Anmerkungen zu Übungen und Aufgabenserien vom 31. 10.

Serie 0

Anmerkungen und Hinweise zu den Übungsaufgaben, die vor Serie 1 verstreut im Skript gestellt wurden.

(0-1) Zum Haskell-Programm zum Symbolischen Differenzieren:

a) Füge Syntax und Regel für Quotienten hinzu.
b) Schlage Regeln zur Vereinfachung vor.

(0-2) Inverse Symbolic Calculator[1]

Der Algorithmus auf der genannten Webseite behauptet, dass näherungsweise

```
sqrt(2+sqrt 3) ==> 1.9318516525781366
```

gilt und diese reelle Zahl näherungsweise Nullstelle des Polynoms $-1 + 4\,x^2 - x^4$ sei.

a) Verifziere diese Behauptung für `sqrt(2+sqrt(3))`.
b) `1.931851652578136 = 1/2/sin(%pi/12)`
c) Bestimme ein Polynom mit der Nullstelle $2^{\frac{1}{2}} + 3^{\frac{1}{3}}$.

Präzisierte Aufgabenstellung zu b) in Maxima-Notation:

`u1:sqrt(2+sqrt(3))` und `u2:1/2/sin(%pi/12)` sind näherungsweise gleich und damit näherungsweise Nullstelle des Polynoms `p:-1+4*x^2-x^4`.

b1) Beweisen Sie, dass u_1 und u_2 exakte Nullstellen von p sind.
b2) Beweisen Sie, dass u_1 und u_2 nicht nur näherungsweise, sondern exakt gleich sind.

(0-1) Siehe Datei `bsp-1.hs`.

Die Signatur von E wurde um den binären Operator `Quotient` sowie, anders als in der Vorlesung, um einen unären Operator `Minus` erweitert, wie er in CAS anzutreffen ist ($a - b$ wird intern stets als $a + (-b)$ oder gar als $a + (-1) * b$ umgeschrieben).

Außerdem wurden zu Testzwecken zwei Ausdrücke

```
e1 = let b = Plus T One in Times b b -- Ausdruck (T+1)*(T+1)
e2 = Quotient One e1 -- Ausdruck 1/e1
```

an Variablen gebunden. Weitere Anmerkungen in der Datei `serie-0.txt`.

[1] `http://oldweb.cecm.sfu.ca/projects/ISC/ISCmain.html`

(0-2) Maxima arbeitet mit rationalen Ausdrücken, die allein Wurzeln als Kerne (vergleiche [ESR, S. 90 ff.]) enthalten (wie u_1), noch relativ zuverlässig. Problematischer wird es mit Ausdrücken mit trigonometrischen Kernen wie u_2, da hier mannigfaltige algebraische Abhängigkeiten bestehen. Noch schwieriger wird es, wenn Wurzelausdrücke und trigonometrische Ausdrücke gemeinsam auftreten wie in Aufgabe b2).

Für eine zielstrebige Lösung solcher Aufgaben ist die genaue Kenntnis der Leistungsfähigkeit verschiedener mathematisch möglicher Simplifikationsansätze sowie der Art und Weise der Umsetzung (oder Nicht-Umsetzung) dieser Simplifikationsansätze im konkreten CAS erforderlich. Eine in vielen Fällen günstige Meta-Strategie ist es, die Zahl der verschiedenen Kerne in einem Ausdruck zugunsten einer – möglicherweise komplizierteren – rationalen Struktur zu reduzieren.

Im Lösungsvorschlag wird die Umwandlung (reellwertiger) trigonometrischer Ausdrücke in (komplexwertige) Exponentialausdrücke mit Kern $\exp\left(\frac{m\,\pi\,\mathrm{I}}{n}\right)$ eingesetzt, die Maxima intern als Potenzen von `%e` mit rationalen Vielfachen von $\pi\,\mathrm{I}$ darstellt und dort Vereinfachungen $\mathrm{e}^{z_1} * \mathrm{e}^{z_2} = \mathrm{e}^{z_1+z_2}$ nach den Potenzgesetzen sofort vornimmt (obwohl die Anwendung der Potenzgesetze im Bereich komplexwertiger meromorpher Funktionen zu mathematisch fehlerhaften Ergebnissen führen kann, siehe [ESR, Abschnitt 3.4]).

Weitere Anmerkungen in der Datei `serie-0.txt`.

Hinweis: Derartige Strategien werden in [ESR, Kapitel 3] näher besprochen.

c) Wir machen den Ansatz $c = a + b$ mit $a = 2^{\frac{1}{2}},\ b = 3^{\frac{1}{3}}$ und rechnen mit Polynomen $f(a,b) \in \mathbb{Q}[a,b]$ in den Variablen a, b sowie den Ersetzungsrelationen

$$R = \left\{a^{2+i} \to 2\,a^i,\ b^{3+i} \to 3\,b^i,\ i \in \mathbb{N}\right\}\,.$$

Beachte an dieser Stelle den Unterschied zwischen exakten und algebraischen Ersetzungssystemen, siehe [ESR, S. 98 ff.].

Normalformen bzgl. R enthalten nur Summanden der Form $c_{ij}a^ib^j$ mit $0 \le i \le 1,\ 0 \le j \le 2$, also maximal sechs verschiedene Terme a^ib^j. Folglich führt ein Ansatz mit unbestimmten Koeffizienten

$$f = \sum_{i=0}^{6} r_i c^i = 0$$

und Koeffizientenvergleich zu einem Gleichungssystem mit 6 Gleichungen für die 7 Variablen $r_i,\ i = 0 \dots 6$, aus dem sich eine brauchbare nichttriviale Lösung berechnen lassen sollte.

Weitere Anmerkungen in der Datei `serie-0.txt`. Dort ist auch eine Lösungsvariante mit „algebraischen Ersetzungsregeln“ ausgeführt, die in Maxima allerdings nicht über Regelsysteme implementiert ist, sondern als Rechnungen im Ring

$$\mathbb{Q}\left[2^{\frac{1}{2}}, 3^{\frac{1}{3}}\right] = \mathbb{Q}[a,b]/(a^2-2, b^3-3)$$

algebraischer Zahlen statt im Polynomring $\mathbb{Q}[a,b]$. Diese Reinterpretation des Rechenbereichs (also Rechnen „in einem anderen Modell“ in der Terminologie der Vorlesung) wird durch das Flag `algebraic`, einige Besonderheiten des Verhaltens von `ratsimp` sowie die Möglichkeit der Definition algebraischer Zahlen über deren Minimalpolynom mittels `tellrat` erreicht.

Hinweis: Das Rechnen mit algebraischen Zahlen wird in [ESR, Kapitel 4] detailliert besprochen.

Serie 1

(Ü1-1) Aus Symmetriegründen können wir uns auf den ersten Quadranten konzentrieren und dort die Funktion $f(x) = \frac{3}{4}\sqrt{16-x^2}$ betrachten. Der zu maximierende Ausdruck ist dann $A(x) = x\,f(x)$.

Dies kann in einem CAS grundsätzlich auf zwei Weisen angeschrieben werden – unter Verwendung von Variablen, in denen *Funktionsausdrücke* wie $f(x)$ mit dem Symbol x (in der Vorlesung nullstelliges Funktionssymbol $x()$) gespeichert werden, oder über *Funktionsdefinitionen* f mit dem formalen Parameter x. Sie sollten diese beiden Formen nicht mixen!

Details dazu in der Datei `serie-1.txt`. Beachten Sie, dass als Ergebnis `sol` von `solve` eine *Substitutionsliste* zurückgegeben wird (siehe [ESR, S. 66 ff.]), mit der sich über `subst` auf einfache Weise lokale Wertzuweisungen zu Symbolen realisieren lassen, ohne den globalen Symbolmodus zu verlassen.

(Ü1-3) Zwei Umsetzungen in Haskell:

- Regeln über Transformationsfunktion (`bsp-2.hs`). Umsetzung des Regelsystems als *Transformationsfunktionen* `rp`, `rm` und `rh` auf symbolischen Ausdrücken. Die symbolischen Testausdrücke können *vor* der Definition des Regelsystems vereinbart werden. Es muss allerdings in den drei Transformationsfunktionen dafür gesorgt werden, dass die Regeln rekursiv auf alle Knoten des Termbaumes angewendet werden.

- Regeln direkt als Funktionsaufrufe (`bsp-2a.hs`). Zugang über eine Umsetzung des Regelsystems als unmittelbare Funktionsaufrufe p, m und h auf symbolischen Ausdrücken in S und Z. Die rekursive Applikation der Regeln auf alle Knoten des Termbaumes wird durch das rekursive Aufrufverhalten der Funktionen „automatisch“ bewirkt. Die symbolischen Testausdrücke können allerdings erst *nach* der Definition der Funktionsaufrufe p, m und h vereinbart werden.

Mehr zu Funktionstransformationen und Transformationsfunktionen finden Sie in [ESR, Abschnitt 2.7]. Der erste Zugang entspricht dem regelbasierten Transformationskonzept (allerdings ohne Trennung in Regelsystem und Simplifikator) [ESR, Abschnitt 3.3], der zweite Zugang dem funktionalen Transformationskonzept [ESR, Abschnitt 3.2].

Anmerkungen zum Maximateil der Aufgabe (`serie-1.txt`):

- Für Regeldefinitionen müssen die formalen Parameter mit `matchdeclare` *global* definiert werden. Um hier Namenskollisionen zu vermeiden, wird in der Musterlösung ein spezielles Namensschema für solche formalen Parameter verwendet.

- Die so definierten Regeln sind *global* wirksam, werden also bei der (erneuten) Auswertung eines der Testausdrücke e_1, e_2, e_3 zusammen mit den internen Regeln angewendet.

- Maxima wertet Ausdrücke im Standardverfahren schrittweise aus; es werden damit unendliche Auswertungsketten vermieden, aber es entstehen nicht notwendig Normalformen im Sinne der Vorlesung (siehe hierzu [ESR, Abschnitt 2.6]). Dieses Verhalten können Sie über das Flag 'infeval' (mit der Gefahr unendlicher Evaluationsketten) verändern.

(S1-1) Dieses einfache Regelsystem kann schnell in Maxima umgesetzt werden. Mit dieser Implementierung können Experimente ausgeführt, Regelmäßigkeiten in diesem experimentellen Material gefunden und zu einer Hypothese verdichtet werden.

Ein Beweis der Hypothese erfordert aber ein strenges mathematisch-deduktives Herangehen.

Details zur Experimentierumgebung in der Datei `serie-1.txt`.

Literatur

[ESR] Skript *Einführung in das symbolische Rechnen* (Prof. Gräbe).

Zugang zu den in den Anmerkungen genannten Materialien

HTWK-Studenten: tba.

UL-Studenten: Im Ordner des OO-Kurses W14.BIS.SR.