Technische Universiteit Eindhoven

Bachelor Eindproject: p−adische getallen en Linear Feedback Shift Registers

Begeleider
Benne de Weger
Student/Auteur
Harm Campmans, 0746865
18 juni 2015

Inhoudsopgave

1 inleiding

Voor u ligt het verslag van het Bachelor Eindproject van Harm Campmans. Het oorspronkelijk idee voor dit project was voor mij om uit te zoeken wat LFSRs en p-adische getallen nou met elkaar te maken hebben.
In dit verslag wordt uitgelegd hoe Linear Feedback Shift Registers, ofwel LFSRs, en Feedback with Carry Shift Registers, ofwel FCSRs, werken. Dit zijn rekenmodellen die een rij getallen als output geven. Deze rij kan pseudowillekeurig zijn wat voor verschillende toepassingen nuttig is. Zo kunnen deze getallen worden gebruikt bij een stroomcijfer waarbij een boodschap symbool voor symbool wordt versleuteld. Zoals vaker in de cryptografie wordt er ook in dit verslag gekeken naar de voorspelbaarheid van de verkregen reeksen. Onvoorspelbaarheid is namelijk van belang voor de veiligheid van een dergelijke vercijfering.
In de theorie over deze systemen wordt gebruik gemaakt van p-adische getallen. Omdat dit voor de lezer wellicht een onbekende soort getallen is, wordt vooraf aan de uitleg van shift registers uitgelegd wat p-adische getallen zijn in hoofdstuk 3. Deze p-adische getallen zijn anders dan de bekende reële getallen, maar zijn wel net als reële getallen een uitbreiding op de breuken. Dit wordt toegelicht in hoofdstuk 4 en kan gezien worden als verdieping op de p-adische getallen. Vervolgens wordt in de hoofdstukken 4 en 5 uitgelegd wat LFSRs en FCSRs zijn, waarna er in hoofdstuk 6 dieper wordt ingegaan op de cryptologische uitdaging die met deze systemen is gecreëerd.

2 p-adische getallen

De theorie over p-adische getallen die in dit hoofdstuk aan bod komt, komt uit het boek P-adic numbers : an introduction[1].

2.1 Hensels analogie

De p-adische getallen waren het eerste geïntroduceerd door de Duitse wiskundige K. Hensel. De motivatie hiervoor lijkt een analogie te zijn geweest. Het betrof de analogie tussen gehele getallen met de hieruit ontstane breuken en de ring $\mathbb{C}[X]$ van polynomen met complexe coefficiënten en zijn lichaam van breuken $\mathbb{C}(X)$.
Om dit te verduidelijken: $f(x)$ behoort tot $\mathbb{C}(X)$ betekent dat $f(X) = \frac{P(X)}{Q(X)}$ met
$P(X), Q(X) \in \mathbb{C}[X]$ en $Q(X)$ is niet de nulfunctie.

2.1.1 overeenkomst

unieke factorisatie van $\mathbb{Z}$ en $\mathbb{C}[X]$.
In $\mathbb{Z}$ hebben alle getallen (op 0 na) een unieke priemfactorisatie. Eventueel zit er ook nog een factor -1 in de factorisatie die niet als priemgetal wordt beschouwd.
In $\mathbb{C}[X]$ zijn elementen te factoriseren als $P(X) = a(X - a_1)(X - a_2)(X - a_3)\cdots$ waarbij de factoren $(X - a_i)$ de priemfactoren zijn. In deze factorisatie van $\mathbb{C}[X]$ geldt dat de factor a evenmin een priemfactor is als de factor -1 in $\mathbb{Z}$. Beide vormen van factorisering geven een unieke priemontbinding op de volgorde van factoren na.

2.1.2 schrijfwijze

De voorgaande vergelijking wordt doorgetrokken, maar eerst wordt een andere schrijfwijze van het polynoom geïntroduceerd. Een polynoom $P(X)$ kan worden geschreven als Taylorreeks. Hierbij wordt vooraf een $\alpha \in \mathbb{C}$ gekozen waarna men $P(X)$ kan schrijven in de vorm$P(X) = \sum_{i=0}^{n} a_i(X - \alpha)^i$.
Voor algemenere functies kunnen we ook een Laurentreeks gebruiken. Een Laurentreeks heeft de vorm $\sum_{i \geq n_0} a_i(X - \alpha)^i$ waarbij n_0 ook een negatief getal kan zijn. Maar laten we voor nu weer kijken naar de polynomen met hun Taylorreeks.
Bij een Taylorreeks gaat het om een benadering van een functie rondom het punt α. Van deze reeks kunnen de eerste n termen uitgerekend worden met behulp van afgeleiden en het invullen van α. Aan de hand van de uitkomsten kunnen we een ne graads polynoom samenstellen die functioneert als benadering van de originele functie. Hoe hoger n is hoe beter de benadering zou moeten zijn, en voor polynomen is dit ook zeker het geval.

2.1.3 doortrekken analogie

We gaan deze schrijfwijze nu gebruiken met een $(X - \alpha)$ die een door ons gekozen priemfactor is. Als $(X - \alpha)$ een priemfactor is van $\mathbb{C}[X]$ dan kunnen we uit deze notatiewijze informatie verkrijgen over de relatie tussen de originele functie en de priemfactor $(X - \alpha)$. We gaan nu een soortgelijke notatie bij $\mathbb{Z}$ introduceren waarbij we het getal $m \in \mathbb{Z}$ proberen te schrijven als $m = \sum_{i \geq 0} a_i p^i = a_0 + a_1 p + a_2 p^2 + ...$ Om van deze reeks termen te berekenen bekijken we het probleem modulo p^k waarbij we beginnen met $k = 1$, daarna $k = 2$ en vervolgens k steeds 1 groter maken. Op deze manier krijgen we "lokale informatie van m rond p".

Zo geeft a_0 alleen maar informatie over $m \mod p^1$, waarna a_1 een beetje informatie toevoegt zodat we nu samen met a_0 de waarde van $m \mod p^2$ weten. Zo voegt iedere coëfficiënt a_i een beetje informatie toe aan het tot dan toe bekende geheel maar a_i betekent op zichzelf niet zo veel. Dit is net als Taylorpolynomen. De zevende term van een Taylorpolynoom voegt wat toe aan het geheel, maar dicht bij $X = \alpha$ is de eerste term het belangrijkst.
Voor lezers die vooral gewend zijn om met reële getallen te werken kan het systeem van p-adische getallen wat vreemde dingen met zich mee brengen:
Als we voor $m \in \mathbb{Z}$ in het tientallig stelsel een globale indicatie willen geven van de waarde van het getal zullen we kijken naar de meest linker coëfficiënt van de decimale schrijfwijze. Zo zou het getal 706 "ongeveer 700 "kunnen zijn. Bij p-adische getallen zijn we echter het meest geïnteresseerd in zijn waarde $\mod p$. Dus als $m = 706$ en $p = 7$ dan is m "ongeveer 6 $\mod 7$ "een logischere uitspraak. Zo is 700 wel groter dan 6 maar voor de 7-adische schrijfwijze is 6 belangrijker dan 700. De lezer moet dus het idee dat "groter belangrijker is "loslaten, de belangrijkste coëfficiënten hebben binnen m immers de kleinste reële waarde. Ook zijn p-adische getallen niet te ordenen wat u wellicht wel gewend bent vanuit de reële getallen. Om dit beter te begrijpen kunt u wellicht meer lezen over de p-adische norm in het volgende hoofdstuk.

2.2 recursieve functies

Het oplossen van een vergelijking als $X = 1 + 3X$ is geen probleem van dergelijke moeilijkheid dat we extra trucjes uit de kast hoeven te halen in de gebruikelijke wiskunde (met de operatie delen). Nu we het delen van p-adische getallen nog niet verkend hebben is het een mooi moment om te kijken of we dit probleem kunnen oplossen met p-adische getallen zonder deze operatie.
We gaan kijken wat de volgende recursieve formule oplevert:

$$x_0 = 1$$
$$x_{n+1} = 1 + 3 \cdot x_n$$

De formule zal een rij opleveren met de volgende waardes als elementen:

$$x_n = \sum_{i=0}^{n} 3^i$$

Laten we nu even aannemen dat deze reeks een p-adisch getal representeert.
Aansluitend op de vorige paragraaf gaan we nu een intuïtieve verklaring geven:
Beschouw

$$Y = \sum_{i=0}^{\infty} x^i$$

Bij de reële getallen convergeert bovenstaande reeks binnen een bepaalde convergentiestraal rond het punt 0, zodat geldt:

$$Y = \sum_{i=0}^{\infty} x^i = \frac{1}{1-x} \forall x \in (-1, 1)$$

Bij reële getallen geldt dat $|x^i|_{\mathbb{R}}$ voldoende hard naar 0 gaat als $i \to \infty$ zodat $|\sum_{i=0}^{\infty} x^i|_{\mathbb{R}}$ convergeert zolang $|x|_{\mathbb{R}} < 1$. De norm van p-adische getallen hebben we tot op heden nog niet gedefiniëerd, maar we hebben al wel vermeld dat getallen die in de reële getallen groot zijn, hier van kleine significantie kunnen zijn. Zo zal $|x^i|_p$ alsnog hard naar 0 kunnen gaan terwijl $|x|_{\mathbb{R}} > 1$ is. Neem nou $x = 3$ en $p = 3$, we zullen dan zien dat hoewel $|x|_{\mathbb{R}} > 1$ dat er voor de 3-adische norm geldt dat $|x|_3 < 1$. Hierdoor gaat $|x^i|_3$ hard genoeg naar 0 als $i \to \infty$ zodat $|\sum_{i=0}^{\infty} x^i|_3$ convergeert voor de waarde $x = 3$, een waarde die in de reële getallen niet zou voldoen aan $\sum_{i=0}^{\infty} x^i = \frac{1}{1-x}$. Hierdoor convergeert onze recursieve formule naar de oplossing van $X = 1 + 3X$:

$$x_n = \sum_{i=0}^{n} 3^i \to \sum_{i=0}^{\infty} 3^i = \frac{1}{1-3} = \frac{1}{-2}$$

2.2.1 breuken

De volgende stap voor deze analogie is het invoeren van de operatie delen waardoor breuken aan het licht komen. Met de geïntroduceerde schrijfwijze weten we dat we de getallen uit $\mathbb{Z}$ kunnen schrijven op p-adische wijze. De vraag ontstaat nu of we al deze p-adische gehele getallen door elkaar kunnen delen met een p-adische uitkomst (met de uitzondering van delen door 0). Het introduceren van delen is niet veel anders dan een methode geven hoe je vermenigvuldigen ongedaan kunt maken.
Een breuk van gehele getallen ($b \neq 0$) ziet er uit als

$$q = \frac{a}{b} \Leftrightarrow q \cdot b = a$$

Als a en/of b deelbaar zijn/is door p dan lossen we het probleem $q^* = \frac{a^*}{b^*}$ op waarbij alle factoren p uit a en b zijn gedeeld. Hierbij worden respectievelijk a^* en b^* verkregen. Als we dit probleem kunnen oplossen weten we dat $q = q^* \cdot p^z$ voor een $z \in \mathbb{Z}$ en dat q dus bestaat.
Vanaf hier kunnen we er dus vanuit gaan dat a en b relatief priem zijn ten opzichte van p. De vraag is dus of we een q kunnen construeren met p-adische schrijfwijze die aan bovenstaande eis voldoet. We zijn op zoek naar $q = q_0 \cdot p^0 + q_1 \cdot p^1 + q_2 \cdot p^2 + \ldots$ die moet voldoen aan $q \cdot b = a$.
Als er een oplossing voor deze vergelijking is dan bestaat deze oplossing ook $\mod p^i$ voor iedere positieve gehele i. We weten dat a en b beide niet deelbaar zijn door p zodat we de eerste coëfficiënt van q als volgt vinden:

$$q_0 = a_0 \cdot (b_0)^{-1} \mod p$$

De inverse van $b_0 \in \{0, 1, \ldots, p-1\}$ bestaat altijd als p priem is. Nadat het eerste beginnetje is gemaakt kan het volgende proces worden herhaald:
$\hat{q}_i$ is het bekende deel van q na i iteraties, ofwel $\hat{q}_i = \sum_{j=0}^{j-1} q_j \cdot p^j$.
En we weten tot nu toe dat $\hat{q}_i \cdot b \equiv a \mod p^i$ geldt en dat $\hat{q}_{i+1} \cdot b \equiv a \mod p^{i+1}$ zal moeten gelden. Hiervoor moeten we alleen nog q_i berekenen.

$$\begin{aligned} \hat{q}_{i+1} \cdot b &\equiv a \mod p^{i+1} \\ q_i \cdot p^i \cdot b &\equiv a - \hat{q}_i \cdot b \mod p^{i+1} \end{aligned}$$

we weten uit $\hat{q}_i \cdot b \equiv a \mod p^i$ dat zowel de linker als de rechterkant deelbaar zijn door p^i

$$\begin{aligned} q_i \cdot b &\equiv \frac{a - \hat{q}_i \cdot b}{p^i} \mod p \\ q_i &\equiv \frac{a - \hat{q}_i \cdot b}{p^i} \cdot (b_0)^{-1} \mod p \end{aligned}$$

zodat we nu ook q_i weten en $\hat{q}_{i+1} = \hat{q}_i + q_i \cdot p^i$ zodat we genoeg weten om aan de volgende iteratie te beginnen.
Aangezien deze berekeningen uit te voeren zijn voor alle gehele a en $b \neq 0$ bewijst dit dat een dergelijke q altijd gevonden kan worden. Eventueel moeten we q nog uitrekenen aan de hand van $q = q^* \cdot p^z$, waarbij $z = n_a - n_b$ zodanig dat $a = p^{n_a} \cdot a^*$ en $b = p^{n_b} \cdot b^*$ met a^*, b^* geheel en ondeelbaar door p. Hierbij krijgen we $q = q_{n_0} \cdot p^{n_0} + q_{n_0+1} \cdot p^{n_0+1} + \ldots$ met $z = n_0$.

Het bestaan van deze rekenmethode is het bewijs dat we van iedere breuk een p-adische expansie kunnen maken. $\mathbb{Q}$ is dus een deelverzameling van $\mathbb{Q}_p$, waarbij

$$\mathbb{Q}_p = \{ \sum_{i=n_0}^{\infty} a_i \cdot p^i | a_i \in \{0, 1, \ldots, p-1\}, n_0 \in \mathbb{Z} \}$$

Verderop zullen we aantonen dat er ook elementen in $\mathbb{Q}_p$ zitten die niet in $\mathbb{Q}$ zitten.

Bij periodieke rijen kan onderscheid gemaakt worden tussen twee soorten uiteindelijk periodiek en strikt periodiek. Een rij $(x_n) = (x_0, x_1, \ldots)$ is periodiek met periode T als $x_n = x_{n+T} \forall n \geq N$. De rij is uiteindelijk periodiek als $N > 0$ en strikt periodiek als $N = 0$. Je zou kunnen zeggen dat een strikt periodieke rij uiteindelijk periodiek is, maar andersom hoeft dat niet het geval te zijn. Bij het delen verkrijgen we altijd een rij coëfficiënten die uiteindelijk periodiek is. Merk hierbij op dat als alle coëfficiënten 0 zijn, dat je dit ook als periodiek kunt beschouwen. Niet alle p-adische getallen hebben een uiteindelijk periodieke rij van coëfficiënten maar de getallen die een breuk representeren wel.

analogie

Ook voor $\mathbb{C}[X]$ geldt dat de breuk van twee eindige polynomen niet per se een eindige rij coëfficiënten oplevert. Dit kan voor sommige vraagstukken onhandig uitpakken, maar voor "lokale informatie"rond α kan een eindige Laurentreeks benadering nog steeds nuttige informatie geven. Voor de breuken van polynomen geldt ook dat de uitkomst een periodieke rij van coëfficiënten heeft. Bij onze uitkomst $f(X) = \frac{P(X)}{Q(X)} = \sum_{i \geq n_0} a_i (X - \alpha)^i$ kunnen we een benadering krijgen met behulp van een staartdeling.

Als we met onze nieuwe notatiewijze breuken gaan maken in $\mathbb{Z}$ krijgen we een p-adisch getal van de vorm

$$q = \frac{a}{b} = \frac{a_0 + a_1 \cdot p + a_2 \cdot p^2 + \cdots}{b_0 + b_1 \cdot p + b_2 \cdot p^2 + \cdots} = c_{n_0} \cdot p^{n_0} + c_{n_0+1} \cdot p^{n_0+1} + \cdots = \sum_{i=n_0}^{\infty} c_i p^i$$

.

Voor de analogie is het goed om te beseffen dat $X - \alpha$ en p beide priemfactoren zijn in hun eigen ruimte. Als je in een Taylorreeks $X - \alpha$ vervangt door p krijg je een p-adisch getal $m = a_0 + a_1 p + a_2 p^2 + ... + a_n p^n$. Beide expansies geven lokale informatie: Als m deelbaar is door p dan zie je dit makkelijk terug in a_0, als hij deelbaar is door p^2 in a_1 (als $a_0 = 0$) etc.

Voor het polynoom P zegt de Taylorreeks rond α veel over het gedrag van de functie rond α. In de analogie die wij trekken zegt de expansie van een getal iets over dat getal "rond p ". Zo kan a_0 ons vertellen of m deelbaar is door p, waarna a_1 ons kan vertellen of hij ook deelbaar is door p^2.

Voorbeeld

Als we nu een 3-adische (p-adisch met $p = 3$) breuk als $\frac{17}{24}$ berekenen, dan krijgen we
$\frac{17}{24} = 1 \cdot p^{-1} + 0 \cdot p^0 + 2 \cdot p^1 + 2 \cdot p^2 + 1 \cdot p^3 + 2 \cdot p^4 + 1 \cdot p^5 + 2 \cdot p^6 + ...$
In dit voorbeeld zie je aan de $a_{-1} \neq 0$ dat de noemer een factor $p = 3$ bevat. Ook is het een mooi voorbeeld van een oneindige reeks coëfficiënten die uiteindelijk periodiek is. Je kunt narekenen dat deze uitkomst vermenigvuldigd met $24 = 2p + 2p^2$ weer $17 = 2 + 2p + p^2$ terug geeft.

2.3 overige getallen in $\mathbb{Q}_p$

p-adische getallen kunnen net als reële getallen ook wortels zijn van polynomen. Bij het oplossen van deze vergelijkingen hebben we vaak niet genoeg aan de bovengenoemde operaties, en gaan we de coëfficiënten stuk voor stuk oplossen.

2.3.1 oplossingen van polynomen

Bij een vergelijking als $X^3 \equiv 17 \mod p^n$ is er een oplossing in $\mathbb{Q}_p$ voor sommige priemgetallen p ($p = 17$ bijvoorbeeld niet). Als de oplossing $X = x_0 \cdot p^0 + x_1 \cdot p^1 + x_2 \cdot p^2 + \ldots$ bestaat is deze te vinden door eerst het stelsel $\mod p^i$ te bekijken waarbij je i steeds groot genoeg neemt om de volgende coëfficiënt te kunnen vinden. Hieronder zie je twee voorbeelden van het vinden van van een oplossing van $X^3 = 17$.

2-adisch	5-adisch
$x_0^3 \equiv 17 \mod p$	$x_0^3 \equiv 17 \mod p$
$x_0 \equiv 1 \mod p$	$x_0 \equiv 3 \mod p$
$(1 + x_1 \cdot p)^3 \equiv 17 \mod p^2$	$(3 + x_1 p)^3 \equiv 17 \mod p^2$
$1 + 3x_1 \cdot p \equiv 17 \mod p^2$	$3^3 + 3^2 \cdot 3 \cdot x_1 p \equiv 17 \mod p^2$
$x_1 \equiv 0 \mod p$	$x_1 \equiv 4 \mod p$
$(1 + x_2 \cdot p^2)^3 \equiv 17 \mod p^3$	$(23 + x_2 p^2)^3 \equiv 17 \mod p^3$
$1 + 3x_2 p^2 \equiv 17 \mod p^3$	$23^3 + 23^2 \cdot 3 \cdot x_2 p^2 \equiv 17 \mod p^3$
$x_2 \equiv 0 \mod 2$	$x_2 \equiv 2 \mod p$
$(1 + x_3 p^3)^3 \equiv 17 \mod p^4$	$(73 + x_3 p^3)^3 \equiv \mod p^4$
$1 + 3x_3 p^3 \equiv 17 \mod p^4$	$73^3 + 73^2 \cdot 3x_3 p^3 \equiv 17 \mod p^4$
$x_3 \equiv 0 \mod p$	$x_3 \equiv 4 \mod p$
$(1 + x_4 p^4)^3 \equiv 17 \mod p^5$	$(573 + x_4 p^4)^3 \equiv 17 \mod p^5$
$1 + 3x_4 p^4 \equiv 17 \mod p^5$	$573^3 + 573^2 \cdot 3x_4 p^4 \equiv 17 \mod p^5$
$x_4 \equiv 1 \mod p$	$x_4 \equiv 4 \mod p$
$(82 + x_5 p^5)^3 \equiv 17 \mod p^6$	$(3073 + x_5 p^5)^3 \equiv 17 \mod p^6$

In de bovenstaande voorbeelden kun je zien dat er steeds een tussenresultaat is van de vorm $\alpha_n = \sum_{i=0}^{n} x_i p^n$. In de voorbeelden zijn dat $(1, 1, 1, 1, 17, \ldots)$ en $(3, 23, 73, 573, 3073, \ldots)$. Deze rijen zijn coherent. De definitie van een coherente rij is dat deze voldoet aan $\alpha_{n+1} \equiv \alpha_n \mod p^n$. Als een oplossing bestaat, gaat deze coherente rij oneindig door (zelfs voor $X^2 = 1$, ook al is het een simpele rij).

Praktische tip bij de controle van de coherentie: 573^3 intikken in een rekenmachine kan leiden tot ongewenste afrondingen. Je kunt 573^2 eerst vereenvoudigen modulo $5^4 = 625$ waarna je het nog eens met 573 vermenigvuldigt om tot $204 * 573 \equiv 573^3 \equiv 17 \mod 625$ te komen.

conclusie uit voorbeeld
Of er oplossingen bestaan voor ieder polynoom, en hoeveel dit er zijn, is voor dit verslag niet heel relevant. De conclusie is hier dat er getallen(oplossingen van vergelijkingen) bestaan in de ruimtes $\mathbb{Q}_p$ die niet in $\mathbb{Q}$ zitten. Merk hierbij op dat de ruimte $\mathbb{Q}_p$ voor iedere p verschillend is.
Merk op dat oplossingen van polynomen ook buiten $\mathbb{R}$ kunnen liggen en dat ook de ruimtes $\mathbb{Q}_p$ onderling verschillend zijn. We zijn in dit verslag niet geïnteresseerd in $\mathbb{R}$ en het wordt alleen gebruikt om tot de verbeelding te spreken.

3 wiskundige basis voor p-adische getallen

De theorie over p-adische getallen die in dit hoofdstuk aan bod komt, komt uit het boek P-adic numbers : an introduction[1].

3.1 p-adische absolute waarde

Nu we in eerdere hoofdstukken hebben gezien dat $\mathbb{Q}_p$ meer bevat dan alleen $\mathbb{Q}$ gaan we in dit hoofdstuk aantonen dat $\mathbb{Q}_p$ een completering is van $\mathbb{Q}$. Hiertoe definiëren we een nieuwe absolute waarde functie die aansluit op de eerder getoonde eigenschappen van de p-adische getallen.

Formeel gezien moet een absolute waarde functie $|\ |: k \to \mathbb{R}$ voldoen aan de volgende eisen:

1. $|x| = 0$ dan en slechts dan als $x = 0$
2. $|xy| = |x| \cdot |y|$ voor alle $x, y \in k$
3. $|x + y| \leq |x| + |y|$ voor alle $x, y \in k$
   En voor een niet-archimedische absolute waarde geldt ook de volgende voorwaarde:
4. $|x + y| \leq \max\{|x|, |y|\}$ voor alle $x, y \in k$

Merk op dat als (4) geldt, dat daaruit (3) volgt.

Bij het introduceren van niet-archimedische absolute waarden kan gebruikt gemaakt worden van een valuatie welke voldoet aan de volgende eisen:

1. $v(a) = +\infty \Leftrightarrow a = 0$
2. $v(x \cdot y) = v(x) + v(y)$
3. $v(x + y) \geq \min\{v(x), v(y)\}$

Ook de p-adische norm zullen we opbouwen vanuit een valuatie. De p-adische valuatie is als volgt gedefinieerd: $v_p(n)$ is het unieke positieve gehele getal dat voldoet aan

$$n = p^{v_p(n)} \cdot n' \ with\ p \nmid n' \in \mathbb{Z}$$

Omdat we deze functie ook op breuken willen toepassen breiden we hem uit volgens de regel

$$v_p(\frac{a}{b}) = v_p(a) - v_p(b)$$

De uitkomst hiervan is hetzelfde voor alle paren breuken met $a_1, a_2, b_1, b_2 \in \mathbb{Z}$ waarvoor geldt dat $\frac{a_1}{b_1} = \frac{a_2}{b_2}$ want uit

$$\begin{aligned} a &= c \cdot ggd(a, b) \\ b &= d \cdot ggd(a, b) \\ a, b, c, d &\in \mathbb{Z} \end{aligned}$$

volgt

$$v_p(\frac{a}{b}) = v_p(c \cdot ggd(a, b)) - v_p(d \cdot ggd(a, b) = v_p(c) - v_p(d) + v_p(ggd(a, b)) - v_p(ggd(a, b)) = v_p(c) - v_p(d)$$

Met deze valuatie definiëren we de p-adische absolute waarde

$$|x|_p = p^{-v_p(x)}$$

waarbij $|0|_p = p^{-\infty} = 0$. Andere grondtallen, zoals e zouden ook een geldige norm opleveren, maar in het vervolg van dit verslag kiezen we voor grondtal $|\mathbb{F}_p| = p$.

Om gevoel te krijgen voor bovenstaande formules volgt hier een voorbeeld:
Bij v_7 telt de valuatie het aantal keer dat het priemgetal 7 in n zit. Zo is $v_7(7) = 1$ en $|7|_7 = \frac{1}{7}$ terwijl een veel groter getal zoals 11^3 een valuatie van $v_7(11^3) = 0$ heeft en $|11^3|_7 = 1$. Bij deze absolute waarde is de uitkomst dus slechts afhankelijk van het aantal 7's in zijn priemfactorisatie. De absolute waarde geeft een waarde afhankelijk van de meest lokale informatie: De uitkomst van de absolute waarde hangt slechts af van de locatie van de het eerste niet-nul element in de p-adische expansie van het getal.

3.2 Cauchyrijen

Bij de eerder gedefinieerde norm komt ook een nieuw beeld van een Cauchyrij om de hoek kijken. De definitie van een Cauchyrij (x_n) is:

$$\forall \epsilon > 0 \; \exists \, N : \forall m, n > N : |x_m - x_n| < \epsilon$$

Bij onze norm betekent dit dat $|x_m - x_n|_p \leq p^c < \epsilon$ waarbij $c = \lfloor \log_p(\epsilon) \rfloor$ omdat onze norm alleen deze waarden kan aannemen. Ieder element uit de Cauchyrij wordt gerepresenteerd als een coherente reeks in de p-adische schrijfwijze. Een "toenemend beginstuk"van deze coherente reeks zal steeds hetzelfde zijn. Er is een x_{N_1} vanaf waar het eerste element van de coherente reeks niet meer verandert (neem $\epsilon = p^{-1}$), een x_{N_2} vanaf waar de eerste twee elementen niet meer veranderen ($\epsilon = p^{-2}$) etcetera.

Verder is het het vermelden waard dat bij sommige (=Archimedische) normen de eis $|x_{n+1} - x_n| \to 0$ geen bewijs geeft dat de rij (x_n) een Cauchyrij is. Bij niet-Archimedische normen geldt echter dat $|x_{n+1} - x_n| \to 0$ dan en slechts dan als (x_n) een Cauchyrij is.

3.3 vinden van completering van $\mathbb{Q}$

Nu willen we deze rijtjes gebruiken om een completering van $\mathbb{Q}$ te creëren.
Een completering moet aan drie voorwaarden voldoen:

1. De norm $|\cdot|_p$ moeten we kunnen gebruiken in de verkregen ruimte.
2. De nieuwe ruimte moet compleet zijn, dat wil zeggen dat alle Cauchyrijen in deze ruimte een limiet hebben die in deze ruimte ligt.
3. $\mathbb{Q}$ moet dicht in de nieuwe ruimte $S \supset \mathbb{Q}$ liggen. Dat wil zeggen dat $B(x, \epsilon) \cap \mathbb{Q} \neq \emptyset \forall x \in S \forall \epsilon > 0$

Om deze ruimte te verkrijgen gaan we de ruimte nemen van alle cauchyrijen.

$$\mathcal{C} = \{(x_n) | \forall \epsilon > 0 \; \exists \, N : \forall m, n > N : |x_m - x_n| < \epsilon\}$$

Waarbij de operaties

$$(x_n) + (y_n) = (x_n + y_n)$$
$$(x_n) \cdot (y_n) = (x_n \cdot y_n)$$

laten zien dat we te maken hebben met een commutatieve ring (maar geen lichaam, want er zijn oninverteerbare elementen ongelijk aan 0).

Vervolgens bekijken we een ideaal van $\mathcal{C}$:

$$\mathcal{N} = \{(x_n) | \lim_{n\to\infty} |x_n|_p = 0\}$$

Voor dit ideaal geldt dat het een maximaal ideaal is.
Als we er namelijk vanuit gaan dat dit niet zo is, zouden we een ideaal moeten kunnen maken dat groter is dan $\mathcal{N}$.
Dit ideaal zou dan dus een element (x_n) bevatten waarvoor geldt dat $x_n \nrightarrow 0$. We kunnen nu bewijzen dat dit ideaal heel $\mathcal{C}$ is. We weten uit $(x_n) \nrightarrow 0$ dat $\exists c, N : \forall n > N : |x_n| > c > 0$. Aan de hand van deze N kunnen we een rij (y_n) definieren:

$$y_i = \begin{cases} 0 & n < N \\ 1/x_n & n \geq N \end{cases}$$

Deze (y_n) is ook Cauchy omdat $|y_{n+1} - y_n| = |\frac{1}{x_{n+1}} - \frac{1}{x_n}| = \frac{|x_{n+1} - x_n|}{|x_n \cdot x_{n+1}|} \leq \frac{|x_{n+1} - x_n|}{c^2} \to 0$ voor diezelfde c en N.
Nu geldt dat $(1) - (x_n) \cdot (y_n) \in \mathcal{N}$ waaruit we kunnen concluderen dat (1) in ieder ideaal zit wat groter is dan $\mathcal{N}$. Maar een ideaal met (1) erin is gelijk aan $\mathcal{C}$ zodat het enige ideaal waar $\mathcal{N}$ een echte deelverzameling van is, de ring $\mathcal{C}$ zelf is.

Nu gaan we $\mathbb{Q}_p$ definiëren vanuit $\mathcal{C}$. We verdelen de elementen van $\mathcal{C}$ in equivalentieklassen uit $\mathcal{C}/\mathcal{N}$. Naast stellingen over maximale idealen kunnen we ook zien hoe deze equivalentieklassen eruit zien aan de hand van de bijbehorende equivalentierelatie.

$$(x_n) \sim (y_n) \Leftrightarrow \lim_{n\to\infty} |x_n - y_n|_p \to 0$$

Om aan te tonen dat dit echt een equivalentierelatie is zullen we de eigenschappen van een equivalentierelatie doorlopen:

1. (reflexiviteit): $(x_n) \sim (x_n)$ geldt omdat $\forall n > 0 : |x_n - x_n|_p = 0 \to 0$
2. (symmetrie): $(x_n) \sim (y_n) \Leftrightarrow (y_n) \sim (x_n)$ geldt omdat $\lim_{n\to\infty} |x_n - y_n|_p = 0 \Leftrightarrow \lim_{n\to\infty} |y_n - x_n| = 0$
3. (transitiviteit): $(x_n) \sim (y_n), (y_n) \sim (z_n) \Rightarrow (x_n) \sim (z_n)$
   Bij deze eigenschap kiezen we een $\epsilon > 0$ en maken we gebruik van de volgende gegevens:

$$\epsilon_1 = \frac{1}{2}\epsilon : \exists N_1 > 0 : \forall n > N_1 : |x_n - y_n| < \epsilon_1$$
$$\epsilon_2 = \frac{1}{2}\epsilon : \exists N_2 > 0 : \forall n > N_2 : |y_n - z_n| < \epsilon_2$$

   zodat we kunnen concluderen dat:

$$\forall \epsilon > 0 \exists N = \max(N_1, N_2) : \forall n > N : |x_n - z_n| \leq |x_n - y_n| + |y_n - z_n| < \frac{1}{2}\epsilon + \frac{1}{2}\epsilon = \epsilon$$

   wat betekent dat $(x_n) \sim (z_n)$.

Door middel van deze equivalentierelatie worden equivalentieklassen gedefinieerd.
Als gegeven is dat (y_n) in een equivalentieklasse zit is de klasse de volgende set:
$\{(x_n) | (x_n) \sim (y_n)\}$
De verzameling $\mathbb{Q}_p$ is isomorf aan deze equivalentieklassen waardoor deze klassen een geschikte manier bieden om $\mathbb{Q}_p$ te definiëren.
Hiertoe moet er nog een manier gevonden worden om de equivalentieklassen te representeren.

3.4 vorm van completering

Nu we weten hoe elementen uit onze completering isomorf moeten zijn aan equivalentieklassen van cauchyrijen komen de volgende constateringen van pas:
Van een Cauchyrij $x = (x_i)$ weten we dat $\forall \epsilon > 0 \; \exists \, N : \forall m, n > N : |x_m - x_n| < \epsilon$ en we kunnen hiervoor dus $\epsilon = \frac{1}{p^j}$ invullen. We vinden dan een N_j zodat $|x_m - x_n| < \frac{1}{p^j} \forall m, n > N_j$.
In het geval van p-adische getallen betekent dit dat de eerste j elementen van de p-adische ontwikkeling gelijk zijn voor alle x_n met $n > N_j$. De Cauchyrij bestaat uit een rij van p-adische getallen met hun ontwikkeling. Deze ontwikkelingen zijn niet per se gelijk. Maar als we dus ver genoeg in deze rij kijken zijn de eerste j elementen van deze ontwikkeling gelijk. Dus hoe verder we kijken in de Cauchyrij, hoe groter het stuk is van deze ontwikkeling dat niet meer zal veranderen. Op deze manier kunnen we een oneindig lange ontwikkeling verkrijgen die een p-adisch getal representeert die we a noemen.
De Cauchyrij $(a, a, a, a, \ldots)$ zit in de equivalentieklasse van x, zodat iedere Cauchyrij x gerepresenteerd wordt door een element $a \in \mathbb{Q}_p$
Zo is het in te zien dat de ruimte $\mathbb{Q}_p$ de completering is van $\mathbb{Q}$.
Daarmee is de vorm van de completering de verzameling van alle p-adische getallen. Iedere denkbare p-adische ontwikkeling zit dus in de verdichting van $\mathbb{Q}$. We kunnen immers een cauchyrij van breuken construeren die er steeds dichterbij komt.
De vorm is dus hetzelfde als eerder aangenomen:

$$\mathbb{Q}_p = \{ \sum_{i=n_0}^{\infty} a_i \cdot p^i | a_i \in \{0, 1, \ldots, p-1\}, n_0 \in \mathbb{Z} \}$$

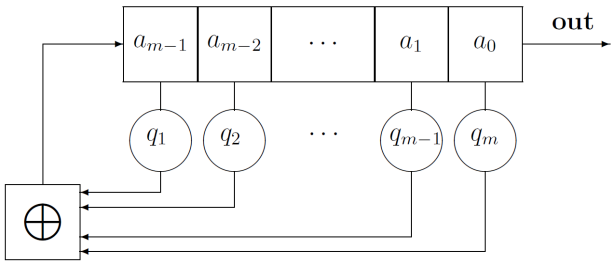
4 Inleiding LFSR

De theorie over LFSRs die in dit hoofdstuk aan bod komt, komt uit het boek Algebraic Shift Register Sequences[2].

4.1 Wat is een LFSR?

Linear Feedback Shift Registers, ofwel LFSRs, zijn virtuele apparaten die een rijtje van m getallen / elementen bevatten. Deze getallen/elementen worden opgeslagen in een geheugen, en de combinatie van de geheugencellen waar ze in zitten wordt een register genoemd. Je kunt een LFSR visualiseren met onderstaand plaatje:

In dit plaatje zijn a_i de getallen die worden onthouden en zijn de q_i de parameters voor het



Figuur 1: Een LFSR in zijn begintoestand

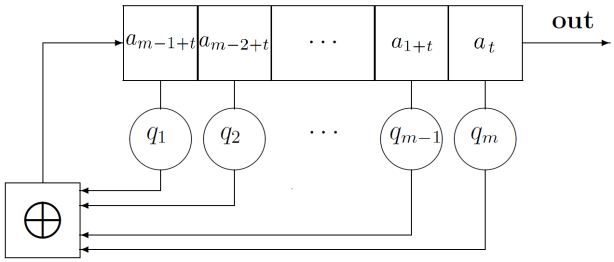
apparaat. Dit systeem kan naast getallen a_i onthouden ook een stap zetten. Bij deze stap, ook wel een klokslag genoemd, wordt een nieuw getal in de rij a_i berekend die een plek in het register zal innemen. De formule voor de eerste klokslag is:

$$a_m := \sum_{i=1}^{m-1} q_i \cdot a_{m-i}$$

Op deze manier rekenen we een nieuw getal voor de oneindige rij a_i uit met de rekenwijze die is vastgelegd door de q_i's. Bij een klokslag zullen alle getallen in het register een plekje opschuiven, daarom heet het een shift register. Zo zal in het LFSR bij de eerste klokslag a_m berekend worden die op de plek van a_{m-1} zal komen. En a_{m-1} zal op de plek komen waar a_{m-2} eerst stond. Zo schuiven alle getallen een plek op, behalve a_0 die uit het systeem zal komen als output.
Zo krijgen we na t klokslagen met de algemenere formule

$$a_k := \sum_{i=1}^{m-1} q_i \cdot a_{k-i}$$

de volgende situatie:



Figuur 2: Een LFSR na t klokslagen

In deze formule legt de set q_i's samen met de begintoestand$(a_0, a_1, \ldots, a_{m-1})$ vast wat de uiteindelijke oneindige rij (a_i) gaat worden. Deze rij kan onbegrensd grote waarden aannemen met de tot nu toe gegeven restricties. Echter rekent een LFSR $\mod n$, dit betekent dat de waarden in de rij geen oneindige grote waarden meer aan kunnen nemen, en de rij uiteindelijk periodiek zal zijn. Met eindig veel elementen in je LFSR en eindig veel mogelijkheden , namelijk n, zijn er immers maar eindig veel toestanden waarin de LFSR zich kan bevinden. Bij een oneindige rij kan een LFSR dus niet steeds een niet-eerder-voorgekomen toestand aannemen, omdat er daar maar eindig veel van zijn. Ook produceert een LFSR gegeven dezelfde set q_i's en dezelfde toestand altijd dezelfde output en hiermee dus ook dezelfde volgende toestand. Hierdoor zal hij vanuit een eerder-voorgekomen toestand ook weer verder gaan zoals hij dat eerder deed; er ontstaat een periodieke output. De rij hoeft niet te beginnen met getallen die herhaald zullen worden, maar dat zal wel het geval zijn onder bepaalde voorwaarden, waar we later op terugkomen.
Naast $\mod n$ rekenen, zal dit systeem werken voor iedere eindige commutatieve ring R.

4.2 toepassingen

LFSRs worden zowel gebruikt voor error-correctie, versleuteling in de cryptografie, radar ranging en mijn favoriet: de pseudotoevalsgeneratoren. Voor pseudotoevalsgeneratoren wil je liever geen periodieke output hebben. Maar aangezien dat blijkbaar onvermijdelijk is, heb je het liefst een zo groot mogelijke periode met zo min mogelijk operaties per rekenstap. De verkregen rij zal verder ook zo min mogelijk afhankelijkheden/patronen moeten vertonen om voorspelbaarheid van de volgende getallen te voorkomen.

4.3 voorbeeld

Als voorbeeld van een LFSR nemen we de volgende parameters: $(q_1, q_2, q_3) = (3, 2, 1)$ en (a_0, a_1, a_2) is $(3, 2, 1)$.
Zo verkrijgen we de volgende toestanden aan de hand van de volgende berekeningen:

$$a_3 = q_1 \cdot a_2 + q_2 \cdot a_1 + q_3 \cdot a_0 = 3 \cdot 1 + 2 \cdot 2 + 1 \cdot 3 = 10 \equiv 0 \mod 5$$

Het systeem geeft $a_0 = 3$ als output en heeft vervolgens toestand:

$$(a_1, a_2, a_3) = (2, 1, 0)$$
$$a_4 = q_1 \cdot a_3 + q_2 \cdot a_2 + q_3 \cdot a_1 = 3 \cdot 0 + 2 \cdot 1 + 3 \cdot 2 = 4 \equiv 4 \mod 5$$

Nu komt output $a_1 = 2$ uit het syteem

$$\begin{aligned}(a_2, a_3, a_4) &= (1, 0, 4)\\ a_5 = 3 \cdot 4 + 2 \cdot 0 + 1 \cdot 1 = 13 &\equiv 3 \mod 5\\ (a_3, a_4, a_5) &= (0, 4, 3)\\ (a_4, a_5, a_6) &= (4, 3, 2)\\ (a_5, a_6, a_7) &= (3, 2, 1)\end{aligned}$$

waarna we een toestand hebben bereikt die we eerder hebben bereikt. Dit betekent dat het systeem vanaf hier in herhaling zal gaan vallen met een periode van 5.

4.4 notatie

Door de werkwijze van de LFSRs geldt de volgende vergelijking voor alle $n \geq m$:

$$0 = q_0 \cdot a_n + q_1 \cdot a_{n-1} + q_2 \cdot a_{n-2} + \ldots + q_m \cdot a_{n-m} \tag{1}$$

Als aan deze eigenschap wordt voldaan noemen we de rij (a_i) lineair recurrent voor de rij van coëfficiënten q. Iedere output van een LFSR is dus een recurrente rij. Hier is $q_0 = -1$, welke geen deel uitmaakt van de LFSR zoals eerder beschreven.
We kunnen de operatie van een LFSR opschrijven als vector. Deze operator zou werken met een inputvector s (van het Engelse 'state'), die de toestand representeert. Deze vector zou bestaan uit de a_i'tjes die we eerder zagen.
$s = (a_0, a_1, \ldots, a_{m-1})^T$
Vervolgens kunnen we een nieuwe s berekenen door $A \cdot s$ te berekenen met

$$A = \begin{bmatrix} 0 & 1 & \cdots & 0 & 0 \\ & & \vdots & & \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ q_m & q_{m-1} & \cdots & q_2 & q_1 \end{bmatrix}$$

Deze matrix wordt de metgezelmatrix genoemd (Engels: companion matrix).
Verder zal de volgende functie (de connectiepolynoom) van pas komen:

$$q(x) = \sum_{i=0}^{m} q_i x^i$$

waarbij $q_0 = -1$.
Het karakteristieke polynoom van de metgezelmatrix kan op de volgende manier uitgerekend worden:

definieer voor $i = 2, 3, 4, \ldots$

$$D_i = \begin{vmatrix} -x & 1 & \cdots & 0 & 0 \\ & & \vdots & & \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & -x & 1 \\ q_i & q_{i-1} & \cdots & q_2 & q_1 - x \end{vmatrix}$$

En merk op dat

1. D_m het karakteristieke polynoom van A is
2. de determinant $\forall i \geq 2$ zonder de linker-kolom en de onderste rij altijd 1 oplevert
3. $D_2 = (-1)(q_2 + q_1 x - x^2)$ waarbij $-x^2 = +q_0 x^2$.

Nu kunnen we aan de hand van de recursie

$$D_i = (-1)^{i-1} q_i \cdot 1 + (-x)\hat{\det}_{i-1}$$

concluderen dat de inductiehypothese $D_i = (-1)^{i-1}(\sum_{j=0}^{i} q_{i-j} x^j)$ geldt voor $i = 2$ en hoger (tot en met m omdat q_{m+1} onbekend is).
En uit inductie volgt dan dat het karakteristieke polynoom van A (namelijk D_m) dus gelijk is aan het reciproke q^* polynoom van $q(x)$ op een eventuele factor -1 na.

$$q^*(x) = x^m \cdot q(\frac{1}{x})$$
$$\det(A - xI) = (-1)^{m-1} q^*(x)$$

Waarbij het relevant is om op te merken dat $q^*(x)$ irreducibel is dan en slechts dan als $q(x)$ irreducibel is.

In het voorafgaande voorbeeld zouden we te maken hebben met een metgezelmatrix van de vorm:

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 2 & 3 \end{pmatrix}, s_0 = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix}$$

Verder is de het karakteristieke polynoom in het voorbeeld $q(x) = -1 + 3x + 2x^2 + x^3$.

4.4.1 orde

Bij de verdere analyse van LFSRs en hun output is de orde van een polynoom van belang.
Definitie: De multiplicatieve orde van een polynoom $p(x)$ is de kleinste $n \in \mathbb{N}$ waarvoor geldt dat

$$p(x) | x^n - 1$$

Met deze definitie kunnen we inzien dat de orde van $q(x)$ en $q^*(x)$ hetzelfde zijn:
R is de ring waar de coëfficiënten in zitten.
Als $q(x)|x^n - 1$ dan is er een $g(x) \in R[x]$ zodat $q(x)g(x) = x^n - 1$.
Als $g^*(x) = x^{\deg(g)} g(\frac{1}{x})$ de reciproke polynoom is van $g(x)$, zien we dat $q^*(x)g^*(x)$ dan gelijk is aan $1 - x^n$, de reciproke polynoom van $x^n - 1$. Hiermee geldt dus dat $q^*(x)|x^n - 1 \Leftrightarrow q(x)|x^n - 1$.

De orde van matrix A is net iets anders gedefiniëerd.
De orde van de matrix A is de kleinste n waarvoor geldt $A^n = I$. Als een dergelijke n niet bestaat, zeggen we dat A geen orde heeft. Als een dergelijke A wel bestaat is ook bewezen dat de inverse $A^{-1} = A^{n-1}$ bestaat.

Nu blijkt er echter een mooie relatie te zijn tussen de orde van $q^*(x)$ en die van A. Hiervoor gaan we de volgende uitdrukking bekijken:

$$q^*(A) \cdot s = q_m \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{m-1} \\ a_m \end{bmatrix} + q_{m-1} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \\ a_{m+1} \end{bmatrix} + \cdots + q_0 \begin{bmatrix} a_m \\ a_{m+1} \\ \vdots \\ a_{2m-2} \\ a_{2m-1} \end{bmatrix}$$

De lineaire recurrentie van (a_i) vertelt ons voor iedere rij van de uitkomst dat de som 0 is, waardoor we weten dat $q^*(A) = 0$.
Met het homomorfisme $\phi : R[x]/(q^*) \to Hom(R^m, R^m)$ die aan $h(x) = \sum_{i=0}^{r} h_i x^i$ het homomorfisme $h(A) = \sum_{i=0}^{r} h_i A^i$ en de bovenstaande conclusie dat $q^*(A) = 0$ zien we dat $(q*)$ ook een ideaal is binnen de ruimte van homomorfismes. Nu weten we dat $\phi(x^k) = A^k$ en daarmee weten we dat de orde van A en q^* gelijk zijn.
Hiermee zien we dat $q^*(A) = 0$ wat we kunnen gebruiken bij het vinden van de orde van A.

Omdat de LFSR uiteindelijk een periodieke rij als output zal geven, is er een kleinste periode t waarvoor geldt $A^t \cdot s = I \cdot s$ voor een s die uiteindelijk in je LFSR zal zitten.
Als er sprake is van een strikt periodieke rij geldt $A^t = I$.
strikte periodiciteit kan dus verkregen worden door de juiste begintoestand s te kiezen of door een LFSR te kiezen waarbij strikte periodiciteit afgedwongen wordt door $q_m \neq 0$. Dat $q_m \neq 0$ strikte periodiciteit afdwingt zal later inzichtelijk worden als we de rij (a_i) anders gaan beschrijven.

In het voorbeeld geldt dat de orde 5 is, wat samenhangt met het feit dat $q(x) = -1+3x+2x^2+x^3$ een deler is van $x^5 - 1$. Er geldt namelijk $(1 + 3x + 1x^2)q(x) = x^5 - 1$.

4.4.2 polynomen die de output beschrijven

De output van een LFSR is een rij coëfficiënten die we in een machtreeks $a(x) = \sum_{i=0}^{\infty} a_i \cdot x^i$ kunnen stoppen. Deze machtreeks voldoet dan dankzij de lineair recurrente eigenschap van de rij (a_i) aan het volgende:

$$q(x) \cdot a(x) = f(x) \in R[x]$$

Dat de uitkomst een polynoom is, en géén machtreeks, zie je wanneer je de coëfficiënt f_k wil uitrekenen met $k \geq m$. We zien dan dat de lineair recurrente eigenschap van (a_i) met betrekking tot q ervoor zorgt dat $\forall k \geq m : f_k = 0$.
De schrijfwijze van $a(x)$ is hierdoor vaak $\frac{f(x)}{q(x)}$ waarbij f en q polynomen zijn met een graad kleiner of gelijk aan m. De machtreeks $a(x)$ draagt de output van het LFSR in zich, en het verbindingspolynoom draagt de coëfficiënten q_i in zich. Het overige polynoom $f(x)$ heeft dan alle overige informatie in zich: de begintoestand van het LFSR.

4.4.3 Op zoek naar $a = f/q$

Als de machtreeks $a(x)$ die strikt periodiek is gegeven is, kunnen we $a(x)$ schrijven als $\frac{f^*(x)}{x^T-1}$ waarbij $f^*(x)$ de periode is. Oftewel de coëfficiënten van f^* zijn die coëfficiënten die één periode van (a_i) omschrijven. De lengte van de periode is dan T, zodat

$$a(x) = \frac{f^*(x)}{x^T - 1} = (-1)f^*(x) \cdot \sum_{i=0}^{\infty}(x^{iT}) = (-1)f^*(x) \cdot (1 + x^T + x^{2T} + x^{3T} + \ldots)$$

Hier wordt de periode steeds achter elkaar geplakt zonder dat ze met elkaar in aanraking komen. Nu hebben we dus een breuk gevonden die precies onze strikt periodieke rij (a_i) representeert in de vorm van $a(x)$. Merk op dat de teller een graad heeft die kleiner is dan de teller van de noemer. Voor uiteindelijk periodieke rijtjes (a_i) die pas periodiek wordt na de eerste k coëfficiënten kunnen we iets soortgelijks doen:

$$a(x) = p(x) + x^k \cdot a^*(x)$$

met $\deg(p) \leq k$ zodat $a^*(x)$ een strikt periodieke rij representeert

$$a^*(x) = \frac{f^*(x)}{x^T - 1}$$
$$a(x) = \frac{p(x)(x^T - 1) + f^*(x)}{x^T - 1} = \frac{f^{**}(x)}{x^T - 1}$$

waarbij $f^{**}(x)$ dus een polynoom is die graad groter dan T kan hebben.

Echter willen we een vorm hebben waarin de noemer van de breuk $q(x)$ is.
We weten dat $a(x)q(x)$ een polynoom is, zodat $\frac{f(x)}{q(x)} = a(x)$ bestaat voor polynomen f en q.
We zien nu dat $\frac{f(x)}{q(x)} = a(x) = \frac{f^*(x)}{x^T-1}$ geldt. Verder weten we uit de definitie van de orde van $q(x)$ dat het de kleinste t is waarvoor $q(x)|x^t - 1$. En de T in $\frac{f^*(x)}{x^T-1}$ is gekozen als de kleinste t waarvoor $x^t \equiv 1 \mod q(x)$ door de lineaire recurrentie. Per definitie is de orde van $q(x)$ dus gelijk aan de (kleinste) periode en dus geldt dat $q(x)|x^T - 1$. Als we $p(x)$ nemen zodat $p(x)q(x) = x^T - 1$, dan geldt $p(x)f(x) = f^*(x)$ waarmee we $f(x)$ kunnen uitrekenen.

Als we dus een LFSR willen beschrijven hebben we dus eigenlijk al genoeg aan een goed gedefinieerde $R[x]$, een $f(x)$ en een $q(x)$.
Dat deze schrijfwijze altijd de output in de vorm van een machtreeks genereert komt goed van pas. Hierdoor kunnen we namelijk een eventuele factorisatie van $q(x) = q_1(x) \cdot q_2(x)$ gebruiken met $deg(q_1), deg(q_2) \geq 1$:

$$a(x) = \frac{f(x)}{q(x)} = \frac{f_1(x)}{q_1(x)} + \frac{f_2(x)}{q_2(x)} = a_1(x) + a_2(x)$$

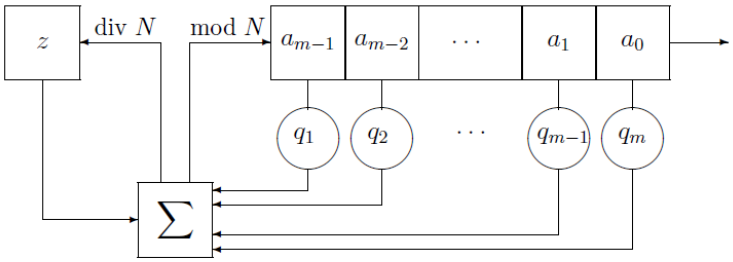
waarbij $a_1(x)$ en $a_2(x)$ periodieke coëfficiënten hebben met een periode kleiner of gelijk aan T. Als je de machtreeksen bij elkaar optelt krijg je $a(x)$ terug. Hierbij geldt dat $periode(a(x)) = kgv(periode(a_1(x)), periode(a_2(x)))$.
Een praktische doelstelling die we onszelf kunnen stellen bij LFSRs is het verkrijgen van een zo onvoorspelbaar mogelijke rij (a_i) met een zo klein mogelijke lengte m van het LFSR. Mochten we twee geschikte kandidaten $q(x)$ vinden, waarvan de periodes relatief priem zijn, dan kunnen we deze altijd nog samenvoegen tot een grotere LFSR met een significant grotere periode door de rijen elementsgewijs bij elkaar op te tellen.
Voor nu is het dus interessant om de grootste periode te verkrijgen met een polynoom dat niet te factoriseren is, oftewel irreducibel.

5 FCSR

De theorie over FCSRs die in dit hoofdstuk aan bod komt, komt uit het boek Algebraic Shift Register Sequences[2]. Naast de LFSRs bestaan er ook zogehete Feedback with Carry Shift Registers, wat we afkorten met FCSR.
Deze FCSRs hebben veel overeenkomsten met LFSRs, maar ook genoeg verschillen. Voor we naar de FCSR gaan kijken wil ik eerst twee definities geven.
De eerste is $(div\ N) : \mathbb{Z} \to \mathbb{Z}$ waarbij $x(div\ N) := \lfloor \frac{x}{N} \rfloor$.
Waarna we $(mod\ N) : \mathbb{Z} \to \{0, 1, \ldots, N-1\}$ definëren als $x(mod\ N) = x - x(div\ N) \cdot N$.
Dit betekent dat $x(mod\ N)$ een geheel getal is tussen 0 en $N-1$. Hiermee is $x(mod\ N)$ een unieke representant voor alle getallen in de equivalentieklasse $xmodN$.

5.1 inleiding FCSR

De verschillen worden het snelst duidelijk in het volgende plaatje:



Figuur 3: Een FCSR in zijn begintoestand

Bij een klokslag is de volgorde van de pijltjes nu minder vanzelfsprekend, maar wel belangrijk. Eerst wordt alle informatie naar het Σ-blok gebracht waar dan de nieuwe z en volgende a_i berekend zullen worden. De rechterkant van bovenstaande afbeelding is gelijk aan een groot deel van de LFSR. Het enige verschil zit aan de linkerkant: een extra geheugencel met de bijbehorende pijlen.
Om te laten zien hoe dit verschil tot uiting komt in het functioneren van dit systeem laat ik zien hoe een rekenstap in zijn werk zal gaan: In tegenstelling tot de rekenwijze van de LFSR berekenen we hier eerst het tussenresultaat $\sigma_m = z_{m-1} + \sum_{i=1}^{m} q_i a_{m-i}$ welke in het afgebeelde Σ blok zal komen.
Met de waarde σ kan vervolgens a_{m-1} en z_m berekend worden.

$$a_m = \sigma_m(mod\ N)$$
$$z_m = \sigma_m(div\ N)$$

Merk op dat wanneer $z_{m-1} = 0$, dat dan $a_m = \sigma_m(mod\ N)$, wat het gedrag is wat we van een LFSR zouden verwachten.
Echter zal over het algemeen deze z niet bij iedere rekenstap 0 zijn, waardoor een FCSR (op den duur) andere coëfficiënten in zijn geheugencellen zal hebben dan een LFSR met dezelfde begintoestand en q_i's. Omdat z kan veranderen gedurende de toestandenreeks is het geen parameter en wordt deze opgenomen in de toestand van de FCSR. Waar we eerst dus spraken over enkel een set a_i'tjes zal nu ook de bijbehorende z_i vermeld moeten worden bij de toestand s.
Deze rekenstap zal herhaald worden volgens de algemenere formules:

$$\sigma_n = z_{n-1} + \sum_{i=0}^{m} q_i a_{n-i}$$
$$a_n = \sigma_n (mod\ N)$$
$$z_n = \sigma_n (div\ N)$$

Hiermee verkrijgen we steeds een toestand $s_n = (z_{n+m-1}; a_{n+m-1}, a_{n+m-2}, \ldots, a_n)$ die samen met de coëfficiënten q_i de hele rij s_i en a_i vastlegt.
Met deze regels ontstaat er een nieuw/ander begrip van lineaire recurrentie, namelijk **lineaire recurrentie met carry** waarop we weer rekenregels gaan baseren. De definitie van lineair recurrent met carry luidt:

$$a_n + z_n \cdot N = \sum_{i=1}^{m} q_i a_{n-i} + z_{n-1}$$

5.2 voorbeeld

Om een indruk te geven van de verschillen tussen de LFSR en de FCSR zal ik een voorbeeld geven van een LFSR en FCSR met dezelfde begintoestand:
Als voorbeeld neem ik $m = 2$, met $q_0 = -1$, $q_1 = 0$ $q_2 = 2$, $a_0 = 1$ en $a_1 = 0$ waarbij we alles modulo 5 bekijken.
LFSR
Omdat $q(x) = -1 + 2x^2 | x^8 - 1$ krijgen we een uitkomst met een periode die een deler is van 8.
Met de gegeven begintoestand krijgen we de output:

$$(a_i) = (1, 0, 2, 0, 4, 0, 3, 0, 1, 0, 2, 0, 4, 0, 3, 0, 1, 0, \ldots)$$

We zien hier dat de periode met lengte 8 te representeren is als
$a(x) \cdot (1 - x^8) = \frac{a(x)}{1+x^8+x^{16}+\ldots} = (1 + 2x^2 + 4x^4 + 3x^6)$.
FCSR
De output van de FCSR die je krijgt als je als de gegeven begintoestand met $z_{m-1} = 0$ neemt is:

$$(a_i) = (1, 0, 2, 0, 4, 0, 3, 1, 1, 3, 2, 1, 0, 3, 0, 1, 1, 2, 2, 4, 4, 3 \ldots)$$

waarbij de periode 42 is wat blijkt uit het feit dat de multiplicatieve orde van 5 $\mod 49$ ($p \mod q$) gelijk is aan 42 en $ggd(f, q) = 1$. De orde vertelt ons namelijk dat $q = -1 + 2 \cdot 5^2 | 5^{42} - 1$ zodat we weten dat de periode een deler is van 42. De definitie van multiplicatieve orde sluit ook uit dat er een deler $d|42$ is met $d < 42$ zodat $5^d \equiv 1 \mod 49 \Leftrightarrow 49|5^d - 1$. De eis $ggd(f, q) = 1$ doet ertoe omdat we een breuk $\frac{f}{q}$ willen vereenvoudigen en de noemer de lengte van de periode bepaalt. Als we de noemer kleiner kunnen maken door het wegstrepen van gemeenschappelijke priemfactoren wordt het probleem makkelijker en de periode korter. In ons geval ($f = a_0 + a_1 \cdot p + z_1 \cdot p^2 = -1$) is de periode van lengte 42. Ook in dit geval is te controleren dat de eerste 22 elementen geen herhaling bevatten zodat 42 de enige mogelijke deler van 42 is die overblijft, maar het is dus mogelijk om kleinere periodes te krijgen door het kiezen van een andere begintoestand. Neem bijvoorbeeld $f = -49$ ofwel $z_1 = 1, a_1 = a_0 = 4$.

5.3 van Polynomen naar N-adisch

Bij de LFSRs kwam het goed uit om de coëfficiënten te representeren in de vorm van een polynoom in $R[x]$ zodat $q(x) \cdot a(x)$ weer een polynoom zou zijn. Bij het narekenen van dit product komt de lineaire recurrentie van de LFSR goed van pas. Bij de FCSRs gaan we iets soortgelijks krijgen, maar dan niet met polynomen, maar met N-adische getallen.

Bij de analyse van LFSRs gebruikten we polynomen met daarin de variabele x. Om een beeld te geven wat er ongeveer gaat gebeuren geef ik de volgende vuistregel: overal waar je eerder een x zag staan zet je nu een N neer. We praten nu over N-adische getallen in plaats van polynomen. Als het zo simpel gezegd wordt lijkt het net of het alleen een verschil in notatie is, maar er zijn wel degelijk verschillen. Zo wordt bij een berekening in $\mathbb{F}_N[x]$ (bij LFSRs) menigmaal gebruikgemaakt van twee beweringen die anders zijn in de N-adische getallenruimte.

1. $N \equiv 0 \mod N$ zonder carry
   Als we de polynomen $N-1$ en 1 bij elkaar optellen in $\mathbb{F}_N[X]$ krijgen we als uitkomst 0. Als we bij N-adische getallen $N-1$ en 1 bij elkaar optellen vinden we de waarde N. Binnen het beeld dat x en N een soortgelijke rol vervullen respectievelijk bij LFSRs en FCSRs is dit vreemd. Je zou immers 1 niet zo vaak bij elkaar op kunnen tellen tot je het polynoom x krijgt, maar bij FCSRs kan dit wel. Het onthouden hoe vaak het modulo getal N van de eerste coëfficiënt af wordt getrokken om tot een getal in $\{0, 1, \ldots, N-1\}$ te komen wordt onthouden en meegenomen in de berekening van de volgende coëfficiënt. Dit getal meenemen naar de berekening van de volgende coëfficiënt wordt 'carry' genoemd. We zien het woord Carry letterlijk terug in de Feedback with Carry Shift Registers, maar ook in de rekenregels van N-adische getallen. Dus het niet gebruiken van een carry bij LFSRs zorgt voor een cruciaal verschil met FCSRs.

2. lineaire recurrentie
   Bij het vermenigvuldigen van de connectiepolynoom $q(x)$ met de machtreeks $a(x)$ die de output van die LFSR representeert wordt een polynoom verkregen. Dit komt doordat er vaak gebruik werd gemaakt van de lineaire recurrentie bij LFSRs:
   $$0 = q_0 \cdot a_n + q_1 \cdot a_{n-1} + q_2 \cdot a_{n-2} + \ldots + q_m \cdot a_{n-m} \tag{2}$$
   Echter geldt deze vergelijking niet meer bij FCSRs, maar geldt de volgende vergelijking:
   $$a_n + z_n \cdot N = \sum_{i=1}^{m-1} q_i a_{n-i} + z_{n-1}$$

We willen N-adische getallen gebruiken om tot een notatiewijze te komen die we ook hadden bij de LFSRs. Hiervoor definiëren we de de getallen $a, q \in \mathbb{Q}_{\mathbb{N}}$ als volgt:

$$a = \sum_{i=0}^{\infty} a_i N^i$$
$$q = \sum_{i=0}^{m} q_i N^i$$

5.3.1 periodiek

Eerst willen we uitleggen waarom a uiteindelijk periodiek zal zijn wanneer het de output van een FCSR betreft. Het aantal toestanden waarin de a_i's verkeren is net als bij de LFSRs eindig, echter is het aantal toestanden waarin z zich bevindt in eerste instantie niet begrensd. Op den duur zal de waarde van z echter onder een systeem-afhankelijke bovengrens blijven. We hoeven enkel aan te tonen dat er een bovengrens bestaat, en hoeven niet de kleinste te vinden. Bij het bestuderen van het gedrag van z_i kunnen we genoegen nemen met de volgende ongelijkheid, waarbij we gebruik maken van $a_i, |q_i| \leq N-1$.

$$|z_n| = |\lfloor \frac{\sigma_n}{N} \rfloor| = |\lfloor \frac{z_{n-1} + \sum_{i=0}^{m} a_i a_{n-i}}{N} \rfloor| < |\frac{z_{n-1} + \sum_{i=0}^{m} a_i a_{n-i}}{N}| + 1$$
$$\leq \frac{|z_{n-1}| + m(N-1)^2}{N} + 1 = \frac{|z_{n-1}| + mN^2 - 2mN + m + N}{N} < \frac{|z_{n-1}|}{N} + mN$$

Uit deze vergelijking volgt dat als $|z_{n-1}| < \frac{mN^2}{N-1}$ dat dan
$|z_n| < \frac{mN}{N-1} + mN = \frac{mN+mN(N-1)}{N-1} = \frac{mN^2}{N-1}$.
Als we nu nog kunnen aantonen dat de rij $|z_n|$ strikt dalend is bij de elementen waar $|z_n| \geq \frac{mN^2}{N-1}$ dan weten we dat de rij op den duur wel elementen heeft die voldoen aan $|z_n| < \frac{mN^2}{N-1}$.

$$|z_{n-1}| > \frac{mN^2}{N-1} \Leftrightarrow mN < \frac{|z_{n-1}|(N-1)}{N}$$
$$|z_n| < \frac{|z_{n-1}|}{N} + mN < \frac{|z_{n_1}|}{N} + \frac{|z_{n-1}|(N-1)}{N} = |z_{n-1}|$$

Het is hiermee niet aangetoond dat de rij snel zal dalen. Het delen door N zal echter zorgen voor een snelle daling bij een grote $|z_{n-1}|$. Propositie 4.7.1 op bladzijde 89 uit Algebraic Shift Register Sequences van Mark Goresky en Andrew Klapper stelt dat als $q \in \{0, ..., N-1\}$ dat $0 \leq z_i < w = \sum_{i=1}^{m} q_i$ bereikt wordt met logaritmische complexiteit. Met initiële $z_{m-1} \geq w$ wordt dit bereikt binnen $\lfloor \log_N(z_{m-1} - w) \rfloor + m$ stappen en met $z_{m-1} < 0$ binnen $\lceil \log_N |z_{m-1}| \rceil + m$ stappen.
Nu we weten dat $\exists N, w : \forall i > N : 0 \leq z_i < w$, weten we dat voor index $i > N$ sprake is van een eindig aantal mogelijke toestanden.
Volgens het ladenprincipe weten we dan dat er minstens twee verschillende indices i, j zijn die dezelfde toestand $s_i = s_j$ beschrijven. Omdat de FCSR met dezelfde toestand altijd dezelfde vervolgtoestand uitrekent, weten we dat $\forall n \geq 0 : s_{i+n} = s_{j+n}$ en zal de s dus periodiek zijn met periode $n|j-i$ en vanaf index $\min(i, j)$ (of eerder). En tot slot: als s_i periodiek is, is a_i dat ook.

5.3.2 gebruiken in N-adische getallen

Nu we weten dat ook de output van de FCSR uiteindelijk periodiek is, kunnen we laten zien dat we de output kunnen vastleggen in een N-adische breuk. Een periodiek N-adisch getal kan gerepresenteerd worden door de periode met opvolgende coëfficiënten h_i en zijn periode n

$$a = \frac{\sum_{i=0}^{n-1} h_i N^i}{1 - N^n} = (h_0 + h_1 N + \ldots + h_{n-1} N^{n-1})(1 + N^n + N^{2n} + N^{3n} + \ldots)$$

Merk hierbij op dat er bij deze schrijfwijze geen coëfficiënten bij elkaar opgeteld hoeven worden om tot een uitkomst te komen. Hierdoor speelt de carry in deze vergelijking geen rol. Een soortgelijk schrijfwijze bestaat ook voor een uiteindelijk periodieke rij a_i. De vergelijking $a = \frac{h}{1-N^n}$ klopt N-adisch gezien, waardoor dus bewezen is dat a te schrijven is als een breuk in de N-adische getallen.
Een stap verder gaan naar $a = \frac{f}{q}$ voor een f met $|f| < |q|$ en verbindingsgetal q is echter moeilijker.

5.3.3 Bewijs dat $a = \frac{f}{q}$

$a = \frac{f}{q}$ met q het verbindingsgetal is stelling 4.3.1 uit Goresky en Klapper en wordt als volgt bewezen:

$$s = a_0 + a_1 N + \ldots + a_{m-1} N^{m-1}$$

$$a = s + \sum_{n=m}^{\infty} a_n N^n$$

$$a_n = \sum_{i=1}^{m} q_i a_{n-i} + (z_{n-1} - N z_n)$$

Deze vergelijkingen leveren samen:

$$a = s + \sum_{n=m}^{\infty} (\sum_{i=1}^{m} q_i a_{n-i}) N^n + \sum_{n=m}^{\infty} (z_{n-1} - N z_n) N^n$$

Waarbij de rechtersom te vereenvoudigen is tot

$$\sum_{n=m}^{\infty} (z_{n-1} - N z_n) N^n = z_{m-1} N^m + \sum_{n=m}^{\infty} ((-N z_n) N^n + z_n N^{n+1}) = z_{m-1} N^m$$

zodat we bijna zonder carry's verder kunnen rekenen met

$$\begin{aligned} a &= s + z_{m-1} N^m + \sum_{n=m}^{\infty} (\sum_{i=1}^{m} q_i N^i a_{n-i} N^{n-i}) \\ &= s + z_{m-1} N^m + \sum_{i=1}^{m} q_i N^i (\sum_{n=m}^{\infty} a_{n-i} N^{n-i}) \\ &= s + z_{m-1} N^m + \sum_{i=1}^{m} q_i N^i (a - \sum_{j=0}^{m-i-1} a_j N^j) \end{aligned}$$

waarbij in het geval van $i = m$ de som leeg is en dus 0

$$= s + z_{m-1} N^m + a \sum_{i=1}^{m} q_i N^i - \sum_{i=1}^{m-1} \sum_{j=0}^{m-i-1} q_i N^i a_j N^j$$

wat genoeg uitgewerkt is om het te herschrijven tot

$$a(1 - \sum_{i=1}^{m} q_i N^i) = a - a \sum_{i=1}^{m} q_i N^i = s + z_{m-1} N^m - \sum_{i=1}^{m-1} (q_i N^i \sum_{j=0}^{m-i-1} a_j N^j)$$

$$a = \frac{s + z_{m-1} N^m - \sum_{i=1}^{m-1} (q_i N^i \sum_{j=0}^{m-i-1} a_j N^j)}{1 - \sum_{i=1}^{m} q_i N^i} = \frac{f}{q}$$

We definiëren het verbindingsgetal q in plaats van het verbindingspolynoom.

$$q = -1 + \sum_{i=1}^{m} q_i N^i = \sum_{i=0}^{m} q_i N^i$$

en we schrijven de output rij (a_i) om naar het N-adische getal $a = \sum_{i=0}^{\infty} a_i N^i$.
In de notatiewijze bij de LFSR kwamen we regelmatig een x tegen, die we bij de FCSR zullen vervangen door een N.
Deze wijziging lijkt wellicht onschuldig, maar representeert een vrij fundamentele wijziging.

5.4 overeenkomsten met LFSR

Als (a_i) lineair recurrent is met betrekking tot het eindige gehele getal q, dan geldt dat er een f is zodat:

$$\frac{f}{q} = a$$

Dit brengt veel overeenkomsten met LFSRs met zich mee.
Hier geldt dat wanneer $q = q_1 \cdot q_2$ met $q_1, q_2 > 1$ dat we a dan kunnen schrijven als

$$a = \frac{f}{q} = \frac{f_1}{q_1} + \frac{f_2}{q_2} = a_1 + a_2$$

Voor praktische toepassingen geldt nu nog steeds dat we dus op zoek zijn naar een irreducibele q die een zo groot mogelijke periode heeft met een zo klein mogelijke lengte m (de grootste index waarvoor $q_i \neq 0$).

5.5 interpretatie voorbeeld

Het voorbeeld dat in paragraaf 2 gegeven werd kan nu met nieuwe inzichten beter toegelicht worden.
Doordat de opeenvolgende coëfficiënten van de FCSR een extra relatie hebben gekregen die betrekking heeft op zijn buren (index n hoger of lager) wordt het systeem een stuk complexer. Je kunt ook stellen dat een LFSR $z = \sigma(div\ N)$ steeds niet benut waardoor informatie verloren gaat en daarmee de LFSR de simpelere versie is van de FCSR.

6 register synthese

De theorie over zowel LFSR- als FCSR-synthese die in dit hoofdstuk aan bod komt, komt uit het boek Algebraic Shift Register Sequences[2].
Tot nu toe hebben we het enkel gehad over situaties waarbij een shift register gegeven was en de output daaruit verkregen werd. Echter kan de situatie ook omgedraaid worden. Dan is de rij a gegeven en wordt er gezocht naar registers die deze reeks genereren. Voor praktische toepassingen is het verder relevant om te zoeken naar dergelijke registers die zo min mogelijk rekenkracht nodig hebben of zo min mogelijk geheugen. Deze eigenschappen hangen in het rekenmodel van de shift registers af van de lengte ofwel span (Engels) van deze registers. Voor LFSRs is de span uit te drukken in het aantal geheugencellen dat gebruikt wordt, terwijl bij FCSRs ook rekening gehouden moet worden met de grootte van de geheugencel waar z in opgeslagen wordt. Hierdoor is span een begrip dat anders wordt gebruikt per soort shift register.
Als we een shift register gegeven hebben en we willen de span hiervan weten zal dit relatief makkelijk zijn. Echter gaat het in deze sectie om register synthese waarbij we ons register nog moeten vinden/construeren. Tijdens deze zoektocht willen we een register vinden met een zo klein mogelijke span die a construeert. In deze probleemstelling kunnen we dus spreken van de span van een rij. Voor de klasse van sequence generators $\mathbb{F}$ spreken we over de $\mathbb{F}$-span $\lambda^{\mathbb{F}}$:

$$\lambda^{\mathbb{F}}(a) = \inf\{size(F) : F \in \mathbb{F} \text{ and } F \text{ outputs } a\}$$

Het liefst vinden we nu een *register synthese algoritme* die met een eindige rij b van coefficiënten een generator F en toestand s_0 kan vinden zodat de outputrij $F(s_0)$ begint met b.

6.1 LFSRs

Hierbij ontstaat de vraag hoe lang de deelrij b van a moet zijn voordat het algoritme met het antwoord komt dat volstaat voor de hele rij a. Een dergelijk algoritme voor LFSRs probeert het volgende systeem op te lossen:

$$\begin{aligned} a_m &= q_1 a_{m-1} + q_2 a_{m-2} + \cdots + q_m a_0 \\ a_{m+1} &= q_1 a_m + q_2 a_{m-1} + \cdots + q_m a_1 \\ &\vdots \\ a_{m+i} &= q_1 a_{m-1+i} + q_2 a_{m-2+i} + \cdots + q_m a_i \end{aligned}$$

Hiervoor is het van belang dat er voor het berekenen van de m onbekende q_i's m vergelijkingen zijn. Er kan dus $i = 0, 1, 2, \ldots, m-1$ ingevuld worden zodat a_0 tot en met a_{2m-1} bekend moeten zijn voordat het algoritme het stelsel kan oplossen. De lezer kan eenvoudig nagaan dat deze regels niet afhankelijk zijn als a geen periode kleiner dan m heeft. Dit betekent dat voor LFSRs geldt dat er een deelrij van a ingevoerd moet worden met een lengte van minstens $2m = 2\lambda^{\mathbb{F}}$ om een geschikte LFSR F te kunnen vinden.

Voor LFSRs is er een register synthese algoritme: het Berlekamp-Massey algoritme. Het algoritme werkt in stappen met index i en zoekt hierbij steeds een geschikte graad i benadering met $f_i(x)$ en $q_i(x)$ zodat

$$q_i(x)a(x) \equiv f_i(x) \mod x^{i+1}$$

Bij een rekenstap kan het zijn dat het antwoord van de vorige stap ook voldoet aan de eis van de huidige stap. Op dat moment volstaat $(f_{i+1}, g_{i+1}) = (f_i, g_i)$. Mocht deze niet volstaan dan geldt dat er een $b \neq 0$ bestaat zodat

$$q_i(x)a(x) \equiv f_i(x) + bx^i \mod x^{i+1}$$

Ook willen we graag gebruik maken van een $m < i$ waar een zelfde probleem optrad met $c \neq 0$ en

$$q_m(x)a(x) \equiv f_m(x) + cx^m \mod x^{m+1}$$

We kunnen dan namelijk

$$f_{i+1} = f_i - (b/c)x^{i-m}f_m, \quad q_{i+1} = q_i - (b/c)x^{i-m}q_m$$

kiezen welke zullen volstaan aan de graad $i+1$ benadering

$$q_{i+1}(x)a(x) \equiv f_{i+1}(x) \mod x^{i+1}$$

Dit levert altijd een benadering voor de vergelijking $q(x)a(x) = f(x)$. Echter kunnen we uitkomsten van verschillende afmetingen krijgen door een andere m te kiezen. De lengte functie voor de LFSR is

$$\Phi(f, q) = \max(\deg(f) + 1, deg(q))$$

Hiermee definiëren we vervolgens het keerpunt i als een index uit het algoritme waarvoor geldt dat

$$\Phi(f_{i+1}, q_{i+1}) > \Phi(f_i, q_i)$$

In de Berlekamp-Massey algoritme kiezen m steeds als het meest recente keerpunt lager dan i. Deze keuze garandeert dat het algoritme een uitkomst geeft met de laagst mogelijke Φ. En als $i \geq 2\lambda(a)$ weten we dat $\Phi(f_i, q_i) = \lambda(a)$.
In het rekenproces wordt steeds een benadering gegeven die de tussenuitkomst iets dichter bij de gewenste uitkomst brengt. Het wordt op een manier gedaan die al eerder bekend was: het construeren van kettingbreuken.

Voorbeeld
De algoritme van Berlekamp-Massey staat uitgewerkt in Appendix A in gewone code zodat het volgende voorbeeld wellicht makkelijker gevolgd kan worden.
In dit voorbeeld proberen we de functie $\frac{-1+x}{-1+x+2x^2} = \frac{1+2x}{1+2x+x^2}$ te benaderen met polynomen die coëfficiënten in $\mathbb{F}_3$ hebben.
(f_i, q_i) is de benadering zodat $f_i(x) - a(x)q_i(x) \equiv 0 mod x^i$. We krijgen de volgende benaderingen:

	f_i	q_i
$i = 0$	0	1
$i = 1$	1	1
$i = 2$	1	1
$i = 3$	1	$1 + x^2$
$i = 4$	$1 + 2x$	$1 + 2x + x^2$
$i = 5$	$1 + 2x$	$1 + 2x + x^2$

De theorie over deze algoritme zegt ons dat we bij het invoeren van $2 \cdot \lambda = 6$ elementen van (a_i) onze f en q gevonden zouden moeten hebben. In bovenstaande tabel zien we dat dit al is bereikt bij 5 elementen. In de praktijk zijn echter de echte functies $f(x)$ en $q(x)$ en daarmee dus ook λ niet zomaar bekend. Een voordeel van dit algoritme is dat je het gewoon kunt hervatten zodra er meer elementen van (a_i) bekend zijn.

6.2 kettingbreuken

De breuk $\frac{f}{q}$ kan ook worden geschreven als een kettingbreuk. Een kettingbreuk is een uitdrukking van de vorm $c_0 + \frac{1}{c_1 + \frac{1}{c_2 + r_2}}$ waarbinnen c_i gehele getallen zijn en r_i de rest met $0 \leq r_i < 1$. In voorgaande uitdrukking kan r_2, mits ongelijk aan 0, weer uitgedrukt worden als $\frac{1}{c_3 + r_3}$ en zo kan men door blijven gaan zodat een rij (c_i) verkregen wordt. Deze rij kan eindig zijn als ergens $r_i = 0$ ontstaat, of oneindig. De kettingbreuk-expansie van een reëel getal x is uiteindelijk periodiek dan en slechts dan als het een oplossing is van een kwadratische vergelijking met gehele coëfficiënten.

Om een indruk te geven waar we mee te maken hebben, geef ik een voorbeeld:

Voorbeeld van kettingbreuk van reëel getal
In dit proces herhalen we de volgende stappen:
We beginnen iedere iteratie met een z_i, waarvan we de $c_i = \lfloor z_i \rfloor$ nemen. Tot slot rekenen we $r_i = z_i - c_i$ uit welke we omdraaien om $z_{i+1} = \frac{1}{r_i}$ te vinden.

$$\begin{aligned}
x = z_0 &= 2 + (\sqrt{7} - 2) \\
z_1 &= \frac{1}{\sqrt{7} - 2} = 1 + \frac{\sqrt{7} - 1}{3} \\
z_2 &= \frac{3}{\sqrt{7} - 1} = 1 + \frac{\sqrt{7} - 1}{2} \\
z_3 &= \frac{2}{\sqrt{7} - 1} = 1 + \frac{\sqrt{7} - 2}{3} \\
z_4 &= \frac{3}{\sqrt{7} - 2} = 4 + (\sqrt{7} - 2) \\
z_5 &= z_1
\end{aligned}$$

Uit bovenstaande vergelijkingen kan de uiteindelijke periodieke rij $(c_i) = (2, 1, 1, 1, 4, 1, 1, 1, 4 \ldots)$ worden gevonden. Bij het hierboven beschreven proces zijn we enkel x aan het herschrijven en verliezen we geen gegevens. Als je echter bij een van de stappen stopt en enkel de tot dan toe berekende c_i gebruikt krijg je een benadering van x. De laatste rest r_i die berekend wordt gebruiken we bij deze benadering dus niet en we doen alsof deze 0 is. Hierdoor houden we een rij over van alleen gehele getallen. Hoe langer deze lijst is hoe beter de benadering. En de kwaliteit van deze benadering is in formules uit te drukken.
Bij het vormen van deze benadering waarbij je gebruik maakt van c_0 t/m c_n en r_n verwaarloost, schrijf je de kettingbreuk uit. Dit kan herschreven worden tot $\frac{f_n}{q_n}$. Voor deze benadering geldt dat:

$$|x - \frac{f_n}{q_n}| < \frac{1}{q_n q_{n+1}}$$

$$q_{n+1} = \begin{cases} c_{n+1} q_n + q_{n-1} & \text{als } r_n \neq 0 \\ q_n & \text{als } r_n = 0 \end{cases}$$

En dat is de beste benadering die je kunt krijgen voor x met een positieve noemer kleiner dan $|q_n|$. Deze constructie werkt voor zowel reële getallen als voor functies. Er moet dan wel een nieuw selectiecriterium komen voor c. De functies die in deze sectie bestudeerd worden zijn van de vorm $\sum_{i=k}^{-\infty} a_i x^i$. En het "gehele "deel van deze functie zal bestaan uit de som over alle termen met index groter of gelijk aan 0. Alle termen met negatieve index vormen dan samen de rest.

Voorbeeld:
Ook deze methode kunnen we demonstreren op $\frac{-1+x}{-1+x+2x^2}$ zoals we deden bij de sectie over Berlekamp-Massey:

$$z_0 = \frac{-1+x}{-1+x+2x^2} = 0 + \frac{-1+x}{-1+x+2x^2}$$
$$z_1 = \frac{-1+x+2x^2}{-1+x} = 2x + \frac{-1}{-1+x}$$
$$z_2 = \frac{-1+x}{-1} = 1 - x$$

zodat we $(c_i) = (0, 2x, 1-x)$ krijgen die ons de volgende benadering geeft:

$$a(x) = 0 + \frac{1}{2x + \frac{1}{1-x}} = \frac{-1+x}{-1+x+2x^2}$$

Deze algoritme heeft als voordeel dat hij in het algemeen 2 keer zo weinig iteraties nodig heeft om een even goede benadering te vinden als de Berlekamp-Massey algoritme. Mocht deze algoritme gebruikt worden op een outputreeks van een LFSR dan dient deze gebruikt te worden op de functie $\hat{a}(x) = x^{-1}a(1/x)$. Mocht een nieuw element van (a_i) gevonden worden na het berekenen van de benaderingen, kan bij Berlekamp-Massey de algoritme hervat worden terwijl dit bij de kettingbreukalgoritme niet zomaar kan. De algoritme kan op $\frac{f}{q}$ gebruikt worden maar ook op een deel van de reeks $a(x)$. Bijvoorbeeld op $a(x) = x^{-1} + 2x^{-3} + 2x^{-4} + x^{-6} + O(x^{-7})$ waarbij wordt bijgehouden in het algoritme wat de termen van lagere orde zijn waar we niets meer van weten.

6.3 LFSRs in Cryptografie

Binnen de cryptografie kunnen LFSRs gebruikt worden bij de versleuteling van gegevens. Hierbij dient de output van de LFSR onvoorspelbaar te zijn zodat een derde partij niet weet welke sleutel is gebruikt. Als onvoorspelbaarheid gewenst is, is het van belang dat de lineaire span groot is. Ook als een rij door een andere soort generator wordt gegenereerd, maar wel een lage lineaire span heeft is het vervolg van de rij voorspelbaar. De Berlekamp-Massey algoritme is het bewijs dat in die situatie een equivalente LFSR gegenereerd kan worden. Hoge lineaire span is dus wel een voorwaarde voor veiligheid van gebruik, maar niet voldoende om veiligheid te garanderen. Zo is het mogelijk dat de uitkomst van een LFSR binnen een ander rekenmodel weer makkelijk te verklaren zou kunnen zijn. Een binaire m-rij, ofwel een maximale lengte rij is een rij geproduceerd door een LFSR met alfabet $\mathbb{F}_2$. Als deze LFSR m registers gebruikt produceert hij een output met periode $2^m - 1$ met slechts een 1 meer dan het aantal 0en. Binnen de output spreken we van een run als een aantal coëfficiënten na elkaar hetzelfde zijn, dus alleen 1en of 0en. Bij een binaire m-rij is de helft van de runs van lengte 1, een kwart van de runs van lengte 2, een achtste van de runs van lengte 3 enzovoorts. Een mooie statistische eigenschap. Niet-binaire m rijen bestaan ook. Deze worden allen gekenmerkt dat ze leven binnen een lichaam $\mathbb{F}_q$ met q elementen en de maximale periode $T = q^d - 1$ hebben waarbij $d = \deg(q(x))$. In deze subsectie wordt gesproken over methoden die gebruikt zijn in een poging tot het genereren van rijen met goede statische eigenschappen zoals die van binaire m-rijen. Ook worden systemen gecombineerd om zo een systeem te maken met een hogere lineaire span dan de individuele systemen.

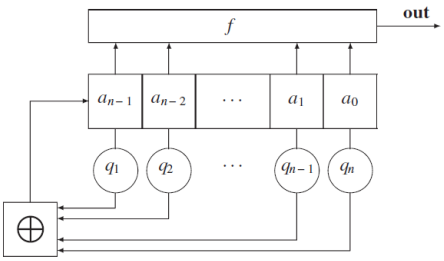
6.3.1 waarom lineair niet ideaal is

Naast de LFSRs bestaan er ook andere lineaire registers welke een toestand hebben die in een k-dimensionale ruimte zit. De nieuwe toestand wordt lineair berekend vanuit de oude toestand, maar hoeft dus geen shift te vertonen. Ook de output wordt lineair berekend vanuit de toestand. Er is een stelling[2] die zegt dat er een LFSR bestaat van maximaal lengte k die dezelfde rijen kan genereren. Dit betekent dat lineaire modificaties op de LFSRs cryptologisch niet zullen helpen als we grote lineaire span willen bereiken. We zullen onze toevlucht dus moeten zoeken in de niet-lineairiteit.

6.3.2 niet-lineair filter

De functie die van de toestand de output berekent was tot nu toe altijd lineair. Als we deze functie niet-lineair maken hebben we nog steeds dezelfde rij (a_i) alleen is dit niet meer onze output. Hoewel de periode hierdoor onbeïnvloed blijft kan de lineaire span hierdoor toenemen. Een voorbeeld om operaties niet-lineair te maken is door het gebruik van Hadamard producten op shifts van (a_i). Bij een Hadamard product is $c = b \cdot a$ de rij van de termsgewijze producten, $c = (a_0 b_0, a_1 b_1, \ldots)$, en een τ-shift van $a = (a_0, a_1, a_2, \ldots)$ is $a^{(\tau)} = (a_\tau, a_{\tau+1}, a_{\tau+2}, \ldots)$. Vervolgens worden verschillende shifts van a met zichzelf vermenigvuldigd met het Hadamard product. Het aantal rijen dat wordt vermenigvuldigd levert de graad.
Een resultaat van Key [6] wat bewezen kan worden met de stelling van Blahut[2] geeft ons een bovengrens voor de lineaire span als we dit toepassen op m-rijen van lengte n. Als de output $c = c^{(1)} + c^{(2)} + \ldots + c^{(s)}$ is met $c^{(i)}$ een Hadamard product van graad $\leq d$ van shifts van a dan geldt dat

$$\lambda(c) \leq \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{d}$$



Figuur 4: Een Feedforward functie ofwel een niet-lineair filter op een LFSR

6.3.3 niet-lineaire samenvoeger

Ook kan er gebruikt worden gemaakt van meerdere LFSRs met output $a^j = (a_0^j, a_1^j, a_2^j, \ldots)$. Vervolgens kunnen we een functie kiezen met definitieve output c kiezen met $c_i = g(a_i^1, a_i^2, \ldots, a_i^k)$ waarbij de functie g niet lineair is. Zo kan een periode bereikt worden van ten hoogste de $kgv(n_i)$ waarbij n_i de periode is van a_i. Ook de lineaire span kan hoger worden. Een voorbeeld van $k = 3$ is de Geffe generator waarbij 3 LFSRs worden gecombineerd om een output $g(a^1, a^2, a^3) = a^1 a^3 + a^2(1 - a^3)$ te verkrijgen met lineaire span $\lambda(a^1)\lambda(a^3) + \lambda(a^2)\lambda(1 + \lambda(a^3))$ en periode $n_1 n_2 n_3$.

6.3.4 statistiek van de samenvoegers

Bij het samenvoegen van LFSRs werd ontdekt dat er stastistische zwakheden konden ontstaan in de output. Hiertoe opperde Rueppel als oplossing voor dit probleem om een geheugencel toe te voegen aan de samenvoeger. Hierbij werd de output verkregen uit als $c_i \equiv a_i + b_i + m_i \mod 2$ met $m_{i+1} = a_i + b_i + m_i(div\ 2)$. Echter was deze sleutelgenerator weer gevoelig voor FCSR-synthese aanvallen. De FCSRs zijn uitgevonden als middel bij de cryptoanalyse van sommatie samenvoegers.

6.3.5 klokgestuurde generatoren

Het is ook een mogelijkheid om meerdere LFSRs te pakken en de uitkomst van de ene LFSR te laten bepalen hoeveel klokslagen de andere LFSR moet doen voor hij zijn laatste output daadwerkelijk naar buiten brengt. Als de output van LFSR-1 t is, en LFSR-2 daardoor $f(t)$ klokslagen doet volgens een functie $f : F \to \mathbb{Z}$, dan heeft de output van de gehele generator een periode $\rho(c)|\rho_1 \cdot \rho_2$ en lineaire span $\lambda(c) \leq n_2\rho_1$ waarbij n_2 de lengte van LFSR-2 is. Dit resultaat is beter dan een gewone LFSR maar niet zo goed als gehoopt. Ook is het mogelijk om de LFSRs tegelijk klokslagen te laten maken, maar de output van LFSR-2 alleen te gebruiken als LFSR-1 een 1 als output geeft. Bij het gebruik van m-rijen met lengte n_1 en n_2 en periodes die relatief priem zijn wordt een lineaire span verkregen van

$$n_1 2^{n_2-2} < \lambda(c) \leq n_1 2^{n_2-1}$$

De output c heeft een bijna uniforme verdeling van subrijen met dezelfde lengte. Echter zijn er praktische problemen vanwege de onregelmatig ge-time-de output.

6.4 FCSR

Alvorens we gaan kijken hoe we register synthese aan gaan pakken bij FCSRs is het handig om wat bijpassende notatie over FCSRs door te lopen. Want ook hier is het bij de synthese van registers belangrijk om een goede afspraak te maken over hoe we de maat van een FCSR definiëren:

$$\lambda = \begin{cases} m + max(\lfloor \log_N(wt(q+1)) \rfloor, \lfloor \log_N |z| \rfloor + 1) \text{ als } z \neq 0, \\ m + \lfloor \log_N(wt(q+1)) \rfloor \text{ als } z = 0. \end{cases}$$

met $wt(q+1) = \sum_{i=1}^{m} q_i$. Deze constructie is tot stand gekomen omdat er voldoende maar toch het liefst zo min mogelijk ruimte gereserveerd wordt voor de geheugencel die z zal gaan onthouden. Als $z > wt(q+1)$ zal z in het verloop van de rekenstappen van de FCSR alleen nog maar dalen. Ook zal $z \leq wt(q+1)$ altijd zo blijven als dit eenmaal het geval is geweest. Zodoende reserveren we $max(\lfloor \log_N(wt(q+1)) \rfloor, \lfloor \log_N |z| \rfloor + 1)$ geheugen waarbij z de beginwaarde z_{m-1} representeert.

De lengte functie zoals hierboven is een getal die het aantal N-adische geheugencellen telt dat nodig is om de toestand te onthouden, terwijl er ook een andere maat is voor de FCSR: de complexiteit. De complexiteit is een reëel getal gebaseerd op de rationale representatie $\frac{f}{q}$ van het geassociëerde N-adische getal a waarbij $a = \frac{f}{q}$ met $ggd(f,q) = 1$. Deze complexiteit wordt als volgt berekend:

$$\phi_N(a) = \log_N(\Phi(f,q))$$

met $\Phi(f,q) = max(|f|,|q|)$. Als a strikt periodiek is zijn er $\lceil \phi_N(a) \rceil$ geheugencellen nodig voor het basis register (dus zonder de cel voor z). En verder blijken de complexiteit en de N-adische lengte ook dicht bij elkaar te liggen.

$$|\lambda_N(a) - \phi_N(a)| \leq \log_N(\phi_N(a)) + 2$$

Echter blijkt nu dat een periodieke rij a met lage complexiteit en lengte een kunstmatige hoge complexiteit kan krijgen door slechts een paar symbolen te veranderen. Dit geeft aan dat de complexiteit en lengte op zichzelf slechts een grove bovengrens is voor de cryptografische veiligheid van de rij a. Rueppel[5] suggereerde dat het complexiteitsprofiel een betere indicator is. Voor dit profiel gebruiken we $a = a_0, a_1, \ldots, a_{k-1}$

$$\psi(a) = \log_N(\min_{(f,q)} \Phi(f,q))$$

waar het minimum genomen wordt over alle $(f,q) \in \mathbb{Z}^2$ met $q \equiv -1 \mod N$ waarvoor geldt dat $\frac{f}{q} \equiv a \mod N^k$.
We krijgen zo een functie $\psi_a(k) : \mathbb{Z}_{>0} \to \mathbb{R}$. Een zeer willekeurige a zal een $\psi_a(k)$ hebben die ongeveer groeit als $k/2$. Een FCSR die op kunstmatige manier een hoge complexiteit heeft verkregen zal bij dit profiel een $\psi_a(k)$ hebben die lang rond zijn originele lagere complexiteit blijft hangen. Dit is een cryptologische zwakte die met dit profiel aan het licht komt.

Maximale complexiteitsorde

De maximale complexiteitsorde van een rij is de afmeting van de kleinste (mogelijk niet-lineaire) feedback shift register zonder geheugen (voor z) welke gebruikt kan worden om de rij te genereren. Hiermee wordt voor een rij (a_i) dus eigenlijk gekeken wat het de beste klasse van feedback shift registers is om de reeks in te reconstrueren. Hierbij is het relevant om te realiseren dat een FCSR als een niet-lineair feedback register zonder geheugen beschouwd kan worden met de N-adische lengte als het aantal cellen. Aangezien in dit verslag niet alle Feedback Shift Registers behandeld worden zal ik hier ook geen formule voor geven, maar het is voor de lezer goed om te beseffen dat er voor een rij (a_i) ook andere klassen Registers kunnen zijn waarin (a_i) een lagere complexiteit heeft en dus makkelijker gegenereerd kan worden. Vanuit cryptologisch oogpunt wil je zorgen dat zelfs het beste model moeite heeft om de rij (a_i) te kraken.

6.4.1 Symmetrische span

Bij het analyseren van feedback registers is het interessant om jezelf de vraag te stellen of het makkelijker is om het eerstvolgende element te voorspellen of het element dat vóór het eerste element kwam. Hiertoe definiëren we de rij a^{rev} die dezelfde coëfficiënten heeft, maar dan in omgekeerde volgorde. Bij de lineaire span blijkt te gelden dat $\lambda(a^{rev}) = \lambda(a)$, maar bij de N-adische span is dat niet het geval. Bij FCSR-synthese is het dus relevant om in beide richtingen te proberen om het probleem op te lossen. Hierdoor ontstaat de *symmetrische N-adische span* die gelijk is aan het minimum van de span van a en a^{rev}.

l-rijen

Net zoals de m-rijen die we bij de LFSRs zagen zijn, zijn de l-rijen bij FCSRs rijen met maximale periode. De l-rijen hebben de maximale periode die mogelijk is met verbindingsgetal q, namelijk $T = q - 1$. Deze periode wordt gehaald dan en slechts dan als N een primitieve wortel modulo q is. Dat betekent dat N^k alle waarden modulo q kan aannemen behalve 0. En van binaire l-rijen met $q = 2^t + e$ priem en $1 \leq e < 2^t$ weten we dat ieder patroon van t coëfficiënten/bits een of twee keer verschijnt in een periode van a. Hij verschijnt twee keer dan en slechts dan als $-ec(mod 2^t) < e$.

6.4.2 roosters en benaderingen

Bij FCSRs en het gebruik hiervan ontstaat dezelfde probleemstelling als bij LFSRs. Men wil een FCSR met zijn parameters en begintoestand kunnen reconstrueren aan de hand van zijn output. Het zou makkelijk zijn als een aangepaste versie van het Berlekamp-Massey algoritme zou volstaan om dit te doen. Dit blijkt echter niet te werken.
Een manier om het probleem van de FCSRs te benaderen gaat via het observeren van zijn benaderingen in een rooster. Dit is gevolg van de bevindingen van Mahler[4] en de Weger[3]. In deze benaderingswijze wordt gekeken naar $a = a_0 + a_1 N + \ldots \in \mathbb{Z}_N$ en zijn ke benaderingsrooster

$$L_k = L_k(a) = \{(h_1, h_2) \in \mathbb{Z}^2 : ah_2 - h_1 \equiv 0 \mod N^k\}$$

Merk hierbij op dat $ah_2 - h_1 \equiv 0 \mod N^k$ hetzelfde is als $a \equiv \frac{h_1}{h_2} \mod N^k$ zolang $ggd(N, h_2) = 1$ geldt. Deze herschrijving is nodig om L_k een lineaire ruimte te maken. De verkregen verzameling is een compleet rooster met volume N^k. Verder geldt dat $L_{k+1} \subset L_k$.
Ook is $\{((a \mod N^k), 1), (N^k, 0)\}$ een basis van L_k. Net zoals bij het complexiteitsprofiel zijn we op zoek naar de kleinste complexiteit waarvoor we een (h_1, h_2) kunnen vinden zodat
$a \cdot h_2 - h_1 \equiv 0 \mod N^k$ geldt. Aangezien zo'n set bestaat moet hij te verkrijgen zijn uit de door ons bekende basis $\{((a \mod N^k), 1), (N^k, 0)\}$ en iedere andere basis die we hiermee kunnen construeren. We gaan dus op zoek naar de kleinste basis voor L_k ten opzichte van de functie $\Phi(f, q)$.
Een paar elementen $u, v \in L$ is een paar opeenvolgende minima, ofwel een minimale basis, als

$$\Phi(u) \leq \Phi(u\prime) \forall u\prime \in L \backslash \{(0,0)\}$$
$$\Phi(v) \leq \Phi(v\prime) \forall v\prime \in L \backslash u\mathbb{Z}$$

Een element noemen we positief als $u_1 u_2 > 0$ en negatief als $u_1 u_2 < 0$. Een paar opeenvolgende minima kan niet uit twee positieve elementen bestaan en ook niet uit twee negatieve. Ook kan het volume worden berekend door de determinant te berekenen van de twee basisvectoren.
De volgende stelling helpt ons met het herkennen van opeenvolgende minima voor een rooster $L \subset \mathbb{R}^2$, wat ook wel een minimale basis wordt genoemd:
Als we een $x \in L \backslash \{(0,0)\}$ vinden met $x < \sqrt{Vol(L)/2}$ Dan is x een geheel veelvoud van u. En als x en y lineair onafhankelijk zijn met $\Phi(x) < \sqrt{Vol(L)}$ en $\Phi(y) < \sqrt{Vol(L)}$, dan zijn x en y opeenvolgende minima voor L. En ieder minimale basis is dezelfde op eventuele wijziging van teken en volgorde na.

De rij a ligt in zijn geheel vast na de eerste k symbolen met

$$k = \begin{cases} 2\lceil \phi_N(a) \rceil + 1 \text{ if } N > 2 \\ 2\lceil \phi_2(a) \rceil + 2 \text{ if } N = 2 \end{cases}$$

rationale benaderingen

Euclides

Bij het zoeken naar een minimale basis voor het rooster komt het uitgebreide algoritme van Euclides in de vorm van Appendix B goed van pas. De algoritme geeft niet altijd een basis als output. Als hij geen output geeft betekent dit dat de algoritme te weinig input heeft gekregen om "zeker"te zijn van de berekende gegevens. Als de algoritme wél een output (f, q) geeft, geeft het een breuk weer die N-adisch gezien overeenkomt met de input. Deze breuk hoeft echter geen FCSR te representeren omdat aan de vereiste $q \equiv -1 \mod N$ niet per se wordt voldaan. De breuk (f, q) kan echter wel herschreven worden naar (uf, uq) met met $uq \equiv -1 \mod N$. Dit verandert de uitkomst van de breuk niet, maar de Φ van de breuk wel. Omdat de algoritme van Euclides niet meteen "door heeft"dat het om een FCSR gaat kan het dus zijn dat de algoritme in Appendix B meer elementen van de rij (a_i) nodig heeft dan de hierboven gegeven van k suggereert. In Appendix B vind je ook een voorbeeld waarin je kunt zien dat k elementen meegeven niet genoeg hoeft te zijn voor de algoritme om een basis als output te geven. k is immers enkel een theoretisch minimum hiervoor.

andere algoritme

Omdat we in de huidige probleemstelling wel zeker zijn dat het een FCSR betreft zijn er aangepaste varianten van deze algoritme die minder input nodig hebben dan het gebruikte Euclides algoritme. Er zijn meerdere varianten die onderling niet zo veel verschillen dus heb ik er een gekozen. Deze algoritme is analoog aan de Berlekamp-Massey algoritme in de zin dat ze aan beide optimalteitseigenschappen voldoen: hij genereert de kleinste FCSR die (a_i) genereert en doet dat door gebruik te maken van enkel de eerst $2\phi_2(a) + 2\log_2(\phi_2(a))$ elementen van (a_i). De werking is gebaseerd op 2-adische approximatietheorie en is aan aanpassing op de procedure omschreven door de Weger[3] en Mahler[4]. De algoritme heeft het voordeel dat het net als de Berlekamp-Massey algoritme verder kan rekenen met extra input (meer a_i's gegeven) dan waar het in eerste instantie mee begon zonder opnieuw te beginnen.

7 conclusie

De oorspronkelijke opzet van dit project was om uit te zoeken wat het verband was tussen p-adische getallen en LFSRs. Twee onderwerpen waarover ik beide in eerste instantie niets wist. Na vele inzichten over zowel de p-adische getallen en LFSRs kan ik stellen dat de p-adische getallen een nuttig middel zijn in de theorie over de FCSRs, wat een aanpassing is op de LFSRs. Waar bij LFSRs gerekend wordt met representanten in de wereld van machtreeksen en polynomen met coëfficiënten in $\mathbb{F}_{\ltimes}$, wordt er bij FCSRs gerekend met p-adische getallen omdat dit getallenstelsel perfect aansluit bij de werking van FCSRs. Het verschil tussen de polynomen en p-adische getallen sluit goed aan op het verschil tussen LFSRs en FCSRs. Echter hebben p-adische getallen meer eigenschappen die in dit verslag niet aan de orde komen. Bij mijn onderzoek naar FCSRs lijken deze overige eigenschappen echter niet heel relevant. Naast het gebruik bij FCSRs zullen p-adische getallen dan ook nog meer toepassingen kennen waarbij wellicht een ander licht op p-adische getallen dient te worden voor men ze nuttig kan gebruiken. De FCSRs zijn ontwikkeld als aanpassing op de LFSRs in een poging om de output minder voorspelbaar te maken. In dit verslag wordt gekeken naar manieren om een FCSR te construeren aan de hand van zijn output wat ons vertelt dat ook FCSRs nog onvoorspelbaarder zouden mogen.

Een eventueel vervolg

Mocht iemand dit project willen voortzetten zou het een goede uitdaging zijn om uit te zoeken wat voor aanpassingen gedaan kunnen worden aan FCSRs om een nog minder voorspelbare rij getallen te kunnen genereren. Hiertoe adviseer ik eerst het boek Algebraic Shift Register Sequences[2] verder te raadplegen waarna meer literatuur over dit onderwerp gezocht kan worden. Een andere optie is om zelf te experimenteren met eigen verzonnen aanpassingen op de FCSRs waarvan de N-adische span onderzocht kan worden met de gegeven algoritmes. Hierin is het doel om een zo hoog mogelijke maximale complexiteitsorde te verkrijgen met een systeem dat zo min mogelijk geheugen en operaties per klokslag nodig heeft.

Verder kan men zelf experimenten welke invloeden de gedaan kunnen worden op eigen ontworpen varianten

8 Appendix A

Data: $a_0, a_1, \ldots$ tot en met a_k
Result: Een (f_i, q_i) zodanig dat $f_i(x) \equiv a(x) \cdot q_i(x) \mod x^{k+1}$
initialization;
if *all* $a_i = 0$ **then**
| return $(0, 1)$
end
$a(x) = \sum_{i=0}^{n-1} a_i x^i$
Laat m minimaal zijn met $a_m \neq 0$
$f_m(x) = 0$
$q_m(x) = 1$
$f_{m+1}(x) = a_m x^m$
$q_{m+1}(x) = \begin{cases} 1 + x^m & \text{if } m > 0 \\ 1 & \text{anders.} \end{cases}$ $c = a_m$
if $b = 1$ **then**
| $b = 1$
else
| $b = 2$
end
for $i = m + 1$ **to** $n - 1$ **do**
test
Let $a(x)q_i(x) - f_i(x) \equiv bx^i \mod x^{i+1}$
if $b = 0$ **then**
| $f_{i+1}(x) = f_i(x)$
| $q_{i+1}(x) = q_i(x)$
else
| $f_{i+1}(x) = f_i(x) - (b/c)x^{i-m} f_m(x)$
| $q_{i+1}(x) = q_i(x) - (b/c)x^{i-m} q_m(x)$
| **if** $\Phi(f_{i+1}, q_{i+1}) > \Phi(f_i, q_i)$ **then**
| | $m = i$
| | $c = b$
| **end**
end
$i = i + 1$
;
return(f_n, q_n)

9 Appendix B

Data: $a_0, a_1, \ldots$ tot en met a_{k-1}
Result: Een (f, q) zodanig dat $a \cdot q \equiv f \mod N^k$
initialization;
if *k is not odd* **then**
  k=k-1
end
$(r_0, x_0, y_0) = (N^k, 1, 0)$
$(r_1, x_1, y_1) = \sum_{i=0}^{k-1} a_i N^i, 0, 1)$
while $r_1 > N^{k/2}$ **do**
  Let $q = \lfloor \frac{r_0}{r_1} \rfloor$
  and $r = r_0 - qr_1$
  such that $r_0 = qr_1 + r$ now holds
  $(x_3, y_3) = (x_0 - qx_1, y_0 - qy_1)$
  $(r_0, x_0, y_0) = (r_1, x_1, y_1)$
  $(r_1, x_1, y_1) = (r, x_3, y_3)$
end
if $|y_1| \leq 2^{k/2}$ **then**
  return (r_1, y_1)
else
  return False
end

voorbeeld

Bij de algoritme wordt de inhoud van de set (r_1, x_1, y_1) steeds doorgegeven aan (r_0, x_0, y_0). Dus in onderstaande notatie schrijf ik enkel steeds de nieuwe (r_1, x_1, y_1) op zodat ik vervolgens met de twee meest-recente toestanden verder reken. Ik neem hierbij hetzelfde voorbeeld als in hoofdstuk 5.2. Hierbij bekijken we de 5-adische getallen en hebben we te maken met $(a_i) = (1, 0, 2, 0, 4, 0, 3, \ldots)$

$$
\begin{aligned}
input = (1,0,2,0,4,0,3) & \\
(N^7, 1, 0) =& (78125, 1, 0) \\
(\sum_{i=0}^{k-1} a_i N^i, 0, 1) =& (49426, 0, 1) \\
& (28699, 1, -1) \\
& (20727, -1, 2) \\
& (7972, 2, -3) \\
& (4783, -5, 8) \\
& (3189, 7, -11) \\
& (1594, -12, 19) \\
& (1, 31, -49)
\end{aligned}
$$

Vervolgens wordt de while loop gestopt en wordt er False gereturned omdat
$|y_1| = 49 \not\leq 11,3 \approx 2^{k/2}$. Anders zou $(r_1, y_1) = (1, -49)$ de output zijn geweest.
De expansie van $\frac{1}{-49}$ bestaat 5-adisch gezien uit de machtreeks met de volgende rij coëfficiënten:

$$(1,0,2,0,4,0,3,1,1,3,2,1,0,3,0,1,1,2,2,4,...)$$

Hoewel dit overeenkomt met de input van de algoritme, geldt voor $\frac{1}{-49}$ echter niet dat de noemer equivalent is aan $-1 \mod N$. Nu zal de breuk $\frac{-1}{49}$ wel voor de hand liggen als aanpassing, maar die logica zit niet ingebakken in de algoritme.

10 Appendix C

Data: $a_0, a_1, \ldots$ tot en met a_{T-1}
Result: Een (f, q) zodanig dat $a \cdot q \equiv f \mod 2^T$
initialization;
$a = \sum_{i=0}^{T-1} a_i 2^i$
Let t be minimal with $a_{t-1} = 1$
$f = (0, 2)$
$g = (2^{t-1}, 1)$
for $k = t$ **to** $T - 1$ **do**
  if $a \cdot g_2 - g_1 \equiv 0 \mod 2^{k+1}$ **then**
    if $\Phi(f) < \Phi(g)$ **then**
      $f = 2f$
    else
      Let d minimize $\Phi(f + dg)$
      $(g, f) = (g, 2(f + 2dg))$
    end
  else
    if $\Phi(g) < \Phi(f)$ **then**
      Let d minimize $\Phi(f + dg)$ with d odd
      $(g, f) = (f + dg, 2g)$
    else
      Let d minimize $\Phi(g + df)$ with d odd
      $(g, f) = (g + df, 2f)$
    end
  end
end
return g

Voorbeeld

Ook bij deze algoritme geef ik een voorbeeld: We nemen de 2-adische rij coëfficiënten $(1,0,0,1,1,1,0,0,0,1)$ als input en kijken wat er gebeurt.

$$a = 569$$

$t = 1$ want $a_0 = 1$

$$f = (0,2)$$
$$g = (2^{t-1},1) = (1,1)$$

En vanaf hier zal ik (k,d,g,f) noemen waarbij k de iteratie is, d zoals berekend en g en f het resultaat zijn van de iteratie

$$(1,-1,\begin{pmatrix}1\\1\end{pmatrix},\begin{pmatrix}-2\\2\end{pmatrix})$$
$$(2,0,\begin{pmatrix}1\\1\end{pmatrix},\begin{pmatrix}-4\\4\end{pmatrix})$$
$$(3,1,\begin{pmatrix}-3\\5\end{pmatrix},\begin{pmatrix}2\\2\end{pmatrix})$$
$$(4,nvt,\begin{pmatrix}-3\\5\end{pmatrix},\begin{pmatrix}4\\4\end{pmatrix})$$
$$(5,-1,\begin{pmatrix}-7\\1\end{pmatrix},\begin{pmatrix}8\\8\end{pmatrix})$$
$$(6,1,\begin{pmatrix}1\\9\end{pmatrix},\begin{pmatrix}-14\\2\end{pmatrix})$$
$$(7,1,\begin{pmatrix}1\\9\end{pmatrix},\begin{pmatrix}-13\\11\end{pmatrix})$$
$$(8,0,\begin{pmatrix}1\\9\end{pmatrix},\begin{pmatrix}-13\\11\end{pmatrix})$$
$$(9,0,\begin{pmatrix}1\\9\end{pmatrix},\begin{pmatrix}-13\\11\end{pmatrix})$$

Het algoritme zal $g = (1,9)$ als output geven wat vervolgens weer overeenkomt met een FCSR met de volgende output:

$$(1,0,0,1,1,1,0,0,0,1,1,1,0,0,0,1,1,1,\ldots)$$

Merk op dat dit overeenkomt met de input van de algoritme en dat we dit algoritme kunnen hervatten na het toevoegen van een $a_1 0$ en herberekenen van a vanaf hier.

Referenties

[1] Fernando Q. Gouvea, *P-adic numbers: an introduction*, Berlin : Springer (2000)

[2] Mark Goresky, Andrew Klapper, *Algebraic Shift Register Sequences* Cambridge University Press (2012), (p. 161,304)

[3] B. M. M. de Weger, *Approximation lattices of p-adic numbers*, Journal of Number Theory 24 (1986), $70 - 88$

[4] K. Mahler, *On a geometrical representation of p-adic numbers*, Ann. of Math. 41 (1940), $8-56$.

[5] R. Rueppel, *Analysis and Design of Stream Ciphers*, Berlin: SpringerVerlag, (1986).

[6] E. Key, *An analysis of the structure and complexity of nonlinear binary sequence generators*, Trans. Info. Theory, IT22 (1976), 7326.