Hilfsmittel Mathematik" Rekursionsgleichungen

M. Neumann, C. Piechotta

30. März 2009

Motivation

Motivation!

Definition der Fibonacci-Zahlenfolge

$$
\begin{aligned}
F(0) &= 1 \\
F(1) &= 1 \\
F(n) &= F(n\text{-}1)+F(n\text{-}2),\ \text{mit } n \geq 2
\end{aligned}
$$

rekursive Berechnungsvorschrift

```
(define fib
    (lambda (n)
        (if(< n 1)
            1
            (+(fib(- n 1))(fib(- n 2))))))
```

rekursive Berechnungsvorschrift

```
(define fib
   (lambda (n)
      (if(< n 1) ← Fehler!
         1
         (+(fib(- n 1))(fib(- n 2))))))
```

rekursive Berechnungsvorschrift

```
(define fib
   (lambda (n)
      (if(< n 2)
         1
         (+(fib(- n 1))(fib(- n 2))))))
```

rekursive Aufwandsgleichung

$$
\begin{aligned}
T(0) &= 1 \\
T(1) &= 1 \\
T(n) &= T(n\text{-}1)+T(n\text{-}2),\ \text{mit } n \geq 2
\end{aligned}
$$

explizite Gleichung

Inhaltsverzeichnis

Raten und Einsetzen

- intelligent guesswork - Intelligentes Raten
- Wertetabelle aufstellen
- explizite Bildungsvorschrift ermitteln

Konstruktive Induktion

1. Vermutung aufstellen
2. für das Anfangsglied beweisen(z.B. n=1)
3. für alle Folgeglieder beweisen(z.B. n+1)

Computeralgebrasysteme CAS

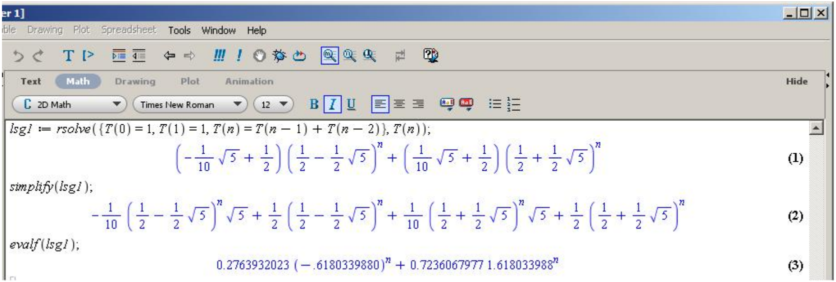
- computergestützte Werkzeuge
- lösen numerische oder symbolische Aufgaben (z.B. Gleichungsumformung)
- basieren auf unterschiedlichen mathematischen Grundlagen
- z.B. Matlab, Mathematica, Maple, Maxima

Lösung rekursiver Gleichungen

- basierend auf komplexen Z-Transformationen
- Implementierung für bestimmte Gleichungsklassen
- $\hookrightarrow$ nur bestimmte Gleichungen lösbar

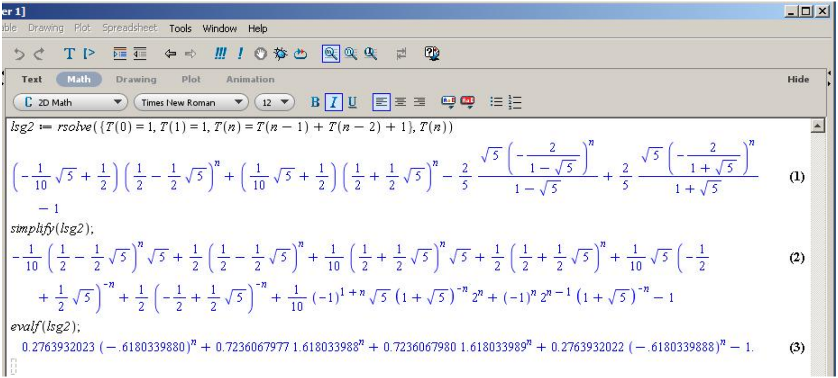
Aufwandsgleichung der Fibonacci-Folge mit Maple

Mit einfachem Aufwand



Aufwandsgleichung der Fibonacci-Folge mit Maple

Mit höherem Aufwand



Iterationsmethode

- Idee $\Rightarrow$ Rekursion $\approx$ Schleife
- Erweitere die Gleichung bis zu einem rekursionsfreien n
- Danach beende die Iteration

Beispiel

- $T(1) = 1$
- $T(n) = 3 \cdot T(\frac{n}{4}) + n, mitn = 4^a \setminus a \in \mathbf{N}$

Expansion der Gleichung

- $T(n) = n + 3 \cdot T(\frac{n}{4})$
- ...
- $T(n) = n + \frac{3}{4}n + \frac{9}{16}n + \frac{27}{64}n + \cdot + \frac{3^{i-1}}{4^{i-1}}n + 3^i \cdot T(\frac{n}{4^i})$
- $T(n) = n \sum_{k=0}^{i-1} (\frac{3}{4})^k + 3^i$
- $T(n) = n \cdot \frac{1 - (\frac{3}{4})^i}{\frac{1}{4}} + 3^i$
- $T(n) = 4n - 4n(\frac{3}{4})^i + 3^i$
- ...
- $\lessapprox n$
- $T(n) = O(n)$

Die Mastermethode

Diese Methode funktioniert in etwa nach dem Aschenputtelprinzip - finde den passenden Schuh.

Allerdings nicht immer mit Happy End!

Der Satz

Sind $n \geq 1$ und $b \geq 1$ Konstanten sowie $f : \mathbf{N} \mapsto \mathbf{R}$ und $T : \mathbf{N} \mapsto \mathbf{R}$ Funktionen, der Form: $T(n) = a \cdot T(\frac{n}{b}) + f(n)$ Dann gilt für T(n):

1. Wenn $f(n) = \mathcal{O}(n^{\log_b a - \epsilon})$ für eine reele Konstante $\epsilon > 0$, dann gilt $T(n) = \Theta(n^{\log_b a})$.
2. Wenn $f(n) = \Theta(n^{\log_b a})$, dann gilt $T(n) = \Theta(n^{\log_b a} \log_2 n)$.
3. Wenn $f(n) = \Omega(n^{\log_b a + \epsilon})$ für beliebige reelle Konstanten $\epsilon > 0$ und wenn $a \cdot f(\frac{n}{b}) \leq c \cdot f(n)$, mit eineer reellen Konstante $c > 0$, dann gilt $T(n) = \Theta(f(n))$.

Beachte

- $T(n)$ und $f(n)$ sind voneinander unabhängig
- $a \geq 1,\ b > 1$
- $f(n)$ wird immer mit $n^{\log_b a}$ verglichen
- $f(n)$ bezeichnet Kombinationsaufwand

Fall 1

$T(n) = 8 \cdot T(\frac{n}{2}) + 1000n^2.$

- $a = 8, b = 2 \Rightarrow \log_2 8 = 3$
- $\hookrightarrow \mathcal{O}(n^3)$
- $f(n) \approx \mathcal{O}(n^2)$
- mit $\mathcal{O}(n^{3-\varepsilon}) \Rightarrow \varepsilon = 1$
- $\varepsilon > 0$

Fall 2

$T(n) = 2 \cdot T(\frac{n}{2}) + n.$

- $a = 2, b = 2 \Rightarrow \log_2 2 = 1$
- $\hookrightarrow \mathcal{O}(n)$
- $f(n) = n^{\log_2 2} \approx \theta(n)$

Fall 3

$T(n) = 2 \cdot T(\frac{n}{2}) + n^2.$

- $a = 2, b = 2 \Rightarrow \log_2 2 = 1$
- $\hookrightarrow \mathcal{O}(n)$
- $f(n) \approx \mathcal{O}(n^2)$
- mit $\mathcal{O}(n^{1+\varepsilon}) \Rightarrow \varepsilon = 1$
- $\varepsilon > 0$
- $Ausserdem : a \cdot f(\frac{n}{b}) \leq c \cdot f(n)) \setminus 0 < c < 1$
- $c = \dfrac{1}{2}$

Fall 4

$T(n) = 2 \cdot T(\frac{n}{2}) + n \log_2 n.$

- $a = 2, b = 2 \Rightarrow \log_2 2 = 1$
- $\hookrightarrow \mathcal{O}(n)$
- Aber: $f(n) : \mathcal{O}(n \log_2 n)$
- $\hookrightarrow$ nicht anwendbar.

Danke

Vielen Dank für eure Aufmerksamkeit.