

Übungsblatt 6, Aufgabe 2 b

(1) **let fun** $iter (n:int) (f:int \rightarrow int) (x:int) : int =$
 if $n = 0$ **then** x **else** $f (iter (n - 1) f x)$
 in $iter\ 2 (\lambda x:int. x * x)$ 3
 ::
int

per Definition des syntaktischen Zuckers aus (2)

(2) **let** $iter = \mathbf{rec}\ iter : int \rightarrow (int \rightarrow int) \rightarrow int \rightarrow int.$
 $\lambda n:int. \lambda f:int \rightarrow int. \lambda x:int. \mathbf{if} = n\ 0$ **then** x **else** $f (iter (-\ n\ 1) f x)$
in $iter\ 2 (\lambda x:int. *\ x\ x)$ 3
 ::
int

mit Regel (LET) aus (3) (27)

(3) **rec** $iter : int \rightarrow (int \rightarrow int) \rightarrow int \rightarrow int.$::
 $\lambda n:int. \lambda f:int \rightarrow int. \lambda x:int. \mathbf{if} = n\ 0$ **then** x **else** $f (iter (-\ n\ 1) f x)$
 int $\rightarrow (int \rightarrow int) \rightarrow int \rightarrow int$

mit Regel (REC) aus (4)

(4) $[iter : int \rightarrow (int \rightarrow int) \rightarrow int \rightarrow int] \triangleright \lambda n:int. \lambda f:int \rightarrow int. \lambda x:int. \mathbf{if} = n\ 0$ **then** x **else** $f (iter (-\ n\ 1) f x) ::$
 int $\rightarrow (int \rightarrow int) \rightarrow int \rightarrow int$

mit Regel (ABSTR) aus (5)

(5) $[iter : int \rightarrow (int \rightarrow int) \rightarrow int \rightarrow int, n:int] \triangleright \lambda f:int \rightarrow int. \lambda x:int. \mathbf{if} = n\ 0$ **then** x **else** $f (iter (-\ n\ 1) f x) ::$
 (int $\rightarrow int) \rightarrow int \rightarrow int$

mit Regel (ABSTR) aus (6)

(6) $[iter : int \rightarrow (int \rightarrow int) \rightarrow int \rightarrow int, n:int, f:int \rightarrow int] \triangleright \lambda x:int. \mathbf{if} = n\ 0$ **then** x **else** $f (iter (-\ n\ 1) f x) ::$
 int $\rightarrow int$

mit Regel (ABSTR) aus (7)

(7) $[iter : int \rightarrow (int \rightarrow int) \rightarrow int \rightarrow int, n:int, f:int \rightarrow int, x:int] \triangleright \mathbf{if} = n\ 0$ **then** x **else** $f (iter (-\ n\ 1) f x) ::$
 int

mit Regel (COND) aus (8) (13) (14)

(8) $[iter : int \rightarrow (int \rightarrow int) \rightarrow int \rightarrow int, n:int, f:int \rightarrow int, x:int] \triangleright = n\ 0 ::$
 bool

mit Regel (APP) aus (9) (12)

(9) $[iter : int \rightarrow (int \rightarrow int) \rightarrow int \rightarrow int, n:int, f:int \rightarrow int, x:int] \triangleright = n ::$
 int $\rightarrow bool$

mit Regel (APP) aus (10) (11)

(10) $[iter : int \rightarrow (int \rightarrow int) \rightarrow int \rightarrow int, n:int, f:int \rightarrow int, x:int] \triangleright = ::$
 int $\rightarrow int \rightarrow bool$

mit Regel (EQ)

(11) $[iter : int \rightarrow (int \rightarrow int) \rightarrow int \rightarrow int, n:int, f:int \rightarrow int, x:int] \triangleright n ::$
 int

mit Regel (ID)

(12) $[iter : int \rightarrow (int \rightarrow int) \rightarrow int \rightarrow int, n:int, f:int \rightarrow int, x:int] \triangleright 0 ::$
 int

mit Regel (CONST) und (INT)

(13) $[iter : int \rightarrow (int \rightarrow int) \rightarrow int \rightarrow int, n:int, f:int \rightarrow int, x:int] \triangleright x ::$
 int

mit Regel (ID)

(14) $[iter : int \rightarrow (int \rightarrow int) \rightarrow int \rightarrow int, n:int, f:int \rightarrow int, x:int] \triangleright f (iter (-\ n\ 1) f x) ::$
 int

mit Regel (APP) aus (15) (16)

(15) $[iter : int \rightarrow (int \rightarrow int) \rightarrow int \rightarrow int, n:int, f:int \rightarrow int, x:int] \triangleright f ::$
 int $\rightarrow int$

mit Regel (ID)

(16) $[iter : int \rightarrow (int \rightarrow int) \rightarrow int \rightarrow int, n:int, f:int \rightarrow int, x:int] \triangleright iter (-\ n\ 1) f x ::$
 int

mit Regel (APP) aus (17) (26)

(17) $[iter : int \rightarrow (int \rightarrow int) \rightarrow int \rightarrow int, n:int, f:int \rightarrow int, x:int] \triangleright iter (-\ n\ 1) f ::$
 int $\rightarrow int$

mit Regel (APP) aus (18) (25)

(18) $[iter : int \rightarrow (int \rightarrow int) \rightarrow int \rightarrow int, n:int, f:int \rightarrow int, x:int] \triangleright iter (-\ n\ 1) ::$
 (int $\rightarrow int) \rightarrow int \rightarrow int$

