PD Dr. U. Köthe *Heidelberg, 27. Oktober 2016*

Übungsblatt 2

Aufgabe 2.1 Uhren [7 Punkte]

Mein Großvater hatte vier Uhren: eine Wanduhr, einen Wecker, eine Küchenuhr und eine Taschenuhr. Eines Tages wollten wir herausfinden, wie genau die Uhren sind. Deshalb haben wir alle vier Uhren exakt auf das Zeitzeichen der 12-Uhr-Nachrichten gestellt. Beim Zeitzeichen der 13-Uhr-Nachrichten ging die Wanduhr bereits 2 Minuten nach. Als die Wanduhr 13 Uhr läutete, zeigte der Wecker schon 13:02. Als es auf dem Wecker 14 Uhr war, zeigte die Küchenuhr erst 13:56. Als die Küchenuhr schließlich 14 Uhr erreichte, war es auf der Taschenuhr bereits 14:04. Ging die Taschenuhr jetzt gegenüber der wahren Zeit vor, nach oder genau? Natürlich dürfen Sie die Antwort nicht raten, sondern müssen einen Lösungsweg angeben und *begründen*.

Tipp: Stellen Sie einen arithmetischen Ausdruck (mit Begründung) auf, an dessen Ergebnis man die Antwort ablesen kann. Ist das Ergebnis des Ausdrucks größer als eins, geht die Taschenuhr vor, ist es kleiner als eins, geht sie nach, und ist es gleich eins, geht sie genau.

Aufgabe 2.2 Arithmetische Ausdrücke [20 Punkte]

(a) Wandeln Sie den folgenden Ausdruck für die Länge eines 3-dimensionalen Vektors

$$\sqrt{x^2 + y^2 + z^2}$$

in die Prefix-Notation um. Die Umwandlung soll in drei Schritten erfolgen: (1) Quadrate durch Multiplikation ausdrücken, (2) fehlende Klammern in die Infix-Notation einfügen, (3) Infix-Notation in Prefix-Notation transformieren. Beachten sie in Schritt (2) die Regeln "Punktrechnung geht vor Strichrechnung" und "Operationen gleicher Wertigkeit werden von links nach rechts ausgewertet", um die Teilausdrücke korrekt zu klammern. Verwenden Sie bei (3) die Funktionsnamen `sqrt` für die Wurzel, `add` für die Addition und `mul` für die Multiplikation (für das Quadrieren steht keine vordefinierte Funktion zur Verfügung).

Transformieren Sie den Ausdruck nun aus der Prefix- in die Baumdarstellung. Berechnen Sie den Wert des Ausdrucks für die Eingabewerte $x = 3$, $y = 4$ und $z = 12$ mit Hilfe des Substitutionsprinzips, d.h. indem Sie in der Baumdarstellung Teilausdrücke von unten nach oben durch ihre Werte ersetzen. Zeichnen Sie die Bäume, die nach den wesentlichen Zwischenschritten entstehen.

(b) Schreiben Sie ein Programm in der simplifizierten Maschinensprache aus der Vorlesung, um den Ausdruck aus (a) mit den gegebenen Eingabewerten zu berechnen. Wir nehmen dabei an, dass die Funktionen `add`, `mul` und `sqrt` von der Hardware bereitgestellt werden. Zur Erinnerung wird die Maschinensprache am Ende der Aufgabe nochmals kurz erklärt.

(c) Betrachten Sie das Polynom $2 + 3x - 4x^2 + 5x^3$, das in Infix-Notation als

```
2 + 3*x - 4*x*x + 5*x*x*x
```

geschrieben werden kann. Aufgrund der Regeln "Punktrechnung geht vor Strichrechnung" und "Operationen gleicher Wertigkeit werden von links nach rechts ausgewertet" konnten hier einige Klammern weggelassen werden. Fügen Sie diese Klammern wieder ein und geben Sie den resultierenden Infix-Ausdruck an. Transformieren Sie den Infix-Ausdruck nun in die Prefix-Notation und in die Baumdarstellung. Benutzen Sie das Substitutionsprinzip, um den Ausdruck für $x = 2$ auszuwerten.

(d) Eine andere Möglichkeit, das Polynom aus (c) zu schreiben, ist das sogenannte Horner-Schema (nach W.G. Horner, der diesen Trick ab 1819 populär gemacht hat):

```
2 + (3 + (-4 + 5*x)*x)*x
```

Beweisen Sie durch Ausmultiplizieren, dass dieser Ausdruck zum ursprünglichen Ausdruck äquivalent ist. Wandeln Sie die Horner-Version ebenfalls in die Prefix-Notation und in die Baumdarstellung um und benutzen Sie das Substitutionsprinzip, um den Ausdruck für $x = 2$ auszuwerten (es sollte das gleiche herauskommen wie zuvor ;-). Warum ist die Benutzung des Horner-Schemas vorteilhaft?

(e) Schreiben Sie das Maschinenprogramm für die Polynom-Version aus (d) mit $x = 2$.

Zur Erinnerung: Die Maschinensprache drückt alle Anweisungen durch Zahlencodes aus, weil Computer nur Zahlen als Symbole akzeptieren. Ist ein Ausdruck in der Baumdarstellung gegeben, wird jedem Knoten des Baumes zunächst eine Speicherzelle zugeordnet. Die Speicherzellen werden durch IDs eindeutig identifiziert (zur Vereinfachung der Korrektur sollen Sie diese IDs so vergeben, dass die Speicherzellen, von 1 beginnend, im Baum von links nach rechts und von oben nach unten durchnummeriert werden). Die Funktionsnamen werden ebenfalls durch Zahlen, die sogenannten *Opcodes*, ersetzt. Wir definieren hier die Opcodes `init=1`, `add=2`, `sub=3`, `mul=4` und `sqrt=6` (es gilt auch `div=5`, aber dies wird für die Aufgabe nicht benötigt).

Die `init`-Anweisung initialisiert eine Speicherzelle mit einem Wert und hat die Struktur

1 <ID Ziel-Speicherzelle> <Wert>

("1" ist der Opcode für `init`). Eine Rechenanweisung hat die Struktur:

<Opcode> <ID SpZ Ergebnis> <ID SpZ Arg1> [<ID SpZ Arg2>]

Dabei drückt die Schreibweise [...] aus, dass dieser Teil des Ausdrucks optional ist – er wird nur dann benötigt, wenn die Funktion zwei Argumente hat. <ID SpZ ...> steht für die Nummer der Speicherzelle, wo der Wert eines Arguments steht bzw. wo der Wert des Ergebnisses abgelegt werden soll.

Beispiel: Das Maschinenprogramm für $x + x$ mit $x = 10$ lautet:

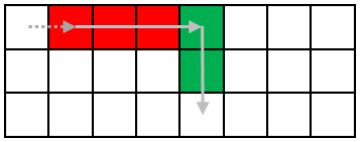
```
1 2 10   # Initialisierung: SpZ2 <- 10
2 1 2 2  # Addition:        SpZ1 <- add(SpZ2, SpZ2)
```

Aufgabe 2.3 Turtle-Graphik [13 Punkte]

Die Turtle-Graphik ist eine einfache Methode, um mit nur drei verschiedenen Kommandos komplexe Graphiken zu erstellen. Man nehme ein Blatt kariertes Papier und stelle sich vor, dass die gewünschte Zeichnung entsteht, indem man eine winzige Schildkröte über das Blatt laufen lässt. Die Kästchen auf dem Papier markieren dabei die Pixel der Zeichnung. Anfangs sitzt die Schildkröte im linken oberen Pixel und schaut nach rechts. Sie kann drei Kommandos ausführen:

1. `walk(N)` Die Schildkröte macht entlang der aktuellen Blickrichtung `N` Schritte. `N` ist eine positive ganze Zahl, und jeder Schritt ist genau ein Pixel lang.
2. `turn(angle)` Die Schildkröte ändert ihre Blickrichtung um den angegebenen Winkel. Mögliche Winkel sind $-90°$ für eine Linksdrehung (gegen den Uhrzeigersinn) und $90°$ für eine Rechtsdrehung.
3. `tail(pen)` Am Schwanz der Schildkröte sind vier Stifte befestigt, ein roter, ein grüner, ein blauer und ein schwarzer. Beim Kommando `tail(red)` senkt die Schildkröte den Schwanz so, dass der rote Stift das Papier berührt. Ebenso macht sie es für die drei anderen Farben. Das Kommando `tail(up)` bewirkt, dass die Schildkröte den Schwanz hebt, so dass kein Stift das Papier berührt.

Macht die Schildkröte einen Schritt, während ein Stift das Papier berührt, wird das Pixel, das die Schildkröte gerade verlassen hat, mit der entsprechenden Farbe gefärbt. Die folgende Zeichnung



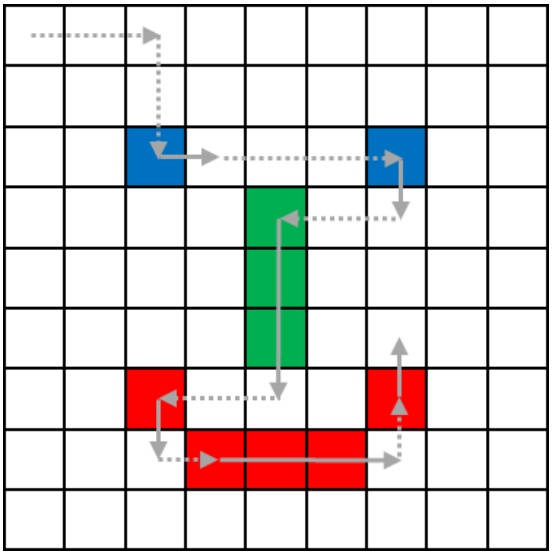
wurde beispielsweise mit der Anweisungsfolge `tail(up); walk(1); tail(red); walk(3); turn(90); tail(green); walk(2)` erstellt. Die Pfeile sind natürlich nicht Bestandteil der Zeichnung, sie symbolisieren vielmehr die Schritte der Turtle: Gestrichelte Pfeile stehen für Schritte ohne Stift, durchgezogene für Schritte mit Stift.

Dummerweise versteht die Turtle nur ihre spezielle Maschinensprache. Wir müssen die Anweisungen deshalb in einen reinen Zahlencode überführen. Für die Kommandos verwenden wir folgende Opcodes: `walk`=1, `turn`=2, `tail`=3. Die Schwanzstellungen werden kodiert als `up`=0, `red`=1, `green`=2, `blue`=3, `black`=4. Schrittzahl und Winkel sind bereits Zahlen und können so bleiben. Das obige Programm lautet in dieser Maschinensprache

```
3 0     # tail(up)
1 1     # walk(1)
3 1     # tail(red)
1 3     # walk(3)
2 90    # turn(90)
3 2     # tail(green)
1 2     # walk(2)
```

Zur besseren Lesbarkeit haben wir die Anweisungen hier untereinander geschrieben. Die Bemerkungen ab dem Zeichen # sind nur als Kommentar für Sie gedacht und werden von der Turtle ignoriert.

(a) Erstellen Sie das Maschinenprogramm, mit dem das folgende Bild gezeichnet wurde:



Geben Sie Ihre Lösung in einem Textfile `turtle.txt` ab. Formatieren Sie das File so, dass pro Zeile eine Anweisung steht (wie im obigen Beispiel, die Kommentare sind dabei optional).

(b) Zeichnen Sie das Bild (8×8 Pixel), das durch folgendes Maschinenprogramm erstellt wird (die Anweisungen sind von oben nach unten auszuführen, die Schildkröte sitzt anfangs im linken oberen Pixel und schaut nach rechts, und die Papierfarbe ist weiß):

```
3 3
1 7
2 90
1 1
2 90
1 3
3 4
1 2
3 3
1 2
2 -90
1 1
2 -90
1 3
3 0
1 2
3 3
1 2
2 90
1 3
```

```
3 2
1 2
2 90
1 7
2 90
1 3
3 3
1 1
2 90
1 3
3 1
1 2
3 3
1 1
2 90
1 1
2 90
1 2
3 0
1 2
```

```
3 3
1 1
2 -90
1 1
3 2
1 1
2 -90
1 2
3 1
1 2
3 2
1 1
2 -90
1 1
2 -90
1 2
3 1
1 2
3 2
1 2
```

Markieren Sie in Ihrem Bild, wo sich die Turtle am Ende befindet. Geben Sie das Ergebnis in einem verbreiteten Bildformat ab (z.B. als JPEG oder PNG aus einem Zeichenprogramm Ihrer Wahl exportiert), oder als abfotografierte Papierzeichnung.

Bitte laden Sie Ihre Lösung spätestens bis 7. November 2016, 11:00 Uhr in Moodle hoch.