**Abteilung für Angewandte Mathematik** 02.07.2018
Prof. Dr. S. Bartels
M.Sc. C. Palus

Übung zur Vorlesung

# Einführung in die Programmierung

SoSe 2018 – Blatt 11 (Bonusblatt)

Abgabe: Briefkästen RZ/E-Mail bis Montag, den 09.07.2018 um 16:00 Uhr

**Aufgabe 1** (4+6 Bonuspunkte)**.** Ein *Alphabet* ist definiert als eine endliche, nichtleere Menge $\mathcal{A}$ von Zeichen. Ein *Wort über* $\mathcal{A}$ ist eine (beliebig lange) Folge von Elementen aus $\mathcal{A}$ und eine *(formale) Sprache über* $\mathcal{A}$ ist eine Menge von Wörtern über $\mathcal{A}$. Die *kleenesche Hülle* $\mathcal{L}^*$ einer Sprache $\mathcal{L}$ ist die Menge aller Wörter, die durch beliebiges Verketten von Wörtern aus $\mathcal{L}$ gebildet werden können, inklusive des *leeren Wortes* $\varepsilon$. *Reguläre Ausdrücke* über $\mathcal{A}$ sind Zeichenketten, mit denen sich formale Sprachen beschreiben lassen. Ihre Syntax ist induktiv wie folgt definiert:

(1) $\emptyset$ und $\varepsilon$ sind reguläre Ausdrücke.
(2) Für alle $z \in \mathcal{A}$ ist $z$ ein regulärerer Ausdruck.
(3) Sind $r$ und $s$ reguläre Ausdrücke, so sind auch die *Alternative* $(r|s)$, die *Verkettung* $(rs)$ und die *kleenesche Hülle* $(r^*)$ reguläre Ausdrücke.

Zu einem regulären Ausdruck $r$ bezeichnen wir die durch ihn beschriebene Sprache mit $\mathcal{L}(r)$. Die Semantik regulärer Ausdrücke lässt sich damit analog zur obigen Syntaxdefinition beschreiben:

(1) $\mathcal{L}(\emptyset) = \emptyset$ und $\mathcal{L}(\varepsilon) = \{\varepsilon\}$, d. h. der leere reguläre Ausdruck definiert die leere Sprache und das leere Wort definiert die Sprache, die nur das leere Wort enthält.
(2) Für alle $z \in \mathcal{A}$ ist $\mathcal{L}(z) = \{z\}$, d. h. jedes Zeichen $z$ definiert die Sprache, die als einziges Wort genau das Wort enthält, welches lediglich aus dem Zeichen $z$ besteht.
(3) 3. Sind $r$ und $s$ reguläre Ausdrücke, so gilt
- $\mathcal{L}(r|s) = \mathcal{L}(r) \cup \mathcal{L}(s)$, d. h. die Alternative definiert die Sprache, die aus der Vereinigung der durch $r$ und $s$ definierten Sprachen entsteht,
- $\mathcal{L}(rs) = \{\alpha\beta : \alpha \in \mathcal{L}(r) \wedge \beta \in \mathcal{L}(s)\}$, d. h. die Verkettung definiert die Sprache, die genau die Wörter enthält, die ein Wort aus der vom ersten Ausdruck beschriebenen Sprache als Anfangsstück haben und deren Reststück ein Wort aus der vom zweiten Ausdruck beschriebenen Sprache ist.
- $\mathcal{L}(r^*) = (\mathcal{L}(r))^*$, d. h. die kleenesche Hülle eines Ausdrucks beschreibt die kleenesche Hülle der durch den Ausdruck beschriebenen Sprache.

Es sei das Alphabet $\mathcal{A}_{\text{Skat}} = \{7, 8, 9, Z, B, D, K, A, \spadesuit, \heartsuit, \diamondsuit, \clubsuit\}$ gegeben, welches sich zur Beschreibung von Spielkarten eignet.

(i) Jede Karte eines Kartenspiels ist eindeutig festgelegt durch ihre Farbe $f \in \{\spadesuit, \heartsuit, \diamondsuit, \clubsuit\}$ und ihren Wert $w \in \{7, 8, 9, Z, B, D, K, A\}$. Geben Sie reguläre Ausdrücke an, welche die folgenden Sprachen definieren:

(a) Die Sprache, die genau jede Spielkarte enthält.
(b) Die Sprache, die genau jede Folge von 10 (evtl. mehrfach darin vorkommenden) Spielkarten enthält.
(c) Die Sprache, die jede Folge von 10 Spielkarten enthält, welche mit vier Buben ($B$) beginnt.
(d) Die Sprache welche jede beliebig lange folge von Spielkarten enthält.

(ii) Betrachten Sie die beiden regulären Ausdrücke $x = ((7|8|9)(\spadesuit, \heartsuit))^*$ und $y = (x|(\diamondsuit|\clubsuit))^*$. Entscheiden Sie jeweils, ob die folgenden Wörter Elemente der durch die Ausdrücke beschriebenen Sprachen $\mathcal{L}(x)$ bzw. $\mathcal{L}(y)$ sind:

(a) $7\heartsuit$, (b) $8\spadesuit3\heartsuit$, (c) $\diamondsuit\diamondsuit$, (d) $\clubsuit\spadesuit8\heartsuit$, (e) $9\heartsuit7\heartsuit7\heartsuit\clubsuit\diamondsuit7\spadesuit$, (f) $8\heartsuit7\clubsuit7\heartsuit\clubsuit$.

Begründen Sie Ihre Entscheidung!

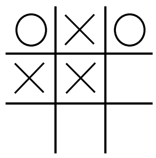
---

Aufgaben 2, 3 und 4 auf der Rückseite.

**Aufgabe 2** (5+5 Bonuspunkte)**.** (i) Plotten Sie sowohl den Graphen als auch die Höhenlinien der Funktion $(x, y) \mapsto (x^2 + y^2)^{\frac{1}{2}} + \sin(x^2 + y^2) + \cos(x^2)$ für $-3 \leq x, y \leq 3$ mit Hilfe der MATLAB-Routinen `surf` und `contour`.
(ii) Gegeben Sei die Funktion $f(x, y, z) = (x^2 + 2.25y^2 + z^2 - 1)^3 - x^2z^3 - 0.1125y^2z^3$. Benutzen Sie die MATLAB-Routine `meshgrid` um ein dreidimensionales Gitter innerhalb des Würfels $[-1.5, 1.5]^3$ zu erzeugen. Berechnen Sie die Funktionswerte an den Gitterpunkten und verwenden Sie dann den Befehl `isosurface(xx,yy,zz,ff,val)` um die approximierte Lösungsmenge der Gleichung $f(x, y, z) = 0$ graphisch darzustellen. Die Parameter sind dabei wie folgt zu wählen:
`xx`, `yy`, `zz`: Die $x$, $y$ und $z$ Koordinaten der Gitterpunkte,
`ff`: die Funktionswerte an den Gitterpunkten,
`val`: die rechte Seite der obigen Gleichung.
Mit dem Befehl `daspect([1 1 1])` können Sie anschließend dafür sorgen, dass die Achsen in dem Plot im gleichen Maßstab skaliert werden.

**Aufgabe 3** (10 Bonuspunkte)**.** Eine Variable `x` vom Typ **`double`** kann in C++ mit Hilfe der Zuweisung `x=double(rand())/RAND_MAX;` mit einer Zufallszahl zwischen 0 und 1 belegt werden. Schreiben Sie ein Programm, welches eine vorgegebene Anzahl $N$ an Paaren $(x_n, y_n)_{1 \leq n \leq N}$ von Zufallszahlen erzeugt und für jedes Zahlenpaar prüft, ob der Punkt $(x_n, y_n)$ innerhalb des Viertelkreises $\{(x, y) : 0 \leq x, 0 \leq y, x^2 + y^2 \leq 1\}$ liegt. Der Anteil der Punkte innerhalb des Viertelkreises liefert für große $N$ eine Approximation an $\pi/4$. Für $1 \leq n \leq N$ bezeichne $\pi_n$ das 4−fache des Anteils der $n$ Punkte $(x_i, y_i)_{i \leq n}$, die innerhalb des Viertelkreises liegen. Wählen Sie für $N$ nacheinander die Werte $10^3$, $10^4$, $10^5$ und $10^6$ und speichern Sie die Werte $(n, \pi_n)_{1 \leq n \leq N}$ aus Ihrer Berechnung jeweils in eine Datei. Lesen Sie diese Dateien anschließend mit dem Befehl `load('DATEINAME')` in MATLAB ein und erstellen Sie jeweils einen Plot der Daten, der die Konvergenz $\pi_n \to \pi$ veranschaulicht.

**Aufgabe 4** (10 Bonuspunkte)**.** Auf der Vorlesungshomepage finden Sie eine MATLAB-Datei namens `tictactoe.m`, in der ein einfaches TICTACTOE-Spiel („Drei gewinnt“) implementiert ist. Der Spielstatus wird dabei zu jedem Zeitpunkt mit einer 3×3-Matrix identifiziert, wobei der Eintrag 1 für ein Kreuz, der Eintrag −1 für einen Kreis und der Eintrag 0 für ein leeres Feld steht. In dem Programm spielt der menschliche Spieler (Symbol X) gegen einen Computer-



$$\texttt{state} = \begin{bmatrix} -1 & 1 & 0 \\ 1 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix}$$

ABBILDUNG 1. Spielzustand nach dem dritten Zug von Spieler X

gegner (Symbol 0), dessen Strategie in der Funktion `computer_move()` implementiert ist. Diese Strategie besteht aktuell darin, zufällig ein freies Feld auszuwählen und dieses mit dem Kreissymbol zu markieren. Aus Sicht des Computers führt diese zufällige Auswahl natürlich häufig zum Verlust des Spiels. Überlegen Sie sich eine bessere Strategie für den Computerspieler und implementieren Sie diese. Um die volle Punktzahl zu erreichen, sollte der Computerspieler mit Ihrer Strategie keine Fehler begehen, das heißt, er sollte kein Spiel verlieren und nach einem Fehler des menschlichen Spielers das Spiel gewinnen. Beachten Sie dabei, dass zu Beginn des Spiels zufällig entschieden wird, welcher der beiden Spieler das Spiel beginnt. Wie Sie die Strategie Ihres Computerspielers genau implementieren, ist jedoch komplett Ihnen überlassen.
Tipp: Eine elegante Variante ist die Verwendung eines rekursiven Spielalgorithmus, der, ähnlich dem rekursiven Lösungsalgorithmus für Sudokus, probeweise einen erlaubten Spielzug ausführt und sich anschließend mit dem neu entstandenen Spielfeld selbst aufruft, um erneut einen Spielzug (diesmal aus Sicht des Gegenspielers) zu finden. Dieses Vorgehen wird rekursiv so lange wiederholt, bis das Spiel sicher gewonnen, verloren oder unentschieden ist und dementsprechend die bestmögliche Wahl getroffen werden kann.

---

Abgabe der Programme per E-Mail, (handschriftlich) kommentierte Ausdrucke der Programme und Rechnungen auf gehefteten, mit Namen versehenen Zetteln in die Briefkästen

Homepage zur Vorlesung: `https://aam.uni-freiburg.de/agba/lehre/ss18/einfprog`