Winfried Auzinger
Dirk Praetorius

SS 2012
17. April 2012

Übungsaufgaben zur VU Computermathematik

Serie 5

Generelle Anmerkung zu den Maple-Übungen:

Viele der Übungsaufgaben sind keine reinen ‘Maple-Aufgaben’, sondern enthalten auch eine mathematische Problemstellung, die Sie, ggf. mit entsprechenden Hinweisen, zunächst verstehen bzw. knacken müssen. Manche andere wieder sind experimenteller Natur (was für Physiker sein Labor ist, ist für den Mathematiker sein Computer). Und bedenken Sie: Der Name der LVA ist Computer*mathematik*.

Für vielen Fragestellungen gibt es innerhalb von Maple schon fertige Lösungen (mit denen man vergleichen kann, wie etwa in Aufgabe 5.8). Es spricht aber nichts dagegen, so etwas als Übungsaufgabe zu verwenden (auch in anderen Übungen berechnen oder beweisen Sie Dinge, die schon andere vor Ihnen berechnet bzw. bewiesen haben).

Es kommt gelegentlich vor, dass Sie einen Befehl benötigen, der in der Vorlesung (noch) nicht besprochen wurde. Für derartige Fälle werden Hinweise gegeben, oft einfach nur das richtige Stichwort, für das Sie Details in der Hilfe nachschlagen können. Lesen Sie die Hinweise genau durch; manches müssen Sie dann noch selbst herausfinden. Orientieren Sie sich auch mit Hilfe des Flyers (siehe Homepage). Manche der Aufgaben (ohne Stern) haben auch den Zweck, dass Sie sich einen in der Vorlesung (aus Zeitgründen) nicht oder noch nicht im Detail besprochenen Stoff aktiv anhand von Beispielen selbst erarbeiten.

Nützen Sie Maple-Hilfe systematisch (für die praktische Arbeit ist das unumgänglich).

Ihre Codes sollten Sie so weit wie möglich auf Korrektheit testen. Dokumentieren Sie Ihre Worksheets auch in angemessener Weise mittels Zwischentexten bzw. Kommentaren (`#`). Das Hauptziel dabei sollte immer sein, dass Sie selbst später noch erkennen können, was sie sich gedacht haben.

Aufgabe 5.1*. Werten Sie mittels `sum`, `binomial` einen Ausdruck der Gestalt

$$S_{n;p} := \sum_{k=0}^{n} k^p \binom{n}{k} \qquad (n, p \in \mathbb{N})$$

aus, und versuchen Sie das Ergebnis auch mittels `simplify` zu vereinfachen. Experimentieren Sie:

- Was passiert, wenn weder `p` noch `n` ein Wert zugewiesen wurde? Hilft Anwendung von `simplify`?
- Deklarieren Sie[1] mittels `assume(n,posint)` und `assume(p,nonnegint)`, dass $n \in N$ und $p \in \mathbb{N}_0$ gilt, und werten Sie die Summe nochmals aus. Hilft das weiter?
- Lassen Sie `n` unspezifiziert und wählen Sie $p = 0, 1, 2, 3, \ldots$
- Stellen Sie anhand dieser Ergebnisse eine Vermutung auf betreffend das asymptotische Verhalten der $S_{n;p}$ für $n \to \infty$ in Abhängigkeit von p. Versuchen Sie inbesondere eine möglichst einfach gebaute Folge $(a_{n;p})$ zu identifizieren, so dass offenbar gilt

  $$\lim_{n\to\infty} \frac{S_{n;p}}{a_{n;p}} = 1\,,$$

  d.h., $S_{n;p}$ und $a_{n;p}$ haben das gleiche asymptotische Verhalten.

Anmerkung: Aufgabe 5.1 ist rein experimentell, plus ein bisschen hinschauen und nachdenken.

[1] Mit `assume` werden Eigenschaften von Variablen festgelegt, ohne dass man ihnen einen konkreten Wert zuweist. Bei Ausgabe des Variablennamens wird dieser dann default-mäßig mit dem Index $\sim$ versehen (als ‘Erinnerung’). Es gibt viele Eigenschaften, die man in dieser Weise an Variablen zuweisen kann (z.B. gerade, ungerade, etc., etc.), was für manche Zwecke ist dies nützlich und sinnvoll ist. Siehe dazu auch Aufgabe 5.5.

Aufgabe 5.2*. Seien $p_1, p_2, \ldots, p_n = 2, 3, \ldots$ die ersten n Primzahlen, und

$$P_n := 1 + \prod_{i=1}^{n} p_i \,.$$

(Ist jedes dieser P_n auch wieder eine Primzahl? Was meinen Sie?)

Verwenden Sie `mul` und `ithprime` dazu, um für $n = 1, 2, 3, \ldots$ die P_n zu berechnen. Suchen Sie das kleinste n so dass P_n *keine* Primzahl ist. Verwenden Sie `ifactor`.

Aufgabe 5.3*.

(i) Verwenden Sie[2] `plots[pointplot]` dazu, um die Folge der Punkte $(x_n, y_n) = (n, p_n)$ in der (x, y)-Ebene grafisch darzustellen ($p_n = n$-te Primzahl). Für größere Werte von n (z.B. $n = 1000$) erkennt man einen ziemlich regelmäßigen Verlauf.

(ii) Generieren Sie (z.B. für $n = 1000$) die Liste der (mittels `evalf` ausgewerteten) Zahlen

$$\frac{n \ln n}{p_n}$$

Welche Vermutung über das asymptotische Verhalten der p_n für $n \to \infty$ drängt sich auf?

Aufgabe 5.4*. Produzieren Sie eine Liste von K Pythagoräischen Tripeln (a, b, c) mit $a^2 + b^2 = c^2$ $(a, b, c \in \mathbb{N})$ in Form einer Liste von Listen $[a, b, c]$,

```
[[3,4,5],[5,12,13],...]
```

(K gibt man beliebig vor, z.B. $K = 10$). Diese Liste können Sie mit Hilfe einer impliziten Schleife generieren:

```
PT := [seq([...,...,...],k=1..K)];
```

Hinweis zur Konstruktion: Für $k = 1 \ldots K$ betrachten Sie die ungerade Zahl $a = 2k + 1$. Überlegen Sie weiter.

Für eine übersichtliche Anzeige verwendet man dann besser eine explizite Schleife (`for`, wird in VO noch genauer besprochen) und die Funktion `print`, etwa so:

```
for k from 1 to K do
    print(PT[k],PT[k][1]^2+PT[k][2]^2,PT[k][3]^2):
end do:
```

Aufgabe 5.5*. Mittels `sum` erhält man die Formel

$$\sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}$$

Ein strenger Beweis dieser Formel beruht auf vollständiger Induktion. Um den Induktionsschritt $n \mapsto n+1$ durchzuführen, ist ein bisschen Rechenarbeit erforderlich. Führen Sie diese Rechnung mit Hilfe von Maple durch.

Sie können es (freiwillig) z.B. auch für $\sum\limits_{k=1}^{n} k^3$ durchführen.

Probieren Sie auch aus, ob `sum` eine allgemeine Darstellungsformel für $\sum\limits_{k=1}^{n} k^p$ findet, falls p kein Wert zugewiesen wurde. Vielleicht geht es mit `assume(p,nonnegint)` (vgl Aufgabe 5.1) ?

Aufgabe 5.6. Die *Stirling'sche Formel* liefert eine einfach auswertbare asymptotische Näherung von $n!$ für große Werte von n:

$$n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \quad \text{für } n \to \infty.$$

[2] Hinweis zur Syntax: `plots[pointplot](...)` aktiviert die Funktion `pointplot` aus dem `plots`-package.

Überprüfen Sie diese Aussage experimentell, indem Sie den Verlauf der beiden Größen mit wachsendem n in Form einer Grafik darstellen. (`[plots]pointplot`)

Hinweis: $\pi =$ `Pi`. Für die Fakultät verwendet Maple die übliche Notation `n!` . Für die Euler'sche Zahl $e = \exp(1)$ gibt es kein vordefiniertes Symbol. Sie können aber die Variable e natürlich selbst mit dem entsprechenden Wert belegen.

Aufgabe 5.7. Definieren Sie zwei Funktionen `ip3 := (x,y) -> ...` und `op3 := (x,y) -> ...`, die zwei Listen `x` und `y` der Länge 3 als Vektoren im $\mathbb{R}^3$ interpretieren und das innere (Skalarprodukt, $x \cdot y$) bzw. das äußere (Vektorprodukt, $x \times y$, letzteres wieder als Liste) zurückgeben.

'Beweisen' Sie damit die Tatsache $a \times b \perp a$, $a \times b \perp b$, und die Identitäten

$$\|a \times b\|^2 = \|a\|^2\,\|b\|^2 - (a \cdot b)^2 \quad \text{(Formel von Lagrange)},$$
$$a \times (b \times c) = (a \cdot c)\,b - (a \cdot b)\,c,$$
$$(a \times b) \cdot (c \times d) = (a \cdot c)(b \cdot d) - (b \cdot c)(a \cdot d),$$

indem Sie beliebige 'generische' Vektoren (Listen) `a,b,c,d` (ohne konkrete Werte) übergeben. Verwenden Sie `simplify` (liefert nicht immer eine zufriedenstellende Vereinfachung) oder ggf. `expand`.

Hinweise:

- $\|x\|$ bezeichnet die (Euklidische) Länge des Vektors x, d.h. $\|x\|^2 = x \cdot x$.
- Multiplikation einer Liste mit einem Skalar (`*`) liefert wieder eine Liste (elementweise Multiplikation).
- Eine 'generische' Liste, z.B. `a`, kann man in folgender Weise erzeugen: [3]

  ```
  a := [alpha[1],alpha[2],alpha[3]];
  ```

  Dann ist `a[1]` $= \alpha_1$, etc. Der Name α_i für die Listenelemente ist willkürlich gewählt. Folgendes geht nicht:

  ```
  a := [a[1],a[2],a[3]];
  ```

  (probieren Sie es aus).
- Dabei ist zu beachten, dass z.B. `alpha[3]` eine sogenannte *indizierte Variable* darstellt, in normaler Schreibweise: α_3. Eine indizierte Variable (mit einem konkreten numerischen Index!) kann man wie eine normale Variable verwenden (man kann z.B. auch Werte zuweisen, was jedoch in dieser Aufgabe nicht verwendet wird).

  Man beachte jedoch den Unterschied – trotz gleicher Schreibweise – zwischen `a[3]` (gibt das dritte Element der Liste `a` zurück) und `alpha[3]` (gibt die indizierte Variable α_3 zurück; dabei ist `alpha` jedoch keine Liste). In dem vorliegenden Beispiel liefert `a[3]` den Wert α_3.

Aufgabe 5.8. *Partielle Summation* ist ein diskretes Analogon zur partiellen Integration: Für zwei Folgen $(a_k), (b_k)$, $k = 0 \ldots n$, gilt [4]

$$\sum_{k=1}^{n} a_k(b_k - b_{k-1}) \;=\; a_n\,b_n - a_0\,b_0 - \sum_{k=1}^{n} (a_k - a_{k-1})b_{k-1}$$

[3] Man könnte annehmen, dass es möglich ist, eine Liste, z.B. `a`, einer vorgegebenen Länge und ohne konkrete Werte zu definieren, und dann mittels `a[k]` auf das `k`-te Listenelement zuzugreifen. Eine derartige Syntax wird jedoch nicht unterstützt. Deshalb der 'workaround' wie hier beschrieben.

Listen sind bequeme aber eher primitive Datenstrukturen, mehr oder weniger 'Datencontainer' für die Zusammenfassung mehrerer Objekte beliebigen, auch gemischten Typs. Für konkrete mathematische Objekte, z.B. Vektoren und Matrizen, gibt es speziellere und flexiblere Datenstrukturen, die in der Vorlesung noch besprochen werden.

[4] Dies beweist man, indem man die rechte Summe nach links bringt und die Summe der beiden Summen vereinfacht. Dann erhält man eine Teleskopsumme.

- Verwenden Sie partielle Summation zur Herleitung einer Formel für die verallgemeinerte geometrische Summe. [5]

  $$\sum_{k=1}^{n} k\, q^k, \quad q \neq 1.$$

  Hinweis: Wählen Sie $a_k = k$; b_k ist leicht zu bestimmen. [6] Damit wird die Summe auf eine normale geometrische Summe zurückgeführt.

- Implementieren Sie die von Ihnen mittels partieller Summation hergeleitete Formel in Maple und vergleichen Sie mit dem Ergebnis von `sum`.

- Wie lautet der Wert der für $|q| < 1$ konvergenten Reihe $\sum_{k=1}^{\infty} k\, q^k$?

  Die Funktion `sum` kann auch mit unendlichen Reihen umgehen. Werten Sie $\sum_{k=1}^{\infty} q^k$ und $\sum_{k=1}^{\infty} k\, q^k$ mittels `sum` aus (∞ = `infinity`). Wundern Sie sich über das Ergebnis?

Anmerkung: Mittels mehrfacher partieller Summation kann man auch $\sum_{k=1}^{n} k^p\, q^k$ bestimmen für $(p = 2, 3, \ldots)$. `sum` kann das natürlich auch.

Aufgabe 5.9. Eine endliche Menge (`? set`) wird in Maple repräsentiert als eine `exprseq` zwischen geschwungenen Klammern, im einfachsten Fall in der expliziten Gestalt

$$\{\ \mathit{element1}, \mathit{element2}, \ldots, \mathit{elementn}\ \}$$

Im folgenden wird angenommen, dass es sich bei `A` und `B` um Mengen natürlicher Zahlen handelt. Definieren Sie Funktionen, die folgende Operationen ausführen:

- $(\mathtt{A},\mathtt{B}) \mapsto (\mathtt{A} \cup \mathtt{B}) \setminus (\mathtt{A} \cap \mathtt{B})$

- $(\mathtt{A},\mathtt{B},\mathtt{m}) \mapsto$ `true`, falls $\mathtt{m} \in \mathbb{N}$ in *genau einer* der beiden Mengen enthalten ist; ansonsten `false` (verwende `xor`, `evalb`) [7]

- $\mathtt{A} \mapsto$ Summe der Elemente von `A`

- $(\mathtt{A},\mathtt{B}) \mapsto \mathtt{A} \times \mathtt{B}$ (kartesisches Produkt), als Menge von geordneten Paaren. Diese werden als Listen `[a,b]` der Länge 2 dargestellt.

  Hinweis: $\mathtt{A} \times \mathtt{B}$ generiert man mittels zweifach geschachtelter Anwendung von `seq(...)`.

Testen Sie dies anhand einiger konkreter Beispiele $\mathtt{A}, \mathtt{B} \subseteq \mathbb{N}$.

Aufgabe 5.10. Deklarieren Sie zwei Funktionen $\mathtt{a} := (\mathtt{i},\mathtt{j}) \mapsto \ldots$ und $\mathtt{b} := \mathtt{i} \mapsto \ldots$, die irgendeine konkrete doppelt indizierte Folge $(a_{i,j})$ und eine einfach indizierte Folge (b_i) definieren. Nun sei $n \in \mathbb{N}$ beliebig vorgegeben. Wir betrachten die Matrix $A = (a_{i,j})_{1 \le i,j \le n}$ und den Vektor $b = (b_i)_{1 \le i \le n}$. Dies definiert ein System $A\,x = b$ von n linearen Gleichungen in n Unbekannten x_i.

Schreiben Sie nun, unter Verwendung von `seq` und `solve`, einen Einzeiler, der für gegebenes $n \in \mathbb{N}$ die Lösung x dieses Gleichungssystems zurückliefert.

Hinweis: Repräsentieren Sie den gesuchten Vektor x in der Form `[seq(x[i],i=1..n)]`. Zum Testen wählen Sie $n = 3$ und wählen unterschiedliche Datenfunktionen `a` gemäß der vier Fälle:

- Die Lösung x ist eindeutig.

---

[5] q ist beliebig; zu beachten ist jedoch, dass – wie bei der normalen geometrischen Reihe – $q = 1$ ein Sonderfall ist.

[6] Maple hilft Ihnen auch dabei: Berechnen Sie die *unbestimmte Summe* `sum(q^k,k)` und überlegen Sie, was das Resultat bedeutet. Denken Sie (in Analogie) an den Begriff des unbestimmten Integrals.

[7] `and, or, xor` sind die Standardoperatoren, die Wahrheitswerte logisch verknüpfen (und, oder, ausschließendes oder). Auswertung einer logischen Aussage (d.h. Entscheidung ob wahr oder falsch) erfolgt mittels `evalb`. Beispiel:

```
evalb((1 in {1,2}) and (1 in {2,3}))
```

liefert den Wahrheitswert `false` (*falsch*). `true` steht für *wahr*.

- Es gibt eine eindimensionale Lösungsschar (d.h., A hat Rang 2).
- Es gibt eine zweidimensionale Lösungsschar (d.h., A hat Rang 1).
- Es gibt gar keine Lösung.

Wie reagiert `solve` in diesen vier Fällen?

Anmerkung: Für lineare Gleichungssysteme (inbesondere für größere Dimension n) verwendet man in der Praxis besser die Datenstrukturen `Matrix` und `Vector` und die Funktionen aus dem Paket `LinearAlgebra`. Dies wird in der Vorlesung noch genauer besprochen.