

3.3 Der BERLEKAMP-MASSEY-Algorithmus

Der Beweis von Satz 1 ist konstruktiv: Er enthält einen Algorithmus, mit dem sukzessive ein linearer Rekurrenzgenerator aufgebaut wird. Beim Schritt von der Folgenlänge n zur Folgenlänge $n + 1$ gibt es drei mögliche Fälle (1, 2a, 2b):

1. **Fall** $d_n = 0$, d. h. der Generator zum Rückkopplungspolynom φ erzeugt auch u_n : Dann bleiben φ und l ungeändert und ebenso ψ, t, r, d_r .
2. **Fall** $d_n \neq 0$, d. h. der Generator zum Rückkopplungspolynom φ erzeugt u_n nicht: Dann wird ein neues Rückkopplungspolynom η gebildet, dessen zugehöriger Generator (u_0, \dots, u_n) erzeugt. Es wird weiter unterschieden:
 - a) $l > \frac{n}{2}$: Dann ist $\lambda_{n+1} = \lambda_n$; es wird φ durch η ersetzt, l bleibt, und ebenso bleiben ψ, t, r, d_r .
 - b) $l \leq \frac{n}{2}$: Dann ist $\lambda_{n+1} = n + 1 - \lambda_n$; es wird φ durch η und l durch $n + 1 - l$ ersetzt, ferner ψ durch φ , t durch l , r durch n und d_r durch d_n .

Damit ist der BERLEKAMP-MASSEY-Algorithmus semiformal beschreibbar:

Input: Eine Folge $u = (u_0, \dots, u_{N-1}) \in K^N$.

Output: Die lineare Komplexität $\lambda_N(u)$,

das Rückkopplungspolynom φ eines linearen Rekurrenzgenerators der Länge $\lambda_N(u)$, der u erzeugt.

Hilfsvariablen: n : aktueller Index, initialisiert mit $n := 0$,

l : aktuelle lineare Komplexität, initialisiert mit $l := 0$,

φ : aktuelles Rückkopplungspolynom $= 1 - a_1T - \dots - a_lT^l$, initialisiert mit $\varphi := 1$,

Invarianzbedingung: $u_i = a_1u_{i-1} + \dots + a_lu_{i-l}$ für $l \leq i < n$,

d : aktuelle Diskrepanz $= u_n - a_1u_{n-1} - \dots - a_lu_{n-l}$,

r : voriger Index, initialisiert mit $r := -1$,

t : vorige lineare Komplexität,

ψ : voriges Rückkopplungspolynom $= 1 - b_1T - \dots - b_tT^t$, initialisiert mit $\psi := 1$,

Invarianzbedingung: $u_i = b_1u_{i-1} + \dots + b_tu_{i-t}$ für $t \leq i < r$,

d' : vorige Diskrepanz $= u_r - b_1u_{r-1} - \dots - b_tu_{r-t}$, initialisiert mit $d' := 1$,

η : neues Rückkopplungspolynom,
 m : neue lineare Komplexität.

Iterationsschritte: Für $n = 0, \dots, N - 1$:

$$\begin{aligned}
 d &:= u_n - a_1 u_{n-1} - \dots - a_l u_{n-l} \\
 \text{Falls } d &\neq 0 \\
 \eta &:= \varphi - \frac{d}{d'} \cdot T^{n-r} \cdot \psi \\
 \text{Falls } l &\leq \frac{n}{2} \text{ [lineare Komplexität wächst]} \\
 m &:= n + 1 - l \\
 t &:= l \\
 l &:= m \\
 \psi &:= \varphi \\
 r &:= n \\
 d' &:= d \\
 \varphi &:= \eta
 \end{aligned}$$

Natürlich kann man sich auch gleich die ganze Folge (λ_n) ausgeben lassen.

Dieser Algorithmus wird jetzt auf das **Beispiel** der Folge 001101110 angewendet. Der Fall $d \neq 0, l \leq \frac{n}{2}$ wird durch „[!]“ bezeichnet.

Eingangsbedingungen	Aktionen
$n = 0 \quad u_0 = 0 \quad l = 0 \quad \varphi = 1$ $r = -1 \quad d' = 1 \quad t = \quad \psi = 1$	$d := u_0 = 0$
$n = 1 \quad u_1 = 0 \quad l = 0 \quad \varphi = 1$ $r = -1 \quad d' = 1 \quad t = \quad \psi = 1$	$d := u_1 = 0$
$n = 2 \quad u_2 = 1 \quad l = 0 \quad \varphi = 1$ $r = -1 \quad d' = 1 \quad t = \quad \psi = 1$	$d := u_2 = 1$ [!] $\eta := 1 - T^3$ $m := 3$
$n = 3 \quad u_3 = 1 \quad l = 3 \quad \varphi = 1 - T^3$ $r = 2 \quad d' = 1 \quad t = 0 \quad \psi = 1$	$d := u_3 - u_0 = 1$ $\eta := 1 - T - T^3$
$n = 4 \quad u_4 = 0 \quad l = 3 \quad \varphi = 1 - T - T^3$ $r = 2 \quad d' = 1 \quad t = 0 \quad \psi = 1$	$d := u_4 - u_3 - u_1 = -1$ $\eta := 1 - T + T^2 - T^3$
$n = 5 \quad u_5 = 1 \quad l = 3 \quad \varphi = 1 - T + T^2 - T^3$ $r = 2 \quad d' = 1 \quad t = 0 \quad \psi = 1$	$d := u_5 - u_4 + u_3 - u_2 = 1$ $\eta := 1 - T + T^2 - 2T^3$

Von jetzt an unterscheiden sich die Ergebnisse je nach Charakteristik des Grundkörpers K . Sei zuerst $\text{char } K \neq 2$. Dann geht es so weiter:

Eingangsbedingungen	Aktionen
$n = 6 \quad u_6 = 1 \quad l = 3$ $\varphi = 1 - T + T^2 - 2T^3$ $r = 2 \quad d' = 1 \quad t = 0 \quad \psi = 1$	$d := u_6 - u_5 + u_4 - 2u_3 = -2$ [!] $\eta = 1 - T + T^2 - 2T^3 + 2T^4$ $m := 4$
$n = 7 \quad u_7 = 1 \quad l = 4$ $\varphi = 1 - T + T^2 - 2T^3 + 2T^4$ $r = 6 \quad d' = -2 \quad t = 3$ $\psi = 1 - T + T^2 - 2T^3$	$d := u_7 - u_6 + u_5 - 2u_4 + 2u_3 = 3$ $\eta = 1 + \frac{1}{2}T - \frac{1}{2}T^2 - \frac{1}{2}T^3 - T^4$
$n = 8 \quad u_8 = 0 \quad l = 4$ $\varphi = 1 + \frac{1}{2}T - \frac{1}{2}T^2 - \frac{1}{2}T^3 - T^4$ $r = 6 \quad d' = -2 \quad t = 3$ $\psi = 1 - T + T^2 - 2T^3$	$d := u_8 + \frac{1}{2}u_7 - \frac{1}{2}u_6 - \frac{1}{2}u_5 - u_4 = -\frac{1}{2}$ [!] $\eta := 1 + \frac{1}{2}T - \frac{3}{4}T^2 - \frac{1}{4}T^3 - \frac{5}{4}T^4 + \frac{1}{2}T^5$ $m := 5$

Als Ergebnis erhalten wir die Folge der linearen Komplexitäten

$$\lambda_0 = 0, \lambda_1 = 0, \lambda_2 = 0, \lambda_3 = 3, \lambda_4 = 3, \lambda_5 = 3, \lambda_6 = 3, \lambda_7 = 4, \lambda_8 = 4, \lambda_9 = 5$$

und die Rekursionsvorschrift

$$u_i = -\frac{1}{2}u_{i-1} + \frac{3}{4}u_{i-2} + \frac{1}{4}u_{i-3} + \frac{5}{4}u_{i-4} - \frac{1}{2}u_{i-5} \quad \text{für } i = 5, \dots, 8.$$

Im Falle $\text{char } K = 2$ sehen die letzten drei Iterationen so aus:

Eingangsbedingungen	Aktionen
$n = 6 \quad u_6 = 1 \quad l = 3$ $\varphi = 1 - T - T^2$ $r = 2 \quad d' = 1 \quad t = 0 \quad \psi = 1$	$d := u_6 - u_5 - u_4 = 0$
$n = 7 \quad u_7 = 1 \quad l = 3$ $\varphi = 1 - T - T^2$ $r = 2 \quad d' = 1 \quad t = 0 \quad \psi = 1$	$d := u_7 - u_6 - u_5 = 1$ [!] $\eta = 1 - T - T^2 - T^5$ $m := 5$
$n = 8 \quad u_8 = 0 \quad l = 5$ $\varphi = 1 - T - T^2 - T^5$ $r = 7 \quad d' = 1 \quad t = 3 \quad \psi = 1 - T - T^2$	$d := u_8 - u_7 - u_6 - u_3 = 1$ $\eta := 1 - T^3 - T^5$

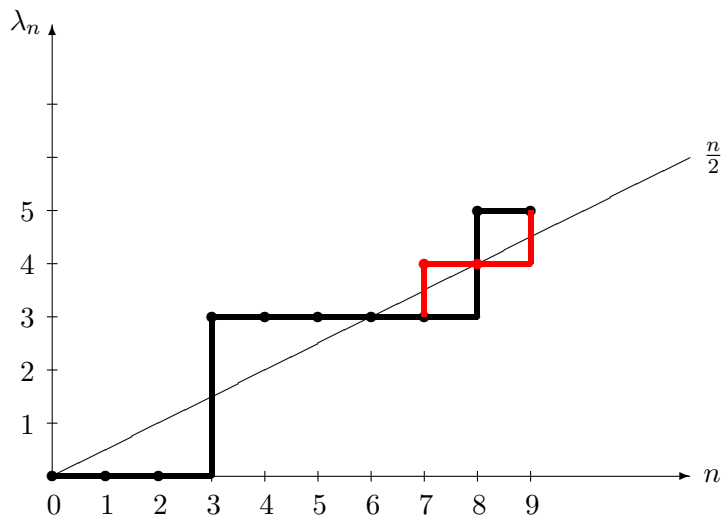
Die Folge der linearen Komplexitäten ist hier

$$\lambda_0 = 0, \lambda_1 = 0, \lambda_2 = 0, \lambda_3 = 3, \lambda_4 = 3, \lambda_5 = 3, \lambda_6 = 3, \lambda_7 = 3, \lambda_8 = 5, \lambda_9 = 5$$

und die Rekursionsvorschrift

$$u_i = u_{i-3} + u_{i-5} \quad \text{für } i = 5, \dots, 8.$$

Die Entwicklung der linearen Komplexität wird auch noch grafisch dargestellt; die rote Linie kennzeichnet den Fall $\text{char } K \neq 2$:



Der Aufwand für den BERLEKAMP-MASSEY-Algorithmus ist $O(N^2 \log N)$.

Die Folge $(\lambda_n)_{n \in \mathbb{N}}$ bzw. (für endliche Bitfolgen) $(\lambda_n)_{0 \leq n \leq N}$ heißt das **Linearitätsprofil** der Bitfolge u .