Wahrscheinlichkeit eines *Run* beim Münzwurf

Ein *Run* der Länge m beim n-maligen Werfen mit einer Münze ($n \geq m$) soll heißen, dass genau m-mal hintereinander Wappen bzw. Zahl geworfen wird. Gesucht wird die Wahrscheinlichkeit für mindestens einen *Run* vorgegebener Länge. Auf Grund der Symmetrie des Zufallsexperimentes kann auf die Kenntnis verzichtet werden, ob zuerst Wappen oder Zahl geworfen wurde; der n-malige Münzwurf kann dann allein durch wechselseitige Angabe der Anzahlen von Wappen und Zahl charakterisiert werden. Zum Beispiel bedeutet für $n = 6$ die Zahlenfolge $(2; 1; 1; 2)$ entweder WWZWZZ oder ZZWZWW. Bis auf den hieraus resultierenden Faktor zwei können daher alle n-maligen Münzwürfe durch die Zerlegungen von n in Summanden dargestellt werden. Für das Beispiel $n = 6$ sind das die Summen:

$1+1+1+1+1+1$	[1]
$2+1+1+1+1$	[5]
$2+2+1+1$	[6]
$2+2+2$	[1]
$3+1+1+1$	[4]
$3+2+1$	[6]
$3+3$	[1]
$4+1+1$	[3]
$4+2$	[2]
$5+1$	[2]
6	[1]

In eckigen Klammern sind die Vielfachheiten für die möglichen Umstellungen der Summanden angegeben. Zusammen mit dem unterschlagenen Faktor 2 ergeben sich tatsächlich $2 \cdot (1+5+6+1+4+6+1+3+2+2+1) = 2 \cdot 32 = 64 = 2^6$ Möglichkeiten.

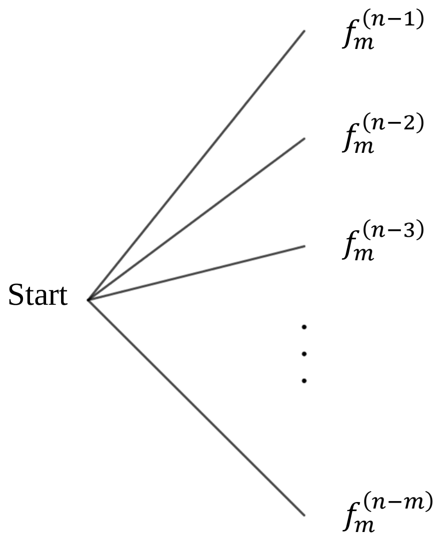
Weiter kann obiger Aufstellung die Wahrscheinlichkeit für einen *Run* mindestens der Länge 4 beim 6-maligen Werfen entnommen werden:

$$P = \frac{2 \cdot (3+2+2+1)}{2^6} = 25\ \%$$

Erstaunlicherweise können die Anzahlen verschiedener Summen (*unter Berücksichtigung der Reihenfolge der Summanden*) mit beschränkten Summanden rekursiv bestimmt werden.[1] Der Autor Holger Stephan der zitierten Datei gebraucht das Bild einer n-stufigen Treppe, die in mehreren Schritten zu ein, zwei, drei etc. Stufen auf einmal gestiegen wird. Die Anzahl der Stufen, die maximal auf einmal genommen werden dürfen, werde mit m bezeichnet. Die Anzahl der Möglichkeiten, die n-stufige Treppe mit dieser Beschränkung zu steigen, werde mit $f_m^{(n)}$

[1] Vgl. https://www.wias-berlin.de/people/stephan/Kombi.pdf – Aufgabe 13

bezeichnet. Dann ergibt sich je nach der Weite des ersten Schrittes folgender Baum für die weiteren möglichen Anzahlen:



Es muss daher gelten:

$$f_m^{(n)} = f_m^{(n-1)} + f_m^{(n-2)} + \cdots + f_m^{(n-m)} = \sum_{k=1}^{m} f_m^{(n-k)}$$

In obiger Formel ist für $k \geq n$ der Summand $f_m^{(n-k)}$ gleich null zu setzen. Wie nachfolgend dargelegt wird, gilt weiter für $n \leq m$ die Beziehung $f_m^{(n)} = 2^{n-1}$. Um dies zu zeigen, sind die Folgen von Wappen und Zahl geeigneter. Da im vorliegenden Falle die obere Schranke m der Summanden größer als die Gesamtanzahl n der Würfe ist, werden durch die Summen stets alle Folgen von Wappen und Zahl – bis auf den oben angesprochenen Faktor zwei – realisiert. Somit folgt wie behauptet $f_m^{(n)} = \frac{1}{2} \cdot 2^n = 2^{n-1}$.

Für das Beispiel mit $n = 6$ ergibt sich $f_1^{(1)} = 2^{1-1} = 1$ und $f_1^{(n)} = \sum_{k=1}^{1} f_1^{(n-k)} = f_1^{(n-1)}$, also rekursiv $f_1^{(n)} = 1$ für alle $n \in \mathbb{N}$. Weiter sind $f_2^{(1)} = 2^{1-1} = 1$ und $f_2^{(2)} = 2^{2-1} = 2$. Rekursiv folgt $f_2^{(n)} = \sum_{k=1}^{2} f_2^{(n-k)} = f_2^{(n-1)} + f_2^{(n-2)}$, woraus sich gerade die Fibonaccifolge $1; 2; 3; 5; 8; 13; \cdots$ ergibt. Nach analogem Muster errechnet sich die Folge $1; 2; 4; 7; 13; 24; \cdots$ für $f_3^{(n)}$ und alle anderen Werte nachstehender Tabelle:

$f_1^{(6)} = 1$

$f_2^{(6)} = 13$

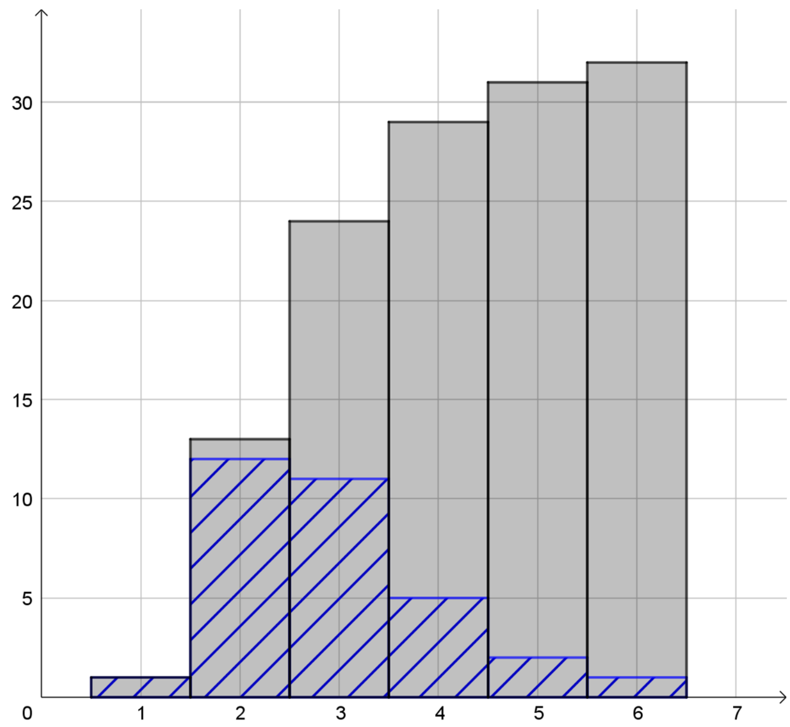
$f_3^{(6)} = 24$

$f_4^{(6)} = 29$

$f_5^{(6)} = 31$

$f_6^{(6)} = 32$

Ein Abgleich mit den eingangs aufgeführten Summenzerlegungen zeigt die Richtigkeit der berechneten Werte. Ferner kann $f_m^{(n)}$ offensichtlich als Verteilungsfunktion interpretiert werden.



Abschließend soll die Wahrscheinlichkeit für einen *Run* mindestens der Länge 6 beim 50-maligen Münzwurf bestimmt werden, die diesen Aufsatz ursprünglich motiviert hat. Rechnet man mit dem Gegenereignis, so ist die gesuchte Wahrscheinlichkeit

$$P = 1 - \frac{2 \cdot f_5^{(50)}}{2^{50}}$$

Da das Rechnen von Hand für derart große Zahlen reichlich mühevoll ist, habe ich folgendes Programm in C aufgesetzt:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define NUMMER 50
#define SUMMAND_MAX 5

main()
{
	double liste[NUMMER];

	int i, j;
	int k = SUMMAND_MAX;


	for( i = 0; i < NUMMER; i++ )
	{
		liste[i] = 0.0;
	}


	for( i = 0; i < k; i++ )
	{
		liste[i] = pow( 2.0, i );
	}

	for( i = k; i < NUMMER; i++ )
```

```
	{
		for( j = 0; j < k; j++ )
		{
			liste[i] = liste[i] + liste[i - 1 - j];
		}
	}

	printf("%.0f\n", liste[NUMMER - 1]);

	return 0;
}
```

Seine Ausgabe lautet mit den eingangs vorgegebenen Parametern: `256641310658978`

Setzt man dies in obige Formel ein, so erhält man den erstaunlich großen Wert:

$P \approx 54{,}4\ \%$

Wenn eine Schulklasse also brav daheim eine Münze 50-mal geworfen hat, dann sollte etwa die Hälfte einen *Run* von mindestens der Länge 6 in ihren Aufzeichnungen haben…

Anmerkung:

Der wesentliche Zusammenhang von Zeichenfolgen Wappen-Zahl des Münzwurfes und Zerlegungen der Wurfanzahl in Summen kann formaler gefasst werden. Hierzu erkläre man auf der Menge $\mathcal{M}_n = \{W, Z\}^n$ aller Zeichenfolgen der Länge n eine Äquivalenzrelation; zwei Zeichenfolgen x und y seien äquivalent, wenn $x = y$ oder $x = \bar{y}$, worin $\bar{y}$ aus y durch Austausch von W und Z entstehe. Hierdurch zerfällt $\mathcal{M}_n$ in 2^{n-1} Äquivalenzklassen, auf denen eine injektive Abbildung φ in die Menge endlicher Folgen natürlicher Zahlen $\mathcal{N} = \bigcup_{k=1}^{\infty} \mathbb{N}^k$ nach obigem Muster erklärt werden kann. Zur Formalisierung dieser Konstruktion seien für ein $x = (x_1|x_2|\cdots|x_n) \in \mathcal{M}_n$ die natürlichen Zahlen $0 = j_0 < j_1 < j_2 < \cdots < j_t = n$ so gewählt, dass

- *für $1 \leq k < t$ gilt: $x_i = x_{j_k}$ für alle $j_{k-1} + 1 \leq i \leq j_k$ und $x_{j_k+1} \neq x_{j_k}$*
- *$x_i = x_n$ für alle $j_{t-1} + 1 \leq i \leq n$.*

Dann ergibt das t-Tupel $(j_1 - j_0|j_2 - j_1|\cdots|j_t - j_{t-1}) \in \mathcal{N}$ das gesuchte Bild unter φ; es bliebe noch die Injektivität von φ zu zeigen. Offensichtlich ergibt die Summe der Einträge des angegebenen t-Tupels $j_t - j_0 = n$. Betrachtet man jetzt noch die Menge $\mathcal{N}_n$ aller t-Tupel in $\mathcal{N}$ für $t \geq 1$, für welche die Summe ihrer Einträge den Wert n hat, so bliebe zu zeigen, dass φ surjektiv auf $\mathcal{N}_n$ abbildet. Auf diese Weise ließen sich die obigen Ausführungen strenger fassen.

Einen spannenden anderen Zugang zu den Werten von $f_m^{(n)}$ bieten die Differenzengleichungen.[2] Zunächst erklärt man auf der Menge $\mathcal{F}$ aller Folgen $\mathbb{N} \longrightarrow \mathbb{R}$ eine (*komponentenweise*) Addition $+ : \mathcal{F} \times \mathcal{F} \longrightarrow \mathcal{F}$, $(f+g)(j) = f(j) + g(j)$ und eine (*skalare*) Multiplikation $\cdot : \mathbb{R} \times \mathcal{F} \longrightarrow \mathcal{F}$, $(\lambda \cdot f)(j) = \lambda \cdot f(j)$ für alle $j \in \mathbb{N}$. Hierdurch wird $\mathcal{F}$ zu einem $\mathbb{R}$-Vektorraum. Weiter erklärt man (*für eine komfortablere Beschreibung rekursiver Beziehungen der Glieder einer Folge*) ein äußeres Produkt $* : \mathbb{R}[s] \times \mathcal{F} \longrightarrow \mathcal{F}$, $\left(\left(\sum_{i=0}^{n} a_i s^i\right) * f\right)(j) = \sum_{i=0}^{n} a_i \cdot \left(s^i * f\right)(j) =$

[2] Vgl. https://www.uibk.ac.at/mathematik/personal/pauer/differenzengleichungen_pauer_2009_endversion.pdf

$\sum_{i=0}^{n} a_i \cdot f(j+i)$ mit einem Polynom in s. Anschaulich gesprochen bewirkt die äußere Multiplikation mit s^i einen *Shift* der Folge um i Stellen nach links, wobei die ursprünglichen Glieder $f(0), f(1), \cdots, f(i-1)$ ersatzlos verloren gehen. Für die Fibonacci-Folge gilt beispielsweise die rekursive Vorschrift $f(n) = f(n-1) + f(n-2)$, die sich nach einer einfachen Substitution auch in der Form $f(j+2) = f(j+1) + f(j) \quad \Leftrightarrow \quad (s^2 * f)(j) = (s^1 * f)(j) + (s^0 * f)(j) \quad \Leftrightarrow \quad \big((s^2 - s - 1) * f\big)(j) = 0$ für alle $j \in \mathbb{N}$ schreiben lässt. Mit der Nullfolge $\sigma : \mathbb{N} \longrightarrow \mathbb{R}, \sigma(j) = 0$ für alle $j \in \mathbb{N}$ kann man noch kürzer und suggestiver $(s^2 - s - 1) * f = \sigma$ schreiben. Offensichtlich gilt $p * \sigma = \sigma$ für alle $p \in \mathbb{R}[s]$. Zusammen mit den nachfolgenden Rechenregeln kann man daher für eine gegebene (*homogene*) Differenzengleichung $p * f = \sigma$ den Algorithmus der Polynomdivision gewinnbringend zur Berechnung der Folgenglieder einsetzen.

Rechenregeln:

(i) $(p + q) * f = (p * f) + (q * f)$
(ii) $(p \cdot q) * f = p * (q * f)$

Beweis:

Die erste Rechenregel ist einfach zu zeigen und bereits in der Definition der äußeren Multiplikation implizit verwendet worden.

Seien nun $p = \sum_{\alpha=0}^{m} a_\alpha s^\alpha$ und $q = \sum_{\beta=0}^{n} b_\beta s^\beta$. Dann ist $p \cdot q = \sum_{k=0}^{m+n} c_k s^k$ mit $c_k = \sum_{\alpha+\beta=k} a_\alpha b_\beta$ und $\big((p \cdot q) * f\big)(j) = \sum_{k=0}^{m+n} c_k \cdot f(j+k)$. Weiter sei $g(j) = (q * f)(j) = \sum_{\beta=0}^{n} b_\beta \cdot f(j+\beta)$. Dann ist $\big(p * (q * f)\big)(j) = (p * g)(j) = \sum_{\alpha=0}^{m} a_\alpha \cdot g(j+\alpha) = \sum_{\alpha=0}^{m} a_\alpha \cdot (q * f)(j+\alpha) = \sum_{\alpha=0}^{m} a_\alpha \cdot \sum_{\beta=0}^{n} b_\beta \cdot f\big((j+\alpha)+\beta\big) = \sum_{\alpha=0}^{m} \sum_{\beta=0}^{n} a_\alpha \cdot b_\beta \cdot f\big(j + (\alpha+\beta)\big)$. Die Doppelsumme lässt sich ersetzen durch eine einfache Summation über $k = \alpha + \beta$ von der unteren Grenze null bis zur oberen Grenze $m+n$ und es folgt $\sum_{\alpha=0}^{m} \sum_{\beta=0}^{n} a_\alpha \cdot b_\beta \cdot f\big(j + (\alpha+\beta)\big) = \sum_{k=0}^{m+n} \big(\sum_{\alpha+\beta=k} a_\alpha \cdot b_\beta\big) \cdot f(j+k)$. Mit den oben definierten Koeffizienten c_k folgt $\big(p * (q * f)\big)(j) = \sum_{k=0}^{m+n} c_k \cdot f(j+k) = \big((p \cdot q) * f\big)(j)$ für alle $j \in \mathbb{N}$ und abschließend $p * (q * f) = (p \cdot q) * f$. □

Es gelte jetzt die Rekursionsformel $f(j+n) = c_{n-1} \cdot f(j+n-1) + \cdots + c_0 \cdot f(j) = \sum_{k=0}^{n-1} c_k \cdot f(j+k) = \Big(\big(\sum_{k=0}^{n-1} c_k\, s^k\big) * f\Big)(j)$ und es seien Anfangswerte $f(0) = a_0, \cdots, f(n-1) = a_{n-1}$ gegeben. Wegen $f(j+n) = (s^n * f)(j)$ und Rechenregel (i) folgt sodann $\Big(\big(s^n - \sum_{k=0}^{n-1} c_k\, s^k\big) * f\Big)(j) = 0$ für alle $j \in \mathbb{N}$. Das voranstehende Polynom vom Grad n heiße p und wir können kürzer schreiben $p * f = \sigma$. Zur Berechnung von Gliedern $f(j)$ mit $j > n$ gebraucht man wegen $f(j) = \big(s^j * f\big)(0)$ eine Polynomdivision $s^j = q \cdot p + r$ mit einem Polynom r vom Grade $deg(r) < n$. Mit den Rechenregeln (i) und (ii) folgt dann $f(j) = \big(s^j * f\big)(0) = \big((q \cdot p + r) * f\big)(0) = \big((q \cdot p) * f\big)(0) + (r * f)(0) = \big(q * (p * f)\big)(0) + (r * f)(0)$. Im ersten Summanden ist $p * f = \sigma$ und in der Folge auch $q * \sigma = \sigma$, also ist insbesondere $\big(q * (p * f)\big)(0) = 0$. Daher ist $f(j) = (r * f)(0) = \sum_{l=0}^{n-1} d_l \cdot f(l) = \sum_{l=0}^{n-1} d_l \cdot a_l$ und es genügt für die Berechnung von $f(j)$ das Restpolynom r zu kennen.

Auf eine gesonderte Anpassung an den vorliegenden Fall einer Folge, die erst mit $f(1)$ startet wurde bewusst verzichtet. Zur Berechnung von $f_m^{(n)}$ ist $f_m^{(0)}$ stets so zu wählen, dass die Rekursionsformel $2^{m-1} = f_m^{(m)} = f_m^{(m-1)} + f_m^{(m-2)} + \cdots + f_m^{(m-m)} = 2^{m-2} + 2^{m-3} + \cdots + 1 + f_m^{(0)} \quad \Leftrightarrow \quad f_m^{(0)} = 1$ erfüllt bleibt.

Beispiel:

Es soll noch einmal $f_5^{(50)} = f_5(50) = (s^{50} * f_5)(0)$ berechnet werden. Die Rekursionsformel $f_5^{(j+5)} = f_5^{(j+4)} + f_5^{(j+3)} + \cdots + f_5^{(j)}$ führt auf das Polynom $p = s^5 - s^4 - s^3 - s^2 - s - 1$ mit $p * f_5 = \sigma$. Mit *geogebra* erhält man in der CAS-Ansicht und dem Befehl *Division* den Polynomrest

$$r = 17180596958496x^4 + 16595567336576x^3 + 15445429382977x^2 + 13184317701200x + 8739089177552.$$

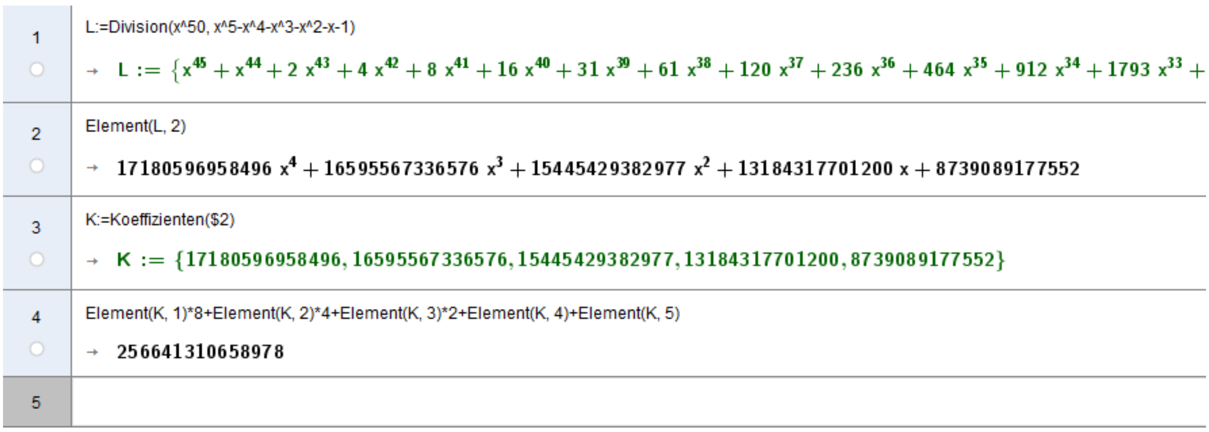
Daher ist der gesuchte Wert gleich

$$17180596958496 \cdot f(4) + 16595567336576 \cdot f(3) + 15445429382977 \cdot f(2) + 13184317701200 \cdot f(1) + 8739089177552 \cdot f(0) =$$

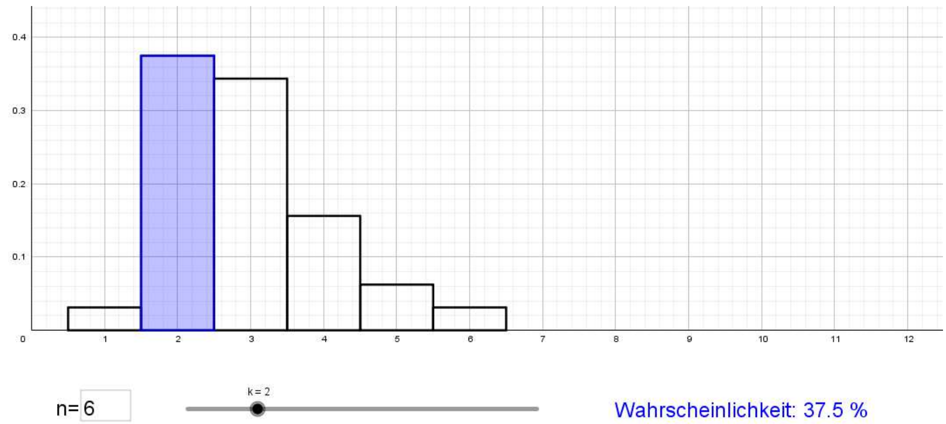
$$17180596958496 \cdot 8 + 16595567336576 \cdot 4 + 15445429382977 \cdot 2 + 13184317701200 \cdot 1 + 8739089177552 \cdot 1 =$$

$$256641310658978.$$

Der Wert dieses neuen Zuganges ist vor allem darin zu sehen, dass mit relativ leichten Mitteln in *geogebra* gearbeitet werden kann.



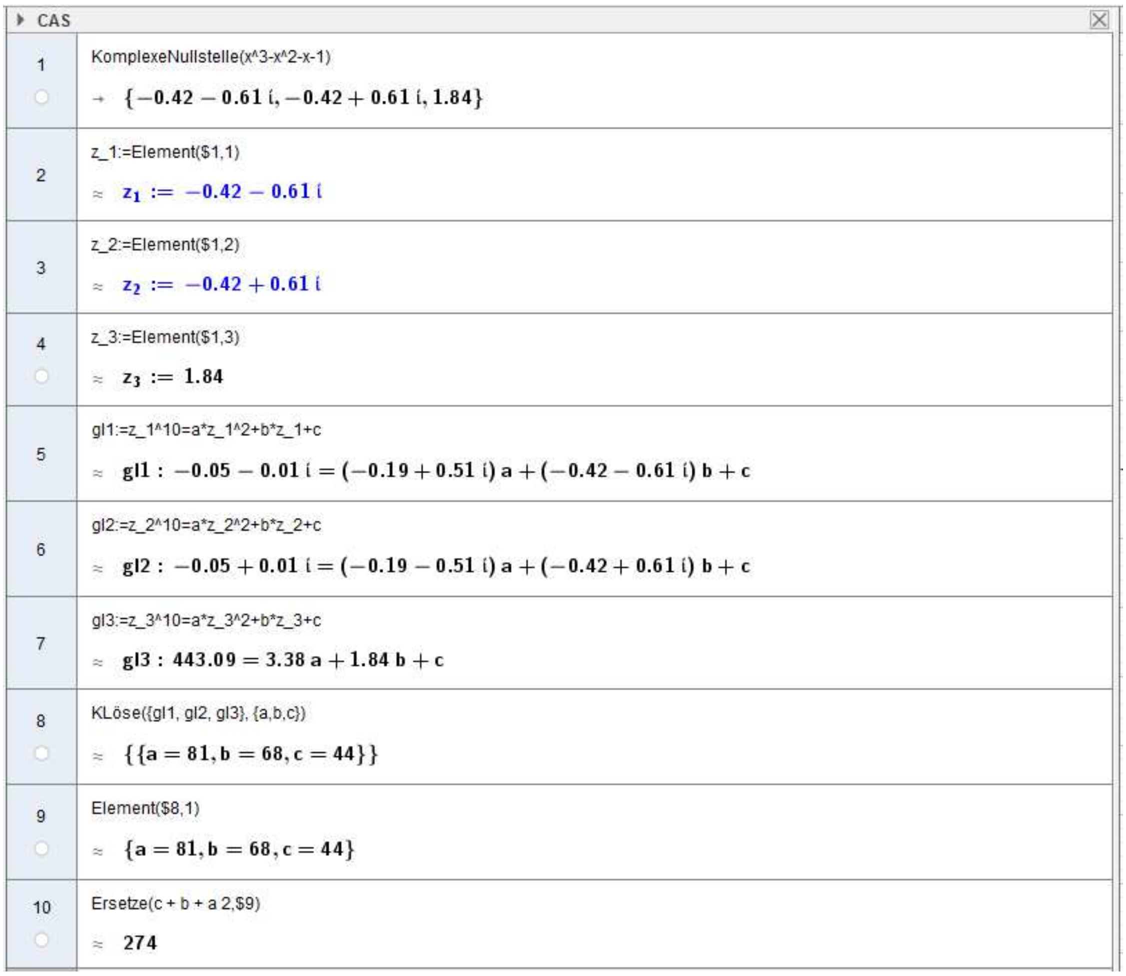
Auf Basis dieser Idee ist dann auch die *geogebra*-Anwendung „*Berechnung der Wkt-Verteilung in Abhängigkeit von n*“ programmiert worden, der nachfolgender Screenshot entstammt:



Im oben zitierten Skript von Franz Pauer wird noch eine weitere Idee ausgeführt, wie die Nullstellen $z_1, z_2, \cdots, z_m$ von p zur Berechnung von $f_m^{(n)}$ benutzt werden können. Hierzu werden diese in die Gleichung $s^n = q \cdot p + r$ eingesetzt und das resultierende lineare Gleichungssystem in den Koeffizienten $r_0^{(n)}, r_1^{(n)}, \cdots, r_{m-1}^{(n)}$ des Restpolynoms wird gelöst. Es ist dann $f_m^{(n)} = r_0^{(n)} + \sum_{k=1}^{m-1} r_k^{(n)} \cdot 2^{(k-1)}$. Für $n = 2$ erhält man so beispielsweise die *Formel von Binet* für die Fibonacci-Folge. In *geogebra* erschließt sich auf diese Weise ein einfacher Zugang zur Entwicklung der Werte $f_m^{(n)}$ mit wachsendem n.

Beispiel:

Die Berechnung von $f_3^{(10)}$ kann in CAS wie folgt vorgenommen werden:

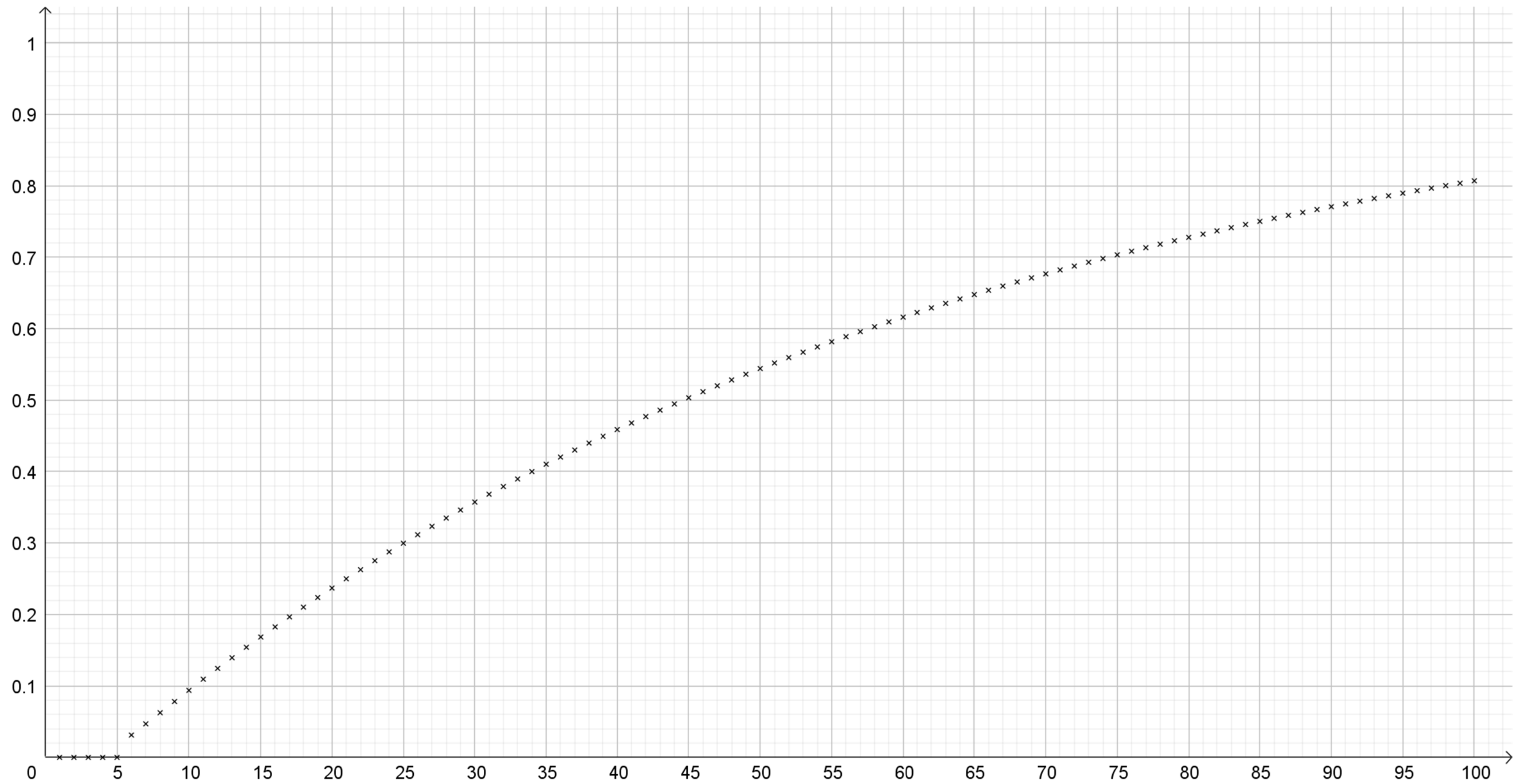


Für den Befehl *Nullstelle(<Polynom>)* ist gesondert zu beachten, dass eine Liste mit Einträgen „$x = (\cdots)$“ erzeugt wird. Für die Konvertierung in Nullstellenwerte z_i muss dann die umständlichere Syntax „$z_i \coloneqq Ersetze(x, \{Element(\cdots, \cdots)\})$“ verwendet werden.

Werden die Gleichungen mit der Unbekannten n angesetzt, so kann auch die Folge $f_3^{(n)}$ berechnet werden:

CAS	
1	KomplexeNullstelle(x^3-x^2-x-1) → $\{-0.42 - 0.61\,i, -0.42 + 0.61\,i, 1.84\}$
2	z_1:=Element($1,1) ≈ $z_1 := -0.42 - 0.61\,i$
3	z_2:=Element($1,2) ≈ $z_2 := -0.42 + 0.61\,i$
4	z_3:=Element($1,3) ≈ $z_3 := 1.84$
5	G1:=z_1^n=a*z_1^2+b*z_1+c ≈ $G1: e^{(-0.3-2.18i)n} = (-0.19 + 0.51\,i)\,a + (-0.42 - 0.61\,i)\,b + c$
6	G2:=z_2^n=a*z_2^2+b*z_2+c ≈ $G2: e^{(-0.3+2.18i)n} = (-0.19 - 0.51\,i)\,a + (-0.42 + 0.61\,i)\,b + c$
7	G3:=z_3^n=a*z_3^2+b*z_3+c ≈ $G3: e^{0.61n} = 3.38\,a + 1.84\,b + c$
8	KLöse({G1, G2, G3}, {a,b,c}) ≈ $\{\{a = (-0.09 + 0.34\,i)\,e^{(-0.3+2.18i)n} + (-0.09 - 0.34\,i)\,e^{(-0.3-2.18i)n} + 0.18\,e^{0.61n}, b = (-0.08 - 0.54\,i)\,e^{(-0.3+2.18i)n} + (-0.08 + 0.54\,i)\,e^{(-0.3-2.}$
9	Element($8,1) ≈ $\{a = (-0.09 + 0.34\,i)\,e^{(-0.3+2.18i)n} + (-0.09 - 0.34\,i)\,e^{(-0.3-2.18i)n} + 0.18\,e^{0.61n}, b = (-0.08 - 0.54\,i)\,e^{(-0.3+2.18i)n} + (-0.08 + 0.54\,i)\,e^{(-0.3-2.18}$
10	f_n:=Element($9,1) 2 + Element($9,2) + Element($9,3) ≈ $f_n: 2\,a + b + c = (0.19 - 0.02\,i)\,e^{(-0.3+2.18i)n} + (0.19 + 0.02\,i)\,e^{(-0.3-2.18i)n} + 0.62\,e^{0.61n}$
11	Folge(Ersetze($10,{n = k}),k,1,10,1) ≈ $\{2a + b + c = 1, 2a + b + c = 2, 2a + b + c = 4, 2a + b + c = 7, 2a + b + c = 13, 2a + b + c = 24, 2a + b + c = 44, 2a + b + c = 81, 2a +$
12	L:=Folge(Ersetze(2a + b + c,Element($11,k)),k,1,10,1) ≈ $L := \{1, 2, 4, 7, 13, 24, 44, 81, 149, 274\}$

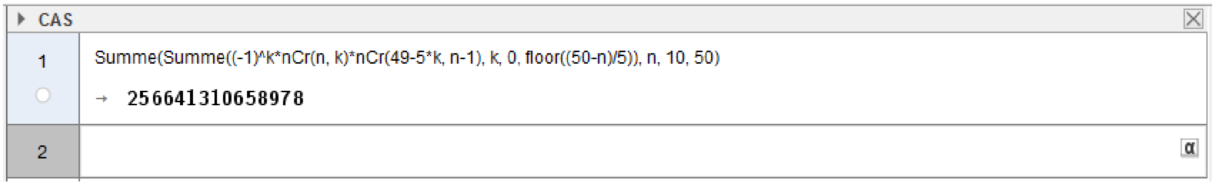
Bedauerlicherweise lassen sich die Gleichungen nicht dynamisch in Abhängigkeit von m erzeugen, sodass die Berechnung stets exemplarisch erfolgen muss. Zu guter Letzt noch ein Rückgriff auf das Eingangsrätsel von einem *Run* mindestens der Länge sechs. Zur Abschätzung seiner Wahrscheinlichkeit in Abhängigkeit von der Anzahl der Münzwürfe benötigt man die Folge $f_5^{(n)}$ beziehungsweise die Werte $P_n = 1 - \frac{2 \cdot f_5^{(n)}}{2^n}$. Der Graph auf der nächsten Seite zeigt die entsprechenden Werte und ihm kann entnommen werden, dass die fragliche Wahrscheinlichkeit bereits für $n = 45$ den Wert von 50 % übersteigt.


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
5
10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100

Aus Überlegungen zu den Augensummen beim Würfeln ergibt sich noch eine weitere Formel zur Berechnung von $f_5^{(50)}$. Die *Augensumme* ist dann 50 und die Schranke für die Summanden ist fünf. Offenkundig müssen dann mindestens zehn und dürfen höchstens 50 (*fünfseitige*) Würfel geworfen werden. Dies ergibt die nachstehende Doppelsumme:

$$\sum_{n=10}^{50} \sum_{k=0}^{\left\lfloor \frac{50-n}{5} \right\rfloor} (-1)^k \cdot \binom{n}{k} \cdot \binom{49-5k}{n-1}$$

Diese ist im CAS von *geogebra* sehr leicht berechenbar und ergibt wunschgemäß den schon bekannten Wert.



Allgemeiner gilt:

$$f_m^{(n)} = \sum_{\nu=\left\lceil \frac{n}{m} \right\rceil}^{n} \sum_{\mu=0}^{\left\lfloor \frac{n-\nu}{m} \right\rfloor} (-1)^\mu \cdot \binom{\nu}{\mu} \cdot \binom{n-m\cdot\mu-1}{\nu-1}$$

Allerdings erfordert die Doppelsumme erhebliche Rechenzeit.