Rekursion in und außerhalb der Informatik

Wo kommt die Rekursion vor?

Technik: Rekursion ist eine durch sich selbst definierte bzw. sich selbst aufrufende Funktion. Sie muss in endlich vielen Schritten terminieren und benötigt deshalb einen Trivialfall.

Linguistik: Rekursion ist in der Linguistik ein mehrfaches Auftreten gleicher Sprachformen. Im Gegensatz zum technischen Ansatz ist hier eine Endlosschleife möglich.

Kunst: In der Kunst gibt es für Bild-in-Bild Werke den Begriff *Mise en abyme.*

Rekursion in der Kunst/ Literatur – Endlose Texte und Bilder

Ein Mops kam in die Küche

Ein Mops kam in die Küche
und stahl dem Koch ein Ei.
Da nahm der Koch den Löffel
und schlug den Mops entzwei.

Da kamen viele Möpse
und gruben ihm ein Grab
und setzten drauf ´nen Grabstein,
darauf geschrieben stand:

Ein Mops kam in die Küche
und stahl dem Koch ein Ei…

Abbildung 1 ein 'Mise en abyme' - ein Bild in Bild

Eisgekühlte Coca-Cola

Ein belegtes Brot mit Schinken, ein belegtes Brot mit Ei.
Das sind zwei belegte Brote, eins mit Schinken und eins mit Ei.

Und dazu

Eisgekühlte Coca-Cola, Coca-Cola eisgekühlt.
Eisgekühlte Coca-Cola, Coca-Cola eisgekühlt

Und dazu

…

Rekursion in der Natur – Fibonacci Zahlen bei Pflanzen





Die Anzahl der im und gegen den Uhrzeigersinn verlaufenden Spiralen vieler Pflanzen sind im Allgemeinen benachbarte *Fibonacci Zahlen* (bzw. Lukas- Zahlen). Das Verhältnis zweier benachbarter Fibonacci Zahlen strebt gegen den Wert $\Phi \approx 1{,}618$[1], den *goldenen Schnitt.*

Interessant wird es auch, wenn man einen Kreis gemäß dem goldenen Schnitt teilt. Die daraus resultierenden Winkel sind $\frac{360°}{\Phi} \approx 222{,}5°$ bzw. $360° - 222{,}5° = 137{,}5°$. Der kleinere der beiden Winkel, Ψ=137,5° wird auch als *goldener Winkel* bezeichnet. Mit diesem kann man selbst Pflanzen wie den oben rechts abgebildeten Hauswurz zeichnen, dessen Blätter die charakteristischen Spiralen bilden.

$\frac{1}{1} = 1$

$\frac{2}{1} = 2$

$\frac{3}{2} = 1{,}5$

$\frac{5}{3} = 1{,}66\ldots$

$\frac{8}{5} = 1{,}6$

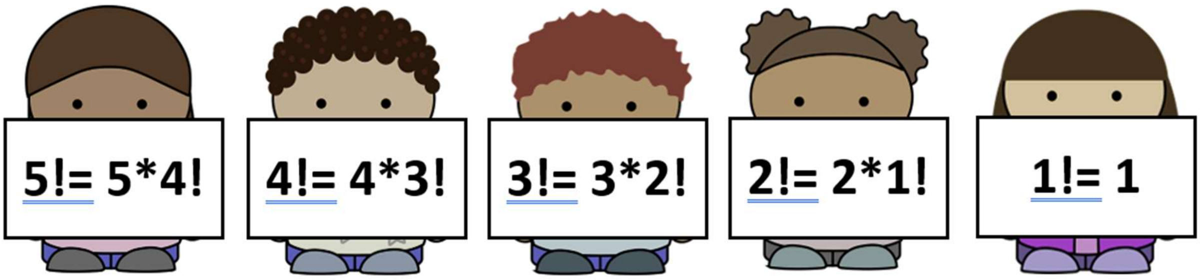
$\frac{13}{8} = 1{,}625$

$\frac{21}{13} = 1{,}615$

$\frac{34}{21} = 1{,}619$

[1] http://www.matheprisma.uni-wuppertal.de/Module/Rekurs/Golden2.htm (25.9.2018)

Rekursion in der Technik – der faule Informatiker



Idee: Ich stehe vor einem großen Problem, welches ich nicht berechnen kann. Dieses Problem wird in ein für mich leichteres Problem geteilt und der aufwändige Teil wird von einer anderen Person (einem Assistenten) berechnet! Dieser Assistent hat denselben Gedanken und wendet dieselbe Strategie an. Jeder folgende Assistent macht das genauso, bis das komplizierte Problem auf ein einfach zu lösendes Problem, einen sogenannten Trivialfall, zurückgeführt worden ist. Der letzte Assistent kann diesen Trivialfall ohne großen Aufwand lösen und teilt das Ergebnis mit, der Nächste berechnet das Ergebnis seines Problems und gibt es weiter. Dies wird so lange fortgeführt, bis das eigentliche Problem gelöst werden kann.

Diesen Gedankengang kann man für rekursive Probleme anwenden, um den Ablauf zu verstehen.

Divide and Conquer – Teile und Herrsche:

„Rekursion ist ein mächtiges Verfahren, das beim Problemlösen in der Informatik eingesetzt wird. In vielen Bereichen ermöglicht Rekursion bei komplexen Problemen sehr klar strukturierte und einfach zu durchschauende Lösungen.“[2]

In der Informatik wird Rekursion als Werkzeug eingesetzt, um komplexe Probleme in einfachere zu teilen und dadurch rückwirkend auf die Lösung zu kommen. Heutige Programmiersprachen sind in der Lage, rekursiv geschriebene Programme auszuführen. Solche rekursiven Programme können viel eleganter und kürzer ausfallen, als die iterative Variante und in manchen Fällen äußerst schnell sein. Doch muss man sich als Programmierer immer bewusst sein, dass Rekursion mehr Speicher benötigt und bei großen Zahlen viel länger braucht als die iterative Variante.

[2] https://www.inf-schule.de/algorithmen/algorithmen/rekursion (25.9.2018)

Faktorielle

Programm zur Berechnung von Faktoriellen in Python

Rekursive Funktion

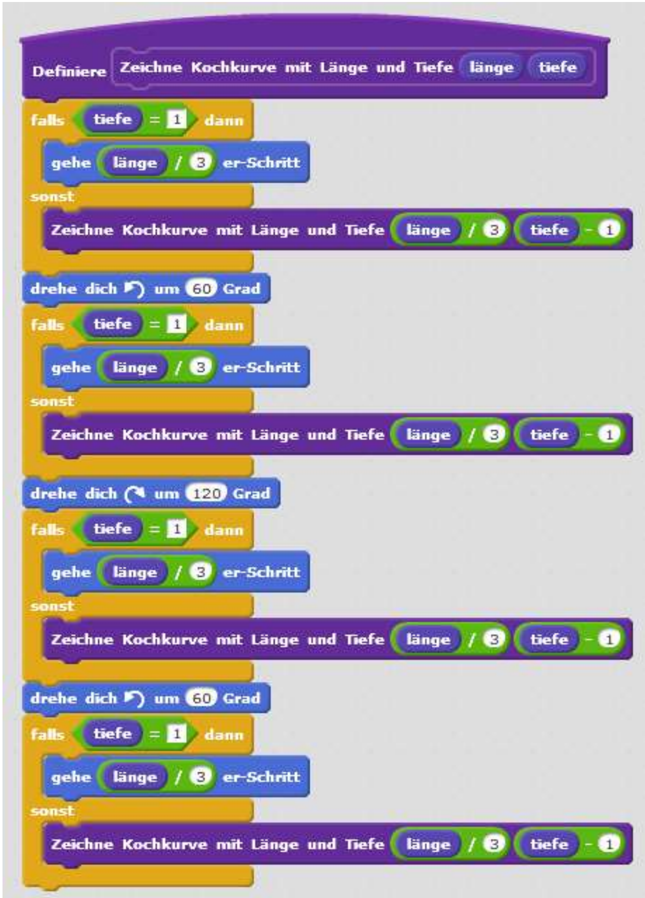
Trivialfall

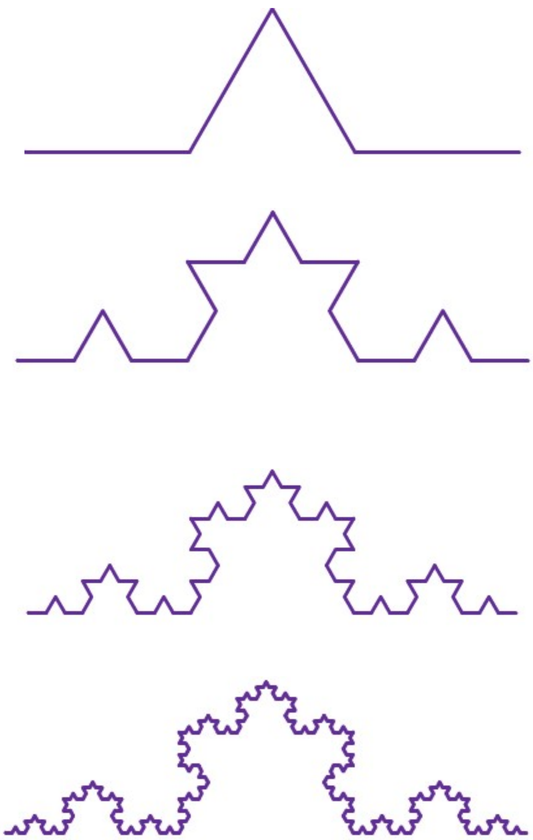
```python
def faktorielle(n):
    if n==1:
        return 1
    else:
        return n*faktorielle(n-1)

print(faktorielle(5))
```

Die grundlegende Struktur von rekursiven Funktionen ist immer gleich. Zuerst wird der Trivialfall abgedeckt, dann erst der nicht triviale Fall. Hier wird schlussendlich 5! ausgegeben.

Kochkurve





Die Berechnung der Kochkurve wurde hier mit Scratch realisiert. Der Zeichenvorgang wird geviertelt. Für jeden Teil wird zuerst der Trivialfall abgefragt, dann erst der nicht triviale Fall.

Die Kochkurve ist ein Fraktal, welches sich dadurch auszeichnet, dass sie unendlich lange, stetig, aber nicht differenzierbar ist. Dies trifft zu, wenn die Tiefe der Kurve gegen ∞ strebt. Links sind die Kochkurven der Tiefe 1, 2, 3 und 4 abgebildet.

Rekursion in der Assembler-Programmierung – Anwendungsbeispiel

Rekursives Assemblerprogramm - Faktorielle (=Fakultät)

Main-Programm

```
main:
li $a0, 4          #a0=4
jal faktorielle    #Berechnet Faktorielle von a0=4
                   und speichert das Ergebnis in v0
move $a0, $v0      #v0=a0
 li $v0, 1         #v0=1
syscall            #Ausgabe von Inhalt a0

 li $v0, 10        #v0=10
syscall            #Beendung des Programms
```

Trivialfall

```
faktorielle:

bne $a0 ,1, faktrek    #wenn a0≠1, dann springe zu faktrek,
                       sonst gehe weiter
li $v0, 1              # v0(Ergebnis) = 1
jr $ra                 # zur Return Adresse springen
                       (Adresse des letzten Sprungs)
```

Sicherung der Daten

```
faktrek:

addi $sp, $sp, -8      #Stackpointer auf richtige Stelle zeigen
sw $ra, 0($sp)         #Returnadresse für Sprung zurück speichern
sw $s0, 4($sp)         #s0 ist ein Zwischenspeicherregister, es muss
                       gesichert werden
```

Berechnung von 4! mithilfe rekursiven Aufrufen

```
move $s0, $a0          #Eingabe s0=a0 (1. Runde 4, dann 3,2…)
addi $a0, $s0, -1      #a0=s0-1 (1. Runde a0=4-1, dann 2, 1…
jal faktorielle        #gehe zu faktorielle mit Eingabeparameter a0=3
                       (bzw. 2,1…)
mult $s0, $v0          #multipliziere Ergebnis mit $s0=4 (3,2,…)
mflo $v0               #Ergebnis der Multiplikation in v0 speichern
lw $ra, 0($sp)         #Register, welche vorher gesichert wurden
lw $s0, 4($sp)         können wieder hergestellt werden, da sie nicht
addi $sp, $sp, 8       mehr benötigt werden und die Rücksprungadresse
                       richtig ist.
```

Rück-sprung

```
jr $ra                 #Rücksprung zu Anfang (main)
```

Weiterführende Links (25.9.2018)

https://deutschelieder.wordpress.com/2016/03/28/ein-hund-kam-in-die-kueche/
Seite der Bamberger Anthologie. Literarische Hintergründe und Textinterpretation des Liedes *Ein Mops kam in die Küche*.

https://www.youtube.com/user/Vihart
Kanal von *Vihart*, einer englischsprachigen Youtuberin, welche Mathematik, Natur und mittels Doodeling verbindet und dadurch interessante Ansätze schafft.
Empfehlenswerte Videos:
die dreiteilige Videoserie Doodling in Math: Spirals, Fibonacci, and Being a Plant

https://www.natuerlich-online.ch/magazin/artikel/der-fibonacci-code/
Artikel über der *Fibonacci-Folge* in Verbindung mit den Konstrukten von *Pflanzen*.

https://www.youtube.com/watch?v=MqZkl8_uF08
Video von Daniel Jung auf seinem Kanal *Mathe by Daniel Jung.* Herleitung des Goldenen Schnittes mithilfe der Fibonacci-Folge.

http://ais.informatik.uni-freiburg.de/teaching/ss09/info_MST/material/mst_08_recursion.pdf
Foliensatz zur Lehrveranstaltung *Einführung in die Informatik – Recursion* von Burgard und Stachniss an der Universität Freiburg; Beschreibung des *Faulen Ansatzes.* Implementierung rekursiver Funktionen mittels Java.

https://www.youtube.com/user/Computerphile/featured
Der englischsprachige Youtube-kanal *Computerphile*, wo verschiedene Forscher und Professoren Themen der Informatik erklären.
Empfehlenswerte Videos:
Programming Loops vs Recursion - Computerphile
What on Earth is Recursion? - Computerphile
Fibonacci Programming - Computerphile
The Most Difficult Program to Compute? - Computerphile

https://scratch-dach.info/wiki/Fraktale
Eine detaillierte Beschreibung und Umsetzung der Kochkurve in Scratch.

https://www.inf-schule.de/algorithmen/algorithmen/rekursion/
Seite von INF-SCHULE, welche eine detaillierte Beschreibung von Rekursion mit Beispielen und Übungen bereitstellt. Die rekursiven Fraktale können mit Python Turtle oder Logo implementiert werden.

https://pixabay.com/
Quellen aller im Informationszettel vorkommenden Bilder