# 3. Zyklische Codes. Endliche Körper

Codierungstheorie
Anton Malevich
21.02.2018

## Definitionen

### Zyklische Codes

Ein linearer Code $C \leqslant K^n$ der Länge $n$ heißt zyklisch, falls für jedes Codewort $(c_0,\ c_1,\ \ldots,\ c_{n-2},\ c_{n-1}) \in C$ stets $(c_{n-1},\ c_0,\ c_1,\ \ldots,\ c_{n-2}) \in C$ ist. Mit einem Codewort $c$ sind also auch alle Worte, die durch zyklische Vertauschungen der Koordinaten aus $c$ entstehen, wieder Codeworte.

Zyklische Codes lassen sich einfach beschreiben, wenn man die Codeworte $c = (c_0, c_1, \ldots, c_{n-1}) \in C$ als *Codepolynome* $c(x) = c_0 + c_1 x + \ldots + c_{n-1} x^{n-1} \in K[x]$ auffasst. Ist $\overline{c} = (c_{n-1}, c_0, c_1, \ldots, c_{n-2})$ die zyklische Vertauschung von $c \in C$, so gilt

```
x c (x) = c̄ (x) + c_{n-1} (x^n - 1) ≡ c̄ (x) mod (x^n - 1).
```

Insbesondere ist also für jedes Polynom $f(x) \in K[x]$ und jedes Codepolynom $c(x) \in C$ das Polynom $f(x)\,c(x) \bmod (x^n - 1)$, ebenfalls ein Codepolynom.

Das normierte Polynom $0 \neq g(x) \in C$ von minimalem Grad in $C$ heißt das *Erzeugerpolynom* von $C$. Das Polynom $h(x) = \frac{x^n-1}{g(x)}$ heißt das *Kontrollpolynom von C*. Es gilt $\dim C = n - \operatorname{Grad} g(x)$ und

```
C = {f (x) g (x) | f (x) ∈ K[x], Grad f (x) < n - Grad g (x)}.
```

Sind $g(x) = g_0 + g_1 x + \ldots + g_{n-k} x^{n-k}$ das Erzeuger- und $h(x) = h_0 + h_1 x + \ldots + h_k x^k$ das Kontrollpolynom von $C$, so sind

$$G = \begin{pmatrix} g_0 & g_1 & \ldots & g_{n-k} & 0 & \ldots & 0 \\ 0 & g_0 & g_1 & \ldots & g_{n-k} & \ldots & 0 \\ \vdots & \vdots & \ddots & & & \ddots & \vdots \\ 0 & \ldots & 0 & g_0 & g_1 & \ldots & g_{n-k} \end{pmatrix} \text{ und } H = \begin{pmatrix} h_k & h_{k-1} & \ldots & h_0 & 0 & \ldots & 0 \\ 0 & h_k & h_{k-1} & \ldots & h_0 & \ldots & 0 \\ \vdots & \vdots & \ddots & & & \ddots & \vdots \\ 0 & \ldots & 0 & h_k & h_{k-1} & \ldots & h_0 \end{pmatrix}$$

eine Erzeuger- bzw. eine Kontrollmatrix von $C$.

### Cyclic Redundancy Check Codes (CRC-Codes)

Sei $C$ ein zyklischer $[n, k]$-Code über $K$ mit dem Erzeugerpolynom $g(x)$ vom Grad $n - k$. Eine Nachricht $a(x) = a_0 + a_1 x + \ldots + a_{k-1} x^{k-1}$ wird bei der CRC-Codierung zur $c(x) = x^{n-k}\,a(x) - r(x) \in C$ codiert, wobei $r(x) = x^{n-k}\,a(x) \bmod g(x)$ ist.

Erhält der Empfänger ein Wort $v(x)$ mit $g(x) \nmid v(x)$, so weiß er, dass Fehler passiert sind.
Ist $g(x) \mid v(x)$, so kann die Nachricht wegen Grad $r(x) < n - k$ unmittelbar aus $x^{n-k}\,a(x)$ abgelesen werden.

Die CRC-Codes können alle Bündelfehler der Länge $b < \operatorname{Grad} g(x)$ erkennen . Das sind Fehlervektoren, in denen die Einsen nur in einem Block der Länge $b$ (auch zyklisch) vorkommen, z.B. $b$ nacheinender folgenden Einsen.

### Polynome und endliche Körper

Ein Polynom $f(x) \in K[x]$ heißt *irreduzibel*, falls es keine $f_1(x)$, $f_2(x)$ gibt mit Grad $f_i(x) \geqslant 1$, sodass $f(x) = f_1(x)\,f_2(x)$.
Ist $f(x)$ irreduzibel, so ist die Menge ${}^{K[x]}/_{\langle f(x)\rangle} = \{v(x) \in K[x] \mid \operatorname{Grad} v(x) < \operatorname{Grad} f(x)\}$, wobei die Addition und Multiplikation modulo $f(x)$ durchgeführt werden, ein Körper.
Ein irreduzibles Polynom $f(x)$ heißt *primitiv*, falls die Potenzen $x^i \bmod f(x)$ für $0 \leqslant i \leqslant |K|^{\operatorname{Grad} f(x)} - 2$ alle unterschiedlich sind.

# Aufgaben

## Aufgabe 1

Zeigen Sie, dass der binäre lineare Code mit Ergeugermatrix *H* zyklisch ist. Bestimmen Sie das Erzeuger- und das Kontrollpolynom von *C*.

$$H = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

```
{{1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0},
 {0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0}, {1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0},
 {1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0}, {1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0},
 {1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0}, {0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1}}
```

## Aufgabe 2 (Fehlerfang-Methode)

Wir betrachten den zyklischen Code aus Aufgabe 1. Die Erzeugermatrix ist

```
MatrixForm[G = Join[IdentityMatrix[Abs[Subtract @@ Dimensions[H]]], H[[All, 1 ;; 5]]^T]]
```

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Warum?

Die Dimension *k* und die Länge *n* des Codes sind

```
{k, n} = Dimensions[G]
```

```
{5, 12}
```

In diesem Code kommen nur gerade Gewichte vor:

```
L = Prepend[Mod[Plus @@@ Rest@Subsets[G], 2], Array[0 &, 12]];
Sort@Union@Flatten@Outer[HammingDistance, L, L, 1, 1]
```

```
{0, 4, 6, 8, 12}
```

Beweisen Sie: Genau dann haben alle Codeworten eines zyklischen Codes *C* gerades Gewicht, wenn $(x + 1)$ das Erzeugerpolynom von *C* teilt.

Die Minimaldistanz des Codes ist 4; er kann somit auf jeden Fall einen Fehler korrigiert werden. Für zyklische Codes gibt es allerdings einen Decodier-Algorithmus, genannt *Fehlerfang-Methode*, der bis zu $\left\lfloor \frac{n-1}{k} \right\rfloor$ Fehler (und sogar bis zu *k* Fehler in nacheinander folgenden Bits) korigieren kann.

Angenommen, es wurde das 31. Codewort versandt und es sind zwei Fehler passiert, sodass *y* empfangen wurde:

```
y = Mod[L[[31]] + {1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0}, 2]
```

```
{1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1}
```

Zuerst berechnen wir $y_j = \sigma^j y$, wobei $\sigma$ die zyklische Vertauschung ist, bis wir ein $j$ finden, sodass das Gewicht des Syndroms klein ist: $\mathrm{wt}\, s_H(y_j) \leqslant \left\lfloor \frac{n-1}{k} \right\rfloor$.

```
syndrome = .
bound = Floor[(n - 1)/k]; j = 0;
yj = NestWhile[(j += 1;
    RotateRight[#]) &, y, (Total[syndrome = Mod[H.#, 2]] > bound) &, 1, n]
j
syndrome
```

```
{0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1}
```

```
5
```

```
{1, 0, 1, 0, 0, 0, 0}
```

Nun setzen wir den Fehlervektor zum

```
f = Join[Array[0 &, k], syndrome]
```

```
{0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0}
```

und decodieren richtig zum

```
c = RotateLeft[Mod[yj - f, 2], j]
Position[L, c]
```

```
{0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1}
```

```
{{31}}
```

Schreiben Sie eine Funktion für den Fehlerfang-Decodierer und decodieren Sie die folgenden 5 empfangenen Vektoren. Welche Nachricht wurde versendet? (Wir setzen voraus, dass das erste Codewort den Buchstaben "*A*" codiert, das zweite "*B*" usw. bis "*Z*".)

```
{1,0,0,0,0,0,0,1,0,0,0,0}
{0,1,0,1,0,1,0,0,0,1,0,1}
{1,0,1,1,0,0,1,1,0,1,0,0}
{1,0,0,1,0,1,1,1,0,0,1,0}
{0,0,1,1,0,1,1,1,1,0,1,1}
```

# Aufgabe 3 (CRC-Codes)

## Teil 1 (Beispiel)

Es sei $C$ der binäre zyklische [15, 10]-Code mit dem Erzeugerpolynom $g(x)$.

```
{n, k} = {15, 10};
m[x_] = x^4 + x^3 + x^2 + x + 1;
g[x_] = Expand[(x + 1) m[x], Modulus → 2]
```

$1 + x^5$

Zeigen Sie, dass $m(x)$ irreduzibel, aber nicht primitiv ist.

Es sei $a(x)$ die Nachricht der Länge 10. Diese wird codiert zu

```
al = {1, 0, 1, 0, 0, 0, 0, 1, 0, 1};
a[x_] = Expand[FromDigits[Reverse@al, x]]
Collect[x^(n-k) a[x] + PolynomialRemainder[x^(n-k) a[x], g[x], x], x, Modulus → 2]
cl = CoefficientList[%, x]
```

$1 + x^2 + x^7 + x^9$

$1 + x^4 + x^5 + x^7 + x^{12} + x^{14}$

```
{1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1}
```

Der Code $C$ kann eine ungerade Anzahl von Fehlern erkennen (Warum?) und alle Bündelfehler der Länge $< 5$. Zum Beispiel,

```
y = Mod[cl + {0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0}, 2]
Expand[FromDigits[Reverse@y, x]]
PolynomialRemainder[%, g[x], x, Modulus → 2]
```

```
{1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1}
```

$1 + x^4 + x^5 + x^6 + x^8 + x^9 + x^{12} + x^{14}$

$x + x^2 + x^3 + x^4$

Es werden aber leider nicht alle Fehler vom Gewicht 2 erkannt. Zum Beispiel,

```
y = Mod[cl + {0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0}, 2]
Expand[FromDigits[Reverse@y, x]]
PolynomialRemainder[%, g[x], x, Modulus → 2]
```

```
{1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1}
```

$1 + x^3 + x^4 + x^5 + x^7 + x^8 + x^{12} + x^{14}$

```
0
```

Welche Originalnachricht wird im letzten Fall falsch decodiert?

## Teil 2

Es sei nun $C$ der binäre zyklische [15, 10]-Code mit dem Erzeugerpolynom $g(x)$.

```
{n, k} = {15, 10};
m[x_] = x^4 + x + 1;
g[x_] = Expand[(x + 1) m[x], Modulus → 2]
```

$1 + x^2 + x^4 + x^5$

Zeigen Sie, dass $m(x)$ primitiv ist.

Zeigen Sie, dass der Code $C$ alle Fehler vom Gewicht 2 erkennt.

# Aufgabe 4 (endliche Körper)

Wir konstruieren einen Körper mit 8 Elementen. Die Elemente sind die Polynome mit Koeffizienten aus $\mathbb{F}_2 = \{0, 1\}$ mit Grad $< 3$, und Addition und Multiplikation sind modulo $m(x)$ durchzuführen.

```
m[x_] = x^3 + x + 1;
Add[A_, B_] := PolynomialMod[A + B, m[x], Modulus → 2];
Multiply[A_, B_] := PolynomialMod[A * B, m[x], Modulus → 2]
```

Die Verknüpfungstabellen sehen wie folgt aus

```
elts = FromDigits[#, x] & /@ Tuples[{0, 1}, 3];
Outer[Add, elts, elts];
Grid[ReplacePart[%, {1, 1} → "+"], Dividers → {{False, True}, {False, True}}]
Outer[Multiply, Rest[elts], Rest[elts]];
Grid[ReplacePart[%, {1, 1} → "×"], Dividers → {{False, True}, {False, True}}]
```

| + | 1 | x | 1 + x | $x^2$ | $1 + x^2$ | $x + x^2$ | $1 + x + x^2$ |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 + x | x | $1 + x^2$ | $x^2$ | $1 + x + x^2$ | $x + x^2$ |
| x | 1 + x | 0 | 1 | $x + x^2$ | $1 + x + x^2$ | $x^2$ | $1 + x^2$ |
| 1 + x | x | 1 | 0 | $1 + x + x^2$ | $x + x^2$ | $1 + x^2$ | $x^2$ |
| $x^2$ | $1 + x^2$ | $x + x^2$ | $1 + x + x^2$ | 0 | 1 | x | 1 + x |
| $1 + x^2$ | $x^2$ | $1 + x + x^2$ | $x + x^2$ | 1 | 0 | 1 + x | x |
| $x + x^2$ | $1 + x + x^2$ | $x^2$ | $1 + x^2$ | x | 1 + x | 0 | 1 |
| $1 + x + x^2$ | $x + x^2$ | $1 + x^2$ | $x^2$ | 1 + x | x | 1 | 0 |

| × | x | 1 + x | $x^2$ | $1 + x^2$ | $x + x^2$ | $1 + x + x^2$ |
|---|---|---|---|---|---|---|
| x | $x^2$ | $x + x^2$ | $1 + x$ | 1 | $1 + x + x^2$ | $1 + x^2$ |
| 1 + x | $x + x^2$ | $1 + x^2$ | $1 + x + x^2$ | $x^2$ | 1 | x |
| $x^2$ | 1 + x | $1 + x + x^2$ | $x + x^2$ | x | $1 + x^2$ | 1 |
| $1 + x^2$ | 1 | $x^2$ | x | $1 + x + x^2$ | 1 + x | $x + x^2$ |
| $x + x^2$ | $1 + x + x^2$ | 1 | $1 + x^2$ | 1 + x | x | $x^2$ |
| $1 + x + x^2$ | $1 + x^2$ | x | 1 | $x + x^2$ | $x^2$ | 1 + x |

Versuchen Sie nun das Gleiche mit dem Polynom $x^3 + x^2 + x + 1$. Was geht schief? Warum?