Die Ackermann-Funktion

Die Ackermann-Funktion A ist für nichtnegative ganze Zahlen definiert durch

$$\begin{aligned} A(0,n) &= n+1, \\ A(m,0) &= A(m-1,1) \text{ für } m>0, \\ A(m,n) &= A(m-1,A(m,n-1)) \text{ für } m,n>0. \end{aligned}$$

Die von Wilhelm Ackermann (1896-1962) 1928 angegebene Funktion wächst außerordentlich schnell und zwingt jeden Rechner schnell in die Knie. Für kleine m und n kann der Wert von $A(m,n)$ allerdings auch iterativ berechnet werden.

$A(m,n)$	$n=0$	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$
$m=0$	1	2	3	4	5	6
$m=1$	2	3	4	5	6	7
$m=2$	3	5	7	9	11	13
$m=3$	5	13	29	61	125	253
$m=4$	13	65533	$2^{65536}-3$	$2^{2^{65536}}-3$	$2^{2^{2^{65536}}}-3$	$2^{2^{2^{2^{65536}}}}-3$
$m=5$	65533					
$m=6$	$\left.2^{2^{2^{\cdots}}}\right\\}$ (65536mal) -3					

$A(4,2)$ hat schon 19.729 Stellen. Für solch große Zahlen gibt es keinen vorhandenen Datentyp in einem realen Rechner. Selbst die mathematische Notation für solch große Zahlen versagt schnell bzw. wird unpraktisch.
Ein Programm, dass diese Zahlen rekursiv berechnen soll, ist sehr einfach formuliert. Beim Programmlauf läuft aber wegen der Tiefe der Rekursion entweder der Rechnerspeicher voll oder es gibt einen Laufzeitfehler wegen Überschreitung der Obergrenzen der `int`- (32 Bit) oder `long`-Typen (64-Bit).
Die Ackermann-Funktion spielt eine Rolle in der theoretischen Informatik im Rahmen der Berechenbarkeitstheorie. Man kann beweisen, dass die Ackermann-Funktion *berechenbar*, aber nicht *primitiv-rekursiv* ist. Weil sie aber *$\mu-$rekursiv* ist, ist sie auch Turing-*berechenbar.* Was eine Turing-Maschine ist, werden wir in der 13 behandeln.