# Die Folge der Fibonacci-Zahlen als primitiv rekursive Funktion

Es wird gezeigt, dass die Folge der Fibonacci-Zahlen primitiv rekursiv ist (vgl. Übung am 28. 11. 2007).
Die Definition der Fibonacci-Zahlen von Übungsblatt 1:

$$\begin{aligned} \mathrm{fib}(0) &= 1 \\ \mathrm{fib}(1) &= 1 \\ \forall n > 1\ (\mathrm{fib}(n) &= \mathrm{fib}(n-1) + \mathrm{fib}(n-2)) \end{aligned}$$

Wenn wir zeigen wollen, dass fib primitiv rekursiv ist, besteht die Schwierigkeit in der 3. Gleichung: Hier wird zur Definition von $\mathrm{fib}(n)$ nicht nur auf $\mathrm{fib}(n-1)$ zurückgegriffen (was das primitive Rekursionsschema zulässt), sondern auch auf $\mathrm{fib}(n-2)$.
Hier hilft ein Blick auf die iterative Definition der Fibonacci-Zahlen, z.B. in Java:

```
static int fib(int n) {
  int f0 = 1, f1 = 1;
  for(int i = 1; i < n; ++i) {
    // Hier ist f0 = fib(i - 1) und f1 = fib(i).
    int oldF0 = f0;
    f0 = f1;
    f1 = oldF0 + f1;
  }
  return f1;
}
```

Während der Schleife werden *zwei* Zwischenergebnisse mitgeführt. Diese Idee lässt sich auf primitiv rekursive Funktionen übertragen, indem man eine Hilfsfunktion $\mathrm{fp} : \mathbb{N}_0 \to \mathbb{N}_0$ definiert, die die Gödelisierung eines Zahlenpaars liefert:

$$\mathrm{fp}(n) = \begin{cases} \langle 1, 1 \rangle, & \text{falls } n = 1 \\ \langle (\mathrm{fp}(n-1))_2, (\mathrm{fp}(n-1))_1 + (\mathrm{fp}(n-1))_2 \rangle, & \text{falls } n > 1 \end{cases}$$

$\mathrm{fp}(n)$ liefert $\langle \mathrm{fib}(n-1), \mathrm{fib}(n) \rangle$. Die Codierungsfunktion $\langle \cdot \rangle : \mathbb{N}_0^k \to \mathbb{N}_0$ und die zugehörigen Decodierungsfunktionen $(\cdot)_i : \mathbb{N}_0 \to \mathbb{N}_0$ für $1 \leq i \leq k$ sind aus der Vorlesung bekannt. Dort wurde auch erwähnt, dass sie primitiv rekursiv sind.
fib kann jetzt folgendermaßen dargestellt werden:

$$\mathrm{fib}(n) = \begin{cases} 1, & \text{falls } n = 0 \\ (\mathrm{fp}(n))_2, & \text{falls } n \neq 0 \end{cases}$$

Die obige Definition von fp entspricht noch nicht ganz dem primitiven Rekursionsschema; insbesondere ist fp(0) nicht definiert. Das wird jetzt behoben:

$$\begin{array}{lcl}
\mathrm{fp}'(0) & = & 0 \\
\mathrm{fp}'(m+1) & = & \langle (\mathrm{fp}'(m))_2, (\mathrm{fp}'(m))_1 + (\mathrm{fp}'(m))_2 \rangle \\[1em]
\mathrm{fp}(n) & = & \begin{cases} \langle 1, 1 \rangle, & \text{falls } n = 1 \\ \mathrm{fp}'(n), & \text{falls } n \neq 1 \end{cases}
\end{array}$$

Die Definition von $\mathrm{fp}'(0)$ ist beliebig, weil sie nie benutzt wird. fp und fib sind mittels Fallunterscheidung definiert und damit auch primitiv rekursiv.