Übungen am 8.11.2001

Übungen zu "Computergestütztes symbolisches Rechnen" im Wintersemester 2001/02, Institut für Informatik der Universität München

Bemerkung: Eingaben werden immer durch SHIFT−RETURNabgeschlossen.

Wir wollen Varietäten auch durch Anschauung näherkommen.

```
In[1]:= << Graphics`ImplicitPlot`
```

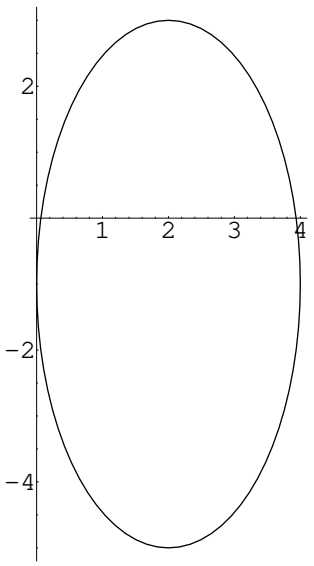
Dieser Befehl lädt erst das Grafik−Paketin *Mathematica*. Was das Paket leistet, erfährt man mittels

```
In[2]:= ? ImplicitPlot
```

```
ImplicitPlot[eqn, {x, a, b}] draws a graph of the set of points that satisfy the equation
   eqn.  The variable x is associated with the horizontal axis and ranges from a to
   b.  The remaining variable in the equation is associated with the vertical axis.
   ImplicitPlot[eqn, {x, a, x1, x2, ..., b}] allows the user to specify values of x where
   special care must be exercised. ImplicitPlot[{eqn1, eqn2, ...}, {x, a, b}] allows
   more than one equation to be plotted, with PlotStyles set as in the Plot function.
   ImplicitPlot[eqn, {x, a, b}, {y, a, b}] uses a contour plot method of generating
   the plot.  This form does not allow specification of intermediate points. More...
```

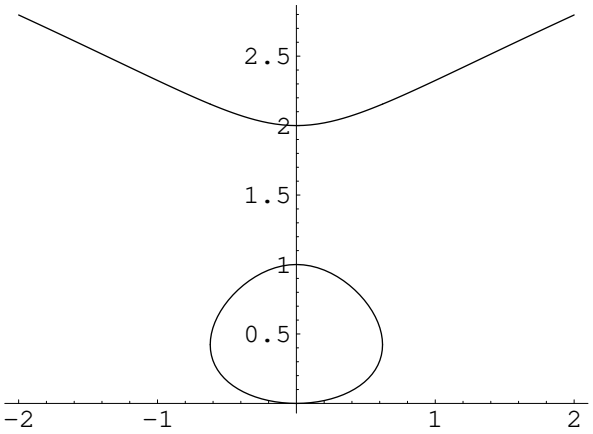
■ Cox et al, Seite 12, Aufgabe 1a:

```
In[3]:= ImplicitPlot[x^2 + 4 y^2 + 2 x - 16 y + 1 == 0, {y, 0, 4}];
```



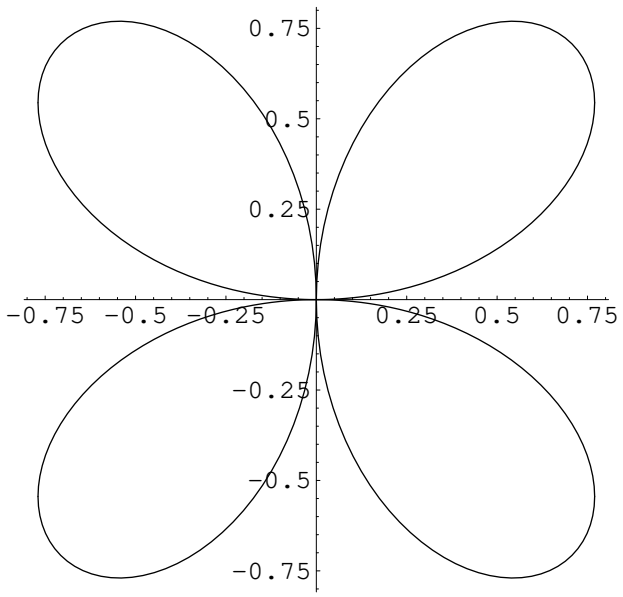
■ Cox et al, Seite 12, Aufgabe 2

```
In[4]:= ImplicitPlot[y^2 - x (x - 1) (x - 2) == 0, {y, -2, 2}];
```



■ Cox et al, Seite 12, Aufgabe 7

```
In[5]:= ImplicitPlot[(x^2 + y^2)^3 - 4 x^2 y^2 == 0, {x, -1, 1}];
```



In[6]:=
```
Roots[(x^2 + y^2)^3 - 4 x^2 y^2 == 0, x]
```

Out[6]=

$$x == \sqrt{-y^2 + \frac{2\,2^{2/3}\,y}{3^{1/3}\left(-9y + \sqrt{3}\,\sqrt{-16+27y^2}\right)^{1/3}} + \frac{2^{1/3}\left(-9y + \sqrt{3}\,\sqrt{-16+27y^2}\right)^{1/3}}{3^{2/3}}} \;||$$

$$x == -\sqrt{-y^2 + \frac{2\,2^{2/3}\,y}{3^{1/3}\left(-9y + \sqrt{3}\,\sqrt{-16+27y^2}\right)^{1/3}} + \frac{2^{1/3}\left(-9y + \sqrt{3}\,\sqrt{-16+27y^2}\right)^{1/3}}{3^{2/3}}} \;||$$

$$x == \sqrt{\left(-y^2 - \frac{2^{2/3}\,y}{3^{1/3}\left(-9y + \sqrt{3}\,\sqrt{-16+27y^2}\right)^{1/3}} - \frac{i\,2^{2/3}\,3^{1/6}\,y}{\left(-9y + \sqrt{3}\,\sqrt{-16+27y^2}\right)^{1/3}} + \frac{i\,y\left(-9y + \sqrt{3}\,\sqrt{-16+27y^2}\right)^{1/3}}{2^{2/3}\,3^{1/6}} - \frac{y\left(-9y + \sqrt{3}\,\sqrt{-16+27y^2}\right)^{1/3}}{6^{2/3}}\right)} \;||$$

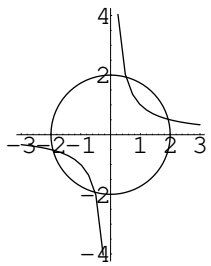
$$x == -\sqrt{\left(-y^2 - \frac{2^{2/3}\,y}{3^{1/3}\left(-9y + \sqrt{3}\,\sqrt{-16+27y^2}\right)^{1/3}} - \frac{i\,2^{2/3}\,3^{1/6}\,y}{\left(-9y + \sqrt{3}\,\sqrt{-16+27y^2}\right)^{1/3}} + \frac{i\,y\left(-9y + \sqrt{3}\,\sqrt{-16+27y^2}\right)^{1/3}}{2^{2/3}\,3^{1/6}} - \frac{y\left(-9y + \sqrt{3}\,\sqrt{-16+27y^2}\right)^{1/3}}{6^{2/3}}\right)} \;||$$

$$x == \sqrt{\left(-y^2 - \frac{2^{2/3}\,y}{3^{1/3}\left(-9y + \sqrt{3}\,\sqrt{-16+27y^2}\right)^{1/3}} + \frac{i\,2^{2/3}\,3^{1/6}\,y}{\left(-9y + \sqrt{3}\,\sqrt{-16+27y^2}\right)^{1/3}} - \frac{i\,y\left(-9y + \sqrt{3}\,\sqrt{-16+27y^2}\right)^{1/3}}{2^{2/3}\,3^{1/6}} - \frac{y\left(-9y + \sqrt{3}\,\sqrt{-16+27y^2}\right)^{1/3}}{6^{2/3}}\right)} \;||$$

$$x == -\sqrt{\left(-y^2 - \frac{2^{2/3}\,y}{3^{1/3}\left(-9y + \sqrt{3}\,\sqrt{-16+27y^2}\right)^{1/3}} + \frac{i\,2^{2/3}\,3^{1/6}\,y}{\left(-9y + \sqrt{3}\,\sqrt{-16+27y^2}\right)^{1/3}} - \frac{i\,y\left(-9y + \sqrt{3}\,\sqrt{-16+27y^2}\right)^{1/3}}{2^{2/3}\,3^{1/6}} - \frac{y\left(-9y + \sqrt{3}\,\sqrt{-16+27y^2}\right)^{1/3}}{6^{2/3}}\right)}$$

■ Cox et al, Seite 12, Aufgabe 3

In[7]:=
```
ImplicitPlot[{x^2 + y^2 - 4 == 0, x y - 1 == 0}, {x, -3, 3}];
```



■ Cox et al, Seite 12, Aufgabe 4e

In[8]:=
```
Reduce[{x^4 - z x == 0, x^3 - y x == 0}, x]
```

Out[8]=

$$x == \frac{y^2}{z} \;\&\&\; y^3 == z^2 \;||\; x == 0$$

```
In[9]:= Reduce[{x^4 - z x == 0, x^3 - y x == 0}, y]

Out[9]= y == x^2 && z == x^3 || x == 0
```

■ Gröbner-Basen

Zur Übung geben wir den Befehl FALSCH ein.

```
In[10]:= GroebnerBase[{x^2 + y^2 - 4, x y - 1}, {x, y}]

Out[10]= GroebnerBase[{-4 + x^2 + y^2, -1 + x y}, {x, y}]

In[11]:= ?Groebner*
```

System`

GroebnerBasis

Global`

GroebnerBase

```
In[12]:= Remove[GroebnerBase]

In[13]:= ?Groebner*

         GroebnerBasis[{poly1, poly2, ... }, {x1, x2, ... }] gives a list of
           polynomials that form a Groebner basis for the set of polynomials polyi. More...

In[14]:= GroebnerBasis[{x^2 + y^2 - 4, x y - 1}, {x, y}]

Out[14]= {1 - 4 y^2 + y^4, -x + 4 y - y^3}
```

Man beachte, daß diese Polynome dieselbe Varietät ergeben, allerdings das erste die Variable x nicht enthält, weshalb man nun leicht eine geschlossene Lösung findet.

```
In[15]:= reg = Solve[1 - 4 y^2 + y^4 == 0, y]

Out[15]= {{y → -√(2 - √3)}, {y → √(2 - √3)}, {y → -√(2 + √3)}, {y → √(2 + √3)}}
```

Dies ergibt Ersetzungsregeln für die 4 möglichen Lösungen. Wir können sie einfach einsetzen in den Ausdruck für x, den wir aus dem zweiten Polynom der Gröbner−Basis erhalten.

```
In[16]:= 4 y - y^3 /. reg

Out[16]= {-4 √(2 - √3) + (2 - √3)^(3/2), 4 √(2 - √3) - (2 - √3)^(3/2),
          -4 √(2 + √3) + (2 + √3)^(3/2), 4 √(2 + √3) - (2 + √3)^(3/2)}
```

Wir vereinfachen dieses Ergebnis, das wir mit % ansprechen.

```
In[17]:= Simplify[%]

Out[17]= {-√(2 - √3) (2 + √3), √(2 - √3) (2 + √3), (-2 + √3) √(2 + √3), -(-2 + √3) √(2 + √3)}
```

Natürlich bekommt man das auch direkt mit dem leistungsfähigen Solve−Befehl:

```
In[18]:= Solve[{x^2 + y^2 - 4 == 0, x y - 1 == 0}, {x, y}]
```

$$\text{Out[18]}= \left\{\left\{x \to -2\sqrt{2-\sqrt{3}} - \sqrt{3\left(2-\sqrt{3}\right)},\ y \to -\sqrt{2-\sqrt{3}}\right\},\ \left\{x \to 2\sqrt{2-\sqrt{3}} + \sqrt{3\left(2-\sqrt{3}\right)},\ y \to \sqrt{2-\sqrt{3}}\right\},\ \left\{x \to 2\sqrt{2+\sqrt{3}} - \sqrt{3\left(2+\sqrt{3}\right)},\ y \to \sqrt{2+\sqrt{3}}\right\},\ \left\{x \to -2\sqrt{2+\sqrt{3}} + \sqrt{3\left(2+\sqrt{3}\right)},\ y \to -\sqrt{2+\sqrt{3}}\right\}\right\}$$

```
In[19]:= Simplify[%]
```

$$\text{Out[19]}= \left\{\left\{x \to -\sqrt{2-\sqrt{3}}\ \left(2+\sqrt{3}\right),\ y \to -\sqrt{2-\sqrt{3}}\right\},\ \left\{x \to \sqrt{2-\sqrt{3}}\ \left(2+\sqrt{3}\right),\ y \to \sqrt{2-\sqrt{3}}\right\},\ \left\{x \to -\left(-2+\sqrt{3}\right)\sqrt{2+\sqrt{3}},\ y \to \sqrt{2+\sqrt{3}}\right\},\ \left\{x \to \left(-2+\sqrt{3}\right)\sqrt{2+\sqrt{3}},\ y \to -\sqrt{2+\sqrt{3}}\right\}\right\}$$

■ Polynome

```
In[20]:= ?*Polynomial*
```

System‘

CharacteristicPolynomial	PolynomialGCD	PolynomialQ	PolynomialRemainder
InterpolatingPolynomial	PolynomialLCM	PolynomialQuotient	Polynomials
PolynomialForm	PolynomialMod	PolynomialReduce	

```
PolynomialQ[expr, var] yields True if expr is
   a polynomial in var, and yields False otherwise. PolynomialQ[expr,
   {var1, ... }] tests whether expr is a polynomial in the vari. More...
```

```
In[21]:= PolynomialQ[x y - 5 x^3 / y, {x, y}]

Out[21]= False

In[22]:= PolynomialQ[x y - 5 x^3 / x, {x, y}]

Out[22]= True

In[23]:= PolynomialQ[Sin[x]^2 + Cos[x]^2, {x}]

Out[23]= False

In[24]:= PolynomialQ[Simplify[Sin[x]^2 + Cos[x]^2], {x}]

Out[24]= True

In[25]:= Simplify[Sin[x]^2 + Cos[x]^2]

Out[25]= 1
```

■ Dreidimensionales Plotten

```
In[26]:= ?*Plot*
```

System‘

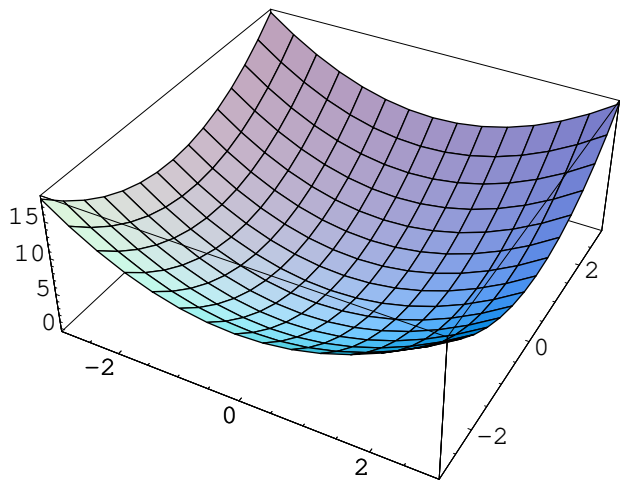
ContourPlot ListDensityPlot ParametricPlot Plot3D PlotJoined PlotRange
DensityPlot ListPlot ParametricPlot3D Plot3Matrix PlotLabel PlotRegion
ListContourPlot ListPlot3D Plot PlotDivision PlotPoints PlotStyle

Graphics‘ImplicitPlot‘

ImplicitPlot

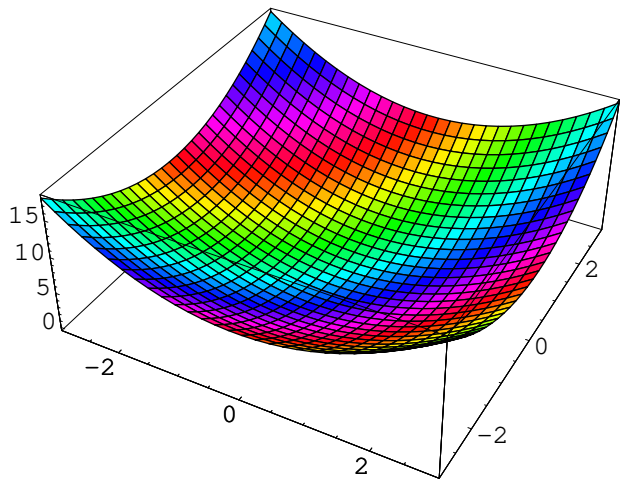
```
Plot3D[f, {x, xmin, xmax}, {y, ymin, ymax}] generates a three-dimensional
   plot of f as a function of x and y. Plot3D[{f, s}, {x, xmin, xmax}, {y,
   ymin, ymax}] generates a three-dimensional plot in which the height of
   the surface is specified by f, and the shading is specified by s. More...
```

```
In[27]:= Plot3D[x^2 + y^2, {x, -3, 3}, {y, -3, 3}];
```



Man beachte den Strichpunkt am Befehlsende, der einen hier uninteressanten Ergebniswert unterdrückt. Die Farbinformation kann darüber hinaus auch programmiert werden. Dazu wollen wir eine höhere Auflösung.

```
In[31]:= Plot3D[{x^2 + y^2, Hue[1 / 2 + (x - y) / 6]}, {x, -3, 3}, {y, -3, 3}, PlotPoints → 30];
```



■ Die Fibonacci−Zahlen

Wir wollen hier auf einige Probleme mit der Definition von Funktionen in *Mathematica* hinweisen. In der Tat sind es nur Probleme im Umgang mit den vielfältigen Möglichkeiten.

```
In[32]:= ? fib

         Information::notfound : Symbol fib not found.

In[33]:= fib[0] = 1

Out[33]= 1

In[34]:= fib[1] = 1

Out[34]= 1

In[35]:= ? fib

        Global`fib

         fib[0] = 1

         fib[1] = 1
```

Wir versuchen, einfach die Rekursionsgleichung als Definition hinzuschreiben.

```
In[36]:= fib[n] = fib[n - 1] + fib[n - 2]

Out[36]= fib[-2 + n] + fib[-1 + n]
```

Leider wird das nur für Argument n benutzt:

```
In[39]:= 6 + fib[2]

Out[39]= 6 + fib[2]

In[40]:= 6 + fib[n]

Out[40]= 6 + fib[-2 + n] + fib[-1 + n]
```

Ein neuer Versuch −nach Löschen des bisherigen Definitionen für fib mit als Patternvariable deklariertem n

```
In[41]:= Clear[fib]; fib[0] = 1; fib[1] = 1;
```

Folgende Eingabe verlangt nach Abbruch mit ALT−.(ALT−Tastelinks und . gleichzeitig drücken!)

```
In[42]:= fib[n_] = fib[n - 1] + fib[n - 2]

         $RecursionLimit::reclim : Recursion depth of 256 exceeded.

         $RecursionLimit::reclim : Recursion depth of 256 exceeded.

         $RecursionLimit::reclim : Recursion depth of 256 exceeded.

         General::stop : Further output of $RecursionLimit::reclim will be suppressed during this calculation.

Out[42]= $Aborted
```

Leider ergab das nur eine definitorische Endlosschleife. Wir dürfen nicht gleich weiterrechnen! Dafür gibt es die verzögerte Zuweisung.

```
In[43]:= Clear[fib]; fib[0] = 1; fib[1] = 1;

In[44]:= fib[n_] := fib[n - 1] + fib[n - 2]

In[45]:= ? fib

         Global`fib

         fib[0] = 1

         fib[1] = 1

         fib[n_] := fib[n - 1] + fib[n - 2]

In[46]:= Table [fib[i], {i, 0, 10}]

Out[46]= {1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89}
```

Nur haben wir keine Einschränkung an den Definitionsbereich angegeben.

```
In[47]:= fib[-1]

         $RecursionLimit::reclim : Recursion depth of 256 exceeded.

         $RecursionLimit::reclim : Recursion depth of 256 exceeded.

         $RecursionLimit::reclim : Recursion depth of 256 exceeded.

         General::stop : Further output of $RecursionLimit::reclim will be suppressed during this calculation.

Out[47]= $Aborted

In[48]:= fib[7 / 2]

         $RecursionLimit::reclim : Recursion depth of 256 exceeded.

         $RecursionLimit::reclim : Recursion depth of 256 exceeded.

         $RecursionLimit::reclim : Recursion depth of 256 exceeded.

         General::stop : Further output of $RecursionLimit::reclim will be suppressed during this calculation.

Out[48]= $Aborted

In[49]:= Clear[fib]; fib[0] = 1; fib[1] = 1;

In[50]:= fib[n_] := fib[n - 1] + fib[n - 2] /; n ≥ 2

In[51]:= fib[-1]

Out[51]= fib[-1]

In[52]:= fib[7 / 2]

Out[52]= fib[1/2] + 2 fib[3/2]
```

Um auch das letzte Problem zu beheben, schränken wir den Bereich für n weiter ein.

```
In[53]:= Clear[fib]; fib[0] = 1; fib[1] = 1;

In[54]:= fib[n_Integer] := fib[n - 1] + fib[n - 2] /; n ≥ 2
```

```
In[55]:= fib[7 / 2]
```

```
Out[55]= fib[7/2]
```

Leider ist dieses Programm äußerst ineffizient, weil alle Teilberechnungen immer wieder ausgeführt werden. So will man nicht auf das Ergebnis im folgenden Fall warten.

```
In[56]:= fib[30]
```

```
Out[56]= $Aborted
```

Die Idee ist nun, die schon berechneten Werte als Funktionswerte abzuspeichern.

```
In[57]:= Clear[fib]; fib[0] = 1; fib[1] = 1;
```

```
In[58]:= fib[n_Integer] := (fib[n] = fib[n - 1] + fib[n - 2]) /; n ≥ 2
```

```
In[59]:= ? fib
```

```
Global`fib

fib[0] = 1

fib[1] = 1

fib[n_Integer] := (fib[n] = fib[n - 1] + fib[n - 2]) /; n ≥ 2
```

```
In[60]:= fib[4]
```

```
Out[60]= 5
```

```
In[61]:= ? fib
```

```
Global`fib

fib[0] = 1

fib[1] = 1

fib[2] = 2

fib[3] = 3

fib[4] = 5

fib[n_Integer] := (fib[n] = fib[n - 1] + fib[n - 2]) /; n ≥ 2
```

```
In[62]:= Table[fib[i], {i, 0, 100}]
```

```
Out[62]= {1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181,
          6765, 10946, 17711, 28657, 46368, 75025, 121393, 196418, 317811, 514229,
          832040, 1346269, 2178309, 3524578, 5702887, 9227465, 14930352, 24157817,
          39088169, 63245986, 102334155, 165580141, 267914296, 433494437, 701408733,
          1134903170, 1836311903, 2971215073, 4807526976, 7778742049, 12586269025,
          20365011074, 32951280099, 53316291173, 86267571272, 139583862445,
          225851433717, 365435296162, 591286729879, 956722026041, 1548008755920,
          2504730781961, 4052739537881, 6557470319842, 10610209857723, 17167680177565,
          27777890035288, 44945570212853, 72723460248141, 117669030460994,
          190392490709135, 308061521170129, 498454011879264, 806515533049393,
          1304969544928657, 2111485077978050, 3416454622906707, 5527939700884757,
          8944394323791464, 14472334024676221, 23416728348467685, 37889062373143906,
          61305790721611591, 99194853094755497, 160500643816367088, 259695496911122585,
          420196140727489673, 679891637638612258, 1100087778366101931, 1779979416004714189,
          2880067194370816120, 4660046610375530309, 7540113804746346429,
          12200160415121876738, 19740274219868223167, 31940434634990099905,
          51680708854858323072, 83621143489848422977, 135301852344706746049,
          218922995834555169026, 354224848179261915075, 573147844013817084101}
```

Leider darf keine Rekursion mehr als 256 übereinanderliegende offene Rechnungen erreichen:

```
In[63]:= fib[355]

        $RecursionLimit::reclim : Recursion depth of 256 exceeded.

Out[63]= 62040070130393226937787020049005111782811456728454536879080533939929867845+
          5412222237103765877667657957123376148335120669380949 7Hold[RuleCondition[
             $ConditionHold[$ConditionHold[fib[101] = fib[101 - 1] + fib[101 - 2]]], 101 ≥ 2]]
```

Leider verdirbt das die weiteren Rechnungen:

```
In[64]:= fib[102]

Out[64]= 573147844013817084101 + Hold[RuleCondition[
            $ConditionHold[$ConditionHold[fib[101] = fib[101 - 1] + fib[101 - 2]]], 101 ≥ 2]]
```

Dennoch können wir problemlos auch bis zum Index 10000 tabellieren, weil ja dann die Zwischenergebnisse schon berechnet sind. Wir wollen die Tabelle aber lieber nicht ausgeben...

```
In[65]:= Clear[fib]; fib[0] = 1; fib[1] = 1;
         fib[n_Integer] := (fib[n] = fib[n - 1] + fib[n - 2]) /; n ≥ 2;
         Table[fib[i], {i, 0, 10000}];

In[66]:= fib[10000]

Out[66]= 5443837311356528133873426099375038013538918455469596702624771584120858286562234905
          1708305154793896054117382267597802631738435958475111624143917470264295916992558605
          3341179060630480897935314761084662590727593678991506779600883065979666419658249305
          7721800381441158841042480997984696487375337180028163763317781927941101369262750905
          7950980071359671802381471066991264421477525447858767456896380800296226513311135905
          9297627266794414001015758000435107774659358053625024617079180592264146790056907505
          2321895868142367849593880756423483754386342639635970733756260098962462668746112005
          4173981940487506244370986865431562684718619562014612664223271181504036701882520505
          3148458758171935335298278378003519025292395178366894676619179538847124410284639305
          5449484614450778762529520961887597272889220768537396475869543159172434537193611205
          6374392633731300589616724805173798630636811500308839674958710261952463135244749905
          5052041983051871683216232838597946272459197714546282183996957892237989121994317705
          5469705216131081096559950638297261253848242007897109054754028438149611930465061805
          6617012298328896435273375079278606944476185352514442107792804597990456129812942305
          8091560550330323389196091622366987599227829231918966880177185755555209946533201205
          8446502371153715141749290913104897203455577507196645425232862022019506091483585205
          2388271101670843305116994211577515125551025165593188816404834412955703882547752105
          1115773957801158683970726025656148249564605387002803313118614853998053970315557205
          7529693399586079850381581446276433858828529535803424850845426446471681531001533105
          8047956743639681565332615250957112748041192819602214884914828438912417852017450705
          3055389287178579235094177433833315068982393544219888054293324403711948672155435705
          6548565499134519271098919802665184564927827827212957649240235507595558205647569305
          6539487331765900020637312657064350970948264971003873351747771340331902810557566705
          9317894700241188030946040343629534719974613922747915497303564126330742308240519905
          9996101549784667340458326852960388301120765629245998136251652347093963049734046405
          4510636530416363082366924225776146828846179184322479343440607991788336067684671105
          185597501
```