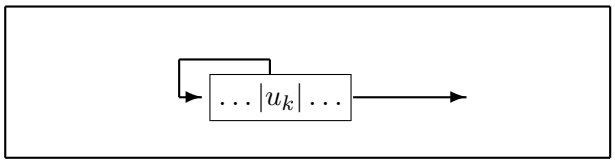
3.1 Die lineare Komplexität von Bitfolgen

Wir betrachten – zunächst unendliche – Bitfolgen $u = (u_i)_{i\in\mathbb{N}} \in \mathbb{F}_2^{\mathbb{N}}$. Gesucht ist ein lineares Schieberegister möglichst geringer Länge, das die Folge produziert.

Gibt es ein solches Schieberegister, muss die Folge periodisch sein. Umgekehrt wird jede periodische Folge stets durch ein lineares Schieberegister erzeugt, dessen Länge die Summe der Längen von Vorperiode und Periode ist – nämlich durch das **zirkuläre Schieberegister**, das als Rückkopplung nur dasjenige Bit wieder einspeist, bei dem die Periode beginnt; ist $u_{l+i} = u_{k+i}$ für $i \geq 0$, so sind die Koeffizienten $a_{l-k} = 1$, $a_i = 0$ sonst.



Damit ist gezeigt:

Hilfssatz 1 *Eine Bitfolge $u \in \mathbb{F}_2^{\mathbb{N}}$ lässt sich genau dann von einem linearen Schieberegister erzeugen, wenn sie periodisch ist.*

Definition. Die **lineare Komplexität** $\lambda(u)$ einer Bitfolge $u \in \mathbb{F}_2^{\mathbb{N}}$ ist die minimale Länge eines linearen Schieberegisters, das u erzeugt.

Ist u konstant 0, wird $\lambda(u) = 0$, ist u nicht periodisch, wird $\lambda(u) = \infty$ gesetzt.

Es handelt sich hierbei also um einen Komplexitätsbegriff, der auf dem sehr speziellen Maschinen-Modell der linearen Schieberegister beruht.

Bemerkungen und Beispiele

1. Falls $\tau(u)$ die Summe aus der Länge der Periode und der Vorperiode von u ist und u von einem linearen Schieberegister der Länge l erzeugt wird, gilt
$$\lambda(u) \leq \tau(u) \leq 2^l - 1 \quad \text{und} \quad \lambda(u) \leq l.$$

2. Die periodisch wiederholte Folge $0, \ldots, 0, 1$ ($l-1$ Nullen) hat die Periode l und die lineare Komplexität l. Ein lineares Schieberegister der Länge $< l$ würde nämlich mit dem Nullvektor als Startwert gefüttert und könnte dann nur noch weitere Nullen produzieren.

Für endliche Bitfolgen $u = (u_0, \ldots, u_{N-1}) \in \mathbb{F}_2^N$ definiert man die lineare Komplexität analog. Insbesondere ist $\lambda(u)$ die minimale Zahl l, so dass es $a_1, \ldots, a_l \in \mathbb{F}_2$ gibt mit

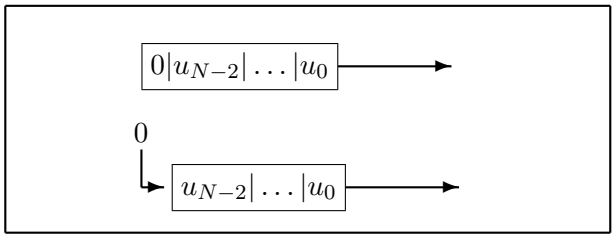
$$u_i = a_1 u_{i-1} + \cdots + a_l u_{i-l} \quad \text{für } i = l, \ldots, N-1.$$

Bemerkungen und Beispiele

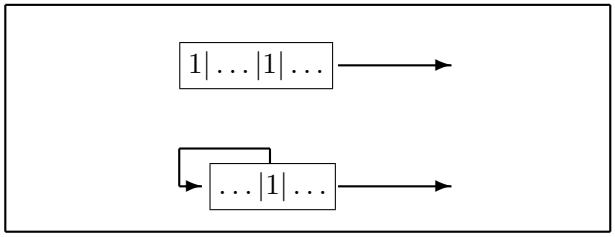
3. Für $u \in \mathbb{F}_2^N$ gilt $0 \leq \lambda(u) \leq N$.

4. $\lambda(u) = 0 \Longleftrightarrow u_0 = \cdots = u_{N-1} = 0$.

5. $\lambda(u) = N \Longleftrightarrow u = (0, \ldots, 0, 1)$. Die Implikation „$\Longleftarrow$“ folgt wie in Bemerkung 2. Für die Umkehrung kann u_{N-1} nicht 0 sein, denn sonst könnte man das lineare Schieberegister der Länge $N-1$ mit Rückkopplung konstant 0 nehmen; die beiden Schieberegister



haben den gleichen Output. Also muss $u_{N-1} = 1$ sein. Gäbe es vorher in der Folge schon eine 1, könnte man das Schieberegister der Länge $N-1$ nehmen, das genau diese Bitposition rückkoppelt; die beiden Schieberegister



haben den gleichen Output.

6. Sind die ersten $2\lambda(u)$ Bits der Bitfolge u bekannt, so lässt sich der Rest von u daraus vorhersagen. (Dem Kryptoanalytiker, der n Bits der Folge kennt, den Rest aber nicht, ist natürlich auch $\lambda(u)$ unbekannt, d. h., er weiß nicht, dass seine Vorhersage von nun an immer korrekt sein wird. Das hindert ihn aber nicht daran, Bit für Bit, und zwar korrekt, vorherzusagen!)