, sind dieses Mal hingegen nicht akzeptabel. Wenn das gewählte Verfahren im Prinzip dem grundlegenden Ansatz entspricht, werden 4 Punkte abgezogen.

(3) *Verfahren nicht unnötig ineffizient*: Wird zuerst ein Rätsel inklusive Operatoren erzeugt und dann durch Probieren aller anderen Belegungen die Eindeutigkeit überprüft, werden 2 Punkte abgezogen. Es werden Verbesserungen erwartet, insbesondere das Vermeiden mehrfacher Berechnungen.

(4) *Verfahren mit korrekten Ergebnissen*: Die Bedingungen der Aufgabenstellung (insbesondere Punkt vor Strich ohne Klammern, Grundrechenarten, ganzzahlige Zwischenergebnisse, eindeutiges nicht-negativ ganzzahliges Endergebnis) sollten richtig verstanden und umgesetzt werden. Eindeutig heißt hier, dass es nur eine mögliche Belegung der $\circ$ gibt, die diese Bedingungen erfüllt und zum richtigen Ergebnis führt.
Wenn das Ergebnis nicht eindeutig ist, so werden bis zu 4 Punkte abgezogen. Wenn die Bedingungen so interpretiert wurden, dass auch negative Zwischenergebnisse ausgeschlossen werden, soll es dafür keinen Punktabzug geben.
Sollte eine Interpretation der Bedingungen zu leichten Einschränkungen bei den Ergebnissen führen, so ist dies akzeptabel, wenn es dokumentiert ist.

(5) *Wahl der Operanden und der Lösung geeignet*: Es sollte erkannt werden, dass die Zahlen und Ergebnisse frei gewählt werden können, und dies für die benötigte Variabilität auch getan werden. Es bietet sich an, die a_i frei zu wählen. Wenn bestimmte Zahlen, wie 0 oder 1, von vornherein ausgeschlossen werden, so muss dies begründet werden. Wenn das Ergebnis b so gewählt wird, dass das Rätsel damit schwieriger wird (zum Beispiel wenn kleine b bevorzugt werden) und diese Beobachtung erwähnt wird, gibt es dafür 2 Pluspunkte.

(6) *Rätsel interessant und unterschiedlich*:
- Laut Aufgabenstellung sollten die erzeugten Rätsel „interessant und unterschiedlich" sein. Es ist also zum Beispiel nicht in Ordnung, das Ergebnis des Rätsels einfach als das Produkt der Operanden zu wählen. Kriterien und Bedingungen für die Erstellung „interessanter" Rätsel sollten genannt und begründet werden. Eine zufällige Wahl der Operanden und Operatoren reicht aus, um das Kriterium „interessant" zu befriedigen, dies sollte allerdings begründet werden. Wenn das Kriterium nicht explizit erwähnt bzw. begründet wird, so wird ein Punkt abgezogen.
- Rein deterministische Ansätze, die nur von der geforderten Anzahl an Operatoren ab-

hängen, sind ungeeignet, da *unterschiedliche* Rätsel benötigt werden.

(7) *Mehrere Lösungsansätze*: Wer mehrere und deutlich unterschiedliche Ansätze ausführlich beschrieben oder sogar realisiert und miteinander verglichen hat, hat Pluspunkte verdient.

2. Theoretische Analyse

(1) *Verfahren / Qualität insgesamt gut begründet*: Die Korrektheit des Verfahrens bezüglich der an die Rätsel gestellten Anforderungen sollte möglichst gut begründet werden, insbesondere die eindeutige Lösbarkeit der generierten Rätsel. Falls ein Algorithmus mit exponentieller Laufzeit gewählt wurde, sollten ebenfalls Gründe genannt werden; formale Argumente können Pluspunkte bringen. Bei der Wahl eines schnelleren Ansatzes sollte auf mögliche Nachteile bei der Generierung der Rätsel eingegangen werden.

(2) *Gute Überlegungen zur Laufzeit des Verfahrens*: Die Laufzeit kann bei einigen Optimierungen nur schwer konkret abgeschätzt werden, allerdings wird erwartet, dass man mit einfachen Überlegungen zeigt, warum der Ansatz besser ist als etwa der grundlegende. Auch hier können für formale Argumente Pluspunkte vergeben werden.

(3) *Weitere gute analytische Überlegungen*: Es wird ein Verfahren verlangt, das den grundlegenden Ansatz verbessert. Hierfür sind vielerlei Optimierungen möglich, nicht nur bei der Berechnung, sondern auch bei der Wahl der Operanden. Hier können vor allem empirische Beobachtungen dazu ausgenutzt werden, wann möglichst schnell ein eindeutiges Ergebnis gefunden wird.

3. Dokumentation

(3) *Erzeugte Rätsel dokumentiert*: Zwar gibt es bei dieser Aufgabe keine vorgegebenen Beispiele, aber es sollten natürlich, je nach Ansatz, beispielhaft mindestens sechs generierte Rätsel mit jeweils verschiedener Operatorenanzahl, darunter die Anzahlen 2 und 3, gezeigt werden. Besser sind solche Rätsel, die repräsentativ für Ausgaben des Programms sind. Für eine erläuterte Dokumentation weiterer, die Leistungsfähigkeit des Rätselgenerators aussagekräftig demonstrierender Beispiele kann es Pluspunkte geben.

(4) *Ergebnisse nachvollziehbar dargestellt*: Für alle dokumentierten Beispielrätsel müssen die Lösungen mit angegeben werden.