(16.7) BEM: Aufwand für die Durchführung eines Primzahltestes

Für die Brauchbarkeit eines Kriteriums oder eines Algorithmus sind die Laufzeit oder der Rechenaufwand entscheidend. Es eignen sich eigentlich nur Algorithmen mit **polynomialer Laufzeit** für die Anwendung. Bei einem solchen Algorithmus ist die Laufzeit für ein $n \in \mathbb{N}$ beschränkt durch einen polynomialen Ausdruck $f(\ln(n))$ in $\ln(n)$. Bei einem Verfahren mit **exponentieller Laufzeit** ist diese beschränkt durch

$$f(n) \;=\; f(e^{\ln(n)})\,.$$

Die in diesem Paragraphen behandelten Verfahren sind alle exponentiell.

In den letzten Jahren hat es in dieser Hinsicht große Verbesserungen gegeben:

- **APR–Test** nach Adleman/Pomerance/Rumely (1983)
  Dies ist ein universeller deterministischer Primzahl–Test mit fast polynomialer Laufzeit, der stark auf algebraischer Zahlentheorie basiert. Zahlen mit bis zu 200 Dezimalstellen werden in ca. 10 min. getestet.
- **ECPP** (elliptic curve primality proving) nach Atkin (1986) und Atkin/Morain (1993). Für den Test einer Zahl mit 15 000 Dezimalstellen wurden 720 Tage benötigt.
- **AKS–Test** nach Agrawal/Kayal/Saxena (2002)
  Dies ist ein universeller deterministischer Primzahl–Test mit polynomialer Laufzeit. Als Charakterisierung einer Primzahl wird benutzt:

$$n \text{ prim} \quad\Longleftrightarrow\quad (1-T)^n \equiv 1 - T^n \;(\mathbf{mod}\; n)\,.$$