

Typechecking

Beispiel zur λ -Abstraktion mit einem Argument vom Produkttyp: `sum_of_squares`

(1) `let val sum_of_squares = $\lambda z:\text{int} * \text{int}.$ + (* (#1 z, #1 z), * (#2 z, #2 z))`
`in sum_of_squares (3, 4)`
`end`
`:: int`

mit Regel (LET-VAL) aus (2) (24)

(2) $\lambda z:\text{int} * \text{int}.$ + (* (#1 z, #1 z), * (#2 z, #2 z)) :: $\text{int} * \text{int} \rightarrow \text{int}$

mit Regel (ABSTR) aus (3)

(3) $[z:\text{int} * \text{int}] \triangleright + (* (\#1 z, \#1 z), * (\#2 z, \#2 z)) :: \text{int}$

mit Regel (APP) aus (4) (5)

(4) $[z:\text{int} * \text{int}] \triangleright + :: \text{int} * \text{int} \rightarrow \text{int}$

mit Regel (AOP)

(5) $[z:\text{int} * \text{int}] \triangleright (* (\#1 z, \#1 z), * (\#2 z, \#2 z)) :: \text{int} * \text{int}$

mit Regel (TUPLE) aus (6) (15)

(6) $[z:\text{int} * \text{int}] \triangleright * (\#1 z, \#1 z) :: \text{int}$

mit Regel (APP) aus (7) (8)

(7) $[z:\text{int} * \text{int}] \triangleright * :: \text{int} * \text{int} \rightarrow \text{int}$

mit Regel (AOP)

(8) $[z:\text{int} * \text{int}] \triangleright (\#1 z, \#1 z) :: \text{int} * \text{int}$

mit Regel (TUPLE) aus (9) (12)

(9) $[z:\text{int} * \text{int}] \triangleright \#1 z :: \text{int}$

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

(10) $[z:\text{int} * \text{int}] \triangleright \#1 :: \text{int} * \text{int} \rightarrow \text{int}$

mit Regel (PROJ)

(11) $[z:\text{int} * \text{int}] \triangleright z :: \text{int} * \text{int}$

mit Regel (ID)

(12) $[z:\text{int} * \text{int}] \triangleright \#1 z :: \text{int}$

mit Regel (APP) aus (13) (14)

(13) $[z:\text{int} * \text{int}] \triangleright \#1 :: \text{int} * \text{int} \rightarrow \text{int}$

mit Regel (PROJ)

(14) $[z:\text{int} * \text{int}] \triangleright z :: \text{int} * \text{int}$

mit Regel (ID)

(15) $[z:\text{int} * \text{int}] \triangleright * (\#2 z, \#2 z) :: \text{int}$

mit Regel (APP) aus (16) (17)

(16) $[z:\text{int} * \text{int}] \triangleright * :: \text{int} * \text{int} \rightarrow \text{int}$

mit Regel (AOP)

(17) $[z:\text{int} * \text{int}] \triangleright (\#2 z, \#2 z) :: \text{int} * \text{int}$

mit Regel (TUPLE) aus (18) (21)

(18) $[z:\text{int} * \text{int}] \triangleright \#2 z :: \text{int}$

mit Regel (APP) aus (19) (20)

(19) $[z:\text{int} * \text{int}] \triangleright \#2 :: \text{int} * \text{int} \rightarrow \text{int}$

mit Regel (PROJ)

(20) $[z:\text{int} * \text{int}] \triangleright z :: \text{int} * \text{int}$

mit Regel (ID)

(21) $[z:\text{int} * \text{int}] \triangleright \#2 z :: \text{int}$

mit Regel (APP) aus (22) (23)

(22) $[z:\text{int} * \text{int}] \triangleright \#2 :: \text{int} * \text{int} \rightarrow \text{int}$

mit Regel (PROJ)

(23) $[z:\text{int} * \text{int}] \triangleright z :: \text{int} * \text{int}$

mit Regel (ID)

(24) $[sum_of_squares:\text{int} * \text{int} \rightarrow \text{int}] \triangleright sum_of_squares (3, 4) :: \text{int}$

mit Regel (APP) aus (25) (26)

(25) $[sum_of_squares:\text{int} * \text{int} \rightarrow \text{int}] \triangleright sum_of_squares :: \text{int} * \text{int} \rightarrow \text{int}$

mit Regel (ID)

(26) $[sum_of_squares:\text{int} * \text{int} \rightarrow \text{int}] \triangleright (3, 4) :: \text{int} * \text{int}$

mit Regel (TUPLE) aus (27) (28)

(27) $[sum_of_squares:\text{int} * \text{int} \rightarrow \text{int}] \triangleright 3 :: \text{int}$

mit Regel (INT)

(28) $[sum_of_squares:\text{int} * \text{int} \rightarrow \text{int}] \triangleright 4 :: \text{int}$

mit Regel (INT)