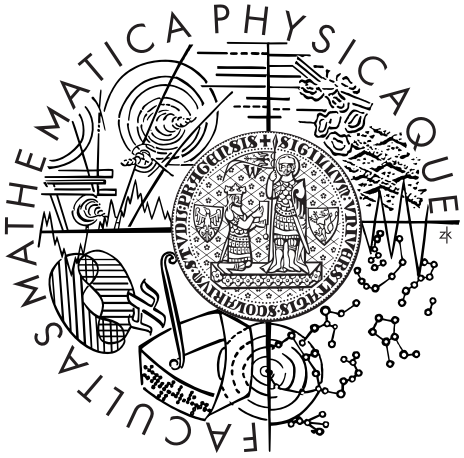
Univerzita Karlova v Praze

Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Jan Fišer

Řetězové zlomky v teorii kódů

Katedra algebry

Vedoucí bakalářské práce:	RNDr. Jan Šťovíček, Ph.D.
Studijní program:	Matematika
Studijní obor:	Matematické metody informační bezpečnosti

Praha 2014

Děkuji panu RNDr. Janu Šťovíčkovi, Ph.D. za odborné vedení, vstřícnost a čas, který věnoval kontrole této práce.

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Praze dne 22. května 2014

Název práce: Řetězové zlomky v teorii kódů

Autor: Jan Fišer

Katedra: Katedra algebry

Vedoucí bakalářské práce: RNDr. Jan Šťovíček, Ph.D., Katedra algebry

Abstrakt: Tato práce nás nejprve seznamuje s Reed-Solomonovy kódy, způsoby jejich konstrukce a kódování. Zároveň uvádíme důkazy jejich nejvýznamnějších vlastností s příslušným teoretickým základem. Ve druhé kapitole zavádíme pojem řetězových zlomků nad tělesem a zkoumáme jejich strukturu. Aplikací provedených obecných pozorování na konkrétní případ Laurentových formálních řad se pak dostáváme k účinnému dekódovacímu algoritmu Reed-Solomonových kódů. Bez kompletních důkazů uvádíme ještě další dva dekódovací algoritmy, které jsou také založeny na řešení klíčové rovnice, a to Berlekampův-Masseyův a Eukleidův. V závěru práce ukazujeme ekvivalenci těchto tří algoritmů.

Klíčová slova: Reed-Solomonovy kódy, řetězové zlomky, ekvivalence dekódovacích algoritmů

Title: Continued fractions in coding theory

Author: Jan Fišer

Department: Department of Algebra

Supervisor: RNDr. Jan Šťovíček, Ph.D., Department of Algebra

Abstract: The first part of the thesis acquaints us with the Reed-Solomon codes, methods of their construction and encoding. At the same time we provide the evidence of their most important properties including the relevant theoretical basis. In the second chapter we introduce the theory of continued fractions over a field and examine their structure. Applying the executed general observations on the specific case of the formal Laurent series we get to efficient Reed-Solomon decoding algorithm. Without complete proofs we also mention other two decoding algorithms that are based on solving the key equation as well, namely Berlekamp-Massey and Euclidean algorithm. In the end we show the equivalence of these three algorithms.

Keywords: Reed-Solomon codes, Continued fractions, Equivalence of decoding algorithms

Obsah

Úvod

Digitální data nás dnes obklopují takřka na každém kroku – ať již voláme známému, posíláme email nebo sledujeme doma televizi. Tato data jsou přenášena z jednoho místa do druhého – od vysílače informačním kanálem až k příjemci. Může se však stát, že se do posílané zprávy při průchodu kanálem přimísí chyba či se nějaká část zprávy jednoduše ztratí. V takovém případě bychom pak byli rádi, aby příjemce dokázal tyto chyby najít a opravit. A tuto otázku řeší právě samoopravné kódy.

V této práci se budeme zabývat rodinou Reed-Solomonových samoopravných kódů (zkráceně RS kódů). Od roku 1960, kdy byly poprvé zveřejněny Irvingem S. Reedem a Gustavem Solomonem v článku „Polynomial codes over certain finite fields“ (Polynomiální kódy nad určitými konečnými tělesy) [3], již našly mnoho uplatnění – od CD, přes digitální televizi až k satelitům ve vesmíru.

Především se pak zaměříme na způsob dekodování RS kódů pomocí teorie řetězových zlomků, se kterými se obvykle setkáváme pouze v rámci aproximace reálných čísel. Zde však podobnou ideu přeneseme od celých čísel do mocninných řad, čímž získáme velice užitečný nástroj k řešení otázky dekódování RS kódů.

Závěrečná kapitola se věnuje porovnání dekódovacích algoritmů (Berlekampova-Masseyova, Eukleidova a algoritmu s řetězovými zlomky) z hlediska postupu výpočtu. Nakonec ukážeme, že ačkoli se mohou zdát tyto tři algoritmy na první pohled odlišné, jsou v jistém smyslu ekvivalentní.

Kapitola 1

Základní pojmy z teorie kódů

Představme si v současné době běžnou situaci – máme nějakou informaci, kterou chceme prostřednictvím vybraného komunikačního kanálu poslat někomu jinému. Tímto kanálem může být například kabelové či rádiové spojení nebo zálohovací médium. Společnou vlastností všech kanálů je tzv. chybovost, která, zjednodušeně řečeno, značí pravděpodobnost výskytu chyby v přenosu.

Danou informaci je nejdříve nutné převést[1] do symbolů kódu. Výsledný řetězec symbolů následně můžeme poslat přes komunikační kanál, ale s jistou pravděpodobností se bude přijatý řetězec od odeslaného lišit (tj. nastane chyba), a tak bude zpráva pro příjemce nečitelná.

V tomto místě přicházejí na řadu samoopravné kódy. Řetězec původní zprávy nejprve rozdělíme do bloků délky k a každý tento blok rozšíříme na délku n ($\geq k$) způsobem předepsaným zvoleným samoopravným kódem. *Kódováním* budeme v této práci rozumět právě tento proces. Rozšířením každého bloku o $n-k$ symbolů docílíme schopnosti opravy či detekce určitého počtu chyb (na úkor vyšší datové náročnosti).

Příjemce poté obdrží zprávu zakódovanou jemu známým samoopravným kódem. Následuje proces *dekódování*, což zde znamená nalezení nejbližšího kódového slova ke slovu přijatému.[2] Pokud při průchodu zprávy kanálem došlo nejvýše k takovému počtu chyb, jež je kód schopen opravit, obdrží příjemce odesílanou zprávu v původním znění.

Základní pojmy teorie kódů nyní shrňme v následujících definicích a pozorováních.

Definice 1.1. *Nechť F je neprázdná konečná množina, $n \in \mathbb{N}$.* Blokový kód *délky n je libovolná neprázdná podmnožina $\mathcal{C} \subseteq F^n$. Prvky F^n nazýváme* slova *a prvky $\mathcal{C}$* kódová slova *nad abecedou F.*

Definice 1.2. *Je-li F q-prvkové konečné těleso, $n \in \mathbb{N}$ a $\mathcal{C}$ podprostor vektorového prostoru F^n dimenze $k \geq 0$, pak $\mathcal{C}$ nazýváme* lineární $[n,k]$ kód*, příp.* $[n,k]_q$ kód.

[1] Tento převod je označován jako kódování. Pojem kódování v tomto textu však budeme užívat pro převod vstupního slova na kódové slovo (viz dále).

[2] Nejbližší kódové slovo chápeme ve smyslu Hammingovy vzdálenosti (přesné definice následují po tomto odstavci).

Definice 1.3. *Buď $\mathcal{C}$ lineární $[n,k]$-kód. Pak matici $\mathbf{C}$ typu $k \times n$ nazýváme* generující maticí *kódu $\mathcal{C}$, obsahuje-li v řádcích právě bázi $\mathcal{C}$.*

Matici $\mathbf{H}$ typu $(n-k) \times n$ nazýváme kontrolní maticí *kódu $\mathcal{C}$, splňuje-li podmínku*

$$\boldsymbol{u} \in \mathcal{C} \Leftrightarrow \mathbf{H}\boldsymbol{u}^T = \mathbf{0}^T.$$

Pozorování 1.1. *Je-li $\mathbf{C}$ generující matice lineárního $[n,k]$ kódu $\mathcal{C}$, pak matice $\mathbf{H}$ typu $(n-k) \times n$ je jeho kontrolní maticí, právě když $\mathbf{C}\mathbf{H}^T = \mathbf{0}_{k\times(n-k)}$.*

V celé práci se věnujeme pouze lineárním kódům, proto nebudou některá tvrzení či definice vysloveny ve zcela obecném znění, ale budou zfromulovány pouze pro lineární kódy.

Definice 1.4. *Buď F těleso, $n \in \mathbb{N}$, $\boldsymbol{u} = (u_1, \ldots, u_n)$, $\boldsymbol{v} = (v_1, \ldots, v_n) \in F^n$.* Hammingovu vzdálenost *$d(\boldsymbol{u}, \boldsymbol{v})$ slov $\boldsymbol{u}$ a $\boldsymbol{v}$ definujeme jako počet souřadnic, ve kterých se tato slova liší, tj.*

$$d(\boldsymbol{u}, \boldsymbol{v}) = |\{i \in \{1, \ldots, n\} \mid u_i \neq v_i\}|.$$

Definice 1.5. *Buď $\mathcal{C}$ lineární $[n,k]_q$ kód, $k \geq 1$. Potom $d(\mathcal{C}) = \min\{d(\boldsymbol{u}, \boldsymbol{v}) \mid \boldsymbol{u}, \boldsymbol{v} \in \mathcal{C}, \boldsymbol{u} \neq \boldsymbol{v}\}$ nazýváme* minimální vzdálenost *kódu $\mathcal{C}$.*
Pro $k = 0$ definujeme $d(\mathcal{C}) = n + 1$. Označíme-li $d = d(\mathcal{C})$, mluvíme o $[n,k,d]$ (resp. $[n,k,d]_q$) kódu $\mathcal{C}$.

Definice 1.6. *Buď F těleso, $n \in \mathbb{N}$, $\boldsymbol{u} = (u_1, \ldots, u_n) \in F^n$.* Hammingovu váhu *$w(\boldsymbol{u})$ slova $\boldsymbol{u}$ definujeme jako počet nenulových souřadnic tohoto slova, tj.*

$$w(\boldsymbol{u}) = |\{i \in \{1, \ldots, n\} \mid u_i \neq 0\}| = d(\boldsymbol{u}, \mathbf{0}),$$

kde $\mathbf{0} = (0, \ldots, 0) \in F^n$.

Pozorování 1.2. *Hammingova vzdálenost d je metrika na F^n, kde F je těleso a $n \in \mathbb{N}$.*

Ústřední vlastností samoopravných kódů je schopnost nalezení a opravy t chyb. Možné hodnoty t jsou přesně určeny minimální vzdáleností kódu. Na druhou stranu minimální vzdálenost kódu je zase omezena rozdílem délky bloku a dimenze kódu, a to tzv. *Singletonovým odhadem.*

Pro vektor $\boldsymbol{u}$ z vektorového prostoru F^n, kde F je těleso a $n \in \mathbb{N}$, definujeme kouli se středem $\boldsymbol{u}$ a poloměrem $r \geq 0$ jako

$$S(\boldsymbol{u}, r) = \{\boldsymbol{v} \in F^n \mid d(\boldsymbol{u}, \boldsymbol{v}) \leq r\}.$$

Definice 1.7. *Buď $\mathcal{C}$ lineární kód, $r \in \mathbb{N}$. Řekneme, že $\mathcal{C}$* detekuje (rozpozná) *r* chyb*, jestliže pro všechna $\boldsymbol{u} \in \mathcal{C}$ platí $S(\boldsymbol{u}, r) \cap \mathcal{C} = \{\boldsymbol{u}\}$.*
$\mathcal{C}$ opraví *r* chyb*, jestliže pro všechna $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{C}$ platí $S(\boldsymbol{u}, r) \cap S(\boldsymbol{v}, r) = \emptyset$.*

Tvrzení 1.1. *Buď $\mathcal{C}$ lineární kód s minimální vzdáleností d a buď $r \in \mathbb{N}$, pak platí*

(i) $\mathcal{C}$ detekuje r chyb právě tehdy, když $r < d$,
(ii) $\mathcal{C}$ opraví r chyb právě tehdy, když $2r < d$.

Důkaz. ad (i) Kód $\mathcal{C}$ detekuje r chyb, právě když vzdálenost libovolných dvou kódových slov je větší než r, což je ekvivalentní tomu, že minimální vzdálenost kódu je větší než r.

ad (ii) Sporem dokážeme implikaci: $\mathcal{C}$ opraví r chyb $\Rightarrow 2r < d$.
Nechť $2r \geq d$, tedy existují kódová slova $\boldsymbol{u}$ a $\boldsymbol{v}$, jejichž vzdálenost je menší než $2r$. Uvažujme slovo $\boldsymbol{w} \in F^n$ ve vzdálenosti $\left\lceil \frac{d}{2} \right\rceil$ od $\boldsymbol{u}$ a ve vzdálenosti $\left\lfloor \frac{d}{2} \right\rfloor$ od $\boldsymbol{v}$, tedy $d(\boldsymbol{u}, \boldsymbol{w}) = \left\lceil \frac{d}{2} \right\rceil$ a $d(\boldsymbol{v}, \boldsymbol{w}) = \left\lfloor \frac{d}{2} \right\rfloor$. Takové slovo lze sestrojit například ze slova $\boldsymbol{u}$ nahrazením poloviny souřadnic, v nichž se slova $\boldsymbol{u}$ a $\boldsymbol{v}$ liší, příslušnými souřadnicemi slova $\boldsymbol{v}$. Je-li $d \leq 2r$, pak v celých číslech platí $\left\lceil \frac{d}{2} \right\rceil \leq r$. Uvážíme-li nyní koule o poloměru r se středy ve slovech $\boldsymbol{u}$ a $\boldsymbol{v}$, zjistíme, že $\boldsymbol{w}$ leží v jejich průniku, což je ve sporu s předpokladem, že $\mathcal{C}$ opraví r chyb.

Podobně sporem dokážeme i opačnou implikaci. Nechť neplatí, že $\mathcal{C}$ opraví r chyb, tedy že existuje slovo $\boldsymbol{w} \in F^n$ a kódová slova $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{C}$ taková, že $\boldsymbol{w}$ leží v průniku koulí $S(\boldsymbol{u}, r)$ a $S(\boldsymbol{v}, r)$. Využitím toho, že Hammingova vzdálenost je metrikou a platí na ní tedy trojúhelníková nerovnost, získáváme $d \leq d(\boldsymbol{u}, \boldsymbol{v}) \leq d(\boldsymbol{u}, \boldsymbol{w}) + d(\boldsymbol{v}, \boldsymbol{w}) = r + r = 2r$. To je spor s předpokladem $2r < d$. □

Tvrzení 1.2 (Singletonův odhad)**.** *Existuje-li lineární $[n, k, d]_q$ kód, potom platí $d \leq n - k + 1$.*

Důkaz. Buď $\mathcal{C} \subseteq F^n$ takový $[n, k, d]_q$ kód. Zřejmě všechna kódová slova $\mathcal{C}$ jsou po dvou různá. Vytvořme nový kód $\mathcal{C}'$, jenž budou tvořit kódová slova kódu $\mathcal{C}$, jimž odstraníme prvních $d - 1$ souřadnic. Slova vzniklá odstraněním $d - 1$ souřadnic budou stále po dvou různá, neboť Hammingova vzdálenost každé dvojice kódových slov kódu $\mathcal{C}$ byla minimálně d. Tedy velikost kódu $\mathcal{C}'$ je rovna velikosti kódu $\mathcal{C}$. Připomeňme ještě, že $\mathcal{C}' \subseteq F^{n-(d-1)}$ a $|F| = q$. Proto platí

$$q^k = |\mathcal{C}| = |\mathcal{C}'| \leq |F^{n-(d-1)}| = q^{n-(d-1)}.$$

□

Význačnou skupinu tvoří kódy, jejichž minimální vzdálenosti uvedenou nerovnost maximalizují:

Definice 1.8. *$[n, k, d]$ kód nazýváme* MDS (maximum distance separable)*, jestliže $d = n - k + 1$.*

Vyslovme ještě tvrzení o tom, jak lze z kontrolní matice kódu vypozorovat jeho minimální vzdálenost. Tato skutečnost nám v další kapitole pomůže jednoduše ukázat, že Reed-Solomonovy kódy jsou MDS.

Tvrzení 1.3. *Buď $\mathcal{C}$ lineární $[n, k, d]$ kód s kontrolní maticí $\mathbf{H}$. Pak $d - 1$ je největší $r \in \mathbb{N}_0$, pro které je každých r sloupců matice $\mathbf{H}$ lineárně nezávislých.*

Důkaz. Je-li největší takové r rovno nule, pak $\mathbf{H}$ musí obsahovat nulový sloupec, a tedy $d = 1$. Nechť je největší takové $r \geq 1$. Kód dimenze 0 (obsahující pouze nulové slovo) má za kontrolní matici regulární čtvercovou matici řádu n, proto

$r = n$. Minimální vzdálenost takového kódu je definována jako $n+1$ a tvrzení tedy platí. Máme-li kód nenulové dimenze, uvažujme slovo $\boldsymbol{u} = (u_1, \ldots, u_n) \in \mathcal{C}$, jehož váha je rovna minimální vzdálenosti kódu (takové slovo určitě existuje). Označme $J = \{i \in \{1, \ldots, n\} \mid u_i \neq 0\}$ a sloupce kontrolní matice $\boldsymbol{h}_1^T, \ldots, \boldsymbol{h}_n^T$. Protože platí $\mathbf{H}\boldsymbol{u}^T = \mathbf{0}^T$, je i $\sum_{i \in J} \boldsymbol{h}_i^T u_i = 0$, čímž jsme nalezli d lineárně závislých sloupců matice $\mathbf{H}$. Tedy $d-1 \geq r$. Současně však existuje slovo $\boldsymbol{v}$ váhy právě $r+1$ takové, že $\mathbf{H}\boldsymbol{v}^T = \mathbf{0}^T$, což znamená, že $\boldsymbol{v} \in \mathcal{C}$, a tedy $d \leq r + 1$. □

1.1 Cyklické kódy

Definice 1.9. *Kód $\mathcal{C}$ se nazývá cyklický, jestliže pro každé slovo $(c_1, \ldots, c_n) \in \mathcal{C}$ je i $(c_n, c_1, \ldots, c_{n-1}) \in \mathcal{C}$.*

Abychom mohli lépe uchopit problematiku cyklických kódů, definujeme pro konečné těleso F a $n \in \mathbb{N}$

$$F[x]_n = \left\{ \sum_{i=0}^{n-1} c_i x^i \mid c_i \in F \right\}$$

s operacemi $+$, $-$ a násobení prvkem $a \in F$ po složkách:

$$\sum_{i=0}^{n-1} c_i x^i + \sum_{i=0}^{n-1} d_i x^i = \sum_{i=0}^{n-1} (c_i + d_i) x^i,$$

$$-\sum_{i=0}^{n-1} c_i x^i = \sum_{i=0}^{n-1} -c_i x^i,$$

$$a \cdot \sum_{i=0}^{n-1} c_i x^i = \sum_{i=0}^{n-1} a c_i x^i.$$

Zobrazení $b : F^n \to F[x]_n$, $b((c_0, \ldots, c_{n-1})) = \sum_{i=0}^{n-1} c_i x^i$, je zřejmě bijekce a s právě definovanými operacemi je b izomorfismus vektorových prostorů. Této skutečnosti využijeme v několika důkazech, kdy budeme podle potřeby uvažovat místo vektoru polynom, na který se bijekcí b zobrazí, případně naopak.

Zavedeme-li na $F[x]_n$ navíc násobení $\cdot$ jako standardní násobení polynomů modulo $x^n - 1$, tj.

$$p \cdot q = \overbrace{(p \cdot q)}^{\text{násobení v } F[x]} \mod x^n - 1,$$

dostáváme $F[x]_n$ jako okruh.

Uvažujme okruhový homomorfismus $\Pi_n : F[x] \to F[x]_n$, $\Pi_n(p) = p \bmod x^n - 1$. Protože se zřejmě jedná o homomorfismus na, z 1. věty o izomorfismu plyne

$$F[x]_n \simeq F[x]/\mathrm{Ker}\ \Pi_n = F[x]/(x^n - 1).$$

Každý ideál I okruhu $F[x]_n$ obsahuje $[0]_{(x^n-1)}$, proto ideál $\Pi_n^{-1}(I)$ okruhu $F[x]$ obsahuje $x^n - 1$. Vzhledem k tomu, že $F[x]$ je obor hlavních ideálů, existuje právě

jeden monický polynom g, jenž dělí $x^n - 1$ a generuje $\Pi_n^{-1}(I)$. Lze tedy psát $I = (gF[x])$ a $F[x]_n$ takto představuje okruh hlavních ideálů.

Z uvedeného pozorování přímo plyne toto tvrzení:

Tvrzení 1.4. *Buď F konečné těleso a $n \in \mathbb{N}$. Pak každý ideál $F[x]_n$ lze psát ve tvaru*

$$C(g) = \{gh \mid h \in F[x], \deg h < n - \deg g\}$$

pro nějaký monický polynom $g \in F[x]$, jenž dělí $x^n - 1$, a naopak pro každý takový polynom g je $C(g)$ ideálem $F[x]_n$.

Důkaz. Zbývá dokázat, že pro každý monický polynom g dělící $x^n - 1$ je $C(g)$ ideálem $F[x]_n$. Uzavřenost $C(g)$ na sčítání je zřejmá, $0 \in C(g)$. Buď $r \in F[x]_n$ a $gh \in C(g)$. Pak $rgh \bmod (x^n - 1) = g(rh \bmod (\frac{x^n-1}{g})) \in C(g)$. □

Poslední tvrzení této kapitoly obsahuje charakterizaci lineárních cyklických kódů.

Tvrzení 1.5. *Buď $\mathcal{C} \subseteq F[x]_n$, pak $\mathcal{C}$ je lineární cyklický kód, právě když existuje monický polynom $g \in F[x]$, jenž dělí $x^n - 1$, a $C(g) = \mathcal{C}$.*

Důkaz. Buď $\mathcal{C} \subseteq F[x]_n$ lineární cyklický kód. Z linearity plyne uzavřenost na sčítání a násobení prvkem tělesa F. Cykličnost znamená, že pro každé kódové slovo u (myšleno jako polynom) je $xu, x^2u, \ldots, x^{n-1}u \in \mathcal{C}$. Pro libovolné $u \in \mathcal{C}$ a $f = \sum_{i=0}^{n-1} f_i x^i \in F[x]_n$ potom platí

$$f \cdot u = \sum_{i=0}^{n-1} \underbrace{f_i \overbrace{x^i u}^{\in \mathcal{C}}}_{\in \mathcal{C}} \in \mathcal{C},$$

a tedy $\mathcal{C}$ je ideál okruhu $F[x]_n$. Podle Tvrzení 1.4 je $\mathcal{C} = C(g)$ pro jednoznačně určený monický polynom g dělící $x^n - 1$.

Opačná implikace se dokáže snadno. Protože $C(g)$ je ideál, můžeme ho považovat za podprostor $F[x]_n \simeq F^n$ a pro libovolné $h \in C(g)$ je $xh \in C(g)$. □

Kapitola 2

Řetězové zlomky

Obsahem této kapitoly je popis struktury řetězových zlomků nad obecným tělesem, abychom těchto znalostí posléze využili při konstrukci dekódovacího algoritmu RS kódů. V této části bylo čerpáno z článku [7], ve kterém W. H. Mills poprvé publikoval algoritmus využívající řetězové zlomky, a [4], který nastiňuje podobnost zmíněného algoritmu s Berlekampovým-Masseyovým algoritmem.

Stále budeme pracovat s pojmy okruh a těleso, proto pro úplnost zopakujme základní definice těchto algebraických struktur:

Definice 2.1. Komutativní okruh s jednotkou $\mathcal{R}$ *je* $(R, +, \cdot, -, \mathbf{0}, \mathbf{1})$ *splňující následující podmínky:*

1. $(R, +, -, \mathbf{0})$ *je abelovská grupa, tj.*
   - $x + y = y + x$ *pro všechna* $x, y \in R$,
   - $(x + y) + z = x + (y + z)$ *pro všechna* $x, y, z \in R$,
   - $x + \mathbf{0} = \mathbf{0} + x = x$ *pro všechna* $x \in R$,
   - $x + (-x) = (-x) + x = \mathbf{0}$ *pro všechna* $x \in R$,
2. $x \cdot (y \cdot z) = (x \cdot y) \cdot z$ *pro všechna* $x, y, z \in R$,
3. $x \cdot (y + z) = x \cdot y + x \cdot z$, $(y + z) \cdot x = y \cdot x + z \cdot x$ *pro všechna* $x, y, z \in R$,
4. *pro všechna* $x, y \in R$ *platí* $x \cdot y = y \cdot x$,
5. *pro každé* $x \in R$ *platí* $x \cdot \mathbf{1} = \mathbf{1} \cdot x = x$.

Definice 2.2. Těleso *je komutativní okruh s jednotkou* $\mathcal{F} = (F, +, \cdot, -, \mathbf{0}, \mathbf{1})$, *v němž pro každé* $\mathbf{0} \neq x \in F$ *existuje* $x^{-1} \in F$ *takové, že* $x \cdot x^{-1} = \mathbf{1}$.

Těleso nazýváme konečné, *jestliže* $|F| < \infty$.

Víme, že počet prvků každého konečného tělesa je roven mocnině nějakého prvočísla a naopak pro každou mocninu prvočísla existuje právě jedno těleso obsahující právě tolik prvků.

Pro nějaké prvočíslo p a $m \in \mathbb{N}$ budeme těleso obsahující p^m prvků značit $\mathbb{F}_{p^m}$.

Teorii řetězových zlomků známe nejspíše pouze z aproximace reálných čísel. Avšak řetězové zlomky lze abstraktně uvažovat nad jakýmkoli tělesem F obsahujícím jako podmnožinu komutativní okruh s jednotkou R, přičemž je třeba zajistit

vhodné jednoznačné přiřazení každého prvku f tělesa F právě jednomu prvku $[f]$ okruhu R. $[f]$ budeme nazývat *celá část* f a *necelou částí* f budeme rozumět $f - [f]$.

Vrátíme-li se k případu reálných čísel, pohybovali jsme se v situaci, kdy okruh R tvořila množina celých čísel a těleso F byla množina reálných čísel. Jednoznačné přiřazení prvku f z F nějakému prvku $[f]$ z R pak znamenalo přiřazení reálnému číslu jeho dolní celou část.

V této souvislosti můžeme pro každý prvek f tělesa F definovat následující rozvoj

$$\begin{aligned} f &= a_0 + r_0 \\ &= a_0 + \frac{\mathbf{1}}{a_1 + r_1} \\ &= a_0 + \cfrac{\mathbf{1}}{a_1 + \cfrac{\mathbf{1}}{a_2 + r_2}} \\ &\vdots \\ &= a_0 + \cfrac{\mathbf{1}}{a_1 + \cfrac{\mathbf{1}}{\ddots \; a_{n-1} + \cfrac{\mathbf{1}}{a_n + r_n}}}, \end{aligned} \tag{2.1}$$

kde pro $n = 0, 1, \ldots$ byly určeny hodnoty a_n a r_n takto:

$$a_0 = [f], \tag{2.2}$$

$$a_n = \left[\frac{\mathbf{1}}{r_{n-1}}\right] \qquad \text{pro } n \geq 1, \text{ pokud } r_{n-1} \neq \mathbf{0} \text{ a je definované,} \tag{2.3}$$

$$r_0 = f - a_0, \tag{2.4}$$

$$r_n = \frac{\mathbf{1}}{r_{n-1}} - a_n \qquad \text{pro } n \geq 1, \text{ pokud } r_{n-1} \neq \mathbf{0} \text{ a je definované.} \tag{2.5}$$

Existuje-li $N \in \mathbb{N}_0$ takové, že $r_N = \mathbf{0}$, pak již a_n, ani r_n pro $n \geq N + 1$ nedefinujeme a řetězový zlomek, jemuž se f rovná, nazýváme *konečný*.

V opačném případě mluvíme o *nekonečném* řetězovém zlomku.

Nyní se dostáváme k aproximaci prvku tělesa F jediným zlomkem s koeficienty z okruhu R. Budeme-li mluvit o n-tém kroku (n-tém členu posloupnosti apod.), budeme tím myslet libovolné $0 \leq n$, případně $0 \leq n \leq N$, bude-li řetězový zlomek konečný.

Lemma 2.1. *Při vytváření řetězového zlomku, můžeme n-tou reprezentaci rovnice (2.1) psát ve tvaru*

$$f = \frac{A_n(a_n + r_n) + B_n}{C_n(a_n + r_n) + D_n}, \tag{2.6}$$

kde A_n, B_n, C_n a D_n jsou výrazy v proměnných $a_0, \ldots, a_{n-1}$, v nichž jsou navíc lineární.

Důkaz. Lemma dokážeme indukcí podle n.
Pro $n = 0$ je

$$f = a_0 + r_0 = \frac{\mathbf{1}(a_0 + r_0) + \mathbf{0}}{\mathbf{0}(a_0 + r_0) + \mathbf{1}}$$

a tvrzení platí triviálně.

Nechť $n \geq 1$ a tvrzení platí pro $n - 1$, tedy nechť existují vhodná A_{n-1}, B_{n-1}, C_{n-1} a D_{n-1} splňující

$$f = \frac{A_{n-1}(a_{n-1} + r_{n-1}) + B_{n-1}}{C_{n-1}(a_{n-1} + r_{n-1}) + D_{n-1}}.$$

Pak rozšířením předchozí rovnice zlomkem $\frac{a_n + r_n}{a_n + r_n}$ a dosazením $r_{n-1} = \frac{\mathbf{1}}{a_n + r_n}$ (což plyne z rovnice (2.5)) získáváme:

$$\begin{aligned} f &= \frac{A_{n-1}(a_{n-1} + \frac{\mathbf{1}}{a_n + r_n}) + B_{n-1}}{C_{n-1}(a_{n-1} + \frac{\mathbf{1}}{a_n + r_n}) + D_{n-1}} \cdot \frac{a_n + r_n}{a_n + r_n} \\ &= \frac{A_{n-1}a_{n-1}(a_n + r_n) + A_{n-1} + B_{n-1}(a_n + r_n)}{C_{n-1}a_{n-1}(a_n + r_n) + C_{n-1} + D_{n-1}(a_n + r_n)} \\ &= \frac{(\overbrace{A_{n-1}a_{n-1} + B_{n-1}}^{A_n})(a_n + r_n) + \overbrace{A_{n-1}}^{B_n}}{(\underbrace{C_{n-1}a_{n-1} + D_{n-1}}_{C_n})(a_n + r_n) + \underbrace{C_{n-1}}_{D_n}}. \end{aligned}$$

Definujme

$$\begin{aligned} A_n &= A_{n-1}a_{n-1} + B_{n-1}, \quad B_n = A_{n-1}, \\ C_n &= C_{n-1}a_{n-1} + D_{n-1}, \quad D_n = C_{n-1}. \end{aligned} \tag{2.7}$$

Takto zvolená A_n, B_n, C_n a D_n zřejmě odpovídají zadání a lemma je tedy dokázáno. □

Položením $r_n = \mathbf{0}$ v rovnici (2.6) dostaneme *n-tou aproximaci prvku f* pomocí řetězových zlomků, již označíme f_n:

$$f_n = \frac{P_n}{Q_n}, \tag{2.8}$$

kde

$$\begin{aligned} P_n &= A_n a_n + B_n, \\ Q_n &= C_n a_n + D_n. \end{aligned} \tag{2.9}$$

Porovnejme nyní rekurentní pravidla (2.7) pro výpočty A_n, B_n, C_n a D_n s předpisy (2.9) pro P_n a Q_n. Ihned vidíme, že můžeme psát

$$\begin{aligned} A_{n+1} &= P_n, \quad B_{n+1} = P_{n-1}, \\ C_{n+1} &= Q_n, \quad D_{n+1} = Q_{n-1}, \end{aligned} \tag{2.10}$$

z čehož dále plynou následující rekurence

$$\begin{aligned} P_{n+1} &= a_{n+1}P_n + P_{n-1}, \\ Q_{n+1} &= a_{n+1}Q_n + Q_{n-1} \end{aligned} \tag{2.11}$$

pro $0 \leq n \ (\leq N)$ s počátečními podmínkami

$$\begin{aligned} P_0 &= a_0, \quad P_{-1} = \mathbf{1}, \\ Q_0 &= \mathbf{1}, \quad Q_{-1} = \mathbf{0}. \end{aligned} \tag{2.12}$$

Ze vztahů (2.2) – (2.5) pro a_n a (2.11) – (2.12) pro výpočet P_n a Q_n získáváme jednoduchou metodu, jak generovat aproximace f_n prvku f. Poznamenejme, že operace v (2.11) jsou pouze okruhové sčítání a násobení prvků z R, a proto P_n i Q_n stále náleží R.

Tyto vztahy navíc implikují důležitou vlastnost uvedenou v Lemmatu 2.2, že P_n a Q_n jsou nesoudělné neboli že aproximační zlomky jsou v základním tvaru.

Lemma 2.2. *Pro všechna $0 \leq n \ (\leq N)$ platí*

$$P_nQ_{n-1} - Q_nP_{n-1} = (-\mathbf{1})^{n+1}. \tag{2.13}$$

Důkaz. Rekurzi (2.11) pro P_n a Q_n, $0 \leq n \ (\leq N)$, lze maticově zapsat jako

$$\begin{aligned} \begin{pmatrix} P_n & P_{n-1} \\ Q_n & Q_{n-1} \end{pmatrix} &= \begin{pmatrix} P_{n-1} & P_{n-2} \\ Q_{n-1} & Q_{n-2} \end{pmatrix} \cdot \begin{pmatrix} a_n & \mathbf{1} \\ \mathbf{1} & \mathbf{0} \end{pmatrix} \\ &= \begin{pmatrix} P_{n-2} & P_{n-3} \\ Q_{n-2} & Q_{n-3} \end{pmatrix} \cdot \begin{pmatrix} a_{n-1} & \mathbf{1} \\ \mathbf{1} & \mathbf{0} \end{pmatrix} \cdot \begin{pmatrix} a_n & \mathbf{1} \\ \mathbf{1} & \mathbf{0} \end{pmatrix} \\ &\vdots \\ &= \begin{pmatrix} a_0 & \mathbf{1} \\ \mathbf{1} & \mathbf{0} \end{pmatrix} \cdot \begin{pmatrix} a_1 & \mathbf{1} \\ \mathbf{1} & \mathbf{0} \end{pmatrix} \cdot \ \cdots \ \cdot \begin{pmatrix} a_n & \mathbf{1} \\ \mathbf{1} & \mathbf{0} \end{pmatrix}. \end{aligned} \tag{2.14}$$

Vypočtením determinantů na obou stranách rovnice dostáváme hledanou rovnost. □

Důsledek 2.3. *P_n a Q_n jsou nesoudělné v R.*

Počítáme-li nějakou aproximaci, obvykle nás zajímá, jak je tento odhad blízký skutečnosti. V případě aproximace řetězovými zlomky proto zavádíme pojem *chyby aproximace*.

Definice 2.3. *Pro $f \in F$ definujeme* n-tou chybu aproximace *jako zlomek*

$$\frac{\Delta_n}{Q_n} = f - f_n = f - \frac{P_n}{Q_n}. \tag{2.15}$$

Algoritmus 1. Aproximace řetězovými zlomky

Input: $f \in \mathcal{F}$

Output: $P, Q \in \mathcal{R}$ takové, že $f = P/Q$

0. $P_{-1} \leftarrow \mathbf{1}$, $Q_{-1} \leftarrow \mathbf{0}$, $\Delta_{-1} \leftarrow -\mathbf{1}$, $n \leftarrow -1$
   $P_0 \leftarrow [f]$, $Q_0 \leftarrow \mathbf{1}$, $\Delta_0 \leftarrow f - [f]$, $n \leftarrow 0$
1. **while** $\Delta_n \neq 0$ **do**
2. $\quad n \leftarrow n + 1$
3. $\quad a_n \leftarrow \left[-\frac{\Delta_{n-2}}{\Delta_{n-1}}\right]$
4. $\quad P_n \leftarrow a_n P_{n-2} + P_{n-1}$
   $\quad Q_n \leftarrow a_n Q_{n-2} + Q_{n-1}$
   $\quad \Delta_n \leftarrow a_n \Delta_{n-2} + \Delta_{n-1}$
5. **end while**
6. **return** $P = P_n$, $Q = Q_n$

Poznámka. Algoritmus terminuje, pokud taková P, Q existují.

Hodnoty Δ_n lze podobně jako P_n nebo Q_n počítat také rekurzivně. Položme

$$\Delta_n = fQ_n - P_n. \tag{2.16}$$

Vzhledem k tomu, že posloupnosti $\{Q_n\}$ i $\{P_n\}$ splňují stejné rekurence (2.11), platí také

$$\Delta_{n+1} = a_{n+1}\Delta_n + \Delta_{n-1}, \tag{2.17}$$

kde

$$\Delta_{-1} = -\mathbf{1}, \quad \Delta_0 = f - a_0 = r_0. \tag{2.18}$$

Vrátíme-li se nyní zpět k rovnici (2.6), z níž vyjádříme $(a_n + r_n)$, za A_n, B_n, C_n a D_n dosadíme podle (2.10), získáme rovnost

$$a_n + r_n = \frac{f \cdot D_n - B_n}{A_n - f \cdot C_n} = -\frac{f \cdot Q_{n-2} - P_{n-2}}{f \cdot Q_{n-1} - P_{n-1}} = -\frac{\Delta_{n-2}}{\Delta_{n-1}}, \tag{2.19}$$

a tedy dle (2.3) a (2.5)

$$a_n = \left[-\frac{\Delta_{n-2}}{\Delta_{n-1}}\right], \quad \text{pokud } \Delta_{n-1} \neq \mathbf{0}. \tag{2.20}$$

Všimněme si, že z předpisů (2.17), (2.18) a (2.20) dostáváme alternativní způsob výpočtu posloupnosti $\{a_n\}$ k původnímu postupu danému rovnicemi (2.2) – (2.5).[3]

Pro jisté odhady v sekci 2.3 se nám bude hodit ještě jedno možné vyjádření Δ_n, konkrétně

$$\Delta_n \overset{(2.19)}{=} -\frac{\Delta_{n-1}}{a_{n+1} + r_{n+1}} \overset{(2.5)}{=} -\frac{\Delta_{n-1}}{\frac{\mathbf{1}}{r_n}} = -\Delta_{n-1} r_n = r_0 \cdot \prod_{i=1}^{n} (-r_i). \tag{2.21}$$

Naposledy zmíněná rovnice (2.21) také přímo ukazuje, že řetězový zlomek bude konečný (tj. $r_N = 0$) právě tehdy, když $\Delta_N = 0$.

[3]Tento postup výpočtu a_n se využívá při praktické implementaci algoritmu s řetězovými zlomky (viz [7]). Při vhodné volbě normy (zachovávající znaménka) lze psát $a_n = -\left[\frac{\Delta_{n-2}}{\Delta_{n-1}}\right]$.

Příklad 2.1. *Algoritmus 1 použijme na vyjádření čísla* $f = \frac{5457}{1853} = 2{,}94495412\ldots$ *pomocí řetězového zlomku (nad tělesem reálných čísel a s celou částí definovanou standardně). Mezivýsledky algoritmu zapíšeme do následující tabulky.*

n	P_n	Q_n	Δ_n	a_n
-1	1	0	-1	$-$
0	2	1	$\frac{1751}{1853} = 0{,}94495412\ldots$	2
1	3	1	$-\frac{102}{1853} = -0{,}0550458\ldots$	$\left[-\frac{-1}{0{,}94495412\ldots}\right] = \left[\frac{1853}{1751}\right] = 1$
2	53	18	$\frac{17}{1853} = 0{,}00917431\ldots$	$\left[-\frac{0{,}94495412\ldots}{-0{,}0550458\ldots}\right] = \left[\frac{1751}{102}\right] = 17$
3	321	109	0	$\left[-\frac{-0{,}0550458\ldots}{0{,}00917431\ldots}\right] = \left[\frac{102}{17}\right] = 6$

Jako podíl celých čísel a pomocí řetězového zlomku lze tedy f *zapsat jako*

$$f = \frac{321}{109} = 2 + \cfrac{1}{1 + \cfrac{1}{17 + \cfrac{1}{6}}}.$$

2.1 Řetězové zlomky nad tělesem formálních mocninných řad

Naším cílem je vytvořit mechanismus, kterým dokážeme dekódovat RS kódy. Proto v tomto místě zadefinujme dvojici tělesa a okruhu, která nás v otázce řetězových zlomků při dekódování RS kódů jediná zajímá. Uvažujme množinu formálních mocninných řad nad tělesem F

$$F(\!(x)\!) = \left\{ \sum_{i=0}^{\infty} f_{d-i} x^{d-i} \mid f_d \neq 0 \right\},$$

kde $f_{d-i} \in F$ pro všechna $i \geq 0$ a $d \in \mathbb{Z}$ nazýváme *stupeň* řady.[4]

Přidejme k této množině nulový prvek (identitu vzhledem ke sčítání) $\mathbf{0} = 0x$, na nějž můžeme nahlížet jako na mocninnou řadu se všemi koeficienty nulovými (jeho stupeň dodefinováváme $-\infty$). A za jednotkový prvek (identitu vzhledem k násobení) zvolme $\mathbf{1} = 1x^0$.

Sčítání a násobení řad definujme obvyklým způsobem, tj.

$$\sum_{i=0}^{\infty} f_{d-i} x^{d-i} + \sum_{i=0}^{\infty} g_{e-i} x^{e-i} = \sum_{i=0}^{\infty} (f_{\max\{d,e\}-i} + g_{\max\{d,e\}-i}) x^{\max\{d,e\}-i}, \quad (2.22)$$

[4]Množina, se kterou pracujeme, je vlastně okruh Laurentových formálních mocninných řad s proměnnou x^{-1}. Korektní označení by tedy bylo $F(\!(x^{-1})\!)$, avšak z důvodu lepší přehlednosti volíme popsaný způsob.

$$\sum_{i=0}^{\infty} f_{d-i}x^{d-i} \cdot \sum_{i=0}^{\infty} g_{e-i}x^{e-i} = \sum_{i=0}^{\infty} h_{m-i}x^{m-i}, \tag{2.23}$$

kde

$$h_{m-i} = \sum_{j+k=m-i} f_j \cdot g_k, \quad m = d + e, \tag{2.24}$$

přičemž pro $i > d$ a $j > e$ položme $f_i = 0$ a $g_j = 0$.

Inverz existuje ke každé nenulové řadě, stupeň a koeficienty inverzní řady dostaneme z (2.23) a (2.24), kde položíme $m = 0$, $h_0 = 1$ a $h_i = 0$ pro $i = -1, -2, \ldots$.

Množina $F(\!(x)\!)$ s právě definovanými operacemi zřejmě tvoří těleso.

Dále uvažujme okruh polynomů

$$F[x] = \left\{ \sum_{i=0}^{d} f_{d-i}x^{d-i} \mid f_d \neq 0 \right\}$$

s obvyklými operacemi nad tělesem F. Pro potřeby řetězových zlomků budeme na $F[x] \subset F(\!(x)\!)$ nahlížet jako na okruh celých částí prvků z $F(\!(x)\!)$. Celou část mocninné řady pak definujeme jako

$$\left[\sum_{i=0}^{\infty} f_{d-i}x^{d-i} \right] = \begin{cases} \sum_{i=0}^{d} f_{d-i}x^{d-i}, & \text{pokud } d \geq 0, \\ 0, & \text{pokud } d < 0. \end{cases} \tag{2.25}$$

Příklad 2.2. *Nechť $\mathcal{F}$ je těleso Laurentových formálních mocninných řad nad tělesem $\mathbb{F}_{16} \cong \mathbb{Z}_2/(\alpha^4 + \alpha + 1)$ a $\mathcal{R}$ okruh polynomů nad tímtéž tělesem. Buď*

$$\begin{aligned} f(x) &= \frac{x^5 + \alpha^9x^4 + \alpha^6x^3 + \alpha^9x^2}{x^3 + \alpha^6x^2 + \alpha^3x + \alpha^{13}} \\ &= x^2 + \alpha^5x + \alpha^9 + \alpha^4x^{-1} + 0x^{-2} + 0x^{-3} + \alpha^2x^{-4} + \ldots. \end{aligned}$$

Zde vidíme dvě různé ekvivalentní reprezentace mocninné řady $f(x)$. První je ve tvaru podílu dvou polynomů z $\mathcal{R}$, druhá reprezentace znázorňuje $f(x)$ jako prvek $\mathcal{F}$.

Celá část $f(x)$ je podle (2.25) rovna $[f(x)] = x^2 + \alpha^5x + \alpha^9$.

2.2 Konvergence řetězových zlomků

Otázku konvergence je třeba řešit vzhledem k nějaké normě definované na tělese $\mathcal{F}$. Zde bude užitečné pracovat s *nearchimedovskou* normou, jejíž definice je:

Definice 2.4. *Buď $\mathcal{F}$ těleso.* Nearchimedovská norma *je zobrazení $\|\cdot\| : \mathcal{F} \to \mathbb{R}_+$ splňující*

N1) $\|A \cdot B\| = \|A\| \cdot \|B\|$,
N2) $\|A\| = 0$, *právě když* $A = \mathbf{0}$,
N3) $\|A + B\| \leq \max(\|A\|, \|B\|)$.[5]

[5]Nerovnost N3) se nazývá *ultrametrická*. Všimněme si, že se jedná o silnější verzi trojúhelníkové nerovnosti.

Příklad 2.3. *Mějme opět těleso $F(\!(x)\!)$. Každému nenulovému prvku $f(x) \in F(\!(x)\!)$ stupně d přiřaďme jeho normu jako $\|f(x)\| = 2^d$ a přirozeně $\|\mathbf{0}\| = 0$. Je lehké podle Definice 2.4 ověřit, že se jedná o nearchimedovskou normu.*

S touto představou normy budou následující lemmata jistě snadno pochopitelná.

Lemma 2.4. *Uvažujme normu $\|\cdot\|$ na tělese $\mathcal{F}$ danou podmínkami N1) – N3). Pak*

V1) $\|\mathbf{1}\| = 1$,
V2) $\|A^{-1}\| = \|A\|^{-1}$,
V3) $\|-A\| = \|A\|$,
V4) jestliže $\|B\| > \|A\|$, *pak* $\|A+B\| = \|B\|$.

Důkaz. V1) $0 \neq x = \|\mathbf{1}\| \overset{N1)}{=} \|\mathbf{1}\cdot\mathbf{1}\| = x \cdot x \Rightarrow x = 1$,
V2) $1 \overset{V1)}{=} \|\mathbf{1}\| = \|A \cdot A^{-1}\| \overset{N1)}{=} \|A\| \cdot \|A^{-1}\| \Rightarrow \|A^{-1}\| = \|A\|^{-1}$,
V3) $\|A\|^2 = \|A^2\| = \|(-A)^2\| = \|-A\|^2 \Rightarrow \|A\| = \|-A\|$,
V4) Nechť $\|B\| > \|A\|$. Pak $\|B\| = \|(A+B)-A\| \leq \max(\|A+B\|, \|A\|) = \|A+B\|$. Porovnáním s N3) získáváme požadovanou rovnost.

□

Lemma 2.5. *Nechť kromě N1) – N3) platí, že*

N4) norma každého prvku $\mathcal{R}$ je nezáporné celé číslo a navíc existuje alespoň jeden prvek z $\mathcal{R}$, jehož norma je 2, a
N5) necelé části prvků tělesa $\mathcal{F}$ mají normu menší než 1, tj.

$$\|f - [f]\| < 1 \quad \textit{pro všechna } f \in \mathcal{F}.$$

Potom pro každý prvek $f \in \mathcal{F}$ platí:

V5) jestliže $\|f\| \geq 1$ *pak* $\|f\| = \|[f]\|$,
V6) $\|f\| = 2^m$, *kde m je rovno celému číslu nebo* $-\infty$,
V7) $\|f - [f]\| \leq \frac{1}{2}$.

Důkaz. V5) Pokud $\|f\| \neq \|[f]\|$, pak je $\max(\|f\|, \|[f]\|) \overset{V4)}{=} \|f-[f]\| \overset{N5)}{<} 1 \overset{N3)}{\leq} \max(\|f\|, \|[f]\|)$, což je spor, a tedy $\|f\| = \|[f]\|$.
V6) Pro spor předpokládejme, že v tělese $\mathcal{F}$ existuje prvek f takový, že $2^m < \|f\| < 2^{m+1}$ pro nějaké $m \in \mathbb{Z}$. Nechť $d \in \mathcal{F}$ je prvek, jehož norma je 2. Pak podle V2) a N1) je $1 < \|fd^{-m}\| \overset{V5)}{=} \|[fd^{-m}]\| < 2$, což je ve sporu s N4).
V7) Důkaz plyne přímo z V6) a N5).

□

Normu chyby aproximace pomocí řetězového zlomku bude užitečné uvažovat ve tvaru

$$\left\| f - \frac{P_n}{Q_n} \right\| = \frac{\|\Delta_n\|}{\|Q_n\|}. \tag{2.26}$$

Za použití vlastnosti, že necelá část prvku tělesa $\mathcal{F}$ má normu nejvýše $\frac{1}{2}$, plyne ze vzorce (2.21)

$$\|\Delta_n\| = \prod_{i=0}^{n} \|r_i\| \leq 2^{-(n+1)}. \tag{2.27}$$

Zlepšit můžeme i dolní odhad $\|Q_n\|$. Z V5) a V7) dostáváme

$$\|a_n\| = \left\| \left[\frac{\mathbf{1}}{r_{n-1}} \right] \right\| = \frac{1}{\|r_{n-1}\|} \geq 2 \quad \text{pro } n \geq 1. \tag{2.28}$$

Předpokládejme, že $\|Q_{n-2}\| < \|Q_{n-1}\|$, pak také $\|Q_{n-2}\| < \|a_n\|\|Q_{n-1}\|$. Využitím V4) v rovnici (2.11) získáváme

$$\|Q_n\| = \|a_n\|\|Q_{n-1}\| \geq 2\|Q_{n-1}\|. \tag{2.29}$$

Jelikož $\|Q_{-1}\| = 0$ a $\|Q_0\| = 1$, vidíme, že $\{\|Q_n\|\}_{n\geq -1}$ je ostře rostoucí posloupnost a předpoklady pro (2.29) byly splněny pro všechna vhodná n. Tyto počáteční podmínky navíc implikují spodní odhad

$$\|Q_n\| \geq 2^n. \tag{2.30}$$

Výsledky (2.27) a (2.30) nám poskytují již velice dobrý odhad aproximační chyby

$$0 \leq \left\| f - \frac{P_n}{Q_n} \right\| = \frac{\|\Delta_n\|}{\|Q_n\|} \leq \frac{1}{2^{2n+1}}. \tag{2.31}$$

Tím je přímo dokázáno i následující tvrzení.

Tvrzení 2.6. *Jsou-li splněny předpoklady N1) – N5), posloupnost aproximací $\{P_n/Q_n\}$ konverguje k f v normě $\|\cdot\|$. Navíc tato posloupnost je konečná, tj. existuje $N \in \mathbb{N}$ takové, že*

$$f = \frac{P_N}{Q_N}$$

právě tehdy, když $\Delta_N = 0$.

Tvrzení 2.7. *Buď $\{P_n/Q_n\}$ posloupnost aproximací $f \in \mathcal{F}$ a nechť platí N1) – N5). Pro každou volbu dvojice prvků P a Q z okruhu $\mathcal{R}$ buď n největší celé číslo takové, že $\|Q_n\| \leq \|Q\|$. Potom*

$$\left\| f - \frac{P_n}{Q_n} \right\| \leq \left\| f - \frac{P}{Q} \right\|$$

a také

$$\left\| f - \frac{P_n}{Q_n} \right\| < \frac{1}{\|Q_n\|\|Q\|}.$$

V případě tělesa reálných čísel a na něm definované normy jako absolutní hodnoty toto tvrzení znamená, že všechny nejlepší racionální aproximace[6] jsou obsaženy v posloupnosti aproximací řetězovými zlomky. Zde je toto tvrzení formulováno pro obecné těleso a na něm definovanou nearchimedovskou normu.

[6]Nejlepší racionální aproximace reálného čísla r je racionální číslo p/q, $q > 0$, jež je číslu r bližší než jakákoli jiná aproximace s menším či rovným jmenovatelem.

Důkaz. Jelikož $\|Q_{-1}\| = 0$ a $\{\|Q_n\|\}$ je ostře rostoucí posloupnost, lze vždy nalézt n takové, že $\|Q_n\| \leq \|Q\| < \|Q_{n+1}\|$.

Pro spor předpokládejme, že $\|f - P/Q\| < \|f - P_n/Q_n\|$. Potom

$$\begin{aligned}\left\|\frac{P}{Q} - \frac{P_n}{Q_n}\right\| &= \left\|\left(f - \frac{P}{Q}\right) - \left(f - \frac{P_n}{Q_n}\right)\right\| \\ &\overset{(V4)}{=} \left\|f - \frac{P_n}{Q_n}\right\| \overset{(V4)}{=} \left\|\left(f - \frac{P_n}{Q_n}\right) - \left(f - \frac{P_{n+1}}{Q_{n+1}}\right)\right\| \\ &= \left\|\frac{P_nQ_{n+1} - P_{n+1}Q_n}{Q_nQ_{n+1}}\right\| \overset{(2.13)}{=} \frac{\mathbf{1}}{\|Q_n\|\|Q_{n+1}\|} < \frac{\mathbf{1}}{\|Q_n\|\|Q\|}.\end{aligned}$$

Převedením výrazu na levé straně rovnice na společného jmenovatele dostáváme po přenásobení celé rovnice $\|Q_n\|\|Q\|$ nerovnost

$$\|PQ_n - P_nQ\| < \mathbf{1}.$$

Protože P, Q, P_n i Q_n jsou prvky $\mathcal{R}$, leží v $\mathcal{R}$ i $PQ_n - P_nQ$. Podle N4) však musí být $\|PQ_n - P_nQ\| = 0$, a tudíž $PQ_n - P_nQ = \mathbf{0}$. Tím jsme dosáhli sporu. □

Vyslovme nyní pomocné lemma a tvrzení, díky kterému dokážeme určit, do jaké míry může být aproximace dvou prvků f a f^* tělesa $\mathcal{F}$ identická.

Lemma 2.8. *Nechť jsou splněny předpoklady N1) – N5). Nechť P_n/Q_n je n-tý člen posloupnosti aproximací řetězovými zlomky a nechť A a B jsou libovolné dva prvky okruhu $\mathcal{R}$ takové, že $A \neq \mathbf{0}$ a*

$$AP_n = BQ_n.$$

Pak

$$\|A\| \geq \|Q_n\|.$$

Důkaz. Rovnice (2.13) zjednodušeně říká, že

$$P_nQ_{n-1} - Q_nP_{n-1} = \pm\mathbf{1}.$$

Vynásobením A a nahrazením AP_n dostáváme

$$\|Q_n\|\|BQ_{n-1} - AP_{n-1}\| = \|A\|.$$

Protože $BQ_{n-1} - AP_{n-1}$ musí být nenulový prvek $\mathcal{R}$, je jeho norma alespoň 1, a tvrzení je tedy dokázáno. □

Terminologie. Uvažujme mocninnou řadu $f(x) \in F(\!(x)\!)$, u níž však známe pouze L jejích prvních koeficientů, tzn.

$$f(x) = \sum_{i=0}^{L-1} f_{d-i}x^{d-i} + X^{d-L},$$

kde d je stupeň mocninné řady $f(x)$, koeficienty $f_{d-i} \in F$ pro $i = 0, 1, \ldots, L-1$ jsou známé a $X^{d-L} \in F(\!(x)\!)$ je neznámá mocninná řada stupně nejvýše $d - L$. Budeme říkat, že o této řadě *nemáme úplnou informaci*.

Je-li řada $f(x) \in F(\!(x)\!)$ rovna podílu dvou polynomů z $F[x]$, nazveme ji *racionální*.

Příklad 2.4. *Pokračujme v Příkladu 2.2. Předpokládejme, že u mocninné řady f známe pouze prvních 9 koeficientů, což zapíšeme jako*

$$f = x^2 + \alpha^5 x + \alpha^9 + \alpha^4 x^{-1} + 0x^{-2} + 0x^{-3} + \alpha^2 x^{-4} + \alpha^8 x^{-5} + \alpha^{12} x^{-6} + X^{-7}.$$

Aplikujme na ni Algoritmus 1 a podívejme se, zda dokážeme nalézt polynomy, jejichž podílu se f rovná. Mezivýsledky opět zaznamenáme do tabulky. Pro zjednodušení budeme v tabulce každý polynom reprezentovat pouze seznamem jeho koeficientů, kde za koeficient absolutního členu vložíme tečku.

n	P_n	Q_n	Δ_n	a_n
-1	$1.$	0	$-1.$	$-$
0	$1\alpha^5\alpha^9.$	$1.$	$.\alpha^4 00\alpha^2\alpha^8\alpha^{12}X^{-7}$	$1\alpha^5\alpha^9.$
1	$\alpha^{11}\alpha\alpha^5 1.$	$\alpha^{11}0.$	$.00\alpha^{13}\alpha^4\alpha^8 X^{-6}$	$\alpha^{11}0.$
2	$\alpha^2\alpha^{11}\alpha^8\alpha^{11}00.$	$\alpha^2\alpha^8\alpha^5 1.$	$.000X^{-4}$	$\alpha^6\alpha^{12}\alpha^9.$

Druhá aproximace je tedy rovna

$$f_2 = \frac{P_2}{Q_2} = \frac{\alpha^2 x^5 + \alpha^{11} x^4 + \alpha^8 x^3 + \alpha^{11} x^2}{\alpha^2 x^3 + \alpha^8 x^2 + \alpha^5 x + 1} = \frac{x^5 + \alpha^9 x^4 + \alpha^6 x^3 + \alpha^9 x^2}{x^3 + \alpha^6 x^2 + \alpha^3 x + \alpha^{13}} = f.$$

Vidíme, že čitatel a jmenovatel druhé aproximace jsou již násobkem polynomů, jejichž podílu se f rovná.

Otázkou zůstává, kolik členů racionální mocninné řady obecně musíme znát, abychom dokázali jednoznačně určit polynomy, jejichž vydělením byla mocninná řada vytvořena. Za předpokladu, že známe stupeň „dělitele“, dostáváme odpověď z následujícího závěrečného tvrzení této kapitoly.

Tvrzení 2.9. *Nechť jsou splněny předpoklady N1) – N5). Buďte P_n/Q_n a P_m^*/Q_m^* se jmenovateli shora omezenými $K \geq 1$ nejlepší racionální aproximace prvků f a f^* tělesa $\mathcal{F}$, tj.*

$$\left\| f - \frac{P_n}{Q_n} \right\| = \min_{P,Q:\|Q\|\leq K} \left\| f - \frac{P}{Q} \right\|,$$

$$\left\| f^* - \frac{P_m^*}{Q_m^*} \right\| = \min_{P^*,Q^*:\|Q^*\|\leq K} \left\| f^* - \frac{P^*}{Q^*} \right\|.$$

Pak

$$\frac{P_n}{Q_n} = \frac{P_m^*}{Q_m^*}, \quad \textit{právě když } \|f - f^*\| < \frac{1}{\|Q_n\| K}.$$

Důkaz. Nechť $P_n/Q_n = P_m^*/Q_m^*$. Z ultrametrické nerovnosti a Tvrzení 2.7 plyne

$$\begin{aligned} \|f - f^*\| &= \left\| f - \frac{P_n}{Q_n} - f^* + \frac{P_m^*}{Q_m^*} \right\| \\ &\leq \max\left(\left\| f - \frac{P_n}{Q_n} \right\|, \left\| f^* - \frac{P_m^*}{Q_m^*} \right\| \right) \\ &< \max\left(\frac{1}{\|Q_n\| K}, \frac{1}{\|Q_m^*\| K} \right). \end{aligned}$$

Z předpokladu $Q_m^* P_n = P_m^* Q_n$ a Lemmatu 2.8 je $\|Q_m^*\| \geq \|Q_n\|$. Proto lze výše uvedenou rovnici zredukovat na

$$\|f - f^*\| < \frac{1}{\|Q_n\| K},$$

čímž je první implikace dokázána.

Nyní dokážeme opačnou implikaci. Nechť platí

$$\|f - f^*\| < \frac{1}{\|Q_n\| K}.$$

Z ultrametrické nerovnosti a Tvrzení 2.7 získáváme

$$\begin{aligned}
\frac{\|Q_m^* P_n - P_m^* Q_n\|}{\|Q_n\| \|Q_m^*\|} &= \left\| \frac{P_n}{Q_n} - \frac{P_m^*}{Q_m^*} \right\| \\
&= \left\| f - f^* - f + \frac{P_n}{Q_n} + f^* - \frac{P_m^*}{Q_m^*} \right\| \\
&\leq \max \left(\|f - f^*\|, \left\| f - \frac{P_n}{Q_n} \right\|, \left\| f^* - \frac{P_m^*}{Q_m^*} \right\| \right) \\
&< \max \left(\frac{1}{\|Q_n\| K}, \frac{1}{\|Q_m^*\| K} \right).
\end{aligned}$$

Přenásobením nerovnice $\|Q_n\| \|Q_m^*\|$ dostáváme

$$\|Q_m^* P_n - P_m^* Q_n\| < \max \left(\frac{\|Q_m^*\|}{K}, \frac{\|Q_n\|}{K} \right) \leq 1.$$

Protože $Q_m^* P_n - P_m^* Q_n \in \mathcal{R}$ a má normu menší než 1, musí být roven $\mathbf{0}$, a tedy

$$\frac{P_n}{Q_n} = \frac{P_m^*}{Q_m^*}.$$

□

Z předchozího víme, že je-li řada $A(x)$ racionální, pak posloupnost aproximací řetězovými zlomky terminuje, jakmile

$$A(x) = \frac{P_N(x)}{Q_N(x)} = A_N(x), \tag{2.32}$$

což nastane po nejvýše $\deg Q_N(x)$ krocích.

Tvrzení 2.9 po dosazení $K = \|Q_N(x)\|$ říká, že $A_N(x) = A_M^*(x)$, právě když $\|A(x) - A^*(x)\| < \|Q_N(x)\|^{-2}$. Hledíme-li na $A^*(x)$ jako na mocninnou řadu, která se shoduje s $A(x)$ v jejích „známých“ koeficientech, můžeme vyslovit důsledek k Tvrzení 2.9.

Důsledek 2.10. *Buď $A(x) = P(x)/Q(x)$ racionální mocninná řada, kde $P(x)$ a $Q(x)$ jsou nesoudělné prvky okruhu polynomů. Jestliže o $A(x)$ nemáme úplnou informaci a její nejvyšší exponent u x s „neznámým“ koeficientem je $-d$, pak $P(x)$ a $Q(x)$ jsou jednoznačně (až na asociovanost) určeny řadou $A(x)$, právě když* $\deg Q(x) < \frac{d}{2}$.

Tímto jsme dokázali správnost algoritmu, který v Kapitole 3 užijeme k řešení klíčové rovnice při dekódování Reed-Solomonových kódů.

Kapitola 3

Reed-Solomonovy kódy

V této kapitole se seznámíme s rodinou Reed-Solomonových samoopravných kódů. Popíšeme základní strukturu těchto kódů a načrtneme způsoby jejich konstrukce a kódování. Podrobněji popíšeme tři dekódovací algoritmy, jež jsou založeny na stejném principu (jak bude ukázáno dále).

Popis RS kódů je dostupný v mnoha publikacích, zde jsme čerpali především z původního Reedova a Solomonova článku [3] a novějšího Welchova [10].

3.1 Základní popis

RS kódy patří mezi lineární blokové kódy s parametry $[n, k, n-k+1]_q$. Abecedu tedy tvoří q-prvkové těleso $\mathbb{F}_q$. V této práci budeme délku bloku uvažovat rovnou $n = q-1$ (některé definice povolují libovolné $n < q$, v tom případě pak ale většinou mluvíme o *zobecněném RS kódu*). Dimenzi kódu $k \leq n$ lze volit libovolně. S takto definovanou délkou bloku se jedná o cyklický kód a jelikož minimální vzdálenost kódu je $n - k + 1$, jedná se navíc o MDS kód. Všechny tyto zmíněné vlastnosti budou dokázány v následujícím textu.

3.2 Konstrukce

Množinu všech kódových slov Reed-Solomonova kódu lze popsat různými způsoby, což odpovídá rozdílným metodám jeho konstrukce. Zde uvedeme dva takové postupy – původní, jenž byl navrhnut Reedem a Solomonem, a v současné době užívaný způsob konstrukce využívající vlastností cyklických kódů.

3.2.1 Původní konstrukce

Původní přístup uvažuje kódovanou zprávu ve tvaru $\mathbf{m} = (m_0, m_1, \ldots, m_{k-1})$, kde $m_i \in \mathbb{F}_q$ pro $i = 0, \ldots, k-1$. Tomuto vektoru přidružíme odpovídající polynom

$m(x) = m_0 + m_1 x + \cdots + m_{k-1}x^{k-1}$. Příslušné kódové slovo $\mathbf{c} \in \mathbb{F}_q^n$ pak vznikne vyhodnocením polynomu $m(x)$ v n různých prvcích $a_0, a_1, \ldots, a_{n-1}$ tělesa $\mathbb{F}_q$, tj.

$$\mathbf{c} = (c_0, c_1, \ldots, c_{n-1}) = (m(a_0), m(a_1), \ldots, m(a_{n-1})).$$

Obecně lze za $a_0, a_1, \ldots, a_{n-1}$ volit libovolné po dvou různé prvky daného tělesa, z praktických důvodů se však pro každé $i = 0, \ldots, n-1$ volí $a_i = \alpha^i$, kde α je primitivním prvkem $\mathbb{F}_q$. Vzhledem k tomu, že α je primitivní prvek, jsou $\alpha^0, \ldots, \alpha^{n-1}$ opravdu po dvou různé a jsou to všechny nenulové prvky tělesa $\mathbb{F}_q$.

Množina všech kódových slov je tedy

$$\begin{aligned}\mathcal{C} &= \left\{(m(\alpha^0), m(\alpha^1), \ldots, m(\alpha^{n-1})) \mid m(x) \in \mathbb{F}_q[x], \deg m < k\right\} \\ &= \left\{\mathbf{m} \cdot \mathbf{G} \mid \mathbf{m} \in \mathbb{F}_q^k\right\}, \text{ kde } \mathbf{G} = \begin{pmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & \alpha & \alpha^2 & \ldots & \alpha^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{k-1} & \alpha^{2(k-1)} & \ldots & \alpha^{(n-1)(k-1)} \end{pmatrix}.\end{aligned}$$

Tvrzení 3.1. *$\mathcal{C}$ je lineární $[n, k, n-k+1]_q$ kód s generující maticí $\mathbf{G}$ a kontrolní maticí*

$$\mathbf{H} = \begin{pmatrix} 1 & \alpha & \alpha^2 & \ldots & \alpha^{n-1} \\ 1 & \alpha^2 & \alpha^{2\cdot 2} & \ldots & \alpha^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{n-k} & \alpha^{2(n-k)} & \ldots & \alpha^{(n-k)(n-1)} \end{pmatrix}.$$

Důkaz. Snadno nahlédneme, že $\mathbf{G}$ je generující matice, neboť každé slovo $\mathbf{m} \cdot \mathbf{G}$ odpovídá lineární kombinaci řádků $\mathbf{G}$ s koeficienty $m_0, \ldots, m_{k-1}$ a naopak. $\mathcal{C}$ je tedy lineární $[n, k]$ kód.

Abychom dokázali, že $\mathbf{H}$ je kontrolní matice, stačí podle Pozorování 1.1 ukázat, že platí $\mathbf{G}\mathbf{H}^T = \mathbf{0}_{k\times(n-k)}$. Prvek matice $\mathbf{G}\mathbf{H}^T$ na pozici $(i, j) \in \{0, \ldots, k-1\} \times \{0, \ldots, n-k-1\}$ je roven

$$\begin{aligned}(\mathbf{G}\mathbf{H}^T)_{i,j} &= \sum_{l=0}^{n-1} \alpha^{il}\alpha^{l(j+1)} = \sum_{l=0}^{n-1} \alpha^{l(i+j+1)} \\ &= \sum_{l=0}^{n-1} (\alpha^{i+j+1})^l = \frac{(\alpha^{i+j+1})^n - 1}{\underbrace{\alpha^{i+j+1}}_{\neq 1} - 1} = \frac{(\overbrace{\alpha^n}^{=1})^{i+j+1} - 1}{\alpha^{i+j+1} - 1} = 0.\end{aligned}$$

Definujme Vandermondovu matici typu $n \times n$ předpisem $\mathbf{V} = (\alpha^{ij})_{i,j=0}^{n-1}$, o níž víme, že je regulární.[7] Protože $\mathbf{H}$ je podmaticí $\mathbf{V}$, je hodnost $\mathbf{H}$ právě $n-k$. Tedy i každých $n-k$ sloupců je lineárně nezávislých, a proto podle Tvrzení 1.3 je minimální vzdálenost kódu $n-k+1$ (tj. jedná se o MDS kód). □

[7] Vandermondova matice řádu n je definována jako $\mathbf{V} = (v_{ij})_{i,j=0}^{n-1} = (a_i^j)_{i,j=0}^{n-1}$ pro nějaká $a_0, a_1, \ldots, a_{n-1}$. Její determinant je roven $\prod_{0\le i\le j\le n-1}(a_j - a_i)$. Proto matice $\mathbf{V}$ je regulární, právě když $a_0, a_1 \ldots, a_{n-1}$ jsou po dvou různá.

Tvrzení 3.2. *$\mathcal{C}$ je cyklický kód.*

Důkaz. $\mathbf{c} = (c_0, c_1, \ldots, c_{n-1}) \in \mathbb{F}_q^n$ je kódovým slovem, právě když $\mathbf{H}\mathbf{c}^T = \mathbf{0}^T$. Označíme-li $c(x) = \sum_{i=0}^{n-1} c_i x^i \in \mathbb{F}_q[x]_n$ polynom příslušný (ve smyslu bijekce definované v části 1.1) kódovému slovu $\mathbf{c}$, pak $c(\alpha^i) = 0$ pro všechna $i = 1, \ldots, n-k$. $\mathcal{C}$ tedy odpovídá množině $\{c(x) \in \mathbb{F}_q[x]_n \mid c(\alpha^i) = 0, i = 1, \ldots, n-k\}$, tj. (dle Tvrzení 1.5) cyklickému kódu $C\left(\prod_{i=1}^{n-k}(x - \alpha^i)\right)$. □

Posledním důkazem jsme se přímo dostali k další možné konstrukci RS kódů, k tzv. konstrukci s generujícím polynomem.

3.2.2 Konstrukce s generujícím polynomem

Při této konstrukci rovnou užíváme vlastnosti, že RS kódy jsou cyklické. Kódovanou zprávu tvoří opět vektor $\mathbf{m} = (m_0, \ldots, m_{k-1})$, resp. jeho příslušný polynom $m(x) = m_0 + m_1 x + \cdots + m_{k-1}x^{k-1}$. Pro dané parametry $[n, k]_q$ dále definujeme *generující polynom*[8]

$$g(x) = \prod_{i=1}^{n-k}(x - \alpha^i) = g_0 + g_1 x + \cdots + g_{n-k-1}x^{n-k-1} + x^{n-k},$$

kde α je primitivním prvkem tělesa $\mathbb{F}_q$. Kódováním zprávy $\mathbf{m}$ pak rozumíme vynásobení polynomu $m(x)$ polynomem $g(x)$. Výsledné kódové slovo je zřejmě stupně menšího než n, a tedy $m(x)g(x) = c(x) \in \mathbb{F}_q[x]_n$.

Množinu všech kódových slov v tomto případě můžeme popsat jako

$$\mathcal{C}' = \left\{ (c_0, c_1, \ldots, c_{n-1}) \mid \sum_{i=0}^{n-1} c_i x^i = c(x) = m(x)g(x), m(x) \in \mathbb{F}_q[x], \deg m < k \right\}.$$

Z pozorování provedeného v předchozí části plyne rovnost $\mathcal{C} = \mathcal{C}'$.

3.3 Kódování

Uvedené způsoby konstrukce naznačují, jak je možno odesílané zprávy kódovat. Přitom rozlišujeme tzv. *systematické* a *nesystematické* kódování. Při systematickém kódování je vstupní slovo obsaženo ve výstupním (kódovém) slově jako podslovo. V případě nesystematického kódování tomu tak není.

[8] Při vytváření generujícího polynomu lze použít libovolných $n-k$ po sobě jdoucích mocnin α, tj. $g(x) = \prod_{i=1}^{n-k}(x - \alpha^{j+i})$ pro nějaké $j \in \mathbb{N}_0$. Zde pro větší přehlednost a korespondenci s předchozím odstavcem volíme $j = 0$.

3.3.1 Kódování s využitím generující matice

Je-li $\mathbf{m} \in \mathbb{F}_q^k$ vstupní slovo, pak přenásobení tohoto slova generující maticí $\mathbf{G}$ (ve smyslu sekce 3.2.1) znamená nesystematické kódování, neboť výsledné kódové slovo zjevně neobsahuje $\mathbf{m}$ jako podslovo.

Matici $\mathbf{G}$ však lze Gaussovou-Jordanovou eliminační metodou převést na matici $\mathbf{G}'$, jež bude ve tvaru $\mathbf{G}' = (\mathbf{I}\ \mathbf{A})$, kde $\mathbf{I}$ značí jednotkovou matici typu $k \times k$ a $\mathbf{A}$ matici typu $k \times (n-k)$.

Řádky matice $\mathbf{G}'$ generují stejný podprostor jako řádky matice $\mathbf{G}$, a tudíž se jedná o generující matice stejného kódu. Užijeme-li nyní ke kódování slova $\mathbf{m}$ matici $\mathbf{G}'$, bude výsledné slovo obsahovat na prvních k pozicích slovo $\mathbf{m}$, a jde tedy o systematické kódování.

3.3.2 Kódování pomocí generujícího polynomu

Uvažujme vstupní slovo jako polynom $m(x) = m_0 + m_1 x + \cdots + m_{k-1}x^{k-1}$. Přenásobení $m(x)$ generujícím polynomem $g(x)$ opět představuje nesystematické kódování.

Naopak vytváříme-li kódové slovo jako

$$c(x) = x^{n-k}m(x) - p(x), \quad \text{kde } p(x) = x^{n-k}m(x) \bmod g(x),$$

popsali jsme způsob systematického kódování. Polynom $p(x)$ je stupně nejvýše $n-k-1$, neboť je zbytkem po dělení polynomem stupně $n-k$. Označíme-li $p(x) = p_0 + p_1 x + \cdots + p_{n-k-1}x^{n-k-1}$ a $c(x) = c_0 + c_1 x + \cdots + c_{n-1}x^{n-1}$, pak kódové slovo $\mathbf{c} = (c_0, c_1, \ldots, c_{n-1}) = (p_0, p_1, \ldots, p_{n-k-1}, m_0, m_1, \ldots, m_{k-1})$.

K důkazu, že $\mathbf{c}$ je opravdu kódové slovo, stačí ukázat, že $g(x)$ dělí $c(x)$, což evidentně platí, neboť

$$c(x) = x^{n-k}m(x) - p(x) \equiv p(x) - p(x) = 0 \mod g(x).$$

Příklad 3.1. *Uvažujme Reed-Solomonův kód s parametry $[6, 2, 5]_7$ nad tělesem $\mathbb{Z}_7$. Jako primitivní prvek tělesa vezměme 3. Pak generujícím polynomem je*

$$g(x) = (x-3)(x-3^2)(x-3^3)(x-3^4) = x^4 + 6x^3 + 3x^2 + 2x + 4.$$

Vstupní slovo nechť je $\mathbf{m} = (1, 3)$, resp. $m(x) = 3x + 1$. Při systematickém kódování pomocí generujícího polynomu je pak příslušné kódové slovo

$$c(x) = x^4(3x+1) - (2x^3 + 3x^2 + x + 5) = 3x^5 + x^4 + 5x^3 + 4x^2 + 6x + 2,$$

což lze jako vektor zapsat $\mathbf{c} = (2, 6, 4, 5, 1, 3)$.

Vidíme, že kódové slovo $\mathbf{c}$ obsahuje na svých posledních dvou pozicích přímo vstupní zprávu $\mathbf{m}$.

3.4 Dekódování

Otázka dekódování RS kódů je v jistém smyslu mnohem složitější než kódování. Existuje mnoho dekódovacích algoritmů, zde popíšeme tyto tři – Berlekampův-Masseyův, Eukleidův a algoritmus využívající řetězové zlomky. Popisy uvádíme v souladu s publikacemi, ve kterých byly poprvé zveřejněny. Konkrétně Eukleidův v [11], Berlekampův-Masseyův v článcích [1] a [5] a Millsův algoritmus s řetězovými zlomky v [7].

Stejně jako v předchozích sekcích uvažujme RS kód s parametry $[n, k, d]_q$ s generujícím polynomem $g(x) = \prod_{i=1}^{n-k}(x - \alpha^i)$, kde α je primitivní prvek $\mathbb{F}_q$. Označme t počet chyb, který je kód nejvýše schopen opravit. Podle Tvrzení 1.1 je $t = \lfloor \frac{n-k}{2} \rfloor$, kde $\lfloor \cdot \rfloor$ značí dolní celou část čísla.

Odesílané kódové slovo nechť je $c(x) = m(x)g(x) = c_0 + c_1x + \cdots + c_{n-1}x^{n-1}$. Chyby při přenosu kanálem bude představovat *chybový polynom* $e(x) = e_0 + e_1x + \cdots + e_{n-1}x^{n-1} \in \mathbb{F}_q[x]$ a přijaté slovo bude ve tvaru $r(x) = c(x) + e(x) = r_0 + r_1x + \cdots + r_{n-1}x^{n-1}$.

Je-li $e_i \neq 0$, $0 \leq i \leq n-1$, řekneme, že došlo k chybě na i-té pozici. Nadále předpokládejme, že došlo k ν chybám, kde $0 \leq \nu \leq t$, na pozicích $j_1 < j_2 < \cdots < j_\nu$.

3.4.1 Výpočet syndromů

Dekódovací proces začneme vyhodnocením přijatého polynomu $r(x)$ v kořenech generujícího polynomu, tj. $\alpha, \alpha^2, \ldots, \alpha^{n-k}$. Pro $i = 1, 2, \ldots, n-k$ nazýváme

$$S_i = r(\alpha^i) = \sum_{j=0}^{n-1} r_j \alpha^{ij} \tag{3.1}$$

i-tý syndrom.

Jelikož $c(x) = m(x)g(x)$ a $g(\alpha^i) = 0$ pro $i = 1, 2, \ldots, n-k$, platí také

$$\begin{aligned} S_i &= r(\alpha^i) \\ &= c(\alpha^i) + e(\alpha^i) \\ &= m(\alpha^i)g(\alpha^i) + e(\alpha^i) \\ &= e(\alpha^i) = \sum_{j=0}^{n-1} e_j \alpha^{ij} \quad \text{pro } i = 1, 2, \ldots, n-k. \end{aligned} \tag{3.2}$$

Poznamenejme, že $r(x)$ je zjevně kódové slovo (nenastala žádná chyba) právě tehdy, když jsou všechny syndromy rovny nule.

Pro zjednodušení situace zavádíme pro $k = 1, 2, \ldots, \nu$ nové označení

$$X_k = \alpha^{j_k},$$

$$Y_k = e_{j_k},$$

kde X_k nazýváme *pozice chyby* a Y_k *velikost chyby.*

Poslední sumu ve (3.2) lze ještě zkrátit vynecháním nulových sčítanců:

$$S_i = \sum_{k=1}^{\nu} e_{j_k} \alpha^{ij_k} = \sum_{k=1}^{\nu} Y_k X_k^i \quad \text{pro } i = 1, 2, \ldots, n-k. \tag{3.3}$$

Vyřešíme-li (3.3), což je soustava $n-k$ nelineárních rovnic o 2ν neznámých, určíme tím chybový polynom, jehož odečtením od přijaté zprávy získáme původní odesílanou zprávu.

Nyní zavedeme dva polynomy – lokační a evaluační. Tyto polynomy budou velice užitečné při hledání pozic a velikostí chyb.

3.4.2 Lokační polynom

Jako první definujme *lokační polynom (error-locator polynomial)*

$$\Lambda(x) = \prod_{k=1}^{\nu}(1 - xX_k) = 1 + \Lambda_1 x + \cdots + \Lambda_\nu x^\nu,$$

jehož kořeny jsou právě všechny inverzy pozic chyb, tj. X_k^{-1}, $1 \le k \le \nu$.

Podívejme se na tento polynom ještě podrobněji. Dosaďme do něj jeden z jeho kořenů a přenásobme vzniklou rovnici $Y_k X_k^{j+\nu}$, kde j je prozatím libovolné.

$$\begin{aligned} 0 &= \Lambda(X_k^{-1}) \\ 0 &= 1 + \Lambda_1 X_k^{-1} + \cdots + \Lambda_\nu X_k^{-\nu} \\ 0 &= Y_k X_k^{j+\nu} + \Lambda_1 Y_k X_k^{j+\nu-1} + \cdots + \Lambda_\nu Y_k X_k^j. \end{aligned} \tag{3.4}$$

Sečtením přes $k = 1$ do ν dostáváme

$$\begin{aligned} 0 &= \sum_{k=1}^{\nu}(Y_k X_k^{j+\nu} + \Lambda_1 Y_k X_k^{j+\nu-1} + \cdots + \Lambda_\nu Y_k X_k^j) \\ 0 &= \sum_{k=1}^{\nu} Y_k X_k^{j+\nu} + \Lambda_1 \sum_{k=1}^{\nu} Y_k X_k^{j+\nu-1} + \cdots + \Lambda_\nu \sum_{k=1}^{\nu} Y_k X_k^j, \end{aligned} \tag{3.5}$$

z čehož nahrazením $S_i = \sum_{k=1}^{\nu} Y_k X_k^i$ plyne soustava rovnic pro $1 \le j \le 2t - \nu$

$$\begin{aligned} S_{j+\nu} + \Lambda_1 S_{j+\nu-1} + \cdots + \Lambda_\nu S_j &= 0 \\ \Lambda_1 S_{j+\nu-1} + \cdots + \Lambda_\nu S_j &= -S_{j+\nu}. \end{aligned} \tag{3.6}$$

Dekódováním přijaté zprávy na základě řešení této soustavy ν lineárních rovnic (pro $1 \le j \le \nu$) o ν neznámých se zabývá Petersonův algoritmus [8].[9]

[9] K tomu poznamenejme, že hodnotu ν také neznáme a je rovna nejmenšímu takovému číslu, pro nějž řešení (3.6) existuje.

3.4.3 Evaluační polynom a klíčová rovnice

Zde se však budeme věnovat algoritmům, které navíc využívají výše zmíněný *evaluační polynom (error-evaluator polynomial)*, jenž definujeme předpisem

$$\Omega(x) = S(x)\Lambda(x) \mod x^{2t}, \tag{3.7}$$

kde $S(x) = S_1 + S_2 x + \cdots + S_{2t} x^{2t-1}$ nazýváme *syndromový polynom.*[10]

Rovnice (3.7) se nazývá *klíčová rovnice (key equation)*. Dekódovací algoritmy, které v této práci popisujeme, řeší právě otázku klíčové rovnice, tj. jak nalézt polynomy $\Lambda(x)$ a $\Omega(x)$ pouze při znalosti $S(x)$.

Evaluační polynom můžeme rozepsat jako

$$\begin{aligned}
\Omega(x) &= S(x)\Lambda(x) \mod x^{2t} \\
&= \left(\sum_{i=1}^{2t}\sum_{k=1}^{\nu} Y_k X_k^i x^{i-1}\right)\left(\prod_{j=1}^{\nu}(1 - X_j x)\right) \mod x^{2t} \\
&= \left(\sum_{i=1}^{2t}\sum_{k=1}^{\nu} Y_k X_k X_k^{i-1} x^{i-1}\right)\left((1 - X_k x)\prod_{j\neq k}(1 - X_j x)\right) \mod x^{2t} \\
&= \sum_{k=1}^{\nu} Y_k X_k (1 - X_k x)\sum_{i=1}^{2t}(X_k x)^{i-1}\prod_{j\neq k}(1 - X_j x) \mod x^{2t} \\
&= \sum_{k=1}^{\nu} Y_k X_k (1 - X_k^{2t} x^{2t})\prod_{j\neq k}(1 - X_j x) \mod x^{2t} \\
&= \sum_{k=1}^{\nu} Y_k X_k \prod_{j\neq k}(1 - X_j x).
\end{aligned} \tag{3.8}$$

V tomto místě si můžeme všimnout, že $\deg\Lambda(x) = \nu$, $\deg\Omega(x) \leq \nu - 1$ a že tyto dva polynomy jsou nesoudělné, neboť nemají žádný společný kořen.

Jak již název napovídá, evaluační polynom používáme k určení velikostí chyb. Známe-li lokační polynom, činíme tak podle vzorce

$$Y_k = -\frac{\Omega(X_k^{-1})}{\Lambda'(X_k^{-1})} \quad \text{pro } k = 1, 2, \ldots, \nu, \tag{3.9}$$

kde $\Lambda'(x)$ značí formální první derivaci $\Lambda(x)$ vzhledem k x. Tuto derivaci můžeme podle základního vzorce pro derivaci součinu psát ve tvaru

$$\Lambda'(x) = -X_k\prod_{j\neq k}(1 - X_j x) + (1 - X_k x)(\prod_{j\neq k}(1 - X_j x))'$$

a z toho ihned plyne platnost (3.9)

$$-\frac{\Omega(X_k^{-1})}{\Lambda'(X_k^{-1})} = -\frac{Y_k X_k \prod_{j\neq k}(1 - X_j X_k^{-1})}{-X_k \prod_{j\neq k}(1 - X_j X_k^{-1})} = Y_k. \tag{3.10}$$

[10] V některé literatuře se můžeme setkat s definicí $S(x) = S_1 x + \ldots S_{2t} x^{2t}$, pak ale měníme i definici $\Omega(x) = (1 + S(x))\Lambda(x) \mod x^{2t+1}$ a u uvedených pozorování také dojde k odpovídajícím změnám.

Tento postup výpočtu velikostí chyb pomocí syndromového, lokačního a evaluačního polynomu nazýváme *Forneyova formule.*

Příklad 3.2. *Pokračujme v Příkladu 3.1. Kódové slovo* $c(x) = 3x^5 + x^4 + 5x^3 + 4x^2 + 6x + 2$ *nyní posíláme komunikačním kanálem. Na dvou pozicích slova dojde k chybám, které popisuje chybový polynom* $e(x) = 4x + 2x^2$*. Přijaté slovo tedy je*

$$r(x) = c(x) + e(x) = 3x^5 + x^4 + 5x^3 + 6x^2 + 3x + 2.$$

Syndromy, které později využijeme při procesu dekódování, jsou $S_1 = r(3) = 2$*,* $S_2 = r(3^2) = 2$*,* $S_3 = 5$*,* $S_4 = 6$*,* $S_5 = 0$ *a* $S_6 = 6$*. Evidentně* $r(x)$ *není kódové slovo, protože všechny syndromy nejsou nulové.*

Pozice chyb jsou $X_1 = 3$ *a* $X_2 = 3^2 = 2$*, velikosti chyb* $Y_1 = 4$ *a* $Y_2 = 2$*. Z toho lze vypočítat lokační polynom*

$$\Lambda(x) = (1 - 3x)(1 - 2x) = 6x^2 + 2x + 1$$

a evaluační polynom

$$\Omega(x) = 4 \cdot 3 \cdot (1 - 2x) + 2 \cdot 2 \cdot (1 - 3x) = 6x + 2.$$

Ve chvíli, kdy tyto polynomy určíme, lze snadno[11] *nalézt pozice chyb, což jsou inverzy kořenů* $\Lambda(x)$*. V našem případě jsou kořeny lokačního polynomu 5 a 4, pozicemi chyb tedy jsou* $5^{-1} = 3 = X_1$ *a* $4^{-1} = 2 = 3^2 = X_2$*.*

Velikosti chyb určíme podle vzorce (3.9):

$$Y_1 = -\frac{\Omega(3^{-1})}{\Lambda'(3^{-1})} = -\frac{6 \cdot 3^{-1} + 2}{-3 \cdot (1 - 2 \cdot 3^{-1})} = -\frac{4}{6} = 4,$$

$$Y_2 = -\frac{\Omega(2^{-1})}{\Lambda'(2^{-1})} = -\frac{6 \cdot 2^{-1} + 2}{-2 \cdot (1 - 3 \cdot 2^{-1})} = -\frac{5}{1} = 2.$$

Chybový polynom je tedy $e(x) = 4x + 2x^2$*, jehož odečtením od přijatého slova* $r(x)$ *dostaneme odesílané kódové slovo* $c(x) = r(x) - e(x) = 3x^5 + x^4 + 5x^3 + 4x^2 + 6x + 2$*. Vzhledem k tomu, že se jedná o systematické kódování, ihned vidíme, že vstupní zpráva byla* $m(x) = 3x + 4$*.*

Nyní popíšeme tři dekódovací algoritmy – Berlekampův-Masseyův, Eukleidův a algoritmus s řetězovými zlomky. V Kapitole 4 ukážeme, že tyto tři algoritmy jsou ekvivalentní, a tudíž se omezíme na kompletní důkaz správnosti pouze u jednoho, konkrétně u algoritmu s řetězovými zlomky.

[11] Hledání všech kořenů lokačního polynomu provádíme evaluací $\Lambda(x)$ v každém prvku tělesa (abecedy). Tento postup nazýváme *Chienovo vyhledávání (Chien search).*

3.4.4 Berlekampův-Masseyův algoritmus

V roce 1966 E. Berlekamp zveřejnil ve svém článku [1] první efektivní algoritmus pro řešení klíčové rovnice. Přibližně o dva roky později J. Massey v [5] ukázal, že tento algoritmus je ekvivalentní nalezení nejkratšího LFSR[12] generujícího posloupnost syndromů $S_1, S_2, \ldots, S_{2t}$. Tento přístup založený na ideji posuvného registru se nyní nazývá Berlekampův-Masseyův (zkráceně BM) algoritmus.

Každý LFSR je dán svojí *délkou* $L \in \mathbb{N}_0$ a *charakteristickým polynomem*[13] $C(x) = 1 + C_1 x + \cdots + C_L x^L$. Řekneme, že LFSR *generuje* posloupnost $s_1, s_2, \ldots, s_N$, kde $N \geq L$, jestliže

$$s_j = -\sum_{i=1}^{L} C_i s_{j-i} \quad \text{pro } j = L+1, L+2, \ldots, N. \tag{3.11}$$

Porovnáme-li poslední soustavu rovnic s (3.6), ihned vidíme, že charakteristický polynom LFSR hledaného BM algoritmem je roven lokačnímu polynomu $\Lambda(x)$.

Struktura algoritmu je iterativní. Pro každé n počínaje $n = 1$ hledáme LFSR minimální délky (takových registrů může existovat více), jenž generuje posloupnost $S_1, S_2, \ldots, S_n$. Charakteristický polynom tohoto LFSR označujeme v pseudokódu $\Lambda^{(n)}(x) = 1 + \Lambda_1^{(n)} x + \cdots + \Lambda_{L^{(n)}}^{(n)} x^{L^{(n)}}$ a v proměnné $L^{(n)}$ uchováváme délku registru.

Je-li $\Lambda^{(n-1)}(x)$ charakteristický polynom registru generujícího $S_1, S_2, \ldots, S_{n-1}$, ověříme, zda negeneruje i posloupnost $S_1, S_2, \ldots, S_n$. Toto ověření v algoritmu reprezentuje tzv. *diskrepance*

$$D^{(n)} = S_n + \sum_{j=1}^{L^{(n-1)}} \Lambda_j^{(n-1)} S_{n-j}. \tag{3.12}$$

Pokud je diskrepance rovna nule, pak registr delší posloupnost evidentně generuje a můžeme položit $\Lambda^{(n)}(x) = \Lambda^{(n-1)}(x)$ a $L^{(n)} = L^{(n-1)}$. V opačném případě užijeme stěžejní trik BM algoritmu – výpočet $\Lambda^{(n)}(x)$ provedeme z charakteristických polynomů LFSR v iteraci $n-1$ a iteraci předcházející poslední změně délky registru podle vzorce

$$\Lambda^{(n)}(x) = \Lambda^{(n-1)}(x) - D^{(n)} T(x), \tag{3.13}$$

kde

$$T(x) = x^{n-u} \Lambda^{(u-1)}(x) / D^{(u)}, \tag{3.14}$$

jestliže poslední změna délky registru proběhla v u-té iteraci.

Délka registru je pak rovna (důkaz viz [5, Theorem 2])

$$L^{(n)} = \max\{L^{(n-1)}, n - L^{(n-1)}\}. \tag{3.15}$$

[12] LFSR (Linear Feedback Shift Register) – česky lineární posuvný registr se zpětnou vazbou.
[13] Koeficient C_L (stejně jako ostatní koeficienty) může být roven nule, a proto $\deg C(x) \leq L$.

Algoritmus 2. Berlekampův-Masseyův algoritmus

Input: posloupnost syndromů $S_1, S_2, \ldots, S_{2t}$

Output: lokační polynom $\Lambda(x)$

7. $n \leftarrow 0$, $\Lambda^{(0)}(x) \leftarrow 1$, $L^{(0)} \leftarrow 0$, $T(x) \leftarrow x$
8. **while** $n < 2t$ **do**
9. $\quad n \leftarrow n + 1$
10. $\quad D^{(n)} \leftarrow S_n + \sum_{j=1}^{L^{(n-1)}} \Lambda_j^{(n-1)} S_{n-j}$
11. $\quad$ **if** $D^{(n)} = 0$ **then**
12. $\quad\quad \Lambda^{(n)}(x) \leftarrow \Lambda^{(n-1)}(x)$
13. $\quad\quad L^{(n)} \leftarrow L^{(n-1)}$
14. $\quad$ **else**
15. $\quad\quad \Lambda^{(n)}(x) \leftarrow \Lambda^{(n-1)}(x) - D^{(n)} T(x)$
16. $\quad\quad$ **if** $2L^{(n-1)} < n$ **then**
17. $\quad\quad\quad L^{(n)} \leftarrow n - L^{(n-1)}$
18. $\quad\quad\quad T(x) \leftarrow \Lambda^{(n-1)}(x)/D^{(n)}$
19. $\quad\quad$ **else**
20. $\quad\quad\quad L^{(n)} \leftarrow L^{(n-1)}$
21. $\quad\quad$ **end if**
22. $\quad$ **end if**
23. $\quad T(x) \leftarrow x \cdot T(x)$
24. **end while**
25. **return** $\Lambda(x) = \Lambda^{(2t)}(x)$

LFSR s takto definovaným charakteristickým polynomem $\Lambda^{(n)}(x)$ a délkou $L^{(n)}$ by měl generovat posloupnost $S_1, S_2, \ldots, S_n$. Podle definice (3.11) musí platit

$$S_j + \sum_{i=1}^{L^{(n)}} \Lambda_i^{(n)} S_{j-i} = 0 \quad \text{pro } j = L^{(n)} + 1, L^{(n)} + 2, \ldots, n. \tag{3.16}$$

Uvědomme si, že (3.11) splňuje jak $\Lambda^{(n-1)}(x)$, tak i $\Lambda^{(u-1)}(x)$. Rozepíšeme-li pak $\Lambda^{(u)}(x)$ podle (3.13), platí

$$\begin{aligned} & S_j + \sum_{i=1}^{L^{(n)}} \Lambda_i^{(n)} S_{j-i} \\ & \quad = S_j + \sum_{i=1}^{L^{(n-1)}} \Lambda_i^{(n-1)} S_{j-i} - \frac{D^{(n)}}{D^{(u)}} \left(S_{j-(n-u)} + \sum_{i=1}^{L^{(u-1)}} \Lambda_i^{(u-1)} S_{j-(n-u+i)} \right) \\ & \quad = \begin{cases} 0 - \frac{D^{(n)}}{D^{(u)}} \cdot 0 = 0 & \text{pro } j = L^{(n)} + 1, L^{(n)} + 2, \ldots, n-1 \\ D^{(n)} - \frac{D^{(n)}}{D^{(u)}} \cdot D^{(u)} = 0 & \text{pro } j = n. \end{cases} \end{aligned} \tag{3.17}$$

Tedy LFSR s charakteristickým polynomem $\Lambda^{(n)}(x)$ opravdu generuje posloupnost $S_1, S_2, \ldots, S_n$. Speciálně LFSR příslušný $\Lambda^{(2t)}(x)$ generuje $S_1, S_2, \ldots, S_{2t}$. Důkaz toho, že $L^{(n)}$ je rovno minimální délce LFSR generujícího posloupnost $S_1, S_2, \ldots, S_n$, a podrobnější popis celého BM algoritmu nalezneme v [5].

Příklad 3.3. *Navažme na příklad 3.2, kde jsme vypočítali syndromy* $S_1 = 2$, $S_2 = 2$, $S_3 = 5$, $S_4 = 6$, $S_5 = 0$ *a* $S_6 = 6$. *K dekódování zprávy, resp. nalezení lokačního polynomu, použijeme Algoritmus 2. Průběh algoritmu zaznamenejme do následující tabulky:*

n	$D^{(n)}$	$\Lambda^{(n)}(x)$	$L^{(n)}$	$T(x)$
0	–	1	0	x
1	2	$5x+1$	1	$4x$
2	5	$6x+1$	1	$4x^2$
3	3	$2x^2+6x+1$	2	$2x^2+5x$
4	5	$6x^2+2x+1$	2	$2x^3+5x^2$
5	0	$6x^2+2x+1$	2	$2x^4+5x^3$
6	0	$6x^2+2x+1$	2	$2x^5+5x^4$

Našli jsme tedy lokační polynom $\Lambda(x) = \Lambda^{(6)}(x) = 6x^2+2x+1$. *Evaluační polynom můžeme dopočítat z definice dle (3.8), tj.*

$$\begin{aligned}\Omega(x) &= \Lambda(x)\cdot S(x) \mod x^6 \\ &= (6x^2+2x+1)\cdot(6x^5+0x^4+6x^3+5x^2+2x+2) \mod x^6 \\ &= 0x^5+0x^4+0x^3+0x^2+6x+2 \\ &= 6x+2.\end{aligned}$$

Celkové dokončení dekódování provedeme postupem popsaným v Příkladu 3.2.

3.4.5 Eukleidův algoritmus

V roce 1975 přišel Yasuo Sugiyama et al. [11] se zajímavou myšlenkou využít známý Eukleidův algoritmus (pro hledání největšího společného dělitele celých čísel) k řešení klíčové rovnice. Příslušný dekódovací algoritmus nazýváme taktéž Eukleidův, případně řidčeji Sugiyamův.

Nejprve si v následujícím pseudokódu připoměňme *rozšířený Eukleidův algoritmus*, jenž kromě největšího společného dělitele počítá i příslušné Bézoutovy koeficienty. Algoritmus 3 je psán ve tvaru pro obecný eukleidovský obor R s normou η.[14]

Označme N krok, kdy algoritmus terminuje (tj. $r_N = 0$). Konečnost algoritmu plyne z toho, že posloupnost norem prvků r_n je ostře klesající posloupností na přirozených číslech, a tedy musí být konečná (podrobněji např. [9]).

Jednoduchou indukcí lze ukázat, že pro každé $n = -1, 0, \ldots, N$ platí

$$r_n = u_n a + v_n b. \tag{3.18}$$

Z rovnice (kterou bychom dokázali analogicky k Lemmatu 2.2)

$$u_{n+1}v_n - v_{n+1}u_n = (-1)^n \quad \text{pro } n = -1, 0, \ldots, N \tag{3.19}$$

navíc plyne, že u_n a v_n jsou vždy nesoudělné.

[14] Nejsou-li čtenáři užité pojmy známy, vysvětlení najde v libovolné učebnici základního kurzu algebry, např. [9].

Algoritmus 3. Rozšířený Eukleidův algoritmus

Input: $a, b \in R$, $\eta(a) \geq \eta(b)$

Output: $\gcd(a, b)$

$u, v \in R$ takové, že $ua + vb = \gcd(a, b)$

0. $n \leftarrow 0$, $r_{-1} \leftarrow a$, $u_{-1} \leftarrow 1$, $v_{-1} \leftarrow 0$,
   $r_0 \leftarrow b$, $u_0 \leftarrow 0$, $v_0 \leftarrow 1$
1. **while** $r_n \neq 0$ **do**
2. $n \leftarrow n + 1$
3. najdi $q_n, r_n \in R$ takové, že $r_{n-2} = q_n \cdot r_{n-1} + r_n$ a $\eta(r_n) < \eta(r_{n-1})$
4. $u_n \leftarrow u_{n-2} - q_n \cdot u_{n-1}$, $v_n \leftarrow v_{n-2} - q_n \cdot v_{n-1}$
5. **end while**
6. **return** $\gcd(a, b) = r_{n-1}$, $u = u_{n-1}$, $v = v_{n-1}$

V naší konkrétní situaci představuje obor R okruh polynomů $\mathbb{F}_q[x]$ a normě η odpovídá stupeň polynomu. Polynomy q a r ve 3. kroku Algoritmu 2 jsou podíl a zbytek při standardním dělení se zbytkem. Budeme-li mluvit o polynomech, budeme nadále psát $a(x)$ místo a, $b(x)$ místo b apod.

Přejděme nyní k hlavnímu problému – klíčové rovnici. Za znalosti syndromového polynomu $S(x)$ chceme nalézt nesoudělné polynomy $\Lambda(x)$ a $\Omega(x)$, jejichž stupně splňují nerovnosti $\deg \Lambda(x) \leq t$ a $\deg \Omega(x) < t$, $\Lambda(0) = 1$ a platí rovnice

$$\Omega(x) = S(x)\Lambda(x) \mod x^{2t}.$$

Aplikujeme-li rozšířený Eukleidův algoritmus na polynomy $a(x) = x^{2t}$ a $b(x) = S(x)$, v každém kroku $n = -1, 0, \ldots, N$ podle (3.18) máme

$$r_n(x) = u_n(x) \cdot x^{2t} + v_n(x) \cdot S(x), \tag{3.20}$$

což je ekvivalentní

$$r_n(x) = v_n(x) \cdot S(x) \mod x^{2t}. \tag{3.21}$$

Pokud tedy nalezneme v Eukleidově algoritmu iteraci, kdy polynomy $v_n(x)$ a $r_n(x)$ vyhovují podmínkám pro $\Lambda(x)$ a $\Omega(x)$, máme efektivní dekódovací algoritmus řešící klíčovou rovnici.

Pro $n = 1, \ldots, N$ zjevně platí

$$\deg v_n(x) = \sum_{i=1}^{n} \deg q_i(x) \tag{3.22}$$

a také

$$\deg q_n(x) = \deg r_{n-2}(x) - \deg r_{n-1}(x). \tag{3.23}$$

Tudíž

$$\begin{aligned}\deg r_{n-1}(x) &= \deg r_{n-2}(x) - \deg q_n(x)\\ &= (\deg r_{n-3}(x) - \deg q_{n-1}(x)) - \deg q_n(x)\\ &\vdots\\ &= \deg r_{-1}(x) - \sum_{i=1}^{n} \deg q_i(x)\\ &= \deg a(x) - \deg v_n(x).\end{aligned} \tag{3.24}$$

Pokud ukončíme Eukleidův algoritmus v nejmenší iteraci $K > 0$, pro kterou $\deg r_K(x) < t$, potom $\deg r_{K-1}(x) \geq t$, což implikuje

$$\deg v_K(x) = \deg a(x) - \deg r_{K-1}(x) \leq 2t - t = t. \tag{3.25}$$

Takto zvolené polynomy $v_K(x)$ a $r_K(x)$ tedy splňují podmínky omezenosti stupňů. Nesoudělnost této dvojice polynomů plyne přímo z (3.18) a (3.19). Přenásobíme-li nakonec $v_K(x)$ a $r_K(x)$ inverzem absolutního členu[15] polynomu $v_K(x)$ (tj. $v_K(0)^{-1}$), budou evidentně splněny všechny tři podmínky a můžeme položit

$$\Lambda(x) = v_K(0)^{-1} \cdot v_K(x), \tag{3.26}$$

$$\Omega(x) = v_K(0)^{-1} \cdot r_K(x). \tag{3.27}$$

Důkaz, že takto nalezené polynomy $\Lambda(x)$ a $\Omega(x)$ jsou až na asociovanost jediné, podává např. [2]. Všimněme si ještě, že výpočet druhého Bézoutova koeficientu $u_n(x)$ je zde nadbytečný. Výsledný dekódovací algoritmus proto vypadá takto:[16]

Algoritmus 4. Eukleidův (Sugiyamův) dekódovací algoritmus

Input: syndromový polynom $S(x) = S_1 + S_2x + \cdots + S_{2t}x^{2t-1}$

Output: lokační polynom $\Lambda(x)$ a evaluační polynom $\Omega(x)$

0. $n \leftarrow 0$, $r_{-1}(x) \leftarrow x^{2t}$, $r_0(x) \leftarrow S(x)$, $v_{-1}(x) \leftarrow 0$, $v_0(x) \leftarrow 1$
1. **while** $\deg r_n(x) \geq t$ **do**
2. $\quad n \leftarrow n + 1$
3. $\quad q_n(x) \leftarrow [r_{n-2}(x)/r_{n-1}(x)]$
4. $\quad r_n(x) \leftarrow r_{n-2}(x) - q_n(x) \cdot r_{n-1}(x)$
5. $\quad v_n(x) \leftarrow v_{n-2}(x) - q_n(x) \cdot v_{n-1}(x)$
6. **end while**
7. **return** $\Lambda(x) = v_n(0)^{-1}v_n(x)$, $\Omega(x) = v_n(0)^{-1}r_n(x)$

[15] Pokud by $v_K(0) = 0$, bylo by i $r_K(0) = 0$. Avšak to je spor s nesoudělností těchto polynomů.

[16] Prakticky by se dal počet proměnných v algoritmu výrazně omezit, zde však z důvodu lepšího odkazování na mezivýpočty volíme tento způsob zápisu. Celočíselné dělení x div y tu značíme $[x/y]$, čímž odkazujeme na stejnou operaci užívanou při výpočtu řetězových zlomků.

Příklad 3.4. *Budeme dekódovat stejnou zprávu jako v Příkladu 3.3, tentokrát však použijeme Algoritmus 4. Syndromový polynom je roven*

$$S(x) = 2 + 2x + 5x^2 + 6x^3 + 0x^4 + 6x^5.$$

Postupné řešení opět zapišme do tabulky:

n	$r_n(x)$	$v_n(x)$	$q_n(x)$
-1	x^6	0	$-$
0	$6x^5 + 0x^4 + 6x^3 + 5x^2 + 2x + 2$	1	$-$
1	$6x^4 + 5x^3 + 2x^2 + 2x + 0$	x	$6x$
2	$6x + 2$	$6x^2 + 2x + 1$	$x + 5$

Po dvou cyklech algoritmu jsme opět dospěli ke správným výsledkům:

$$\Lambda(x) = v_2(0)^{-1} v_2(x) = 6x^2 + 2x + 1,$$

$$\Omega(x) = v_2(0)^{-1} r_2(x) = 6x + 2.$$

3.4.6 Algoritmus s řetězovými zlomky

V Kapitole 2 jsme vystavěli dobrý teoretický základ teorie řetězových zlomků, který nyní užijeme k řešení klíčové rovnice. Podívejme se na ni však ještě z jiného pohledu než doposud. Pro formální mocninné řady platí

$$\frac{1}{1 - X_k x} = 1 + X_k x + X_k^2 x^2 + \dots. \tag{3.28}$$

Nechť X_k a Y_k značí pozici a velikost k-té chyby. Přenásobením rovnice (3.28) $Y_k X_k$ a sečtením přes $k = 1$ do ν dostáváme podle vzorce (3.3)

$$\sum_{k=1}^{\nu} \frac{Y_k X_k}{1 - X_k x} = S_1 + S_2 x + S_3 x^2 + \dots. \tag{3.29}$$

Levou stranu rovnice ještě převeďme na společného jmenovatele:

$$\frac{\sum_{k=1}^{\nu} Y_k X_k \prod_{j \neq k} (1 - X_j x)}{\prod_{k=1}^{\nu} (1 - X_k x)} = S_1 + S_2 x + S_3 x^2 + \dots, \tag{3.30}$$

což je podle definice lokačního a evaluačního polynomu ekvivalentní

$$\frac{\Omega(x)}{\Lambda(x)} = S_1 + S_2 x + S_3 x^2 + \dots + S_{2t} x^{2t-1} + \dots. \tag{3.31}$$

Poslední rovnice zjevně odpovídá klíčové rovnici. Mocninnou řadu na pravé straně rovnice (3.31) na okamžik označme $S'(x)$.

Abychom při práci se $S'(x)$ mohli použít aparát řetězových zlomků vybudovaný v Kapitole 2, zaměníme proměnnou x za x^{-1}. Řada $S'(x^{-1})$ už leží v tělese $F(\!(x)\!)$ Laurentových formálních mocninných řad (viz sekce 2.1).

Zaveďme *syndromovou řadu* $S^*(x) = x^{-1} \cdot S'(x^{-1})$ z důvodů, které budou zřejmé v zápětí. Víme, že $S^*(x)$ je racionální mocninná řada, neboť $S^*(x) = \widehat{\Omega}(x)/\widehat{\Lambda}(x)$, kde $\widehat{\Omega}(x)$ a $\widehat{\Lambda}(x)$ jsou polynomy z $F[x]$ definované jako

$$\widehat{\Omega}(x) = x^{\nu-1} \cdot \Omega(x^{-1}), \tag{3.32}$$

$$\widehat{\Lambda}(x) = x^{\nu} \cdot \Lambda(x^{-1}). \tag{3.33}$$

Pozorování 3.1. *Polynomy $\widehat{\Lambda}(x)$ a $\Lambda(x)$ jsou stejného stupně, mají pouze obrácené pořadí koeficientů. Podobně je tomu u $\widehat{\Omega}(x)$ a $\Omega(x)$, jen zde není zaručena rovnost stupňů (neboť absolutní člen $\Omega(x)$ může být roven 0).*

Rovnost $S^*(x) = \widehat{\Omega}(x)/\widehat{\Lambda}(x)$ ověříme snadno:

$$\frac{\widehat{\Omega}(x)}{\widehat{\Lambda}(x)} = \frac{x^{\nu-1} \cdot \Omega(x^{-1})}{x^{\nu} \cdot \Lambda(x^{-1})} = x^{-1} \cdot S'(x^{-1}) = S^*(x). \tag{3.34}$$

Podívejme se podrobněji, jak vypadají $\widehat{\Lambda}(x)$ a $\widehat{\Omega}(x)$. Z definic máme

$$\widehat{\Lambda}(x) = x^{\nu} \cdot \Lambda(x^{-1}) = x^{\nu} \cdot \prod_{k=1}^{\nu}(1 - x^{-1}X_k) = \prod_{k=1}^{\nu}(x - X_k), \tag{3.35}$$

$$\begin{aligned} \widehat{\Omega}(x) &= x^{\nu-1} \cdot \Omega(x^{-1}) = x^{\nu-1} \cdot \sum_{k=1}^{\nu} Y_k X_k \prod_{j \neq k}(1 - X_j x^{-1}) \\ &= \sum_{k=1}^{\nu} Y_k X_k \prod_{j \neq k}(x - X_j). \end{aligned} \tag{3.36}$$

Opět se jedná o dvojici nesoudělných polynomů, protože očividně nemají žádné společné kořeny. Navíc kořeny $\widehat{\Lambda}(x)$ jsou právě pozice chyb X_k.

$S^*(x) = \widehat{\Omega}(x)/\widehat{\Lambda}(x)$ je tedy racionální mocninná řada, kde $\widehat{\Omega}(x)$ a $\widehat{\Lambda}(x)$ jsou nesoudělné, čímž jsou předpoklady Důsledku 2.10 splněny. O $S^*(x)$ nemáme úplnou informaci, známe pouze prvních $2t$ koeficientů této řady. Nejvyšší exponent s neznámým koeficientem je $-2t-1$. Podle Důsledku 2.10 jsou polynomy $\widehat{\Omega}(x)$ a $\widehat{\Lambda}(x)$ řadou $S^*(x)$ jednoznačně (až na asociovanost) určeny, právě když $\deg \widehat{\Lambda}(x) < \frac{2t+1}{2}$.

Protože $\deg \Lambda(x) = \nu \leq t < \frac{2t+1}{2}$, stačí na syndromovou řadu $S^*(x)$ aplikovat Algoritmus 1 (v malé úpravě) a na výstupu dostaneme polynomy $\widehat{\Omega}(x)$ a $\widehat{\Lambda}(x)$ (případně jejich násobek o konstantu). Polynomy $\Omega(x)$ a $\Lambda(x)$ pak dostaneme z výstupních přenásobením vhodnou mocninou x a inverzem vedoucího koeficientu $\Lambda(x)$, jenž v Algoritmu 5 značíme jako λ.

Dle Tvrzení 2.9 a následné diskuze budou polynomy $\widehat{\Omega}(x)$ a $\widehat{\Lambda}(x)$, jejichž podílu se $S^*(x)$ rovná, určeny po $\deg \widehat{\Lambda}(x) = \nu$ opakováních cyklu algoritmu řetězových zlomků. V ν-té iteraci již tedy bude $\Delta_\nu(x) = 0$, což bude i terminační podmínka Algoritmu 5. Protože $\nu \leq t$, stane se tak nejpozději v t-tém cyklu algoritmu. Dekódovací algoritmus a ilustrační příklad jsou uvedeny na následující stránce.

Algoritmus 5. Dekódovací algoritmus s řetězovými zlomky

Input: $S^*(x) = S_1x^{-1} + S_2x^{-2} + \cdots + S_{2t}x^{-2t} + X^{-2t-1}$

Output: lokační polynom $\Lambda(x)$ a evaluační polynom $\Omega(x)$

0. $P_{-1}(x) \leftarrow 1$, $Q_{-1}(x) \leftarrow 0$, $\Delta_{-1}(x) \leftarrow -1$
   $P_0(x) \leftarrow 0$, $Q_0(x) \leftarrow 1$, $\Delta_0(x) \leftarrow S^*(x)$, $n \leftarrow 0$
1. **while** $\Delta_n(x)$ má známé nenulové koeficienty **do**
2. $\quad n \leftarrow n + 1$
3. $\quad a_n(x) \leftarrow -\left[\frac{\Delta_{n-2}(x)}{\Delta_{n-1}(x)}\right]$
4. $\quad P_n(x) \leftarrow a_n(x)P_{n-2}(x) + P_{n-1}(x)$
   $\quad Q_n(x) \leftarrow a_n(x)Q_{n-2}(x) + Q_{n-1}(x)$
   $\quad \Delta_n(x) \leftarrow a_n(x)\Delta_{n-2}(x) + \Delta_{n-1}(x)$
5. **end while**
6. **return** $\Lambda(x) = x^\nu \lambda Q_n(x^{-1})$, $\Omega(x) = x^{\nu-1}\lambda P_n(x^{-1})$,
   kde $\nu = \deg Q_n(x)$ a $\lambda = \mathrm{lc}^{-1}(Q_n(x))$

Příklad 3.5. *Na řešení stejného zadání jako v Příkladu 3.4 použijeme v tomto případě Algoritmus 5. Syndromová řada je rovna*

$$S^*(x) = 2x^{-1} + 2x^{-2} + 5x^{-3} + 6x^{-4} + 0x^{-5} + 6x^{-6} + X^{-7},$$

kde X^l, $l \in \mathbb{Z}$, značí mocninnou řadu v proměnné x stupně nejvýše l (pro nás s neznámými koeficienty).

Průběžné hodnoty proměnných zanesme do následující tabulky:

n	$P_n(x)$	$Q_n(x)$	$\Delta_n(x)$	$a_n(x)$
-1	1	0	-1	$-$
0	0	1	$2x^{-1} + 2x^{-2} + 5x^{-3} + 6x^{-4} + 0x^{-5} + 6x^{-6} + X^{-7}$	0
1	1	$4x + 3$	$5x^{-2} + 4x^{-3} + 4x^{-4} + 3x^{-5} + X^{-6}$	$4x + 3$
2	$x + 3$	$4x^2 + x + 3$	X^{-5}	$x + 3$

Úpravami uvedenými v kroku 6 Algoritmu 5 dostáváme očekávané řešení

$$\Lambda(x) = x^2 \cdot 4^{-1} \cdot (4x^{-2} + x^{-1} + 3) = 1 + 2x + 6x^2,$$

$$\Omega(x) = x \cdot 4^{-1} \cdot (x^{-1} + 3) = 2 + 6x.$$

Kapitola 4

Ekvivalence algoritmů

Souvislost Eukleidova algoritmu s řetězovými zlomky je při bližším pohledu evidentní. Oproti tomu iterace Berlekampova-Masseyova algoritmu nedávají ani při podrobnějším zkoumání příliš najevo, že by nějak korespondovaly se zbylými dvěma algoritmy. Přesto zde platí až pozoruhodná ekvivalence.

V této kapitole využijeme poznatků provedených W. L. Eastmanem v knize [2]. Již její název „Eukleidovské dekodéry BCH kódů“ poukazuje na společný teoretický základ zmiňovaných tří dekódovacích algoritmů. Nedávno vydaný článek [6] T. D. Mattera pak prezentuje názorné algoritmy, díky nimž jsou důkazy ekvivalence ihned srozumitelnější.

V závěru Kapitoly 3 jsme zavedli operaci stříšky $\widehat{\ }$, která ve spojení s polynomy znamenala převrácení jejich koeficientů (a případně změnu stupně polynomu), tj.

$$p(x) = \sum_{i=0}^{n} p_i x^i \Longrightarrow \widehat{p}(x) = \sum_{i=0}^{n} p_{n-i} x^i = x^n p(x^{-1}). \tag{4.1}$$

Pozorování 4.1. *Buďte* $a(x) = \sum_{i=0}^{n_1} a_i x^i$, $b(x) = \sum_{j=0}^{n_2} b_j x^j$ *polynomy nad nějakým tělesem. Nechť*

$$a(x)b(x) = c(x) = \sum_{k=0}^{n} c_k x^k, \tag{4.2}$$

kde

$$n = n_1 + n_2,\ c_k = \sum_{i+j=k} a_i b_j. \tag{4.3}$$

Pak

$$\widehat{a}(x)\widehat{b}(x) = d(x) = \sum_{k=0}^{n} d_k x^k, \tag{4.4}$$

kde

$$n = n_1 + n_2,\ d_k = \sum_{\substack{(n_1-i)+\\+(n_2-j)=k}} a_i b_j = \sum_{i+j=n-k} a_i b_j = c_{n-k}. \tag{4.5}$$

Tedy platí rovnost

$$\widehat{a}(x)\widehat{b}(x) = \widehat{c}(x) = \widehat{a(x)b(x)}. \tag{4.6}$$

Podívejme se na pravou stranu klíčové rovnice (3.7). Bez operace mod x^{2t} je rovna

$$\begin{aligned}\Lambda(x)S(x) &= \sum_{k=0}^{2t+\nu-1}\sum_{i=0}^{k}\Lambda_i S_{k+1-i}x^k \\ &= \sum_{k=0}^{\nu-1}\sum_{i=0}^{k}\Lambda_i S_{k+1-i}x^k + \sum_{k=\nu}^{2t-1}\sum_{i=0}^{k}\Lambda_i S_{k+1-i}x^k + \sum_{k=2t}^{2t+\nu-1}\sum_{i=0}^{k}\Lambda_i S_{k+1-i}x^k \\ &\overset{(3.6)}{=} \sum_{k=0}^{\nu-1}\sum_{i=0}^{k}\Lambda_i S_{k+1-i}x^k + 0 + \sum_{k=2t}^{2t+\nu-1}\sum_{i=0}^{k}\Lambda_i S_{k+1-i}x^k \\ &= \Omega(x) + x^{2t}A(x),\end{aligned} \tag{4.7}$$

kde $A(x)$ je nějaký polynom stupně nejvýše $\nu - 1$.

Aplikací (4.6) na (4.7) obdržíme[17]

$$\widehat{\Lambda}(x)\widehat{S}(x) = \widehat{\Lambda(x)S(x)} = x^{2t}\widehat{\Omega}(x) + \widehat{A}(x). \tag{4.8}$$

Poznámka. *Na okraj poznamenejme, že znalost $A(x)$ je ekvivalentní znalosti $\Omega(x)$, což ukazují následující dvě rovnice:*

$$S(x) = \frac{\Omega(x) + x^{2t}A(x)}{\Lambda(x)} = \left[\frac{x^{2t}A(x)}{\Lambda(x)}\right], \tag{4.9}$$

protože $\deg\Lambda(x) = \nu$ *a* $\deg\Omega(x) \le \nu - 1$ *([ ] značí celou část). Podobně*

$$\widehat{S}(x) = \frac{x^{2t}\widehat{\Omega}(x) + \widehat{A}(x)}{\widehat{\Lambda}(x)} = \left[\frac{x^{2t}\widehat{\Omega}(x)}{\widehat{\Lambda}(x)}\right]. \tag{4.10}$$

Příklad 4.1. *Funkci polynomu $A(x)$ ilustrujme na úloze se stejným zadáním jako Příklad 3.5, tj. syndromový polynom nechť je*

$$S(x) = 2 + 2x + 5x^2 + 6x^3 + 0x^4 + 6x^5$$

a lokační polynom

$$\Lambda(x) = 1 + 2x + 6x^2.$$

Vynásobením těchto dvou polynomů získáváme

$$\Lambda(x)S(x) = 2 + 6x + 0x^2 + 0x^3 + 0x^4 + 0x^5 + 5x^6 + 1x^7,$$

a tedy v souladu s (4.7)

$$\Omega(x) = 2 + 6x, \quad A(x) = 5 + x.$$

Platnost předcházející Poznámky předvedeme pro rovnici (4.9):

$$\begin{aligned}\left[\frac{x^6A(x)}{\Lambda(x)}\right] &= \left[\frac{x^7 + 5x^6}{6x^2 + 2x + 1}\right] \\ &= \left[6x^5 + 0x^4 + 6x^3 + 5x^2 + 2x + 2 + 6x^{-1} + 0x^{-2} + 6x^{-3} + \dots\right] \\ &= 6x^5 + 0x^4 + 6x^3 + 5x^2 + 2x + 2 = S(x).\end{aligned}$$

[17] Definujeme $\widehat{\Omega}(x) = x^{\nu-1}\Omega(x^{-1})$ (resp. $\widehat{A}(x) = x^{\nu-1}A(x^{-1})$) i v případě, jsou-li nižšího stupně než $\nu - 1$.

Ve světle nově zavedeného polynomu $A(x)$ se podívejme na Eukleidův dekódovací algoritmus. V popisu Algoritmu 4 jsme vynechali výpočet druhého Bézoutova koeficientu u_n. Pro něj by při terminaci algoritmu v kroku K podle (3.20) platilo

$$u_K(x)x^{2t} + v_K(x)S(x) = r_K(x) \tag{4.11}$$

neboli (po přenásobení $v_n(0)^{-1}$, abychom znormalizovali $\Lambda(x)$ a $\Omega(x)$)

$$v_K(0)^{-1}u_K(x)x^{2t} + \Lambda(x)S(x) = \Omega(x), \tag{4.12}$$

a tedy dle (4.8)

$$A(x) = -v_K(0)^{-1}u_K(x). \tag{4.13}$$

Užijeme-li však Algoritmus 4 na polynom $\widehat{S}(x)$, při terminaci v kroku K' máme

$$u'_{K'}(x)x^{2t} + v'_{K'}(x)\widehat{S}(x) = r'_{K'}(x). \tag{4.14}$$

Podle (4.11) a sekce 3.4.5 je

$$\begin{array}{rcccccl} \lambda v'_{K'}(x) & = & \widehat{\Lambda}(x) & = & x^{\nu}\Lambda(x^{-1}) & = & x^{\nu}v_K(0)^{-1}v_K(x^{-1}), \\ -\lambda u'_{K'}(x) & = & \widehat{\Omega}(x) & = & x^{\nu-1}\Omega(x^{-1}) & = & x^{\nu-1}v_K(0)^{-1}r_K(x^{-1}), \\ \lambda r'_{K'}(x) & = & \widehat{A}(x) & = & x^{\nu-1}A(x^{-1}) & = & -x^{\nu-1}v_K(0)^{-1}u_K(x^{-1}). \end{array} \tag{4.15}$$

kde $\lambda = \mathrm{lc}^{-1}(v'_{K'}(x))$ stejně jako v algoritmu řetězových zlomků značí inverz vedoucího koeficientu $v'_{K'}(x)$.

Příklad 4.2. *Na polynom $\widehat{S}(x) = 2x^5+2x^4+5x^3+6x^2+0x+6$ daný Příkladem 4.1 aplikujme Algoritmus 4 upravený pro výpočet i druhého Bézoutova koeficientu $u_n(x)$.*

n	$r_n(x)$	$u_n(x)$	$v_n(x)$	$q_n(x)$
-1	x^6	1	0	$-$
0	$2x^5 + 2x^4 + 5x^3 + 6x^2 + 0x + 6$	0	1	$-$
1	$2x^4 + 3x^3 + 3x^2 + 4x + 3$	1	$3x + 4$	$4x + 3$
2	$6x + 4$	$6x + 4$	$4x^2 + 1x + 3$	$x + 3$

Upravením výsledných polynomů dle (4.15) opět dostáváme řešení klíčové rovnice.

Důsledek 4.1. *Algoritmus 4 se vstupem $S(x)$ je ekvivalentní tomuto algoritmu se vstupem $\widehat{S}(x)$ ve smyslu (4.15).*

Operace $\widehat{\ }$ nezachovává sčítání, proto mezivýsledky v průběhu Algoritmu 4 se vstupy $S(x)$ a $\widehat{S}(x)$ na sebe takto jednoduše převést nelze.

Provedli jsme několik pozorování týkajících se Eukleidova algoritmu, která nyní uplatníme při zkoumání souvislosti tohoto algoritmu s řetězovými zlomky.

Lemma 4.2. *Nechť $f = P/Q$ je racionální prvek tělesa $F(\!(x)\!)$, kde P, Q jsou polynomy z okruhu $F[x]$ takové, že $\deg P < \deg Q$. Potom výpočet aproximací f řetězovými zlomky Algoritmem 1 je identický výpočtu největšího společného dělitele P a Q rozšířeným Eukleidovým algoritmem.*

Důkaz. Pro řetězové zlomky užijeme značení dané (2.1) a pro rozšířený Eukleidův algoritmus značení z Algoritmu 3. Identita ze zadání je myšlena ve smyslu

- $r_n/Q = (-1)^n \cdot \Delta_n$,
- $u_n = (-1)^{n+1} \cdot P_n$,
- $v_n = (-1)^n \cdot Q_n$ pro $n = -1, 0, \ldots, N$,
- $a_n = q_n$ pro $n = 1, 2, \ldots, N$,

kde N značí krok terminace (jenž je stejný pro oba algoritmy).

Tyto rovnosti dokážeme indukcí podle n. Platí

$$\begin{aligned} r_{-1}/Q &= Q/Q = 1 = -\Delta_{-1}, & r_0/Q &= P/Q = f = f - [f] = \Delta_0, \\ u_{-1} &= 1 = P_{-1}, & u_0 &= 0 = -[f] = -P_0, \\ v_{-1} &= 0 = -Q_{-1}, & v_0 &= 1 = Q_0. \end{aligned} \tag{4.16}$$

Buď $n \geq 1$ a nechť jsou zadané rovnosti splněny pro $n-1$. Pak

$$\begin{aligned} a_n &= \left[\frac{r_{n-2}}{r_{n-1}}\right] = \left[\frac{(-1)^{n-2}\Delta_{n-2}}{(-1)^{n-1}\Delta_{n-1}}\right] = \left[-\frac{\Delta_{n-2}}{\Delta_{n-1}}\right] = q_n, \\ r_n &= r_{n-2} - q_n r_{n-1} = (-1)^{n-2} Q \Delta_{n-2} - a_n (-1)^{n-1} Q \Delta_{n-1} \\ &= (-1)^n Q \left(\Delta_{n-2} + a_n \Delta_{n-1}\right) = (-1)^n Q \Delta_n. \end{aligned} \tag{4.17}$$

Pro u_n a v_n je důkaz analogický k r_n, a tedy lemma platí. □

Příklad 4.3. *Porovnej průběhy algoritmů uvedených v Příkladu 3.5 a Příkladu 4.2.*

Vstupem do Algoritmu 5 – dekódovacího algoritmu s řetězovými zlomky – je syndromová řada $S^*(x) = \Omega(x)/\Lambda(x)$. Podle Tvrzení 2.9 nám dá Algoritmus 5 stejný výsledek, použijeme-li místo $S^*(x)$ řadu $\widehat{S}(x)/x^{2t}$.[18]

Přitom dle Lemmatu 4.2 je výpočet aproximací $\widehat{S}(x)/x^{2t}$ (jedná se o podíl dvou polynomů, tedy racionální prvek tělesa $F(\!(x)\!)$) řetězovými zlomky identický výpočtu největšího společného dělitele $\widehat{S}(x)$ a x^{2t}.

Dále rozšířený Eukleidův algoritmus se vstupními polynomy $\widehat{S}(x)$ a x^{2t} obsahuje (v prvních K' iteracích) celý proces Eukleidova dekódovacího algoritmu užitého na $\widehat{S}(x)$. A nakonec podle Důsledku 4.1 je Algoritmus 4 (Eukleidův dekódovací) se vstupem $\widehat{S}(x)$ ekvivalentní tomuto algoritmu se vstupem $S(x)$.

Důsledek 4.3. *Dekódování RS kódů pomocí řetězových zlomků je ekvivalentní dekódování Eukleidovým algoritmem.*

Zbývá dokázat shodnost těchto dvou algoritmů s Berlekampovým-Masseyovým. Nejnázornější se jeví začít s rozšířeným Eukleidovým algoritmem aplikovaným na polynom $\widehat{S}(x)$ (jenž je dle předchozí části v podstatě identický algoritmu řetězových zlomků). Vybíráme tento, neboť chceme, aby do procesu dekódování vstupovaly syndromy S_i v pořadí od $i = 1$ do $2t$. Takto pracuje právě Berlekampův-Masseyův narozdíl od standardního Eukleidova dekódovacího algoritmu.

[18] $\widehat{S}(x)/x^{2t}$ lze považovat za mocninnou řadu s nekonečně mnoha nulovými koeficienty počínaje koeficientem u x^{-2t-1}. Předpoklad Tvrzení 2.9 je splněn, neboť $\|S^*(x) - \widehat{S}(x)/x^{2t}\| < 2^{-2t}$.

Učiňme několik pozorování, která využijeme k modifikaci rozšířeného Eukleidova algoritmu. Tento nový algoritmus bude obsahovat všechny původní kroky, z nichž však některé budou rozepsány do více iterací tak, aby měl mezivýsledky totožné s BM algoritmem.

Pozorování 4.2.

(1) Eukleidův algoritmus můžeme inicializovat $r_{-1}(x) = a(x) + b(x)$*, chceme-li upravit výchozí hodnotu* $v_{-1}(x) = 1$*.*

(2) Rovnost $r_{n-2}(x) = u_{n-2}(x)a(x) + v_{n-2}(x)b(x)$ *lze přenásobit libovolnou nenulovou konstantou* C*, abychom dostali*

$$Cr_{n-2}(x) = Cu_{n-2}(x)a(x) + Cv_{n-2}(x)b(x). \tag{4.18}$$

V případě takovéto úpravy se pak změní i hodnoty Bézoutových koeficientů

$$u_n(x) = Cu_{n-2}(x) - q_n(x)u_{n-1}(x), \tag{4.19}$$

$$v_n(x) = Cv_{n-2}(x) - q_n(x)v_{n-1}(x). \tag{4.20}$$

Jak zanedlouho ukážeme, mezivýsledky Eukleidova a BM algoritmu se liší pouze o násobek nějakou konstantou. A právě vhodnou volbou konstanty C *budou výsledky odpovídat buď jednomu nebo druhému algoritmu.*

Vzpomeňme si na školské dělení se zbytkem: vydělíme vedoucí koeficienty dělence a dělitele a tento podíl zapíšeme do výsledku, tímto číslem zpátky přenásobíme dělitele atd. Přesně toto budou popisovat následující body pozorování a lze si je takto i snadno představit.

(3) Vedoucí členy $r_{n-2}(x)$ *a* $r_{n-1}(x)$ *buďte* $D_{n-2}x^\delta$ *a* $D_{n-1}x^\gamma$*. (*D *je úmyslně vybráno, aby korespondovalo s diskrepancí Berlekampova-Masseyova algoritmu.) Je-li* $r_{n-2}(x)$ *vynásobeno* C*, pak vedoucí člen* $q_n(x)$ *je dán jako*

$$q_n^{(1)}(x) = \frac{CD_{n-2}x^\delta}{D_{n-1}x^\gamma}. \tag{4.21}$$

Pokud $q_n(x) = q_n^{(1)}(x)$*, pak* $r_n^{(1)}(x)$ *bude stupně menšího než* $\deg r_{n-1}(x)$ *a můžeme postoupit k následujícímu cyklu* $n+1$ *rozšířeného Eukleidova algoritmu. V opačném případě je vedoucí koeficient* $r_n^{(1)}(x)$ *diskrepance, jíž užijeme k pozměnění* $q_n^{(1)}(x)$*.*

(4) Předpokládejme, že máme daný $(j-1)$*-tý odhad pro* $q_n(x)$ *a* $r_n(x)$*, tj.* $q_n^{(j-1)}(x)$ *a* $r_n^{(j-1)}(x)$*, a chtěli bychom určit* j*-tý odhad těchto polynomů. Pokud vedoucí členy* $r_n^{(j-1)}(x)$ *a* $r_{n-1}(x)$ *jsou po řadě*[19] D_nx^γ *a* $D_{n-1}x^\delta$*, pak*

$$\frac{D_n}{D_{n-1}}x^{\gamma-\delta}r_{n-1}(x) \tag{4.22}$$

je polynom, jenž má shodný vedoucí člen s $r_n^{(j-1)}(x)$*. Proto volíme*

$$q_n^{(j)}(x) = q_n^{(j-1)}(x) + \frac{D_n}{D_{n-1}}x^{\gamma-\delta}, \tag{4.23}$$

[19]Proměnné δ a γ jsou v tomto případě použity v jiném kontextu než v bodě (3).

$$r_n^{(j)}(x) = Cr_{n-2}(x) - q_n^{(j)}(x)r_{n-1}(x), \tag{4.24}$$

abychom dostali polynom $r_n^{(j)}(x)$ *stupně menšího než* $\deg r_n^{(j-1)}(x)$*. Zjevně po konečném počtu iterací dostaneme kvocient a zbytek rovné* $q_n(x)$ *a* $r_n(x)$*.*

Předchozí pozorování mohou být použita na libovolnou aplikaci Eukleidova algoritmu. Nás však zajímá pouze případ, kdy $a(x) = x^{2t}$*. Navíc, jestliže chceme nalézt pouze „lokační“ polynom, můžeme se omezit jen na výpočet* $v_n(x)$*.*

(5) Nechť $a(x) = x^{2t}$*. Pak pro každé* $n \geq 1$ *je polynom* $r_n(x)$ *stupně menšího než* $2t$*. Pokud* $b(x) = \widehat{S}(x)$*, pak*

$$r_n(x) = v_n(x)\widehat{S}(x) \mod x^{2t}. \tag{4.25}$$

Navíc je-li L_n *stupeň* $v_n(x)$*, pak koeficient stupně* γ *v* $r_n(x)$ *je dán sumou*

$$\sum_{j=0}^{L_n} (v_n)_j \cdot S_{2t-(\gamma-j)}, \tag{4.26}$$

kde $(v_n)_j$ *značí koeficient stupně* j *ve* $v_n(x)$ *a* $S_{2t-(\gamma-j)}$ *koeficient stupně* $\gamma - j$ *v* $\widehat{S}(x) = S_{2t} + S_{2t-1}x + \cdots + S_1 x^{2t-1}$*.*

Nyní předvedeme upravenou verzi Eukleidova algoritmu (viz Algoritmus 6), která bude produkovat stejné mezivýsledky jako Berlekampův-Masseyův algoritmus.

Každý běh cyklu „while“ začne vypočtením vedoucího koeficientu $r_n(x)$ z polynomů $v_n(x)$ a $\widehat{S}(x)$ podle vzorce (4.26)

$$\sum_{j=0}^{L_n} (v_n)_j \cdot S_{2t-(\gamma-j)},$$

kde γ je stupeň předpokládaného vedoucího členu $r_n(x)$. Je-li výsledek roven nule, snižujeme γ (tím, že navyšujeme i), dokud nenalezneme nenulový koeficient. Jakmile tento koeficient nalezneme, porovnáme jeho stupeň se stupněm $r_{n-1}(x)$, jenž je uložen v proměnné δ.

Pokud $\gamma < \delta$, je $r_n(x)$ stupně menšího než dělitel $r_{n-1}(x)$, je tedy přípustným zbytkem po dělení a můžeme postoupit k dalšímu kroku dělení v Eukleidově algoritmu. Zvýšíme hodnotu proměnné n a spočítáme náš první odhad $v_n(x)$ užitím (4.20) a (4.21), kde C je zvolená nenulová konstanta. Odhady podílu $q_n^{(j)}(x)$, ani výsledný podíl $q_n(x)$ do paměti neukládáme, stejnou informaci totiž obsahují $v_n^{(j)}(x)$ a $v_n(x)$ (pro které navíc používáme jen jednu proměnnou $v_n(x)$, kterou postupně přepisujeme). V dalším opakování „while“ cyklu zjistíme, zda byl náš odhad $v_n(x)$ správný.

Mezitím je také užitečné ukládat si normalizovaný (tj. s vedoucím koeficientem rovným 1) polynom $v_{n-1}(x)$ v proměnné $T(x)$. Taktéž ukládáme stupeň posledního polynomu r_n do proměnné δ a stejně tak i stupeň $v_n(x)$ do L_n.

Algoritmus 6. Modifikovaný rozšířený Eukleidův algoritmus

Input: $\widehat{S}(x) = S_1 x^{2t-1} + S_2 x^{2t-2} + \cdots + S_{2t}$

Output: lokační polynom $\Lambda(x)$ (při volbě $C = D_{n-2}/D_{n-1}$)

0. $n \leftarrow 0$, $v_0(x) \leftarrow 1$, $L_0 \leftarrow 0$, $T(x) \leftarrow 1$,
   $i \leftarrow 0$, $v_{-1}(x) \leftarrow 1$, $\delta \leftarrow 2t$, $D_{-1} \leftarrow 1$, $\gamma \leftarrow 6$
1. **while** $i < 2t$ a $\gamma \geq t$ **do**
2. $\quad i \leftarrow i + 1$, $\gamma \leftarrow 2t + L_n - i$
3. $\quad$ spočítej koeficient stupně γ v $r_n(x) = v_n(x) \cdot S(x)$, tj.
   $\quad D_n = \sum_{j=0}^{L_n} (v_n)_j \cdot S_{2t-(\gamma-j)}$
4. $\quad$ **if** $D_n \neq 0$ **then**
5. $\quad\quad$ **if** $\gamma < \delta$ **then**
6. $\quad\quad\quad n \leftarrow n + 1$
7. $\quad\quad\quad v_n(x) \leftarrow C \cdot v_{n-2}(x) - C \cdot (D_{n-2}/D_{n-1}) \cdot x^{\delta-\gamma} \cdot v_{n-1}(x)$
8. $\quad\quad\quad T(x) \leftarrow v_{n-1}(x)/D_{n-1}$
9. $\quad\quad\quad \delta \leftarrow \gamma$, $L_n \leftarrow \deg v_n$
10. $\quad\quad$ **else**
11. $\quad\quad\quad v_n(x) \leftarrow v_n(x) - D_n \cdot T(x) \cdot x^{\gamma-\delta}$
12. $\quad\quad$ **end if**
13. $\quad$ **end if**
14. **end while**
15. **return** $\Lambda(x) = x^{\deg v_n(x)} \cdot v_n(x^{-1})$

Poznámka. Je-li $C = D_{n-1}/D_{n-2}$, pak krok 7 lze nahradit $v_n(x) = D_{n-1}T(x) - x^{\delta-\gamma} v_{n-1}(x)$.

Pokud $\gamma \geq \delta$, pak $r_n(x)$ je stupně vyššího nebo rovného $r_{n-1}(x)$, tedy nemůže se jednat o zbytek po dělení polynomem $r_{n-1}(x)$. Je tedy nutné nalézt jiný polynom $r_n(x)$, který bude stupně menšího než dělitel. Aktualizovaný odhad $v_n(x)$ je dán podle (4.20) jako

$$v_n^{(j)}(x) = C v_{n-2}(x) - q_n^{(j)}(x) \cdot v_{n-1}(x) x^{\gamma-\delta}. \tag{4.27}$$

Nicméně nám však stačí aktualizovat $v_n(x)$ jen v souvislosti s novou částí $q_n(x)$. Dosadíme-li za $q_n^{(j)}(x)$ z (4.23), pak se rovnice (4.27) zjednoduší na

$$v_n^{(j)}(x) = v_n^{(j-1)}(x) - D_n \cdot T_{n-1}(x) x^{\gamma-\delta}, \tag{4.28}$$

kde $T_{n-1}(x) = v_{n-1}(x)/D_{n-1}$. Správnost tohoto odhadu opět ověříme v dalším cyklu algoritmu.

Zvolíme-li $C = 1$, jedná se o optimalizovaný rozšířený Eukleidův algoritmus. Vylepšení se týká nahrazení operace dělení polynomů v kroku 3 Algoritmu 4 více dílčími iteracemi.

Algoritmus 6 implementuje BM algoritmus, jestliže volíme $C = D_{n-2}/D_{n-1}$ a inicializujeme $v_{-1}(x) = 1$, $v_0(x) = 1$. Tato volba konstanty C zajišťuje vlastnost algoritmu, že všechny návrhy $\Lambda^{(i)}(x)$ na lokační polynom mají vždy absolutní člen roven 1. V mezivýsledcích Algoritmu 6 však mají vycházet polynomy, které mají převrácené pořadí koeficientů oproti $\Lambda^{(i)}(x)$. Proto chceme, aby každý polynom

$v_n(x)$ byl monický. Rovnici (4.20) a krok 7 Algoritmu 6 lze pak zjednodušit na

$$v_n^{(1)}(x) = D_{n-1} \cdot T(x) - x^{\delta-\gamma} \cdot v_{n-1}(x). \tag{4.29}$$

Prohlédněme si ještě podmínku $\gamma < \delta$ v kroku 5. Nahrazením $\gamma = 2t + L_n - i$, $\delta = 2t + L_{n-1} - i_0$ a $L_n = i_0 - L_{n-1}$ dostáváme podmínku $2L_n < i$, jež je totožná s tou v 9. kroku BM algoritmu.

Celkově by tedy měly polynomy $v_n(x)$ v cyklu i Algoritmu 6 být „převrácené“ k $\Lambda^{(i)}(x)$ z Algoritmu 2. Diskrepance $D^{(n)}$ by měly být shodné s D_n (při prvním výpočtu, viz Příklad 4.4) a polynomy $T(x)$ by si měly navzájem odpovídat až na převrácené pořadí koeficientů a příslušný násobek x.

V následujícím závěrečném příkladu podrobně projdeme celý Algoritmus 6, přičemž mezi sebou porovnáme mezivýsledky tohoto algoritmu a Algoritmu 2.

Příklad 4.4. *V Příkladu 3.3 jsme zaznamenali průběh Berlekampova-Masseyova algoritmu; k řešení stejného zadání nyní použijme Algoritmus 6, kde za konstantu C zvolíme D_{n-1}/D_{n-2}. V následující tabulce znázorňující průběh modifikovaného rozšířeného Eukleidova algoritmu se zaměřme na tučně zvýrazněné mezivýsledky. (Tabulka byla vyplňována shora dolů, zleva doprava a byly do ní zapisovány hodnoty všech proměnných, jakmile byly aktualizovány.)*

$i = 0$	$\gamma = 6$	$D_{-1} = 1$	$n = 0$	$\mathbf{v_0(x) = 1}$	$\mathbf{T(x) = 1}$	$\delta = 6$	$\mathbf{L_0 = 0}$
$i = 1$	$\gamma = 5$	$D_0 = 2$	$n = 1$	$\mathbf{v_1^{(1)}(x) = 6x + 2}$	$\mathbf{T(x) = 4}$	$\delta = 5$	$\mathbf{L_1 = 1}$
$i = 2$	$\gamma = 5$	$\mathbf{D_1 = 2}$		$\mathbf{v_1^{(2)}(x) = 6x + 1}$			
$i = 3$	$\gamma = 4$	$D_1 = 4$	$n = 2$	$\mathbf{v_2^{(1)}(x) = x^2 + 6x + 2}$	$\mathbf{T(x) = 5x + 2}$	$\delta = 4$	$\mathbf{L_2 = 2}$
$i = 4$	$\gamma = 4$	$\mathbf{D_2 = 5}$		$\mathbf{v_2^{(2)}(x) = x^2 + 2x + 6}$			
$i = 5$	$\gamma = 3$	$D_2 = 0$					
$i = 6$	$\gamma = 2$	$D_2 = 0$					

Vidíme, že zvýrazněné mezivýsledky odpovídají těm z Algoritmu 2, jak jsme očekávali. Jediná odchylka se vyskytuje u polynomů $v_1^{(1)}(x)$ a $v_1^{(2)}(x)$. Je to z důvodu, že v tomto kroku ještě nejsou znormalizované. Po přenásobení inverzem vedoucího koeficientu (tj. 6^{-1}) se již „převráceným“ polynomům $v_1(x)$ a $v_2(x)$ rovnají.

Podobnost modifikovaného rozšířeného Eukleidova algoritmu se standardním Eukleidovým algoritmem by podle toho, jak jsme modifikaci prováděli, měla být zřejmá.

Důsledek 4.4. *Dekódovací algoritmy s řetězovými zlomky, Berlekampův-Masseyův a Eukleidův jsou ekvivalentní ve smyslu Kapitoly 4.*

Závěr

V této práci jsme se seznámili s Reed-Solomonovými kódy, různými způsoby jejich kódování i dekódování na dostatečné teoretické úrovni, abychom mohli navázat hlavním tématem, jímž bylo využití řetězových zlomků při dekódování těchto samoopravných kódů.

Nahlédnutím do problematiky řetězových zlomků a odkrytím malé části této obsáhlé teorie se nám podařilo dojít k odpovědi, jak využít aproximace řetězovými zlomky k řešení tzv. klíčové rovnice. Zde by šel nalezený dekódovací algoritmus zobecnit i na analogické problémy, které nalézáme v dalších oblastech samoopravných kódů (a pravděpodobně i v jiných odvětvích matematiky).

Celkem jsme popsali tři dekódovací algoritmy – Berlekampův-Masseyův, Eukleidův a již zmíněný algoritmus s řetězovými zlomky. Všechny se snaží nalézt řešení klíčové rovnice, ale každý z nich k tomuto problému přistupuje svým vlastním způsobem: Berlekampův-Masseyův skrze nalezení nejkratšího lineárního posuvného registru se zpětnou vazbou generujícího danou posloupnost, Eukleidův analogicky ke stejnojmennému algoritmu hledajícímu největšího společného dělitele a nakonec algoritmus s řetězovými zlomky prostřednictvím postupných aproximací zadané mocninné řady.

Souvislost Eukleidova algoritmu a řetězových zlomků je obecně známa. Oproti tomu spojitost řetězových zlomků s hledáním nejkratšího posuvného registru může být překvapivá. Zde jsme tento blízký vztah ukázali na konkrétním případu dekódování pomocí Eukleidova a Berlekampova-Masseyova algoritmu. Tento vztah však určitě platí i na obecnější úrovni a bylo by zajímavé jej prozkoumat podrobněji.

Literatura

[1] E. Berlekamp. Nonbinary BCH decoding. *IEEE Transactions on Information Theory*, 14(2):242, 1968.

[2] W. L. Eastman. *Euclidean Decoders for Bose-Chaudhuri-Hocquenghem Codes*. Rome Air Development Center, Air Force Systems Command, 1988.

[3] I. S. Reed a G. Solomon. Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, 8:300–304, 1960.

[4] L. Welch a R. A. Scholtz. Continued fractions and Berlekamp's algorithm. *Information Theory, IEEE Transactions on*, 25(1):19–27, 1979.

[5] J. Massey. Shift-register synthesis and BCH decoding. *Information Theory, IEEE Transactions on*, 15(1):122–127, 1969.

[6] T. D. Mateer. On the equivalence of the Berlekamp-Massey and the Euclidean algorithms for algebraic decoding. pages 139–142, 2011.

[7] W. H. Mills. Continued fractions and linear recurrences. *Mathematics of Computation*, 29(129):173–180, 1975.

[8] W. Peterson. Encoding and error-correction procedures for the Bose-Chaudhuri codes. *IEEE Transactions on Information Theory*, 6(4):459–470, 1960.

[9] D. Stanovský. *Základy algebry*. Matfyzpress, 2010.

[10] L. R. Welch. The original view of Reed-Solomon coding and the Welch-Berlekamp decoding algorithm, 1997.

[11] Y. Sugiyama, M. Kasahara, S. Hirasawa a T. Namekawa. A method for solving key equation for decoding Goppa codes. *Information and Control*, 27(1):87–99, 1975.