11.1 Übertragung von Resultaten aus $\mathbb{R}[x]$

In § 9 haben wir verschiedene Resultate in $\mathbb{R}[x]$ gefunden:

- Horner-Schema
- Polynomdivision
- Abspaltung von Nullstellen
- Zahl der Nullstellen eines Polynoms n-ten grades
- Teilbarkeit, gemeinsamer Teiler, ggT
- Euklidischer Algorithmus

Diese Resultate gelten allgemein für jeden Polynomring $K[x]$ über einem Körper K.

11.2. Beispiele

c) Wir betrachten das Polynom $p(x) = 4x^2 + 3x - 1$ im Körper $(\mathbb{Z}_5, +, \cdot)$. Wegen

$$p([0]) = [4]\cdot[0]^2 + [3][0] - [1] = [4]$$
$$p([1]) = [4]\cdot[1]^2 + [3]\cdot[1] - [1] = [1]$$
$$p([2]) = [4]\cdot[2]^2 + [3]\cdot[2] - [1] = [3]$$
$$p([3]) = [4]$$
$$p([4]) = [0]$$

hat p in $\mathbb{Z}_5$ eine Nullstelle in $[4]$.

b) Polynomdivision in $(\mathbb{Z}_3, +, \cdot)$:

+	[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]

$$([2]x^3 + [2]x + [1]) : (x + [2]) = [2]x^2 + [2]x + [1] \quad \text{Rest } [2]$$
$$-([2]x^3 + x^2)$$
$$[2]x^2 + [2]x$$
$$-([2]x^2 + x)$$
$$x + [1]$$
$$-(x + [2])$$
$$[2]$$

Probe: $[2] + ([2]x^2 + [2]x + [1])(x + [2])$

$= [2] + [2]x^3 + x^2 + [2]x^2 + x + x + [2]$

$= [2]x^3 + [2]x + [1]$ ✓

21/11/03

11.3. Anwendung der Polynomdivision in $\mathbb{Z}_2[x]$ zur Datensicherung

Bei der Datenübertragung können Bits „umklappen".

Einfachste Abhilfe zur Fehlererkennung: Einführung von zusätzlichen Prüfbits

Bsp.: Datenblock von 1 Byte, an den ein Prüfbit angehängt wird, dass Summe der Bits modulo 2 berechnet:

1 1 0 1 1 0 0 1 | 1

Byte | Prüfbit

Empfängt man 1 1 0 1 1 0 1 1 | 1, muss also ein Fehler aufgetreten sein.

Nachteile:

- Man muss $\frac{1}{8}$ mehr Daten übertragen.
- Keine Fehlermeldung, wenn 2 Bits umklappen.

Alternativ kann man Polynomdivision in $\mathbb{Z}_2[x]$ zur Datensicherung verwenden. Vorgehensweise:

1. Interpretiere Bits eines zu übertragenden Datenblocks als Polynom $f(x) \in \mathbb{Z}_2[x]$.

   Bsp.: Byte als Datenblock: 11011001

   Polynom: $f(x) = x^7 + x^6 + x^4 + x^3 + 1$

2. Ein festes „Generatorpolynom" $g(x) \in \mathbb{Z}_2[x]$ mit $\deg(g) = n$ dient als Divisor. Typischerweise ist $\deg(g) \ll \deg(f)$.

3. Betrachte statt $f(x)$ das Polynom $h(x) = x^n \cdot f(x)$.

   (D.h. in der Bitfolge zu $f(x)$ werden n Nullen angehängt.)

   Bsp.: $n = 4$: $\underbrace{\underbrace{11011001}_{f(x)}\,\underbrace{0000}_{x^n}}_{h(x)}$

4. Berechne Polynomdivision $h(x) : g(x)$ in $\mathbb{Z}_2[x]$.

   $$h(x) = q(x)\,g(x) + r(x)$$

   mit $\deg(r) < n$.

5. Sende $h(x) - r(x) = q(x)\,g(x)$. $(= h(x) + r(x)$ in $\mathbb{Z}_2[x])$

   Wegen $\deg(r) < n$ unterscheiden sich $h(x) - r(x)$ und $h(x)$ höchstens in den letzten n Bits:

   $$\underbrace{\underbrace{11011001}_{f(x)}\,\underbrace{1011}_{r(x)}}_{h(x)-r(x)}$$

   $r(x)$ dient der Datensicherung.

6. Der Empfänger erhält das Polynom $p(x)$
Tritt bei Division durch $g(x)$ ein Rest auf, so ist $p(x) \neq h(x) - r(x)$, d.h. ein Übertragungsfehler ist aufgetreten.

Bei geschickter Wahl von $g(x)$

- ist es sehr unwahrscheinlich, dass bei einem Übertragungsfehler die Division $p(x) : g(x)$ ohne Rest aufgeht.
- kann man aus dem Divisionsrest den Fehler lokalisieren und korrigieren.

Konkrete Spezifikation im X.25 - Übertragungsprotokoll:

- Datenblock hat 4096 Byte = 32.768 Bit
  $\Rightarrow \deg(f) = 32.767$
- Generatorpolynom: $g(x) = x^{16} + x^{12} + x^5 + 1$
  $\Rightarrow \deg(g) = 16$ (2 Byte zur Datensicherung)

Es werden also nur $\frac{2}{4096} \approx 0{,}49$ Promille zusätzliche Daten übertragen. Erkannt werden können u.A. alle 1-, 2- und 3-Bit-Fehler sowie alle Fehler mit ungerader Fehlerzahl!

Bem.: Algorithmen zur Polynomdivision in $\mathbb{Z}_2[x]$ lassen sich effizient in Soft- und Hardware realisieren.