Iteration I

Iteration: wiederholte Berechnung mit identischer Berechnungsvorschrift

Beispiel: Tabulation einer Funktion

gegeben: Wertevorrat $x = \{-5.0, -4.8, -4.6, \ldots, 4.8, 5.0\}$

gesucht: Funktionswerte $f(x) = x^3 - 20x$

Ausdrücke

	A	B	...
1	-5	=a1**3-20*a1	
2	-4.8	=a2**3-20*a2	
3	-4.6	=a3**3-20*a3	
4	-4.4	=a4**3-20*a4	
...	...	...	
50	4.8	=a50**3-20*a50	
51	5	=a51**3-20*a51	
...			

Werte

	A	B	...
1	-5	-25	
2	-4.8	-14.592	
3	-4.6	-5.336	
4	-4.4	2.816	
...	...	...	
50	4.8	14.592	
51	5	25	
...			

Diagramm

Darstellungsbereich: a1:b51

$\rightarrow$ Diagramm $\rightarrow$ X/Y-Diagramm

Iteration I

Iteration durch *Kopieren* der Berechnungsvorschrift

Relative Referenz: Inkrementieren der Referenzzeiger beim Kopieren

Ausdrücke

	A	B	C	...
1	3	2	=a1+b1	
2	2	4	=a2+b2	
3	1	3	=a3+b3	
...				

Absolute Referenz: Referenzzeiger bleiben beim Kopieren unverändert

zeilenkonstante Referenz: ⟨Spalte⟩$⟨Zeile⟩

Ausdrücke

	A	B	C	...
1	3	2	=a1+b$1	
2	2	4	=a2+b$1	
3	1	3	=a3+b$1	
...				

Iteration I

Absolute Referenz:

spaltenkonstante Referenz: $\langle$Spalte$\rangle\langle$Zeile$\rangle$ mit vorangestelltem $: \$\langle\text{Spalte}\rangle\langle\text{Zeile}\rangle$

Ausdrücke

	A	B	C	...
1	3	2	3	
2	2	4	2	
3	=a1+$a2	=b1+$a2	=c1+$a2	
...				

richtungsunabhängig absolute Referenz: $\$\langle\text{Spalte}\rangle\$\langle\text{Zeile}\rangle$

Ausdrücke

	A	B	C	...
1	3			
2	2	=a2+a1		
3	4		=b3+a1	
...				

Wozu absolute Referenz? lokale Manipulierbarkeit bei globaler Modifikation der Berechnungsvorschrift

Iteration I

z.B. Konstanten in einem funktionalen Ausdruck: $f(x) = a_3 x^3 + a_2 x^2 + a_1 x + a_0$

Ausdrücke

	A	B	C	D
1			x_i	$f(x_i)$
2	a_0	0	-5	=b5*c2**3+b4*c2**2+b3*c2+b2
3	a_1	-20	-4.8	=b5*c3**3+b4*c3**2+b3*c3+b2
4	a_2	0	-4.6	=b5*c4**3+b4*c4**2+b3*c4+b2
5	a_3	1	-4.4	=b5*c5**3+b4*c5**2+b3*c5+b2
...			...	...
51			4.8	=b5*c51**3+b4*c51**2+b3*c51+b2
52			5	=b5*c52**3+b4*c52**2+b3*c52+b2
...				

Werte

	A	B	C	D
			x_i	$f(x_i)$
	a_0	0	-5	-25
	a_1	-20	-4.8	-14.592
	a_2	0	-4.6	-5.336
	a_3	1	-4.4	2.816
			...	...
			4.8	14.592
			5	25

Simulation: Was passiert, wenn ...? Experimente: gezielte Variation der freien Parametern

Induktion

Induktion:

Ableitung des Allgemeinen aus dem Besonderen

Schlußfolgern vom Spezialfall auf den allgemeinen Fall

Beweisverfahren auf der Basis der Nachfolgerrelation
(Übergang von n zu n+1)

Induktionsanfang: faktuelle Information für einen ausgezeichneten Anfangswert, z.B. $x_0 = 0$

Induktionsschritt: Übergang auf das nachfolgende Element, z.B. $x_n = x_{n-1} + 1$

Induktion

Beispiel: kumulative Addition zur Berechnung eines Wertevorrats (Induktionsvorschrift)

$$x_n = \begin{cases} -5 & \text{für } n = 1 \\ x_{n-1} + 0.2 & \text{sonst} \end{cases} \qquad \begin{matrix} \text{Induktionsanfang} \\ \text{Induktionsschritt} \end{matrix}$$

Ausdrücke

	A	B	...
1	n	x	
2	1	-5	
3	=a2+1	=b2+0.2	
4	=a3+1	=b3+0.2	
5	=a4+1	=b4+0.2	
...	...	...	
51	=a50+1	=b50+0.2	
52	=a51+1	=b51+0.2	
...			

Werte

	A	B	...
1	n	x	
2	1	-5	
3	2	-4.8	
4	3	-4.6	
5	4	-4.4	
...	...	...	
51	50	4.8	
52	51	5	
...			

Induktion durch *Kopieren* des Induktionsschritts

Induktion

Kombination mit der iterativen Berechnung der Funktionswerte

Ausdrücke

	A	B	C	D
1			x_i	$f(x_i)$
2	a_0	0	-5	=b5*c2**3+b4*c2**2+b3*c2+b2
3	a_1	-20	=c2+b6	=b5*c3**3+b4*c3**2+b3*c3+b2
4	a_2	0	=c3+b6	=b5*c4**3+b4*c4**2+b3*c4+b2
5	a_3	1	=c4+b6	=b5*c5**3+b4*c5**2+b3*c5+b2
6	Δx	0.2	=c5+b6	=b5*c5**3+b4*c5**2+b3*c5+b2
...			...	...
51			=c50+b6	=b5*c51**3+b4*c51**2+b3*c51+b2
52			=c51+b6	=b5*c52**3+b4*c52**2+b3*c52+b2
...				

Werte

A	B	C	D
		x_i	$f(x_i)$
a_0	0	-5	-25
a_1	-20	-4.8	-14.592
a_2	0	-4.6	-5.336
a_3	1	-4.4	2.816
Δx	0.2	-4.2	9.912
	...	...	
		4.8	14.592
		5	25

Exkurs: Modellierung und Simulation

Modellierung: Aufstellen eines funktionalen Zusammenhangs

freie Parameter: Wertevorrat der Funktion

Modellparameter: Konstanten des funktionalen Audrucks

Beispiel: $s = \frac{1}{2}at^2 + v_0t + s_0$

freier Parameter: t

Modellparameter: a, v_0, s_0

Simulation: Studieren des Modellverhaltens unter Variation von Modellparametern

Simulation in der Kalkulationstabelle:

freier Parameter $\rightarrow$ relative Referenz

Modellparameter $\rightarrow$ absolute Referenz

Beispiel: Persönliche Zahlungsbilanz

monatliche Einnahmen / Ausgaben

Induktionsanfang: anfängliches Guthaben

Induktionsschritt: monatlicher Zuwachs (möglicherweise auch negativ!):
neuer Bestand = alter Bestand + Einnahmen − Ausgaben

Ausdrücke

	A	B	C	D	...
1	Monat	Einnahmen	Ausgaben	Bilanz	
2				7638.47	
3	Januar	3125.00	2327.67	=d2+b3-c3	
4	Februar	2981.56	3376.78	=d3+b4-c4	
5	März	3225.98	1978.42	=d4+b5-c5	
6	April	3100.34	2785.68	=d5+b6-c6	
...					

Werte

	A	B	C	D	...
1	Monat	Einnahmen	Ausgaben	Bilanz	
2				7638.47	
3	Januar	3125.00	2327.67	8435.80	
4	Februar	2981.56	3376.78	8040.50	
5	März	3225.98	1978.42	9288.14	
6	April	3100.34	2785.68	9602.80	
...					

Beispiel: Prognose

Frage: Kann ich mir noch in diesem Jahr ein Auto leisten?

Annahmen:

Preis 15000 DM

lineares Fortschreiben der Zahlungsbilanz

Lösung: lineare Regression

Schätzen einer Regressionsgeraden nach dem Verfahren der kleinsten Fehlerquadrate:

$$y = a_x + b_x x$$

$$b_x = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{\sum_{i=1}^{n} x_i y_i - \frac{1}{n}[\sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i]}{\sum_{i=1}^{n} x_i^2 - \frac{1}{n}[\sum_{i=1}^{n} x_i]^2}$$

eingebaute Funktionen:

achsenabschnitt(Y-Werte;X-Werte)
steigung(Y-Werte;X-Werte)

Beispiel: Prognose

Ausdrücke

	...	D	E	F	...
1		Bilanz	n	Prognose	
2		7638.47	0	=d17+e17*e2	
3		=d2+b3-c3	=e2+1	=d17+e17*e3	
4		=d3+b4-c4	=e3+1	=d17+e17*e4	
5		=d4+b5-c5	=e4+1	=d17+e17*e5	
6		=d5+b6-c6	=e5+1	=d17+e17*e6	
7			=e6+1	=d17+e17*e7	
8			=e7+1	=d17+e17*e8	
9			=e8+1	=d17+e17*e9	
10			=e9+1	=d17+e17*e10	
11			=e10+1	=d17+e17*e11	
12			=e11+1	=d17+e17*e12	
13			=e12+1	=d17+e17*e13	
14			=e13+1	=d17+e17*e14	
15					
16		a_x	b_x		
17		=achsenabschnitt(d3:d6)	=steigung(e3:e6)		
...					

Werte

D	E	F
Bilanz	n	Prognose
7638.47	0	7644.96
8435.80	1	8123.06
8040.58	2	8601.16
9288.14	3	9079.26
9602.80	4	9557.36
	5	10035.46
	6	10513.56
	7	10991.66
	8	11469.76
	9	11947.86
	10	12425.96
	11	12904.06
	12	13382.16
a_x	b_x	
7644.96	478.1	

Frage: Wann kann ich mir endlich das Auto leisten?

Beispiel: Geometrische Folge

Frage: Welche Frequenzen haben die Töne einer chromatischen Tonleiter (gleichschwebend-temperierte Stimmung)?

Induktionsanfang: beliebiger Grundton (z.B. Kammerton A mit 440 Hz)

Induktionsschritt: Geometrische Folge: $f_n = qf_{n-1}$

Wie groß ist q?

analytisch: $q = \sqrt[12]{2}$

Zusätzliche Randbedingung: Oktave entspricht einer Frequenzverdopplung: $f_12 = 2f_0$

Lösung durch Simulation unterschiedlicher Werte für q

Beispiel: Geometrische Folge

Ausdrücke

	A	B	C	...
1	q	n	f_n	
2		0	440	
3		=c2+1	=a2*c2	
4		=c3+1	=a2*c3	
5		=c4+1	=a2*c4	
...		...	...	
13		=c12+1	=a2*c8	
14		=c13+1	=a2*c9	
...				

Werte

	A	B	C	...
1	q	n	f_n	
2		0	440	
3		1	0	
4		2	0	
5		3	0	
...		...	...	
13		11	0	
14		12	0	
...				

Werte

	A	B	C	...
1	q	n	f_n	
2	1.06	0	440	
3		1	466.4	
4		2	494.38	
5		3	524.38	
...		...	...	
9		7	835.25	
10		8	885.37	
...				

Beispiel: Wachstumsmodelle

Wachstum ohne Ressourcenbeschränkung: lineare Rekursionsgleichung

z.B. Zinseszins: $x_n = (1+z)x_{n-1}$ z: Zinssatz

lineare Modelle führen auf exponentielles Wachstum!

Wachstum mit Ressourcenbeschränkung: nichtlineare Gleichungen

z.B. logistische Gleichung: $x_n = 4cx_{n-1}(1-x_{n-1})$

Modellierung der Populationsgröße abgeschlossener biologischer Systeme, z.B.

Bakterien in einem Reagenzglas

Mücken auf einer einsamen Insel

c: abhängig vom Futtervorrat, Temperatur im Laboratorium usw.

x: Zahl der Individuen normiert auf die maximale Zahl, die das Gebiet am Leben erhalten kann

nichtlineare Wachstumsmodelle führen für spezielle Parametersetzungen entweder

zu einem stabilen Systemzustand (ein Attraktor)

zu oszillierendem Systemverhalten (mehrere Attraktoren) oder

zu chaotischem Systemverhalten

Rekursion

Rekursion: Rückführung eines komplexen Problems auf ein einfacheres.

Jede Induktionsvorschrift kann rekursiv interpretiert werden.

$$n! = \begin{cases} 1 & \text{für } n = 0 \qquad \text{Induktionsanfang} \\ (n-1)! * n & \text{sonst} \qquad \text{Induktionsschritt} \end{cases}$$

$$\begin{array}{cccc}
 & 10! = 9! * 10 & 10! = 3628800 & \\
 & 9! = 8! * 9 & 9! = 362880 & \\
\text{rekursiver} & 8! = 7! * 8 & 8! = 40320 & \Uparrow \\
\text{Abstieg} & 7! = 6! * 7 & 7! = 5040 & \text{rekursiver} \\
\Downarrow & 6! = 5! * 6 & 6! = 720 & \text{Aufstieg} \\
 & \cdots & \cdots & \\
 & 1! = 0! * 1 & 1! = 1 & \\
 & 0! = 1 & &
\end{array}$$

Rekursion

Rekursion und Induktion sind duale Sichtweisen

Induktion	Rekursion
Induktionsschritt	Rekursionsschritt
Induktionsanfang	Rekursionsabschluß
Generalisierung vom Induktionsanfang aus	Rückführung auf den Rekursionsabschluß

Induktion ist eine Rekursion, bei der die Berechnung ausschließlich beim rekursiven Aufstieg erfolgt.

$\rightarrow$ direkte iterative Implementierung ist möglich, wenn der Verknüpfungsoperator kommutativ (symmetrisch) ist

Rekursion

rekursive Berechnung erfordert die Verwaltung der Rekursionsaufrufe (partiell ausgewertete funktionale Ausdrücke) → Kellerspeicher

Kellerspeicher: (Stack) einseitig beschränkter Speicher mit Zugriff auf das oberste Element

last in first out (LIFO)

Aufruf: 5! Ergebnis: 120

5!=4!*5	4!=3!*4	3!=2!*3	2!=1!*2	1!=0!*1	2!=1*2	3!=2*3	4!=6*4	5!=24*5	
	5!=4!*5	4!=3!*4	3!=2!*3	2!=1!*2	3!=2!*3	4!=3!*4	5!=4!*5		
		5!=4!*5	4!=3!*4	3!=2!*3	4!=3!*4	5!=4!*5			
			5!=4!*5	4!=3!*4	5!=4!*5				
				5!=4!*5					
push	push	push	push	push	pop	pop	pop	pop	pop

Rekursion

Rekursion liegt in der Kalkulationstabelle immer dann vor, wenn beim Kopieren eines funktionalen Ausdrucks eine Referenz auf den jeweils vorangehenden Ausdruck hergestellt wird.

Für das Zustandekommen rekursiver Abhängigkeitsbeziehungen durch Kopieren ist die relative Referenz von zentraler Bedeutung.

Der Kellerspeicher wird in der Kalkulationstabelle durch Kopieren manuell simuliert.

induktiv (vorwärtgerichtet):

Ausdrücke

	A	B	...
1	n	fak(n)	
2	0	1	
3	=a2+1	=b2*a3	
4	=a3+1	=b3*a4	
5	=a4+1	=b4*a5	
...	...	...	
11	=a10+1	=b10*a11	
12	=a11+1	=b11*a12	
...			

Werte

	A	B	...
1	n	fak(n)	
2	0	1	
3	1	1	
4	2	2	
5	3	6	
...	...	...	
11	9	362880	
12	10	3628800	
...			

rekursiv (rückwärtsbezogen):

Ausdrücke

	A	B	...
1	n	fak(n)	
2	5	=b3*a2	
3	=a2-1	=b4*a3	
4	=a3-1	=b5*a4	
5	=a4-1	=b6*a5	
...	...	...	
11	=a10-1	=b12*a11	
12	=a11-1	1	
...			

Werte

	A	B	...
1	n	fak(n)	
2	10	3628800	
3	9	362880	
4	8	40320	
5	7	5040	
...	...	...	
11	1	1	
12	0	1	
...			

Ausdrücke

	A	B	...
1	n	fak(n)	
2	5	=wenn(a2>0;b3*a2;1)	
3	=a2-1	=wenn(a3>0;b4*a3;1)	
4	=a3-1	=wenn(a4>0;b5*a4;1)	
5	=a4-1	=wenn(a5>0;b6*a5;1)	
...	...	...	
11	=a10-1	=wenn(a11>0;b12*a11;1)	
12	=a11-1	=wenn(a12>0;b13*a12;1)	
13	=a12-1	=wenn(a13>0;b14*a13;1)	
...			

Werte

	A	B	...
1	n	fak(n)	
2	10	3628800	
3	9	362880	
4	8	40320	
5	7	5040	
...	...	...	
11	1	1	
12	0	1	
13	-1	1	
...			

Schlichte Rekursion

schlichte Rekursion: Berechnung nur auf einem Ast der Rekursion

aufsteigende Rekursion: $\sim$ Induktion
rekursiver Abstieg erfolgt nur, um den
Induktionsanfang/Rekursionsabschluß zu finden
alle Berechnungen erfolgen beim rekursiven Aufstieg

absteigende Rekursion: z.B. wenn Induktionsanfang/Rekursionsabschluß
noch unbekannt ist
die gesamte Berechnung erfolgt nur beim rekursiven Abstieg
das Ergebnis liegt beim Rekursionsabschluß vor
der rekursive Aufstieg dient nur der Übermittlung des Ergebnisses

$\rightarrow$ repetitive Rekursion / Endrekursion / tail recursion

keine Berechnung im aufsteigenden Ast

$\rightarrow$ iterative Implementierung ist möglich

Schlichte Rekursion

Beispiel: Größter gemeinsamer Teiler zweier natürlicher Zahlen

$$\mathrm{gcd}(x,y) = \begin{cases} x & \text{wenn } x = y \\ \mathrm{gcd}(y,x) & \text{wenn } x < y \\ \mathrm{gcd}(x-y,y) & \text{sonst} \end{cases}$$

Ausdrücke

	A	B	C	...
1	x	y	gcd(x,y)	
2	15	9	=wenn(a2=b2;a2;0)	
3	=wenn(a2<b2,b2,a2-b2)	=wenn(a2<b2,a2,b2)	=wenn(a3=b3;a3;0)	
4	=wenn(a3<b3,b3,a3-b3)	=wenn(a3<b3,a3,b3)	=wenn(a4=b4;a4;0)	
5	=wenn(a4<b4,b4,a4-b4)	=wenn(a4<b4,a4,b4)	=wenn(a5=b5;a5;0)	
6	=wenn(a5<b5,b5,a5-b5)	=wenn(a5<b5,a5,b5)	=wenn(a6=b6;a6;0)	
7	=wenn(a6<b6,b6,a6-b6)	=wenn(a6<b6,a6,b6)	=wenn(a7=b7;a7;0)	
...				

Werte

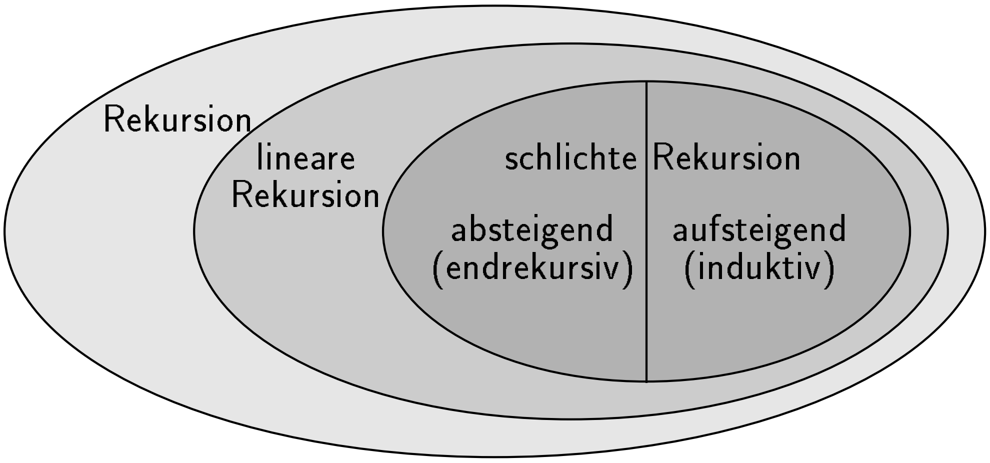
A	B	C
x	y	gcd(x,y)
15	9	0
6	9	0
9	6	0
3	6	0
6	3	0
3	3	3

Weitere Beispiele: last(Liste), length(Liste)

Systematik

Lineare Rekursion: nur ein Rekursionsbezug pro Ausdruck

Verzweigende Rekursion: Mehrere Rekursionsbezüge in einem Ausdruck (nur für komplexe Datenstrukturen typisch)



Schlichte Rekursion

aufsteigende (induktive) Berechnungsvorschriften können immer in absteigende (endrekursive) Verfahren umgewandelt werden

Trick: Ergebnisakkumulator

Beispiel: fak(n)

aufsteigend rekursiv (induktiv):

$$\text{fak}(n) = \begin{cases} 1 & \text{für } n = 0 \\ \text{fak}(n-1) * n & \text{sonst} \end{cases}$$

absteigend rekursiv (endrekursiv):

$$\text{fak}(n) = \text{fak1}(n, 1)$$

$$\text{fak1}(n, a) = \begin{cases} a & \text{für } n = 0 \\ \text{fak}(n-1, n * a) & \text{sonst} \end{cases}$$

Iteration / Rekursion

Iteration	Rekursion
Wiederholte Verwendung eines Ausdrucks durch Kopieren	
ohne Referenz auf den vorangehenden Zelleninhalt	mit Referenz auf den vorangehenden Zelleninhalt

Ziel: iterative Abarbeitung induktiver und endrekursiver Berechnungsvorschriften

Ist das auch in Kalkulationstabellen möglich?

→ keine Referenzbeziehungen zwischen Kopie und Original beim Kopieren!

Iteration II

Warum Iteration? Effizienzvorteile:

- weniger Speicherbedarf bei zyklischen Berechnungen
- geringerer Aufwand zur Beschreibung eines Prozeßzustandes bei Unterbrechnung der Abarbeitung (Restart-Fähigkeit)

schlichte Rekursion (induktiv oder repetitiv) kann auch iterativ abgearbeitet werden

Iteration durch wiederholte Abarbeitung selbstbezüglicher (zyklischer) Audrücke

- **Wiederverwendung** der Variablen bei der zyklischen Abarbeitung
  $\rightarrow$ Fehlerzustand: zyklische Referenz
  $\rightarrow$ Speichereffizienz!
- **Aktualisierung** der Variablen durch manuelles Wiederanstoßen der Berechnung (F9)

Einstellung: Extras $\rightarrow$ Optionen $\rightarrow$ Berechnen:
"manuell" einschalten und Iteration auf 1 setzen

Iteration II

Beispiel: Zählvariable: $x = \{1, 2, 3, \ldots\}$

Ausdrücke

	A	...
1	=a1+1	
...		

Werte

	A	
1	1	
...		

Werte

	A	
1	2	
...		

Werte

	A	
1	3	
...		

Funktioniert nur, weil die Standardwertebelegung einer Zelle Null ist!

Beispiel: Zählvariable mit beliebigem Anfangswert $x = \{5, 6, 7, \ldots\}$

Ausdrücke

	A	...
1	=wenn(a1=0;5;a1+1)	
...		

Werte

	A	
1	5	
...		

Werte

	A	
1	6	
...		

Werte

	A	
1	7	
...		

Funktioniert nicht, falls die Null Element der zu erzeugenden Zahlenmenge ist!

Iteration II

Beispiel: Fakultät, induktiv

Ausdrücke

	A	B	...
1	n	=b1+1	
2	n!	=wenn(b1=0;1;b2*b1)	
...			

Werte

B	B	B
1	2	3
1	2	6

Achtung: Ergebnis ist abhängig von der Reihenfolge der Zellenauswertungen!
Standardreihenfolge: links → rechts, oben → unten

Beispiel: Fakultät, endrekursiv

Ausdrücke

	A	B	...
1	n	3	
2	z	=wenn(b2=0;b1;b2-1)	
3	a	=wenn(b3=0;b1;b3*b2)	
4	n!	=wenn(b2=1;b3;b4)	
...			

Werte

B	B	B
3	3	3
0	3	2
0	3	3
0	0	0

Werte

B	B	B
3	3	3
1	0	3
6	6	0
0	6	6

Achtung: Variablenbelegung ist abhängig von den aktuellen Anfangswerten!

Iteration II

Beispiel: Größter gemeinsamer Teiler

$$\mathrm{gcd}(x,y) = \begin{cases} x & \text{wenn } x = y \\ \mathrm{gcd}(y,x) & \text{wenn } x < y \\ \mathrm{gcd}(x-y,y) & \text{sonst} \end{cases}$$

Ausdrücke

	A	B	C	...
1		n	m	
2	A-werte	15	9	
3	Akkumulator	=wenn(b3=0;b2;b5)	=wenn(c3=0;c2;c5)	
4	Vertauscht	=wenn(b3>c3;b3;c3)	=wenn(b3>c3;c3;b3)	
5	Differenz	=b4-c4	=c4	
6	Resultat	=wenn(b3=c3;b3;0)		
...				

Werte

	B	C	
	15	9	
	15	9	
	15	9	
	6	9	
	0		

Werte

	B	C	
	15	9	
	6	9	
	9	6	
	3	6	
	0		

Werte

	B	C	
	15	9	
	3	6	
	6	3	
	3	3	
	0		

Werte

	B	C	
	15	9	
	3	3	
	3	3	
	0	3	
	3		

Grafische Darstellung geometrischer Figuren

Visualisierung von funktionalen Zusammenhängen:

Approximation als Liniendiagramm mit äquidistanten Stützpunkten:
Diagramm → Liniendiagramm

variabler Stützpunktabstand: Darstellung als X/Y-Diagramm
Diagramm → X/Y-Diagramm
vgl. Polynomdarstellung mit variabler Schrittweite

Visualisierung nichtfunktionaler Zusammenhänge:

Umwandlung in eine Funktion durch Parametrisierung: x(t),y(t)

Darstellung durch ein X/Y-Diagramm

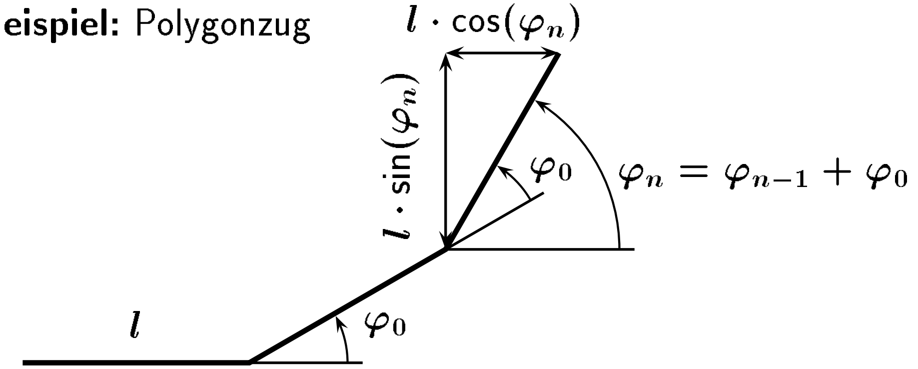
Beispiel: Rechteck

Werte

	A	B	C	...
1	t	x(t)	y(t)	
2	1	1	1	
3	2	5	1	
4	3	5	4	
5	4	1	4	
6	5	1	1	
...				

Grafische Darstellung geometrischer Figuren

Beispiel: Polygonzug



Ausdrücke

	A	B	C	D	
1	länge	phi(t)	x(t)	y(t)	
2	3	=a4	1	1	
3	phi0	=b2+a4	=c2+a2*cos(b3)	=d2+a2*sin(b3)	
4	=pi()/2	=b3+a4	=c3+a2*cos(b4)	=d3+a2*sin(b4)	
5		=b4+a4	=c4+a2*cos(b5)	=d4+a2*sin(b5)	
6		=b5+a4	=c5+a2*cos(b6)	=d5+a2*sin(b6)	
...		...	...	...	

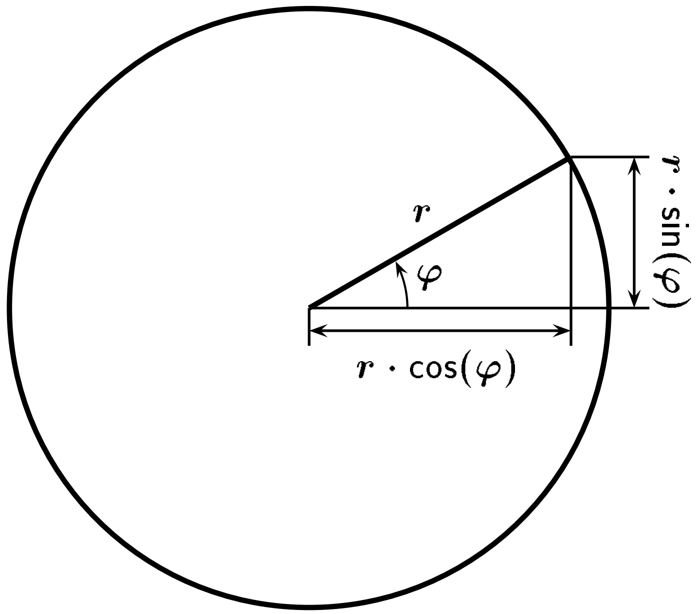
Werte

	A	B	C	D	...
	länge	phi(t)	x(t)	y(t)	
	2	1.570	1	1	
	phi0	3.141	-1	1	
	=pi/2	4.712	-1	-1	
		6.283	1	-1	
		7.853	1	1	
		...	...	...	

Kreisfiguren

Beispiel: Kreis

Parameter: Drehwinkel φ



Ausdrücke

	A	B	C	D
1	radius	phi(t)	x(t)	y(t)
2	3	0	=\$a\$2*cos(b2)	=\$a\$2*sin(b2)
3	**Δ**phi	=b2+\$a\$4	=\$a\$2*cos(b3)	=\$a\$2*sin(b3)
4	=pi()/10	=b3+\$a\$4	=\$a\$2*cos(b4)	=\$a\$2*sin(b4)
5		=b4+\$a\$4	\$a\$2*cos(b5)	=\$a\$2*sin(b5)
6		=b5+\$a\$4	=\$a\$2*cos(b6)	=\$a\$2*sin(b6)
...		...	...	...

Werte

	A	B	C	D	...
1	länge	phi(t)	x(t)	y(t)	
2	3	0	3	0	
3	**Δ**phi	0.314	2.853	0.927	
4	0.314	0.628	2.427	1.763	
5		0.942	1.763	2.427	
6		1.256	0.927	2.853	
...		...	...	...	

Kreisfiguren

Variation 1: Spirale

Ausdrücke

	A	B	C	D	E	...
1	radius	phi(t)	rad(t)	x(t)	y(t)	
2	3	0	=a2	=\$a\$2*cos(b2)	=\$a\$2*sin(b2)	
3	Δphi	=b2+\$a\$4	=c2-\$a\$6	=\$a\$2*cos(b3)	=\$a\$2*sin(b3)	
4	=pi()/10	=b3+\$a\$4	=c3-\$a\$6	=\$a\$2*cos(b4)	=\$a\$2*sin(b4)	
5	Δrad	=b4+\$a\$4	=c3-\$a\$6	\$a\$2*cos(b5)	=\$a\$2*sin(b5)	
6	0.1	=b5+\$a\$4	=c4-\$a\$6	=\$a\$2*cos(b6)	=\$a\$2*sin(b6)	
...		...	...	...		

Werte

	A	B	C	D	E	...
	länge	phi(t)	x(t)	y(t)		
	3	0	3	3	0	
	Δphi	0.314	2.9	2.758	0.896	
	0.314	0.628	2.8	2.265	1.645	
	Δrad	0.942	2.7	1.587	2.184	
	0.1	1.256	2.6	0.803	2.472	
		...	...	...		

Kreisfiguren

Variation 2: x- und y-Koordinaten werden durch Ortsvektoren mit unterschiedlicher Drehgeschwindigkeit (Kreisfrequenz) ermittelt.

Ausdrücke

	A	B	C	D
1	radius	phi(t)	x(t)	y(t)
2	3	0	=a2*cos(a6*b2)	=a2*sin(a8*b2)
3	**Δ**phi	=b2+a4	=a2*cos(a6*b3)	=a2*sin(a8*b3)
4	=pi()/10	=b3+a4	=a2*cos(a6*b4)	=a2*sin(a8*b4)
5	ω_x	=b4+a4	=a2*cos(a6*b5)	=a2*sin(a8*b5)
6	2	=b5+a4	=a2*cos(a6*b6)	=a2*sin(a8*b6)
7	ω_y	...	...	...
8	3			
...				

Werte

A	B	C	D
länge	phi(t)	x(t)	y(t)
3	0	3	0
Δphi	0.314	2.427	2.427
0.314	0.628	0.927	2.853
ω_x	0.942	-0.927	0.927
2	1.256	-2.427	-1.763
ω_y	...	...	...
3			

Variation 3: zusätzlich kann für jede Koordinate ein individueller Anfangsdrehwinkel (Phasenwinkel) eingestellt werden.

→ LISSAJOUS'sche Figuren

Animation

Animation: Erzeugen einer Bildfolge mit kleinen, aber systematischen Variationen ausgewählter Parameter

Neuberechnen der internen Bildrepräsentation durch *Iteration*

Beispiel: analoge Uhr mit zwei Zeigern

Ausdrücke

	A	B	C	
1		x(t)	y(t)	
2	=a2+a3	=sin(a2)	=cos(b2)	
3	=pi()/20	0	0	
4		=0.8*sin(1/12*a2)	=0.8*cos(1/12*b2)	
5				
6		1	1	
7				
8		-1	-1	
...				

Erzeugung der Bildfolge durch wiederholte Auswertung der Kalkulationstabelle (F9)

Datenbanken

Listen Sammlung gleichartiger Daten in aufeinanderfolgenden Tabellenspalten (Datenbereich)

Ausdrücke

	A	B	C	D
1	Typ	Straße	Nummer	Jahr
2				
3	EFH	Gärtnerstr	15	1965
4	EFH	Bahnhofsstr	27	1943
5	EFH	Bahnhofsstr	29	1955
6	MFH	Bahnhofsstr	28	1991
7	Bahnhof	Bahnhofsstr	30	1901
8	Kaufhaus	Bahnhofsstr	26	1997
9	EFH	Gärtnerstr	17	1982
...				

Dateneingabe durch Masken
$\rightarrow$ Daten $\rightarrow$ Maske

Datenbanken

Selektion durch Angabe von Filterkriterien (Kriterienbereich)
$\rightarrow$ Daten $\rightarrow$ Filter $\rightarrow$ Spezialfilter

Konjunktion von Selektionsbedingungen:

```
select * from obj where type='EFH' and str='Gaertnerstr'
```

Ausdrücke

...	F	G	...
	Typ	Straße	
	EFH	Gärtnerstr	

Disjunktion von Selektionsbedingungen:

```
select * from obj where type='EFH' or type='MFH'
```

Ausdrücke

	F	G	
	Typ	Straße	
	EFH		
	MFH		

Datenbanken

Konjunktion und Disjunktion von Selektionsbedingungen:

```
select * from obj where type='EFH' and (jahr>1950 or
jahr<1990)
```

Ausdrücke

...	F	G	...
	Typ	Jahr	
	EFH	>1950	
	EFH	<1990	
...			

Filterkriterium kann auch berechnet werden

Ausdrücke

...	F	...
	Jahr	
	=d3>mittelwert(d3:d9)	
...		

Begrenzung: nur einzelne Tabellen können gefiltert werden!

Datenbanken

Aggregatfunktionen

für beliebige Bereiche:
summewenn, zaehlewenn

für Datenbanken (auch externe):
dbanzahl, dbmax, dbmin
dbmittelwert, dbsumme
...

Sortieren

Datenimport

Export von SQL-Anfragen: sql.request

Grenzen der Tabellenkalkulation

Steuerstrukturen nur rudimentär

Zyklenbildung

a) durch Kopieren der Berechnungsvorschrift
b) durch manuelle Wiederauswertung bei zyklischen Berechnungsvorschriften (globale Iteration)

Verzweigung: Konditional ist immer an eine Variable gebunden

Probleme bei iterativer Programmierung, wenn eine Bedingung für zwei Variablenbelegungen relevant ist, aber eine wechselseitige Abhängigkeit besteht:

Ausdrücke

	A	B	...
1	=wenn(a1=b1;a1+1;0)	=wenn(a1=b1;b1+1;0)	

Rekursion: kein infiniter Regreß möglich

Grenzen der Tabellenkalkulation

Prozedurabstraktion fehlt

- kein Unterprogrammkonzept
- keine Modularisierung
- extreme Probleme bei der Strukturierung großer Datenverarbeitungaufgaben
- sehr hohe Fehleranfälligkeit

Datenabstraktion fehlt

direkt verfügbar: nur atomare Datentypen: Zahlen, Zeichenketten

beschränkt verfügbar: Records (Datenbanksätze)

Grenzen der Tabellenkalkulation

indirekt verfügbar: Felder (Arrays)

Bereiche können einen Namen besitzen

Zugriff auf die einzelnen Zellen eines Bereiches kann über einen variablen Index erfolgen

index(Bereich;X-offset;Y-offset)

Ausdrücke

	A	B	C	...
1	3	4	=index(a1:a5;c2;c3)	
2	5	5	2	
3	4	7	3	
4	2	5		
5	4	4		
...				

vollständig fehlen: Listen, Terme (rekursive Datenstrukturen)

Beschränkung auf funktionale Abhängigkeiten

nur gerichtete Werteausbreitung

Erweiterung: Constraint Propagation