Algebraische Strukturen Teil 2

Algebraische Strukturen

Algebraische Strukturen:

- Gruppen
- Ringe
- Körper
- *Vektorräume: Einführung*

Außerdem:

- *Beispiel für Ringe:* Rechnen im Polynomring; die Polynomdivision zum Lösen von Gleichungssystemen
- *Beispiel für Körper:* Die komplexen Zahlen

Anwendung aus der Informatik:

- Reed Solomon fehlerkorrigierende Codes

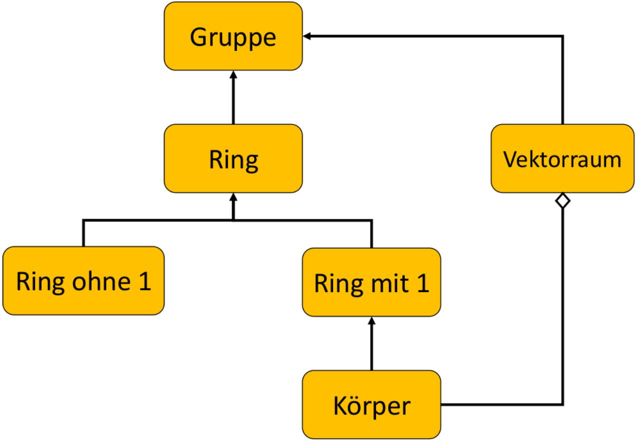
Permutationen: Ergänzung

Beispiel:
Permutation

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 1 & 3 & 4 & 6 & 5 \end{pmatrix}$$

schreiben wir alternativ als $(1\ 2)(5\ 6)$.

Übersicht über algebraische Strukturen



Polynom-Ringe

Polynom-Ringe

Definition:
Sei R ein Ring mit $a_0, a_1, \ldots, a_n \in R$. Die Abbildung

$$f : R \to R, x \mapsto a_n x^n + a_{n-1} x^{n-1} + \ldots a_1 x + a_0$$

heißt **Polynom(funktion)** über R. Wenn $a_n \neq 0$ ist, ist n der **Grad** von f.

Ein Polynom ist eine altbekannte Sache, wenn man unter R die Menge der reellen Zahlen $\mathbb{R}$ versteht,
z. B. $f(x) = x + 1$ oder $f(x) = x^3 + 3 \cdot x^2 - 2$.

Polynom-Ringe

Ein Polynom macht aber (alleine) noch keinen Ring:

Definition:
Sei R ein Ring und wir bezeichnen die Menge aller Polynome definert in X über R mit $R[X]$. Wir haben die Verknüpfungen $+$ und $\cdot$ folgendermaßen definiert:

$$(1)\quad p + q = (p + q)(x) := p(x) + q(x)$$
$$(2)\quad p \cdot q = (p \cdot q)(x) := p(x) \cdot q(x)$$

für alle $x \in X$. $R[X]$ heißt dann **Polynomring** über R.

Polynomdivision

Polynomdivision

In den reellen Zahlen ist Division nicht besonders schwierig. Im Polynomring ist Division etwas schwieriger, aber auch hier funktioniert es. Zuerst: Über was reden wir genau?

Satz:
Sei K ein Körper, $K[X]$ der Polynomring über K. Dann gibt es für $f, g \in K[X]$ $(g \neq 0)$ $q, r \in K[X]$, sodass gilt $f = q \cdot g + r$ und $grad(r) < grad(g)$.
Erinnerung: Folie 58 in letzer Vorlesung für $(Z, +, .)$.
Beweis: Buch Hartmann, Kapitel 5.

Polynomdivision

Erinnern wir uns zuerst nochmal wie Division funktioniert:

$$
\begin{array}{r}
285 : 9 = 31 \\
\underline{-27} \\
15 \\
\underline{-9} \\
6
\end{array}
$$

Die Aussage hier ist: $285 = 9 \cdot 31 + 6$.
Das ist dem vorherigen Satz nicht unähnlich. Wir haben etwas großes (285) in das Produkt zweier Zahlen plus einen kleineren Rest aufgespalten. Polynomdivision funktioniert nun gar nicht so anders.

Polynomdivision

Vielleicht haben Sie das in der Schule schon einmal gemacht:

$$
\begin{array}{l}
(x^3-5x^2+10x-8) : (x-1) = x^2-4x+6 \\
\underline{-(x^3\ \ -x^2)\qquad\qquad} \\
\qquad -4x^2+10x-8 \\
\underline{\quad -(-4x^2\ \ +4x)\qquad} \\
\qquad\qquad\qquad 6x-8 \\
\underline{\qquad\qquad\quad -(6x-6)} \\
\qquad\qquad\qquad\quad -2
\end{array}
$$

Wir sehen: Das Prinzip hier ist quasi dasselbe. Wir spalten ein großes Polynom in das Produkt zweier kleinerer und einen Rest auf:
$x^3 - 5x^2 + 10x - 8 = (x^2 - 4x + 6)(x - 1) - 2$
Insbesondere: Grad(-2)$<$Grad($x - 1$)

Polynomdivision

Versuchen Sie es selbst:

$$(3x^3+2x-5) : (x^2+2x-1) = ?$$

Polynomdivision

Lösung:

$$
\begin{array}{l}
(3x^3 \qquad +2x-5) : (x^2+2x-1) = 3x - 6 \\
\underline{-(3x^3+6x^2-3x)\qquad} \\
\qquad -6x^2+5x \;\; -5 \\
\underline{\qquad -(-6x^2-12x+6)} \\
\qquad\qquad\qquad 17x-11
\end{array}
$$

Polynomdivision

Sehr interessant sind diejenigen Divisionen bei denen der Rest gleich 0 ist: D.h., wir haben ein Polynom f, so dass gilt $f = q \cdot g$ für zwei Polynome q, g mit kleinerem Grad als f,
z.B. wir können $(x^2 - 1)$ in $(x+1)(x-1)$ ohne Rest zerlegen, wie wir leicht nachrechnen können.

Satz
Wenn $f : \mathbb{R} \to \mathbb{R}$ eine Nullstelle x_0 hat, d.h. es gilt $f(x_0) = 0$, so ist f durch $(x - x_0)$ ohne Rest teilbar, d.h. es gibt $q \in K[\mathbb{R}]$ so that $f(x) = (x - x_0) \cdot q(x)$ für alle $x \in \mathbb{R}$.

Das bedeutet, wir haben die Nullstelle x_0 abgespalten!

Polynomdivision

Beweis:

Zu zeigen:
Wir haben f mit einer Nullstelle x_0. Die Behauptung ist nun, dass wir f aufspalten können in ein Polynom $(x - x_0)$ und in ein unbekanntes Polynom q mit Rest 0.

Ausgeschrieben:
$f(x) = (x - x_0) \cdot q(x)$

Polynomdivision

Beweis:

Wir wissen, dass für jede x $f(x) = (x - x_0)q(x) + r(x)$ und Grad$(r(x))$ $<$Grad$(x - x_0) = 1$.

Da Grad$(r(x)) = 0$ gilt, ist $r(x) = a_0 x^0 = a_0$ d.h. eine Konstante.

Also: $f(x) = (x - x_0)q(x) + a_0$.

Wir wissen $f(x_0) = 0 \Rightarrow 0 = f(x_0) = (x_0 - x_0)q(x_0) + a_0 = a_0$

$\Rightarrow f(x) = (x - x_0)q(x)$.

Polynomdivision

Wir haben nun bei f eine **Nullstelle abgespalten**.
Wir wissen, dass das in $\mathbb{R}$ nicht immer geht, so können wir etwa bei $x^2 + 1$ keine Nullstelle abspalten.
Es wäre sonst nämlich $x^2 + 1 = 0 \Leftrightarrow x^2 = -1$ und das ist in $\mathbb{R}$ nicht definiert.

Aber: $x^2 = -1$ ist in $\mathbb{C}$, der **Menge der komplexen Zahlen** definiert!
Und zwar mit $x = \pm i$. Wir können f über $\mathbb{C}$ betrachten ($\mathbb{C}$ ist ja nur die Erweiterung von $\mathbb{R}$ um i). In $\mathbb{C}$ zerfällt f also in $(x - i)(x + i)$. Da ein Polynom vom Grad 2 maximal 2 Nullstellen hat, können wir f nicht weiter aufspalten.

Merke: In $\mathbb{C}$ gilt: Wir können jedes Polynom in Polynome vom Grad 1 aufspalten. Das bedeutet es wenn wir sagen, dass $\mathbb{C}$ **algebraisch abgeschlossen** ist.
$\Rightarrow f = a_0(x - x_0)(x - x_1) \cdots (x - x_n)$

Komplexe Zahlen

Komplexe Zahlen

Komplexe Zahlen:

Was ist die Lösung von $x^2 = -1$?

Komplexe Zahlen

Die Menge der natürlichen Zahlen $\mathbb{N}$ wurde eingeführt, weil man Gleichungen der Form $5 - 7$ lösen wollte, die Menge der rationalen Zahlen $\mathbb{Q}$, weil man Gleichungen der Form $3/4$ lösen wollte.

Manche quadratische Gleichungen wie $x^2 + x + 6 = 0$ können in $\mathbb{R}$ gelöst werden. Die große Lösungsformel (Mitternachtsformel) gibt uns die Nullstellen aus, aber etwa $x^2 + 4x + 8 = 0$ hat keine Lösungen in $\mathbb{R}$.

Sogar simple Gleichungen wie $x^2 + 1 = 0$ haben in $\mathbb{R}$ keine Lösungen.

$\Rightarrow$ Gibt es einen größeren Definitionsbereich als $\mathbb{R}$, wo solche Gleichungen gelöst werden können?

Man führt ein: $x^2 + 1 = 0$ hat als Lösung $x = \pm i$.

Komplexe Zahlen

Die Einführung von $i^2 = -1$ führt dazu, dass auch alle anderen Gleichungen der Form $ax^2 + bx + c = 0$ gelöst werden können.

Die Erweiterung der Menge der reelen Zahlen $\mathbb{R}$ mit i kennen wir als **komplexe Zahlen** und bezeichnen wir als $\mathbb{C}$.

Ein Element $z \in \mathbb{C}$ wird dargestellt als $z = a + b \cdot i$ mit $a, b \in \mathbb{R}$.

Komplexe Zahlen

Definition

Die Zahl $i \in \mathbb{C}$ ist die **imaginäre Einheit**. Sie ist definiert als $i^2 = -1$.

Für $z = a + bi$ bezeichnet man a als **Realteil** von z und b als **Imaginärteil**. Man schreibt $a = \Re(z)$ und $b = \Im(z)$.

Beispiel

Sei $z = 2 + 3i$. Dann gilt:
$\Re(z) = 2$ und $\Im(z) = 3$.

Wichtig: $\Im(z) = 3$ und nicht $\Im(z) = 3i$!

Komplexe Zahlen

Die komplexen Zahlen $\mathbb{C}$ sind also eine Erweiterung der reelen Zahlen $\mathbb{R}$, sodass quadratische Polynome in **lineare Faktoren** zerlegt werden können, d.h. man kann alle Gleichungen der Form $ax^2 + bx + c = 0$ lösen:

$ax^2 + bx + c = 0 \Leftrightarrow x_1$ und x_2 Nullstellen $\Leftrightarrow (x - x_1)(x - x_2) = 0.$

Das gilt nicht nur für quadratische Polynome. Alle Polynome beliebigen Grades zerfallen über $\mathbb{C}$ in solche Linearfaktoren.

Wir nennen einen solchen Körper, über dem alle Polynome in Linearfaktoren zerfallen, **algebraisch abgeschlossen**.

Komplexe Zahlen

Beispiel

$x^2 + 1 = (x + i)(x - i)$.

Die Nullstellen des Polynoms sind $-i$ und i.

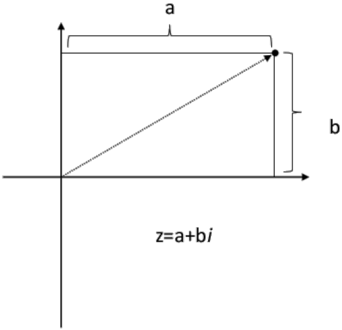
$x^4 - x^3 - 62x^2 + 120x + 448 = (x - 8)(x + 7)(x + 2)(x - 4)$

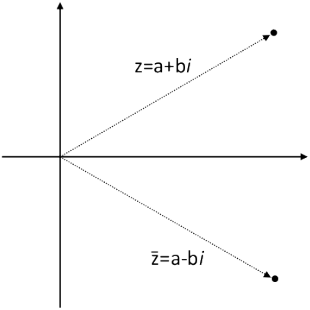
Die Nullstellen des Polynoms sind 8, -7, -2 und 4.

Den einzigen nicht-trivialen **algebraisch abgeschlossenen** Körper den wir kennen sind die komplexen Zahlen.
Sie sind der **algebraische Abschluss** der reelen Zahlen.

Komplexe Zahlen

Wir haben also $z = a + b \cdot i$ als komplexe Zahl, $z \in \mathbb{C}$, mit $a, b \in \mathbb{R}$
$\Rightarrow$ Man kann also z auf der **Zahlenebene** darstellen.





Die **Konjugation** von z, die wir als $\bar{z}$ bezeichnen wird oft verwendet.
$\bar{z}$ ist definiert als $\bar{z} = a - b \cdot i$

Komplexe Zahlen

Wir haben für $z \in \mathbb{C}$ die folgenden simplen Rechenregeln:

Sei $z_1 = a + b \cdot i$ und $z_2 = c + d \cdot i$ aus $\mathbb{C}$. Dann gilt:

$$z_1 + z_2 = (a + c) + (b + d)i$$

$$z_1 \cdot z_2 = (ac - bd) + (ad + bc)i$$

"Herleitung"

$$\begin{aligned} z_1 \cdot z_2 &= (a + bi) \cdot (c + di) = ac + bci + adi + bdi^2 = \\ &= ac + bci + adi - bd = (ac - bd) + (ad + bc)i \end{aligned}$$

Beachten Sie: $i^2 := -1$ (nach Definition).

Man kann leicht zeigen, dass $(\mathbb{C}, +, \cdot)$ ein Körper ist. (Buch Hartmann)

Komplexe Zahlen

Anmerkung:
Beachten muss man, dass man in $\mathbb{C}$ keine Totalordnung der Zahlen mehr hat. Es gilt nicht mehr $z_1 < z_2$, weil nicht mehr klar ist, wie $<$ genau definiert werden sollte.

Der Betrag einer Zahl ist aber noch sinnvoll zu definieren:
Wir haben: $|z| = |a + bi| = \sqrt{a^2 + b^2}$

Man kann zeigen: $|z|^2 = z \cdot \bar{z}$.

Anwendung: Reed Solomon Algorithm (RSA)

Allgemein

Reed Solomon Codes (RS-Codes) sind blockbasierte fehlerkorregierende Codes mit einem weiten Anwendungsfeld:

- in digitaler Kommunikation, z.B. bei verschiedenen Speichermedien wie CD's, DVD's und QR Codes
- aber auch High-speed Modems
- und Satelliten Kommunikation (z.B. Korrektur der Fehler durch Rauschen und einen weiten Übertragungsweg).

© ORF1

Allgemein

- RSA - Verschlüsselungsverfahren, besteht aus Encoder und Decoder.
- Reed-Solomon **Encoder** verwendet digitale Datenblöcke und fügt extra redundante Bits hinzu, da während der Übertragung Fehler entstehen können.
- Der Reed-Solomon **Decoder** empfängt die Daten und verarbeitet jeden Block anschliessend und überprüft, ob die Übertragungsfehler entstanden sind und versucht die Fehler (zu einem gewissen Grad) zu korrigieren.

Allgemein

- RS-Codes (Codieralgorithmus) wurden um 1960 von I.S. Reed und G. Solomon am MIT in der USA entwickelt und effizienten Decodieralgorithmus stellten 1969 E. Berlekamp und J. Massey vor.
- Erstmals angewandt wurden RS-Codes im Voyager-Programm der NASA im Jahr 1977. `https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19780022919.pdf`
- Erste kommerzielle Anwendung fanden sie 1982 bei der Fehlerkorrektur von CD's.

Mathematische Konzepte

Informationsübertragung mit Fehlerkorrektur ist möglich durch folgende Konzepte aus der Mathematik:

- Abbildungen,
- Galois Körper,
- Polynommultiplikation und Division.

Übersicht

0101 1101 1010 0011

Ausgangsdaten

Codierung →

1101 0110 1000 1101 0100 1000 0101 0011

Codierte Daten mit redundanten Bits

Quelle/Sender

Ziel/Empfänger

1101 0110 1001 1101 0100 1000 0101 0010

Codierte Daten mit Fehlern

Fehlerkorrektur ←

1101 0110 1000 1101 0100 1000 0101 0011

Codierte Daten nach Fehlerkorrektur

Dekodierung ↓

0101 1101 1010 0011

nach Decodierung

Z.B. Richtung $\rightarrow$ erstellt den QR Code, druckt ihn auf das Papier, $\leftarrow$ scannt den QR Code auf das Smartphone ein und korrigiert die Fehler.

Galois Körper im Reed Solomon Code

Wähle 3 Parameter:

- Alphabetgröße s,
- Zahl der Blöcke k (jeder mit s Bits),
- Fehlerrate t.

Relativ frei wählbar. Wichtig: $k + 2t \leq 2^s$

Alphabet für die Codierung sind die Elemente des Körpers $GF(2^s)$.

Galois Körper $\mathbb{F}_2$

Beispiel:

Wir haben bereits $\mathbb{F}_2$ der aus Elementen 0 und 1 besteht.

Die Verknüpfungen sind definiert als:

	$+$	$\cdot$
	$0+0=0$	$0 \cdot 0=0$
	$0+1=1$	$0 \cdot 1=0$
	$1+0=1$	$1 \cdot 0=0$
	$1+1=0$	$1 \cdot 1=1$

Ihr kennt $\mathbb{F}_2$ auch als $\mathbb{Z}/2\mathbb{Z}$.
Auch $\mathbb{F}_2$ ist ein Galois Körper, $GF(2)$.
Für Reed Solomon brauchen wir Körper mit mehr Elementen.
Oft wird $GF(2^8)$ verwendet mit 256 Elementen.
Die Blockgröße s ist ein Byte (8 Bit).

Galois Körper $\mathbb{F}_2$

auch Galois-Felder (englisch Galois fields),
benannt nach Evariste Galois (1811-1832).
= endlicher Körper, d. h. Körper mit endlich vielen Elementen
Beispiel $\mathbb{Z}_2 = \{0, 1\}$ (siehe oben) mit üblicher Multiplikation und Addition als XOR (exklusives Oder).

Beispiel:
$\mathbb{Z}_3 = \{0, 1, 2\}$ mit Addition und Multiplikation modulo 3:

+	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

*	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

Körper mit 2^n Elementen: $GF(2^n)$

für die Informatik interessant.
Die Elemente (Galois-Zahlen) werden als Bitfolgen der Länge n interpretiert.
Anwendungen in der Kodierungstheorie: Reed-Salomon Codes, Hamming-Codes

Bei RSA: $s = n$

Bemerkung:
Der im Folgenden vorgestellte Ansatz kann verallgemeinert werden auf Körper mit p^n Elementen, wobei $p \in \mathbb{N}$ eine beliebige Primzahl ist.

Körper mit 2^n Elementen: $GF(2^n)$

$=$ Rechnen mit Bitfolgen, sodass gängige Rechenregeln (Körperaxiome) erfüllt sind.

Addition:

Die Addition ist als bitweises XOR, d.h. exklusives Oder (bitweise Addition in $\mathbb{Z}_2$) definiert.

Beispiel:

$$1001 + 0101 = 1100$$

$$1101.0111 + 0011.1101 = 1110.1010$$

Eigenschaften:

- Addition ohne Übertrag!
- $(GF(2^n), +)$ ist eine abelsche Gruppe.
- Jedes Element ist zu sich selbst invers: $x + x = 0$.
  Konsequenz: $x - y = x + y$ ("Minus gleich plus")

Mögliche Darstellung von Bitfolgen $a_{n-1}a_{n-2}\dots a_0$

- als Dualzahl, Beispiel 10011101,
- als Polynom über $\mathbb{Z}_2$:

$$a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1x + a_0,$$

im Beispiel

$$x^7 + x^4 + x^3 + x^2 + 1.$$

Die Darstellung als Polynom wird benutzt, um Addition und Multiplikation von Galois-Zahlen zu definieren.
Addition von Polynomen über $\mathbb{Z}_2$:

$$(x^3 + x^2 + 1) + (x^2 + 1) = x^3 + (1+1)x^2 + (1+1) = x^3$$

entspricht bitweiser Addition in $GF(2^n)$
(hier mit $n = 4 : 1101 + 0101 = 1000$).

Mögliche Darstellung von Bitfolgen $a_{n-1}a_{n-2}\dots a_0$

Multiplikation von Polynomen

Beispiel in $GF(2^4)$:

$(x^3 + x^2 + 1).(x^2 + 1)$

$= x^5 + x^4 + x^2 + x^3 + x^2 + 1$

$= x^5 + x^4 + x^3 + 1$ (wegen $x^2 + x^2 = (1 + 1)x^2 = 0$)

Das Ergebnis (111001) ist zu groß, es stellt kein Element von $GF(2^4)$ dar!

Ausweg:

Modifikation der Multiplikation durch Benutzung von Modulpolynomen.

Polynomdivision in $\mathbb{Z}_2[x]$

Beispiel in $\mathbb{Z}_2[x]$

$$p(x) = x^4 + x + 1 = 10011, q(x) = x^2 + 1 = 101.$$

Es wird ausgenutzt, dass in $\mathbb{Z}_2$ gilt $+ = -$.

$$\begin{array}{l}
(\ x^4 \qquad\quad + x + 1) : (x^2 + 1) = x^2 + 1 \\
+x^4 + x^2 \qquad\qquad\qquad\qquad\quad \text{Rest } x \\
\hline
\qquad\quad x^2 + x + 1 \\
\qquad\ \ +x^2 \qquad + 1 \\
\hline
\qquad\qquad\quad x + 0
\end{array}$$

Polynomdivision in $\mathbb{Z}_2[x]$

Um Schreibarbeit zu sparen, können die Polynome als Bitfolgen (Galoiszahlen) dargestellt werden:

$$
\begin{array}{l}
(1\ 0\ 0\ 1\ 1) : (1\ 0\ 1) = 1\ 0\ 1 \text{ Rest } 1\ 0 \\
+1\ 0\ 1 \\
\hline
\ \ 0\ 0\ 1\ 1\ 1 \\
\ \ \ \ \ \ +1\ 0\ 1 \\
\hline
\ \ \ \ \ \ \ \ \ 0\ 1\ 0.
\end{array}
$$

Irreduzible Polynome

Man sagt, q ist Teiler von p (kurz $q|p$), wenn es ein Polynom $k \in \mathbb{K}[x]$ gibt $p(x) = k(x).q(x)$, d. h. der Rest bei der Division $p(x) : q(x)$ gleich 0 ist.
Ein Polynom $p \in \mathbb{K}[x]$ heißt irreduzibel, wenn es keinen Teiler q hat mit $0 < \deg q < \deg p$.
Andernfalls heißt p reduzibel.
Jedes normierte Polynom lässt sich eindeutig als Produkt von normierten irreduziblen Polynomen darstellen.
Bemerkungen:

- Diese Zerlegung entspricht der Darstellung einer ganzen Zahl als Produkt von Primzahlen in $\mathbb{K}$.
- Ist $p(x) = p_1(x) \dots p_k(x)$, so ist $\deg p(x) = \deg p_1(x) + \dots + \deg p_k(x)$.

Der Restklassenring $\mathbb{Z}_2[x]_{m(x)}$

Man betrachtet ein fest gewähltes Modulpolynom $m(x)$ mit Grad n. $\mathbb{Z}_2[x]_{m(x)}$ ist dann die Menge aller Polynomen über $\mathbb{Z}_2$ mit Grad $< n$, also

$$\mathbb{Z}_2[x]_{m(x)} = \{a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \dots + a_1 x + a_0; a_0, a_1 \dots a_{n-1} \in \mathbb{Z}_2\}.$$

$\mathbb{Z}_2[x]_{m(x)}$ hat 2^n Elemente.
Zu $p, q \in \mathbb{Z}_2[x]_{m(x)}$ ist auch die Summe $p + q = p - q$ ein Element von $\mathbb{Z}_2[x]_{m(x)}$.
Eine Multiplikation innerhalb von $\mathbb{Z}_2[x]_{m(x)}$ wird definiert durch

$$p(x).q(x) \mod m(x).$$

Die Benutzung des Modulo-Operators stellt sicher, dass das Ergebis wieder in $\mathbb{Z}_2[x]_{m(x)}$ liegt.

Der Restklassenring $\mathbb{Z}_2[x]_{m(x)}$

- Formal kann $\mathbb{Z}_2[x]_{m(x)}$ als Menge von Äquivalenzklassen konstruiert werden.
- Zu jedem $n \geq 2$ gibt es irreduzible Polynome vom Grad n, mit denen $GF(2^n)$ konstruiert werden kann.
- Analog kann zu jeder Primzahl p und jedem $n \geq 2$ ein Körper mit p^n Elementen konstruiert werden. Es gilt sogar: Es gibt genau dann einen endlichen Körper mit k Elementen, wenn k eine Primzahlpotenz ist, also $k = p^n$ mit einer Primzahl p und $n \in \mathbb{N}(n \geq 1)$.
- $GF(2^8)$ kann mit dem Modulpolynom

$$m(x) = x^8 + x^4 + x^3 + x + 1$$

definiert werden.

Bemerkungen

- Die Definition der Multiplikation in $GF(2^n)$ hängt von der Wahl des irreduziblen Modulpolynoms $m(x)$ ab, die für $n \geq 3$ nicht eindeutig ist. Unterschiedliche Modulpolynome bei gleichem n führen jedoch zu isomorphen algebraischen Strukturen.
- Die Elemente $\neq 0$ eines Galoiskörpers G bilden bezüglich der Multiplikation eine zyklische Gruppe, d.h. es gibt ein $p \in G$, sodass alle Elemente von $G \setminus \{0\}$ Potenzen von p sind. Jeder Galoiskörper hat somit die Form

$$GF(2^n) = \{0, 1, p, p^2, p^3, \ldots, p^{m-1}, p^m\} \text{ mit } m = 2^n - 2$$

  mit einem geeigneten p. Ein solches p heißt Erzeuger der multiplikativen Gruppe von $GF(2^n)$ und ist nicht eindeutig bestimmt. Diese Eigenschaft kann bei praktischen Rechnungen ausgenutzt werden.

Zurück zu Anwendung:

Reed-Solomon Codes

zur fehlertoleranten Datenübertragung. Dazu:

- Zerlege Daten in Blöcke zu jeweils $k.s$ Bit, wobei $k < 2^s$.
- Jeder Block definiert k Elemente $a_0, \ldots, a_{k-1} \in GF(2^s)$.
- $c(x) = \sum_{j=0}^{s} a_j x^j$ definiert nun ein Polynom über $GF(2^s)$, also eine Funktion $c : GF(2^s) \to GF(2^s)$.
- Berechne und übertrage $y_i = c(x_i)$ für jedes $x_i \in GF(2^s)$, d.h. es sind $m = 2^s$ Elemente $y_1, \ldots, y_m \in GF(2^s)$, also insgesamt $s2^s$ Bit zu übertragen.
- Sind nur k der y_i bekannt, so lassen sich die ursprünglichen Daten $a_0, \ldots, a_s$ daraus eindeutig berechnen (vgl. Newton Interpolation!). Ist also $m > k$, so können die Daten a_i auch dann noch korrekt rekonstruiert werden, wenn ein Teil der y_i verloren geht.

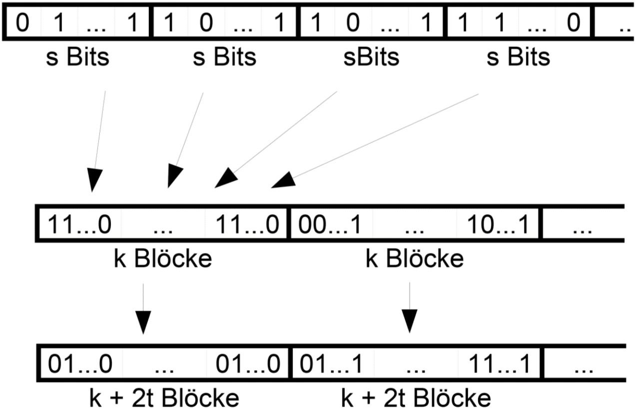
Beispiel

Beispiel mit $s = k = 2$
Zu übertragen sei die Datenblock 101011, der in drei Elemente von $GF(2^2)$ zerlegt wird: $a_0 = 10, a_1 = 10, a_2 = 11$.
Man betrachtet nun $c(x) = 11.x^2 + 10.x + 10$. Es gilt

x	00	01	10	11
$c(x)$	10	11	11	10

(z. B. $c(10) = 11.10^2 + 10.10 + 10 = 11.11 + 11 + 10 = 10 + 01 = 11$)
Übertragen werden die 8 Bit 10 11 11 10.
Empfangen wird z. B. 10 ?? 11 10. Mit Newton- oder Lagrange-Interpolation wird nun das Polynom $c(x)$ berechnet, womit a_0, a_1 und a_2 rekonstruiert werden können.

Übersicht Codierung



1. Eingabe des RS-Code: k Blöcke; 2. Erstellung von $k + 2t$ Blöcken, diese erhalten die redundante Information, die zur Decodierung notwendig ist.

Algorithmus Codierung (Grundprinzip)

1. Einteilung der Quelldaten in k Blöcke der Länge von je s Bits.
2. Bijektive Abbildung dieser Blöcke auf die Elemente von $GF(2^s)$.
3. Erzeugung eines Polynoms des Grads $k-1$ aus der Folge von Elementen
$$c(x) = a_{k-1}x^{k-1} + ... + a_1 x + a_0.$$
4. Erzeugung eines Polynoms vom Grad $2t$, man nennt es auch das Generatorpolynom:
$$g(x) = (x - \alpha^1) \cdot (x - \alpha^2) \cdot, \ldots \cdot (x - \alpha^{2t}),$$
α's sind Elementen des GF.
5. Erzeugung des Polynoms $d := c(x) \cdot g(x)$.
6. Abbildung der Koeffizienten von $d(x)$ auf die jeweiligen Bitfolgen (= das was auf das Speichermedium gespeichert wird bzw. durch einen Übertragungskanal gesendet wird).

Abbildung von Bitsequenzen zu Polynomkoeffizienten

ord	Polynomdarstellung	Tupel (Bitsequenz)
0	0	0000
α^0	1	1000
α^1	x	0100
α^2	x^2	0010
α^3	x^3	0001
α^4	$1 + x^3$	1001
α^5	$1 + x + x^3$	1101
α^6	$1 + x + x^2 + x^3$	1111
α^7	$1 + x + x^2$	1110
α^8	$x + x^2 + x^3$	0111
α^9	$1 + x^2$	1010
α^{10}	$x + x^3$	0101
α^{11}	$1 + x^2 + x^3$	1011
α^{12}	$1 + x$	1100
α^{13}	$x + x^2$	0110
α^{14}	$x^2 + x^3$	0011

$s = 4$, $\alpha^i \in GF(2^4)$, $|GF(2^4)| = 16$. Bitsequenzen sind als Polynome interpretiert. Bijektivität notwenig für Decodierung!

Algorithmus zur Codierung (Beispiel)

Hinweis: $0101 = \alpha^{10}$, $1101 = \alpha^{5}$, $1010 = \alpha^{9}$, $0011 = \alpha^{14}$, $c(x) = a_{k-1}x^{k-1} + \ldots . + a_1 x + a_0$.

Parameter s = 4, k = 4, t = 2 => $g(x) = (x-\alpha^1)(x-\alpha^2)(x-\alpha^3)(x-\alpha^4)$ Generatorpolynom vom Grad 2t = 4

0101110110100011

Eingabe:
Ausgangsdaten

→ 0101 1101 1010 0011

Einteilung in Blöcke zu je 4 Bits

→ $\alpha^{10}\ \alpha^{5}\ \alpha^{9}\ \alpha^{14}$

Abbildung auf Elemente von GF(2^4)

→ $c(x) = x^3\alpha^{14} + x^2\alpha^9 + x^1\alpha^5 + x^0\alpha^{10}$

Erzeugung eines Polynoms vom Grad k-1

→ $f(x) := c(x)\, g(x) = x^7\alpha^{14} + x^6\alpha^{10} + x^5\alpha^0 + x^4\alpha^1 + x^3\alpha^5 + x^2\alpha^0 + x^1\alpha^{13} + x^0\alpha^5$

Multiplikation dieses Polynoms mit dem Generatorpolynom

→ $\alpha^{5}\ \alpha^{13}\ \alpha^{0}\ \alpha^{5}\ \alpha^{1}\ \alpha^{0}\ \alpha^{10}\alpha^{14}$

Die Koeffizienten sind Elemente von GF(2^4)

→ 1101 0110 1000 1101 0100 1000 0101 0011

Abbildung auf die Bitfolge

Ausgabe:
Codierte Daten

Algorithmus Decodierung (Grundprinzip)

1. Lese die Zeichenfolge ein und interpretiere sie als Polynom $f(x)$.
2. Überprüfe auf Fehler: wenn keine Fehler, dann gilt $f(x) = d(x)$.
   Da $d(x)$ durch Multiplikation aus $c(x)$ und $g(x)$ entstanden ist und wenn kein Fehler vorliegt, gilt:
   $$\forall i = 1, .., 2t : f(\alpha^i) = 0.$$
   (α^i sind die Nullstellen von Polynom g.)
3. Fehlerkorrektur ist möglich bei maximal t Fehlern:
   $$f(x) = d(x) + e(x).$$
   Lösung des linearen Gleichungssystems.
4. Rekonstruktion von $d(x)$ aus $f(x) - e(x)$.
5. Erhalte die Originaldaten mittels Polynomdivision als $c(x) := d(x) : g(x)$.

Algorithmus zur Decodierung (Beispiel)

1101 0110 1001 1101 0100 1000 0101 1110

Eingabe: Codierte Daten evtl. mit Fehlern

$\alpha^5\ \alpha^{13}\ \alpha^4\ \alpha^5\ \alpha^1\ \alpha^0\ \alpha^{10}\alpha^7$

Die Koeffizienten sind Elemente von GF(2^4)

$d(x) = x^7\alpha^7 + x^6\alpha^{10} + x^5\alpha^0 + x^4\alpha^1 + x^3\alpha^5 + x^2\alpha^4 + x^1\alpha^{13} + x^0\alpha^5$

Stelle das entsprechende Polynom auf

$d(\alpha^i)$?= 0 für i = 1,…, 2t, hier t = 2

Prüfe auf Fehler

$f(x) = x^7\alpha^{14} + x^6\alpha^{10} + x^5\alpha^0 + x^4\alpha^1 + x^3\alpha^5 + x^2\alpha^0 + x^1\alpha^{13} + x^0\alpha^5$

Weniger als 2t Fehler: Korrektur möglich durch f(x) = d(x) + e(x)

$c(x) = x^3\alpha^{14} + x^2\alpha^9 + x^1\alpha^5 + x^0\alpha^{10}$

Erhalte c(x) durch Division f(x)/g(x)

0101 1101 1010 0011

Abbildung auf die Bitfolge

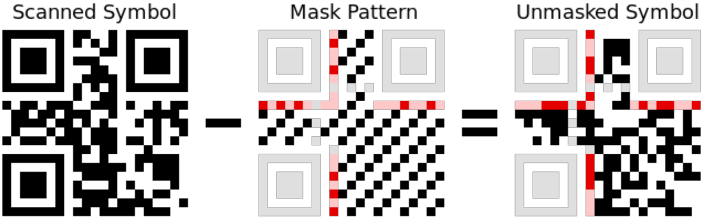
Ausgabe:
Rekonstruierte Daten ohne Fehler

Reed Solomon für QR Codes

QR Codes sind mit Hilfe von Reed-Salomon Code Verfahren erstellt.
QR Codes bestehen aus:

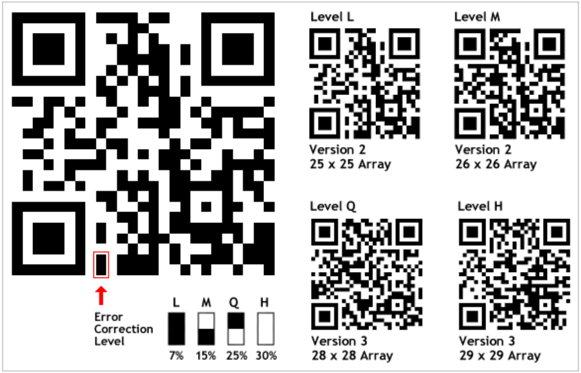
- schwarzen (1) und weissen (0) Quadraten (Interpretation: Bits),
- einem *locator pattern* zur Platzierung des Scanners.

QR Codes



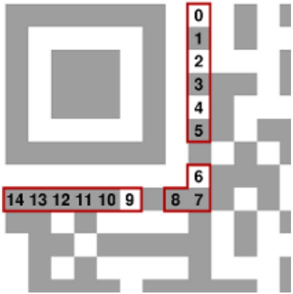
Masking: wird verwendet um Symbole zu maskieren welche den Scanner verwirren könnten, z.B. nehmen weisse Flächen die Stellen der "locator patterns" ein.
Die roten Flächen enkodieren Formatinformationen inklusive der Information welche Maske verwendet wurde und welches Fehler Korrektur Level verwendet wurde. Andere schwarz-weiss Bereiche beinhalten die Daten die der QR Code übertragen sollte.

QR Codes



Reed Solomon Fehlerkorrektur: Abhängig vom gewählten Fehler Korrektur Level kann auch bei beschädigtem QR Code die Gesamtheit der Daten rekonstruiert werden. Bei Level L darf z.B. 7% des QR codes beschädigt sein.

Lesen von QR Codes



Schwarze und weisse Quadrate entsprechen jeweils einem Bit.
Die markierte Sequenz der Bits 0-14 enthält Formatinformation. Von dieser Sequenz enthalten nur 5 Bit (10-14) tatsächlich Information. Der Rest ist für Fehlerkorrektur reserviert. Bits 14 und 13 enthalten das Fehler Korrektur Level und 12,11 und 10 enthalten die Masken ID.

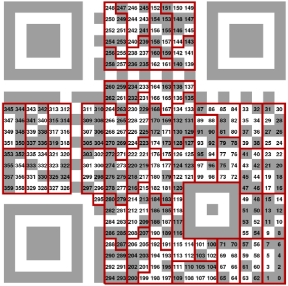
Lesen von QR Codes: Formatinformation

Die oben genannten 15 Bits kann man als Koeffizienten eines Polynoms in GF betrachten deren Koeffizienten entweder 0 oder 1 sind.

$$p(x) = x^{14} + x^{13} + x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x$$

Die eigentliche Information ist in den Termen 10 bis 14 enthalten.
$p(x)$ ist durch das Generatorpolynom $g(x)$ teilbar, damit ist Fehlerkorrektur möglich.

QR Codes: Inhalt



Gruppierungen von je 8 Bits werden zu Bytes zusammengefasst. Wichtig ist hierbei auch die Numerierung die die "Ableserichtung" anzeigt. Diese müssen beider der Sender und der Empfänger wissen.

Vektorräume - Einleitung

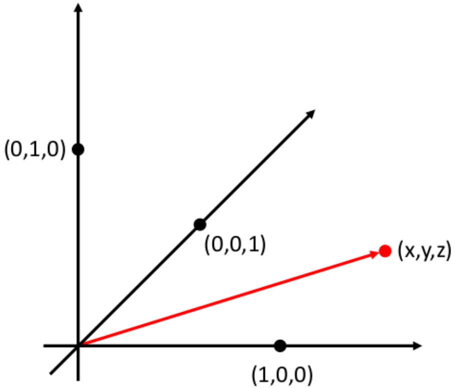
Einleitung

Algebraische Strukturen sind eine sehr mathematische Konstruktion, d.h. man kann damit rechnen, aber man kann sich wenig drunter vorstellen.

> „In der Mathematik versteht man die Dinge nicht.
> Man gewöhnt sich nur an sie.“
>
> – John von Neumann
> (ungarisch-US. Mathematiker des 20. Jh.)

Vektoren und Vektorräume hingegen sind sehr viel einfacher, schlicht weil sie unserem alltäglichen Verständnis entsprechen. Unser drei-dimensionaler Raum kann man einfach als $\mathbb{R}^3$ verstehen und eine Landkarte als $\mathbb{R}^2$.

Vektorraum



Vektorraum

Anmerkungen:
Diese Art Koordinaten aufzuzeichnen nennt man „kartesisches Koordinatensystem“ (René Descartes).

Man muss sich nicht auf $\mathbb{R}^2$ oder $\mathbb{R}^3$ beschränken, man kann auch mit $(3, 1, 4, 1, 5, 9, 2, 6, 5, \dots) \in \mathbb{R}^{34}$ arbeiten. Für gegebene n schreiben wir $(x_1, x_2, \dots, x_n) \in \mathbb{R}^n$. Explizit vorstellen kann man es sich halt nicht mehr.

Man kann statt $(1, 2, 3, \dots)$ auch $\begin{pmatrix} 1 \\ 2 \\ 3 \\ \vdots \end{pmatrix}$ schreiben. Das ist Geschmackssache (zumindest fast).

Vektorraum

Man kann auf Vektorräumen natürlich auch Verknüpfungen definieren:

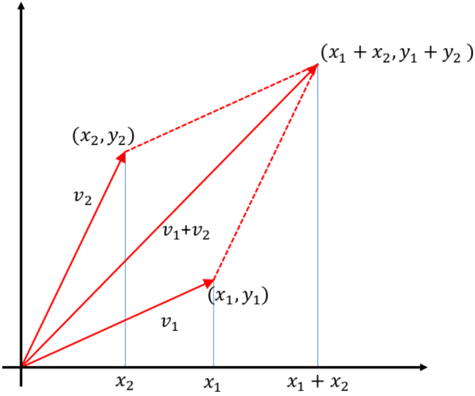
Vektoraddition:

$$v_1 + v_2 = \begin{pmatrix} x_1 \\ y_1 \\ z_1 \\ \vdots \end{pmatrix} + \begin{pmatrix} x_2 \\ y_2 \\ z_2 \\ \vdots \end{pmatrix} := \begin{pmatrix} x_1 + x_2 \\ y_1 + y_2 \\ z_1 + z_2 \\ \vdots \end{pmatrix}$$

Mit einem neutralen Element $(0, 0, 0, \dots)$ bildet das dann eine Gruppe.
$\Rightarrow$ Übungsaufgabe: Beweisen Sie das !

Man kann die Vektoraddition so verstehen, dass der „Pfeil" v_2 an den „Pfeil" v_1 dran gehängt wird. Das Ganze kann man sehr schön graphisch darstellen:

Vektorraum



Vektorraum

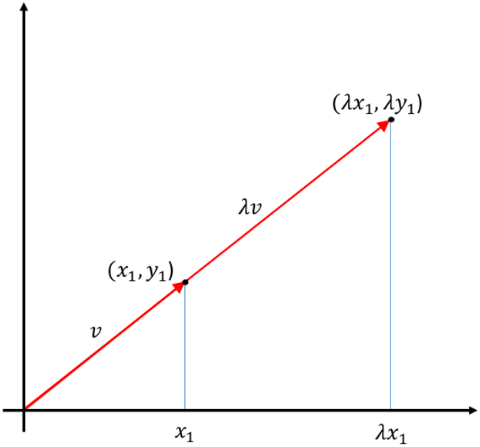
Eine andere sehr wichtige Verknüpfung ist die **Skalarmultiplikation**. Skalarmultiplikation ist die Streckung des Vektors um einen Faktor c ($c \in \mathbb{R}$).

Definiert wird das so:

$$\lambda v_1 = \lambda \begin{pmatrix} x_1 \\ y_1 \\ z_1 \\ \vdots \end{pmatrix} := \begin{pmatrix} \lambda x_1 \\ \lambda y_1 \\ \lambda z_1 \\ \vdots \end{pmatrix}$$

Wenn $|\lambda| < 1$ ist, dann wird natürlich der Vektor nachher „kürzer" sein als vorher, dh $|\lambda v_1| < |v_1|$.

Vektorraum



Vektorraum

Wir haben die Verknüpfungen kennen gelernt und nun die Definition:

Definition:
Sei K ein Körper. Ein **Vektorraum** V mit Skalaren aus K besteht aus einer kommutativen Gruppe $(V, +)$ und einer skalaren Multiplikation $\cdot : K \times V \to V, (\lambda, v) \mapsto \lambda \cdot v$, sodass für alle $v, w \in V$, $\lambda, \mu \in K$ gilt:

(V1) $\lambda(\mu v) = (\lambda\mu)v$

(V2) $1 \cdot v = v$

(V3) $\lambda(v + w) = \lambda v + \lambda w$

(V4) $(\lambda + \mu)v = \lambda v + \mu v$.

Vektorraum

Anmerkungen:
Wir sehen, dass wir hierfür immer einen Körper K brauchen um von einem Vektorraum sprechen zu können. Wenn $K = \mathbb{R}$ ist, sprechen wir von einem **reellen Vektorraum**, wenn $K = \mathbb{C}$ gilt, von einem **komplexen Vektorraum**. Man sagt dann z.B. Vektorraum über $\mathbb{R}$ oder $\mathbb{R}$-Vektorraum und schreibt $V(\mathbb{R})$.

Man könnte natürlich auch ganz andere Körper verwenden, z. B. den Galois Körper $v \in V(GF(2))$ darstellen, z.B. $v = (0, 1, 1, 1, 0, 0, 0, 1)$.

Vektorraum

Definition:
Sei V ein K-Vektorraum und $U \subset V$ mit $U \neq \emptyset$. Ist U mit den Verknüpfungen von V selbst wieder ein K-Vektorraum, so heißt U **Untervektorraum** von V. Man sagt auch **Teilraum** oder **Unterraum**.

Satz:
Sei V ein K-Vektorraum und $U \subset V$, $U \neq \emptyset$. U ist genau dann ein Teilraum von V, wenn gilt:

(U1) $u + v \in U$ für alle $u, v \in U$

(U2) $\lambda u \in U$ für alle $\lambda \in K, u \in U$

Diese Bedingungen bedeuten lediglich, dass der Teilraum abgeschlossen ist bzgl. der Verknüpfungen.
Beweis (voraussichtlich) in der Übung.

Vektorraum

Beispiel:
Betrachten wir das ganze anhand des $V = \mathbb{R}^2$. Ein Teilraum U ist uninteressant, wenn er nichts enthält (d.h. $U = \emptyset$), also haben wir $u \in U$. Es muss gelten $\lambda u \in U$ für alle $\lambda \in \mathbb{R}$, d.h. alle Vektoren, die die gleiche Richtung wie u haben, in U liegen.
Wenn wir einen zweiten Vektor $v \in U$ haben, dann ist auch $\mu v \in U$ und nach $(U1)$ auch $\lambda u + \mu v$.

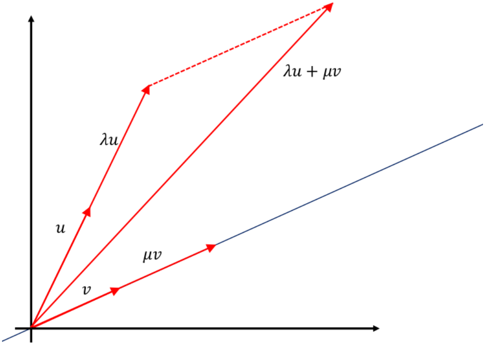
Definition:
Für beliebige u_i, λ_i gilt:

$$\lambda_1 u_1 + \lambda_2 u_2 + \lambda_3 u_3 + \dots$$

ist eine **Linearkombination** von u_i. Und die Menge aller solcher Linearkombinationen ist $\textbf{Span}(u_1, u_2, u_3, \dots)$.

Vektorraum

Beispiel:
Der Span(u_i) bildet immer einen Teilraum von V
Man nennt den Span auch oft **Lineare Hülle**.



Literatur

- Hartmann Peter, Kap. 5.3 (Körper), Kap. 5.4 (Polynomdivision) Kap. 6 (Vektorräume), insbesondere Kap. 6.3 (Lineare Abbildungen)
- Ausarbeitung *Endliche Körper* von Steffen Lohrke, FH Wedel
- Buch *Reed Solomon Codes and Their Applications* von Stephen B. Wicker, Vijay K. Bhargava, Einführungskapitel online verfügbar über citeSeerX
- Buch Gerald Teschl, Gerald, Teschl, Susanne: Mathematik für Informatiker, Band 1, Diskrete Mathematik und Lineare Algebra, 4. Auflage.