

# Systematisches Programmieren

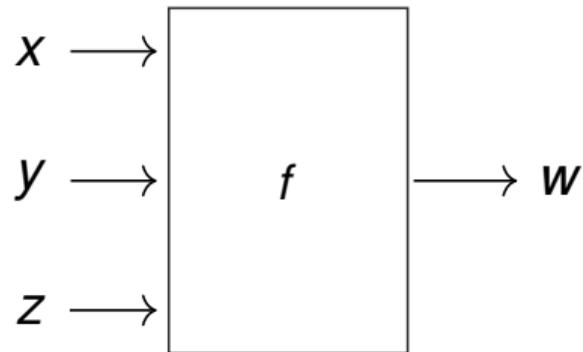
## Funktionen und Ausdrücke

Dr. Maurice Chandoo

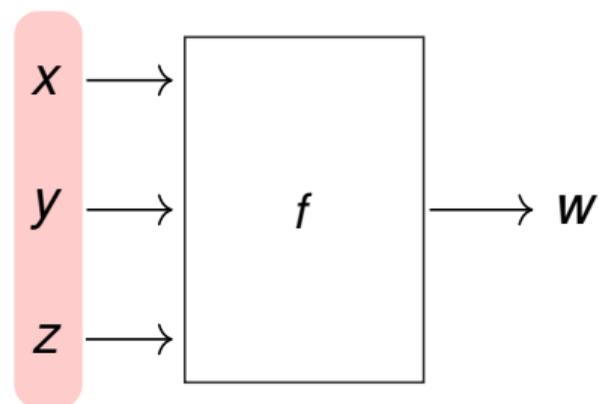
Leibniz Universität Hannover  
Institut für Theoretische Informatik

Sommersemester 2020

# Funktionen in der Programmierung

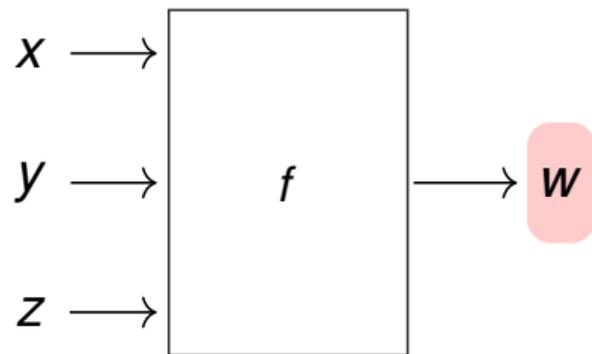


# Funktionen in der Programmierung



Eingabeparameter

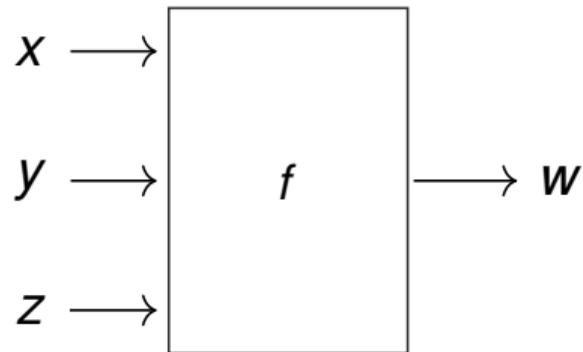
# Funktionen in der Programmierung



Eingabeparameter

Rückgabewert

# Funktionen in der Programmierung

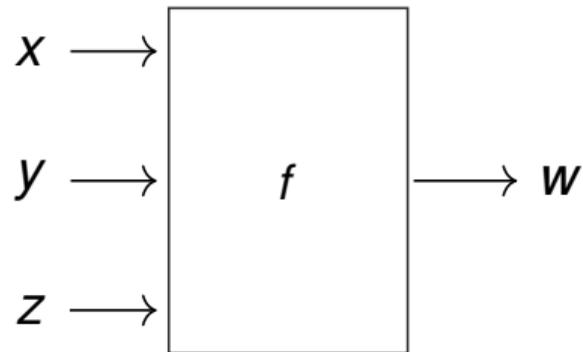


Eingabeparameter

Rückgabewert

# Funktionen in der Programmierung

- ▶ Funktionen sind **eindeutig**

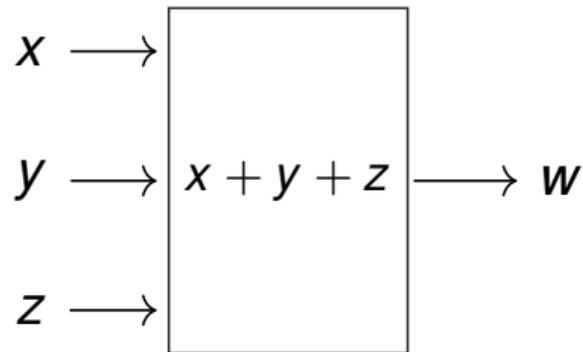


Eingabeparameter

Rückgabewert

# Funktionen in der Programmierung

- ▶ Funktionen sind **eindeutig**

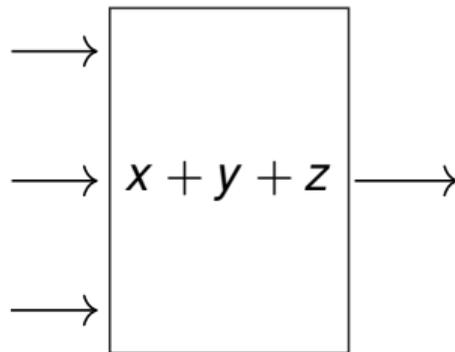


Eingabeparameter

Rückgabewert

# Funktionen in der Programmierung

- ▶ Funktionen sind **eindeutig**

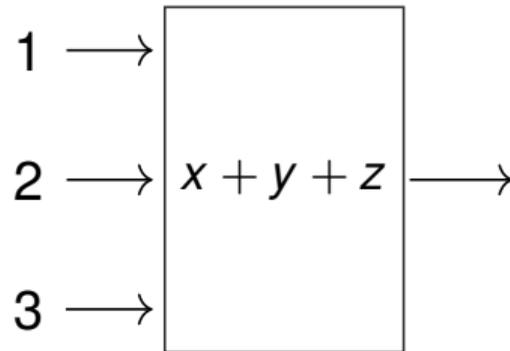


Eingabeparameter

Rückgabewert

# Funktionen in der Programmierung

- ▶ Funktionen sind **eindeutig**

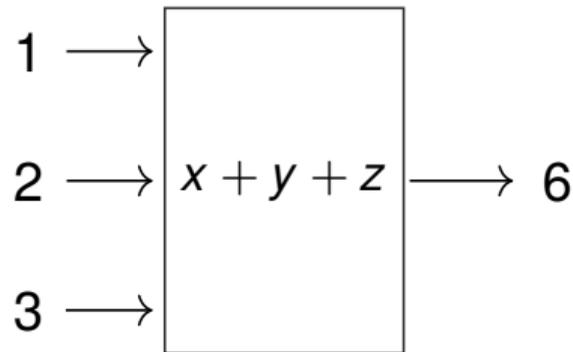


Eingabeparameter

Rückgabewert

# Funktionen in der Programmierung

- ▶ Funktionen sind **eindeutig**

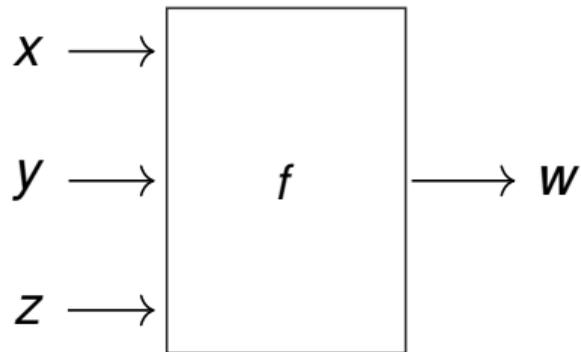


Eingabeparameter

Rückgabewert

# Funktionen in der Programmierung

- ▶ Funktionen sind **eindeutig**

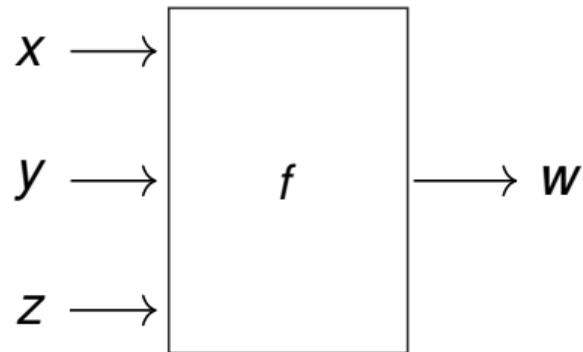


Eingabeparameter

Rückgabewert

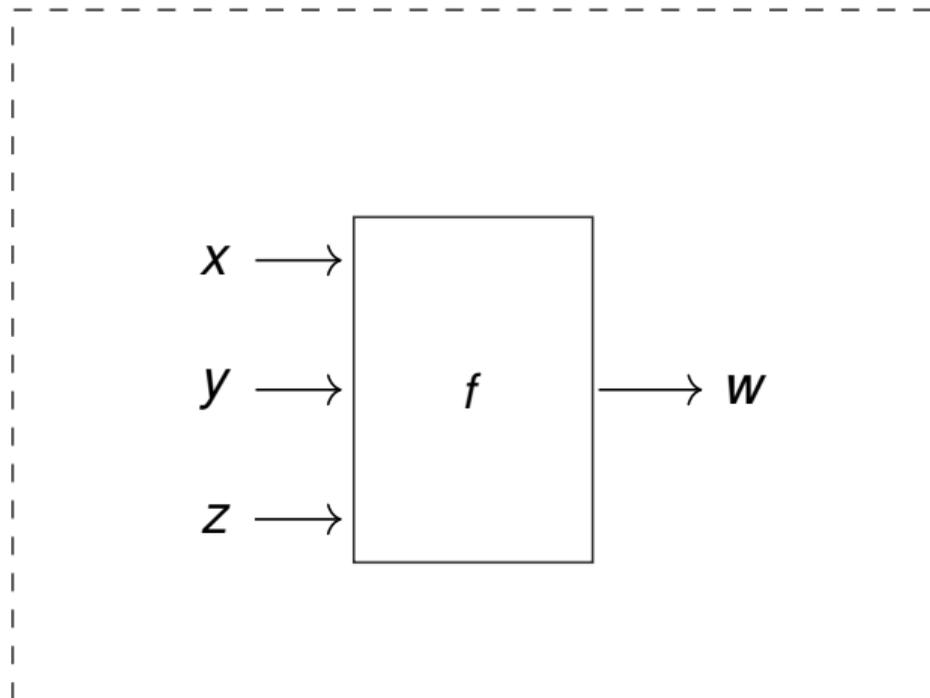
# Funktionen in der Programmierung

- ▶ Funktionen sind **eindeutig**



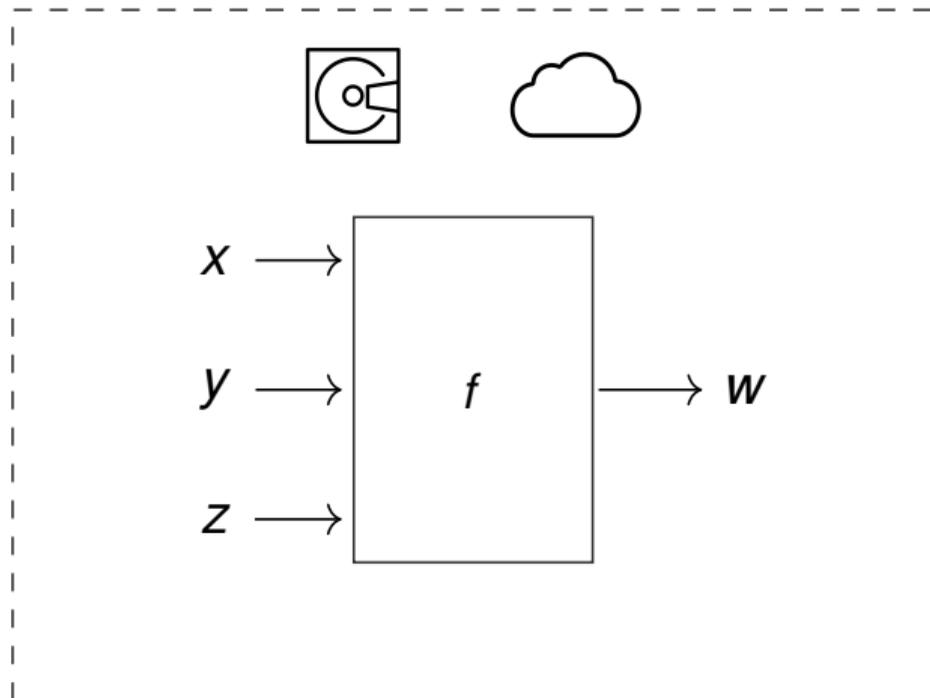
# Funktionen in der Programmierung

- ▶ Funktionen sind **eindeutig**
- ▶ Ausführungsumgebung



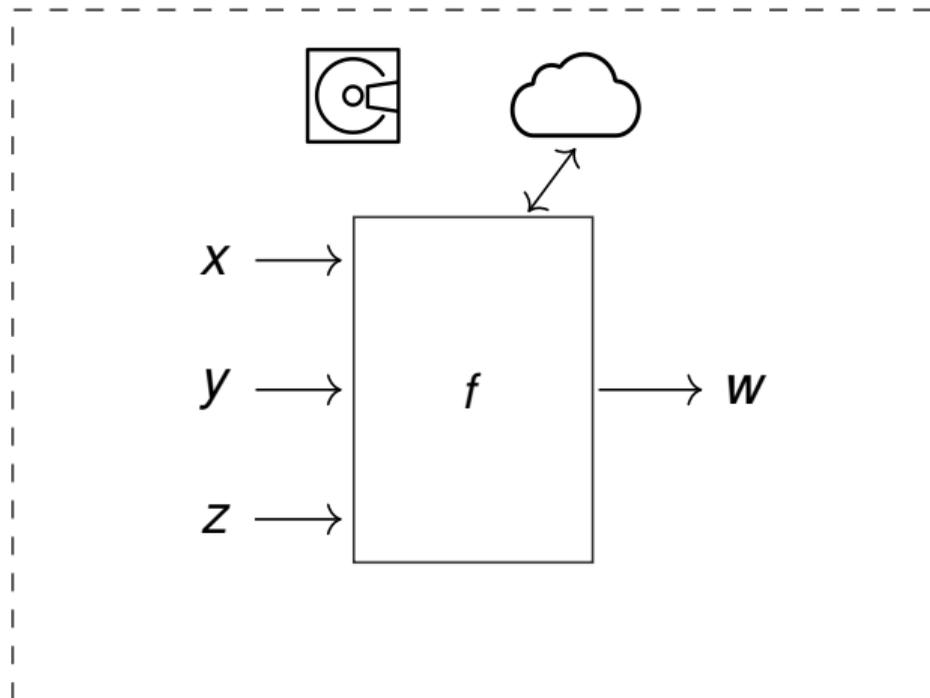
# Funktionen in der Programmierung

- ▶ Funktionen sind **eindeutig**
- ▶ Ausführungsumgebung  
Speicher, Netzwerk, ...



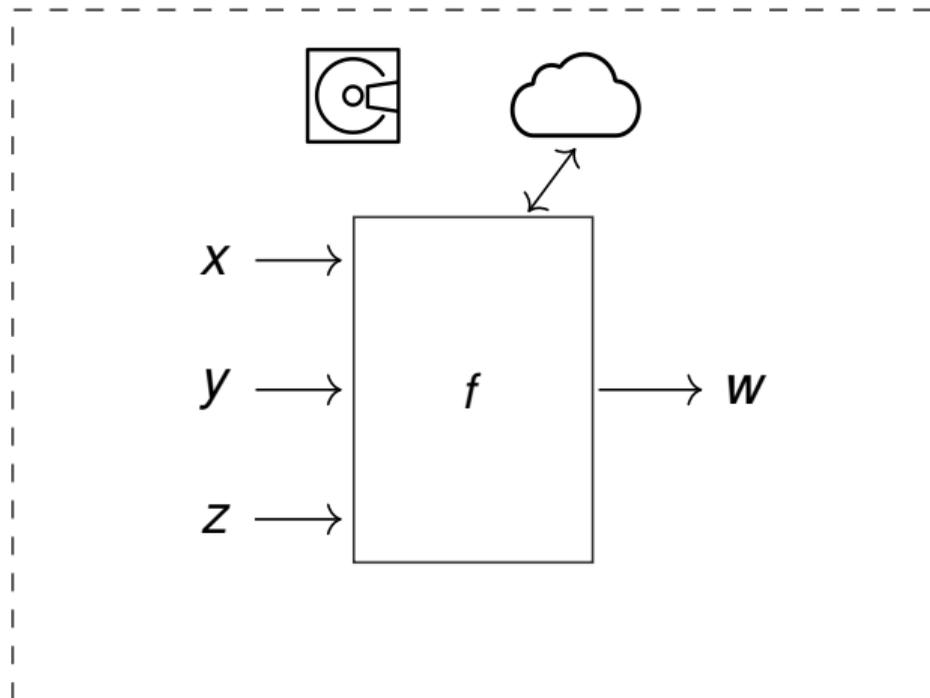
# Funktionen in der Programmierung

- ▶ Funktionen sind **eindeutig**
- ▶ Ausführungsumgebung  
Speicher, Netzwerk, ...



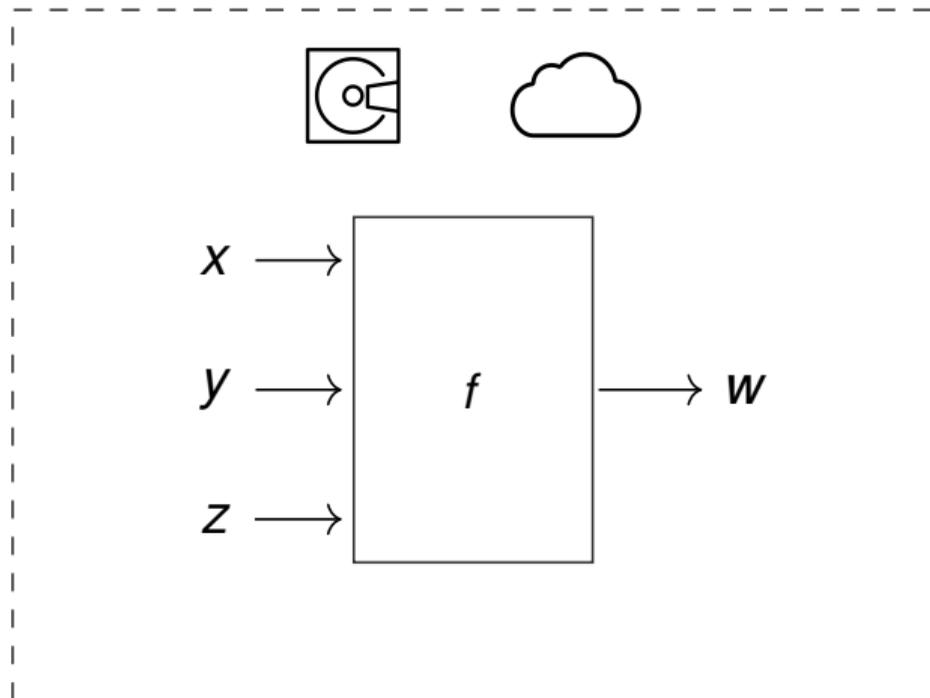
# Funktionen in der Programmierung

- ▶ Funktionen sind **eindeutig**
- ▶ Ausführungsumgebung  
Speicher, Netzwerk, ...
- ▶ Interaktion mit Umgebung kann  
Eindeutigkeit verletzen



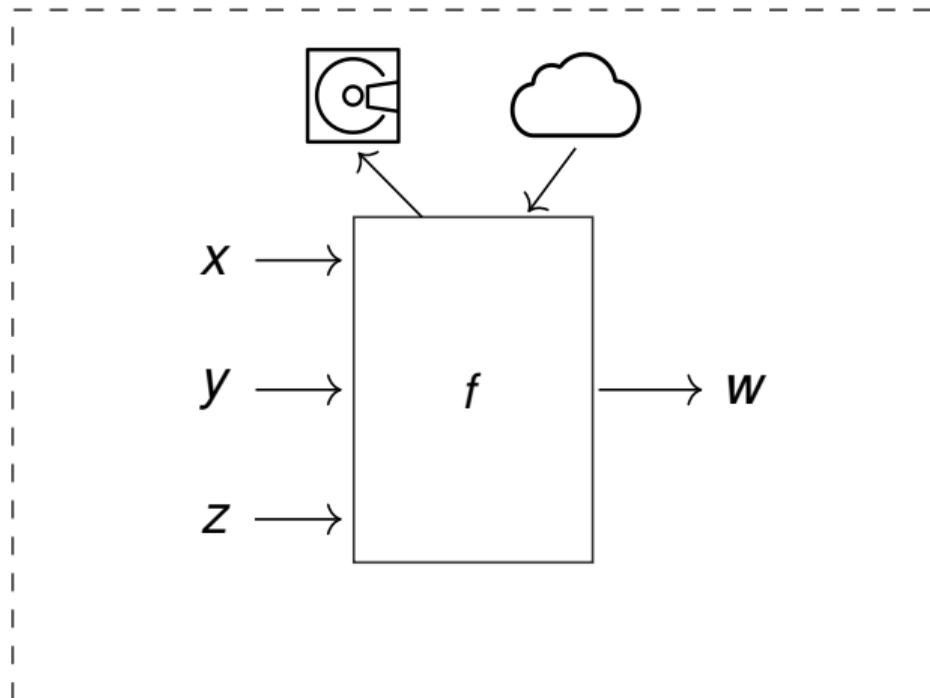
# Funktionen in der Programmierung

- ▶ Funktionen sind **eindeutig**
- ▶ Ausführungsumgebung  
Speicher, Netzwerk, ...
- ▶ Interaktion mit Umgebung kann  
Eindeutigkeit verletzen



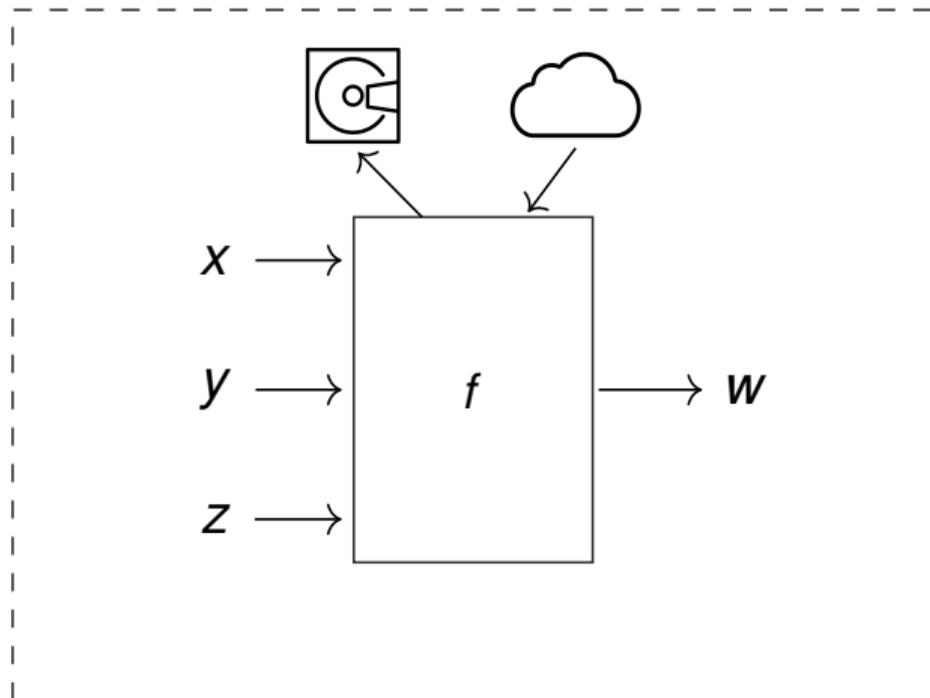
# Funktionen in der Programmierung

- ▶ Funktionen sind **eindeutig**
- ▶ Ausführungsumgebung  
Speicher, Netzwerk, ...
- ▶ Interaktion mit Umgebung kann  
Eindeutigkeit verletzen



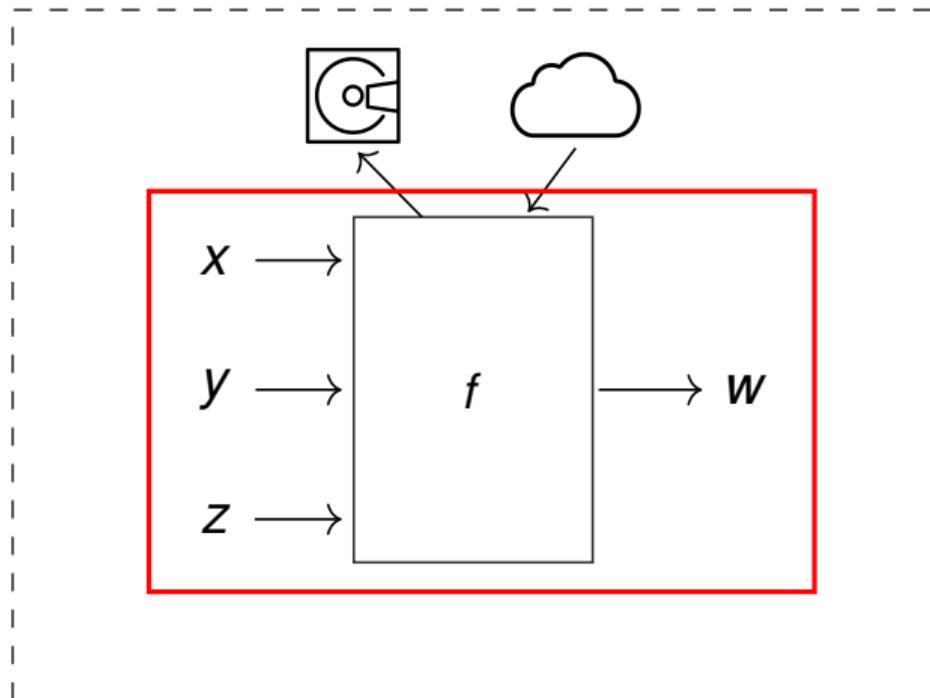
# Funktionen in der Programmierung

- ▶ Funktionen sind **eindeutig**
- ▶ Ausführungsumgebung  
Speicher, Netzwerk, ...
- ▶ Interaktion mit Umgebung kann  
Eindeutigkeit verletzen
- ▶ **Seiteneffekt**  
Veränderung der Umgebung



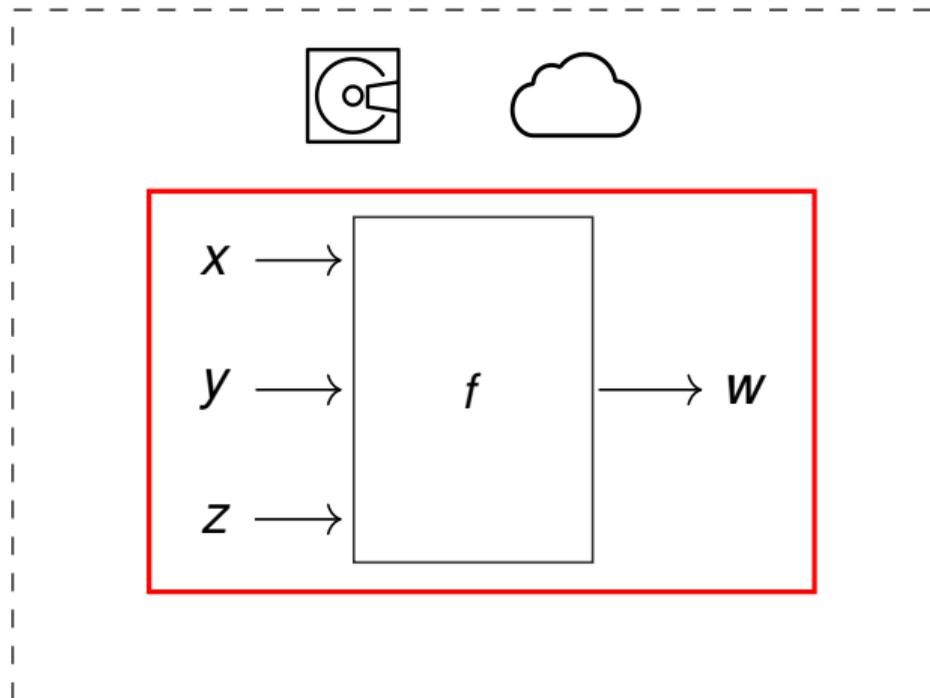
# Funktionen in der Programmierung

- ▶ Funktionen sind **eindeutig**
- ▶ Ausführungsumgebung  
Speicher, Netzwerk, ...
- ▶ Interaktion mit Umgebung kann  
Eindeutigkeit verletzen
- ▶ **Seiteneffekt**  
Veränderung der Umgebung



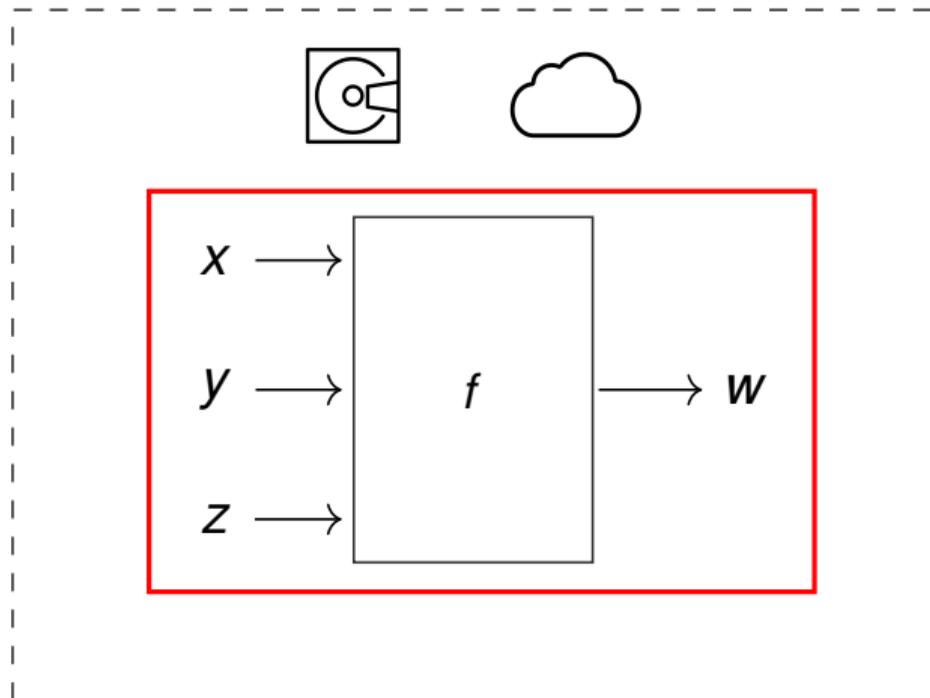
# Funktionen in der Programmierung

- ▶ Funktionen sind **eindeutig**
- ▶ Ausführungsumgebung  
Speicher, Netzwerk, ...
- ▶ Interaktion mit Umgebung kann  
Eindeutigkeit verletzen
- ▶ **Seiteneffekt**  
Veränderung der Umgebung



# Funktionen in der Programmierung

- ▶ Funktionen sind **eindeutig**
- ▶ Ausführungsumgebung  
Speicher, Netzwerk, ...
- ▶ Interaktion mit Umgebung kann  
Eindeutigkeit verletzen
- ▶ **Seiteneffekt**  
Veränderung der Umgebung
- ▶ **pure Funktion**  
keine Interaktion mit Umgebung



# Funktionen beschreiben

▶  $f(n) =$  größte Primzahl  $\leq n$

# Funktionen beschreiben

►  $f(n) =$  größte Primzahl  $\leq n$

$n$	$f(n)$
2	2
3	3
4	3
5	5
6	5

# Funktionen beschreiben

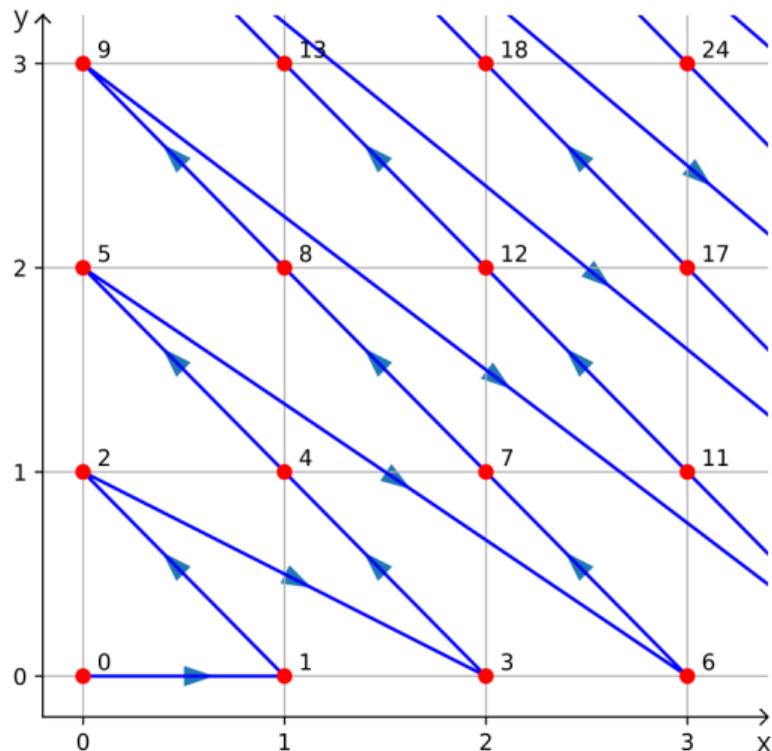
▶  $f(n) =$  größte Primzahl  $\leq n$

# Funktionen beschreiben

- ▶  $f(n) =$  größte Primzahl  $\leq n$
- ▶  $c(x, y)$

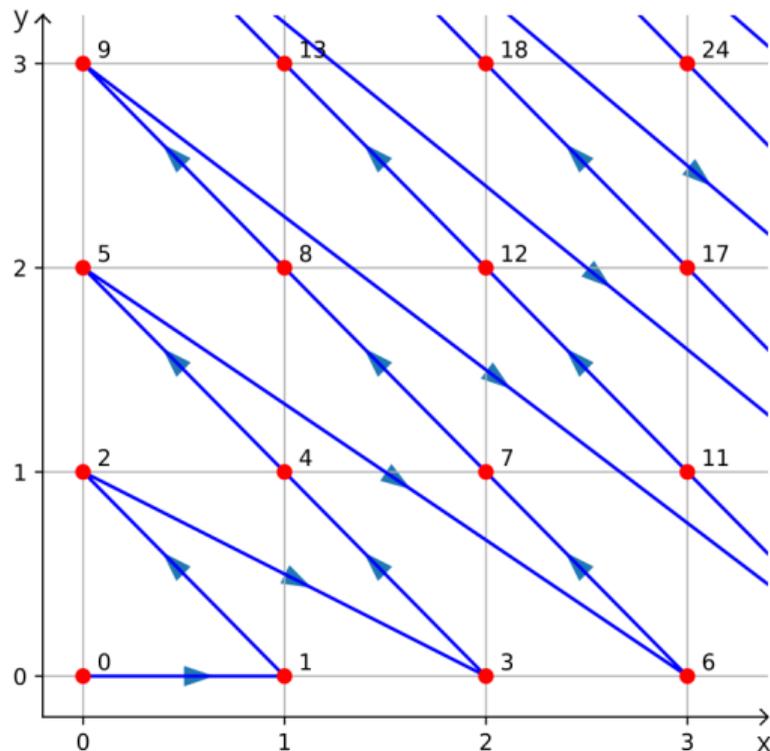
# Funktionen beschreiben

- ▶  $f(n) = \text{größte Primzahl} \leq n$
- ▶  $c(x, y)$



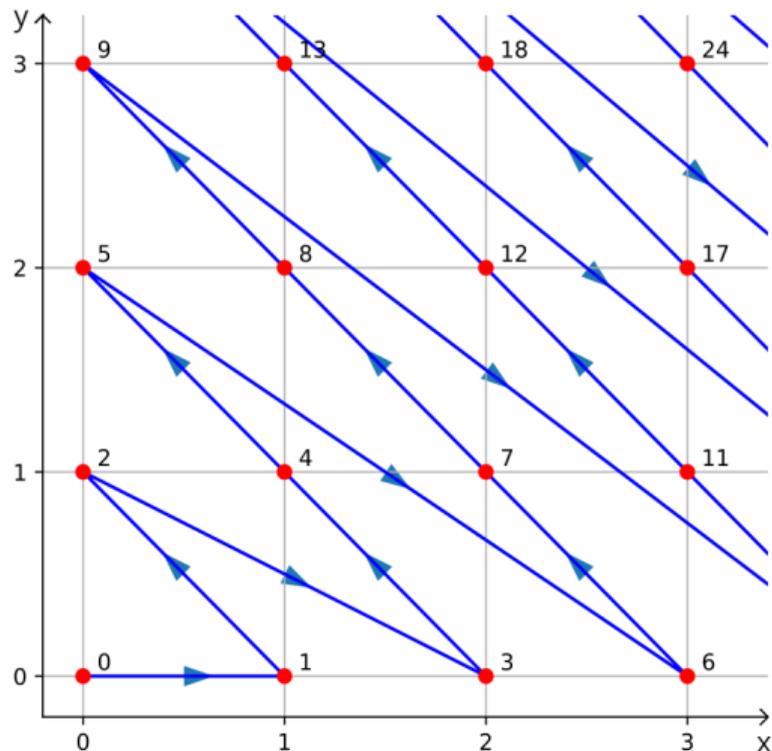
# Funktionen beschreiben

- ▶  $f(n) = \text{größte Primzahl } \leq n$
- ▶  $c(x, y) = \text{Anz. der Schritte bis } (x, y)$



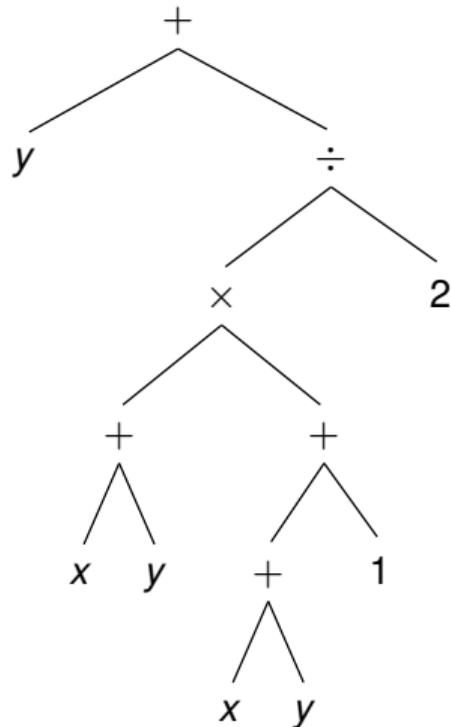
# Funktionen beschreiben

- ▶  $f(n) =$  größte Primzahl  $\leq n$
- ▶  $c(x, y) =$  Anz. der Schritte bis  $(x, y)$
- ▶  $c(x, y) = y + \frac{(x+y)(x+y+1)}{2}$



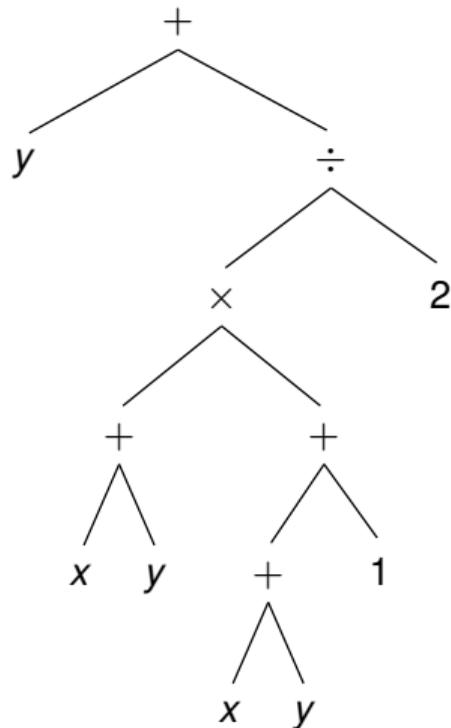
# Funktionen beschreiben

- ▶  $f(n) =$  größte Primzahl  $\leq n$
- ▶  $c(x, y) =$  Anz. der Schritte bis  $(x, y)$
- ▶  $c(x, y) = y + \frac{(x+y)(x+y+1)}{2}$



# Funktionen beschreiben

- ▶  $f(n) =$  größte Primzahl  $\leq n$
- ▶  $c(x, y) =$  Anz. der Schritte bis  $(x, y)$
- ▶  $c(x, y) = y + \frac{(x+y)(x+y+1)}{2}$
- ▶ Ausdrücke beschreiben Funktionen



# Ausdrücke

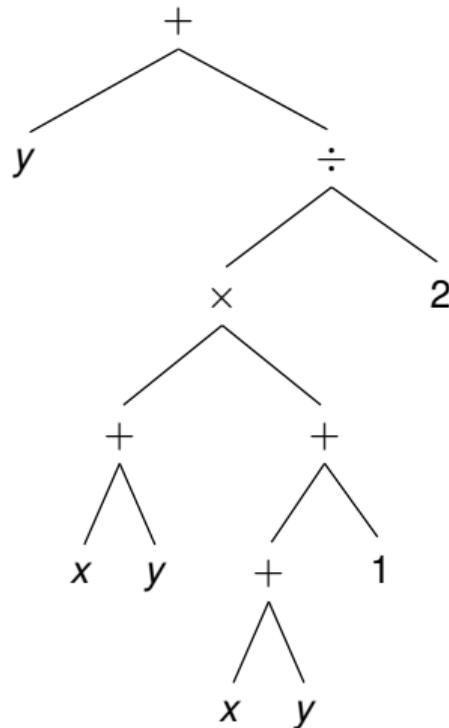
- ▶ Ausdruck

Baum beschriftet mit Variablen & Funktionen

# Ausdrücke

## ► Ausdruck

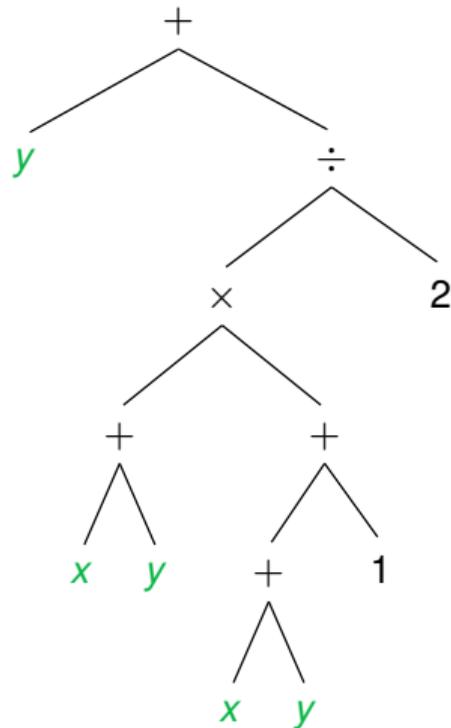
Baum beschriftet mit Variablen & Funktionen



# Ausdrücke

## ► Ausdruck

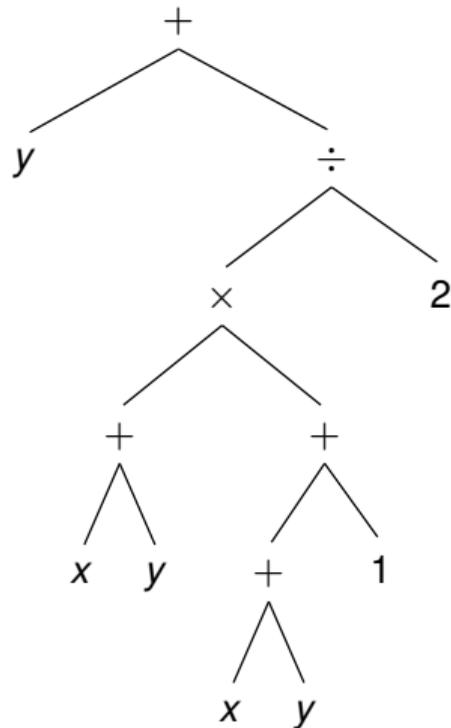
Baum beschriftet mit Variablen & Funktionen



# Ausdrücke

## ► Ausdruck

Baum beschriftet mit Variablen & Funktionen



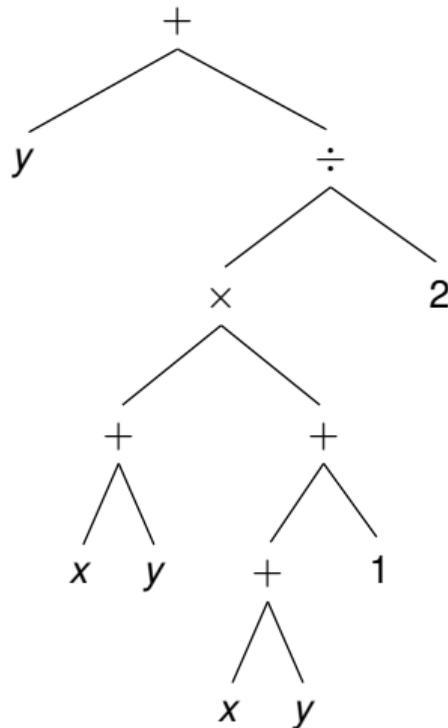
# Ausdrücke

- ▶ **Ausdruck**

Baum beschriftet mit Variablen & Funktionen

- ▶ **Konstante**

Funktion ohne Eingabeparameter



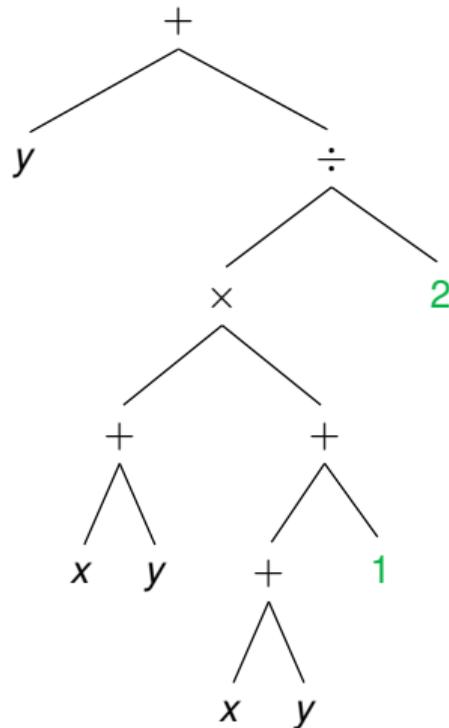
# Ausdrücke

- ▶ **Ausdruck**

Baum beschriftet mit Variablen & Funktionen

- ▶ **Konstante**

Funktion ohne Eingabeparameter



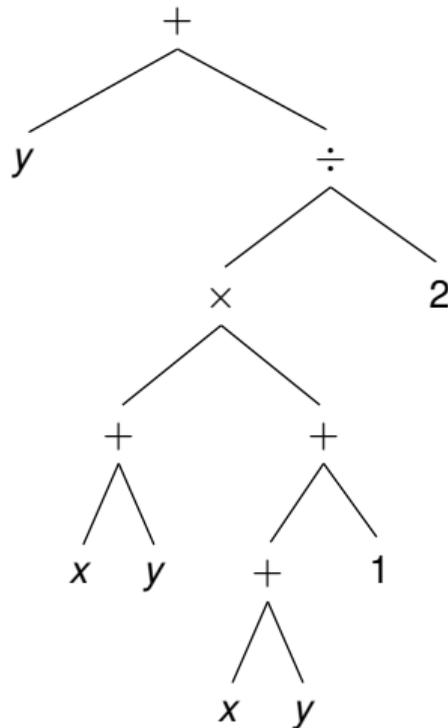
# Ausdrücke

- ▶ **Ausdruck**

Baum beschriftet mit Variablen & Funktionen

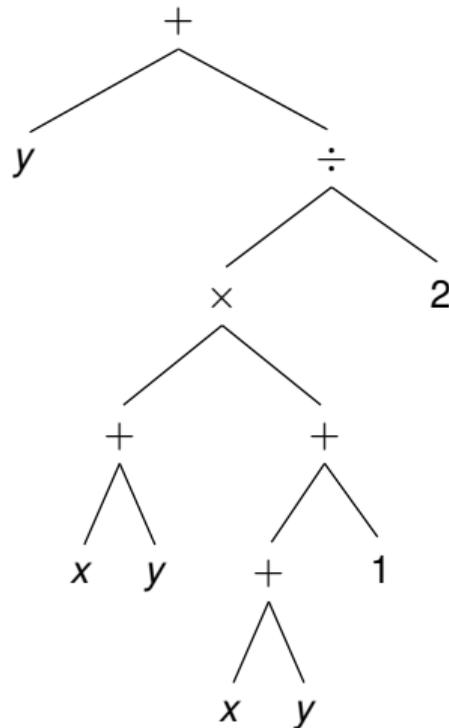
- ▶ **Konstante**

Funktion ohne Eingabeparameter



# Ausdrücke

- ▶ **Ausdruck**  
Baum beschriftet mit Variablen & Funktionen
- ▶ **Konstante**  
Funktion ohne Eingabeparameter
- ▶ **Basisfunktionen**  
vorgegebene Funktionen



# Ausdrücke

- ▶ **Ausdruck**  
Baum beschriftet mit Variablen & Funktionen
- ▶ **Konstante**  
Funktion ohne Eingabeparameter
- ▶ **Basisfunktionen**  
vorgegebene Funktionen

$$f(x) = g(x) + h(x)$$

$$g(x) = x \times h(2)$$

$$h(x) = x + 1$$

# Ausdrücke

- ▶ **Ausdruck**  
Baum beschriftet mit Variablen & Funktionen
- ▶ **Konstante**  
Funktion ohne Eingabeparameter
- ▶ **Basisfunktionen**  
vorgegebene Funktionen

$$f(x) = g(x) + h(x)$$

$$g(x) = x \times h(2)$$

$$h(x) = x + 1$$

Aufrufgraph

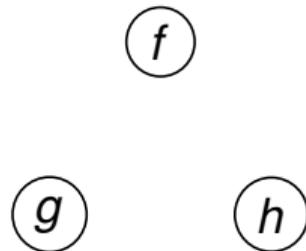
# Ausdrücke

- ▶ **Ausdruck**  
Baum beschriftet mit Variablen & Funktionen
- ▶ **Konstante**  
Funktion ohne Eingabeparameter
- ▶ **Basisfunktionen**  
vorgegebene Funktionen

$$f(x) = g(x) + h(x)$$

$$g(x) = x \times h(2)$$

$$h(x) = x + 1$$



Aufrufgraph

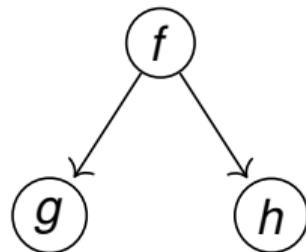
# Ausdrücke

- ▶ **Ausdruck**  
Baum beschriftet mit Variablen & Funktionen
- ▶ **Konstante**  
Funktion ohne Eingabeparameter
- ▶ **Basisfunktionen**  
vorgegebene Funktionen

$$f(x) = g(x) + h(x)$$

$$g(x) = x \times h(2)$$

$$h(x) = x + 1$$



Aufrufgraph

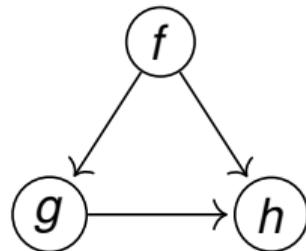
# Ausdrücke

- ▶ **Ausdruck**  
Baum beschriftet mit Variablen & Funktionen
- ▶ **Konstante**  
Funktion ohne Eingabeparameter
- ▶ **Basisfunktionen**  
vorgegebene Funktionen

$$f(x) = g(x) + h(x)$$

$$g(x) = x \times h(2)$$

$$h(x) = x + 1$$



**Aufrufgraph**

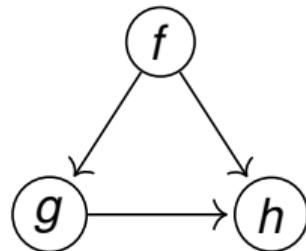
# Ausdrücke

- ▶ **Ausdruck**  
Baum beschriftet mit Variablen & Funktionen
- ▶ **Konstante**  
Funktion ohne Eingabeparameter
- ▶ **Basisfunktionen**  
vorgegebene Funktionen
- ▶ Ausdruck **rekursiv**, falls dessen Funktion im Aufrufgraph zu einem Zyklus führt

$$f(x) = g(x) + h(x)$$

$$g(x) = x \times h(2)$$

$$h(x) = x + 1$$



**Aufrufgraph**

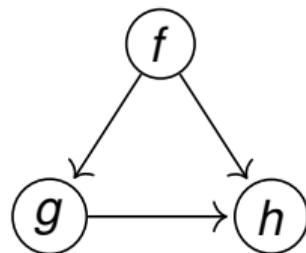
# Ausdrücke

- ▶ **Ausdruck**  
Baum beschriftet mit Variablen & Funktionen
- ▶ **Konstante**  
Funktion ohne Eingabeparameter
- ▶ **Basisfunktionen**  
vorgegebene Funktionen
- ▶ Ausdruck **rekursiv**, falls dessen Funktion im Aufrufgraph zu einem Zyklus führt

$$f(x) = g(x) + h(x)$$

$$g(x) = x \times h(2)$$

$$h(x) = g(x)$$



**Aufrufgraph**

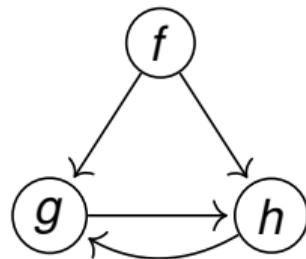
# Ausdrücke

- ▶ **Ausdruck**  
Baum beschriftet mit Variablen & Funktionen
- ▶ **Konstante**  
Funktion ohne Eingabeparameter
- ▶ **Basisfunktionen**  
vorgegebene Funktionen
- ▶ Ausdruck **rekursiv**, falls dessen Funktion im Aufrufgraph zu einem Zyklus führt

$$f(x) = g(x) + h(x)$$

$$g(x) = x \times h(2)$$

$$h(x) = g(x)$$



**Aufrufgraph**

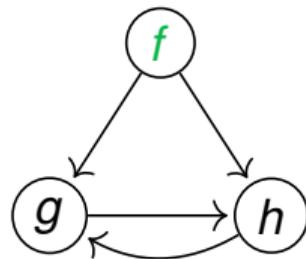
# Ausdrücke

- ▶ **Ausdruck**  
Baum beschriftet mit Variablen & Funktionen
- ▶ **Konstante**  
Funktion ohne Eingabeparameter
- ▶ **Basisfunktionen**  
vorgegebene Funktionen
- ▶ Ausdruck **rekursiv**, falls dessen Funktion im Aufrufgraph zu einem Zyklus führt

$$f(x) = g(x) + h(x)$$

$$g(x) = x \times h(2)$$

$$h(x) = g(x)$$



Aufrufgraph

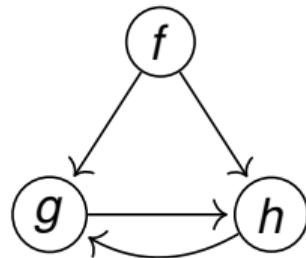
# Ausdrücke

- ▶ **Ausdruck**  
Baum beschriftet mit Variablen & Funktionen
- ▶ **Konstante**  
Funktion ohne Eingabeparameter
- ▶ **Basisfunktionen**  
vorgegebene Funktionen
- ▶ Ausdruck **rekursiv**, falls dessen Funktion im Aufrufgraph zu einem Zyklus führt

$$f(x) = g(x) + h(x)$$

$$g(x) = x \times h(2)$$

$$h(x) = g(x)$$



**Aufrufgraph**

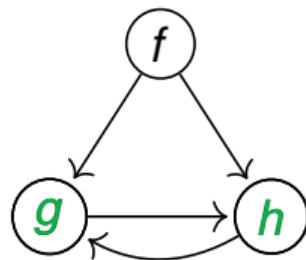
# Ausdrücke

- ▶ **Ausdruck**  
Baum beschriftet mit Variablen & Funktionen
- ▶ **Konstante**  
Funktion ohne Eingabeparameter
- ▶ **Basisfunktionen**  
vorgegebene Funktionen
- ▶ Ausdruck **rekursiv**, falls dessen Funktion im Aufrufgraph zu einem Zyklus führt

$$f(x) = g(x) + h(x)$$

$$g(x) = x \times h(2)$$

$$h(x) = g(x)$$



Aufrufgraph

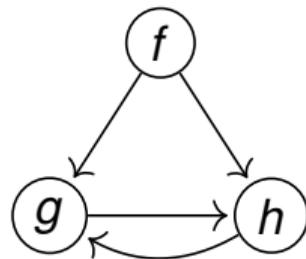
# Ausdrücke

- ▶ **Ausdruck**  
Baum beschriftet mit Variablen & Funktionen
- ▶ **Konstante**  
Funktion ohne Eingabeparameter
- ▶ **Basisfunktionen**  
vorgegebene Funktionen
- ▶ Ausdruck **rekursiv**, falls dessen Funktion im Aufrufgraph zu einem Zyklus führt

$$f(x) = g(x) + h(x)$$

$$g(x) = x \times h(2)$$

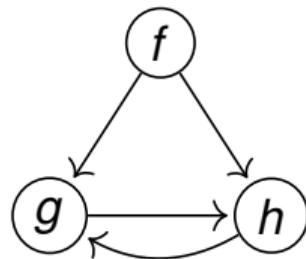
$$h(x) = g(x)$$



Aufrufgraph

# Ausdrücke

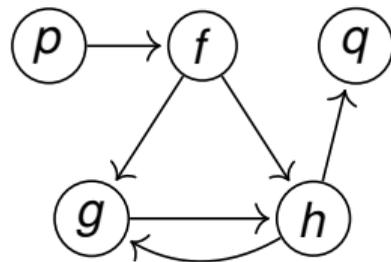
- ▶ **Ausdruck**  
Baum beschriftet mit Variablen & Funktionen
- ▶ **Konstante**  
Funktion ohne Eingabeparameter
- ▶ **Basisfunktionen**  
vorgegebene Funktionen
- ▶ Ausdruck **rekursiv**, falls dessen Funktion im Aufrufgraph zu einem Zyklus führt



**Aufrufgraph**

# Ausdrücke

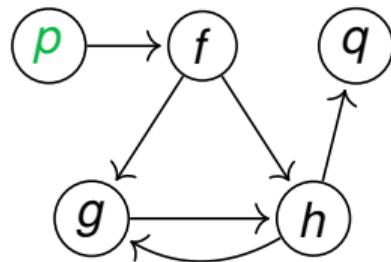
- ▶ **Ausdruck**  
Baum beschriftet mit Variablen & Funktionen
- ▶ **Konstante**  
Funktion ohne Eingabeparameter
- ▶ **Basisfunktionen**  
vorgegebene Funktionen
- ▶ Ausdruck **rekursiv**, falls dessen Funktion im Aufrufgraph zu einem Zyklus führt



Aufrufgraph

# Ausdrücke

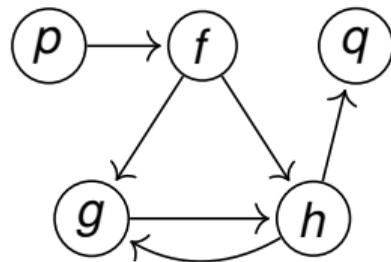
- ▶ **Ausdruck**  
Baum beschriftet mit Variablen & Funktionen
- ▶ **Konstante**  
Funktion ohne Eingabeparameter
- ▶ **Basisfunktionen**  
vorgegebene Funktionen
- ▶ Ausdruck **rekursiv**, falls dessen Funktion im Aufrufgraph zu einem Zyklus führt



Aufrufgraph

# Ausdrücke

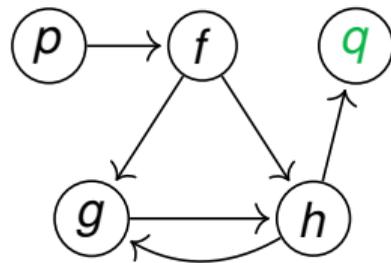
- ▶ **Ausdruck**  
Baum beschriftet mit Variablen & Funktionen
- ▶ **Konstante**  
Funktion ohne Eingabeparameter
- ▶ **Basisfunktionen**  
vorgegebene Funktionen
- ▶ Ausdruck **rekursiv**, falls dessen Funktion im Aufrufgraph zu einem Zyklus führt



**Aufrufgraph**

# Ausdrücke

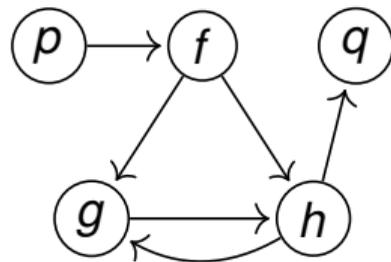
- ▶ **Ausdruck**  
Baum beschriftet mit Variablen & Funktionen
- ▶ **Konstante**  
Funktion ohne Eingabeparameter
- ▶ **Basisfunktionen**  
vorgegebene Funktionen
- ▶ Ausdruck **rekursiv**, falls dessen Funktion im Aufrufgraph zu einem Zyklus führt



Aufrufgraph

# Ausdrücke

- ▶ **Ausdruck**  
Baum beschriftet mit Variablen & Funktionen
- ▶ **Konstante**  
Funktion ohne Eingabeparameter
- ▶ **Basisfunktionen**  
vorgegebene Funktionen
- ▶ Ausdruck **rekursiv**, falls dessen Funktion im Aufrufgraph zu einem Zyklus führt



Aufrufgraph

# Rekursiver Ausdruck

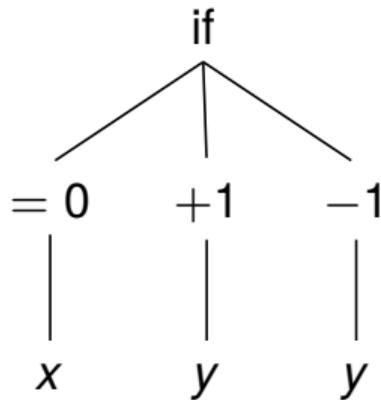
- ▶ gesucht: Ausdruck für Addition

# Rekursiver Ausdruck

- ▶ gesucht: Ausdruck für Addition
- ▶ Basisfunktionen:  $+1$ ,  $-1$ ,  $= 0$ , if

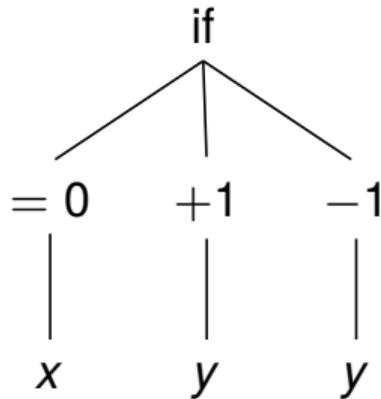
# Rekursiver Ausdruck

- ▶ gesucht: Ausdruck für Addition
- ▶ Basisfunktionen:  $+1$ ,  $-1$ ,  $= 0$ ,  $\text{if}$



# Rekursiver Ausdruck

- ▶ gesucht: Ausdruck für Addition
- ▶ Basisfunktionen:  $+1$ ,  $-1$ ,  $= 0$ ,  $\text{if}$



$$(x, y) \mapsto \begin{cases} y + 1, & \text{falls } x = 0 \\ y - 1, & \text{falls } x \neq 0 \end{cases}$$

# Rekursiver Ausdruck

- ▶ gesucht: Ausdruck für Addition
- ▶ Basisfunktionen:  $+1$ ,  $-1$ ,  $= 0$ , if

# Rekursiver Ausdruck

- ▶ gesucht: Ausdruck für Addition
- ▶ Basisfunktionen:  $+1$ ,  $-1$ ,  $= 0$ , if
- ▶ Ausdruck für  $\text{add}(x, y)$

# Rekursiver Ausdruck

- ▶ gesucht: Ausdruck für Addition
- ▶ Basisfunktionen:  $+1$ ,  $-1$ ,  $= 0$ ,  $\text{if}$
- ▶ Ausdruck für  $\text{add}(x, y)$

$$\text{add}(x, y) = x + \underbrace{1 + \dots + 1}_{y\text{-mal}}$$

# Rekursiver Ausdruck

- ▶ gesucht: Ausdruck für Addition
- ▶ Basisfunktionen:  $+1$ ,  $-1$ ,  $= 0$ , if
- ▶ Ausdruck für  $\text{add}(x, y)$
- ▶ rekursiver Ausdruck notwendig

$$\text{add}(x, y) = x + \underbrace{1 + \dots + 1}_{y\text{-mal}}$$

# Rekursiver Ausdruck

- ▶ gesucht: Ausdruck für Addition
- ▶ Basisfunktionen:  $+1$ ,  $-1$ ,  $= 0$ , if
- ▶ Ausdruck für  $\text{add}(x, y)$
- ▶ rekursiver Ausdruck notwendig

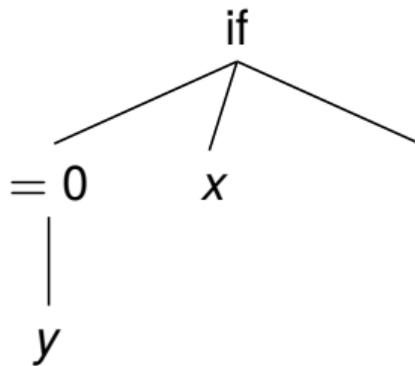
# Rekursiver Ausdruck

- ▶ gesucht: Ausdruck für Addition
- ▶ Basisfunktionen:  $+1$ ,  $-1$ ,  $= 0$ , if
- ▶ Ausdruck für  $\text{add}(x, y)$
- ▶ rekursiver Ausdruck notwendig

$$x + 0 = x$$

# Rekursiver Ausdruck

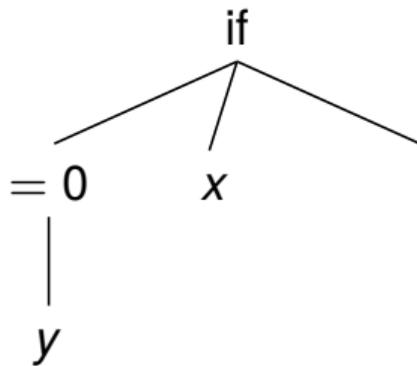
- ▶ gesucht: Ausdruck für Addition
- ▶ Basisfunktionen:  $+1$ ,  $-1$ ,  $= 0$ ,  $\text{if}$
- ▶ Ausdruck für  $\text{add}(x, y)$
- ▶ rekursiver Ausdruck notwendig



$$x + 0 = x$$

# Rekursiver Ausdruck

- ▶ gesucht: Ausdruck für Addition
- ▶ Basisfunktionen:  $+1$ ,  $-1$ ,  $= 0$ , `if`
- ▶ Ausdruck für `add(x, y)`
- ▶ rekursiver Ausdruck notwendig

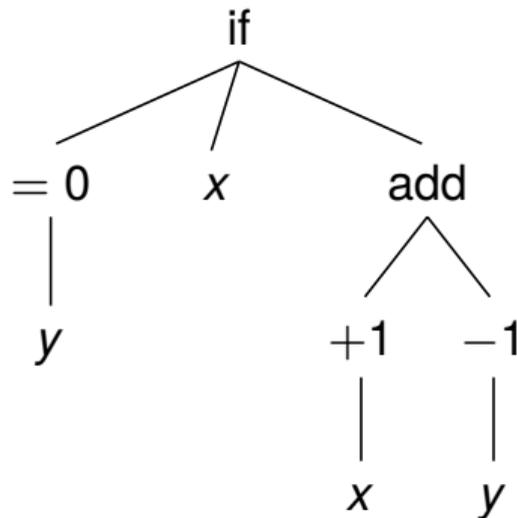


$$x + 0 = x$$

$$x + y = (x + 1) + (y - 1)$$

# Rekursiver Ausdruck

- ▶ gesucht: Ausdruck für Addition
- ▶ Basisfunktionen:  $+1$ ,  $-1$ ,  $= 0$ ,  $\text{if}$
- ▶ Ausdruck für  $\text{add}(x, y)$
- ▶ rekursiver Ausdruck notwendig

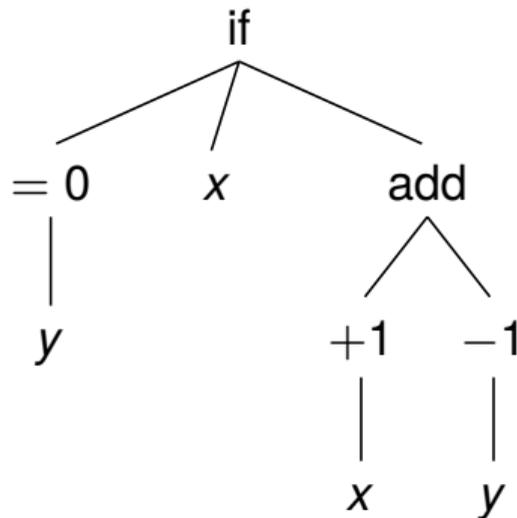


$$x + 0 = x$$

$$x + y = (x + 1) + (y - 1)$$

# Rekursiver Ausdruck

- ▶ gesucht: Ausdruck für Addition
- ▶ Basisfunktionen:  $+1$ ,  $-1$ ,  $= 0$ ,  $\text{if}$
- ▶ Ausdruck für  $\text{add}(x, y)$
- ▶ rekursiver Ausdruck notwendig
- ▶ ÜA: auf negative Zahlen erweitern

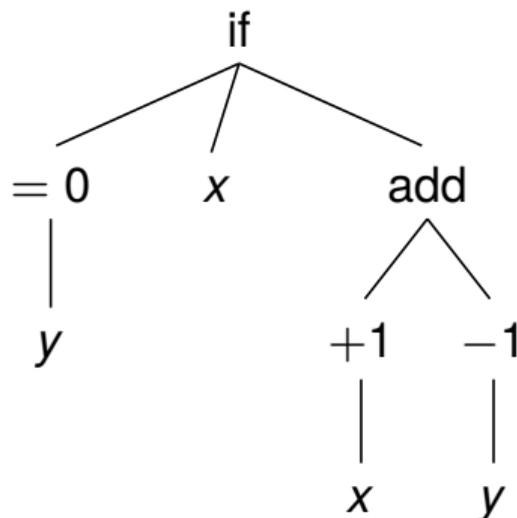


$$x + 0 = x$$

$$x + y = (x + 1) + (y - 1)$$

# Rekursiver Ausdruck

- ▶ gesucht: Ausdruck für Addition
- ▶ Basisfunktionen:  $+1$ ,  $-1$ ,  $= 0$ ,  $\text{if}$
- ▶ Ausdruck für  $\text{add}(x, y)$
- ▶ rekursiver Ausdruck notwendig
- ▶ ÜA: auf negative Zahlen erweitern  
zusätzliche Basisfunktion  $< 0$

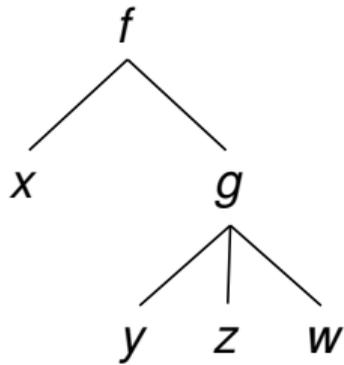


$$x + 0 = x$$

$$x + y = (x + 1) + (y - 1)$$

# Repräsentationen für Ausdrücke

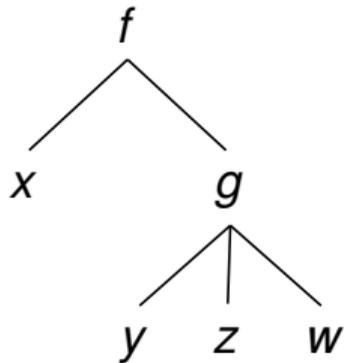
Baum



# Repräsentationen für Ausdrücke

Baum

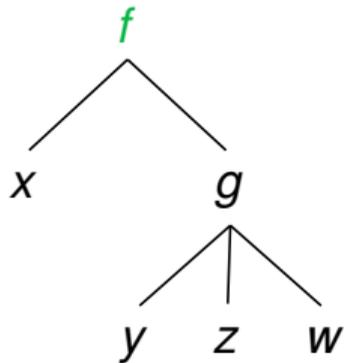
Präfix-Notation



# Repräsentationen für Ausdrücke

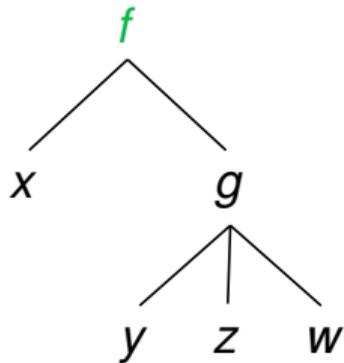
Baum

Präfix-Notation



# Repräsentationen für Ausdrücke

Baum

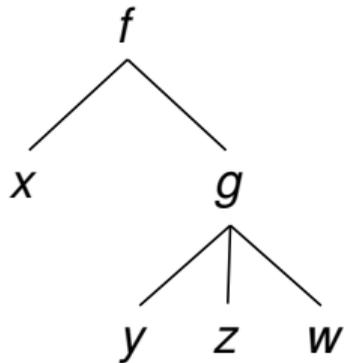


Präfix-Notation

( f ( y z w ) )

# Repräsentationen für Ausdrücke

Baum

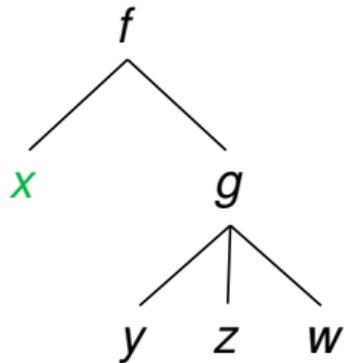


Präfix-Notation

( )

# Repräsentationen für Ausdrücke

Baum

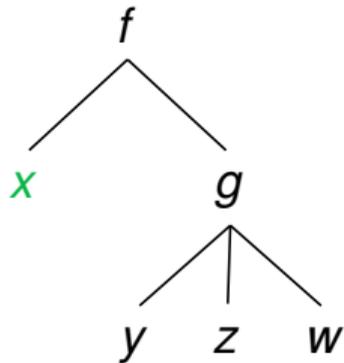


Präfix-Notation

( )

# Repräsentationen für Ausdrücke

Baum

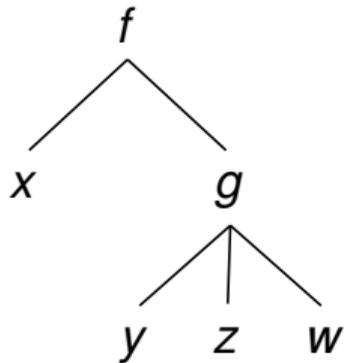


Präfix-Notation

( ( ) )

# Repräsentationen für Ausdrücke

Baum

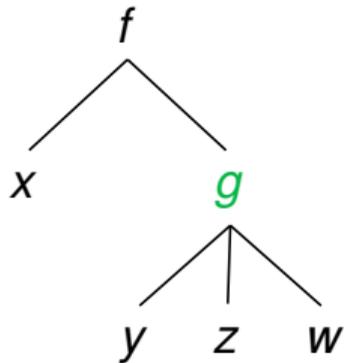


Präfix-Notation

( ( ) )

# Repräsentationen für Ausdrücke

Baum

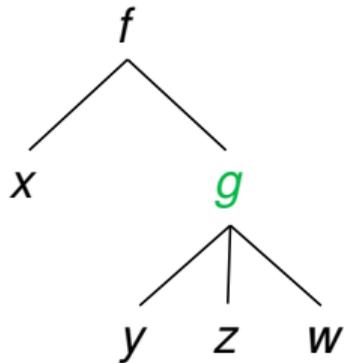


Präfix-Notation

( ( ) )

# Repräsentationen für Ausdrücke

Baum

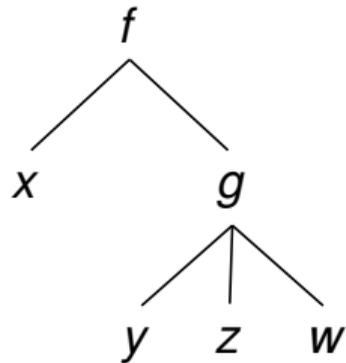


Präfix-Notation

( ( ) (            ) )

# Repräsentationen für Ausdrücke

Baum

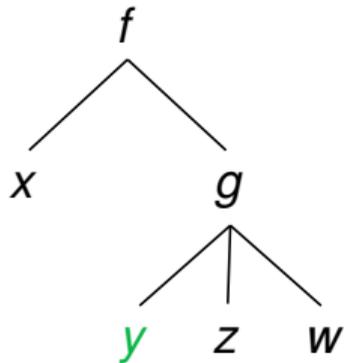


Präfix-Notation

( ( ) (                    ) )

# Repräsentationen für Ausdrücke

Baum

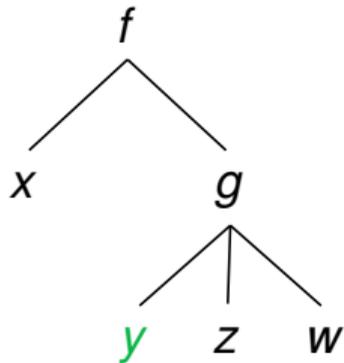


Präfix-Notation

( ( ) (                    ) )

# Repräsentationen für Ausdrücke

Baum

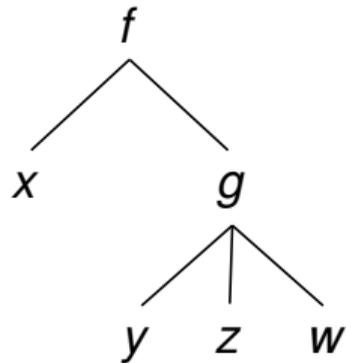


Präfix-Notation

( ( ) ( ( ) ) )

# Repräsentationen für Ausdrücke

Baum

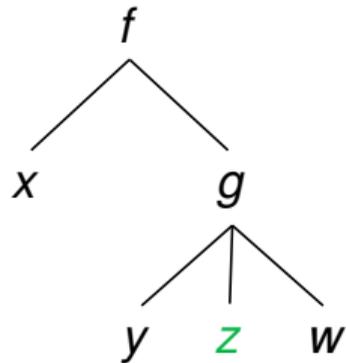


Präfix-Notation

( ( ) ( ( ) ) )

# Repräsentationen für Ausdrücke

Baum

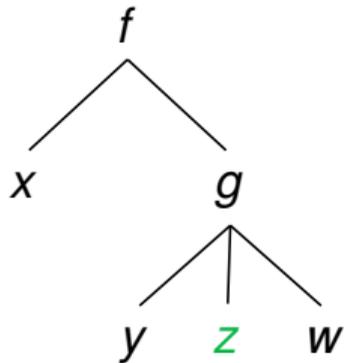


Präfix-Notation

( ( ) ( ( ) ) )

# Repräsentationen für Ausdrücke

Baum

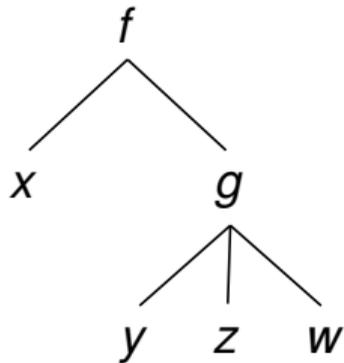


Präfix-Notation

( ( ) ( ( ) ( ) ) )

# Repräsentationen für Ausdrücke

Baum

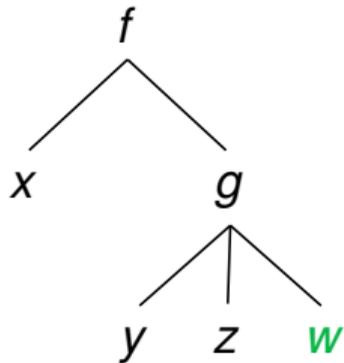


Präfix-Notation

$(( ( ) ( ( ) ( ) ) ) )$

# Repräsentationen für Ausdrücke

Baum

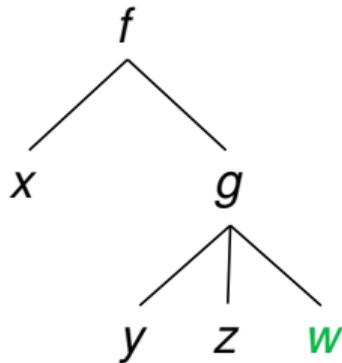


Präfix-Notation

( ( ) ( ( ) ( ) ) )

# Repräsentationen für Ausdrücke

Baum

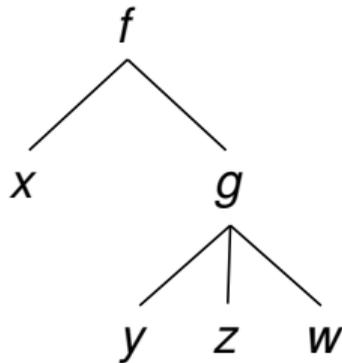


Präfix-Notation

( ( ) ( ( ) ( ) ( ( ) ) ) )

# Repräsentationen für Ausdrücke

Baum

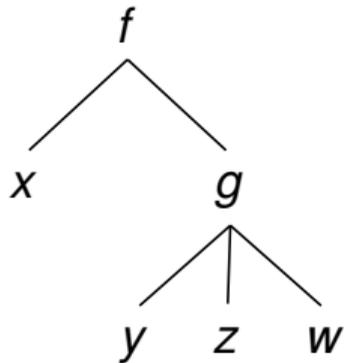


Präfix-Notation

( ( ) ( ( ) ( ) ( ) ) )

# Repräsentationen für Ausdrücke

Baum

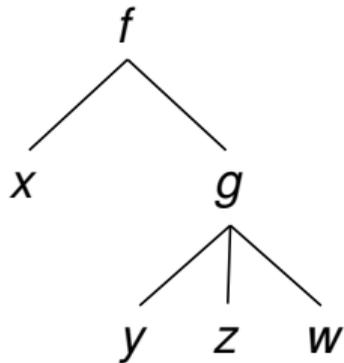


Präfix-Notation

$(f(x)(g(y)(z)(w)))$

# Repräsentationen für Ausdrücke

Baum



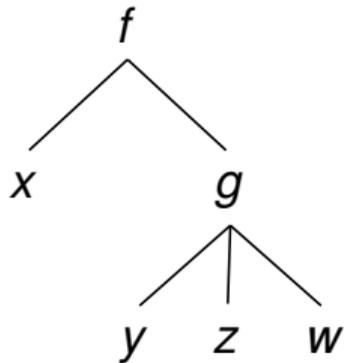
Präfix-Notation

$(f(x)(g(y)(z)(w)))$



# Repräsentationen für Ausdrücke

Baum



Präfix-Notation

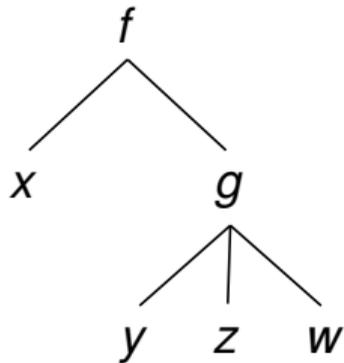
$(f(x)(g(y)(z)(w)))$



$(f(x)(g(y)(z)(w)))$

# Repräsentationen für Ausdrücke

Baum



Präfix-Notation

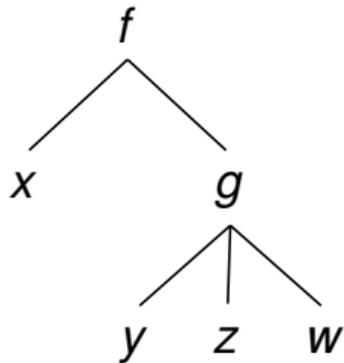
$(f(x)(g(y)(z)(w)))$



$(f x (g y z w ))$

# Repräsentationen für Ausdrücke

Baum



Präfix-Notation

$(f (x) (g (y) (z) (w)))$

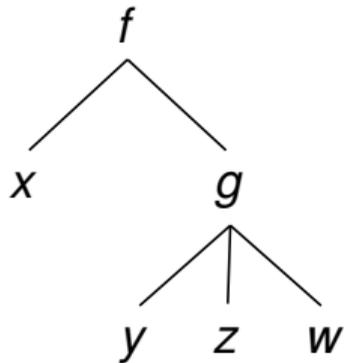


$(f x (g y z w ))$

Math. Notation

# Repräsentationen für Ausdrücke

Baum



Präfix-Notation

$(f(x)(g(y)(z)(w)))$



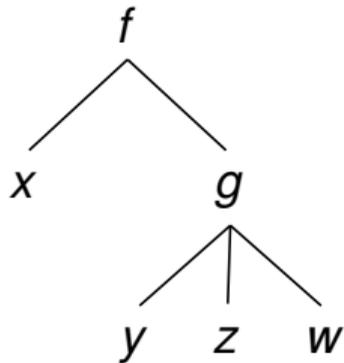
$(f x (g y z w))$

Math. Notation



# Repräsentationen für Ausdrücke

Baum



Präfix-Notation

$(f(x)(g(y)(z)(w)))$



$(f x (g y z w))$

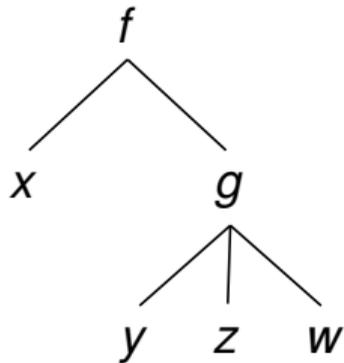
Math. Notation

$(f x (g y z w))$



# Repräsentationen für Ausdrücke

Baum



Präfix-Notation

$(f(x)(g(y)(z)(w)))$



$(f x (g y z w ))$

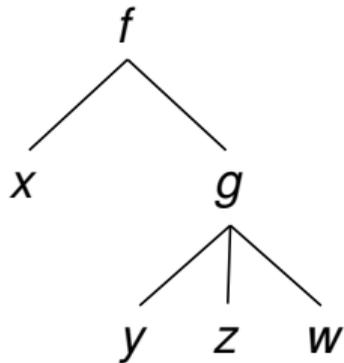
Math. Notation

$f(x g(y z w ))$



# Repräsentationen für Ausdrücke

Baum



Präfix-Notation

$(f(x)(g(y)(z)(w)))$



$(f x (g y z w))$

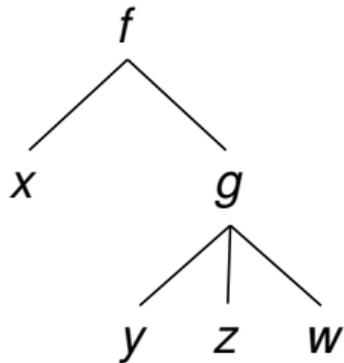
Math. Notation

$f(x, g(y, z, w))$



# Repräsentationen für Ausdrücke

Baum



Präfix-Notation

$(f(x)(g(y)(z)(w)))$



$(f x (g y z w))$

Math. Notation

$f(x, g(y, z, w))$



# Infix-Notation

- ▶ 2-stellige Funktion zwischen Parametern

# Infix-Notation

$f(x, y)$      $x f y$

- ▶ 2-stellige Funktion zwischen Parametern

# Infix-Notation

$$+(x, y) \quad x + y$$

- ▶ 2-stellige Funktion zwischen Parametern

# Infix-Notation

$$+(x, y) \quad x + y$$

- ▶ 2-stellige Funktion zwischen Parametern
- ▶ **Assoziativität**  
 $x f y f z$

# Infix-Notation

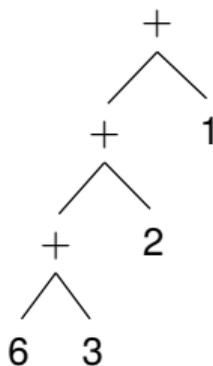
$$6 + 3 + 2 + 1$$

- ▶ 2-stellige Funktion zwischen Parametern
- ▶ **Assoziativität**  
 $x f y f z$

# Infix-Notation

$$6 + 3 + 2 + 1$$

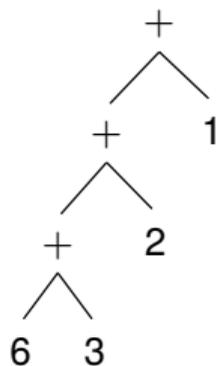
- ▶ 2-stellige Funktion zwischen Parametern
- ▶ **Assoziativität**  
 $x f y f z$



# Infix-Notation

$$6 + 3 + 2 + 1$$

- ▶ 2-stellige Funktion zwischen Parametern
- ▶ **Assoziativität**  
 $x f y f z$



linksassoziativ

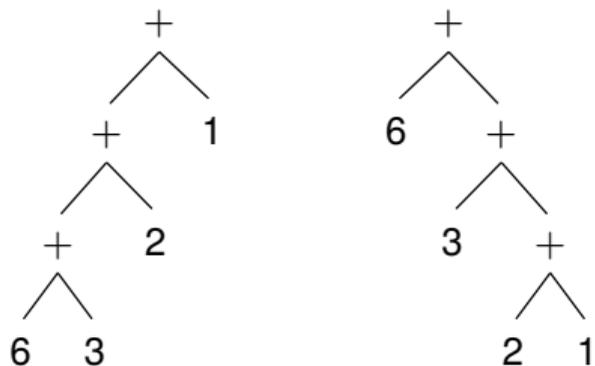
$$(x f y) f z$$

# Infix-Notation

$$6 + 3 + 2 + 1$$

- ▶ 2-stellige Funktion zwischen Parametern

- ▶ **Assoziativität**  
 $x f y f z$



**linksassoziativ**

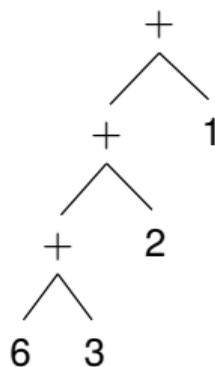
$(x f y) f z$

# Infix-Notation

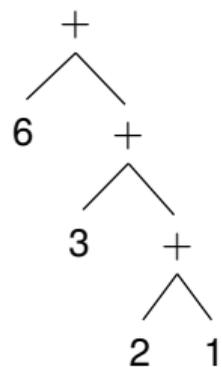
$$6 + 3 + 2 + 1$$

- ▶ 2-stellige Funktion zwischen Parametern

- ▶ **Assoziativität**  
 $x f y f z$



linksassoziativ  
 $(x f y) f z$



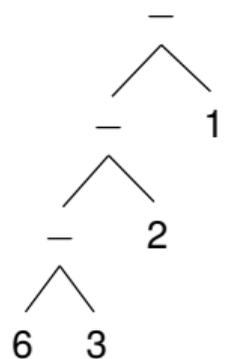
rechtsassoziativ  
 $x f (y f z)$

# Infix-Notation

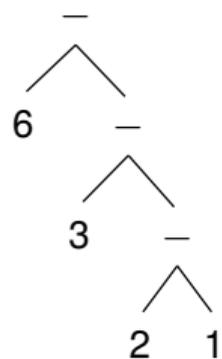
6 - 3 - 2 - 1

- ▶ 2-stellige Funktion zwischen Parametern

- ▶ **Assoziativität**  
 $x f y f z$



linksassoziativ  
 $(x f y) f z$

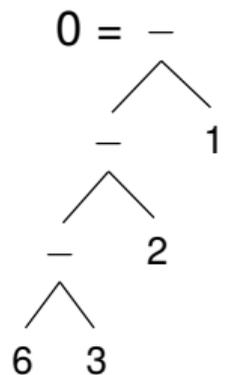


rechtsassoziativ  
 $x f (y f z)$

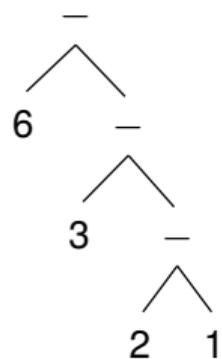
# Infix-Notation

- ▶ 2-stellige Funktion zwischen Parametern
- ▶ **Assoziativität**  
 $x f y f z$

6 - 3 - 2 - 1



linksassoziativ  
 $(x f y) f z$

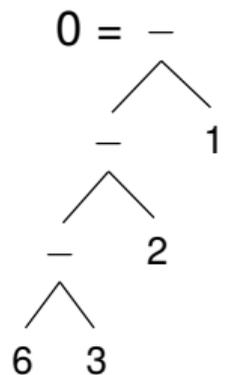


rechtsassoziativ  
 $x f (y f z)$

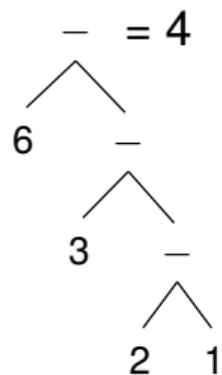
# Infix-Notation

- ▶ 2-stellige Funktion zwischen Parametern
- ▶ **Assoziativität**  
 $x f y f z$

$$6 - 3 - 2 - 1$$



linksassoziativ  
 $(x f y) f z$

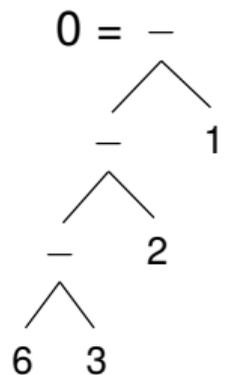


rechtsassoziativ  
 $x f (y f z)$

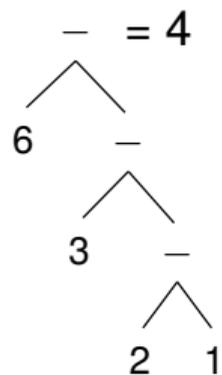
# Infix-Notation

- ▶ 2-stellige Funktion zwischen Parametern
- ▶ Assoziativität  
 $x f y f z$
- ▶ Präzedenz  
 $x f y g z$

$$6 - 3 - 2 - 1$$



linksassoziativ  
 $(x f y) f z$



rechtsassoziativ  
 $x f (y f z)$

# Infix-Notation

- ▶ 2-stellige Funktion zwischen Parametern
- ▶ Assoziativität  
 $x f y f z$
- ▶ Präzedenz  
 $x f y g z$

# Infix-Notation

- ▶ 2-stellige Funktion zwischen Parametern
- ▶ Assoziativität  
 $x f y f z$
- ▶ Präzedenz  
 $x f y g z$

$$2 + 3 \times 4$$

# Infix-Notation

- ▶ 2-stellige Funktion zwischen Parametern
- ▶ Assoziativität  
 $x f y f z$
- ▶ Präzedenz  
 $x f y g z$

$$2 + 3 \times 4$$

$$(2 + 3) \times 4$$

# Infix-Notation

- ▶ 2-stellige Funktion zwischen Parametern
- ▶ Assoziativität  
 $x f y f z$
- ▶ Präzedenz  
 $x f y g z$

$$2 + 3 \times 4$$

$$(2 + 3) \times 4 \quad 2 + (3 \times 4)$$

# Zuweisungen und Ausdrücke

## Sequenz von Zuweisungen

# Zuweisungen und Ausdrücke

## Sequenz von Zuweisungen

$$a \leftarrow (x + y) \times y$$

$$a \leftarrow (a \times a) + 1$$

$$b \leftarrow 5$$

$$c \leftarrow a - b$$

# Zuweisungen und Ausdrücke

## Sequenz von Zuweisungen

$$1: a \leftarrow (x + y) \times y$$

$$2: a \leftarrow (a \times a) + 1$$

$$3: b \leftarrow 5$$

$$4: c \leftarrow a - b$$

# Zuweisungen und Ausdrücke

Sequenz von Zuweisungen

induzierte Ausdrücke

$$1: a \leftarrow (x + y) \times y$$

$$2: a \leftarrow (a \times a) + 1$$

$$3: b \leftarrow 5$$

$$4: c \leftarrow a - b$$

# Zuweisungen und Ausdrücke

Sequenz von Zuweisungen

induzierte Ausdrücke

$$1: a \leftarrow (x + y) \times y$$

$$2: a \leftarrow (a \times a) + 1$$

$$3: b \leftarrow 5$$

$$4: c \leftarrow a - b$$

# Zuweisungen und Ausdrücke

Sequenz von Zuweisungen

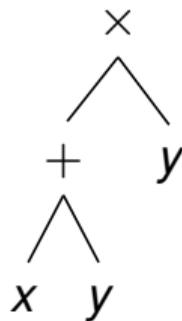
1:  $a \leftarrow (x + y) \times y$

2:  $a \leftarrow (a \times a) + 1$

3:  $b \leftarrow 5$

4:  $c \leftarrow a - b$

induzierte Ausdrücke



# Zuweisungen und Ausdrücke

## Sequenz von Zuweisungen

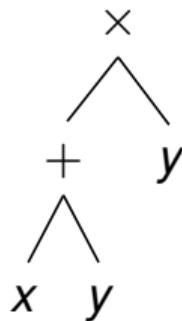
1:  $a \leftarrow (x + y) \times y$

2:  $a \leftarrow (a \times a) + 1$

3:  $b \leftarrow 5$

4:  $c \leftarrow a - b$

## induzierte Ausdrücke



# Zuweisungen und Ausdrücke

Sequenz von Zuweisungen

induzierte Ausdrücke

$$1: a \leftarrow (x + y) \times y$$

$$2: a \leftarrow (a \times a) + 1$$

$$3: b \leftarrow 5$$

$$4: c \leftarrow a - b$$

# Zuweisungen und Ausdrücke

Sequenz von Zuweisungen

induzierte Ausdrücke

$$1: a \leftarrow (x + y) \times y$$

$$2: a \leftarrow (a \times a) + 1$$

$$3: b \leftarrow 5$$

$$4: c \leftarrow a - b$$

# Zuweisungen und Ausdrücke

Sequenz von Zuweisungen

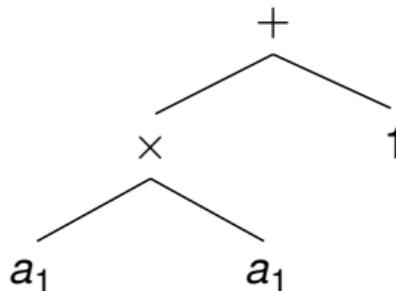
1:  $a \leftarrow (x + y) \times y$

2:  $a \leftarrow (a \times a) + 1$

3:  $b \leftarrow 5$

4:  $c \leftarrow a - b$

induzierte Ausdrücke



# Zuweisungen und Ausdrücke

Sequenz von Zuweisungen

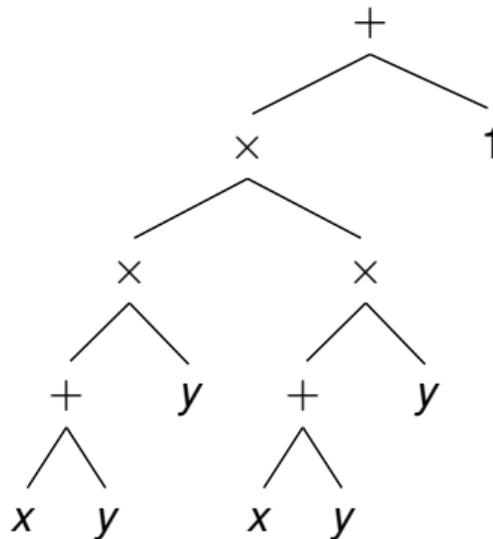
1:  $a \leftarrow (x + y) \times y$

2:  $a \leftarrow (a \times a) + 1$

3:  $b \leftarrow 5$

4:  $c \leftarrow a - b$

induzierte Ausdrücke



# Zuweisungen und Ausdrücke

Sequenz von Zuweisungen

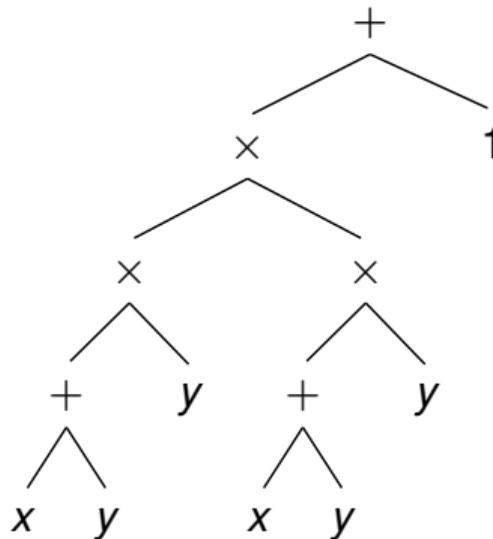
1:  $a \leftarrow (x + y) \times y$

2:  $a \leftarrow (a \times a) + 1$

3:  $b \leftarrow 5$

4:  $c \leftarrow a - b$

induzierte Ausdrücke



# Zuweisungen und Ausdrücke

Sequenz von Zuweisungen

induzierte Ausdrücke

$$1: a \leftarrow (x + y) \times y$$

$$2: a \leftarrow (a \times a) + 1$$

$$3: b \leftarrow 5$$

$$4: c \leftarrow a - b$$

# Zuweisungen und Ausdrücke

Sequenz von Zuweisungen

induzierte Ausdrücke

$$1: a \leftarrow (x + y) \times y$$

$$2: a \leftarrow (a \times a) + 1$$

$$3: b \leftarrow 5$$

$$4: c \leftarrow a - b$$

# Zuweisungen und Ausdrücke

Sequenz von Zuweisungen

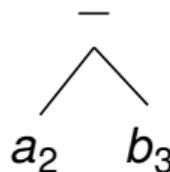
$$1: a \leftarrow (x + y) \times y$$

$$2: a \leftarrow (a \times a) + 1$$

$$3: b \leftarrow 5$$

$$4: c \leftarrow a - b$$

induzierte Ausdrücke



# Zuweisungen und Ausdrücke

Sequenz von Zuweisungen

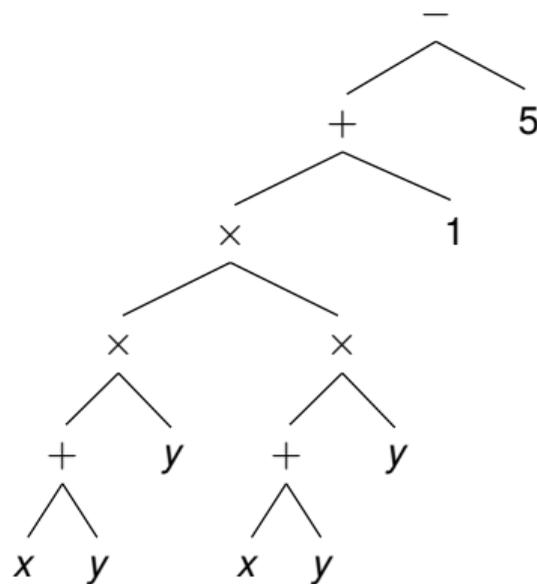
1:  $a \leftarrow (x + y) \times y$

2:  $a \leftarrow (a \times a) + 1$

3:  $b \leftarrow 5$

4:  $c \leftarrow a - b$

induzierte Ausdrücke



# Zuweisungen und Ausdrücke

Sequenz von Zuweisungen

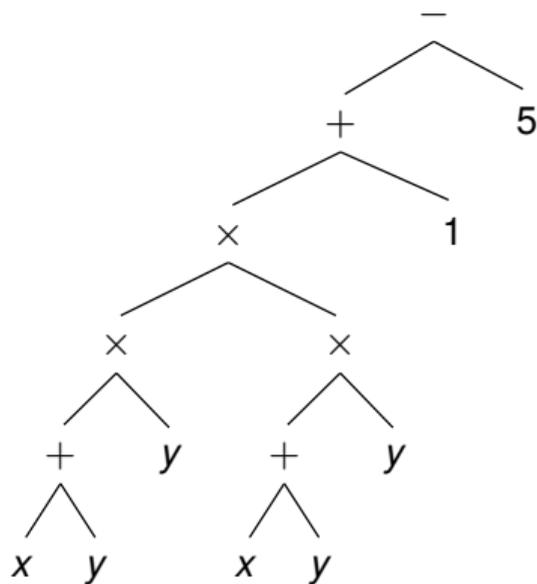
1:  $a \leftarrow (x + y) \times y$

2:  $a \leftarrow (a \times a) + 1$

3:  $b \leftarrow 5$

4:  $c \leftarrow a - b$

induzierte Ausdrücke



# Zuweisungen und Ausdrücke

Sequenz von Zuweisungen

$$1: a \leftarrow (x + y) \times y$$

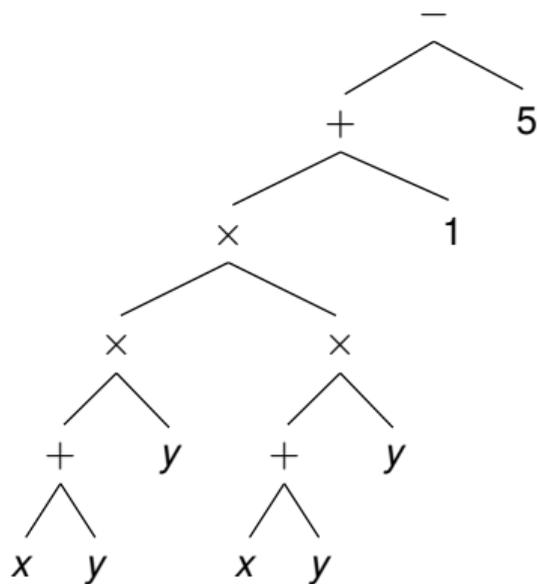
$$2: a \leftarrow (a \times a) + 1$$

$$3: b \leftarrow 5$$

$$4: c \leftarrow a - b$$

bottom-up Konstruktion eines Ausdrucks

induzierte Ausdrücke



# Quellen

- ▶ Computer Disk Icon (S. 2), Jonas Roßner, <https://openmoji.org>
- ▶ Cloud Icon (S. 2), Vanessa Boutzikoudi, <https://openmoji.org>
- ▶ Cantor's Pairing Function (S. 3), crh23, <https://commons.wikimedia.org>