Wie erfinde ich mathematische Algorithmen???

Wolfgang Windsteiger

RISC Linz

Universität Linz / Schloss Hagenberg

Wie beweise ich mathematische Algorithmen???

Ausgangslage

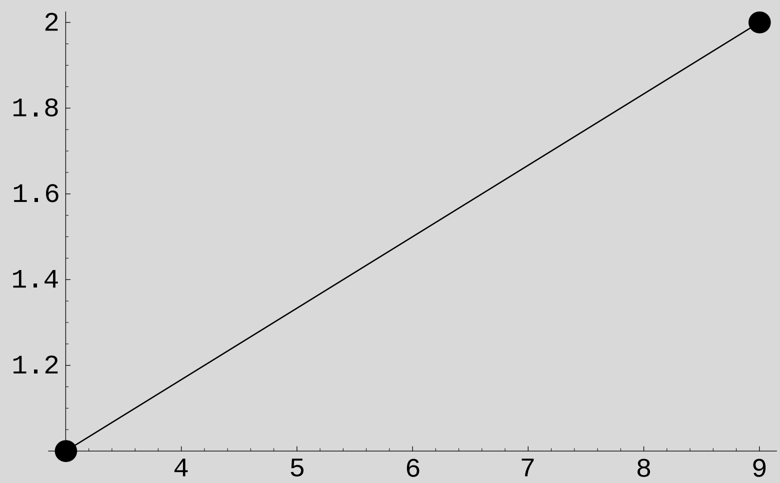
- Problemstellung bekannt, gesucht ist eine *Lösung des Problems*.
  — Beispiel: Gegeben $P = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$ und $Q = \begin{pmatrix} 9 \\ 2 \end{pmatrix}$. Gesucht eine Gerade g durch P und Q.

Lösung

— Beispiel: Gegeben $P = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$ und $Q = \begin{pmatrix} 9 \\ 2 \end{pmatrix}$. Gesucht eine Gerade g durch P und Q.

Lösung 1

```
Show[
  Graphics[{PointSize[0.03], Line[{{3, 1}, {9, 2}}], Point[{3, 1}], Point[{9, 2}]}], Axes → True]
```



- Graphics -

Lösung 2

$$g : \begin{pmatrix} 3 \\ 1 \end{pmatrix} + t \cdot \begin{pmatrix} 6 \\ 1 \end{pmatrix}$$

Lösung 3

$$g[x] := \frac{x}{6} + \frac{1}{2}$$

?

- Welche Lösung ist *die richtige Lösung???*

4 of 15

Wir sehen schon …

- Wir müssen bei jedem Problem *genau beschreiben*, was wir als *Lösung des Problems* erwarten/akzeptieren.
- Welche Bestandteile muss so eine Beschreibung beinhalten?

5 of 15

Gegeben … Gesucht …

- Eingaben
- Ausgaben

- Eigenschaft

Beispiel

- Eingaben: $P = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$, $Q = \begin{pmatrix} 9 \\ 2 \end{pmatrix}$
- Ausgaben: g
- Eigenschaft: "g ist eine Gerade zwischen P und Q" (???)

" g ist eine Gerade zwischen $\begin{pmatrix} x_1 \\ y_1 \end{pmatrix}$ und $\begin{pmatrix} x_2 \\ y_2 \end{pmatrix}$ "

- *Eine* Möglichkeit:

g ist ein *lineares Polynom*, dessen *Wert an der Stelle* x_1 gleich y_1 ist und dessen *Wert an der Stelle* x_2 gleich y_2 ist.

" g ist lineares Polynom "

- *Eine* Möglichkeit:

$g = \langle p_1, p_2 \rangle \quad \longrightarrow \quad p_1 + p_2\, x$

" Wert von $g = \langle p_1, p_2 \rangle$ an der Stelle a "

- *Eine* Möglichkeit:

$p_1 + p_2\, a$

```
Wert[〈p1_, p2_〉, a_] := p1 + p2 * a
```

Lösung

z.B. durch Probieren

$g = \left\langle \frac{1}{2}, \frac{1}{6} \right\rangle$

```
g = 〈1/2, 1/6〉;
```

weil:

- Wert von g an der Stelle 3:

```
Wert[g, 3]
```

```
1
```

- Wert von g an der Stelle 3:

```
Wert[g, 9]
```

2

6 of 15

Lösung eines *konkreten Problems*

- Konkrete Eingaben sind *gegeben*
- Gewünschte Eigenschaft des Resultats *gegeben*
- Konkrete Lösung *gesucht*, die die gewünschte Eigenschaft aufweist

ABER …

- Eingaben: $P = \begin{pmatrix} 7 \\ 13 \end{pmatrix}, Q = \begin{pmatrix} -3 \\ 2 \end{pmatrix}$
- Ausgaben: g
- Eigenschaft: "g ist eine Gerade zwischen P und Q"

7 of 15

ALGORITHMEN

- Was man *wirklich will*, sind Verfahren/Methoden, die eine gewisse Problemstellung *für alle möglichen Eingabewerte* lösen ⟶ *Algorithmen*
- Angabe eines Algorithmus, der *für alle konkreten Eingabewerte* eine *konkrete Lösung* ermittelt.

8 of 15

Lösung eines *Problems*

- Eingaben: x
- Eigenschaft: Ψ_x (Eigenschaft der Eingaben: *Eingabebedingung*)
- Ausgabe: y
- Eigenschaft: $\Phi_{x,y}$ (Beziehung zwischen Eingabe und Ausgabe: *Ausgabebedingung*)

Wir suchen einen Algorithmus α, sodass $\alpha[x]$ eine Lösung des Problems ist, aber *für alle* x, d.h.

$$\underset{x}{\forall}\ \Psi_x \Rightarrow \Phi_{x,\alpha[x]}$$

Eine Problemstellung P mit Eingaben x und Ausgaben y schreibe ich als $P_{x,y}$ und meine damit:

Wenn x die Eingabebedingung erfüllt, dann erfüllt y die Ausgabebedingung, d.h.

$\Psi_x \Rightarrow \Phi_{x,y}$

9 of 15

Beispiel

- Eingaben: x_1, y_1, x_2, y_2
- Eigenschaft: $x_1, y_1, x_2, y_2 \in \mathbb{R}$
- Ausgabe: p_1, p_2
- Eigenschaft: $\text{Wert}[\langle p_1, p_2\rangle, x_1] = y_1 \wedge \text{Wert}[\langle p_1, p_2\rangle, x_2] = y_2$

Wir suchen einen Algorithmus “Gerade”, sodass Gerade[x_1, y_1, x_2, y_2] eine Lösung des Problems ist, d.h.

$\underset{x}{\forall}\ x_1, y_1, x_2, y_2 \in \mathbb{R} \Rightarrow \text{Wert}[\text{Gerade}[x_1, y_1, x_2, y_2], x_1] = y_1 \wedge \text{Wert}[\text{Gerade}[x_1, y_1, x_2, y_2], x_2] = y_2$

Obige Problemstellung P schreibe ich als $P_{\langle x_1, y_1, x_2, y_2\rangle, \langle p_1, p_2\rangle}$ und meine damit:

$x_1, y_1, x_2, y_2 \in \mathbb{R} \Rightarrow \text{Wert}[\langle p_1, p_2\rangle, x_1] = y_1 \wedge \text{Wert}[\langle p_1, p_2\rangle, x_2] = y_2$

10 of 15

Wie löse ich ein Problem $P_{x,y}$???

- Nachschauen
- Problemlöse-Muster: “Zurückführen auf ein anderes Problem”, “Ansatz”, etc.

Rekursion

- Es gibt verschiedene Rekursions-Schemata:

$P_{S[x],y} \Rightarrow P_{x,F[y]}$ $\quad \alpha[x] := F[\alpha[S[x]]]$

$P_{S[x],y} \Rightarrow P_{x,F[x,y]}$ $\quad \alpha[x] := F[x, \alpha[S[x]]]$ $\quad$ ($\longrightarrow$ Potenzieren)

$P_{x,y} \Rightarrow P_{C[x],F[y]}$ $\quad \alpha[C[x]] := F[\alpha[x]]$

$P_{x_1,y_1} \wedge P_{x_2,y_2} \Rightarrow P_{C[x_1,x_2],F[y_1,y_2]}$ $\quad \alpha[C[x_1, x_2]] := F[\alpha[x_1], \alpha[x_2]]$ $\quad$ ($\longrightarrow$ Differenzieren)

Potenzieren

```
Pot[a_, 0] := 1
```

```
Pot[a_, n_] := a * Pot[a, n - 1]
```

```
Pot[2, 16]
```

```
65536
```

Neue Problemstellung: Evaluieren von Polynomen

- Eingaben: $\langle p_1, p_2, p_3, \ldots, p_n \rangle$, a
- Eigenschaft: p_i, $a \in \mathbb{R}$
- Ausgabe: W
- Eigenschaft: $\text{Wert}[\langle p_1, p_2, p_3, \ldots, p_n \rangle, a] = W$ wobei: $\text{Wert}[p, a] := \sum_{i=1}^{n} p_i \, a^{i-1}$
- Suche Algorithmus “Eval”, sodass: $\text{Wert}[\langle p_1, p_2, p_3, \ldots, p_n \rangle, a] = \text{Eval}[\langle p_1, p_2, p_3, \ldots, p_n \rangle, a]$

$$\text{Wert}[\text{p\_}, \text{a\_}] := \sum_{i=1}^{\text{Length}[p]} p[[i]] * a^{i-1}$$

$$\text{Wert}[\langle 3435, 34, -245, 4 \rangle, 3]$$

1440

- Wir wollen einen “anderen (besseren) Algorithmus”!

Wie finde ich einen Algorithmus ???

- Eingaben: $\langle p_1, p_2, p_3, \ldots, p_n \rangle$, a
- Problemstellung $P_{\langle\langle p_1, p_2, p_3, \ldots, p_n \rangle, a \rangle, W}$: $\text{Wert}[\langle p_1, p_2, p_3, \ldots, p_n \rangle, a] = W$
- Probiere verschiedene Schemata
- Verwende mathematisches Wissen zum Beweisen der Schemata

$P_{\langle p, a \rangle, W} \Rightarrow P_{\langle C[p, p_0], a \rangle, F[p_0, a, W]}$ $\quad \alpha[C[p, p_0], a] := F[p_0, a, \alpha[p, a]]$

- Mathematisches Wissen in unserem Fall:

$\text{Wert}[p + q, a] = \text{Wert}[p, a] + \text{Wert}[q, a]$

$\text{Wert}[\langle p_0 \rangle, a] = p_0$

$\text{Wert}[\text{Shift}[p], a] := a * \text{Wert}[p, a]$

$\text{Shift}[\langle p_1, p_2, \ldots, p_n \rangle] + \langle p_0 \rangle = \langle p_0, p_1, p_2, \ldots, p_n \rangle$

$\text{Shift}[\langle p_1, p_2, \ldots, p_n \rangle] := \langle 0, p_1, p_2, \ldots, p_n \rangle$

Nehme an:

$P_{\langle p, a \rangle, W}$

$\text{Wert}[p, a] = W$

Zu beweisen:

$P_{\langle C[p,p_0],a\rangle,F[p_0,a,W]}$

$\text{Wert}[C[p, p_0], a] \stackrel{?}{=} F[p_0, a, W]$

Wähle C:

$C[p, p_0] := \text{Shift}[p] + \langle p_0 \rangle$

dann

$\text{Wert}[C[p, p_0], a] =$

$$\text{Wert}[\text{Shift}[p] + \langle p_0\rangle, a] = \underbrace{\text{Wert}[\text{Shift}[p], a]}_{a*\text{Wert}[p,a]} + \underbrace{\text{Wert}[\langle p_0\rangle, a]}_{p_0} = a * \underbrace{\text{Wert}[p, a]}_{W} + p_0 = a * W + p_0 \stackrel{?}{=} F[p_0, a, W]$$

Wähle F:

$F[p_0, a, W] := a * W + p_0$

Lösungsalgorithmus:

$\alpha[C[p, p_0], a] := F[p_0, a, \alpha[p]]$

$\text{Eval}[\text{Shift}[p] + \langle p_0\rangle, a] := a * \text{Eval}[p, a] + p_0$

```
Clear[Eval]
```

```
Eval[〈p0_, p__〉, a_] := a * Eval[〈p〉, a] + p0
```

```
Eval[〈p0_〉, a_] := p0
```

```
Eval[〈3435, 34, −245, 4〉, 3]
```

```
1440
```

12 of 15

13 of 15

Neue Problemstellung

- Eingaben: $\langle x_1, x_2, x_3, \ldots, x_n\rangle$, $\langle a_1, a_2, a_3, \ldots, a_n\rangle$
- Eigenschaft: $x_i, a_i \in \mathbb{R}$
- Ausgabe: $\langle p_1, p_2, p_3, \ldots, p_n\rangle$
- Eigenschaft:

$$\underset{i=1,\ldots,n}{\forall} \text{Wert}[\langle p_1, p_2, p_3, \ldots, p_n\rangle, x_i] = a_i$$

- Suche Algorithmus "Interpol", sodass:

$$\underset{i=1,\ldots,n}{\forall}\ \mathrm{Wert}[\mathrm{Interpol}[\langle x_1, x_2, x_3, \ldots, x_n\rangle, \langle a_1, a_2, a_3, \ldots, a_n\rangle], x_i] = a_i$$

14 of 15

Probiere Rekursionsschema

$$P_{S_1[\langle x,a\rangle],y_1} \ \wedge\ P_{S_2[\langle x,a\rangle],y_2} \ \Rightarrow\ P_{\langle x,a\rangle,F[\langle x,a\rangle,y_1,y_2]}$$

Ein rekursiver Lösungsalgorithmus g für P hat dann die Gestalt

$$\mathrm{Interpol}[x, a] := F[x, \mathrm{Interpol}[S_1[x, a]], \mathrm{Interpol}[S_2[x, a]]]$$

mit

$$S_1[\langle x, a\rangle] := \langle x_{1\rightarrow\square}, a_{1\rightarrow\square}\rangle$$
$$S_2[\langle x, a\rangle] := \langle x_{-1\rightarrow\square}, a_{-1\rightarrow\square}\rangle$$
$$F[\langle x, a\rangle, y_1, y_2] := \frac{\langle -x_1, 1\rangle * y_1 - \langle -x_{-1}, 1\rangle * y_2}{x_{-1} - x_1}$$

Wir müssen beweisen:

$$P_{\langle\langle x_1\rangle,\langle a_1\rangle\rangle,\langle a_1\rangle} \quad \text{(elementarer Fall)}$$

$$P_{S_1[\langle x,a\rangle],y_1} \ \wedge\ P_{S_2[\langle x,a\rangle],y_2} \ \Rightarrow\ P_{\langle x,a\rangle,F[\langle x,a\rangle,y_1,y_2]}$$

15 of 15

Der Algorithmus

```
Interpol[⟨x1_⟩, ⟨a1_⟩] := ⟨a1⟩
```

```
Interpol[x_, a_] := (⟨−x⟦1⟧, 1⟩ ∗ Interpol[Drop[x, 1], Drop[a, 1]] –
        ⟨−x⟦−1⟧, 1⟩ ∗ Interpol[Drop[x, −1], Drop[a, −1]]) / (x⟦−1⟧ – x⟦1⟧)
```

Hilfsprogramme: Polynom-Arithmetik

```
coef[p_, i_] /; Abs[i] < Length[p] := p⟦i + 1⟧
```

```
coef[p_, i_] := 0
```

```
Clear[AngleBracket]
```

$$\text{AngleBracket /: x\_AngleBracket} * \text{y\_AngleBracket} :=$$
$$\text{AngleBracket @@ Table}\left[\sum_{j=0}^{i} \text{coef}[x, j] * \text{coef}[y, i - j], \{i, 0, \text{Length}[x] + \text{Length}[y] - 2\}\right]$$

```
AngleBracket /: x_AngleBracket + y_AngleBracket :=
  AngleBracket @@ Table[coef[x, i] + coef[y, i], {i, 0, Max[Length[x], Length[y]] − 1}]
```

```
AngleBracket /: λ_?NumberQ * x_AngleBracket := Map[λ * # &, x]
```

Beispiele

— Gegeben $P = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$ und $Q = \begin{pmatrix} 9 \\ 2 \end{pmatrix}$. Gesucht eine Gerade g durch P und Q.

```
Interpol[⟨3, 9⟩, ⟨1, 2⟩]
```

$\left\langle \frac{1}{2}, \frac{1}{6} \right\rangle$

— Polynom durch gegebene 10 Punkte:

```
x = AngleBracket @@ Table[i, {i, 1, 10}]
```

```
⟨1, 2, 3, 4, 5, 6, 7, 8, 9, 10⟩
```

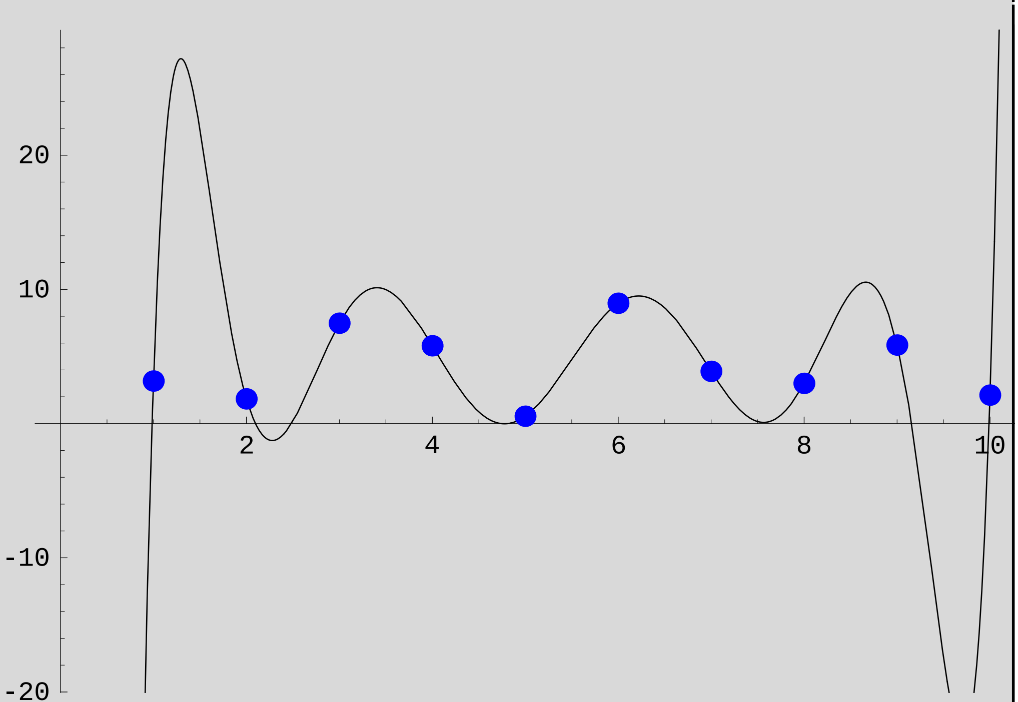
```
a = AngleBracket @@ Table[Random[Real, {0, 10}], {i, 1, 10}]
```

```
⟨3.17214, 1.84198, 7.48278, 5.79924,
  0.549365, 8.96882, 3.89361, 2.99861, 5.85349, 2.12407⟩
```

```
p = Interpol[x, a]
```

```
⟨−1727.38, 4814.52, −5357.58, 3183.16, −1126.57,
  248.798, −34.5779, 2.93699, −0.139172, 0.0028173⟩
```

```
Plot[Eval[p, y], {y, 0, 11}, Epilog →
  {PointSize[0.02], RGBColor[0, 0, 1], Map[Point, Transpose[{List @@ x, List @@ a}]]}]
```



- Graphics -