Lösung zu Aufgabe 1

Bei der Auswertung des Ausdrucks

$$2 \cdot (25 + 3 \cdot (4 + 5)) - 314 - 4 \cdot 63$$

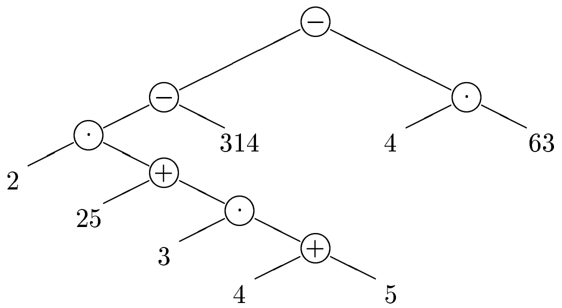
hat man, ohne daß Rechengesetze wie Kommutativität oder Assoziativität zur Umformung der Formel verwendet werden, folgende Wahlmöglichkeit: Man kann entweder im linken Teil der Formel beginnen und zunächst $4 + 5 = 9$, $3 \cdot 9 = 27$, etc. berechnen, oder man kann im rechten Teil der Formel mit der Berechnung von $4 \cdot 63 = 252$ beginnen. In der Tat kann man die Berechnung von $4 \cdot 63 = 252$ zwischen beliebige Schritte der Auswertung des linken Teils der Formel einschalten. Da der linke Teil der Formel aus sechs Operationen besteht, ergeben sich folglich sechs nichtdeterministische (aber determinierte!) Möglichkeiten der Auswertung:

Schritt	1. Möglichkeit	2. Möglichkeit	3. Möglichkeit
1	$4 \cdot 63 = 252$	$4 + 5 = 9$	$4 + 5 = 9$
2	$4 + 5 = 9$	$4 \cdot 63 = 252$	$3 \cdot 9 = 27$
3	$3 \cdot 9 = 27$	$3 \cdot 9 = 27$	$4 \cdot 63 = 252$
4	$25 + 27 = 52$	$25 + 27 = 52$	$25 + 27 = 52$
5	$2 \cdot 52 = 104$	$2 \cdot 52 = 104$	$2 \cdot 52 = 104$
6	$104 - 314 = -210$	$104 - 314 = -210$	$104 - 314 = -210$
7	$-210 - 252 = -462$	$-210 - 252 = -462$	$-210 - 252 = -462$

Schritt	4. Möglichkeit	5. Möglichkeit	6. Möglichkeit
1	$4 + 5 = 9$	$4 + 5 = 9$	$4 + 5 = 9$
2	$3 \cdot 9 = 27$	$3 \cdot 9 = 27$	$3 \cdot 9 = 27$
3	$25 + 27 = 52$	$25 + 27 = 52$	$25 + 27 = 52$
4	$4 \cdot 63 = 252$	$2 \cdot 52 = 104$	$2 \cdot 52 = 104$
5	$2 \cdot 52 = 104$	$4 \cdot 63 = 252$	$104 - 314 = -210$
6	$104 - 314 = -210$	$104 - 314 = -210$	$4 \cdot 63 = 252$
7	$-210 - 252 = -462$	$-210 - 252 = -462$	$-210 - 252 = -462$

In diesen Tabellen wandert die Berechnung von $4 \cdot 63 = 252$ langsam nach unten, wird also, je nach Möglichkeit, im ersten, zweiten, dritten, usw. Schritt ausgewertet. Da die Berechnung in verschiedenen Schritten erfolgt, handelt es sich um verschiedene Auswertemöglichkeiten.

Sehr übersichtlich läßt sich der zu berechnende Ausdruck durch einen Baum darstellen, der in seiner Struktur die Rangfolge der Operatoren wiedergibt (Ein solcher Baum wird z.B. von einem Compiler aufgebaut, wenn ein Ausdruck berechnet werden soll. Man nennt ihn auch Parsebaum, da er beim Einlesen des Quelltextes (engl. *to parse* — grammatisch zergliedern) erzeugt wird.):



Parsebaum des Ausdrucks aus Aufgabe 1. Der Ausdruck im rechten Ast $(4 \cdot 63)$ kann vor jedem Ausdruck im linken Ast (d.h. vor $4 + 5$, $3 \cdot 9$, etc.) ausgewertet werden.

Aus diesem Baum lassen sich die verschiedenen Möglichkeiten der Auswertung besonders leicht ablesen. Die beiden Äste des Baumes sind unabhängig voneinander. Man kann im linken oder im rechten beginnen. Die Berechnung des rechten Astes kann nun vor jeder Auswertung einer Operation des linken Astes und vor der Auswertung der Wurzel geschehen. Man erhält folglich sechs Auswertemöglichkeiten, da der linke Ast, inklusive der Wurzel, sechs Operatorknoten enthält.

Lösung zu Aufgabe 2

Zur Vereinfachung der Beschreibung nennen wir das Entnehmen zweier Süßigkeiten aus dem Glas und das Ausführen einer der beiden Anweisungen abhängig von der Art der entnommenen Süßigkeiten eine Operation, d.h. wir unterscheiden:

1. Zwei Karamelbonbons werden dem Glas entnommen und ein Lakritz wird wieder eingefüllt.
2. Zwei Lakritz werden dem Glas entnommen und ein Lakritz wird wieder eingefüllt.
3. Ein Lakritz und ein Karamelbonbon werden dem Glas entnommen und das Karamelbonbon wird wieder zurückgelegt.

Zur Lösung der Aufgabe betrachten wir die Eigenschaften dieser drei Operationen.

Terminierung

Zu Beginn seien in dem Glas n Süßigkeiten enthalten. Durch jede Operation werden dem Glas zwei Süßigkeiten entnommen und nur eine zurückgelegt. Folglich verringert sich die Zahl der Süßigkeiten im Glas mit jeder Operation um eins. Sind zu Beginn n Süßigkeiten enthalten, so nach einem Schritt $n-1$, nach zwei Schritten $n-2$, usw. Nach $n-1$ Schritten ist nur noch eine Süßigkeit im Glas, womit der Vorgang beendet ist. Folglich terminiert der Algorithmus in $n-1$ Schritten (nach Ausführen von $n-1$ Operationen).

Übrigbleibende Süßigkeit

Wir stellen zunächst fest, daß die Art der übrigbleibenden Süßigkeit nicht unabhängig von den zu Beginn im Glas enthaltenen Süßigkeiten ist, denn (wir betrachten nur zwei sehr einfache Fälle mit $n = 2$): Sind zu Beginn ein Karamelbonbon und ein Lakritz im Glas, so bleibt, da nur Operation 3 ausgeführt werden kann, ein Karamelbonbon übrig. Sind zu Beginn dagegen zwei Karamelbonbons oder zwei Lakritz enthalten, so kann nur Operation 1 oder nur Operation 2 ausgeführt werden und folglich bleibt ein Lakritz übrig. (Natürlich können wir auch, noch einfacher, annehmen, daß $n = 1$ und entweder ein Lakritz oder ein Karamelbonbon im Glas ist. Dann brauchen wir nicht einmal eine Operation auszuführen, um die unterschiedlichen Ergebnisse zu erhalten.)

Dennoch kann man eine allgemeine Aussage über die übrigbleibende Süßigkeit machen. Wir betrachten dazu die Zahl der Karamelbonbons im Glas, die wir mit k bezeichnen. Durch Operation 1 wird k offenbar um zwei verringert, da die beiden entnommenen Karamelbonbons gegessen werden. Durch Operation 2 ändert sich k dagegen nicht, da gar keine Karamelbonbons entnommen werden, und auch Operation 3 läßt k unverändert, da das entnommene Karamelbonbon wieder zurückgelegt wird. Wir stellen also fest, daß die Zahl der Karamelbonbons im Glas entweder gleich bleibt oder sich um zwei verringert. Entscheidend ist, daß sie sich nicht um eins verringern kann, denn dies ermöglicht die folgende Aussage:

> Ist die Zahl der Karamelbonbons zu Beginn gerade, so bleibt ein Lakritz,
> ist sie ungerade, so bleibt ein Karamelbonbon übrig.

Dies sehen wir wie folgt ein: Sei k zu Beginn gerade. Da jede Operation k entweder unverändert läßt oder um zwei verringert, bleibt k stets gerade. Insbesondere kann k nie den Wert 1 annehmen. Dies ist aber eine notwendige Voraussetzung, damit *ein* Karamelbonbon übrig bleiben kann. Folglich muß in diesem Fall ein Lakritz übrig bleiben. Sei k nun zu Beginn ungerade. Da jede Operation k entweder unverändert läßt oder um zwei verringert, bleibt k stets ungerade. Schließlich erreicht k den Wert 1. Dieses eine Karamelbonbon kann aber durch keine der Operationen aus dem Glas entfernt werden, da die einzige Operation, die Karamelbonbons entfernt, die Operation 1, nicht mehr anwendbar ist. Folglich bleibt in diesem Fall ein Karamelbonbon übrig.

Man beachte bei dieser Überlegung, daß stets $k \leq n$ gilt (d.h. die Zahl der Karamelbonbons im Glas nie größer sein kann als die Gesamtzahl der Süßigkeiten im Glas) und in jedem Schritt, wie oben festgestellt, n um eins verringert wird. Folglich muß sich irgendwann auch k verringern und damit entweder den Wert 0 (k zu Beginn gerade) oder den Wert 1 (k zu Beginn ungerade) erreichen.

Lösung zu Aufgabe 3

Bei der Lösung dieser Aufgabe kann man sich z.B. an den Standardfunktionen der Programmiersprache C zur Behandlung von Zeichenketten orientieren. Diese umfaßt u.a. die folgenden Funktionen (definiert in den Header-Dateien `string.h` und `stdlib.h`):

`strcpy`	Copies a string Kopiert eine Zeichenkette (entspricht der Zuweisung bei ganzen Zahlen)
`strlen`	Computes the length of a string Bestimmt die Länge einer Zeichenkette
`strchr`	Scans a string for a character Sucht in einer Zeichenkette nach einem gegebenen Zeichen (Suche von links)
`strrchr`	Locates the last occurrence of a character in a string Sucht in einer Zeichenkette nach dem letzten Auftreten eines gegebenen Zeichens (Suche von rechts)
`strstr`	Locates a substring Sucht nach dem Auftreten einer Zeichenkette in einer anderen Zeichenkette
`strcat`	Concatenates two strings Verbindet zwei Zeichenketten, d.h. hängt eine Zeichenkette an eine andere an
`strcmp`	Compares two strings Vergleicht zwei Zeichenketten
`atoi`, `strtol`	Converts a string to `long int` Wandelt eine Zeichenkette in eine ganze Zahl vom Typ `long int`
`atof`, `strtod`	Converts a string to `double` Wandelt eine Zeichenkette in eine Fließkommazahl vom Typ `double`

Weiter kann man an eine Funktion zum Einfügen einer Zeichenkette in eine andere (an einer bestimmten Position) oder zum Löschen einer Zeichenkette aus einer anderen, eine Funktion zum Spiegeln einer Zeichenkette (Vertauschen des ersten Zeichen gegen das letzte, des zweitens gegen das vorletzte, etc.), eine Funktion zum Rotieren einer Zeichenkette (z.B. nach links: das zweite Zeichen rückt an die Stelle des ersten, das dritte Zeichen an die Stelle des zweiten, usw., das erste Zeichen rückt an die Stelle des letzten) und viele andere Funktionen denken.

Christian Borgelt