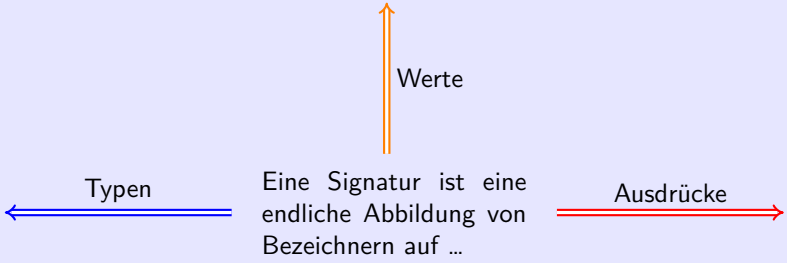
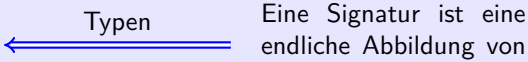
Teil III

Werte I

5. Quiz: Signaturen

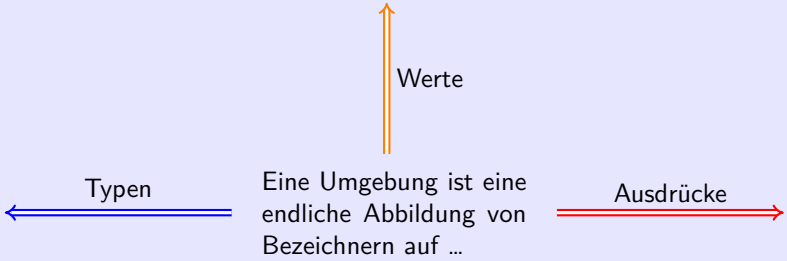


5. Quiz: Signaturen



Eine Signatur ist eine endliche Abbildung von Bezeichnern auf ...

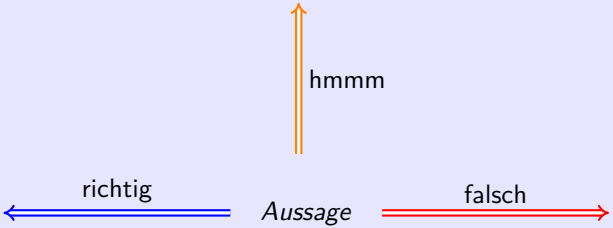
5. Quiz: Umgebungen



5. Quiz: Umgebungen

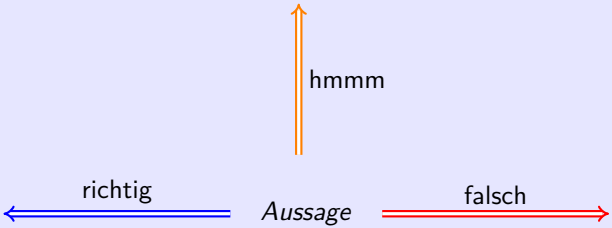
Eine Umgebung ist eine endliche Abbildung von Bezeichnern auf ...

5. Quiz: Ausdrücke und Definitionen



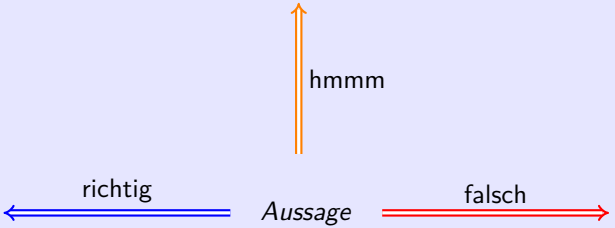
$e \leqslant 2$ ist ein Ausdruck.

5. Quiz: Ausdrücke und Definitionen



richtig ⟸ $e \leqslant 2$ ist ein Ausdruck.

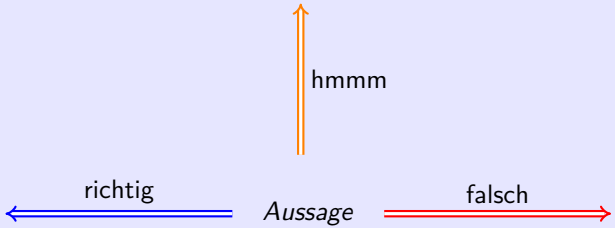
5. Quiz: Ausdrücke und Definitionen



. . . . . .

let $e \leqslant 2$ ist eine Definition.

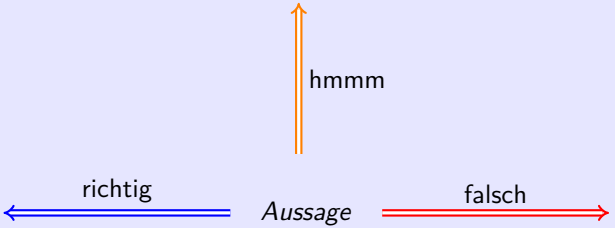
5. Quiz: Ausdrücke und Definitionen



let $e \leqslant 2$ ist eine Definition.

falsch

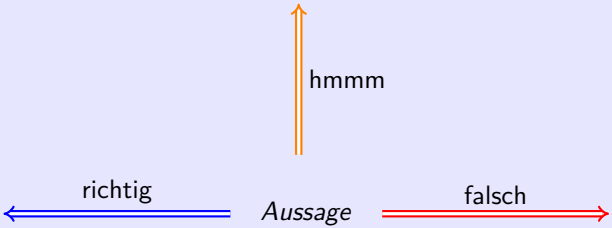
5. Quiz: Ausdrücke und Definitionen



. . . . . .

let $e = 2$ ist eine Definition.

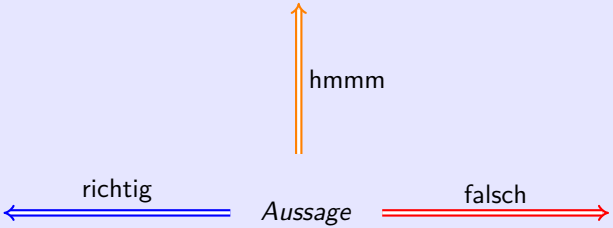
5. Quiz: Ausdrücke und Definitionen



richtig

let $e = 2$ ist eine Definition.

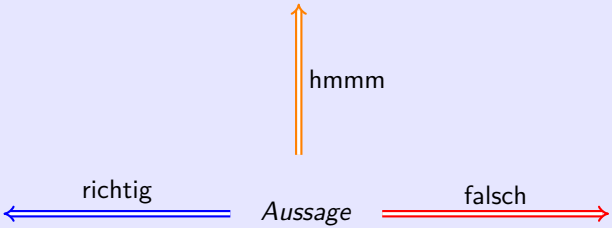
5. Quiz: Ausdrücke und Definitionen



. . . . . .

let $e = 2$ ***in*** e ist ein Ausdruck.

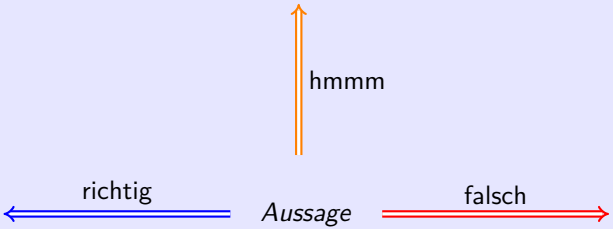
5. Quiz: Ausdrücke und Definitionen



richtig

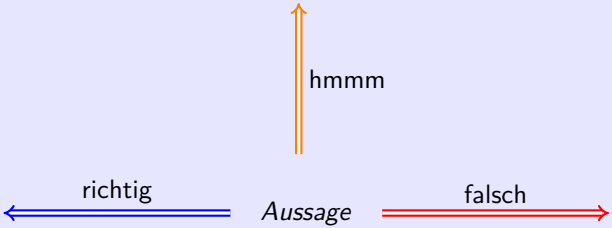
let $e = 2$ ***in*** e ist ein Ausdruck.

5. Quiz: Ausdrücke und Definitionen



Das ***in***-Konstrukt verbindet Definitionen und Ausdrücke.

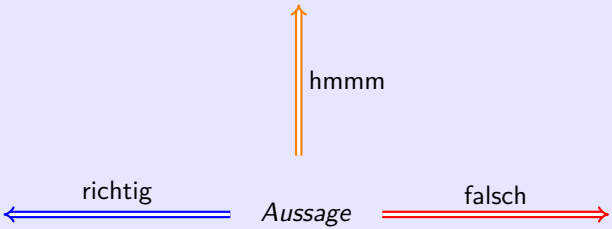
5. Quiz: Ausdrücke und Definitionen



Das ***in***-Konstrukt verbindet Definitionen und Ausdrücke.

5. Quiz: Ausdrücke und Definitionen

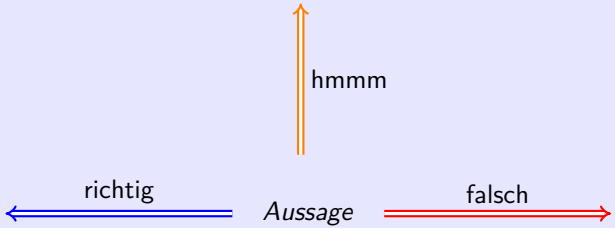
$$(\textbf{\textit{let}}\ x = 2\ \textbf{\textit{in fun}}\ (y : \mathit{Nat}) \to y + x)\ (x\ \%\ 5)$$



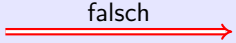
An der Wurzel des Syntaxbaums für den Ausdruck steht ***let*** $x = 2$.

5. Quiz: Ausdrücke und Definitionen

$$(\textbf{\textit{let}}\ x = 2\ \textbf{\textit{in}}\ \textbf{\textit{fun}}\ (y : \mathit{Nat}) \to y + x)\ (x \,\%\, 5)$$

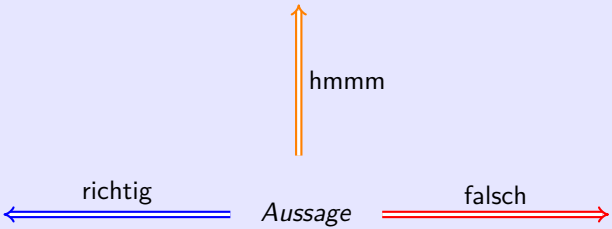


An der Wurzel des Syntaxbaums für den Ausdruck steht ***let*** $x = 2$.



5. Quiz: Ausdrücke und Definitionen

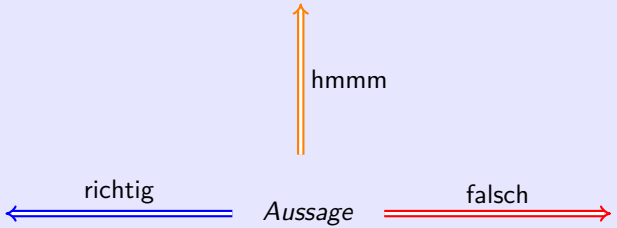
$$(\textbf{\textit{let}}\ x = 2\ \textbf{\textit{in}}\ \textbf{\textit{fun}}\ (y : \mathit{Nat}) \to y + x)\ (x\ \%\ 5)$$



An der Wurzel des Syntaxbaums für den Ausdruck steht das ***in***-Konstrukt.

5. Quiz: Ausdrücke und Definitionen

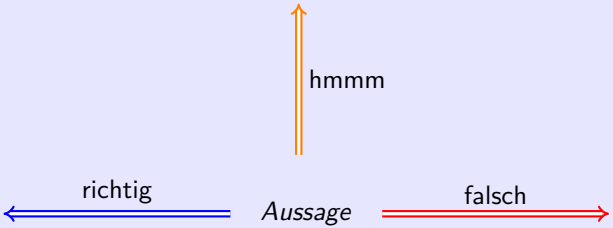
$$(\textbf{\textit{let}}\ x = 2\ \textbf{\textit{in}}\ \textbf{\textit{fun}}\ (y : \mathit{Nat}) \to y + x)\ (x \,\%\, 5)$$



An der Wurzel des Syntaxbaums für den Ausdruck steht das ***in***-Konstrukt.

5. Quiz: Ausdrücke und Definitionen

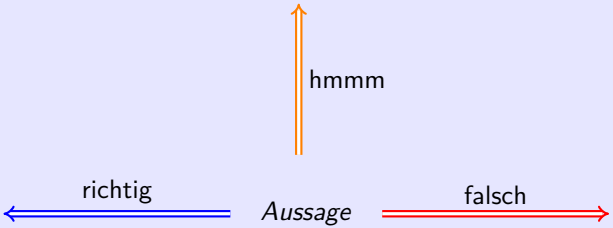
$$(\textbf{\textit{let}}\ x = 2\ \textbf{\textit{in fun}}\ (y : \mathit{Nat}) \to y + x)\ (x\ \%\ 5)$$



An der Wurzel des Syntaxbaums für den Ausdruck steht die Funktionsapplikation.

5. Quiz: Ausdrücke und Definitionen

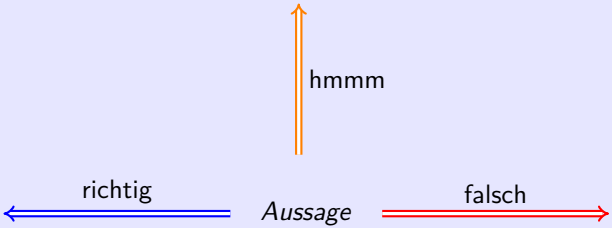
$$(\textbf{\textit{let}}\ x = 2\ \textbf{\textit{in fun}}\ (y : \mathit{Nat}) \to y + x)\ (x\ \%\ 5)$$



An der Wurzel des Syntaxbaums für den Ausdruck steht die Funktionsapplikation.

5. Quiz: Ausdrücke und Definitionen

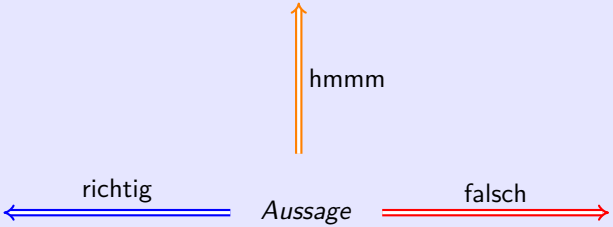
$$(\textbf{\textit{let}}\ x = 2\ \textbf{\textit{in}}\ \textbf{\textit{fun}}\ (y : \mathit{Nat}) \to y + x)\ (x \,\%\, 5)$$



Der Ausdruck ist unter der leeren Signatur $\varnothing$ typkorrekt.

5. Quiz: Ausdrücke und Definitionen

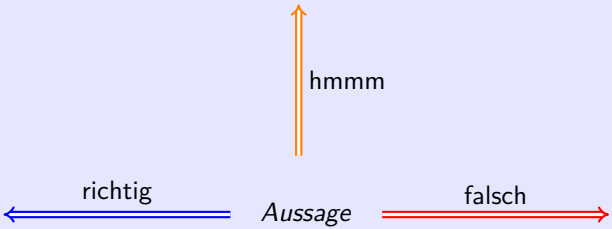
$$(\textbf{\textit{let}}\ x = 2\ \textbf{\textit{in fun}}\ (y : \mathit{Nat}) \to y + x)\ (x\ \%\ 5)$$



Der Ausdruck ist unter der leeren Signatur $\varnothing$ typkorrekt.

5. Quiz: Ausdrücke und Definitionen

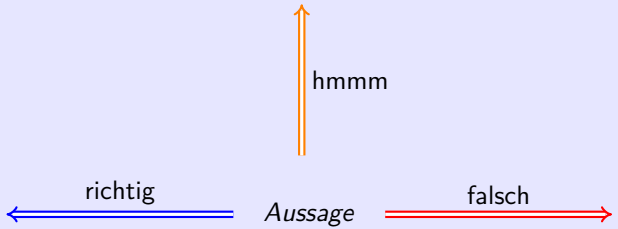
$$(\textbf{\textit{let}}\ x = 2\ \textbf{\textit{in fun}}\ (y : \mathit{Nat}) \to y + x)\ (x \,\%\, 5)$$



Der Ausdruck wertet zu 4 aus.

5. Quiz: Ausdrücke und Definitionen

$$(\textbf{\textit{let}}\ x = 2\ \textbf{\textit{in fun}}\ (y : \mathit{Nat}) \to y + x)\ (x\,\%\,5)$$



Der Ausdruck wertet zu 4 aus.

6. A tribute to Alonzo Church (1903—1995)

I never had any mathematical conversations with anybody, because there was nobody else in my field.
— Alonzo Church

6. Boolesche Werte — da capo

Zur Erinnerung: wir haben Abkürzungen für Negation, Konjunktion und Disjunktion eingeführt.

$$\begin{aligned} \mathit{not}\ a &= \textbf{\textit{if}}\ a\ \textbf{\textit{then}}\ \mathit{false}\ \textbf{\textit{else}}\ \mathit{true} \\ a\ \&\&\ b &= \textbf{\textit{if}}\ a\ \textbf{\textit{then}}\ b\ \textbf{\textit{else}}\ \mathit{false} \\ a\ ||\ b &= \textbf{\textit{if}}\ a\ \textbf{\textit{then}}\ \mathit{true}\ \textbf{\textit{else}}\ b \end{aligned}$$

Was ist mit Implikation ($\rightarrow$, $\leq$), Nicht-Umkehrimplikation ($<$), Äquivalenz ($\leftrightarrow$, $=$) und Nicht-Äquivalenz ($\neq$)?

$$\begin{aligned} a \rightarrow b &= \\ a < b &= \\ a = b &= \\ a \neq b &= \end{aligned}$$

6. Boolesche Werte — da capo

Zur Erinnerung: wir haben Abkürzungen für Negation, Konjunktion und Disjunktion eingeführt.

$$\begin{aligned} \mathit{not}\ a &= \textbf{\textit{if}}\ a\ \textbf{\textit{then}}\ \mathit{false}\ \textbf{\textit{else}}\ \mathit{true} \\ a\ \&\&\ b &= \textbf{\textit{if}}\ a\ \textbf{\textit{then}}\ b\ \textbf{\textit{else}}\ \mathit{false} \\ a\ ||\ b &= \textbf{\textit{if}}\ a\ \textbf{\textit{then}}\ \mathit{true}\ \textbf{\textit{else}}\ b \end{aligned}$$

Was ist mit Implikation ($\rightarrow$, $\leq$), Nicht-Umkehrimplikation ($<$), Äquivalenz ($\leftrightarrow$, $=$) und Nicht-Äquivalenz ($\neq$)?

$$\begin{aligned} a \rightarrow b &= \textbf{\textit{if}}\ a\ \textbf{\textit{then}}\ b\ \textbf{\textit{else}}\ \mathit{true} \\ a < b &= \\ a = b &= \\ a \neq b &= \end{aligned}$$

6. Boolesche Werte — da capo

Zur Erinnerung: wir haben Abkürzungen für Negation, Konjunktion und Disjunktion eingeführt.

$$\begin{aligned} \mathit{not}\ a &= \textbf{\textit{if}}\ a\ \textbf{\textit{then}}\ \mathit{false}\ \textbf{\textit{else}}\ \mathit{true} \\ a\ \&\&\ b &= \textbf{\textit{if}}\ a\ \textbf{\textit{then}}\ b\ \textbf{\textit{else}}\ \mathit{false} \\ a\ ||\ b &= \textbf{\textit{if}}\ a\ \textbf{\textit{then}}\ \mathit{true}\ \textbf{\textit{else}}\ b \end{aligned}$$

Was ist mit Implikation ($\rightarrow$, $\leq$), Nicht-Umkehrimplikation ($<$), Äquivalenz ($\leftrightarrow$, $=$) und Nicht-Äquivalenz ($\neq$)?

$$\begin{aligned} a \rightarrow b &= \textbf{\textit{if}}\ a\ \textbf{\textit{then}}\ b\ \textbf{\textit{else}}\ \mathit{true} \\ a < b &= \textbf{\textit{if}}\ a\ \textbf{\textit{then}}\ \mathit{false}\ \textbf{\textit{else}}\ b \\ a = b &= \\ a \neq b &= \end{aligned}$$

6. Boolesche Werte — da capo

Zur Erinnerung: wir haben Abkürzungen für Negation, Konjunktion und Disjunktion eingeführt.

$$\begin{aligned} \mathit{not}\ a &= \textbf{\textit{if}}\ a\ \textbf{\textit{then}}\ \mathit{false}\ \textbf{\textit{else}}\ \mathit{true} \\ a\ \&\&\ b &= \textbf{\textit{if}}\ a\ \textbf{\textit{then}}\ b\ \textbf{\textit{else}}\ \mathit{false} \\ a\ ||\ b &= \textbf{\textit{if}}\ a\ \textbf{\textit{then}}\ \mathit{true}\ \textbf{\textit{else}}\ b \end{aligned}$$

Was ist mit Implikation ($\rightarrow$, $\leq$), Nicht-Umkehrimplikation ($<$), Äquivalenz ($\leftrightarrow$, $=$) und Nicht-Äquivalenz ($\neq$)?

$$\begin{aligned} a \rightarrow b &= \textbf{\textit{if}}\ a\ \textbf{\textit{then}}\ b\ \textbf{\textit{else}}\ \mathit{true} \\ a < b &= \textbf{\textit{if}}\ a\ \textbf{\textit{then}}\ \mathit{false}\ \textbf{\textit{else}}\ b \\ a = b &= \textbf{\textit{if}}\ a\ \textbf{\textit{then}}\ b\ \textbf{\textit{else}}\ \mathit{not}\ b \\ a \neq b &= \end{aligned}$$

6. Boolesche Werte — da capo

Zur Erinnerung: wir haben Abkürzungen für Negation, Konjunktion und Disjunktion eingeführt.

$$\begin{aligned} \mathit{not}\ a &= \mathbf{if}\ a\ \mathbf{then}\ \mathit{false}\ \mathbf{else}\ \mathit{true} \\ a\ \&\&\ b &= \mathbf{if}\ a\ \mathbf{then}\ b\ \mathbf{else}\ \mathit{false} \\ a\ ||\ b &= \mathbf{if}\ a\ \mathbf{then}\ \mathit{true}\ \mathbf{else}\ b \end{aligned}$$

Was ist mit Implikation ($\rightarrow$, $\leq$), Nicht-Umkehrimplikation ($<$), Äquivalenz ($\leftrightarrow$, $=$) und Nicht-Äquivalenz ($\neq$)?

$$\begin{aligned} a \rightarrow b &= \mathbf{if}\ a\ \mathbf{then}\ b\ \mathbf{else}\ \mathit{true} \\ a < b &= \mathbf{if}\ a\ \mathbf{then}\ \mathit{false}\ \mathbf{else}\ b \\ a = b &= \mathbf{if}\ a\ \mathbf{then}\ b\ \mathbf{else}\ \mathit{not}\ b \\ a \neq b &= \mathbf{if}\ a\ \mathbf{then}\ \mathit{not}\ b\ \mathbf{else}\ b \end{aligned}$$

6. Daten als Programme

Folks, now that you've got functions, why not ditch Booleans and natural numbers?

Ein Wahrheitswert ist entweder wahr (1) oder falsch (0).

Eine natürliche Zahl ist entweder 0 oder der Nachfolger einer natürlichen Zahl $(n + 1)$.

6. Wahrheitswerte als Funktionen

true

⇓

fun *true false* → *true*

fun *a b* → *a*

false

⇓

fun *true false* → *false*

fun *a b* → *b*

☞ Ausnahmsweise ohne Angaben von Typen ...

6. Wahrheitswerte als Funktionen

true

false

fun *true false* $\rightarrow$ *true*

fun *true false* $\rightarrow$ *false*

fun *a b* $\rightarrow$ *a*

fun *a b* $\rightarrow$ *b*

☞ Ausnahmsweise ohne Angaben von Typen ...

6. Wahrheitswerte als Funktionen

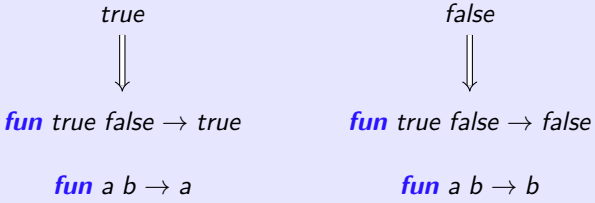
fun $a\ b \rightarrow a$ *fun* $a\ b \rightarrow b$

☞ Ausnahmsweise ohne Angaben von Typen ...

6. Wahrheitswerte als Funktionen

☞ Ausnahmsweise ohne Angaben von Typen ...

6. Wahrheitswerte als Funktionen



☞ Ausnahmsweise ohne Angaben von Typen ...

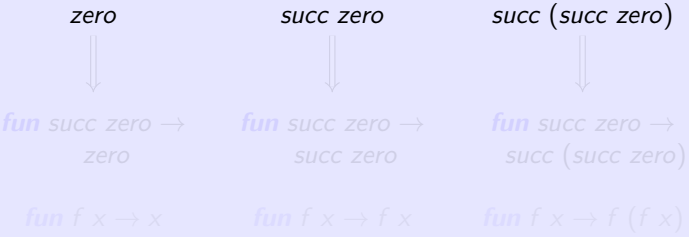
6. Wahrheitswerte als Funktionen

let *false* = ***fun*** *a b* → *b*

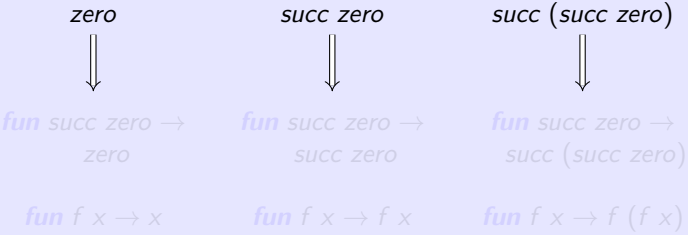
let *true* = ***fun*** *a b* → *a*

let *if-then-else cond x y* = *cond x y*

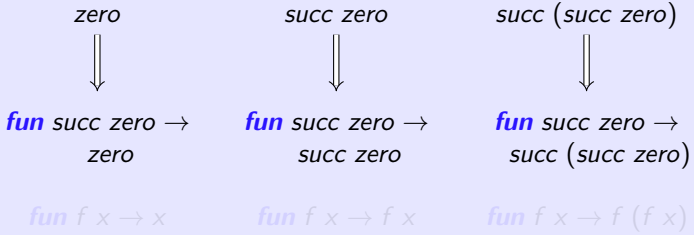
6. Natürliche Zahlen als Funktionen



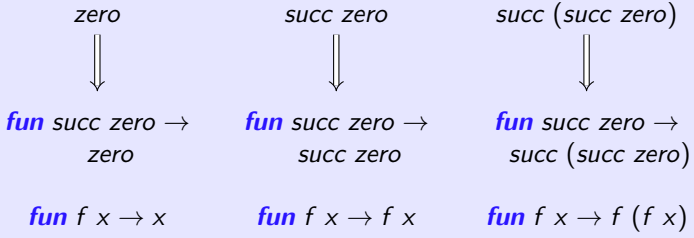
6. Natürliche Zahlen als Funktionen



6. Natürliche Zahlen als Funktionen



6. Natürliche Zahlen als Funktionen



6. Natürliche Zahlen als Funktionen

$$
\begin{aligned}
\mathit{zero} &= \textbf{\textit{fun}}\ f\ x \to x \\
\mathit{succ}\ n &= \textbf{\textit{fun}}\ f\ x \to f\ (n\ f\ x) \\
\mathit{plus}\ n_1\ n_2 &= \textbf{\textit{fun}}\ f\ x \to n_1\ f\ (n_2\ f\ x) \\
\mathit{times}\ n_1\ n_2 &= \textbf{\textit{fun}}\ f\ x \to n_1\ (n_2\ f)\ x
\end{aligned}
$$