**Aufgabe 8-3:**

a)

Sei D(k) eine Funktion, die die Differenz zwischen *n* und der neuen Eingabe (für sumFromTo) nach *k*-ten Aufruf von sumFromTo berechnet.

Hier ist *n* eine Konstante, *k* ist eine Variable.

Im Allgemeinen gilt: **D(k) = n - (m + k) = - k + n – m**

Je größer *k* ist, desto kleiner wird D(k). Diese Funktion D(k) fällt strikt monoton mit jedem rekursiven Aufruf von sumFromTo. Wenn D(k) negativ ist, wird der Basisfall von sumFromTo erreicht.

Also terminiert die Funktion sumFromTo.

b)

Sei S(i) eine Funktion, die die Summe der neuen Eingaben und (für binomi) nach *i*-ten Aufruf von binomi berechnet. Jeder Aufruf ruft binomi mit unterschiedlichen Parameter 2-mal auf.

Hier sind *n* und *k* Konstanten, *i* ist eine Variable.

Im Allgemeinen gilt: **S(i) = (n – i) + (k – i) + (n – i) + k = -3i + 2n + 2k**

Je größer *i* ist, desto kleiner wird S(i). Diese Funktion S(i) fällt strikt monoton mit jedem rekursiven Aufruf von binomi. Der Basisfall von binomi wird erreicht, wenn i gleich k oder i gleich (n-k).

Also terminiert die Funktion binomi.

c)

Sei D(k) eine Funktion, die die Differenz zwischen *x* und dem Quadrat der neuen Eingabe (für sqrt/above) nach *k*-ten Aufruf von sqrt/above berechnet.

Hier ist *x* eine Konstante, *k* ist eine Variable.

Im Allgemeinen gilt: **D(k) = x – (t + k)^2 = -k^2 – 2\*t\*k + x – t^2**

Je größer *k* ist, desto kleiner wird D(k). Diese Funktion D(k) fällt strikt monoton mit jedem rekursiven Aufruf von sqrt/above. Wenn D(k) negativ ist, wird der Basisfall von sqrt/above und somit sqrt/nat erreicht.

Also terminiert die Funktion sqrt/nat.