Aufgabe 4-1

a) Definieren Sie die in der Vorlesung vorgestellte Funktion `fakt(n:nat): RECURSIVE nat`, mit $\texttt{fakt}(n) = n!$. Geben Sie dabei eine geeignete Maßfunktion an.

b) Beweisen Sie die auftretenden Typkorrektheitsbedingungen! Die TTCs werden mit dem Kommando `C-c C-q s` oder `M-x show-tccs` angezeigt; der Beweisvorgang wird wie bei anderen Formeln auch durch `M-x pr` gestartet.

c) Formulieren und beweisen Sie ein Theorem für folgende Eigenschaft:

$$\forall\, x > 3 : \texttt{fakt}(x) > 2^x$$

Hinweise:

- Für die Zweierpotenzen 2^x können Sie die in PVS vordefinierte Funktion $\texttt{exp2}(x)$ verwenden.
- Im Beweis benötigen Sie als Lemma vermutlich folgenden Sachverhalt:

  $$\forall\, w, x, y, z \in \mathbb{N} : y > 0 \wedge x \geq y \wedge z > w \Rightarrow x * z > y * w$$

  In der *Prelude*-Theorie von PVS ist diese Eigenschaft bereits unter dem Namen `gt_times_gt_pos1` bewiesen.

Zentrale PVS-Kommandos: `(INDUCT)`, `(EXPAND)`, `(LIFT-IF)`, `(CASE-REPLACE)`, `(LEMMA)`.

Aufgabe 4-2

Diese Aufgabe behandelt einfache rekursive Funktionen auf natürlichen Zahlen sowie deren Eigenschaften, die mit Hilfe einfacher Induktionsbeweise gezeigt werden sollen.

Geben Sie bei der Definition der rekursiven Funktionen jeweils eine geeignete Maßfunktion an. Beweisen Sie auch alle auftretenden Typkorrektheitsbedingungen!

a) Definieren Sie eine Funktion `sum(n:nat): RECURSIVE nat`, die die Summe der natürlichen Zahlen von 0 bis n berechnet:

$$\texttt{sum}(n) = \sum_{i=0}^{n} i$$

Zeigen Sie: $\texttt{sum}(n) = \frac{1}{2} n * (n + 1)$

b) Definieren Sie nun eine Funktion `sumF(n:nat,f:[nat->nat]): RECURSIVE nat`, die die Summe der Funktionswerte von f an den Stellen von 0 bis n berechnet:

$$\texttt{sumF}(n, f) = \sum_{i=0}^{n} f(i)$$

Zeigen Sie:

$$\begin{aligned} \texttt{sumF}(n, \mathbf{id}) &= \texttt{sum}(n) \\ \texttt{sumF}(n, \lambda\, x.f(x) + g(x)) &= \texttt{sumF}(n, f) + \texttt{sumF}(n, g) \end{aligned}$$

id ist die Identitätsfunktion ($\mathbf{id}(n) = n$) und in PVS vordefiniert.

c) Definieren Sie die Funktionen `square(n:nat) : nat` und `cube(n:nat) : nat` mit $\texttt{square}(n) = n^2$ und $\texttt{cube}(n) = n^3$.
Zeigen Sie:

$$\begin{aligned} \texttt{sumF}(n, \texttt{square}) &= \frac{1}{6} n * (n+1) * (2n+1) \\ \texttt{sumF}(n, \texttt{cube}) &= \texttt{square}(\texttt{sum}(n)) \end{aligned}$$

d) Definieren Sie eine Funktion `exp2(n:nat): RECURSIVE nat`, mit $\texttt{exp2}(n) = 2^n$. Zeigen Sie: $\texttt{sumF}(n, \texttt{exp2}) = \texttt{exp2}(n+1) - 1$.

e) Verallgemeinern Sie die Funktion `sumF` und definieren in PVS eine rekursive Funktion `sumFI` mit der Signatur `sumI(m:nat,n:nat,f:[nat -> nat]) : RECURSIVE nat`, so dass $\texttt{sumFI}(m, n, f)$ die Summe der Funktionswerte von f in dem Intervall zwischen m und n bildet:

$$\texttt{sumFI}(m, n, f) = \sum_{i=m}^{n} f(i)$$

Zeigen Sie $\texttt{sumFI}(0, n, f) = \texttt{sumF}(n, f)$.

Beweisen Sie ferner:

$$\begin{aligned} \texttt{sumFI}(m, n, \lambda\, i.x * f(i)) &= x * \texttt{sumFI}(m, n, f) \\ \texttt{sumFI}(m, n, \lambda\, i.f(i + x)) &= \texttt{sumFI}(m + x, n + x, f) \\ \mid \texttt{sumFI}(m, n, f) \mid &\leq \texttt{sumFI}(m, n, \lambda\, i. \mid f(i) \mid) \end{aligned}$$

Hinweis: Bei der letzten Aussage benötigen Sie vermutlich als Hilfslemma die sogenannte „Dreiecksungleichung“:

$$\mid x + y \mid \;\leq\; \mid x \mid + \mid y \mid$$

Aufgabe 4-3

Definieren Sie eine rekursive Funktion `iterate(f:[nat->nat], n:nat) : RECURSIVE [nat -> nat]` in PVS, die eine gegebene Funktion f genau n-mal iteriert:

$$iterate(f, n)(x) = \underbrace{f(\ldots f(f(}_{n-mal} x)) \ldots)$$

a) Beweisen Sie nach der Typprüfung die enstandenen TCCs.

b) Zeigen Sie (`o` ist die Funktionskomposition und in PVS bereits vordefiniert):

```
iterate_compose : THEOREM
  (iterate(f,n) o iterate(f,m))(x) = iterate(f,n+m)(x)
```

c) Zeigen Sie: Die m-fache Iteration der Funktion $\lambda\, x.x + n$ entspricht der Multiplikation $n \cdot m$.

```
iterate_add : THEOREM
  iterate((LAMBDA (x): x + n), m)(0) = n * m
```

d) Durch Iteration welcher Funktion erhält man die Exponentiation n^m? Formulieren Sie eine entsprechende Aussage und beweisen Sie diese. (Die Exponentiation `n^m` ist in PVS ebenfalls vordefiniert.)