

4 - Algorithmen & Objektivität

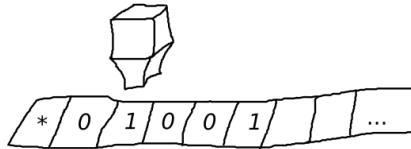
Marianne Maertens

Technische Universität Berlin

Wintersemester 2025/2026

Themen

1. Wiederholung
2. Was ist ein Algorithmus
3. Repräsentation von Algorithmen
4. Turing-Maschinen
5. Algorithmen und Objektivität



Geschichte der Informatik

- enge Verbindung von:

Automatik

automatos (altgriech.) - selbst (autos) +
denken (men)

Rechengeräte - praktische Fragen,
Arbeitserleichterung

Ingenieurdisziplin - Rechnen
mechanisieren

→ **Entwicklung des Computers**
(Hardware)

Information

informare (lat.) - durch Unterweisung
bilden

Erkenntnisse - theoretische,
philosophische Fragen

Mathematik - Denken formalisieren

→ **Entwicklung von Algorithmen**
(Software)

Lesen Sie den Text

Wie werden Algorithmen im Text definiert?

Welche Bedeutung wird ihnen zugeschrieben?



Lesen Sie den Text

Wie werden Algorithmen im Text definiert?

- klar definierte Folge von Schritten, die eindeutig beschreibt, wie alle Probleme eines bestimmten Typs gelöst werden können

Welche Bedeutung wird ihnen zugeschrieben?



Lesen Sie den Text

Wie werden Algorithmen im Text definiert?

- klar definierte Folge von Schritten, die eindeutig beschreibt, wie alle Probleme eines bestimmten Typs gelöst werden können

Welche Bedeutung wird ihnen zugeschrieben?

- die Intelligenz, die zur Lösung des Problems erforderlich ist, steckt im Algorithmus



Lesen Sie den Text

Wie werden Algorithmen im Text definiert?

- klar definierte Folge von Schritten, die eindeutig beschreibt, wie alle Probleme eines bestimmten Typs gelöst werden können

Welche Bedeutung wird ihnen zugeschrieben?

- die Intelligenz, die zur Lösung des Problems erforderlich ist, steckt im Algorithmus
- das Intelligenzniveau von Maschinen hängt davon ab, wie viel Intelligenz durch Algorithmen abgebildet werden kann



Lesen Sie den Text

Wie werden Algorithmen im Text definiert?

- klar definierte Folge von Schritten, die eindeutig beschreibt, wie alle Probleme eines bestimmten Typs gelöst werden können

Welche Bedeutung wird ihnen zugeschrieben?

- die Intelligenz, die zur Lösung des Problems erforderlich ist, steckt im Algorithmus
- das Intelligenzniveau von Maschinen hängt davon ab, wie viel Intelligenz durch Algorithmen abgebildet werden kann
- Maschinen können nur solche Aufgaben ausführen, für die es einen Algorithmus gibt. Wenn für eine Problemlösung kein Algorithmus existiert, liegt die Lösung des Problems außerhalb der Fähigkeit von Maschinen.



Euklidischer Algorithmus (300 v. Chr.)

- Algorithmus zum Finden des größten gemeinsamen Teilers zweier Zahlen
- ältester Algorithmus, der den Namen verdient
- wird noch heute benutzt, weil es keinen besseren gibt

Gegeben: 2 Zahlen

Gesucht: deren größter gemeinsamer Teiler (ggT)

Wie funktioniert der klassische Euklidische Algorithmus?

Der Euklidische Algorithmus - geometrisch

gegeben: 75 und 27

Der Euklidische Algorithmus - geometrisch

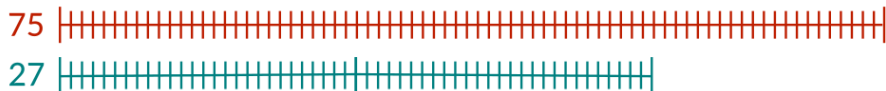
gegeben: 75 und 27



Quelle https://www.youtube.com/watch?v=i_Zb43TE0Ac

Der Euklidische Algorithmus - geometrisch

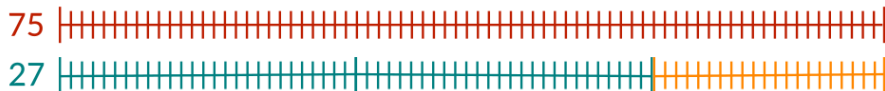
gegeben: 75 und 27



Quelle https://www.youtube.com/watch?v=i_Zb43TE0Ac

Der Euklidische Algorithmus - geometrisch

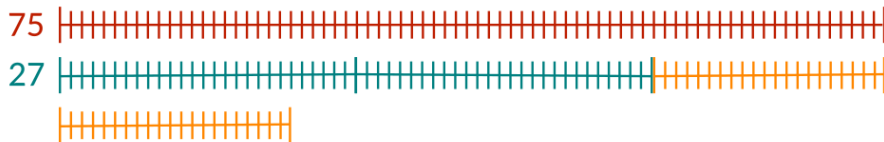
gegeben: 75 und 27



Quelle https://www.youtube.com/watch?v=i_Zb43TE0Ac

Der Euklidische Algorithmus - geometrisch

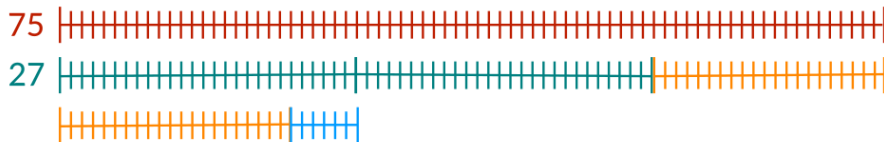
gegeben: 75 und 27



Quelle https://www.youtube.com/watch?v=i_Zb43TE0Ac

Der Euklidische Algorithmus - geometrisch

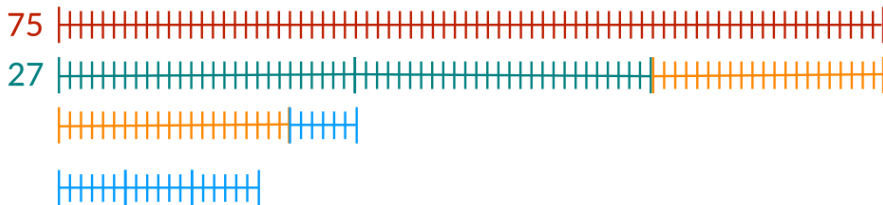
gegeben: 75 und 27



Quelle https://www.youtube.com/watch?v=i_Zb43TE0Ac

Der Euklidische Algorithmus - geometrisch

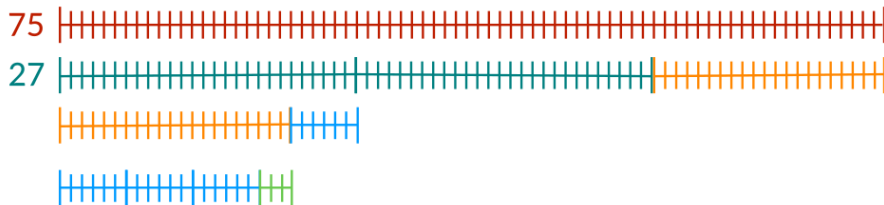
gegeben: 75 und 27



Quelle https://www.youtube.com/watch?v=i_Zb43TE0Ac

Der Euklidische Algorithmus - geometrisch

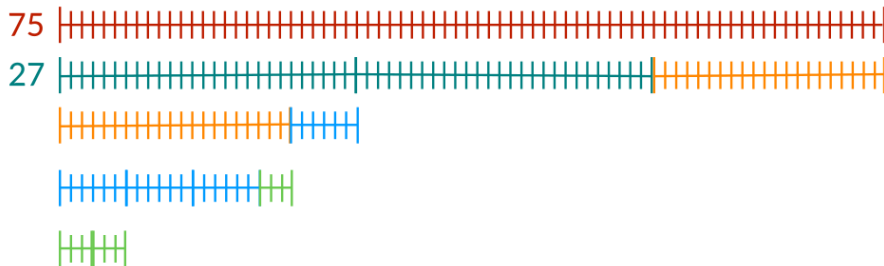
gegeben: 75 und 27



Quelle https://www.youtube.com/watch?v=i_Zb43TE0Ac

Der Euklidische Algorithmus - geometrisch

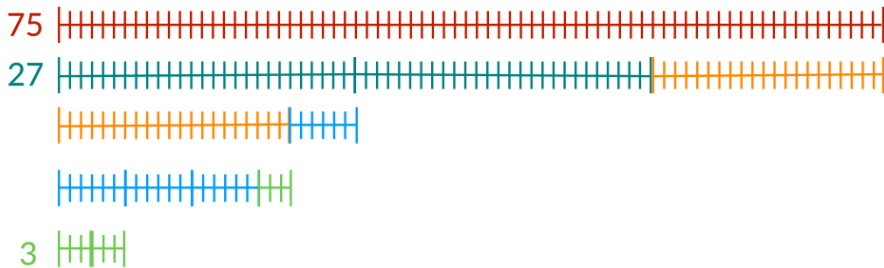
gegeben: 75 und 27



Quelle https://www.youtube.com/watch?v=i_Zb43TE0Ac

Der Euklidische Algorithmus - geometrisch

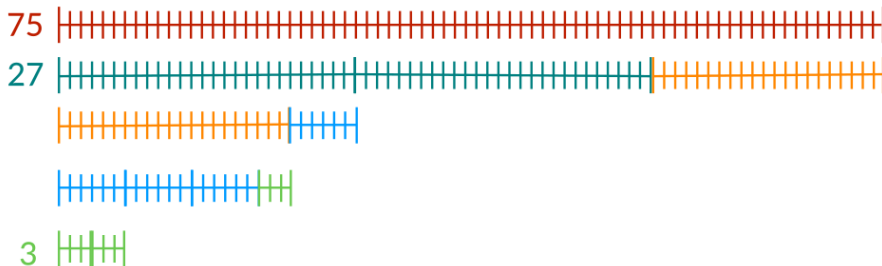
gegeben: 75 und 27



Quelle https://www.youtube.com/watch?v=i_Zb43TE0Ac

Der Euklidische Algorithmus - geometrisch

gegeben: 75 und 27



Quelle https://www.youtube.com/watch?v=i_Zb43TE0Ac

$$\text{ggT}(75, 27) = 3$$

Das Sieb des Eratosthenes (276-194 v.Chr.)



- Algorithmus zum Finden aller Primzahlen in einer Liste von Zahlen der Länge n
- Primzahl ist eine ganze Zahl, $x \in \mathbb{N} > 1$, die ausschließlich durch sich selbst und durch 1 **teilbar** ist

Das Sieb des Eratosthenes (276-194 v.Chr.)



- Algorithmus zum Finden aller Primzahlen in einer Liste von Zahlen der Länge n
- Primzahl ist eine ganze Zahl, $x \in \mathbb{N} > 1$, die ausschließlich durch sich selbst und durch 1 **teilbar** ist

1 2 3 4 5 6 7 8

9 10 11 12 13 14 15

16 17 18 19 20 21 22

23 24 25 26

Das Sieb des Eratosthenes (276-194 v.Chr.)



- Algorithmus zum Finden aller Primzahlen in einer Liste von Zahlen der Länge n
- Primzahl ist eine ganze Zahl, $x \in \mathbb{N} > 1$, die ausschließlich durch sich selbst und durch 1 **teilbar** ist

~~1~~ 2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26

Das Sieb des Eratosthenes (276-194 v.Chr.)



- Algorithmus zum Finden aller Primzahlen in einer Liste von Zahlen der Länge n
- Primzahl ist eine ganze Zahl, $x \in \mathbb{N} > 1$, die ausschließlich durch sich selbst und durch 1 **teilbar** ist

~~1~~ 2 3 4 5 6 7 8

9 10 11 12 13 14 15

16 17 18 19 20 21 22

23 24 25 26

Das Sieb des Eratosthenes (276-194 v.Chr.)



- Algorithmus zum Finden aller Primzahlen in einer Liste von Zahlen der Länge n
- Primzahl ist eine ganze Zahl, $x \in \mathbb{N} > 1$, die ausschließlich durch sich selbst und durch 1 **teilbar** ist

~~1~~ 2 3 ~~4~~ 5 ~~6~~ 7 ~~8~~
9 ~~10~~ 11 ~~12~~ 13 ~~14~~ 15
~~16~~ 17 ~~18~~ 19 ~~20~~ 21 ~~22~~
23 ~~24~~ 25 ~~26~~

- streiche Vielfache der aktuellen Primzahl ("sieben")

Das Sieb des Eratosthenes (276-194 v.Chr.)



- Algorithmus zum Finden aller Primzahlen in einer Liste von Zahlen der Länge n
- Primzahl ist eine ganze Zahl, $x \in \mathbb{N} > 1$, die ausschließlich durch sich selbst und durch 1 **teilbar** ist

~~1~~ 2 3 ~~4~~ 5 ~~6~~ 7 ~~8~~
~~9~~ ~~10~~ 11 ~~12~~ 13 ~~14~~ ~~15~~
~~16~~ 17 ~~18~~ 19 ~~20~~ ~~21~~ ~~22~~
23 ~~24~~ 25 ~~26~~

- streiche Vielfache der aktuellen Primzahl ("sieben")

Das Sieb des Eratosthenes (276-194 v.Chr.)



- Algorithmus zum Finden aller Primzahlen in einer Liste von Zahlen der Länge n
- Primzahl ist eine ganze Zahl, $x \in \mathbb{N} > 1$, die ausschließlich durch sich selbst und durch 1 **teilbar** ist

~~1~~ 2 3 ~~4~~ 5 ~~6~~ 7 ~~8~~
~~9~~ ~~10~~ 11 ~~12~~ 13 ~~14~~ ~~15~~
~~16~~ 17 ~~18~~ 19 ~~20~~ ~~21~~ ~~22~~
23 ~~24~~ ~~25~~ ~~26~~

- streiche Vielfache der aktuellen Primzahl ("sieben")

Das Sieb des Eratosthenes (276-194 v.Chr.)



- Algorithmus zum Finden aller Primzahlen in einer Liste von Zahlen der Länge n
- Primzahl ist eine ganze Zahl, $x \in \mathbb{N} > 1$, die ausschließlich durch sich selbst und durch 1 **teilbar** ist

1 2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26

- streiche Vielfache der aktuellen Primzahl ("sieben")
- größte Siebzahl einer Liste mit n Zahlen ist \sqrt{n}
- übrige Zahlen sind Primzahlen

Herkunft des Begriffs Algorithmus

- ~ 780 – 850 Abu Dscha'far Muhammad ibn Musa al-Chwarizmi
- “Dixit *Algorizmi*” = “also sprach al-Chwarizmi”
 - als Gütezeichen einer Rechnung (ab ca. 1200)



Herkunft des Begriffs Algorithmus

- ~ 780 – 850 Abu Dscha'far Muhammad ibn Musa al-Chwarizmi
- “Dixit *Algorizmi*” = “also sprach al-Chwarizmi”
 - als Gütezeichen einer Rechnung (ab ca. 1200)

Definition

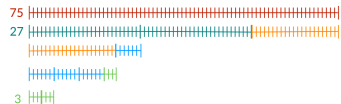
- Eine eindeutige Handlungsanweisung zur Lösung eines Problems. Dabei wird eine bestimmte Eingabe in eine bestimmte Ausgabe überführt.
- Intelligenz steckt in der Anweisung. Der Ausführende muss nicht verstehen, was er tut.

(Hisāb *al-gābr* wa-'l-muqābala - *Algebra*)



Übung

1. Bestimmen Sie $\text{ggT}(8,5)$ mit Hilfe des klassischen Euklidischen Algorithmus. Schreiben Sie die Schritte in Form von Gleichungen auf.
2. Beschreiben Sie in Worten, was der Algorithmus macht.



Übung

1. Bestimmen Sie $\text{ggT}(8,5)$ mit Hilfe des klassischen Euklidischen Algorithmus. Schreiben Sie die Schritte in Form von Gleichungen auf.
2. Beschreiben Sie in Worten, was der Algorithmus macht.

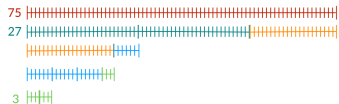
$$8 = 1 * 5 + 3$$

$$5 = 1 * 3 + 2$$

$$3 = 1 * 2 + 1$$

$$2 = 2 * 1 + 0$$

$$\text{ggT}(8,5) = 1$$



Übung

1. Bestimmen Sie $\text{ggT}(8,5)$ mit Hilfe des klassischen Euklidischen Algorithmus. Schreiben Sie die Schritte in Form von Gleichungen auf.
2. Beschreiben Sie in Worten, was der Algorithmus macht.

$$\begin{array}{l} 8 = 1 * 5 + 3 \\ 5 = 1 * 3 + 2 \\ 3 = 1 * 2 + 1 \\ 2 = 2 * 1 + 0 \end{array}$$

$$\text{ggT}(8,5) = 1$$

1. größere Zahl als Vielfaches der kleineren Zahl darstellen plus Rest



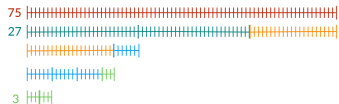
Übung

1. Bestimmen Sie $\text{ggT}(8,5)$ mit Hilfe des klassischen Euklidischen Algorithmus. Schreiben Sie die Schritte in Form von Gleichungen auf.
2. Beschreiben Sie in Worten, was der Algorithmus macht.

$$\begin{aligned} 8 &= 1 * 5 + 3 \\ 5 &= 1 * 3 + 2 \\ 3 &= 1 * 2 + 1 \\ 2 &= 2 * 1 + 0 \end{aligned}$$

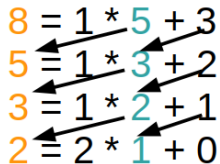
$$\text{ggT}(8,5) = 1$$

1. größere Zahl als Vielfaches der kleineren Zahl darstellen plus Rest
2. wenn der Rest null ist, dann ist die kleinere Zahl der ggT



Repräsentation von Algorithmen

mathematisch - arithmetisch

$$\begin{array}{l} 8 = 1 * 5 + 3 \\ 5 = 1 * 3 + 2 \\ 3 = 1 * 2 + 1 \\ 2 = 2 * 1 + 0 \end{array}$$


natürliche Sprache

1. größere Zahl als Vielfaches der kleineren Zahl plus Rest
2. wenn der Rest null ist → kleinere Zahl ist der ggT

Repräsentation von Algorithmen

mathematisch - arithmetisch

$$\begin{array}{l} 8 = 1 * 5 + 3 \\ 5 = 1 * 3 + 2 \\ 3 = 1 * 2 + 1 \\ 2 = 2 * 1 + 0 \end{array}$$

natürliche Sprache

1. größere Zahl als Vielfaches der kleineren Zahl plus Rest
2. wenn der Rest null ist → kleinere Zahl ist der ggT

mathematisch - geometrisch



Repräsentation von Algorithmen

mathematisch - arithmetisch

$$\begin{array}{l} 8 = 1 * 5 + 3 \\ 5 = 1 * 3 + 2 \\ 3 = 1 * 2 + 1 \\ 2 = 2 * 1 + 0 \end{array}$$

natürliche Sprache

1. größere Zahl als Vielfaches der kleineren Zahl plus Rest
2. wenn der Rest null ist → kleinere Zahl ist der ggT

mathematisch - geometrisch



Pseudocode

```
ggT_mod(a,b)
  while a % b != 0 # so lange der Rest nicht
    null ist
      if b > a:
        a, b = b, a
      a = a % b # größere Zahl als Vielfaches der
        Kleineren
  return b # kleinere Zahl ist ggT
```


Formalisieren des Denkens: Zeichensysteme

- Leibniz (1646-1716) träumte von einer Maschine, die Symbole manipulieren kann, um den Wahrheitswert mathematischer Aussagen zu bestimmen
- Voraussetzung: klare, formale Sprache
- 3. Jh. v. Chr. indischer Mathematiker Pingala beschreibt Zahlensystem bestehend aus 2 Zeichen
- Ende 17Jh. entwirft Leibniz das binäre Zahlensystem
 - kritisch bei der Entwicklung elektronischer Rechenmaschinen
 - Zahlen werden durch elektrische Zustände dargestellt (Strom an/aus)

Handwritten table from Pingala's 'Tabulara' manuscript, showing powers of 2 and their corresponding binary representations. The table is written in Devanagari script. The left column lists powers of 2 from 10^0 to 10^9 . The right column lists the corresponding binary representations, which are powers of 2 from 2^0 to 2^{10} .

10^0	Tabulara	ika	stabil
1	1	2	2^0
100	4	8	2^3
1000	8	16	2^4
10000	16	32	2^5
100000	32	64	2^6
1000000	64	128	2^7
10000000	128	256	2^8
100000000	256	512	2^9
1000000000	512	1024	2^{10}

[https:](https://de.wikipedia.org/wiki/Dualsystem#Entwicklung_des_Dualsystems)

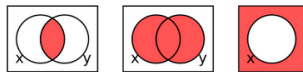
[//de.wikipedia.org/wiki/Dualsystem#](https://de.wikipedia.org/wiki/Dualsystem#Entwicklung_des_Dualsystems)

Entwicklung_des_Dualsystems

Formalisieren des Denkens: Boolesche Logik

- 1847 entwickelt George Boole (1815-1864) das nach ihm benannte Logikkalkül
- es war die Grundlage für die Zwei-Elementige Boolesche Algebra
- verallgemeinert Eigenschaften der logischen Operatoren UND (Konjunktion), ODER (Disjunktion), NICHT (Negation) sowie der mengentheoretischen Verknüpfungen Durchschnitt, Vereinigung, Komplement

→ FORSA - Formale Sprachen und Automaten



$x \wedge y$

$x \vee y$

$\neg x$

Venn-Diagramme für
Konjunktion, Disjunktion
und Negation

[https://de.wikipedia.org/wiki/
Boolesche_Algebra](https://de.wikipedia.org/wiki/Boolesche_Algebra)

Konjunktion			Disjunktion			Negation	
\wedge	0	1	\vee	0	1		\neg
0	0	0	0	0	1	0	1
1	0	1	1	1	1	1	0

Programmierbare Maschinen

- Ada Lovelace (1815-1852) schreibt 1842: *“Die Grenzen der Arithmetik wurden in dem Augenblick überschritten, in dem die Idee zur Verwendung der [Programmier]Karten entstand ... die Analytical Engine hat keine Gemeinsamkeit mit schlichten Rechenmaschinen.”*



Diagram for the computation by the Engine of the Numbers of Bernoulli. See Note G. (page 722 of sup.)

Number of Operations.	Form of Operation.	Variables entered.	Variables resulting.	Indication of change in the value of any Variable.	Remainder of Bernoulli.	Step.	Working Variable.	Result Variable.
1	$x_1 = x_1 + 1$	x_1	x_1	$+1$	x_1	1	x_1	
2	$y_1 = x_1^2$	x_1	y_1		y_1	2	y_1	
3	$z_1 = y_1 + x_1$	y_1, x_1	z_1		z_1	3	z_1	
4	$w_1 = z_1^2$	z_1	w_1		w_1	4	w_1	
5	$v_1 = w_1 + z_1$	w_1, z_1	v_1		v_1	5	v_1	
6	$u_1 = v_1^2$	v_1	u_1		u_1	6	u_1	
7	$t_1 = u_1 + v_1$	u_1, v_1	t_1		t_1	7	t_1	
8	$s_1 = t_1^2$	t_1	s_1		s_1	8	s_1	
9	$r_1 = s_1 + t_1$	s_1, t_1	r_1		r_1	9	r_1	
10	$q_1 = r_1^2$	r_1	q_1		q_1	10	q_1	
11	$p_1 = q_1 + r_1$	q_1, r_1	p_1		p_1	11	p_1	
12	$o_1 = p_1^2$	p_1	o_1		o_1	12	o_1	
13	$n_1 = o_1 + p_1$	o_1, p_1	n_1		n_1	13	n_1	
14	$m_1 = n_1^2$	n_1	m_1		m_1	14	m_1	
15	$l_1 = m_1 + n_1$	m_1, n_1	l_1		l_1	15	l_1	
16	$k_1 = l_1^2$	l_1	k_1		k_1	16	k_1	
17	$j_1 = k_1 + l_1$	k_1, l_1	j_1		j_1	17	j_1	
18	$i_1 = j_1^2$	j_1	i_1		i_1	18	i_1	
19	$h_1 = i_1 + j_1$	i_1, j_1	h_1		h_1	19	h_1	
20	$g_1 = h_1^2$	h_1	g_1		g_1	20	g_1	
21	$f_1 = g_1 + h_1$	g_1, h_1	f_1		f_1	21	f_1	
22	$e_1 = f_1^2$	f_1	e_1		e_1	22	e_1	
23	$d_1 = e_1 + f_1$	e_1, f_1	d_1		d_1	23	d_1	
24	$c_1 = d_1^2$	d_1	c_1		c_1	24	c_1	
25	$b_1 = c_1 + d_1$	c_1, d_1	b_1		b_1	25	b_1	
26	$a_1 = b_1^2$	b_1	a_1		a_1	26	a_1	
27	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	27	a_1	
28	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	28	a_1	
29	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	29	a_1	
30	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	30	a_1	
31	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	31	a_1	
32	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	32	a_1	
33	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	33	a_1	
34	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	34	a_1	
35	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	35	a_1	
36	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	36	a_1	
37	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	37	a_1	
38	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	38	a_1	
39	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	39	a_1	
40	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	40	a_1	
41	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	41	a_1	
42	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	42	a_1	
43	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	43	a_1	
44	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	44	a_1	
45	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	45	a_1	
46	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	46	a_1	
47	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	47	a_1	
48	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	48	a_1	
49	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	49	a_1	
50	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	50	a_1	
51	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	51	a_1	
52	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	52	a_1	
53	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	53	a_1	
54	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	54	a_1	
55	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	55	a_1	
56	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	56	a_1	
57	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	57	a_1	
58	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	58	a_1	
59	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	59	a_1	
60	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	60	a_1	
61	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	61	a_1	
62	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	62	a_1	
63	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	63	a_1	
64	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	64	a_1	
65	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	65	a_1	
66	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	66	a_1	
67	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	67	a_1	
68	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	68	a_1	
69	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	69	a_1	
70	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	70	a_1	
71	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	71	a_1	
72	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	72	a_1	
73	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	73	a_1	
74	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	74	a_1	
75	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	75	a_1	
76	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	76	a_1	
77	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	77	a_1	
78	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	78	a_1	
79	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	79	a_1	
80	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	80	a_1	
81	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	81	a_1	
82	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	82	a_1	
83	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	83	a_1	
84	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	84	a_1	
85	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	85	a_1	
86	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	86	a_1	
87	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	87	a_1	
88	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	88	a_1	
89	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	89	a_1	
90	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	90	a_1	
91	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	91	a_1	
92	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	92	a_1	
93	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	93	a_1	
94	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	94	a_1	
95	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	95	a_1	
96	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	96	a_1	
97	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	97	a_1	
98	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	98	a_1	
99	$a_1 = a_1 + 1$	a_1	a_1	$+1$	a_1	99	a_1	
100	$a_1 = a_1 - 1$	a_1	a_1	-1	a_1	100	a_1	

Programmierbare Maschinen

- Ada Lovelace (1815-1852) schreibt 1842: *“Die Grenzen der Arithmetik wurden in dem Augenblick überschritten, in dem die Idee zur Verwendung der [Programmier]Karten entstand ... die Analytical Engine hat keine Gemeinsamkeit mit schlichten Rechenmaschinen.”*



Diagram for the computation by the Engine of the Numbers of Bernoulli. See Note G. (page 722 of sigs.)

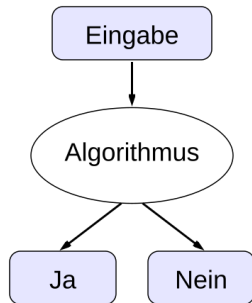
Number of Operations.	Sign of Operation.	Variables used.	Variables receiving results.	Indication of change in the value of any Variable.	Statement of Result.	Step.	Working Variables.	Result Variables.
1	x	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_1 = 2x$	1	x_1	
2	-	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_2 = x_1 - 1$	2	x_2	
3	x	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_3 = x_2 + 1$	3	x_3	
4	-	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_4 = x_3 - 1$	4	x_4	
5	x	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_5 = x_4 + 1$	5	x_5	
6	-	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_6 = x_5 - 1$	6	x_6	
7	x	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_1 = x_1 + x_2$	7	x_1	
8	-	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_2 = x_2 - x_1$	8	x_2	
9	x	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_3 = x_3 + x_2$	9	x_3	
10	-	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_4 = x_4 - x_3$	10	x_4	
11	x	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_5 = x_5 + x_4$	11	x_5	
12	-	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_6 = x_6 - x_5$	12	x_6	
13	x	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_1 = x_1 + x_3$	13	x_1	
14	-	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_2 = x_2 - x_4$	14	x_2	
15	x	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_3 = x_3 + x_5$	15	x_3	
16	-	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_4 = x_4 - x_6$	16	x_4	
17	x	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_5 = x_5 + x_4$	17	x_5	
18	-	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_6 = x_6 - x_5$	18	x_6	
19	x	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_1 = x_1 + x_6$	19	x_1	
20	-	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_2 = x_2 - x_1$	20	x_2	
21	x	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_3 = x_3 + x_2$	21	x_3	
22	-	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_4 = x_4 - x_3$	22	x_4	
23	x	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_5 = x_5 + x_4$	23	x_5	
24	-	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_6 = x_6 - x_5$	24	x_6	
25	x	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_1 = x_1 + x_6$	25	x_1	
26	-	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_2 = x_2 - x_1$	26	x_2	
27	x	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_3 = x_3 + x_2$	27	x_3	
28	-	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_4 = x_4 - x_3$	28	x_4	
29	x	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_5 = x_5 + x_4$	29	x_5	
30	-	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_6 = x_6 - x_5$	30	x_6	
31	x	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_1 = x_1 + x_6$	31	x_1	
32	-	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_2 = x_2 - x_1$	32	x_2	
33	x	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_3 = x_3 + x_2$	33	x_3	
34	-	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_4 = x_4 - x_3$	34	x_4	
35	x	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_5 = x_5 + x_4$	35	x_5	
36	-	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_6 = x_6 - x_5$	36	x_6	
37	x	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_1 = x_1 + x_6$	37	x_1	
38	-	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_2 = x_2 - x_1$	38	x_2	
39	x	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_3 = x_3 + x_2$	39	x_3	
40	-	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_4 = x_4 - x_3$	40	x_4	
41	x	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_5 = x_5 + x_4$	41	x_5	
42	-	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_6 = x_6 - x_5$	42	x_6	
43	x	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_1 = x_1 + x_6$	43	x_1	
44	-	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_2 = x_2 - x_1$	44	x_2	
45	x	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_3 = x_3 + x_2$	45	x_3	
46	-	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_4 = x_4 - x_3$	46	x_4	
47	x	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_5 = x_5 + x_4$	47	x_5	
48	-	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_6 = x_6 - x_5$	48	x_6	
49	x	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_1 = x_1 + x_6$	49	x_1	
50	-	$x_1, x_2, x_3, x_4, x_5, x_6$	$x_1, x_2, x_3, x_4, x_5, x_6$		$x_2 = x_2 - x_1$	50	x_2	

Note: When a negative sign appears in the Statement of Result, it is to be understood that the variable is to be subtracted from the value it has at the end of the preceding operation.

“Die Analytical Engine hat keine Ambitionen etwas (Neues) hervorzubringen. Sie kann [nur] das tun, was wir ihr befehlen; sie hat jedoch keine Fähigkeit zur Erkenntnis analytischer Verhältnisse oder Wahrheiten.”
(Lady Lovelace's objection, Turing, 1950)

Das “Entscheidungsproblem”

- Anfang 20. Jh.: Kann es einen Algorithmus geben, der für eine Aussage (Eingabe) entscheidet, ob sie universell wahr oder falsch ist (Hilbert & Ackermann, 1928)
- Church (1935) und Turing (1936) haben bewiesen, dass es so einen Algorithmus nicht geben kann



<https://de.wikipedia.org/wiki/Entscheidbarkeit>

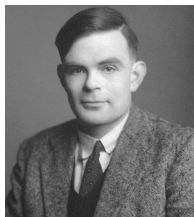
Konzept der Berechenbarkeit

Alan Turing (1912-1954)

ON COMPUTABLE NUMBERS, WITH AN APPLICATION TO
THE ENTSCHEIDUNGSPROBLEM

By A. M. TURING.

[Received 28 May, 1936.—Read 12 November, 1936.]



- Wenn man schon nicht alles **beweisen** kann, was kann man dann **berechnen**? (Und was nicht?)
- Was heißt das überhaupt, Dinge zu berechnen?

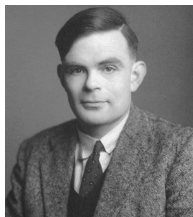
Konzept der Berechenbarkeit

Alan Turing (1912-1954)

ON COMPUTABLE NUMBERS, WITH AN APPLICATION TO
THE ENTSCHEIDUNGSPROBLEM

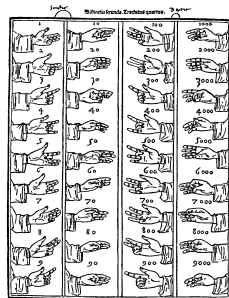
By A. M. TURING.

[Received 28 May, 1936.—Read 12 November, 1936.]



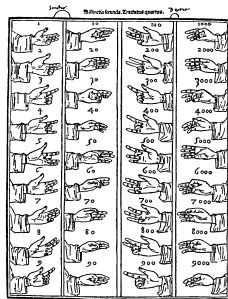
- Wenn man schon nicht alles **beweisen** kann, was kann man dann **berechnen**? (Und was nicht?)
 - Was heißt das überhaupt, Dinge zu berechnen?
- versucht zu **formalisieren** wie ein Mensch rechnet
- **Turingmaschine** - Idee war nicht, Modell für Computer aufzustellen

Was brauchen Menschen, um etwas zu berechnen?



Was brauchen Menschen, um etwas zu berechnen?

- Liste von Anweisungen (Algorithmus/Programm)
- Lesen, Schreiben (radieren/überschreiben)
- Schmierpapier für Zwischenergebnisse
- Gedächtnis
- “Eingabe” & “Ausgabe”
- Speichermedium



Turing's Regeln

1. Alphabet: endliche Menge von Zeichen

$$\Sigma = \{0, 1\}$$

Turing's Regeln

1. Alphabet: endliche Menge von Zeichen
 $\Sigma = \{0, 1\}$
2. Speicherband, das in Felder unterteilt ist, 1
Zeichen pro Feld, beliebig lang - "Tickertape"

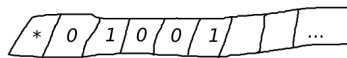


Turing's Regeln

1. Alphabet: endliche Menge von Zeichen

$$\Sigma = \{0, 1\}$$

2. Speicherband, das in Felder unterteilt ist, 1 Zeichen pro Feld, beliebig lang - "Tickertape"



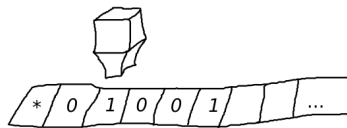
3. Leerzeichen \square oder "B" (blank)
4. Sonderzeichen '*' für den Anfang des Tapes (* und $\square \notin \Sigma$)

Turing's Regeln

1. Alphabet: endliche Menge von Zeichen

$$\Sigma = \{0, 1\}$$

2. Speicherband, das in Felder unterteilt ist, 1 Zeichen pro Feld, beliebig lang - "Tickertape"



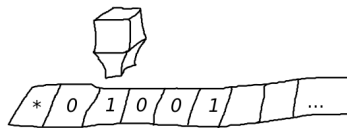
3. Leerzeichen \square oder "B" (blank)
4. Sonderzeichen '*' für den Anfang des Tapes (* und $\square \notin \Sigma$)
5. Schreib-/Lesekopf, der jeweils ein Feld lesen oder überschreiben kann (darf * nicht überschreiben und nicht weiter nach links gehen)

Turing's Regeln

1. Alphabet: endliche Menge von Zeichen

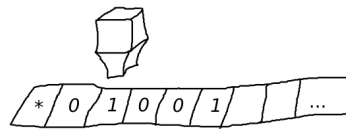
$$\Sigma = \{0, 1\}$$

2. Speicherband, das in Felder unterteilt ist, 1 Zeichen pro Feld, beliebig lang - "Tickertape"



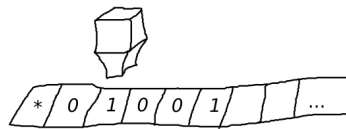
3. Leerzeichen \square oder "B" (blank)
4. Sonderzeichen '*' für den Anfang des Tapes (* und $\square \notin \Sigma$)
5. Schreib-/Lesekopf, der jeweils ein Feld lesen oder überschreiben kann (darf * nicht überschreiben und nicht weiter nach links gehen)
6. "Programm", das den Kopf steuert - Berechnungsvorschrift

Wie soll so ein “Programm” aussehen?



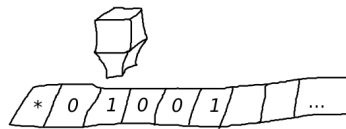
Wie soll so ein “Programm” aussehen?

read	write	move	go to
1	1	R	0



Wie soll so ein “Programm” aussehen?

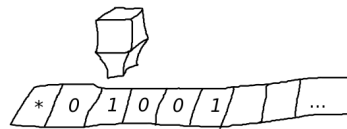
read	write	move	go to
1	1	R	0



1. read: Lies aktuelles Feld

Wie soll so ein “Programm” aussehen?

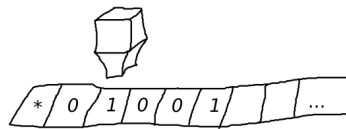
read	write	move	go to
1	1	R	0



1. read: Lies aktuelles Feld
2. write: Schreibe etwas auf das Feld

Wie soll so ein “Programm” aussehen?

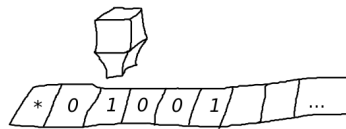
read	write	move	go to
1	1	R	0



1. read: Lies aktuelles Feld
2. write: Schreibe etwas auf das Feld
3. move: Gehe 1 Schritt nach links, rechts oder halte an

Wie soll so ein “Programm” aussehen?

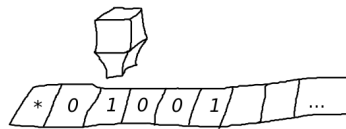
read	write	move	go to
1	1	R	0



1. read: Lies aktuelles Feld
2. write: Schreibe etwas auf das Feld
3. move: Gehe 1 Schritt nach links, rechts oder halte an
4. go to: wähle den nächsten Programmschritt aus

Wie soll so ein “Programm” aussehen?

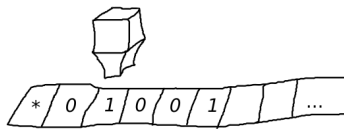
read	write	move	go to
1	1	R	0



1. read: Lies aktuelles Feld
2. write: Schreibe etwas auf das Feld
3. move: Gehe 1 Schritt nach links, rechts oder halte an
4. go to: **wähle** den nächsten Programmschritt aus
(2, 3 und 4 hängen von 1 ab)

Wie soll so ein “Programm” aussehen?

read	write	move	go to
1	1	R	0
0	1	R	2
*	*	R	0
B	0	L	-1

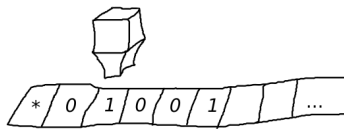


1. read: Lies aktuelles Feld
2. write: Schreibe etwas auf das Feld
3. move: Gehe 1 Schritt nach links, rechts oder halte an
4. go to: **wähle** den nächsten Programmschritt aus
(2, 3 und 4 hängen von 1 ab)

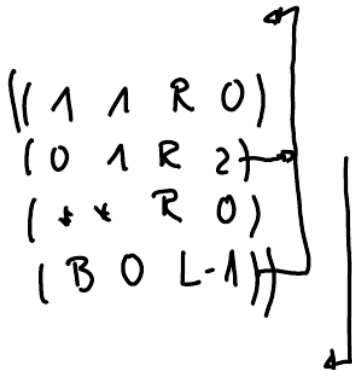
Wie soll so ein "Programm" aussehen?

read	write	move	go to
1	1	R	0
0	1	R	2
*	*	R	0
B	0	L	-1

= Zustand



1. read: Lies aktuelles Feld
2. write: Schreibe etwas auf das Feld
3. move: Gehe 1 Schritt nach links, rechts oder halte an
4. go to: **wähle** den nächsten Programmschritt aus
(2, 3 und 4 hängen von 1 ab)



Turing-Maschine: Subtraktion von 1 von Binärzahl

$$10-1=9$$

$$\begin{array}{r} 1010 \\ - \quad 1 \\ \hline 1001 \end{array}$$

$$17-1=16$$

$$\begin{array}{r} 10001 \\ - \quad 1 \\ \hline 10000 \end{array}$$

Turing-Maschine: Subtraktion von 1 von Binärzahl

- wenn Input 0, dann 0

$$10-1=9$$

$$\begin{array}{r} 1010 \\ - \quad 1 \\ \hline 1001 \end{array}$$

$$17-1=16$$

$$\begin{array}{r} 10001 \\ - \quad 1 \\ \hline 10000 \end{array}$$

Turing-Maschine: Addition von 1 zu einer Binärzahl

Überlegen Sie sich eine Turing-Maschine zur Addition von 1 zu einer Binärzahl für die nachfolgenden Beispielrechnungen!

$$4+1=5$$

$$\begin{array}{r} 100 \\ + 1 \\ \hline 101 \end{array}$$

$$5+1=6$$

$$\begin{array}{r} 101 \\ + 1 \\ \hline 110 \end{array}$$

Turing-Maschine: Addition von 1 zu einer Binärzahl

Überlegen Sie sich eine Turing-Maschine zur Addition von 1 zu einer Binärzahl für die nachfolgenden Beispielrechnungen!

$$4+1=5$$

$$\begin{array}{r} 100 \\ + 1 \\ \hline 101 \end{array}$$

$$5+1=6$$

$$\begin{array}{r} 101 \\ + 1 \\ \hline 110 \end{array}$$

Wenden Sie Ihre Turing-Maschine nun auf folgende Zahl an! Ändern Sie Ihre Turing-Maschine, so dass sie auch für diese Eingabe die korrekte Ausgabe liefert!

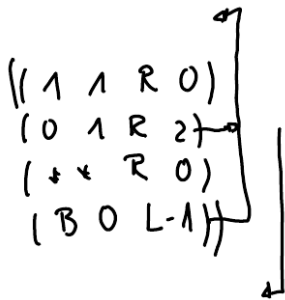
$$7+1=8$$

$$\begin{array}{r} 111 \\ + 1 \\ \hline 1000 \end{array}$$

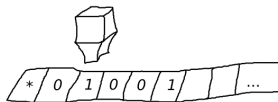
Turing-“Maschine”

read	write	move	go to
1	1	R	0
0	1	R	2
*	*	R	0
B	0	L	-1

= Zustand



- theoretisches Modell einer Rechenmaschine
- Modell der theoretischen Informatik, das den Begriff der Berechenbarkeit formalisiert
- Berechnung besteht aus schrittweisen Manipulationen von Zeichen/Symbolen, Zeichen können u.a. als Zahlen interpretiert werden

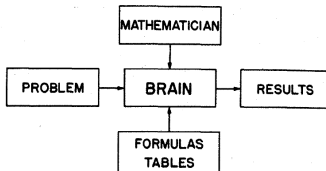
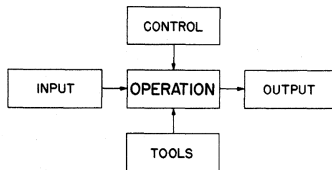


1940/50er Jahre

Grace Hopper (1906-1992)

- 1952 The education of a computer

While the materialization is new, the idea of **mechanizing mathematical thinking** is not new. Its lineage starts with the abacus and descends through Pascal, Leibnitz, and Babbage. More

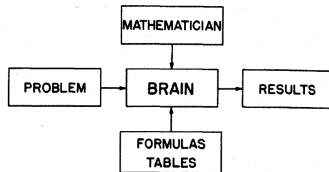
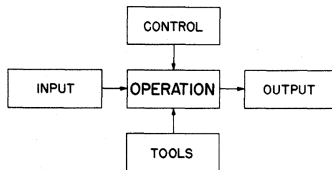


1940/50er Jahre

Grace Hopper (1906-1992)

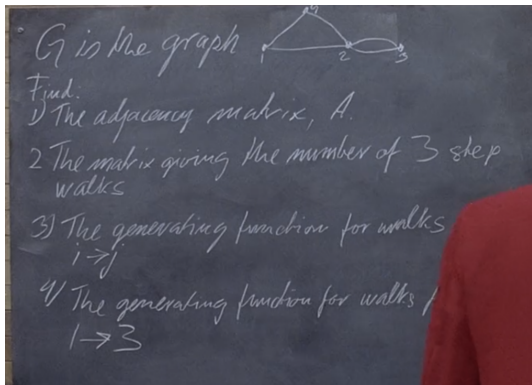
- 1952 The education of a computer

While the materialization is new, the idea of **mechanizing mathematical thinking** is not new. Its lineage starts with the abacus and descends through Pascal, Leibnitz, and Babbage. More



"I was lazy and hoped that the programmer may return to being a mathematician"

Algorithmen & Objektivität



- so beschrieben, dass bei gleicher Eingabe (Frage) immer dasselbe Ergebnis herauskommt
 - jede/r kann sie ausführen
 - unabhängig von Nutzerin
- Objektivität

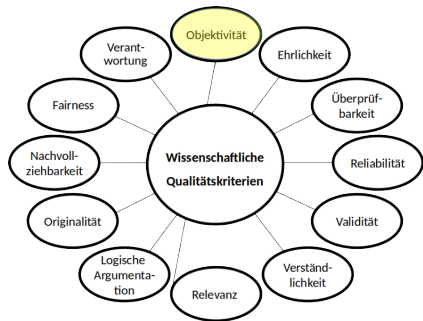
aus Good Will Hunting (Drama 1997), Quelle <https://bthmas.wordpress.com/2016/03/26/good-will-hunting-problem/>

Objektivität

1. Was bedeutet Objektivität?
2. Was steht Objektivität im Wege?
3. Welches Problem wird in dem Affenexperiment beschrieben?



Objektivität



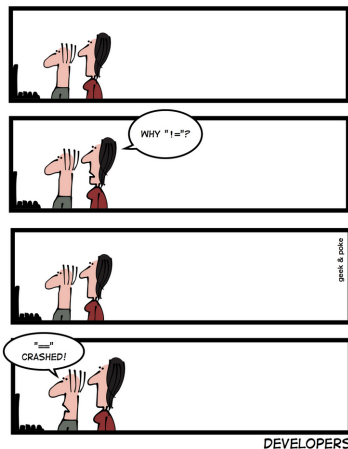
©PROF. BALZERT-STIFTUNG 2022

1. Ergebnisse sind unabhängig vom Beobachter, d.h. frei von persönlichen Urteilen, Einstellungen, nicht frei von fachlichem Vorwissen
2. menschliche Wahrnehmungs- und Urteilstendenzen z.B. Cognitive Biases, da sie die Unvoreingenommenheit beeinträchtigen können
3. Affenexperiment: Bestätigungsfehler (confirmation bias, selektive Wahrnehmung, Wunschdenken)

<https://www.visualcapitalist.com/50-cognitive-biases-in-the-modern-world/>

Intelligente Maschinen - Wo stehen wir?

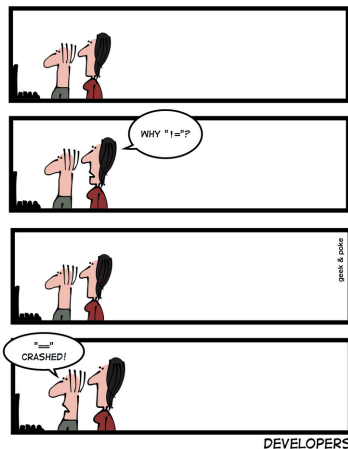
- Denken zu definieren ist subjektiv oder indirekt - von außen beobachtbar machen, messbar machen - wie gut ist die Messung?



(Quelle: <http://geek-and-poke.com/>)

Intelligente Maschinen - Wo stehen wir?

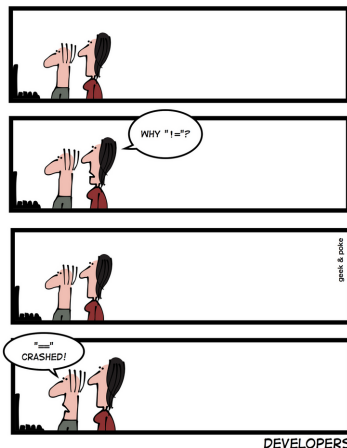
- Denken zu definieren ist subjektiv oder indirekt - von außen beobachtbar machen, messbar machen - wie gut ist die Messung?
- Intelligenz haben wir noch nicht definiert



(Quelle: <http://geek-and-poke.com/>)

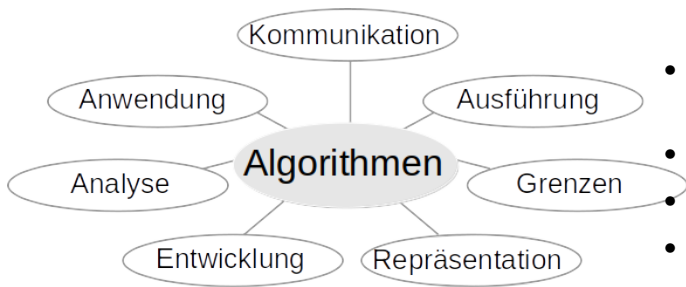
Intelligente Maschinen - Wo stehen wir?

- Denken zu definieren ist subjektiv oder indirekt - von außen beobachtbar machen, messbar machen - wie gut ist die Messung?
- Intelligenz haben wir noch nicht definiert
- klassische Maschinen und Algorithmen sind bereichsspezifisch



(Quelle: <http://geek-and-poke.com/>)

Algorithmen als Forschungsgegenstand



- Welche Probleme lassen sich algorithmisch lösen?
- Wie kann man Algorithmenentwicklung vereinfachen?
- Wie repräsentiert man Algorithmen?
- Wie kann man Algorithmen analysieren?
- Können Algorithmen intelligentes Verhalten erzeugen?
- Wie steigert man ihre Effizienz?