# Funktionale Programmierung
## Sommersemester 2021

Prof. Dr. Jakob Rehof
TU Dortmund
LS XIV Software Engineering

# Diese Vorlesung:

- DoberkatFP: Folien 98 – 122
  - Faltungen (fold)
- Homomorphismen

# Homomorphismus

*Homomorphismus* heißt in der Mathematik:

- eine *strukturtreue* (*strukturbewahrende*) Abbildung

Aus dem Altgriechischen:

- *homós* = gleich (ähnlich)
- *morphé* = Form

# Homomorphismus

Seien $\langle \mathbf{A}, \circ \rangle$ und $\langle \mathbf{B}, \bullet \rangle$ zwei algebraische Strukturen mit Operationen (jeweils) $\circ$ und $\bullet$, wobei

$$\circ : \mathbf{A} \times \mathbf{A} \to \mathbf{A}$$

$$\bullet : \mathbf{B} \times \mathbf{B} \to \mathbf{B}$$

Eine Abbildung $\varphi : \mathbf{A} \to \mathbf{B}$ ist dann ein Homomorphismus (von $\mathbf{A}$ nach $\mathbf{B}$), wenn für alle $x, y \in \mathbf{A}$ gilt:

$$\varphi(x \circ y) = \varphi(x) \bullet \varphi(y)$$

Weitere algebraische Struktur wird automatisch erhalten in $\varphi(\mathbf{A}) \subseteq \mathbf{B}$, Beispiel Neutralelement:

$$\varphi(x) = \varphi(1 \circ x) = \varphi(1) \bullet \varphi(x)$$

Also, $\varphi(1) \bullet \varphi(x) = \varphi(x)$.

# Homomorphismus

Setzen wir $(\varphi \times \varphi)(x, y) = (\varphi(x), \varphi(y))$, können wie auch die Homomorphismus-eigenschaft durch die *Kommutatvität des Diagramms ausdrücken:*

$$
\begin{tikzcd}
A \times A \arrow[r, "\varphi \times \varphi"] \arrow[d, "\circ"'] & B \times B \arrow[d, "\bullet"] \\
A \arrow[r, "\varphi"] & B
\end{tikzcd}
$$

# Homomorphismus

Bekannte Homomorphismen

- Exponentialfunktionen. Mit $\varphi(x) = e^x$ haben wir $\varphi(x + y) = \varphi(x) \cdot \varphi(y)$
- Lineare Abbildungen sind Homomorphismen zwischen Vektorräumen: $f(\mathbf{x} + \mathbf{y}) = f(\mathbf{x}) + f(\mathbf{y})$, $af(\mathbf{x}) = f(a\mathbf{x})$
- Substitutionen sind Homomorphismen in Termalgebren: $S(\mathbf{t}(x, y)) = \mathbf{t}(S(x), S(y))$
- Graphhomomorphismus $f : \langle V_G, E_G\rangle \to \langle V_H, E_H\rangle$: $(u, v) \in E_G \Rightarrow (f(u), f(v)) \in E_H$

Das Prinzip auf Datenstrukturen übertragen, zum Beispiel:

- Auf Paaren hätten wir danach $f^*(x, y) = (f(x), f(y))$
- Auf Listen hätten wir danach $f^*(x : xs) = f(x) \bullet f^*(xs)$