Inhoud

Samenvatting

In deze scriptie wil ik een algoritme beschrijven om de ring van gehelen in een algebraïsch functielichaam te berekenen. Als het algebraïsch functielichaam een eindige uitbreiding van $L(x)$ is, zoek ik de elementen daarin, die geheel zijn over $L[x]$.

Ik heb naar aanleiding van het college algebraïsche getaltheorie een algoritme geschreven dat onder andere de ring van gehelen in een algebraïsch getallenlichaam berekent. Het algoritme dat in deze scriptie beschreven wordt, heb ik geschreven nadat Prof. Levelt mij gevraagd had, of de methode voor het algebraïsch getallenlichaam aangepast kon worden voor het geval van een algebraïsch functielichaam. Het resultaat ziet U hier.

Vlak voor het beëindigen van deze scriptie heb ik in het proefschrift van B.M. Trager een ander algoritme gezien. Voor de volledigheid heb ik zijn methode kort beschreven in paragraaf 6. De vergelijking met Trager's algoritme blijkt afhankelijk te zijn van het gekozen voorbeeld. In sommige gevallen is het sneller, en in andere gevallen langzamer.

Berekening van de ring van gehelen in een algebraïsch functielichaam

Mark van Hoeij

november 1992

Zij $R \subset S$ een ringuitbreiding. Dan zijn de gehelen in S over R die elementen van S die nulpunt zijn van een monische veelterm over R. De verzameling $\overline{R}$ van elementen in S die geheel zijn over R heet de gehele afsluiting van R in S. $\overline{R}$ is een deelring van S.

Berekening van een ring van gehelen is nodig bij elementaire integratie [3]. Ik wil hier een algoritme aangeven om zo'n ring van gehelen te berekenen. Ik bekijk in deze scriptie de volgende situatie:

- L is een algebraïsch gesloten lichaam van karakteristiek 0.
- x is transcendent over L.
- y is algebraïsch over $L(x)$ met minimumveelterm f. Door y eventueel met een element van $L[x]$ te vermenigvuldigen mag ik aannemen dat y geheel is over $L[x]$, oftewel dat de minimumveelterm f van y over $L(x)$ zijn coëfficiënten in $L[x]$ heeft.
- n is de graad van f, dus $L(x,y)$ is een $L(x)$-vectorruimte van dimensie n.
- Gezocht is nu $\overline{L[x]}$, de gehele afsluiting van $L[x]$ in het algebraïsch functielichaam $L(x,y)$.

Ik zal in deze scriptie een algoritme geven om $\overline{L[x]}$ te bepalen.

1 Inleidende theorie

Deze paragraaf dient als geheugenopfrisser voor de benodigde voorkennis. In paragraaf 1.1 geef ik wat theorie die analoog is met de algebraïsche getaltheorie. Alleen dan met $\mathbb{Z}$ vervangen door $L[x]$, $\mathbb{Q}$ vervangen door $L(x)$, en het algebraïsch getallenlichaam vervangen door het algebraïsch functielichaam $L(x,y)$. De definities en bewijzen kunnen met deze vervangingen letterlijk overgenomen worden uit de algebraïsche getaltheorie. In paragraaf 1.2 kijk ik naar ringen van gehelen vanuit de theorie van de algebraïsche krommen.

Niet alle theorie in deze paragraaf is nodig voor het algoritme dat ik zal geven. Het is voor een goed begrip van de situatie echter wel nuttig om deze theorie in het achterhoofd te houden bij het lezen van de andere paragrafen.

1.1 Enkele definities

In deze paragraaf zal ik enkele benodigde definities geven en bewijzen (stelling 2) dat $\overline{L[x]}$ als $L[x]$-moduul isomorf is met $(L[x])^n$. Het bewijs dat de ring van gehelen in een algebraïsch getallenlichaam (zie [1]) als $\mathbb{Z}$-moduul isomorf is met $\mathbb{Z}^n$ kan eenvoudig omgezet worden in een bewijs dat $\overline{L[x]}$ als $L[x]$-moduul isomorf is met $(L[x])^n$. Ik geef hier echter een ander bewijs van stelling 2, dat tevens illustreert hoe het algoritme dat ik verderop zal geven ongeveer werkt.

Notatie: Als $b = \{b_1, \ldots, b_N\}$ een verzameling elementen van $L(x,y)$ is dan bedoel ik met $L[x]b$ het door de elementen van b voortgebrachte $L[x]$-moduul.

Definitie 1 *Een* rooster *in $L(x,y)$ is een vrij $L[x]$-moduul $M \subset L(x,y)$ van rang n.*

Stel $b = \{b_1, \ldots, b_n\}$ is een verzameling van n $L(x)$-lineair onafhankelijke elementen. Dan is $L[x]b$ een rooster. Deze definitie is analoog met de definitie van een rooster in een algebraïsch getallenlichaam.

Stelling 1 *Elk eindig voortgebracht $L[x]$-moduul $M \subset L(x,y)$ is een vrij moduul met hoogstens n voortbrengers, dus rang $\leq n$.*

Bewijs: Stel $L(x)M \subset L(x,y)$ heeft als $L(x)$-vectorruimte dimensie N, d.w.z. er zijn precies N $L(x)$-lineair onafhankelijke elementen in M. Dan $N \leq n$. Merk op: als $L[x,y] \subset M \subset \overline{L[x]}$, wat voor de modulen die ik in deze scriptie bekijk steeds zal gelden, dan heb je de n $L(x)$-lineair onafhankelijke elementen $1, y, \ldots, y^{n-1}$ in M. Dan geldt dus $N = n$, oftewel M is een rooster.

Ik zal bewijzen dat M precies N vrije voortbrengers als $L[x]$-moduul heeft. Minder dan N kan niet omdat er N $L(x)$-lineair onafhankelijke elementen in M zitten. Die zijn dan ook over $L[x]$ onafhankelijk. Merk ook op dat als je precies N voortbrengers hebt, dan moeten het wel vrije voortbrengers zijn omdat die voortbrengers al over $L(x)$ onafhankelijk zijn.

Stel M wordt als $L[x]$-moduul voortgebracht door $b = \{b_1, \ldots, b_K\}$, met K minimaal en $K > N$. Er moeten N $L(x)$-lineair onafhankelijke elementen in b zitten. Ik mag dus aannemen dat $b_1, \ldots, b_N$ $L(x)$-lineair onafhankelijk zijn, en $L(x)M$ voortbrengen. Daarom zijn er a_{ij} en k_i in $L[x]$, $i = N+1, \ldots, K$ en $j = 1, \ldots, N$ zó dat

$$b_i = \frac{a_{i1}b_1 + \ldots + a_{iN}b_N}{k_i}$$

Omdat de graden van k_i eindig zijn zie je dat

$$M/(L[x]\{b_1, \ldots, b_N\}) \tag{1}$$

een L-vectorruimte van eindige dimensie is. Kies nu $b_1, \ldots, b_K$ zó dat deze dimensie minimaal is. Ik zal nu gaan bewijzen dat deze dimensie 0 is, oftewel, dat M dan door $b_1, \ldots, b_N$ wordt voortgebracht als L-moduul.

Ik ga nu eenzelfde redenering houden als ik in een van de punten van stelling 6 ook zal doen. Daar zal ik dat iets uitgebreider doen dan hier.

Stel de dimensie van (1) is niet nul. Dan is er een element $a = (a_1b_1 + \ldots + a_Nb_N)/k$ in $M \setminus (L[x]\{b_1, \ldots, b_N\})$, met a_i en k in $L[x]$. Dan mag ik door eventueel a met een element van $L[x]$ te vermenigvuldigen aannemen dat k irreducibel is in $L[x]$, en dat nog steeds $a \in M \setminus (L[x]\{b_1, \ldots, b_N\})$. Dus niet alle $a_i \equiv 0 \bmod k$. Kies i zo dat $a_i \not\equiv 0 \bmod k$. Door a met een element van $L[x]$ te vermenigvuldigen mag ik aannemen dat $a_i \equiv 1 \bmod k$. Door vervolgens bij a een b_i-voud op te tellen mag ik aannemen dat $a_i = 1$. Dan is nog steeds $a \in M \setminus (L[x]\{b_1, \ldots, b_N\})$. Zij nu $M' := L[x]\{b_1, \ldots, b_{i-1}, a, b_{i+1}, \ldots, b_N\}$. Dan zie je dat $b_i \in M'$ en dus $M \subset M'$. $M \neq M'$ omdat $a \notin M$. Dus de dimensie van (1) is niet minimaal, want de dimensie van M/M' is kleiner. Tegenspraak.

□

Voor de volgende definitie van de discriminant, en de beweringen die ik over de discriminant doe, is kennis van algebraïsche getaltheorie nodig. Zie daarvoor [1], bladzijde 19 tot en met 38. Ik zal die theorie hier niet herhalen omdat het daar beter uitgelegd wordt dan ik dat kan. Ook zal ik de beweringen over de discriminant niet bewijzen, de bewijzen zijn precies hetzelfde als in het geval van een algebraïsch getallenlichaam.

Definitie 2 *Zij $b = \{b_1, \ldots, b_n\}$ een verzameling van n elementen uit $L(x, y)$. De* discriminant *disc(b) van b is dan de determinant van de matrix*

$$(\text{spoor}(b_i b_j))_{1 \leq i,j \leq n}$$

waarbij het spoor genomen wordt over de lichaamsuitbreiding $L(x, y) : L(x)$.

Voor de discriminant gelden een aantal eigenschappen:

- De discriminant is een element van $L(x)$.
- Als $b_1, \ldots, b_n$ geheel zijn over $L[x]$ dan is de discriminant een element van $L[x]$.
- De discriminant van $\{1, y, y^2, \ldots, y^{n-1}\}$ is gelijk aan de discriminant van f als veelterm in y.
- Als de vector $(c_1, \ldots, c_n)$ uit de vector $(b_1, \ldots, b_n)$ ontstaat door vermenigvuldiging met een overgangsmatrix C met coëfficiënten in $L(x)$ dan is

$$\text{disc}(c_1, \ldots, c_n) = \text{disc}(b_1, \ldots, b_n)(\det C)^2 \tag{2}$$

Uit het laatste kun je de volgende feiten afleiden:

- De discriminant is ongelijk aan nul als $\{b_1, \ldots, b_n\}$ een basis van $L(x, y)$ als $L(x)$-vectorruimte is. Dit kun je inzien als je weet dat de discriminant van $\{1, y, y^2, \ldots, y^{n-1}\}$ niet nul is.
- Als $b_1, \ldots, b_n$ en $c_1, \ldots, c_n$ geheel zijn over $L[x]$ en het door $\{b_1, \ldots, b_n\}$ voortgebrachte $L[x]$-moduul $L[x]\{b_1, \ldots, b_n\}$ een deelmoduul is van $L[x]\{c_1, \ldots, c_n\}$, dan geldt dat $\text{disc}(c_1, \ldots, c_n)$ een deler van $\text{disc}(b_1, \ldots, b_n)$ in $L[x]$ is.

Als $b_1, \ldots, b_n$ en $c_1, \ldots, c_n$ hetzelfde moduul M voortbrengen, dan moet de overgangsmatrix C over $L[x]$ inverteerbaar zijn, en dus $\det C \in L$. De discriminanten van $b_1, \ldots, b_n$ en $c_1, \ldots, c_n$ verschillen dus een factor uit L. Deze discriminanten zijn ongelijk aan 0 als $b_1, \ldots, b_n$ $L(x)$-lineair onafhankelijk zijn, oftewel als M een rooster is. Je kunt dus de discriminant van een rooster $M \subset \overline{L[x]}$ als volgt definiëren:

Definitie 3 *Zij $M \subset \overline{L[x]}$ een rooster. Kies vrije voortbrengers $b_1, \ldots, b_n$ van M, dan $d :=$disc$(\{b_1, \ldots, b_n\}) \in L[x]$. Dan is de* discriminant *van M gelijk aan d gedeeld door de kopterm van d.*

De discriminant van M is dus gelijk aan de discriminant van de voortbrengers, maar dan monisch gemaakt. Dit monisch maken dient om een eenduidige definitie krijgen. Uit de eigenschappen van de discriminant van een verzameling $\{b_1, \ldots, b_n\}$ volgen nu:

Lemma 1 *Stel M is een rooster en $M \subset \overline{L[x]}$, dan is* $\mathrm{disc}(M) \in L[x]$.

Lemma 2 *Stel M_1 en M_2 zijn door n elementen voortgebrachte $L[x]$-modulen, $M_1 \subset M_2$, en $M_1 \neq M_2$. Stel M_1 en M_2 zijn deelmodulen van $\overline{L[x]}$. Dan is:*

$$\mathrm{disc}(M_2)|\mathrm{disc}(M_1)$$

en $\mathrm{disc}(M_1)/\mathrm{disc}(M_2)$ *is een kwadraat in $L[x]$.*

Bewijs: Laat $\{c_1, \ldots, c_n\}$ $L[x]$-voortbrengers van M_1 zijn, en $\{b_1, \ldots, b_n\}$ voortbrengers van M_2, en zij C de overgangsmatrix tussen $(b_1, \ldots, b_n)$ en $(c_1, \ldots, c_n)$. Dan zegt vergelijking 2:

$$\mathrm{disc}(M_1) = \mathrm{disc}(M_2)(\det C)^2$$

Uit $M_1 \neq M_2$ volgt dat de overgansmatrix C niet inverteerbaar over $L[x]$, dus $\det C \notin L$. Dus $\mathrm{disc}(M_2)$ is een echte deler van $\mathrm{disc}(M_1)$.

□

Stelling 2 *De ring van gehelen $\overline{L[x]}$ is een rooster en dus Noethers.*

Bewijs: Zij $M_1 := L[x, y] \subset \overline{L[x]}$. M_1 is een rooster. Stel $M_1 \neq \overline{L[x]}$ oftewel de dimensie van $\overline{L[x]}/M_1$ als L-vectorruimte is niet nul. Dan kun je volgens het bewijs van stelling 1 een rooster $M_2 \neq M_1$ (daar heet M_2 echter M') zo dat $M_1 \subset M_2 \subset \overline{L[x]}$. Als nu weer dimensie$(\overline{L[x]}/M_2) \neq 0$ dan kun je een rooster M_3 vinden, etc.

Merk nu op dat telkens de discriminant van M_{i+1} een strikte deler van de discriminant van M_i is. Dus moet voor zekere i gelden dat $\overline{L[x]}$ gelijk aan het rooster M_i is. Dus $\overline{L[x]}$ is Noethers.

□

Definitie 4 *Zij $L_1 : L_2$ een algebraïsche lichaamsuitbreiding en zij $a \in L_1$. Dan is vermenigvuldiging met a in L_1 een L_2-lineaire afbeelding. De* karakteristieke veelterm *van a over de lichaamsuitbreiding $L_1 : L_2$ is nu gedefiniëerd als de karakteristieke veelterm van deze lineaire afbeelding, in een of andere variabele.*

Als nu R een deelring van L_2 is, $a \in L_1$ en g is de karakteristieke veelterm van a in de dummy-variabele z dan geldt:

$$g(z) \in R[z] \iff a \text{ is geheel over } R$$

1.2 Een stukje algebraïsche meetkunde

De algebraïsche meetkunde biedt een heel ander gezichtspunt op ringen van gehelen dan de algebraïsche getaltheorie. Ik zal voor het algoritme niet veel theorie van algebraïsche krommen gebruiken, maar zal in deze paragraaf toch een paar feiten daarover opsommen omdat die nuttig zijn om in gedachten te houden bij de rest van de scriptie. Voor bewijzen en meer theorie over algebraïsch krommen verwijs ik naar [2].

De nulpunts-kromme $C(f)$ van f in het affiene L-vlak is een algebraïsche kromme. Dit affiene vlak V, met x en y als coördinaten, kun je opvatten als deel van een projectief vlak. $C(f)$ zijn precies die punten (x, y) in V waarvan het paar (x, y) aan f voldoet. Met $C(f)$ bedoel ik dus alleen het affiene deel van de algebraïsche kromme. $\overline{L[x]}$ bestaat uit precies die algebraïsche functies in $L(x, y)$ die geen polen hebben in $C(f)$. Nu is:

$$\overline{L[x]} = \bigcap_{O \in X_L \text{ boven } V} O$$

en

$$L[x, y] = \bigcap_{P \in C(f)} O_P$$

waar O_P de lokale ring in P is, en X_L de verzameling plaatsen (= discrete waarderingsringen) van L is.

Als $C(f)$ geen singulariteiten in V heeft, dan zijn deze twee doorsneden hetzelfde omdat dan de lokale ringen in de punten van $V \cap C(f)$, en de plaatsen $O \in X_L$ boven (met centrum in) V hetzelfde zijn. Dan geldt dus $\overline{L[x]} = L[x, y]$. De verzameling $S(f)$ van singulariteiten in $V \cap C(f)$ is precies de nulpuntsverzameling van $\{f, \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\}$. Als $S(f) = \emptyset$ dan is de ring van gehelen dus $L[x, y]$.

Stel $g_1, \ldots, g_k \in \overline{L[x]}$. Zoals f een kromme beschrijft in het affiene x-y-vlak, zo ook beschrijft f een kromme $C_R(f)$ in de affiene ruimte R die $x, y, g_1, \ldots, g_k$ als coördinaten heeft. Nu geldt ook weer

$$\overline{L[x]} = \bigcap_{O \in X_L \text{ boven } R} O$$

en

$$L[x, y, g_1, \ldots, g_k] = \bigcap_{P \in R \cap C_R(f)} O_P$$

Ook hier zie je dat $\overline{L[x]} = L[x, y, g_1, \ldots, g_k]$ als f geen singulariteiten heeft in R. Het vinden van $\overline{L[x]}$ is dus equivalent met het zodanig uitbreiden van V dat de kromme geen singulariteiten meer heeft.

Opmerking: je kunt in principe zolang als $L[x, y, g_1, \ldots, g_{k-1}] \neq \overline{L[x]}$ met het opblaasproces een gehele $g_k \in \overline{L[x]} \setminus L[x, y, g_1, \ldots, g_{k-1}]$ vinden. Met opblazen bedoel ik hier het volgende: je neemt een singulariteit, en kiest 'willekeurig' twee functies die in die singulariteit een nulpunt van de orde 1 hebben. Dan neem je g_k als het quotiënt van die twee functies, eventueel vermenigvuldigd met een element van $L[x]$ om te zorgen dat je een gehele krijgt.

Als nu g_k al in $L[x, y, g_1, \ldots, g_{k-1}]$ zit dan kies je twee andere functies die een nulpunt van orde 1 hebben.

Het opblazen wordt echter wel steeds reken-intensiever naarmate de dimensie van R stijgt. Wel kun je denk ik, als je het bovenstaande wat nauwkeuriger uitwerkt, op deze manier bewijzen dat het opblaasproces een eindig proces is. Je krijgt immers een strikt stijgende rij $L[x]$-modulen $L[x, y], L[x, y, g_1], L[x, y, g_1, g_2], \ldots$ binnen het Noetherse $L[x]$-moduul $\overline{L[x]}$, en zo'n rij moet eindigen.

Het was oorspronkelijk mijn bedoeling om ook een algoritme voor de ring van gehelen te schrijven dat zich richt op het wegwerken van singulariteiten. Het idee was om een rij

$$L[x, y] \subset L[x, y, z_1] \subset L[x, y, z_2] \cdots \subset L[x, y, z_k] = \overline{L[x]}$$

te construeren, zó dat $C(f)$ in de bijbehorende affiene ruimten telkens 'minder singulier' zou zijn. Dit is mede door tijdgebrek niet meer gelukt. Uit deze constructie zou tevens de volgende in de meetkunde bekende stelling volgen.

Stelling 3 *Er bestaat een $z \in \overline{L[x]}$ zó dat $\overline{L[x]} = L[x, y, z]$.*

Bewijs: We weten al dat $\overline{L[x]}$ als $L[x]$-moduul eindig voortgebracht is, zeg door $b_1, \ldots, b_n$. In de keuze van z beperk ik me nu tot L-lineaire combinaties van $b_1, \ldots, b_n$, zeg $b = a_1 b_1 + \ldots + a_n b_n$. Nu leg ik de volgende eisen op aan z:

1. z moet in alle plaatsen die hun centrum hebben in $S(f)$ (de verzameling singulariteiten van $C(f)$ in V) verschillende waarden hebben. Merk op dat alle nog mogelijke z geheel zijn, en dus in het affiene vlak V geen pool hebben. De waarden die z in de plaatsen met centrum in $S(f)$ aanneemt zijn dus elementen van L.
2. Als z de waarde w aanneemt in een plaats met centrum in $S(f)$ dan moet $z - w$ valuatie 1 hebben in die plaats.

Deze eisen zijn lineaire ongelijkheden voor de a_i's, en wel eindig veel lineaire ongelijkheden. Elke lineaire ongelijkheid sluit een deelruimte van dimensie 1 aan mogelijkheden voor z uit. Als nu $n > 1$ dan blijven er dus nog mogelijkheden over. Kies z als een van die mogelijkheden.

Zij R de affiene ruimte met x, y en z als coördinaten. Stel dat $C_R(f)$ nog singulariteiten in R heeft. Dat kan alleen als verschillende plaatsen hetzelfde centrum hebben, maar dat wordt door de eerste verzameling eisen uitgesloten, of als een plaats vertakt, maar dat wordt uitgesloten door de tweede verzameling eisen.

Conclusie: $C_R(f)$ heeft geen singulariteiten, en dus is $L[x, y, z]$ de ring van gehelen.

□

2 Globale schets van de methode van berekening

1. Begin met $i = 1$, $B_1 := \{1, y, \ldots, y^{n-1}\}$ en $M_1 = L[x]B_1$, het door de elementen van B_1 voortgebrachte rooster.

2. Zolang als $\overline{L[x]} \neq M_i$ doe:
   (a) kies een $a \in \overline{L[x]} \setminus L[x]B_i$
   (b) $M_{i+1} := L[x]B_i + L[x]a$, het door de elementen van B_i en a voortgebrachte deelmoduul van $\overline{L[x]}$. M_{i+1} is weer een rooster.
   (c) $B_{i+1} :=$ een verzameling van n vrije voortbrengers van het $L[x]$-moduul M_{i+1}.
   (d) $i := i + 1$
3. Nu is $M_1 \subset M_2 \ldots \subset \overline{L[x]}$ een strikt stijgende rij $L[x]$-modulen binnen het Noetherse $L[x]$-moduul $\overline{L[x]}$. Daarom moet voor zekere k gelden $M_k = \overline{L[x]}$.

Voor de discriminanten van $B_1, B_2, \ldots$ geldt elke keer:

$$\mathrm{disc}(B_i) = \mathrm{disc}(B_{i+1})(\det(\text{overgangsmatrix}))^2 \in L[x]$$

Hieruit volgt 'nog eens' dat de rijtjes $M_1, M_2, \ldots$ en $B_1, B_2, \ldots$ eindigen. Een bovengrens voor de lengte van de rijtjes wordt gegeven door een half maal het aantal factoren in de discriminant van M_1. Ik heb de woorden 'nog eens' tussen aanhalingstekens gezet omdat ik het Noethers zijn van $\overline{L[x]}$ juist bewezen heb met behulp van de discriminant.

Door lemma 2 toe te passen zie je dat $M_i = \overline{L[x]}$ als de discriminant van B_i kwadraatvrij is. Omgekeerd hoeft echter niet te gelden dat de discriminant van $\overline{L[x]}$ kwadraatvrij is.

Voorbeeld: neem $L = \overline{\mathbb{Q}}$. Ik zoek die elementen van het algebraïsch functielichaam

$$\overline{\mathbb{Q}}(x)[y]/(y^3 + xy + x^2)$$

die geheel zijn over $\overline{\mathbb{Q}}[x]$. Hier voldoet y aan f waar $f(z) = z^3 + xz + x^2$. De discriminant van de verzameling $\{1, y, y^2\}$ is $-x^3(4 + 27x)$. De enige factor die hier in het kwadraat voorkomt is x. Door de karakteristieke veelterm van y^2/x uit te rekenen zie je dat dit element geheel is over $\overline{\mathbb{Q}}[x]$. Dus $\{1, y, y^2\}$ brengt de ring van gehelen niet voort, want

$$\frac{y^2}{x} \notin \overline{\mathbb{Q}}[x]1 + \overline{\mathbb{Q}}[x]y + \overline{\mathbb{Q}}[x]y^2$$

De elementen van

$$\overline{\mathbb{Q}}[x]1 + \overline{\mathbb{Q}}[x]y + \overline{\mathbb{Q}}[x]y^2 + \overline{\mathbb{Q}}[x]\frac{y^2}{x} = \overline{\mathbb{Q}}[x]1 + \overline{\mathbb{Q}}[x]y + \overline{\mathbb{Q}}[x]\frac{y^2}{x}$$

zijn allemaal geheel. De discriminant van $\{1, y, y^2/x\}$ is $-x(4+27x)$, is kwadraatvrij, daarom kunnen er ook niet meer gehelen zijn. Dus $\overline{L[x]}$ wordt als $L[x]$-moduul voortgebracht door $\{1, y, y^2/x\}$. In dit voorbeeld wordt als grondlichaam $\overline{\mathbb{Q}}$ genomen, maar feitelijk gaat de berekening over $\mathbb{Q}$. Het geval van een niet algebraïsch gesloten lichaam komt in paragraaf 5 aan de orde.

Ik schrijf vanaf nu steeds b i.p.v. B_i en $L[x]b$ i.p.v. M_i. Het probleem dat overblijft is in de beschreven methode om $\overline{L[x]}$ te vinden is: hoe weet je of $\overline{L[x]} = L[x]b$ (nodig voor stap 2

van het algoritme), en zoniet, hoe vind je een nieuwe gehele $a \in \overline{L[x]} \setminus L[x]b$ (nodig voor stap 2a). Ik zal in paragraaf 3 een methode geven om zo'n a te berekenen, en vervolgens daarmee in paragraaf 4 dit algoritme uitwerken.

In het algoritme geldt steeds:

$$L[x,y] \subset L[x]b \subset \overline{L[x]}$$

en

$$\mathrm{disc}(\overline{L[x]}) \mid \mathrm{disc}(b) = \mathrm{disc}(L[x]b) \mid \mathrm{disc}(L[x,y]) = \mathrm{disc}(\{1, y, \ldots, y^{n-1}\}) = \mathrm{disc}(f).$$

Als $L[x]b$ geen ring is kun je gemakkelijk nieuwe gehelen (elementen van $\overline{L[x]} \setminus L[x]b$) vinden door elementen van b met elkaar te vermenigvuldigen. $\overline{L[x]}$ is immers een ring.

Algemeen heb je in elke stap van het algoritme:

- b is een verzameling van n gehelen, zeg $b = \{b_1, \ldots, b_n\}$
- Als er een $a \in \overline{L[x]} \setminus L[x]b$ is, dan is a te schrijven als:

  $$a = \frac{a_1 b_1 + \ldots + a_n b_n}{k}$$

  met $a_1, \ldots, a_n$ en k in $L[x]$, omdat b een $L(x)$-basis van $L(x,y)$ is.
- Door eventueel a met een element uit $L[x]$ te vermenigvuldigen mag ik aannemen dat k irreducibel is in $L[x]$, en dat niet alle a_i nul zijn modulo k.
- Omdat $L[x]/(k)$ een lichaam is, kan ik een a met een element van $L[x]$ vermenigvuldigen, zó dat één van de $a_i \not\equiv 0$ modulo k gelijk wordt aan 1 modulo k. Dan blijft a geheel als ik die a_i vervang door 1; het al dan niet geheel zijn van a wordt door de a_i's modulo k bepaald. Ik mag dus aannemen $\exists_i a_i = 1$.
- Als $a_i = 1$, en b' is de verzameling b met b_i vervangen door a dan is de determinant van de overgangsmatrix tussen b en b' gelijk aan k. Daarom geldt $\mathrm{disc}(b) = k^2 \mathrm{disc}(b')$, dus $k^2 = \mathrm{disc}(b)/\mathrm{disc}(b') \mid \mathrm{disc}(b)/\mathrm{disc}(\overline{L[x]})$. In het bijzonder heb je: $k^2 | \mathrm{disc}(b)$. Deze b' is tevens de verzameling die in stap 2c gevraagd wordt.

Uit dit en het vorige punt volgt het $\Longrightarrow$ gedeelte van de volgende

Stelling 4 *Stel dat k irreducibel is, dan geldt de volgende equivalentie*

$$\exists_{a_1,\ldots,a_n \in L[x]} \exists_i \left[a_i \not\equiv 0 \bmod k \bigwedge \frac{a_1 b_1 + \ldots + a_n b_n}{k} \in \overline{L[x]} \right] \Longleftrightarrow k^2 \Big| \frac{\mathrm{disc}(b)}{\mathrm{disc}(\overline{L[x]})}$$

Bewijs: nu nog alleen het $\Longleftarrow$ gedeelte van de stelling. Stel $\overline{L[x]}$ wordt voortgebracht door $c = \{c_1, \ldots, c_n\}$ en $k^2 | (\mathrm{disc}(b))/(\mathrm{disc}(c))$. Dan geldt dat k een deler is van de determinant van de overgangsmatrix van c naar b. Dus moet er een element $c_j \in c$ zijn dat van de vorm

$$c_j = \frac{a_1 b_1 + \ldots + a_n b_n}{l} \bigwedge \exists_i a_i \not\equiv 0 \bmod k$$

is, met $a_1, \ldots, a_n$ en l in $L[x]$, en $k | l$. Door c_j dan met l/k te vermenigvuldigen krijg je een uitdrukking van de juiste vorm.

□

Stelling 5 *Stel $L[x,y] \subset L[x]b \subset \overline{L[x]}$, stel dat k irreducibel is, en niet nul in een van de singulariteiten van $C(f) \bigcap V$. Stel $l \in L[x]b$ en $l/k \in \overline{L[x]}$. Dan geldt: $l/k \in L[x]b$.*

Bewijs: ik doe eerst het geval dat $L[x]b = L[x,y]$. Stel dus dat $l \in L[x,y]$ en dat l/k geheel is. Dan geldt:

$$\frac{l}{k} \in \overline{L[x]} = \bigcap_{O \in X_L \text{ boven } V} O \subset \bigcap_{O \in X_L \text{ boven } V \bigwedge k(O)=0} O = \bigcap_{P \in V \cap C(f) \bigwedge k(P)=0} O_P$$

, omdat waar $k = 0$ geen singulariteiten liggen, en dus punten en plaatsen overeenkomen. Omdat je in O_P door k mag delen als $k(P) \neq 0$ geldt:

$$\frac{l}{k} \in \bigcap_{P \in V \cap C(f) \bigwedge k(P) \neq 0} O_P$$

combineren van deze twee geeft

$$\frac{l}{k} \in \bigcap_{P \in V \cap C(f)} O_P = L[x,y]$$

Nu het algemene geval: stel $l \in L[x]b$, $l/k \in \overline{L[x]} \setminus L[x]b$ en k is niet nul in een van de singulariteiten. Dan zegt stelling 4

$$k^2 \mid \frac{\text{disc}(b)}{\text{disc}(\overline{L[x]})}$$

. Nu is disc(b) een deler van disc$(L[x,y])$ dus heb je

$$k^2 \mid \frac{\text{disc}(L[x,y])}{\text{disc}(\overline{L[x]})}$$

. Dan vind je, met het $\Longleftarrow$ gedeelte van stelling 4 toegepast voor $b = \{1, y, \ldots, y^{n-1}\}$ een l' in $L[x,y]$, met

$$\frac{l'}{k} \in \overline{L[x]} \setminus L[x,y]$$

in tegenspraak met het deel van stelling 5 dat ik al bewezen had.

□

Ik vat alle punten hier nog eens samen in de volgende

Stelling 6 *Zij b een verzameling van n elementen van $L(x,y)$, zo dat $L[x,y] \subset L[x]b \subset \overline{L[x]}$. Als $\overline{L[x]} \setminus L[x]b \neq \emptyset$ dan is er een $a \in \overline{L[x]} \setminus L[x]b$ van de vorm:*

$$a = \frac{a_1 b_1 + \ldots + a_n b_n}{k} \bigwedge \exists_i a_i = 1 \bigwedge k \text{ is irreducibel} \bigwedge k^2 | \text{disc}(b) \bigwedge \exists_{P \in S(f)} k(P) = 0 \qquad (*)$$

3 Het vinden van gehelen in $\overline{L[x]} \setminus L[x]b$

Vanwege stelling 6 hoef ik alleen te zoeken naar gehelen van de vorm $(*)$. Als er zulke gehelen niet meer zijn dan geldt $\overline{L[x]} \setminus L[x]b = \emptyset$ en is de ring van gehelen gevonden. Kies nu een monische $k \in L[x]$ die aan $(*)$ voldoet.

De vraag is nu dus voor welke $a_i \in L[x]$ geldt dat $a = (a_1b_1 + \ldots + a_nb_n)/k$ van de vorm $(*)$ geheel is. Het al dan niet geheel zijn wordt door de a_i's modulo k bepaald. Ik kan dus een representantenverzameling R van de restklassen van $L[x]$ modulo k kiezen, en me beperken tot a_i's die in R zitten.

Deze a_i's waarvoor a geheel is vind ik door een stelsel vergelijkingen V in $a_1, \ldots, a_n$ te zoeken dat equivalent is met het geheel zijn van $a = (a_1b_1 + \ldots + a_nb_n)/k$. Omdat ik me beperk tot a_i's in R hoeft deze equivalentie niet voor alle $a_i \in L[x]$ te gelden, maar alleen maar voor a_i's in R. Dit stelsel vergelijkingen zal ik dan m.b.v. een Gröbnerbasis berekening oplossen. Voor de verzameling V die ik zoek moet dan gelden:

$$(a_1, \ldots, a_n) \text{ voldoet aan } V \iff a \text{ is geheel}$$

In deze paragraaf is L algebraïsch gesloten, en de graad van k dus 1 (k is irreducibel). Ik kan hier dus $R = L$ als representantenverzameling kiezen. In paragraaf 5 wil ik op dezelfde manier als hier gehelen vinden. Alleen zal dan L niet algebraïsch gesloten zijn. Dan kan graad(k) groter dan 1 zijn, en kan ik L niet meer als representantenverzameling kiezen. Om paragraaf 5 het gemakkelijkst met paragraaf 3 te kunnen vergelijken zal ik daarom ook hier R blijven schrijven i.p.v. L.

3.1 Vergelijkingen vinden

Het idee om de vergelijkingen te vinden is: bereken van a de karakteristieke veelterm g, in de dummy-variabele z, over de eindige lichaamsuitbreiding

$$L(x, y) : L(x)$$

Dus

$$a := \frac{a_1b_1 + \ldots + a_nb_n}{k} \in L(x, y)$$

en

$$g(z) \in L(x)[z]$$

In a staat alleen k in de noemer, dus zit $k^n g(z)$ in $L[x, z]$. Definiëer nu

$$W := \text{de verzameling coëfficiënten van } k^n g(z) \text{ als veelterm in } z$$

Dan heb je:

$$a \text{ geheel} \iff g(z) \in L[x, z] \iff k^n g(z) \equiv 0 \bmod k^n \iff \forall_{r \in W} r \equiv 0 \bmod k^n.$$

Tot nu toe heb ik nog niet gebruikt dat de a_i's in de representantenverzameling L zitten, of dat graad$(k) = 1$. Dat ga ik nu wel doen. Het volgende voorbeeld laat zien hoe je het feit

dat de variabele x in de $\tilde{a}_i$'s niet voorkomt kunt gebruiken om x uit de vergelijkingen weg te werken. Zo zet ik W om in een stelsel V waar x niet meer in voorkomt.

Voorbeeld: Stel $k = x - 1$, $n = 2$ en $r = x^3 + a_1x^2 + a_1x + a_1 + a_2 \in W$. Dit geeft als vergelijking:

$$x^3 + a_1x^2 + a_1x + a_1 + a_2 \equiv 0 \bmod (x-1)^2$$

oftewel

$$x^3 + a_1x^2 + a_1x + a_1 + a_2 = 0 \text{ in } L[x]/((x-1)^2)$$

rest nemen modulo $(x-1)^2$ geeft:

$$(3a_1 + 3)x + (a_2 - 2) = 0 \text{ in } L[x]/((x-1)^2) \qquad (A)$$

Als ik me nu beperk tot oplossingen a_i die geen x bevatten, dan is deze vergelijking equivalent met

$$(3a_1 + 3 = 0) \bigwedge (a_2 - 2 = 0) \text{ in } L \qquad (B)$$

Deze laatste twee vergelijkingen komen dan in de verzameling V.
10pt

Ik definiëer V nu als volgt:

$$V := \bigcup_{r \in W} \text{coeffs}(r \bmod k^n)$$

of in woorden: ik neem eerst van elke $r \in W$ de rest modulo k^n, en vervolgens stop ik de coëfficiënten (als veelterm in x) van $r \bmod k^n$ in V. Dan heb je:

$$a \text{ geheel} \iff \forall_{r \in V} r = 0$$

voor a_i's in R. Door V uit te rekenen met onbepaalde a_i's vind je veelterm vergelijkingen in de a_i's.

Ik heb nu het stelsel W van 'is nul in $L[x]/(k^n)$' vergelijkingen omgezet in een stelsel V van 'is nul in L' vergelijkingen. Het voordeel daarvan is dat 'is nul in een lichaam' vergelijkingen gemakkelijker met een Gröbnerbasis-berekening op te lossen zijn dan 'is nul in een ring' vergelijkingen.

Het voorbeeld was echter niet helemaal realistisch. De ring van gehelen is immers een $L[x]$-moduul. De verzameling oplossingen $\{(a_1, \ldots, a_n) | a \text{ geheel}\}$ moet dus een L-vectorruimte zijn. De gevonden vergelijkingen moeten dus homogeen zijn.

In het voorbeeld zag je wel dat één 'is nul in $L[x]/(k^n)$' vergelijking $x^3 + a_1x^2 + a_1x + a_1 + a_2 = 0$ van W equivalent is met n (in het voorbeeld twee) vergelijkingen (B) van V. Je ziet dat die equivalentie alleen geldt bij beperking tot oplossingen in de R. Als ik alle oplossingen in $L[x]$ zou toestaan geldt de equivalentie tussen (A) en (B) niet meer.

3.2 Vergelijkingen oplossen

Het stelsel vergelijkingen is het gemakkelijkst op te lossen met een Gröbner-basis als je weet dat er slechts 0 of 1 oplossingen zijn. Dit bereik je door enkele vergelijkingen toe te voegen, want dan wordt het aantal oplossingen i.h.a. kleiner. Dat kan als volgt:

1. $l := n$
2. $G :=$Gröbnerbasis$(V \cup \{a_1 = 0, \ldots, a_{l-1} = 0, a_l = 1\})$
3. Als $1 \in G$ en $l > 1$ dan $l := l - 1$, terug naar 2
4. Twee mogelijkheden:
   (a) $1 \in G$ dan heeft V alleen de triviale oplossing $a_1 = 0, \ldots, a_n = 0$, je vindt dan geen nieuwe gehelen
   (b) $1 \notin G$, dan heeft de verzameling G nog slechts één oplossing. Als $a_1, \ldots, a_n$ die oplossing is dan is $(a_1 + \ldots + a_n)/k$ geheel, en $\notin L[x]b$.

Ik moet nog bewijzen dat de twee mogelijkheden bij 4 de enige mogelijkheden zijn. Stel er is een niet-triviale (niet-nul) oplossing van V. Kies dan l maximaal zó dat $V \cup \{a_1 = 0, \ldots, a_{l-1} = 0\}$ nog een niet-triviale oplossing heeft (l kan 1 zijn). Neem zo'n oplossing $a_1, \ldots, a_n$, dan is $a_l \neq 0$. Deel deze oplossing door a_l dan heb je een oplossing van $V \cup \{a_1 = 0, \ldots, a_{l-1} = 0, a_l = 1\})$. Deze oplossing is uniek. Stel immers dat $V \cup \{a_1 = 0, \ldots, a_{l-1} = 0, a_l = 1\})$ twee oplossingen heeft, dan is het verschil een niet-triviale oplossing van $V \cup \{a_1 = 0, \ldots, a_l = 0\}$, in tegenspraak met de maximaliteit van l.

Hoe los je een stelsel vergelijkingen op waarvan je weet dat er slechts één oplossing is? Stel je wilt bijvoorbeeld a_j berekenen voor zekere vaste j. Er is slechts één mogelijkheid voor a_j, zeg w. Dan moet $(a_j - w)^m$ in het door het stelsel voortgebrachte ideaal zitten voor zekere m (Nullstellensatz). Neem m minimaal met die eigenschap. Kies nu een lexicografische ordening op de a_i's waarbij a_j het kleinst is. Bereken met deze ordening de Gröbner-basis. In deze Gröbner-basis moet een element g zitten waarvan de kopterm de kopterm van $(a_j - w)^m$ deelt, dus de kopterm van g is een deler van a_j^m. Dan bevat g alleen de onbepaalde a_j, omdat de andere onbepaalden in de gekozen ordening groter zijn dan de kopterm van g. Dan moet g wel gelijk zijn aan $(a_j - w)^m$. De oplossing w vind je dan eenvoudig, $w = -1/m$ maal de coëffiënt van a_j^{m-1} in g.

4 Het algoritme

Ik zal het programma zoals dat in paragraaf 2 beschreven staat hier uitwerken. Ik zal in het programma ook het geval meenemen dat L niet algebraïsch gesloten is. In paragraaf 5 zal ik verder op dat geval ingaan.

Ik zal in paragraaf 4.1 een volledig algoritme geven, in een beter leesbare taal dan het eigenlijke algoritme zelf. De bedoeling is wel dat de implementatie van het algoritme in paragraaf 4.1 geen problemen oplevert voor wie bekend is met een computeralgebra systeem. Ik zal daarom alleen stappen maken die met enkele Maple commando's uit te voeren zijn.

In paragraaf 4.2 zal ik dan nog wat verfijningen op het algoritme geven, die het algoritme sneller maken. In paragraaf 4.3 zal ik dan het eigenlijke algoritme geven, dat ik in Maple geschreven heb.

4.1 Een compleet programma

De input van dit programma zal zijn een $f \in L[x,y]$, monisch in y en irreducibel. Het algebraïsch functielichaam $L(x,y)$ wordt hier gegeven door f als minimumveelterm van y over $L(x)$. De output van het programma moet dan $\overline{L[x]}$ zijn.

1. Initialisatie
   (a) $b_i := y^{i-1}$ voor $i = 1, \ldots, n$
   (b) $b := \{b_1, \ldots, b_n\}$, dus nu is $L[x]b = L[x,y]$
   (c) $d :=$disc(f,y), de discriminant van f als veelterm in y. Dit is gelijk aan disc(b).
   (d) $S :=$ de verzameling monische irreducibele $k \in L[x]$ waarvoor geldt $k^2|d$
2. Voor elke $k \in S$ doe:
   (a) *vergelijkingen vinden:*
   $a := (a_1 b_1 + \ldots + a_n b_n)/k$, waar $a_1, \ldots, a_n$ onbepaalden zijn
   (b) $g :=$ karakteristieke veelterm van a in de dummy-variabele z.
   (c) $W :=$coeffs$(k^n g, z)$ de verzameling coëfficiënten van $k^n g$ als veelterm in z
   (d) $W := W$ mod k^n, ik vervang elk element in W door de rest van dat element bij deling door k^n
   (e)
   - *als de graad van k 1 is:*
     $V := \bigcup_{i \in W}$ coeffs(i,x), de verzameling coëfficiënten van de elementen van W als veeltermen in x, ga verder naar 2f
   - *als de graad van k groter dan 1 is:*
     Dit geval waarin L niet algebraïsch gesloten is neem ik alvast op in het programma, ik moet echter nog wel bewijzen dat het werkt. Zie daarvoor paragraaf 5.
     Zij w een onbepaalde.
     $W' :=$subs$(x = x + w, W)$, ik vervang x door $x + w$
     $W' := W'$ mod w^n, ik vervang elk element in W' door de rest van dat element bij deling door w^n
     $V := \bigcup_{i \in W'}$ coeffs(i,w), de verzameling coëfficiënten van de elementen van W' als veeltermen in w
     $V := V \bigcup \{k\}$, dit veroorzaakt dat het Gröbnerbasis-algoritme een berekening over $L[x]/(k)$ zal uitvoeren i.p.v. over L.
   (f) *vergelijkingen oplossen:*
   $l := n$
   (g) $G :=$Gröbnerbasis$(V \cup \{a_1 = 0, \ldots, a_{l-1} = 0, a_l = 1\})$
   (h) Als $1 \in G$ en $l > 1$ dan $l := l - 1$, terug naar 2g
   (i) Als $1 \in G$ ga dan terug naar stap 2, maar dan met de volgende k

(j) Nu heeft $V \cup \{a_1 = 0, \ldots, a_{l-1} = 0, a_l = 1\}$ precies 1 oplossing $(\tilde{a}_1, \ldots, \tilde{a}_n)$. Die ga ik in de stappen 2j tot en met 2o berekenen:
$\tilde{a}_i := 0$ voor $i = 1 \ldots l$
$\tilde{a}_l := 1$

(k) $l := l + 1$

(l) Als $l = n + 1$ dan $\tilde{a} := (\tilde{a}_1 b_1 + \ldots + \tilde{a}_n b_n)/k$, ga naar 2p

(m) $\tilde{a}_l$ *berekenen:*
$G :=$Gröbnerbasis$(V \cup \{a_1 = \tilde{a}_1, \ldots, a_l = \tilde{a}_{l-1}\})$, met een lexicografische waarin:

- *Als de graad van k 1 is:*
  a_l het kleinst is.
- *Als de graad van k groter dan 1 is:*
  dan komt x nog voor als variabele in de verzameling V. Ik kies dan x het kleinst in de ordening, en a_l het op een na kleinst.

(n) $g :=$ het enige element van G dat alleen a_l (a_l en x, als graad$(k) > 1$) bevat.

(o) $m :=$ de graad van g als veelterm in a_l
$\tilde{a}_l := -\text{coeff}(g, a_l^{m-1})/m$, $-1/m$ maal de coëfficiënt van a_l^{m-1}.
Ga terug naar 2k

(p) *de verzameling b aanpassen tot voortbrengers van* $L[x]b + L[x]\tilde{a}$*:*
$b_l := \tilde{a}$, ik vervang een element van b door de nieuwe gehele $\tilde{a}$

(q) $d := d/k^2$, dit is nu weer de discriminant van de zojuist veranderde verzameling $b = \{b_1, \ldots, b_n\}$

(r) Als $k^2 | d$ ga dan terug naar 2a

(s) Als $k^2 \not| d$ ga dan terug naar 2 met de volgende k

3. Nu moet gelden $\overline{L[x]} = L[x]b$. De ring van gehelen wordt hier dus gegeven door een verzameling van n elementen, die de ring van gehelen als $L[x]$-moduul voortbrengen.

Opmerking: je kunt de verzameling S in stap 1d beperken tot alleen die k's die 0 zijn in een singulariteit. Dat kost een extra Gröbnerbasisberekening, maar S wordt er misschien kleiner door. Je berekent S dan als volgt:

1. $G :=$Gröbnerbasis$(\{f, \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\})$, met lexicografische ordening, zó dat x lager in de ordening is dan y
2. $g :=$ het enige element van G dat geen y bevat
3. $S :=$ de verzameling irreducibele factoren van g

Afhankelijk van de input van het programma kan het sneller zijn om stap 1d door deze 3 stappen te vervangen.

4.2 Verfijningen op het programma

Ik heb in het algoritme nog wat verfijnd om het sneller te maken. Een belangrijke verbetering is te halen uit de triviale constatering dat de ring van gehelen een ring moet zijn. Daarom

kun je zolang als het deelmoduul

$$L[x]\{b_1, \ldots, b_n\}$$

geen ring is nieuwe gehelen vinden door de b_i's met elkaar te vermenigvuldigen. Elke nieuwe gehele in $\overline{L[x]} \setminus L[x]\{b_1, \ldots, b_n\}$ die je zo vindt bespaart een berekening van een karakteristieke veelterm.

Het aanpassen van de verzameling $\{b_1, \ldots, b_n\}$ tot een stelsel vrije voortbrengers van een groter deelrooster van $\overline{L[x]}$ gaat het gemakkelijkst als je steeds zorgt dat:

$$\text{graad}(b_i) = i - 1$$

waarbij de graad van b_i de graad als veelterm van y bedoeld is. Als namelijk telkens $\text{graad}(b_i) = i - 1$ geldt, dan is het eenvoudig om na te gaan of een gegeven element $a \in L(x, y)$ in het door de b_i's voortgebrachte rooster zit.

Verder kun je gebruiken dat

$$\text{spoor}(ab_i) \in L[x]$$

als a geheel is. Dit levert voor elke b_i een $L[x]/(k)$-lineaire vergelijking in de a_i's op. Met elke niet-triviale $L[x]/(k)$-lineaire vergelijking kun je één a_i in de overige a_i's uitdrukken. Deze a_i kun je dus door substitutie vervangen door andere a_i's. Daardoor daalt het aantal a_i's, en dat is gunstig omdat berekeningen i.h.a. sneller gaan als er minder onbepaalden in de uitdrukkingen voorkomen.

Nog een verbetering van het algoritme is als volgt te bereiken. In stap 2a tot en met 2e in paragraaf 4.1 wordt een stelsel vergelijkingen bepaald. Vervolgens wordt in stap 2g voor een aantal a_i's 0 ingevuld en voor één a_i 1 ingevuld. Deze twee bewerkingen kunnen beter omgewisseld worden. Dus eerst een aantal a_i's wegwerken door er nullen en een 1 voor in te vullen, en dan pas de vergelijkingen gaan zoeken. Als ik die twee omwissel dan moet wel elke keer dat stap 2g aangeroepen wordt de verzameling vergelijkingen V opnieuw bepaald worden. Toch gaat het algoritme op deze manier veel sneller.

4.3 Het algoritme

Dit is het algoritme zoals ik dat in Maple geschreven heb.

```
# De globale variabelen in het algoritme zijn:
# x,y : dit moeten onbepaalden zijn
# graad, dat is de graad van f, waar f de minimumveelterm van y over L(x) is
# discriminant

# Input f, de minimumveelterm van y over L(x), het functielichaam is
# dus L(x)[y])/(f). f moet een polynoom zijn, d.w.z. y moet geheel zijn
# Output: een stelsel voortbrengers van de ring van gehelen als L[x]-moduul
ringvangehelen:=proc(f) local k,dubbele_factoren,
Lx_basis_deelmoduul_vd_ring_van_gehelen,oude_basis;
```

```
graad:=degree(f,y);
Lx_basis_deelmoduul_vd_ring_van_gehelen:=[1,'y^dummy'$dummy=1..graad-1];
discriminant:=factor(discrim(f,y));
dubbele_factoren:=kwadraatdelers(discriminant);
for k in dubbele_factoren
do oude_basis:=NULL;
 while oude_basis<>Lx_basis_deelmoduul_vd_ring_van_gehelen
 # d.w.z. while er was nog een gehele gevonden
 and multipliciteit(discriminant,k)>1
 do oude_basis:=Lx_basis_deelmoduul_vd_ring_van_gehelen;
  Lx_basis_deelmoduul_vd_ring_van_gehelen:=
  nieuwegehelen(k,Lx_basis_deelmoduul_vd_ring_van_gehelen,f)
 od
od;
Lx_basis_deelmoduul_vd_ring_van_gehelen
# Dat deelmoduul is nu de hele ring van gehelen
end:


# Input: de factor k en de basis [b1,...,bn]
# Output: een basis die uitgebreid is met gehelen van de vorm
# (a1*b1+...+an*bn)/k, als er die zijn tenminste
nieuwegehelen:=proc(k,b,f) local verz,i,j,klverz,codimensie,basis,
basis_ring,verzc,aantal_ai,gesubstitueerden,aa,a,sp,s,t;
basis:=b;
a:=sum('basis[dummy]*a.dummy',dummy=1..graad);
# in de scriptie heb ik het steeds over a=sum(..)/k, ik zal nog
# voor die factor k corrigeren.
gesubstitueerden:={};
for i from 1 to graad
do
sp:=expand(rem(spoor(a*basis[i],f),k,x));
# Dit geeft een L[x]/(k)-lineaire vergelijking voor de a.i, daarmee
# zal ik nu het aantal a.i gaan verkleinen d.m.v. substitutie.
  if sp<>0 then
   j:=de_kleinste_ai(sp);
   # Nu kan ik a.j vervangen door substitutie
   gcdex(k,coeff(sp,a.j),1,x,'s','t');
   sp:=expand(rem(expand(sp*t),k,x));
   a:=subs(a.j=a.j-sp,a);
   gesubstitueerden:=gesubstitueerden union {sp};
  fi
```

```
od;
verzc:=[op(de_ai_gesorteerd(a)),x];
# Dit zijn de a.i tjes die nog over zijn, plus x
aantal_ai:=nops(verzc)-1;
# Het aantal a.i tjes dat nog over is.

# In de vorige versie deed ik eerst (op deze plek in het algoritme) de
# vergelijkingen zoeken met de volgende regel:
#     verz:=vergelijkingen_zoeken(a,k,f);
# Daarna kwam pas de substitutie waarin een aantal a.i tjes door 0
# vervangen worden, en een a.i door 1. Nu doe ik dat omgekeerd, ik voer
# eerst die substitutie uit (dan niet in verz maar in a) en bepaal dan pas
# verz=vergelijkingen_zoeken(a,k,f). Dat blijkt veel sneller te zijn.

codimensie:=0;
# Dat wordt (hoeft hij nu nog niet te zijn) de codimensie van de
# oplossingsruimte in de L[x]/(k)-vectorruimte van nog overgebleven
# mogelijkheden. De dimensie van de overgebleven mogelijkheden is aantal_ai
while codimensie < aantal_ai and multipliciteit(discriminant,k)>1
do
  verz:=vergelijkingen_zoeken(subs({'verzc[dummy]=0'$
  dummy=codimensie+2..aantal_ai,verzc[codimensie+1]=1},a),k,f);
  klverz:=gbasis(verz,verzc);
  if not klverz=[1] then
     # Nu is er nog precies 1 oplossing, de procedure losop zal die
     # dan vinden.
     i:=ai_numero(verzc[codimensie+1]);
     klverz:={op(klverz),-1+a.i,
     'verzc[dummy]'$dummy=codimensie+2..aantal_ai};
     aa:=losop(klverz union gesubstitueerden,x);
     aa:=normaalvorm(sum('aa[dummy]*basis[dummy]',dummy=1..graad)/k,f);
     basis:=['basis[dummy]'$dummy=1..i-1,aa,'basis[dummy]'$dummy=i+1..graad];
     discriminant:=discriminant/k^2;
     # De volgende twee regels dienen voor versnelling v.h. algoritme
     basis_ring:=maak_er_een_ring_van(basis,i,k,f);
     if basis_ring<>basis then RETURN(basis_ring) fi;
     # Nu de a.i wegwerken:
     gesubstitueerden:=gesubstitueerden union {a.i};
     a:=subs(a.i=0,a);
     verzc:=['verzc[dummy]'$dummy=1..codimensie,
     'verzc[dummy]'$dummy=codimensie+2..aantal_ai,x];
     aantal_ai:=aantal_ai-1;
```

```
    # Vanaf hier tot ‘else’ dient voor versnelling, kan weggelaten worden
    # Wat ik hier doe is namelijk een lineaire vergelijking voor de a.i’tjes
    # zoeken door spoor(a*(de zojuist gevonden nieuwe gehele)) uit te
    # rekenen. Dat is niet nodig, alleen voor versnelling v.h. algoritme
    sp:=expand(rem(spoor(a*aa,f),k,x));
    if sp<>0 then
       i:=de_kleinste_ai(sp);
       gcdex(k,coeff(sp,a.i),1,x,’s’,’t’);
       sp:=expand(rem(expand(sp*t),k,x));
       a:=subs(a.i=a.i-sp,a);
       gesubstitueerden:=gesubstitueerden union {sp};
       verzc:=[op(de_ai_gesorteerd(a)),x];
       aantal_ai:=aantal_ai-1;
       # De codimensie kan nu ook dalen, dus, voor de zekerheid doe ik:
       if codimensie>0 then codimensie:=codimensie-1 fi
    fi
  else
    # In dit geval ‘else’ is er geen nieuwe gehele gevonden en klopte de
    # waarde van ‘codimensie’ blijkbaar niet, die moet ik nu dus aanpassen.
    codimensie:=codimensie+1
  fi
od;
basis
end:

# Input: een stelsel b1..bn, waarvan b.i gewijzigd is, waardoor
# L[x]{b1,..bn} misschien geen ring meer is
# Output: L[x]-voortbrengers van de ring voortgebracht door b1,..,bn
maak_er_een_ring_van:=proc(basis,i,k,f) _maak_ring(basis[i],basis,2,f) end:
_maak_ring:=proc(a,basis,n,f) local v,i,aa;
if n>graad then RETURN(basis) fi;
aa:=normaalvorm(a*basis[n],f);
v:=schrijf_uit_op_basis(aa,basis);
i:=graad;while i>0 and not member(x,indets(denom(v[i]))) do i:=i-1 od;
if i=0 then RETURN(_maak_ring(a,basis,n+1,f)) fi;
gcdex(denom(v[i]),numer(v[i]),1,x,’s’,’t’);
v:=[’normal(v[dummy]*t)’$dummy=1..i-1,1/denom(v[i])];
discriminant:=discriminant/factor(denom(v[i]))^2;
v:=normaalvorm(sum(’v[dummy]*basis[dummy]’,dummy=1..i),f);
_maak_ring(a,[’basis[dummy]’$dummy=1..i-1,v,
’basis[dummy]’$dummy=i+1..graad],n,f)
end:
```

```
# Ik neem hier aan dat basis[1]=1 en dat graad(basis[i],y)=i-1
schrijf_uit_op_basis:=proc(a,basis) local q,c,aa,result;aa:=a;result:=NULL;
for d from graad-1 by -1 to 0 do
c:=normal(coeff(aa,y,d));
c2:=normal(coeff(basis[d+1],y,d));
q:=normal(c/c2);result:=q,result;
aa:=collect(aa-q*basis[d+1],y) od;[result] end:

ai_numero:=proc(ai) local i;
for i from 1 to graad do
if ai=a.i then RETURN(i) fi od;
ERROR(helaas) end:

# Ik wil hier with(linalg) doen, maar helaas raak je dan Maple's
# belangrijkste procedure kwijt. Daarom geeft ik die hier nu
# een andere naam.
doorzoek:=op(trace):with(linalg):
lprint(`Gebruik nu de procedure doorzoek i.p.v. trace`):
with(grobner):

# Input: een expressie in L(x,y)
# Output: een expressie in L(x)[y] die in L(x)[y]/(f) hetzelfde is, en die
# bovendien een graad in y heeft, kleiner dan de graad van f.
normaalvorm:=proc(expressie,f) local e,n;
options remember;
if type(expressie,polynom(anything,y)) then
RETURN(_normaalvorm(expressie,f)) fi;
e:=normal(expressie);n:=inverse(matrixvorm(denom(e),f));
n:=sum('n[1,dummy]*y^(dummy-1)',dummy=1..graad);
normaalvorm(n*numer(e),f) end:

_normaalvorm:=proc(expressie,f) local ff;
ff:=expand(expressie);
sum('expand(coeff(ff,y,dummy)*onthoud_normaalvorm_yn(dummy,f))',
dummy=0..degree(ff,y))
end:

onthoud_normaalvorm_yn:=proc(n,f)
options remember;
if n<graad then RETURN(y^n) fi;
_normaalvorm(y^(n-graad)*(y^graad-f),f) end:
```

```
# Ik neem hier aan dat f monisch is.

# Input: een expressie in L(x,y) (die moet je als element van
#                                 L(x)[y]/(f) interpreteren)
# Output: de matrix van de L(x)-lineaire afbeelding: "vermenigvuldigen
# met expressie"
matrixvorm:=proc(expressie,f) local ff,m,i,j;
m:=matrix(graad,graad);
ff:=normaalvorm(expressie,f);
for i from 0 to graad-1 do
ff:=expand(subs(y^graad=onthoud_normaalvorm_yn(graad,f),ff));
for j from 0 to graad-1 do
m[i+1,j+1]:=coeff(ff,y,j) od;
ff:=expand(ff*y) od;m end:

# Input: een expressie in L(x,y) in normaalvorm (graad in y is < graad f)
# Output: de naam zegt het al
karakteristiekeveelterm:=proc(expressie,z,f)
charpoly(matrixvorm(expressie,f),z) end:

# Input: een expressie in L(x,y)
# Output: de naam zegt het al
spoor:=proc(expressie,f) local ff;ff:=collect(expressie,y);
normal(sum('onthoud_spoor(y^dummy,f)*coeff(ff,y,dummy)',
dummy=0..degree(ff,y)))
end:

# Berekent het spoor:
onthoud_spoor:=proc(expressie,f)
options remember;
normal(trace(matrixvorm(expressie,f))) end:

# Input: een produkt
# Output: die factoren uit dat produkt die meer dan een keer voorkomen
kwadraatdelers:=proc(ff) local result,i;
if type(ff,`+`) then RETURN({}) fi;
if type(ff,`^`) then RETURN({op(1,ff)}) fi;
result:={};
for i in {op(ff)} do
if type(i,`^`) then result:={op(result),op(1,i)} fi od;
result end:
```

```
# Input: een produkt en een factor
# Output: het aantal keer dat die factor voorkomt in dat produkt
multipliciteit:=proc(ff,factor) local i,v;
v:={op(ff)}; if member(factor,v) then RETURN(1) fi;
for i in v do
if type(i,`^`) and op(1,i)=factor then RETURN(op(2,i)) fi;
od; 0 end:

# Input: een stelsel vergelijkingen met slechts 1 oplossing
# Output: die ene oplossing zet hij in de waarden a.i
losop:=proc(v,x) local d,l,j,vv,i,w,result,verder;
w:=['a.dummy'$dummy=1..graad,x];
vv:={op(gbasis(v,w,plex))};result:=[];
for j from graad by -1 to 1 do verder:=true;
for i in vv while verder do
l:=leadmon(i,w,plex);
if l[2]=a.j then result:=[a.j-i/l[1],op(result)];
   vv:=expand(subs(a.j=a.j-i/l[1],vv));verder:=false
 elif type(l[2],`^`) and op(l[2])[1]=a.j then d:=op(l[2])[2];i:=i/l[1];
   vv:=expand(subs(a.j=-coeff(i,a.j,d-1)/d,vv));
   result:=[-coeff(i,a.j,d-1)/d,op(result)];verder:=false
fi od od;result
end:

onthoud_macht:=proc(a,n)
options remember;
if n=0 then RETURN(1) fi;
expand(a*onthoud_macht(a,n-1)) end:

vergelijkingen_zoeken:=proc(a,k,f) local z,w,ff,verz;
ff:=karakteristiekeveelterm(a,z,f);
lprint(nu_heb_ik_de_karakteristieke_veelterm_bepaald);
verz:=['rem(coeff(ff,z,dummy),
onthoud_macht(k,graad-dummy),x)'$dummy=0..graad];
if {op(verz)}={0} then RETURN({k}) fi;
# Nu pas ik het isomorfisme x-->x+w toe:
verz:=expand(subs(x=x+w,verz));
verz:=['expand(rem(rem(verz[dummy],
w^(graad+1-dummy),w),k,x))'$dummy=1..graad+1];
verz:={k,'coeffs(verz[dummy],w)'$dummy=1..graad+1};
# De oplossingen van dit stelsel vormen een L[x]/(k)-vectorruimte.
# De oplossingen zijn precies die a.i tjes
```

```
# waarvoor sum(a1b1+...+anbn)/k geheel is.
verz
end:

de_kleinste_ai:=proc(b) local jj,bb;bb:=indets(b);
for jj from 1 to graad do if member(a.jj,bb) then RETURN(jj) fi od;
ERROR(helaas) end:

de_ai_gesorteerd:=proc(b) local jj,v,bb;bb:=indets(b);
v:=[]; for jj from 1 to graad do
if member(a.jj,bb) then v:=[op(v),a.jj] fi od;v end:
```

4.4 De snelheid van het algoritme

Ik zal hier een lijstje polynomen geven, waarmee ik het algoritme heb getest. Voor elke f_i (behalve f_7, daar stopte de berekening vanwege een out of memory) heb ik de ring van gehelen van $L(x)[y]/(f)$ bepaald. Deze ring van gehelen wordt dan gegeven door een stelsel dat deze ring als $L[x]$-moduul voortbrengt.

De f_i's zijn zodanig gekozen dat je gemakkelijk kunt zien dat de nulpuntskrommen $C(f_i)$ singulariteiten hebben. Als je zomaar een veelterm in $L[x,y]$ kiest dan zijn er i.h.a. geen singulariteiten, en is de ring van gehelen gelijk aan $L[x,y]$.

$f_1 := y^7 + (y-2x)(y-x)(y+x^2)(y+2x^2)(y^2-x)$	299 seconden
$f_2 := y^8 + (y-x^2)(y-x)(y-2x)$	8 seconden
$f_3 := y^6 + (y-x^2-x-1)(y-x-1)(y+x^2+x+1)$	7 seconden
$f_4 := y^5 + (y-(x+1)(x^2+x+1))^3$	282 seconden
$f_5 := y^5 + (y+1)(x^2+x+1)^5$	6 seconden
$f_6 := y^6 + (y+1)(x^2+x+1)^5$	441 seconden
$f_7 := (y-1)^8 + x(x^2+x+1)^7(y-1) + (x^2+x+1)^6$	out of memory na 3300 seconden
$f_8 := (y-1)^8 + x(x^2+x+1)^7(y-1) + (x^2+x+1)^8$	26 seconden

Meer dan 75% van de rekentijd die het algoritme gebruikt gaat zitten in het berekenen van de karakteristieke veeltermen. De reden dat sommige voorbeelden veel sneller gaan dan andere voorbeelden is dan ook dat daar minder karakteristieke veeltermen bepaald hoeven te worden, of dat er karakteristieke veeltermen van minder ingewikkelde uitdrukkingen bepaald hoeven te worden.

Door de aanpassingen op het algoritme hoeft bij het voorbeeld met f_8 hoeft maar één keer een karakteristieke veelterm berekend te worden. Het algoritme vindt dan als nieuwe gehele:

$$\frac{y-1}{x^2+x+1}$$

Vervolgens gaat het algoritme het moduul

$$L[x]\{1, y, \ldots, y^7\} + L[x]\frac{y-1}{x^2+x+1}$$

uitbreiden tot een ring. In dit voorbeeld f_8 is de ring van gehelen dan al gevonden.

In het voorbeeld f_7 echter moeten veel meer karakteristieke veeltermen berekend worden, en is de ring van gehelen niet na één stap gevonden. De uitdrukkingen waarvan de karakteristieke veelterm bepaald moet worden, worden in de loop van het algoritme echter steeds ingewikkelder. Dat is de reden waarom de berekening bij f_7 uit de hand loopt.

5 Het vinden van gehelen als L niet algebraïsch gesloten is

Ik moet nog bewijzen dat de methode waarmee het programma uit paragraaf 4 vergelijkingen vindt, en oplost correct is in het geval dat de graad van k groter dan 1 is. Het geval waarin L niet algebraïsch gesloten is, maar k toch graad 1 heeft gaat hetzelfde als in paragraaf 3.

In paragraaf 5.1 zal ik laten zien hoe je de vergelijkingen vindt. Dit worden dan geen vergelijkingen over L maar over $L[x]/(k)$. In paragraaf 5.2 zeg ik dan in het kort hoe je een Gröbnerbasis over $L[x]/(k)$ kunt uitrekenen, er vanuit gaande dat je al een Gröbnerbasis algoritme hebt dat over L werkt.

5.1 Vergelijkingen vinden

Definiëer a, g en W weer als in paragraaf 3. Evenals in paragraaf 3 is W een verzameling veeltermen in $a_1, \ldots, a_n$ die nul zijn in $L[x]/(k^n)$ precies als a geheel is. Dat geldt dan voor alle a_i's in $L[x]$. In paragraaf 3.1 kon het stelsel 'is nul in een ring' vergelijkingen W worden omgezet in een stelsel 'is nul in een lichaam' vergelijkingen V.

Die stap, van (A) naar (B) kon worden gemaakt omdat ik me beperkte tot a_i's in $L[x]$ die geen x bevatten, dus a_i's in L. Ook nu wil ik een dergelijke stap maken, maar ik kan me nu niet meer beperken tot a_i's in L. De graad van k is nu immers groter dan 1, en dan is L geen volledige representantenverzameling voor $L[x]/(k)$.

Het idee is nu om het stelsel 'is nul in een ring' vergelijkingen W door middel van een ring-isomorfisme om te zetten in een stelsel 'is nul in een andere ring' vergelijkingen W'. Dit stelsel W' kan ik dan wel omzetten in een stelsel V van 'is nul in een lichaam' vergelijkingen. Om de equivalentie tussen het geheel zijn van a en het stelsel V te kunnen bewijzen zal ik weer een beperking opleggen aan de a_i's. Deze a_i's moeten in een nog te kiezen verzameling R zitten. Achteraf zal dan blijken dat V equivalent is met het geheel zijn van a voor alle $a_i \in L[x]$.

Stelling 7 *Zij w een onbepaalde. Het L-homomorfisme*

$$\phi : \; L[x]/(k^n) \longrightarrow (L[x]/(k))\,[w]/(w^n)$$

gedefiniëerd wordt door:

$$\phi(x) = x + w$$

is een isomorfisme van ringen.

Bewijs: merk op dat $x+w$ in $(L[x]/(k))\,[w]/(w^n)$ als minumumveelterm k^n heeft. De deelring $R \subset (L[x]/(k))\,[w]/(w^n)$ voortgebracht door $x+w$ is dus isomorf met $L[x]/(k^n)$. De dimensies als L-vectorruimte van R en $(L[x]/(k))\,[w]/(w^n)$ zijn beide gelijk aan graad(k) maal n. Dus zijn R en $(L[x]/(k))\,[w]/(w^n)$ hetzelfde en is ϕ een isomorfisme van ringen.

□

Lemma 3 *Als $b \in L[x]/(k) \subset (L[x]/(k))[w]/(w^n)$ dan geldt*

$$b = \phi^{-1}(b) \bmod k$$

Bewijs: van een willekeurig element $c \in L[x]/(k^n)$ zie je eenvoudig dat

$$\phi(c) = c \bmod k \;+\; \text{een veelvoud van } w$$

Kies nu $c = \phi^{-1}(b)$. Dan bevat $\phi(c)$ geen term met w, want $\phi(c) = b$, dus heb je:

$$\phi(c) = c \bmod k$$

of anders geschreven: $b = \phi^{-1}(b) \bmod k$

□

Ik kies nu een representantenverzameling R van $L[x]$ modulo k als volgt: voor elk element $b \in L[x]/(k)$ kies ik een representant in $L[x]$ die equivalent is met $\phi^{-1}(b)$ modulo k^n. Je kunt zo'n representant bijvoorbeeld met minimale graad kiezen. Dat is hier echter niet nodig.

Ik ga weer vergelijkingen zoeken, die voor $a_i \in R$ equivalent zijn met het geheel zijn van a. Eerst bereken ik de verzameling W op dezelfde manier als in paragraaf 3. In paragraaf 3 gebruikte ik het feit dat graad$(k) = 1$ alleen bij de omzetting van W naar V. Daarom kun je nu W op dezelfde manier berekenen en heb je:

$$a \text{ geheel} \iff \forall_{r \in W} r \equiv 0 \bmod k^n$$

oftewel

$$a \text{ geheel} \iff W = \{0\} \text{ in } L[x]/(k^n)$$

voor a_i's in $L[x]$.

Ik ga nu het isomorfisme ϕ toepassen op het stelsel W van 'is nul in $L[x]/(k^n)$' vergelijkingen. Het stelsel W' dat ik dan krijg is gelijk aan W met $x+w$ voor x gesubstitueerd. Dan heb je:

$$W = \{0\} \text{ in } L[x]/(k^n) \iff W' = \{0\} \text{ in } (L[x]/(k))[w]/(w^n)$$

als je $\phi(a_i)$ voor a_i substitueerd in W'. Bij beperking tot $a_i \in R$ heb je $\phi(a_i) = a_i \bmod k$, en geldt de laatste equivalentie zonder dat je in W' a_i door $\phi(a_i)$ hoeft te vervangen. Voor zulke a_i's geldt $\phi(a_i) = a_i$ in $(L[x]/(k))[w]/(w^n)$. De speciale keuze van de representantenverzameling zorgt ervoor dat het beeld van a_i onder ϕ geen term w bevat.

Onder die aanname, dat $\phi(a_i)$ $(= a_i \bmod k)$ geen w bevat kan ik het stelsel $W' = \{0\}$ in $(L[x]/(k))[w]/(w^n)$ omzetten in een stelsel V (met een groter aantal vergelijkingen) over $L[x]/(k)$, door eerst van de vergelijkingen uit W' de rest modulo w^n te nemen, en vervolgens daarvan de coëfficiënten (elke coëfficiënt wordt 1 vergelijking) naar w te nemen. Op deze manier komt w niet meer voor in de vergelijkingen. V is een stelsel vergelijkingen in $a_1, \ldots, a_n$ over $L[x]/(k)$. Ik definiëer V nu als volgt:

$$V := \bigcup_{r \in W'} \text{coeffs}(r \bmod w^n)$$

waar coeffs de verzameling coëfficiënten als veelterm in w voorstelt, en $r \bmod w^n$ de rest van r bij deling door w^n. Door nu V uit te rekenen voor onbepaalde a_i's vind je veeltermvergelijkingen in de a_i's.

Voorbeeld: Stel y heeft als minimumveelterm f, waar $f(y) = y^2 - x$. Stel ik zoek een gehele van de vorm $(a_1 + a_2 y)/(x^2 + 1)$. Dus $k = x^2 + 1$, $n = 2$, $b_1 = 1$, $b_2 = x$, en $b = \{b_1, b_2\}$. Eerst bereken ik de karakteristieke veelterm g, en die vermenigvuldig ik met k^n. Dan is

$$k^n g(z) = z^2x^4 - 2a_2x^3z + 2z^2x^2 + a_2^2x^2 - 2a_1zx^2 - 2a_2xz + 2a_1a_2x - 2a_1z + a_1^2 + z^2$$

W is de verzameling coëfficiciënten van $k^n g(z)$ als veelterm in z

$$W = \{a_1^2 + 2a_1a_2x + a2^2x^2, -2a_2x^3 - 2a_1x^2 - 2a_2x - 2a_1, x^4 + 2x^2 + 1\}$$

Dan wordt W' gelijk aan W met x door $x + w$ vervangen:

$$\begin{aligned} W' \quad = \quad & \{a_1^2 + 2a_1a_2(x+w) + a2^2(x+w)^2, (x+w)^4 + 2(x+w)^2 + 1, \\ & -2a_2(x+w)^3 - 2a_1(x+w)^2 - 2a_2(x+w) - 2a_1\} \end{aligned}$$

Nu neem ik van deze verzameling de rest modulo w^n, dan krijg ik:

$$\begin{aligned} \{x^4 + 4x^3w + 2x^2 + 4xw + 1, a_1^2 + 2a_1a_2x + 2a_1a_2w + a_2^2x^2 + 2a_2^2xw, \\ -2a_2x^3 - 6a_2x^2w - 2a_1x^2 - 4a_1xw - 2a_2x - 2a_2w - 2a_1\} \end{aligned}$$

Nu neem ik V als de verzameling coëfficiënten (als veelterm in w) van de elementen van deze laatste verzameling.

$$\begin{aligned} V \quad = \quad & \{-6a_2x^2 - 4xa_1 - 2a_2, x^4 + 2x^2 + 1, a_1^2 + 2a_1a_2x + a_2^2x^2, \\ & -2a_2x^3 - 2a_1x^2 - 2a_2x - 2a_1, 4x + 4x^3, 2a_1a_2 + 2a_2^2x\} \end{aligned}$$

V is nu een stelsel vergelijkingen over $L[x]/(k)$, modulo k is V gelijk aan:

$$\{0, a_1^2 + 2a_1a_2x - a_2^2, 2a_1a_2 + 2a_2^2x, 4a_2 - 4xa_1\}$$

Dit stelsel vergelijkingen over $L[x]/(k)$ heeft alleen een triviale oplossing, d.w.z. alle $a_i = 0$ in $L[x]/(k)$. Dit was al te verwachten vanwege stelling 6. Je ziet namelijk eenvoudig dat de kromme $C(f)$ geen singulariteiten in het affiene vlak heeft (f is hier $y^2 - x$). Een ander argument is dat k^2 geen deler van de discriminant van f is.

Achteraf kun je nu inzien dat V equivalent is met het geheel zijn van a, niet alleen voor $a_i \in R$ maar voor alle $a_i \in L[x]$. Dit volgt uit:

- De vergelijkingen in V zijn vergelijkingen modulo k.
- Het geheel zijn van a is voor de a_i's modulo k bepaald.
- R is een volledige representantenverzameling voor $L[x]$ modulo k.

5.2 Vergelijkingen oplossen

Ik ga er van uit dat ik beschik over een Gröbnerbasis algoritme dat over het lichaam L werkt. Ik wil echter de Gröbnerbasis van V over het lichaam $L[x]/(k)$ uit rekenen. Om dat te doen voeg ik de veelterm k toe aan V, en reken dan de Gröbnerbasis van $V \bigcup \{k\}$ uit. Het resultaat kan ik interpreteren als een Gröbnerbasis van V over $L[x]/(k)$.

Stel dat je ook nog wilt dat de veeltermen in de Gröbnerbasis (met als grondlichaam $L[x]/(k)$) van V kopcoëfficiënt 1 hebben. Dan moeten de veeltermen van de Gröbnerbasis (met als grondlichaam L) van $V \bigcup \{k\}$ geen x bevatten. Dit bereik je door een lexicografische ordening kiezen waarin x het kleinst is.

5.3 Een toepassing voor het geval van een algebraïsch gesloten grondlichaam

Het voorbeeld in paragraaf 2 laat een vaak (niet alleen bij ringen van gehelen) voorkomende situatie zien. Het grondlichaam is in dat voorbeeld $\overline{\mathbb{Q}}$. De coëfficiënten van f in dat voorbeeld zaten al in $\mathbb{Q}$. Hoewel de ring van gehelen van $\overline{\mathbb{Q}}(x,y)$ gevraagd werd in dat voorbeeld, waren tijdens de berekening geen algebraïsche uitbreidingen op $\mathbb{Q}$ nodig. De volgende stelling laat zien dat dat niet toevallig was:

Stelling 8 *Stel $b = \{b_1, \ldots, b_n\}$ brengt $\overline{L[x]}$ als $L[x]$-moduul voort. Dan brengt b tevens $\overline{\overline{L}[x]}$ (de ring van gehelen in $\overline{L}(x,y)$) als $\overline{L}[x]$-moduul voort.*

Bewijs: duidelijk is dat $\overline{L}[x]b$ een deelmoduul is van $\overline{\overline{L}[x]}$. Stel nu dat er een $a \in \overline{\overline{L}[x]} \setminus \overline{L}[x]b$ is. Dan is a van de vorm:

$$a = \frac{a_1 b_1 + \ldots + a_n b_n}{k}$$

met $k \in \overline{L}[x]$. Door eventueel een k-voud bij de a_i's op te tellen mag ik aannemen dat de graad (als veelterm in x) van de a_i's kleiner is dan de graad van k.

Zij k' het produkt van alle beelden van k onder de L-automorfismen van $\overline{L}$. k' is de norm van k. Dan geldt: $k|k'$ en $k' \in L[x]$.

Nu vermenigvuldig ik teller en noemer van a met k'/k, dan is a van de vorm

$$a = \frac{a'_1 b_1 + \ldots + a'_n b_n}{k'}$$

waar $a'_i = a_i k'/k$ niet allemaal 0, graad(a'_i) <graad(k') en $k' \in L[x]$. Kies i zo dat $a'_i \neq 0$. Zeg

$$a'_i = \sum_{j=0}^{\text{graad}(k')-1} a'_{ij} x^j$$

Kies dan j zo dat $a'_{ij} \neq 0 \in \overline{L}$. Door eventueel a met $1/a'_{ij}$ te vermenigvuldigen mag ik aannemen: $a'_{ij} = 1$.

Zij L' een algebraïsche uitbreiding van L die alle coëfficiënten a'_{ij} bevat. Definiëer nu s als het spoor van a over de lichaamsuitbreiding $L' : L$. Dan is s van de vorm

$$\frac{s_1 b_1 + \ldots + s_n b_n}{k'}$$

met $s_1, \ldots, s_n \in L[x]$ van graad kleiner dan de graad van k'. De coëfficiënt van x^j in s_i is gelijk aan spoor(1) en dat is ongelijk aan 0 (hier gebruik ik dat de karakteristiek 0 is). Dus $s_i \neq 0$, en van graad kleiner dan de graad van k'. Dus

$$s \notin L[x]\{b_1, \ldots, b_n\}$$

Dus s is niet geheel. Maar s is de som van geconjugeerden van a, en als a geheel is dan zijn de geconjugeerden dat ook. Dus s is niet geheel, en wel de som van gehelen. Tegenspraak.

□

Een toepassing van deze stelling is de volgende: stel er wordt een ring van gehelen gevraagd van $K(x)[y]/(f)$, waar K algebraïsch gesloten is. Neem dan L een deellichaam van K, zó dat $K = \overline{L}$, en zó dat alle coëfficiënten van f al in L zitten. Bereken een stelsel $L[x]$-voortbrengers van $\overline{L[x]}$. Dit is dan tevens een stelsel $K[x]$-voortbrengers van $\overline{K[x]}$. Het voordeel van deze aanpak is dat rekenen in een kleiner lichaam i.h.a. veel sneller gaat dan in een groot lichaam.

6 Het algoritme van Trager

Vlak voor het beëindigen van deze scriptie heb ik in het proefschrift van Trager [3] een ander algoritme gezien voor het berekenen van de ring van gehelen in een algebraïsch functielichaam. Ik zal zijn methode in deze paragraaf in het kort beschrijven, en zal de bewijzen weglaten. Trager's algoritme is gebaseerd op een algoritme van Ford en Zassenhaus [4], dat de ring van gehelen in een algebraïsch getallenlichaam berekent.

Ik zal eerst de stelling geven waarop Trager's algoritme gebaseerd is. Laat R een hoofdideaalring zijn, in dit geval $L[x]$, Ford en Zassenhaus nemen $R = \mathbb{Z}$. Zij V een integriteitsgebied, dat een eindige gehele uitbreiding van R is. Dan is V een vrij moduul met rang gelijk aan de graad van $QF(V)$ als $QF(R)$-vectorruimte, waar QF het quotiëntlichaam is. Zij $\overline{v} = [v_1, \ldots, v_n]$ een basis van V als R-moduul. De discriminant d van $\overline{v}$ brengt een ideaal in $(d) \subset V$ voort. Omdat $d \in R$ brengt d ook een ideaal in R voort, maar hier wordt met (d) een ideaal in V bedoeld.

Dat ideaal heet de discriminant van V over R. Had je een andere basis $\overline{w}$ van V genomen dan geldt: disc$(\overline{w})$ is een eenheid maal disc$(\overline{v})$. Dit brengt dus hetzelfde ideaal voort, dus de discriminant van V over R is goed gedefiniëerd.

Als $\mathbf{m}$ een ideaal is in een ring S, dan wordt $Id(\mathbf{m})$ (de 'idealizer' in het Engels) gedefiniëerd als de verzameling van alle $u \in QF(S)$ waarvoor $u\mathbf{m} \subset \mathbf{m}$. $Id(\mathbf{m})$ bevat S en is de grootste ring waarin $\mathbf{m}$ nog een ideaal is.

Onder de bovenstaande voorwaarden geldt:

Stelling 9 *V is geheel gesloten dan en slechts dan als voor elk priemideaal $\mathcal{P}$ dat de discriminant van V bevat geldt: $Id(\mathcal{P}) = V$.*

Bewijs: Zie [3]

□

Met deze stelling kun je de ring van gehelen berekenen door de idealizers te berekenen van de eindig vele priemidealen die die discriminant bevatten. Dan geldt ofwel dat al die idealizers zijn gelijk aan V, dan is V de ring van gehelen, ofwel minstens een van die idealizers is groter dan V. Dan vervang je V door die idealizer en heb je een groter deel van de ring van gehelen gevonden. Omdat de discriminant van die idealizer dan een echte deler van de discriminant van V is, moet dit proces eindigen.

Vervolgens bewijst Trager uit stelling 9 het volgende

Lemma 4 *V is geheel gesloten dan en slechts dan als de idealizer van het wortelideaal van de discriminant gelijk is aan V*

Dit geeft het volgende algoritme om de gehele afsluiting van V te vinden:

1. bepaal het wortelideaal van de discriminant van V over R
2. bereken de idealizer $\hat{V}$ van dat wortelideaal
3. als $V \subset \hat{V}$, $V \neq \hat{V}$, vervang dan V door $\hat{V}$ en ga terug naar stap 1

Vervolgens geeft hij de volgende verfijningen:

- In de eerste stap is V gelijk aan $R[y]$, waar y aan de minimumveelterm f voldoet. De discriminant van V kan dan het gemakkelijkst bepaald worden door de discriminant van de veelterm f te bepalen.
- Factoren die slechts één keer in de discriminant voorkomen kunnen worden genegeerd.
- Wanneer je van stap 3 terug naar stap 1 terug gaat, hoef je alleen rekening te houden met die factoren die in de discriminant van $\hat{V}$ minder vaak voorkomen dan in de discriminant van V.

Dit samen geeft het volgende algoritme om de gehele afsluiting van V te vinden:

0. Neem d de discriminant van f, en $k := d$
1. Neem q het product van die priemfactoren p_i waarvoor geldt: $p_i|k$, en $p_i^2|d$. Als q een eenheid is dan is V de ring van gehelen. Merk op dat voor deze stap geen volledige ontbinding van d of k nodig is.
2. Bepaal $J_q(V)$, het wortelideaal van (q) in V
3. Bepaal $\hat{V}$, de idealizer van $J_q(V)$, en de matrix M; de overgangsmatrix van de basis van $\hat{V}$ naar V.
4. Zij k de determinant van M, als k een eenheid is dan is V de ring van gehelen.
5. Neem $d := d/k^2$ en $V := \hat{V}$, en ga terug naar stap 1.

Er blijven nog twee problemen over. De eerste is het berekenen van het wortelideaal van (q), en het tweede is het bepalen van de idealizer van dat wortelideaal.

6.1 Het wortelideaal van de discriminant

De discriminant is een hoofdideaal voortgebracht door een element $d \in R$. Als $(p_1, \ldots, p_k)$ de verschillende priemfactoren van d zijn, dan is het wortelideaal van (d) de doorsnede van de wortelidealen van (p_i). Daarom nu eerst een methode om het wortelideaal van (p) te bepalen, waar (p) priem is. Zo'n ideaal wordt een p-wortelideaal genoemd. u is een element van het p-wortelideaal dan en slechts dan als alle coëfficiënten van de karakteristieke veelterm van u deelbaar zijn door p.

We definiëren het p-spoor-wortelideaal nu als de verzameling van alle $u \in V$ waarvoor $\forall_{w \in V} p | \text{spoor}(uw)$ geldt.

Stelling 10 *Als de karakteristiek van $R/(p)$ nul is, of groter dan de rang van V over R, dan is het p-spoor-wortelideaal gelijk aan het p-wortelideaal*

In ons geval is de karakteristiek 0, en kun je deze stelling toepassen.

6.2 De berekening van het p-spoor-wortelideaal

Laat $\overline{w} = [w_1, \ldots, w_n]$ een R-basis van V zijn, en laat de vector

$$\overline{u} = \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix}$$

gedefiniëerd zijn door u, uitgeschreven op de basis $\overline{w}$.

Dan is u een element van het p-spoor-wortelideaal, dan en slechts dan als:

$$\begin{aligned} \forall_{w \in V} \left[\text{spoor}(uw) \equiv 0 \bmod p\right] \quad &\Longleftrightarrow \\ \forall_{i=1\ldots n} \left[\text{spoor}(uw_i) \equiv 0 \bmod p\right] \quad &\Longleftrightarrow \\ \forall_{i=1\ldots n} \left[\textstyle\sum_{j=1}^{n} u_j \text{spoor}(w_j w_i) \equiv 0\right] \bmod p \end{aligned}$$

Met behulp van de spoor matrix $SP_{\overline{w}}$ kun je dit schrijven als:

$$SP_{\overline{w}} \cdot \overline{u} = \begin{pmatrix} \text{spoor}(w_1^2) & \cdots & \text{spoor}(w_1 w_n) \\ \vdots & & \vdots \\ \text{spoor}(w_n w_1) & \cdots & \text{spoor}(w_n^2) \end{pmatrix} \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix} \in pR^n \qquad (3)$$

waar pR^n de verzameling van elementen die deelbaar zijn door p is. Nu is $J_p(V)$ gelijk aan de verzameling oplossingen $u \in V$ waarvoor de u_i in R zitten en aan (3) voldoen. Er geldt:

$$J_q(V) = \bigcap_{p|q} J_p(V)$$

dus $J_q(V)$ wordt bepaald door de verzameling oplossingen $u_i \in R$ van (3), maar dan met pR^n vervangen door qR^n.

Om nu te garanderen dat alle oplossingen u_i van (3) in R zitten moet je nog wat vergelijkingen toevoegen. Neem M_q gelijk aan de matrix $SP_{\overline{w}}$ met de matrix qI_n eronder gezet. Dan is $J_q(V)$ de verzameling van alle $u \in QF(V)$ waarvoor geldt $M_q \cdot \overline{u} \in qR^{2n}$.

In onze toepassing is $R = L[x]$, dat is een euclidische ring. Daarom kun je van M_q een driehoeksmatrix maken met uitsluitend elementaire rij operaties. Dit heet de Hermietse rij reductie, en lijkt op de Gauss eliminatie, maar dan voor euclidische ringen i.p.v. voor lichamen. Trager geeft een eenvoudig algoritme voor de Hermietse reductie. Daarmee krijg je dan een vierkante matrix $\hat{M}$. Er geldt dan:

$$u \in J_q(V) \iff \hat{M}\overline{u} \in qR^n$$

Een R-basis van $J_q(V)$ wordt dan gegeven door de kolommen van

$$\left(\frac{1}{q}\hat{M}\right)^{-1}$$

6.3 Het berekenen van de idealizer

Laat $(m_1, \ldots, m_n)$ een R-basis voor een ideaal $\mathbf{m}$ in V zijn, gevraagd is nu een R-basis van de idealizer van $\mathbf{m}$. De idealizer van $\mathbf{m}$ is de verzameling $u \in QF(V)$ waarvoor geldt $u\mathbf{m} \subset \mathbf{m}$. Laat $\overline{v} = (v_1, \ldots, v_n)$ een vast gekozen R-basis van V zijn. Er geldt

$$u \in \mathrm{Id}(\mathbf{m}) \iff \forall_{1 \le i \le n} um_i = \sum_j r_{ij} m_j$$

waar r_{ij} elementen van R moeten zijn.

Vermenigvuldiging met m_i is een $QF(R)$-lineaire afbeelding in $QF(V)$. Laat M_i de matrix zijn behorende bij deze lineaire afbeelding, vanaf de basis $\overline{v}$ naar de basis $(m_1, \ldots, m_n)$. Als dan $u = \sum_j u_j v_j$ dan is

$$um_i \in \mathbf{m} \iff M_i \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix} \in R^n$$

Neem nu M de $n^2 \times n$ matrix die je krijgt als je alle M_i onder elkaar zet. Dan is de idealizer de verzameling $u \in QF(V)$ waarvoor Mu een vector met n^2 elementen van R is. Door Hermietse rij reductie kun je de laatste $n^2 - n$ kolommen nul maken. Zij $\hat{M}$ de overgebleven $n \times n$ matrix, dan geven de kolommen van $\hat{M}^{-1}$ een basis van de idealizer.

Referenties

[1] Dr. F.J. Keune *algebraïsche getaltheorie* 1981 Katholieke Universiteit Nijmegen

[2] Prof. Dr. J.H.M. Steenbrink *College algebraïsche krommen* 1991 Katholieke Universiteit Nijmegen

[3] B.M. Trager *Integration of Algebraic Functions* 1984 proefschrift, Dpt. of EECS, Massachusetts Institute of Technology

[4] D.J. Ford *On the Computation of the Maximal Order in a Dedekind Domain* 1978 proefschrift, Ohio State University, Dept. of Mathematics