Definition 2.2.1: "groß O", "klein O", "groß Omega", klein Omega", "Theta"

Es sei f: $\mathbb{R}^+ \rightarrow \mathbb{R}^+$ eine Funktion über den positiven reellen Zahlen $\mathbb{R}^+ \subset \mathbb{R}$ (oft wird diese Definition auf die natürlichen Zahlen eingeschränkt, also auf Funktionen f: $\mathbb{N} \rightarrow \mathbb{N}$; unten muss dann nur g: $\mathbb{R}^+ \rightarrow \mathbb{R}^+$ durch g: $\mathbb{N} \rightarrow \mathbb{N}$ ersetzt werden).

$$\mathbf{O(f)} := \{g: \mathbb{R}^+ \rightarrow \mathbb{R}^+ \mid \exists c \in \mathbb{R}^+ \, \exists\, n_0 \in \mathbb{N} \, \forall n \geq n_0 : g(n) \leq c \cdot f(n)\},$$

$$\mathbf{o(f)} := \{g: \mathbb{R}^+ \rightarrow \mathbb{R}^+ \mid \forall c \in \mathbb{R}^+ \, \exists\, n_0 \in \mathbb{N} \, \forall n \geq n_0 : c \cdot g(n) \leq f(n)\},$$

$$\mathbf{\Omega(f)} := \{g: \mathbb{R}^+ \rightarrow \mathbb{R}^+ \mid \exists c \in \mathbb{R}^+ \, \exists\, n_0 \in \mathbb{N} \, \forall n \geq n_0 : f(n) \leq c \cdot g(n)\},$$

$$\boldsymbol{\omega}\mathbf{(f)} := \{g: \mathbb{R}^+ \rightarrow \mathbb{R}^+ \mid \forall c \in \mathbb{R}^+ \, \exists\, n_0 \in \mathbb{N} \, \forall n \geq n_0 : c \cdot f(n) \leq g(n)\},$$

$$\mathbf{\Theta(f)} := \{g: \mathbb{R}^+ \rightarrow \mathbb{R}^+ \mid \exists c_1, c_2 \in \mathbb{R}^+ \, \exists\, n_0 \in \mathbb{N} \, \forall n \geq n_0 : c_1 \cdot f(n) \leq g(n) \leq c_2 \cdot f(n)\}.$$

Landau-Symbole

Aufgabe 3 (Programmanalyse)

Gegeben Sei folgendes Programmstück mit konstantem n und bekanntem $z < 2^n$.

```
procedure Aufg3 is
  a:array[1..n] of natural :=(0| others => 0);
  begin
    for i in (1 .. z) loop
      for j in (1 .. n) loop
        if A(j)=1 then A(j):=0
                  else A(j):=1; j:=n;
        end if;
      end loop;
    end loop;
  end Aufg3;
```

Was tut dieses Programmstück? Schätzen Sie seine Laufzeit möglichst genau ab.

Aufgabe 4 ehem. Prüfungsaufgabe

Zeigen Sie:

$$O(f(n) + g(n)) = O(\max\{f(n), g(n)\}) \qquad \text{für Funktionen } f, g : \mathbb{N} \to \mathbb{N}$$

Aufgabe 5

Wir definieren:

$$f_1(n) = \log\log\log n$$

$$f_2(n) = \log\log n$$

$$f_3(n) = \sqrt{\log n}$$

$$f_4(n) = \log_2 n$$

$$f_5(n) = (\log_2 n)^2$$

$$f_6(n) = n^{\frac{1}{3}}$$

$$f_7(n) = n$$

$$f_8(n) = n\log_2 n$$

$$f_9(n) = n^{\frac{3}{2}}$$

$$f_{10}(n) = n^2$$

$$f_{11}(n) = n^{10}$$

$$f_{12}(n) = n^{\log_2 n}$$

$$f_{13}(n) = 2^{\sqrt{n}}$$

$$f_{14}(n) = 1.1^n$$

$$f_{15}(n) = 2^n$$

$$f_{16}(n) = e^n$$

$$f_{17}(n) = n!$$

$$f_{18}(n) = n^n$$

$$f_{19}(n) = 2^{n^2}$$

$$f_{20}(n) = 2^{2^n}$$

Zeigen Sie für $1 \leq i \leq 19$:

$$\lim_{n\to\infty} \frac{f_i(n)}{f_{i+1}(n)} = 0.$$

Aufgabe 6

Geben Sie einfache $O(\dots)$-Ausdrücke für die folgenden Funktionen an und $\Omega(\dots)$-Ausdrücke, soweit möglich.

(a) $g(n) = 10\log_2 n + 4\sqrt{3n}$

(b) $g(n) = 2n(\log_2 n + 5\log_2\log_2 n)$

(c) $g(n) = n^2\left(\lceil \log_2(n+5)\rceil + 7\right)$

(d) $g(n) = n^2 2^n + 2n^4 3^n + 10n^n$

(e) $g(n) = n^{10} + n^2 \cdot 3^n + n^5 \cdot 2^n \cdot \log n$

(f) $g(n) = |\sin n| + |\cos n|$

(g) $g(n) = e^n - \lfloor e^n \rfloor$

Aufgabe 7 (Eigenschaften von Graphen)

Ein ungerichteter Graph (V, E) heißt *bipartit*, wenn es eine Aufteilung der Knotenmenge V in zwei disjunkte Mengen V_1 und V_2 mit $V_1 \cup V_2 = V$ gibt, so dass jede Kante aus E einen Endpunkt in V_1 und einen in V_2 hat.

Beschreiben Sie einen möglichst effizienten Algorithmus, der für einen gegebenen ungerichteten Graphen entscheidet, ob dieser bipartit ist, und ggf. eine entsprechende Partition von V berechnet. Dabei können Sie annehmen, dass der Graph durch sein Knotenarray und seine Adjazenzlisten gegeben ist. Zeigen Sie die Korrektheit des Algorithmus und bestimmen Sie seine Laufzeit in O-Notation.

Beschreiben Sie einen möglichst effizienten Algorithmus, der entscheidet, ob ein ungerichteter Graph (V, E) einen Kreis ungerader Länge enthält. Begründen Sie seine Korrektheit und bestimmen sie seine Laufzeit in O-Notation.

Aufgabe 8 Programmanalyse

Analysieren Sie das folgende Programm. Welchen Aufwand hat es bei Eingabe von n Elementen?

```
PROCEDURE Exchange IS NEW Swap_Generic(ValueType => ElementType);

AnotherPassNeeded: Boolean := True;
CurrentBottom: IndexType := List'Last;

BEGIN
  WHILE AnotherPassNeeded LOOP
    AnotherPassNeeded := False;
    FOR Current IN List'First .. CurrentBottom - 1 LOOP
      IF KeyOf(List(Current + 1)) < KeyOf(List(Current)) THEN
        Exchange(List(Current), List(Current + 1));
        AnotherPassNeeded := True;
      END IF;
    END LOOP;
  END LOOP;
END;
```

Aufgabe 9 (Abschätzung in O - Notation - ehem. Prüfungsaufgabe)

Mit O(g) bezeichnen wir, wie in der Vorlesung, die Menge der Funktionen, die von der Ordnung g sind:

$O(g) = \{f: N_0 \to R_+ |$ es gibt ein $c > 0$ und ein $n_0 \in N_0$ mit $f(n) \le c\, g(n)$ für alle $n \ge n_0\}$.

Entscheiden Sie, welche der folgenden Behauptungen richtig und welche falsch sind:
(für jede falsche Antwort wird ein Punkt abgezogen!!!)

a) $3n^2 + 17n + 23 \in O(n^3)$
b) $3n^2 + 17n + 23 \in O(n^2)$
c) $3n^2 + 17n + 23 \in O(n)$
d) $2^n + 25n \in O(n^{200})$
e) $10^n \in O(2^n)$
f) $115n^{17} + 12n^{12} + 4n^5 + 123 \in O(n^{20})$

Aufgabe 10 Ordnen von Funktionen nach asymptotischem Verhalten

Sortieren Sie die folgenden Funktionen in aufsteigender Reihenfolge gemäß der O-Notation:

$$\log_2(\frac{1}{2}n), \quad \sqrt{n^3}, \quad n!, \quad 10^4 n, \quad n^2 + \log_2 n^n, \quad \ln(2), \quad n^2, \quad n^3 + 5n^2, \quad 2^n$$

Ordnen Sie folgende Funktionen bezogen auf die O-Notation und begründen Sie Ihre Wahl:

$$f_1(n) = \begin{cases} n & \text{falls } n \text{ ungerade} \\ 3 * n^3 & \text{sonst} \end{cases} \qquad f_2(n) = \begin{cases} 3 * n^3 & \text{falls } n \text{ ungerade} \\ n^2 & \text{sonst} \end{cases}$$

$$f_3(n) = 2^n + 3^n \qquad f_4(n) = \sqrt{\frac{n}{2}}$$

Aufgabe 11 Beweise von Aussagen über asymptotisches Verhalten

(a) Zeigen oder widerlegen Sie folgende Behauptungen:

(i) $o(g) \cap \mathcal{O}(g) = \emptyset$

(ii) $f \in \mathcal{O}(g) \Rightarrow f \in o(g)$

(iii) $\sum_{i=0}^{n} i^2 \in \mathcal{O}(n^2)$

(b) Zeigen Sie folgende Behauptungen.

(i) $(n+a)^b \in \Theta(n^b)$, wobei a, b feste Konstanten sind (keine Kenngrößen), $a, b \in \mathbb{N}$, $a, b \geq 1$

(ii) $\lceil n^{1.5} + n \log_e n \rceil \in \mathcal{O}(\lceil n^{1.5} \rceil)$

Geben Sie eine geeignete Konstante c und ein geeignetes n_0 an, so dass die Aussage folgt.

Zeigen Sie mit Hilfe der Definition: Sind $f, g : \mathbf{N} \rightarrow \mathbf{N}$ mit $f(n) = O(n^k)$ und $g(n) = O(n^l)$ und Konstanten $k, l \in \mathbf{N}$ gegeben, so folgt $g(f(n)) = O(n^{kl})$.

Aufgabe 1 (Analyse einer Turingmaschine)

Gegeben ist das Alphabet $A = \{\#; 0; 1; B\}$. Das Band ist wie folgt beschrieben:

$\cdots$	B	B	0	1	0	0	1	1	B	B	$\cdots$

Der Schreib/Lesekopf befindet sich unter der ersten 0 (ganz links), die TM befindet sich im Zustand q_0. Was bewirkt das folgende Programm?

$$\delta : \begin{cases} (q_0, 0) & \mapsto & (q_0, 0, R) \\ (q_0, 1) & \mapsto & (q_0, 1, R) \\ (q_0, B) & \mapsto & (q_1, \#, L) \\ (q_1, 0) & \mapsto & (q_1, 0, L) \\ (q_1, 1) & \mapsto & (q_1, 1, L) \\ (q_1, B) & \mapsto & (q_e, \#, S) \end{cases}$$

Aufgabe 2 (Entwurf von Turingmaschinen)

a) Entwerfen Sie eine Turingmaschine, die alle Wörter der Form $a^{2n} b^n$ mit $n \in \mathbb{N}$ erkennt.

b) Entwerfen Sie eine Turingmaschine, die alle aus a's und b's bestehenden Wörter erkennt, die korrekten Klammerstrukturen entsprechen, wobei a einer öffnenden und b einer schließenden Klammer entspricht. Die Turingmaschine soll also beispielsweise aababbab akzeptieren, aber abbabaab zurückweisen.

c) Geben Sie eine Turing-Maschine M an, die zu einer gegebenen binären Zahl 1 addiert.

Aufgabe 3 (Analyse einer Turingmaschine)

Betrachten Sie die folgende Turingmaschine: $Q = \{s, q_0, q_1, q_2, q_{\text{accept}}, q_{\text{reject}}\}, \Sigma = \{0, 1\}, \Gamma = \{0, 1, \sqcup, \triangleright\}$

δ	0	1	$\sqcup$	$\triangleright$
s	$(q_0, 0, R)$	$(q_2, \sqcup, R)$	$(q_{\text{accept}}, \sqcup, L)$	$(s, \triangleright, R)$
q_0	$(q_{\text{reject}}, 1, R)$	$(q_1, 1, R)$	$(q_{\text{reject}}, 1, R)$	$(q_{\text{reject}}, \triangleright, R)$
q_1	$(q_{\text{reject}}, 0, R)$	$(s, 1, R)$	$(q_{\text{reject}}, \sqcup, R)$	$(q_{\text{reject}}, \triangleright, R)$
q_2	$(q_2, 0, R)$	$(q_2, \sqcup, R)$	$(q_2, 1, R)$	$(q_{\text{reject}}, \triangleright, R)$

Welche Sprache wird durch die Turingmaschine akzeptiert? Terminiert die Turingmaschine für jede Eingabe? Ändern Sie die Turingmaschine so ab, dass sie für jede Eingabe terminiert.

Aufgabe 5 (Konstruktion einer 2 Band Turingmaschine)

Beschreiben Sie formal eine 1-Band-Turingmaschine, die die Eingabe $x\#y$ mit $x, y \in \{0, 1\}^*$ in $x\#y\#x$ überführt.

Beschreiben Sie formal eine 1-Band-Turingmaschine, die bei Eingabe $x\#y$ mit $x, y \in \{0, 1\}^l, l \in \mathbb{N}$ (x und y seien Binärzahlen) folgendes leistet: Schreiben Sie x und y übereinander und das #-Symbol davor. Führen Sie dazu die folgenden vier Buchstaben des Bandalphabets ein: $\substack{0\\0}, \substack{0\\1}, \substack{1\\0}, \substack{1\\1}$. Ihr Band sieht also nach der Berechnung so aus: $\triangleright\#\substack{x\\y}$. Beispielsweise soll bei Eingabe 10110#01101 die Ausgabe $\#\substack{10110\\01101}$ berechnet werden.

Aufgabe 4 (Konfiguration einer Turingmaschine - ehemalige Prüfungsaufgabe)

Gegeben sei eine deterministische Turingmaschine $M = (Q, \Sigma, \Gamma, q_0, \delta, F)$ mit $Q = \{q_0, q_1, q_2\}$, $\Sigma = \{0, 1\}$, $\Gamma = \Sigma \cup \{B\}$, $F = \{q_2\}$ und $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, N, R\}$ wie folgt:

δ	0	1	B
q_0	$(q_1, 0, R)$	$(q_1, 1, R)$	$(q_2, 0, N)$
q_1	$(q_0, 0, R)$	$(q_0, 1, R)$	$(q_2, 1, N)$
q_2	–	–	–

Ein waagerechter Strich bedeutet, dass die Turingmaschine stoppt, d. h. $\delta(q, a) = (q, a, N)$ für $q \in Q$ und $a \in \Gamma$.

a) Geben Sie die vollständige Konfigurationsfolge der Maschine M angesetzt auf die Eingabe 11001 an.

b) Beschreiben Sie in Worten die von M berechnete Funktion.

Zeigen Sie mit Hilfe der Definition: Sind $f, g : \mathbf{N} \to \mathbf{N}$ mit $f(n) = O(n^k)$ und $g(n) = O(n^l)$ und Konstanten $k, l \in \mathbf{N}$ gegeben, so folgt $g(f(n)) = O(n^{kl})$.

Aufgabe 6 (platzbeschränkte Turingmaschinen)

Sei M eine Turingmaschine, $x \in \Sigma^*$ die Eingabe und $n = |x|$ die Länge der Eingabe. Sei $s(n)$ die maximale Kopfposition, die die Turingmaschine erreicht, d.h. $s(n)$ ist die maximale Anzahl von Zellen, die die Maschine beschreibt. Unter einer $s(n)$-platzbeschränkten Turingmaschine M verstehen wir dann eine DTM, die nicht mehr als $s(n)$ Zellen benutzt.

1. Zeigen Sie: Hält M gestartet mit x, so hat die Rechnung von M nicht mehr als $|\Gamma|^{s(n)} \cdot s(n) \cdot |Q|$ Schritte benötigt.

2. Zeigen Sie, dass die Menge $H^{s(n)} = \{\langle M\rangle x | M$ gestartet mit x benutzt höchstens $s(n)$ Platz und hält$\}$ entscheidbar ist (Platz beschränktes Halteproblem).

Aufgabe 7

Konstruieren Sie eine Turingmaschine, welche die Sprache

$L = \{a^{2^i} :\ i \in \mathbb{N}_0\}$

akzeptiert.

Beschreiben Sie auch umgangssprachlich, wie Ihre Turingmaschine funktioniert.

2.3.1 Definition (nach A. M. Turing, 1912-1954, engl. Mathem.)

$M = (Q, \Sigma, \Gamma, \delta, q_0, F, \mathbf{b}, k)$ heißt (nichtdeterministische)

k-Band-Turingmaschine $\Leftrightarrow$

(1) Q ist eine nicht-leere endliche Menge ("Zustandsmenge"),

(2) Σ ist eine endliche Menge ("Eingabealphabet"),

(3) Γ ist eine endliche Menge ("Bandalphabet") mit $\Sigma \subset \Gamma$,

(4) $q_0 \in Q$ ist der Anfangszustand,

(5) $F \subset Q$ ist die Menge der (akzeptierenden) Endzustände,

(6) $\mathbf{b} \in \Gamma \setminus \Sigma$ ist das "Blank"-Symbol,

(7) $k \in \mathbb{N}$ ist die Anzahl der Bänder,

(8) $\delta \subseteq Q \setminus F \times \Gamma^k \times Q \times \Gamma^k \times \{-1, 0, 1\}^k$ ist die Überführungsrelation.