



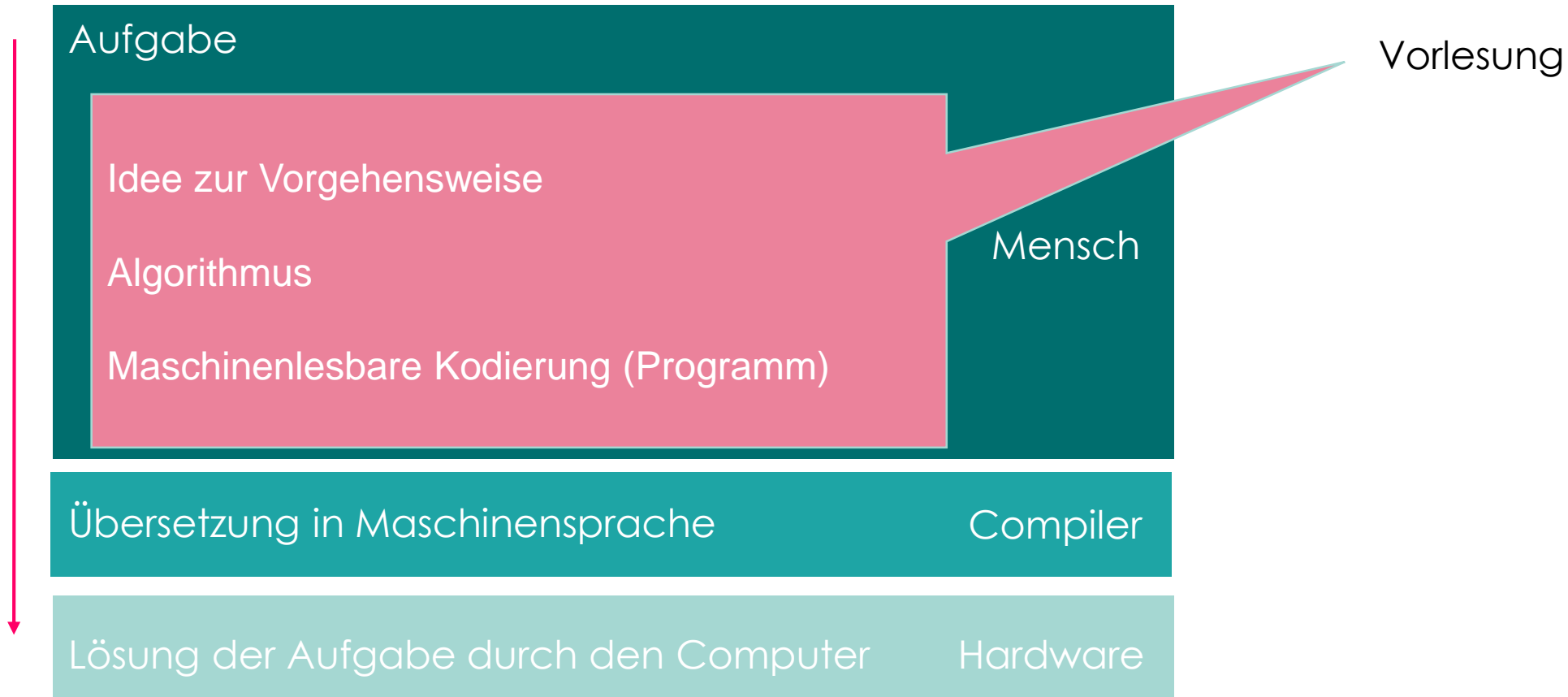
University
of Basel

Einige Grundlagen der Informatik

M. Lüthi – Universität Basel

Was heisst Programmieren

Exaktes Instruieren eines Computers, eine bestimmte Aufgabe zu lösen.



Arten von Programmen

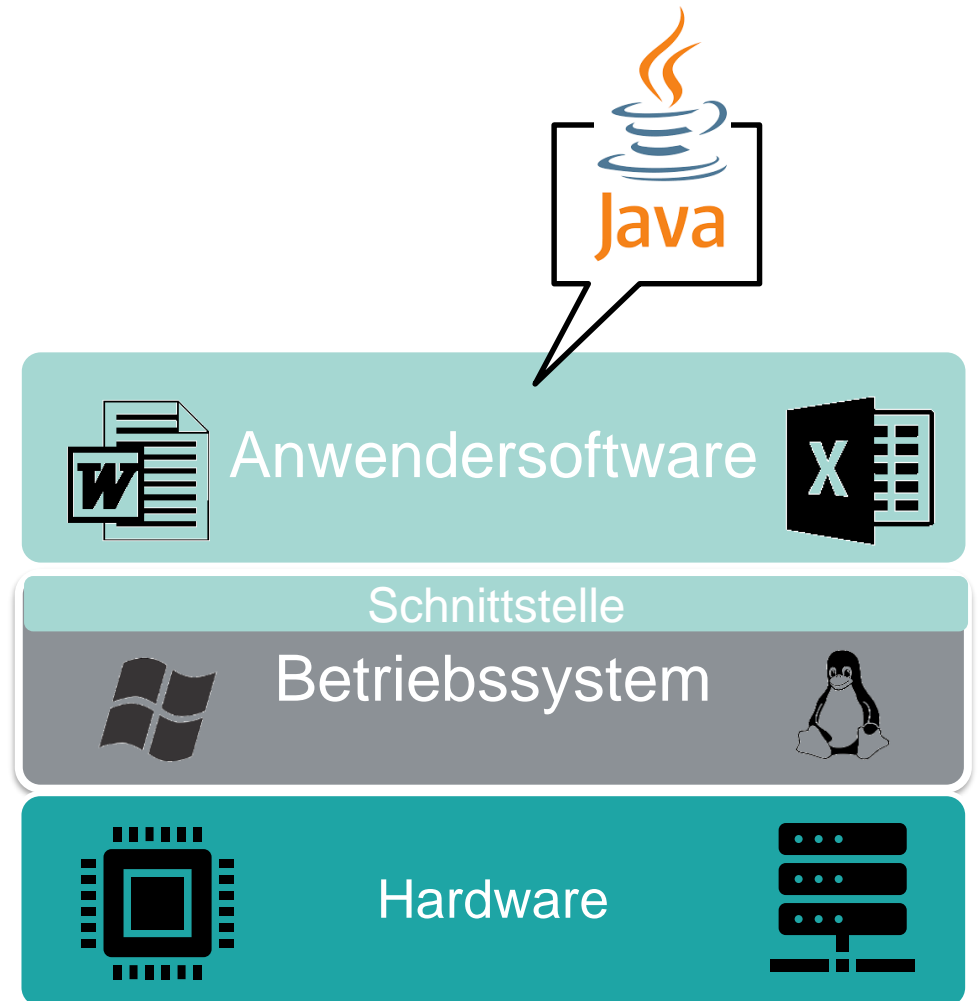
Anwendersoftware:

erlaubt die Lösung allgemeiner Aufgabenstellungen.

z. B. Textverarbeitung, Tabellenkalkulation, Bildbearbeitung, Buchhaltung, Produktionsplanung, Lohn und Gehaltsabrechnung, Spiele...

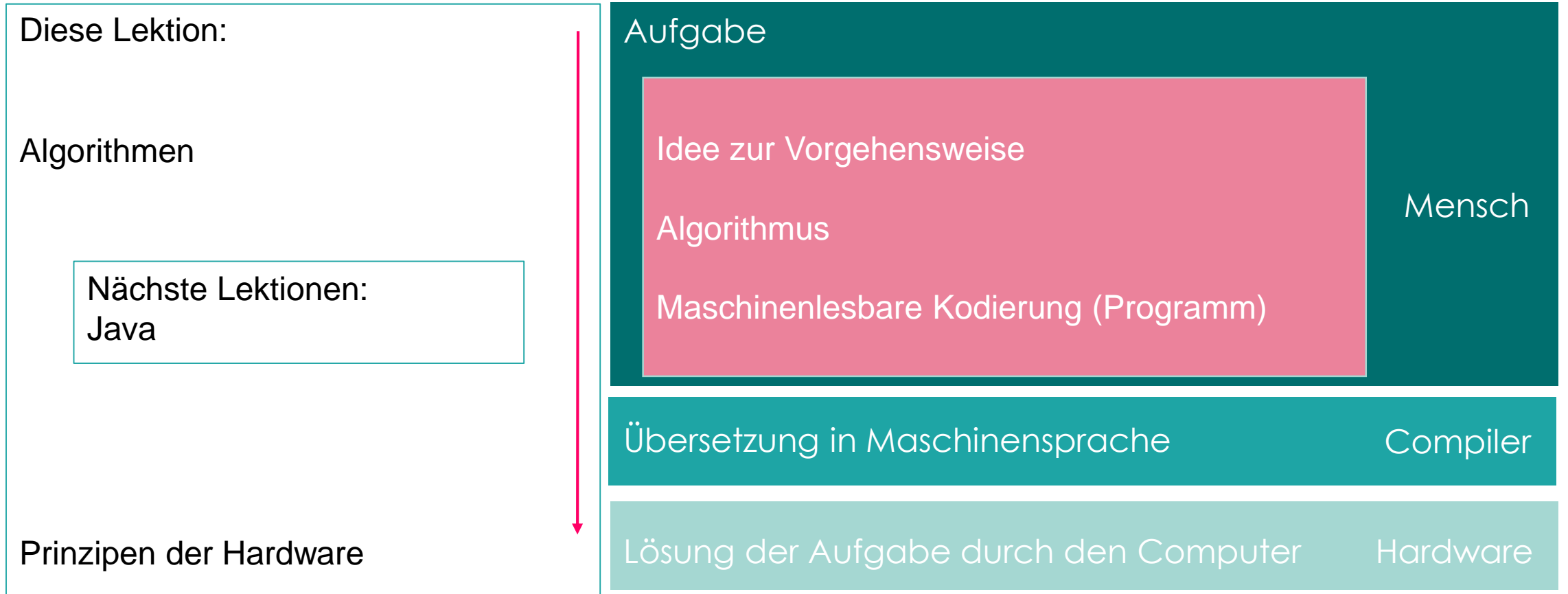
– Systemsoftware:

hilft beim Betrieb des Rechners und bei der Konstruktion der Anwendersoftware. Systemsoftware umfasst neben Datenbanksystemen, Übersetzern (compiler) etc. in jedem Fall das Betriebssystem.



Agenda

Trennung Algorithmus und Hardware in Java nicht ganz perfekt.

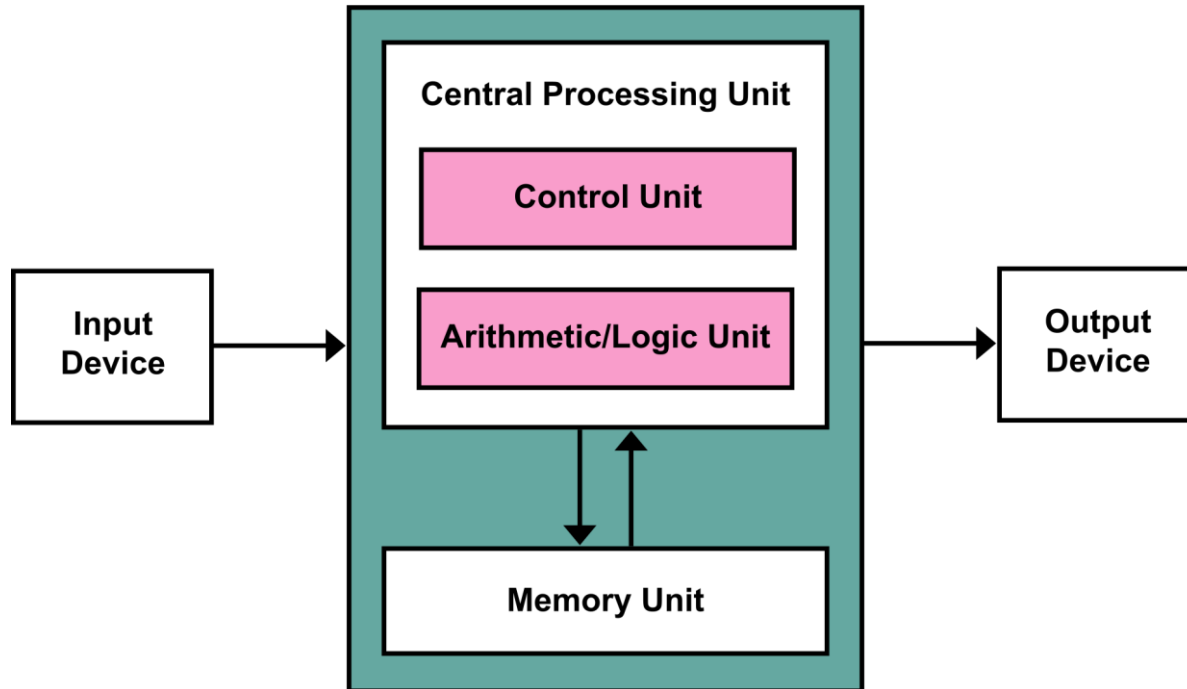




University
of Basel

Hardware

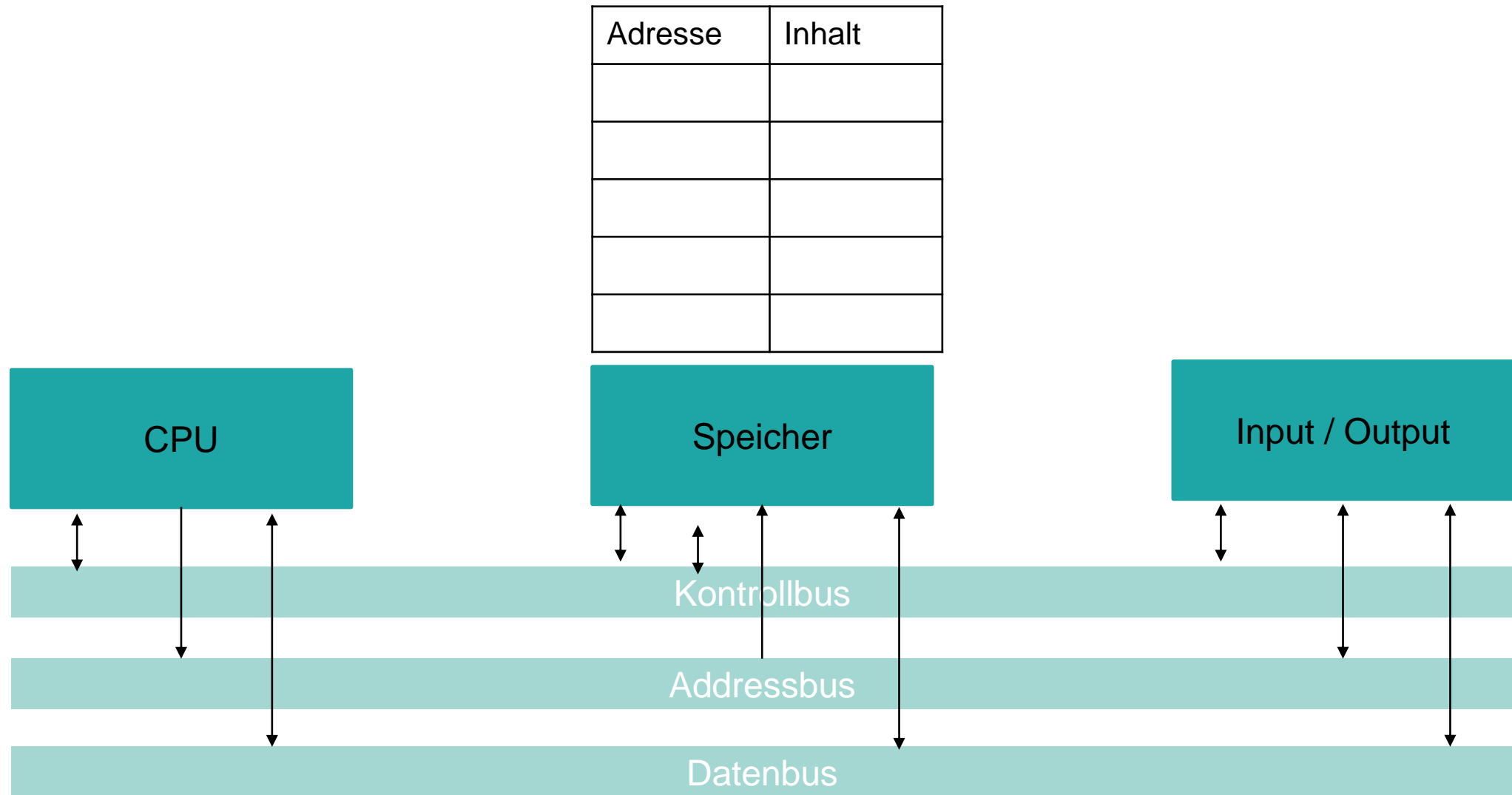
Von-Neumann Architektur



By Kapoht - Own work, CC BY-SA 3.0,
<https://commons.wikimedia.org/w/index.php?curid=25789639a>

Grundsätzlicher Aufbau moderner Computer

Speicherzugriff



Speicher

Kleinste Speichereinheit hat 2 Zustände

- 1 Bit
- Zustände werden i.A. mit 0 und 1 bezeichnet

Mit 2 Speichereinheiten $2^2=4$ Zustände darstellbar

Bit 0	Bit 1	Zustand
0	0	0
0	1	1
1	0	2
1	1	3

Mit 8 Bit $2^8=256$ Zustände darstellbar

8 Bit = 1 Byte

Heutzutage sind Bytes die kleinsten adressierbaren Speichereinheiten
Kleinere Einheiten müssen aus einem Byte extrahiert werden

Intermezzo: Zahlensystem

Ein **Zahlensystem** (number system) besteht aus

- endlich vielen Ziffern (digits) und
- einer Vorschrift
wie werden Zeichenreihen als Zahl interpretiert?

Arabische Zahlensysteme zur Basis β

- Natürliche Zahl z wird geschrieben als Polynom

$$z = \sum_{i=0}^{n-1} z_i \beta^i, \quad 0 \leq z_i \leq \beta$$

Intermezzo: Binärsystem

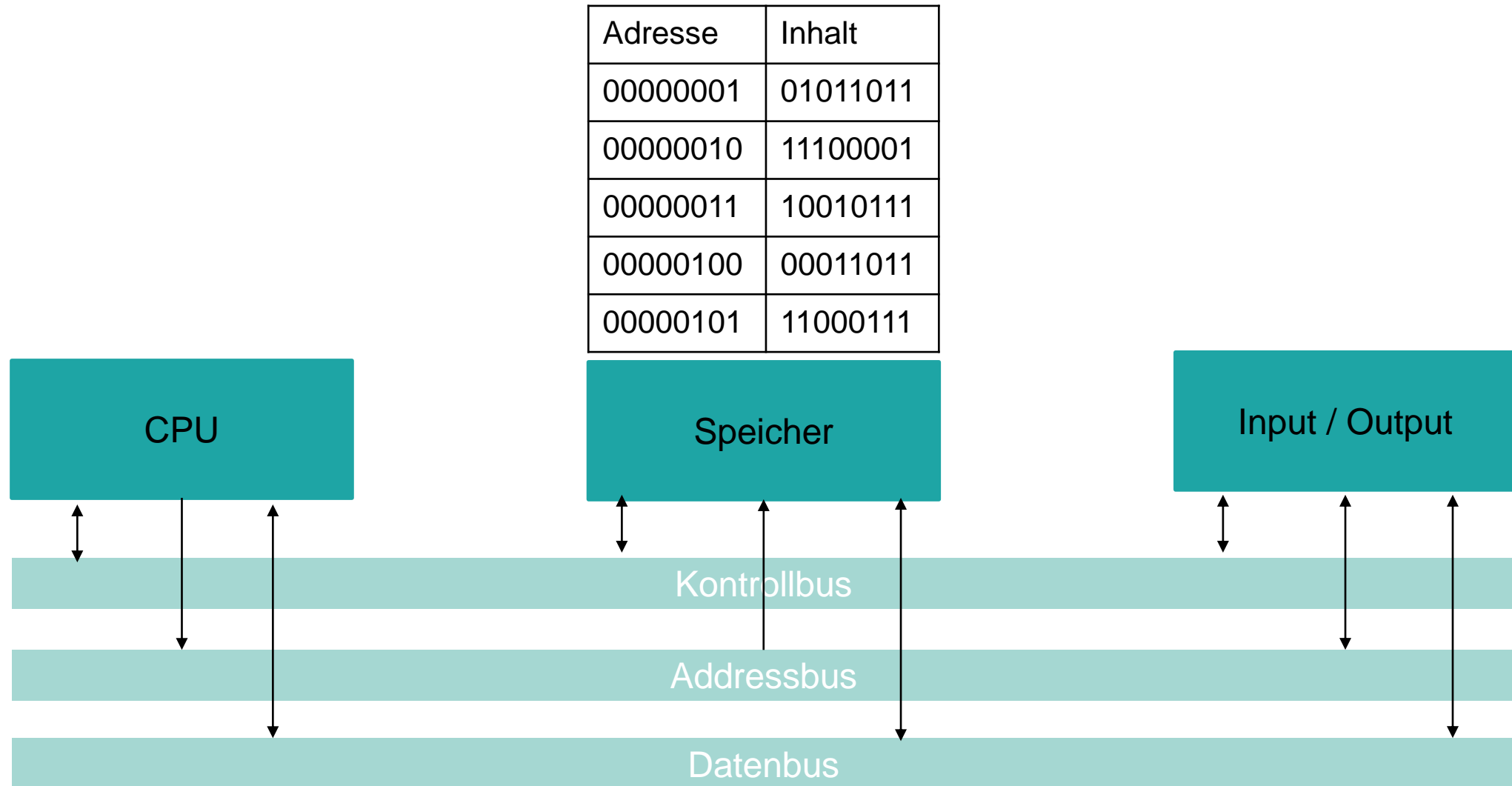
Basis $\beta = 2$

$$z = \sum_{i=0}^{n-1} z_i 2^i, \quad 0 \leq z_i \leq 1$$

Beispiele:

- $1_d = 1 \cdot 2^0 = 1_b$
 - $7_d = 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 111_b$
 - $9_d = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 1001_b$
-

Speicher & Adressierung



Quiz: Speicher

Wodurch wird der maximal mögliche Speicher in einem Computer begrenzt?

Wie viele Bits können auf einmal aus dem Speicher gelesen werden oder in den Speicher geschrieben werden?

Welcher Dezimalzahl entspricht die Zahl 10101111?

Welcher Binärzahl entspricht die Zahl 12?

Welcher Zahl im 16-er System (Hexadezimalsystem) entspricht die Dezimalzahl 65?

Variablen

Variablen sind Namen für Speicheradressen

- Einfacher zu merken als Zahl

Variable hat einen Wert

- Inhalt des Speichers an dieser Adresse

Variable hat einen Typ

- Wie gross ist Speicherstelle?
- Wie muss Bitmuster interpretiert werden?

Kleine Zahl $x = 11$
Grosse Zahl $y = 256$

Zeichen $z = 'a'$

Adresse	Inhalt
00000001	00001011
00000010	00000001
00000011	00000001
00000100	10010001
00000101	11000111

Elementare Datentypen

byte	8 Bit Zahl	$-2^7 \dots 2^7 - 1$	(-128, , 127)
short	16 Bit-Zahl	$-2^{15} \dots 2^{15} - 1$	(-32768, , 32767)
int	32 Bit-Zahl	$-2^{31} \dots 2^{31} - 1$	(-2 147 483 648, , 2 147 483 647)
long	64 Bit-Zahl	$-2^{63} \dots 2^{63} - 1$	
float	32 Bit IEEE-754-1985 Gleitkommazahl		
Double	64 Bit IEEE-754-1985 Gleitkommazahl		
char	16 Bit Unicode		
boolean	Wahrheitswert, <i>false</i> oder <i>true</i>		

Gleitkommazahlen

Darstellung einer Floating-Point-Zahl (IEEE 754-1985)

$$z = (-1)^v \cdot \text{Mantisse} \cdot 2^{\text{Exponent}}$$

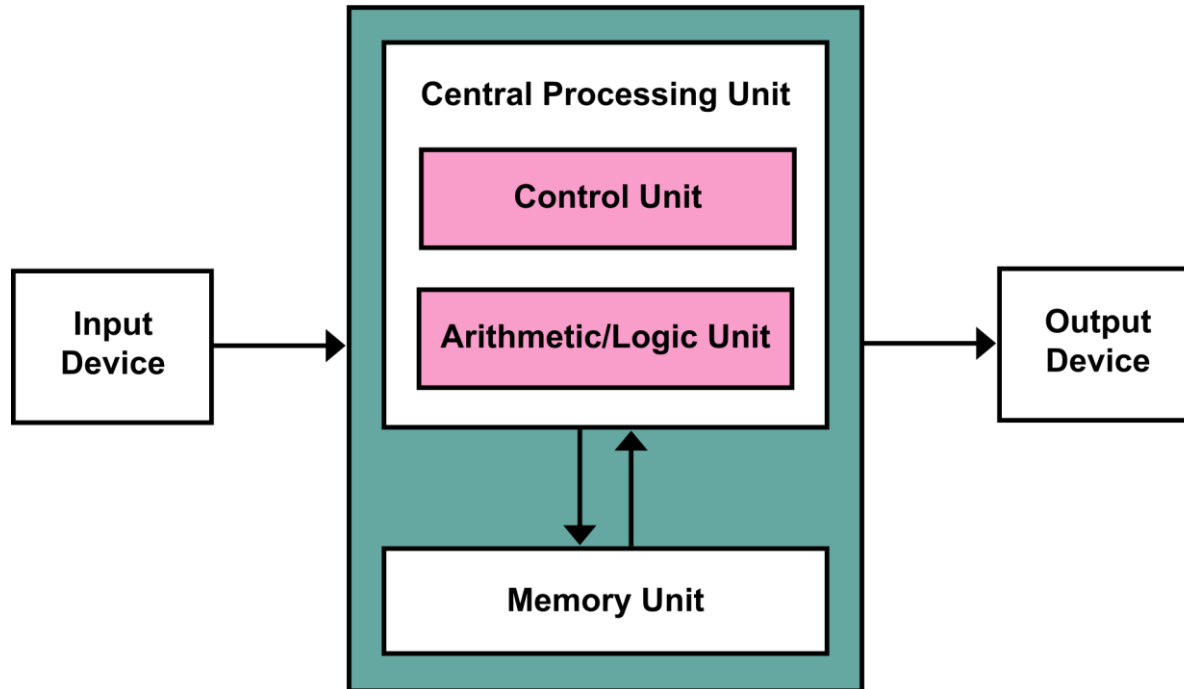
float

v	Exponent	Mantisse
1 Bit	8 Bit	23 Bit

double

v	Exponent	Mantisse
1 Bit	11 Bit	52 Bit

Von-Neumann Architektur : Befehle



By Kapoht - Own work, CC BY-SA 3.0,
<https://commons.wikimedia.org/w/index.php?curid=25789639a>

Befehle: Operationen mit den Speicherzellen

```
org100h
start:
mov dx, hello
mov ah, 09h
int 21h
mov al, 0
mov ah, 4Ch
int 21 h
section.data
hello: db 'hello, world', 13, 10, '$'
```

Jeder Befehl steht für
eindeutiger Bitsequenz

Charakteristik der Von-Neumann Architektur. Gleicher Speicher für Daten und Befehle

Diskutieren Sie

Jedes Programm, das wir schreiben, wird irgendwann in Maschinensprache übersetzt.

- Weshalb programmiert man typischerweise nicht direkt in Maschinensprache?
 - Was könnten Vorteile sein, direkt in Maschinensprache zu programmieren?
 - Gibt es Programme, die in Maschinensprache geschrieben werden müssen?
-



University
of Basel

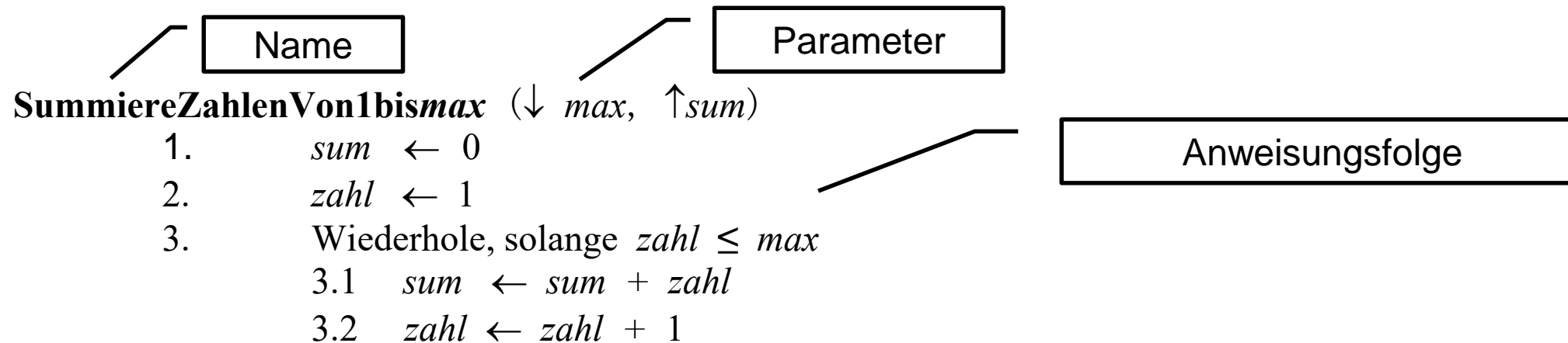
Algorithmen

Algorithmus

Algorithmus:

Schrittweises, präzises Verfahren zur Lösung eines Problems

Problem: Summiere die Zahlen von 1 bis max $sum = \sum_{i=1}^{max} i$



Programm

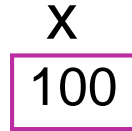
Beschreibung eines Algorithmus in einer Programmiersprache

Variablen

Variablen: Behälter für Werte



Variablen können Wert ändern
 $x \leftarrow x + 1$



Variablen haben Datentyp (Menge erlaubter Werte)

Variablentyp

Werte



Zahl



...

- in eine Zahlenvariable passen nur Zahlen



Zeichen



...

- in eine Zeichenvariable passen nur Zeichen

Anweisungen

Beispiel: Wertzuweisung

$x \leftarrow x + 1$



Variable

The diagram shows the assignment statement $x \leftarrow x + 1$. A pink callout box labeled 'Variable' points to the variable 'x' on the left. A teal callout box labeled 'Ausdruck' points to the expression 'x + 1' on the right.

Ausdruck

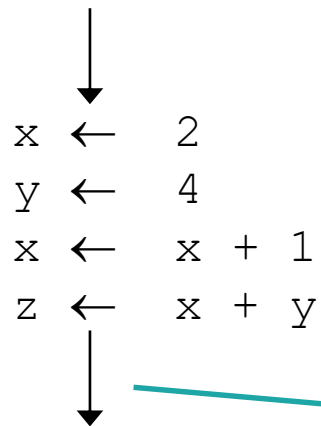
1. Werte Ausdruck aus
2. Weise seinen Wert der Variablen zu

Beliebige, klar definierte Anweisungen sind möglich

- *Keine Einschränkung durch Programmiersprache oder Hardware*
-

Anweisungsfolge (Sequenz)

Ablaufdiagramm



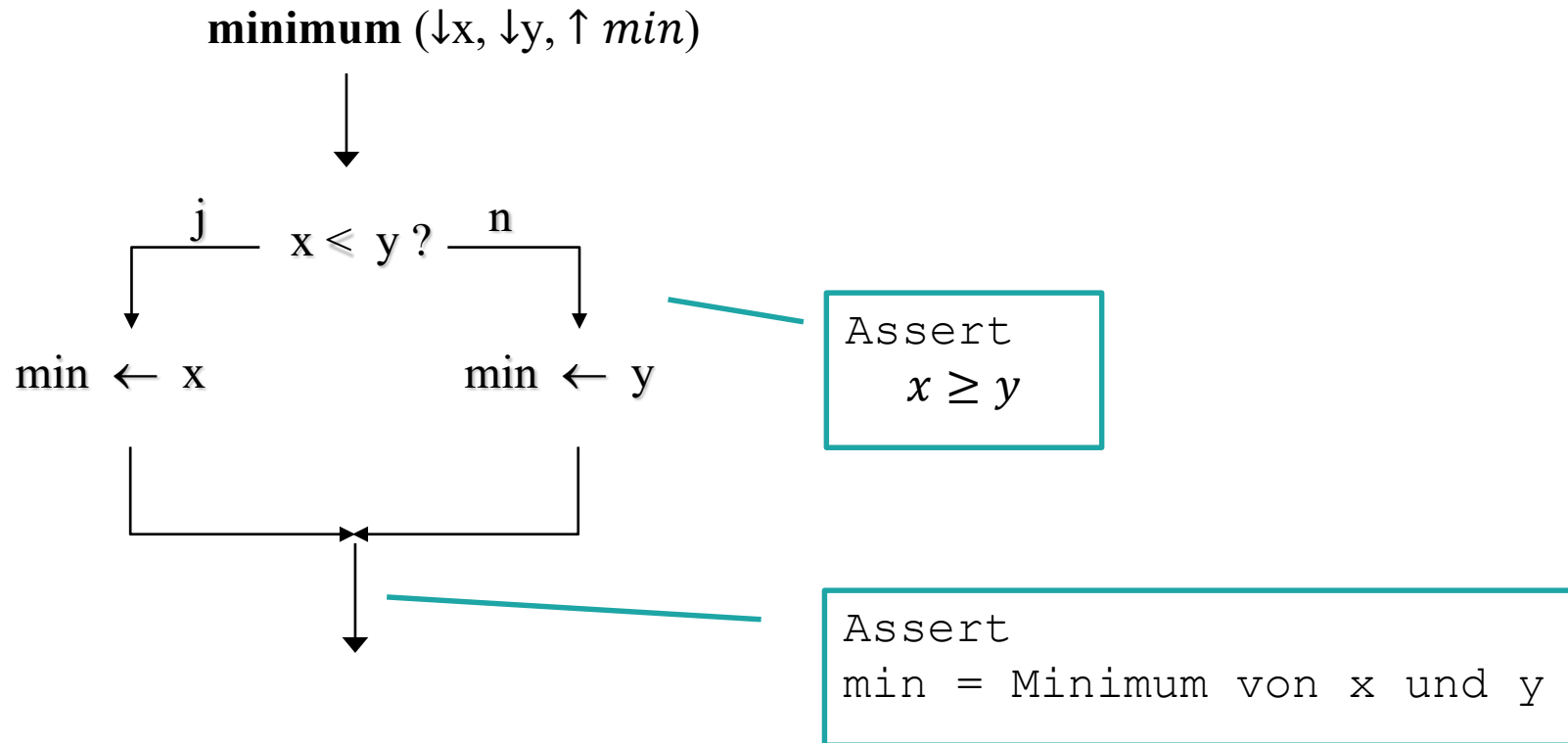
Assert
 $x = 3, y = 4, z = 7$

Assertion (Zusicherung)

Aussage über den Zustand des Algorithmus
an einer bestimmten Stelle

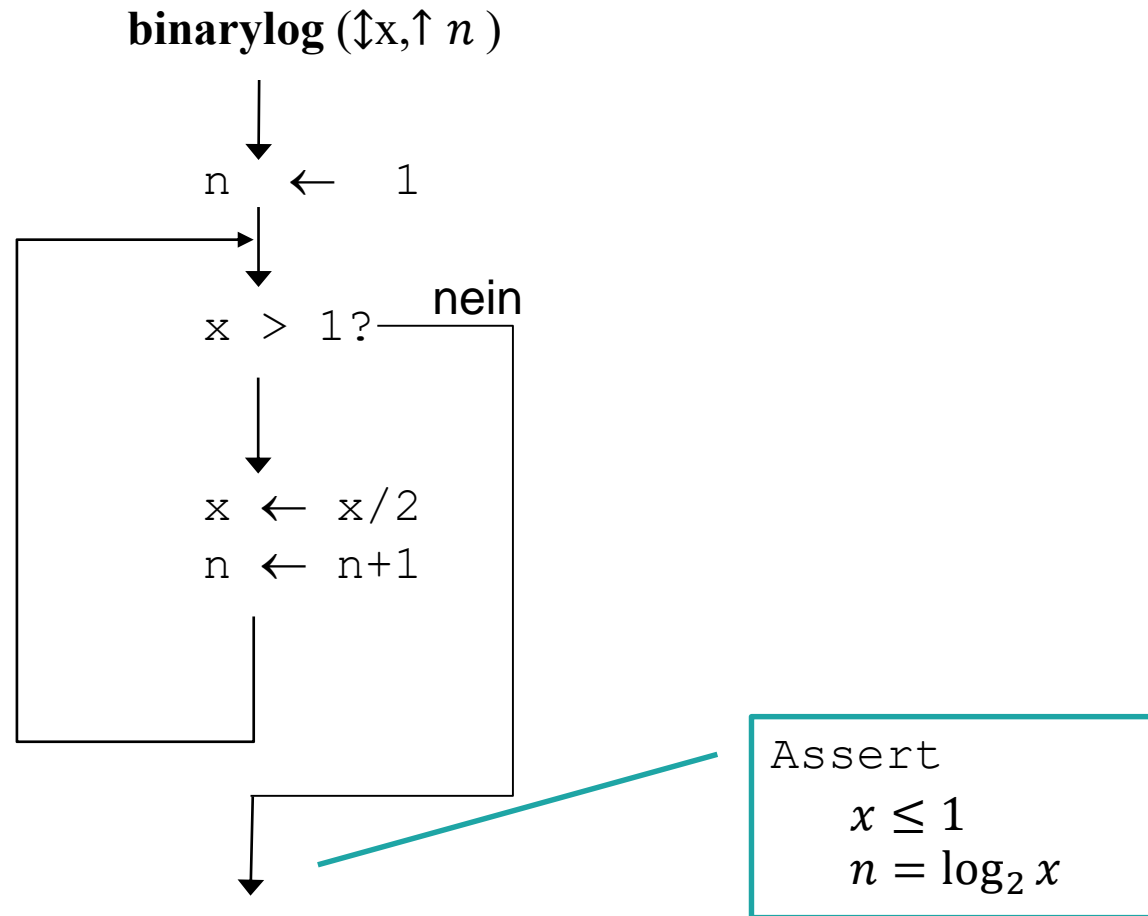
Auswahl (Verzweigung)

Beispiel: Suche das Minimum der zwei Zahlen x und y



Wiederholung (Schleife, Iteration)

Beispiel: Suche die grösste ganze Zahl n mit 2^n kleiner oder gleich x .



Beispiel: Vertausche zwei Variableninhalte

swap ($\uparrow x$, $\uparrow y$)



$h \leftarrow x$

$x \leftarrow y$

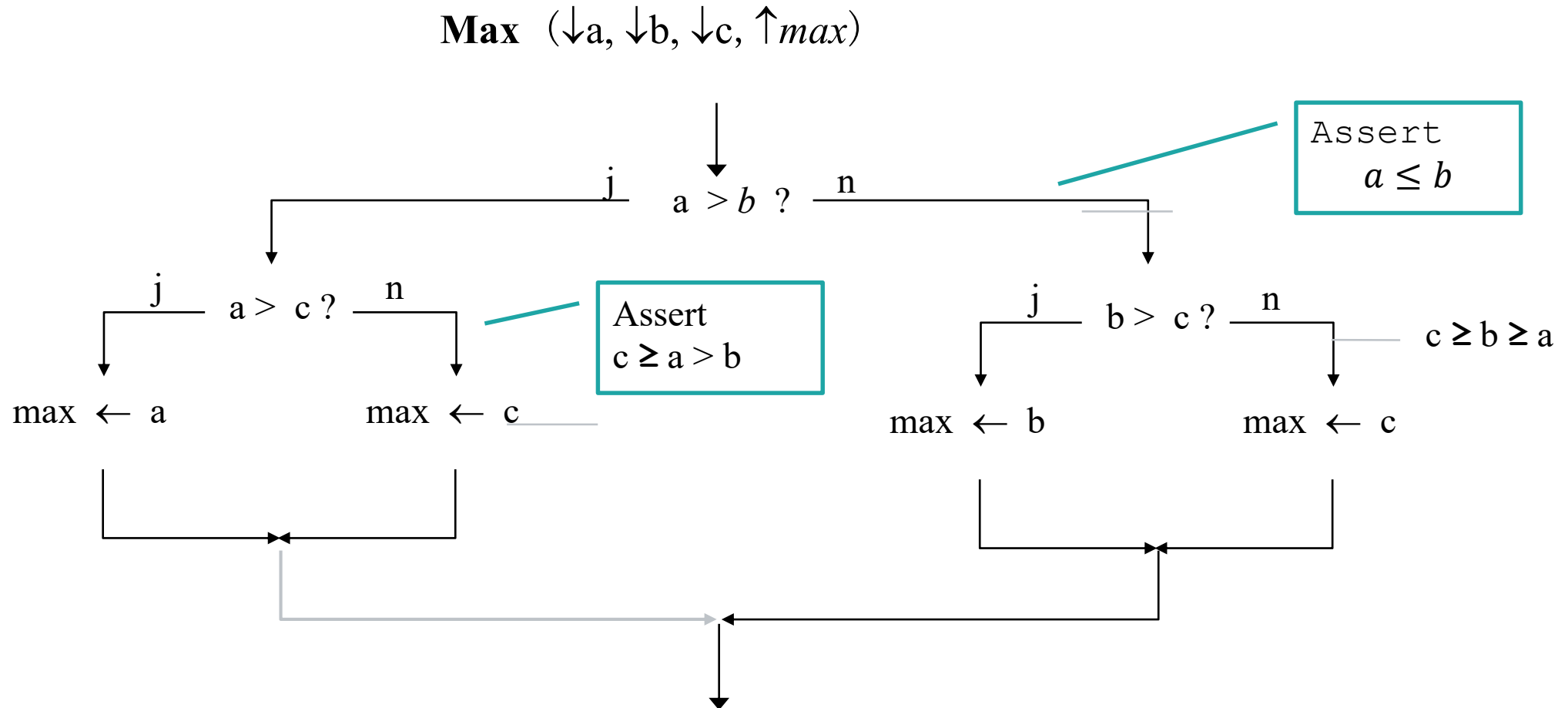
$y \leftarrow h$



Schreibtischtest

x	y	h
3 2	2 3	3

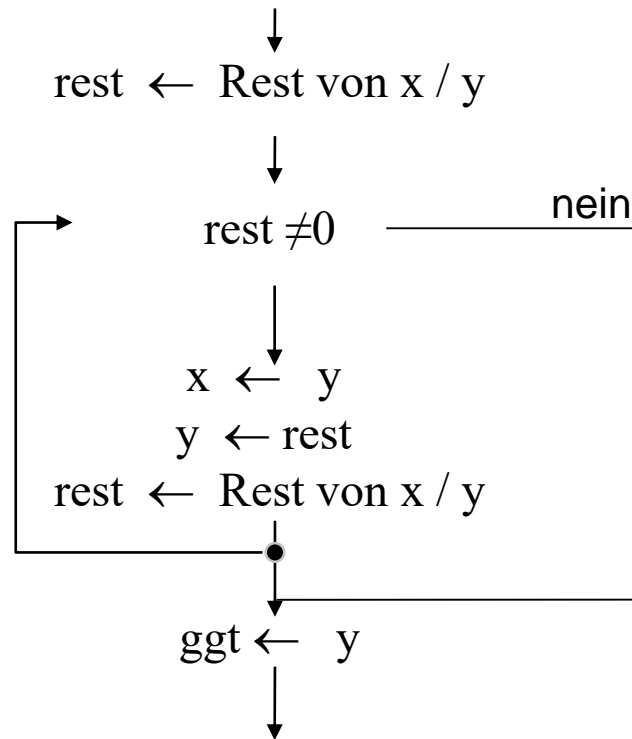
Beispiel: Bestimme Maximum dreier Zahlen



Beispiel: Euklidischer Algorithmus

Berechnet den größten gemeinsamen Teiler zweier Zahlen x und y

GGT ($\downarrow x, \downarrow y, \uparrow \text{ggt}$)

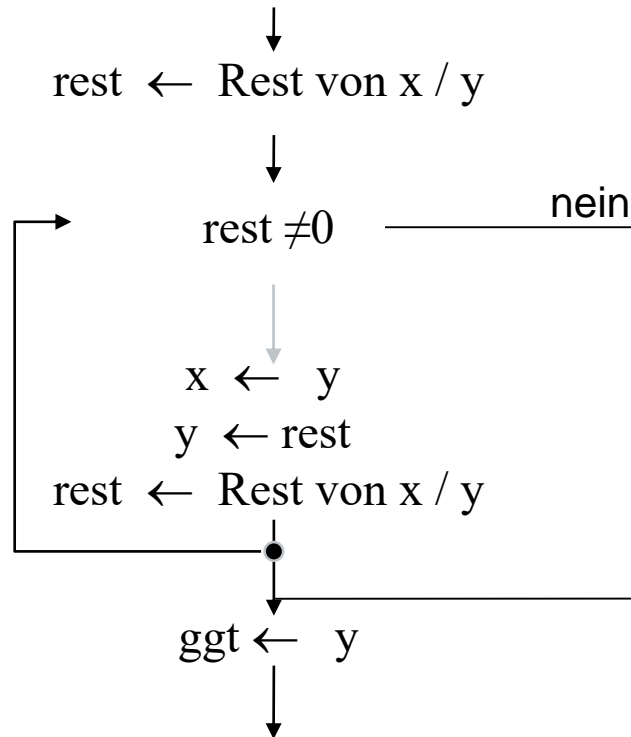


Schreibtischtest

x	y	rest
28	20	8
20	8	4
8	4	0

Euklidischer Algorithmus: Korrektheitsbeweis

GGT ($\downarrow x, \downarrow y, \uparrow \text{ggt}$)



Warum funktioniert dieser Algorithmus?

$(\text{ggt teilt } x) \ \& \ (\text{ggt teilt } y)$

$\text{ggt teilt } (x - y)$

$\text{ggt teilt } (x - q \cdot y)$

ggt teilt rest

$\text{ggt}(x, y) = \text{ggt}(y, \text{rest})$



University
of Basel

Grammatiken

Beschreibung von Programmiersprachen

Syntax

Regeln, nach denen Sätze gebaut werden dürfen

z. B.: Zuweisung = Variable " \leftarrow " Ausdruck.

Semantik

Bedeutung der Sätze

z. B.: werte Ausdruck aus und weise ihn der Variablen zu

Grammatik

Menge von Syntaxregeln

z. B. Grammatik der ganzen Zahlen

Ziffer = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9".

Zahl = Ziffer {Ziffer}.

EBNF (Erweiterte Backus- Naur- Form)

Metazeichen	Bedeutung	Beispiel	beschreibt
=	trennt Regelseiten		
.	schließt Regel ab		
	trennt Alternativen	$x \mid y$	x, y
()	klammert Alternativen	$(x \mid y) z$	xz, yz
[]	wahlweises Vorkommen	$[x] y$	xy, y
{ }	0.. n- maliges Vorkommen	$\{x\} y$	$y, xy, xxy, xxxy, \dots$

Beispiele

Grammatik der Gleitkommazahlen

Ziffer = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9".

Zahl = Ziffer {Ziffer}.

Gleitkommazahl = Zahl "." Zahl [" E" ["+" | "-"] Zahl].

Quiz

Welche der folgenden Ausdrücke sind syntaktisch korrekte Gleitkommazahlen

- a) 1
- b) 1.1
- c) .12
- d) 1.3E7
- e) +1.3E-7

Welche der folgenden Ausdrücke sind syntaktisch korrekt gemäss folgender Grammatik: $A\{[_]A\}$

- a) A_A
 - b) AA_AAA
 - c) _A
 - d) A_
 - e) A__A
-