UNIVERSITA' DEGLI STUDI DI GENOVA

Dipartimento di Matematica

Anno Accademico 2000/2001

TESI DI LAUREA

Applicazioni dell'Algebra Computazionale allo studio di alcuni Operatori Differenziali

Relatore
Prof. Lorenzo Robbiano

Correlatore
Prof. Giancarlo Mauceri

Candidato: Alberto Damiano

Dedicare questa tesi a tutte le persone a cui vorrei sarebbe assolutamente impossibile. Gli avvenimenti di questi ultimi anni della mia vita hanno fatto sì che i miei sentimenti per molte persone che erano accanto a me si trasformassero e si consolidassero notevolmente. Non smetterò mai di ritenermi molto fortunato per tutto quello che ho avuto e continuo ad ottenere dagli altri ogni giorno.

Primi tra tutti vorrei ringraziare i miei genitori Roberto ed Anna che mi hanno permesso di realizzare uno dei miei sogni: laurearmi in matematica. Senza il loro continuo appoggio ed entusiasmo non sarei mai potuto andare avanti. Sono certo che il loro sostegno nel mio percorso di studi futuri sarà sempre determinante.

Tengo in modo particolare a ringraziare il professor Robbiano per la sua estrema pazienza e l'amore per la materia che è riuscito a trasmettermi durante i corsi. Grazie anche a tutta la sua "squadra" di CoCoA, ad Anna Bigatti per la sua disponibilità, a John Abbot e Massimo Caboara per la cortesia e la simpatia dimostratami. Non vorrei mancare di ringraziare anche tutti i docenti che mi hanno seguito durante il percorso di laurea. Ricorderò sempre con grande affetto G. Mauceri, G. Monti Bragadin, E. Carletti, F. De Mari, S. Testa, M. Pedroni, G. Caviglia, M. E. Rossi, G. Valla, A. V. Geramita, M. Beltrametti, V. Del Prete, M. P. Cavaliere, per avermi fatto appassionare alla loro materia, e avermi fatto capire quanta profonda umanità si celi anche dietro alle più fredde argomentazioni. Se già la professoressa A. Ramondo durante il periodo del liceo mi aveva fatto intuire tutta questa bellezza, loro me lo hanno confermato ulteriormente.

Ma la persona a cui più di tutte vorrei dedicare il mio lavoro e che saluto con grandissimo affetto è la mia amica Maria, perché ha vissuto ogni giorno insieme a me non solo la preparazione degli esami, lo stress delle lezioni e la vita del Dipartimento, ma ha anche condiviso con me alcuni momenti davvero importanti per la mia maturazione, non solo scolsatica. Sono certo che entrambi ricorderemo, non senza un po' di nostalgia, tutto il nostro percorso di questi quattro anni. Fondamentali, pur nella loro semplicità, sono stati i riti di "pausa-caffè" seduti sulle scale del quinto piano insieme a Michela e Sabrina! La loro amicizia così come quella di Fabrizio, Edoardo, Andrea e tutti gli altri miei compagni di corso è un regalo che vale più di qualunque altro.

Grazie a tutti i miei amici del Giovedì sera, con il loro calore e il loro affetto sono riuscito anche ad affrontare con più serenità gli impegni universitari. Un grazie particolare a Simone per avermi dato sempre nuovi spunti di riflessione, a Stefania per la sua incredibile simpatia, a Daniela per aver sempre creduto in me, ad Aldo per avermi dimostrato che la differenza di età è una questione puramente numerica, a Luisa per avermi sempre considerato come un figlio e a Francesca e Antonio per aver reso più sopportabili anche i momenti di maggiore stress. Come non citare anche i miei coinquilini che in questi quattro anni genovesi hanno dovuto sopportare un matematico in casa: Matteo, Cristina, Marco e Andrea, Andrea "Benfe", Davide, Valerio.

Gli amici veri avranno sempre un posto nel mio cuore, e quindi concludo con una dedica speciale a Jeorghia che è accanto a me da quando ancora, forse, non sapevo nemmeno contare.

Indice

1. Introduzione e Motivazioni

Un classico problema dell' analisi è la soluzione di sistemi di equazioni differenziali. Le difficoltà di tale problema sono molteplici, si pensi ad esempio che già nel caso di una sola variabile reale è possibile cercare esplicitamente le soluzioni solo per classi molto ristrette di equazioni. I problemi aumentano notevolmente nel caso di più variabili e di equazioni alle derivate parziali. In certi casi, però si possono dedurre alcune informazioni sulla classe di funzioni che soddisfa un certo sistema di equazioni differenziali studiando proprietà algebriche dell'operatore differenziale associato. Nel nostro lavoro ci soffermeremo in particolare sullo studio dell'operatore differenziale detto di Cauchy-Fueter, che genera un sistema omogeneo del primo ordine di equazioni differenziali alle derivate parziali, rispetto a quattro variabili reali.

Le proprietà algebriche dell'operatore saranno indagate mediante alcuni strumenti di algebra omologica, in particolare utilizzando i funtori Hom ed Ext, opportunamente interpretati dal punto di vista computazionale. Lo studio di tali oggetti matematici, e la possibilità di poterli rappresentare sul calcolatore, è stato suggerito da alcuni problemi che nascono appunto nel campo dell'analisi, in particolare analisi quaternionica, in riferimento ad alcuni articoli di ricerca recenti (si vedano [SS] e [DS]) dovuti, fra gli altri, al professor Daniele Struppa e alla professoressa Irene Sabadini. In tali articoli si affrontano alcune problematiche tipiche legate alle funzioni di variabile quaternionica, quali ad esempio la regolarità, e si osserva come esse possano essere interpretate dal punto di vista puramente algebrico. In particolare il funtore Ext gioca un ruolo fondamentale nella deduzione di alcune proprietà analitiche delle funzioni regolari, che altro non sono che il nucleo dell'operatore differenziale di Cauchy-Fueter. Il secondo capitolo della nostra tesi introdurrà brevemente alcuni concetti fondamentali relativi alle funzioni regolari e presenterà i principali risultati che mettono in luce lo stretto legame che è stato individuato in quest'ultimo secolo tra l'algebra omologica e l'analisi quaternionica.

Nel terzo capitolo faremo un passo indietro per poter inquadrare le definizioni dei funtori Hom ed Ext in un contesto algebrico appropriato. Verranno presentate le definizioni formali degli oggetti necessari ai nostri scopi, quali le sizigie e le risoluzioni libere di moduli finitamente generati sull'anello dei polinomi. Tali concetti costituiscono le fondamenta per la costruzione non solo formale dei funtori Hom ed Ext, ma anche computazionale. Grazie infatti alla potenza delle Basi di Gröbner che permettono di esplicitare il calcolo del modulo delle sizigie e quindi delle risoluzioni libere, almeno nel caso di moduli omogenei, saremo in grado di dedurre gli algoritmi appropriati per la rappresentazione dei funtori Hom ed Ext. Tali funtori si applicano a moduli finitamente generati sull'anello dei polinomi, e forniscono come risultato un modulo dello stesso tipo. Di tale modulo verrà costruita algoritmicamente una presentazione, cioè un isomorfismo con un quoziente di un modulo libero.

Tale scelta di rappresentazione si è rivelata particolarmente utile e adatta agli strumenti computazionali in nostro possesso. Durante tutto il percorso, per non rendere la trattazione ecccessivamente pesante, si affiancheranno alle definizioni astratte alcuni esempi che illustrano come sia possibile rendere concreti e maneggiabili gli oggetti algebrici in gioco. Gli esempi sono stati elaborati con l'utilizzo di CoCoA, il software prodotto a Genova che permette di gestire il calcolo simbolico, in particolare l'algebra commutativa dei polinomi.

Nel quarto capitolo verrà presentato l'algoritmo che permette la rappresentazione del funtore Ext, così come la definizione stessa suggerisce. In particolare illustreremo nel dettaglio il pacchetto di funzioni scritto con CoCoA che è stato messo a punto per il calcolo di Ext. Nel caso di input omogeneo si è trattato semplicemente di trasporre l'algoritmo in linguaggio CoCoA, mentre nel caso di dati non omogenei si è dovuto ricorrere ad un trucco per poter simulare, con un isomorfismo, la gestione degli shift, che nell'attuale versione di CoCoA non è ancora possibile in modo automatico.

In conclusione, nell'ultimo capitolo, vedremo alcuni esempi di applicazione non solo di tipo puramente algebrico, ma con particolare riferimento a quelli che nascono dai problemi di analisi quaternionica citati nel secondo capitolo, in modo da giustificare le motivazioni del nostro lavoro.

2. Funzioni regolari di variabile quaternionica

Questo capitolo è dedicato alla presentazione di alcuni concetti di base relativi alle funzioni di variabile quaternionica, in particolare vengono introdotti gli operatori differenziali che risultano associati a particolari sistemi di equazioni differenziali. Nel primo paragrafo vedremo come introdurre una opportuna definizione di differenziabilità per le funzioni di variabile quaternionica, prendendo come spunto la caratterizzazione della differenziabilità su $\mathbb{C}$ data dalle condizioni di Cauchy-Riemann. Le funzioni differenziabili sul corpo dei quaternioni $\mathbb{H}$, dette funzioni regolari, saranno infatti soluzioni di un opportuno sistema omogeneo di equazioni differenziali del primo ordine. Nei paragrafi successivi si metteranno invece in luce i legami tra alcune proprietà analitiche di tale classe di funzioni, e le proprietà algebriche degli operatori differenziali associati al sistema di equazioni. I riferimenti per le dimostrazioni delle proposizioni riportate nel capitolo saranno sempre gli articoli [SS] e [ABL], salvo i casi in cui riporteremo espressamente la prova oppure indicheremo un rifermento più preciso.

2.1 Operatore di Cauchy-Fueter

Cominceremo con il considerare funzioni di una sola variabile quaternionica, rimandando l'estensione al caso di più variabili quaternioniche alla fine del paragrafo.

Definizione 2.1.1. Ricordiamo che il **corpo dei quaternioni** $\mathbb{H}$ è un' algebra su $\mathbb{R}$ di dimensione 4, generata dagli elementi 1,i,j,k con le seguenti relazioni:

$$\begin{cases} i^2 = j^2 = k^2 = 1 \\ ij = -ji = k \\ jk = -kj = i \\ ki = -ik = j \end{cases}$$

Si tratta quindi di un algebra non commutativa i cui elementi sono del tipo seguente:

$$q = x_0 + x_1 i + x_2 j + x_3 k \quad x_i \in \mathbb{R} \quad \forall i = 0, \ldots, 3$$

così come è possibile studiare l'analisi su $\mathbb{R}$ o su $\mathbb{C}$ (su $\mathbb{R}^n$ o $\mathbb{C}^n$ nel caso di più variabili) possiamo studiare il calcolo differenziale di funzioni definite su un aperto U di $\mathbb{H}$, a valori in $\mathbb{H}$

$$f : U \longrightarrow \mathbb{H}$$

Se indichiamo con $c : \mathbb{R}^4 \longrightarrow \mathbb{H}$ l'applicazione lineare che realizza l'isomorfismo di $\mathbb{R}^4$ e $\mathbb{H}$ come $\mathbb{R}$-spazi vettoriali, cioè

$$c(e_0) = 1 \quad c(e_1) = i \quad c(e_2) = j \quad c(e_3) = k$$

possiamo "vedere" f come una funzione da $\mathbb{R}^4$ in $\mathbb{R}^4$, dove $\{e_0, \ldots, e_4\}$ rappresenta la base canonica di $\mathbb{R}^4$:

$$\begin{array}{ccc} H & \xrightarrow{f} & H \\ \uparrow{\scriptstyle c} & & \downarrow{\scriptstyle c^{-1}} \\ R^4 & \xrightarrow{\widetilde{f}} & R^4 \end{array}$$

dove $\widetilde{f} = c^{-1} \circ f \circ c$

Spesso indicheremo con $x_0, \ldots, x_3$ le variabili reali e confonderemo f con $\widetilde{f}$ scrivendo indifferentemente $f(q)$ oppure $f(x_0, \ldots, x_3)$, così come $f_0, \ldots, f_3$ indicheranno le componenti scalari reali di $\widetilde{f}$. Vogliamo ora introdurre una definizione appropriata di differenziabilità su $\mathbb{H}$ così come si fa usualmente su $\mathbb{R}$ e su $\mathbb{C}$. Consideriamo per ora una funzione $f(t)$, di variabile t reale o complessa, e ricordiamo la seguente

Definizione 2.1.2. f si dice **differenziabile** (in senso classico) se esiste

$$\lim_{h \to 0} \frac{f(t+h) - f(t)}{h}$$

Tale numero, se esiste, viene detto derivata di f rispetto a t.

Come ha analizzato Fueter in alcuni suoi lavori, in particolare [FU1], un'analoga definizione nel caso di $\mathbb{H}$ condurrebbe ad ottenere una classe di funzioni differenziabili estremamente esigua. Vale infatti la seguente

Proposizione 2.1.3. *Le uniche funzioni differenziabili su $\mathbb{H}$ (nel senso classico) sono le funzioni lineari, cioè quelle del tipo $f(q) = qa + b$, dove $a, b \in \mathbb{H}$*

Alla luce della proposizione 2.1.3 si rende dunque necessaria una nozione di derivabilità diversa da quella appena introdotta. Prenderemo infatti spunto dalla nozione di olomorfia per le funzioni di variabile complessa, arrivando a definire delle condizioni analoghe a quelle di Cauchy-Riemann. Ricordiamo brevemente cosa si intende quando si dice che una funzione $f : \mathbb{C} \longrightarrow \mathbb{C}$ è **olomorfa**. Se indichiamo con $z = x + iy$ la variabile complessa e con u e v rispettivamente la parte reale e la parte immaginaria di f, in modo che $f = u + iv$, possiamo costruire l'operatore di derivazione

$$\frac{\partial}{\partial \overline{z}} = \frac{\partial}{\partial x} + i \frac{\partial}{\partial y}$$

In tal modo abbiamo la seguente catena di uguaglianze:

$$\frac{\partial f}{\partial \overline{z}} = \frac{\partial f}{\partial x} + i\frac{\partial f}{\partial y} = \frac{\partial u + iv}{\partial x} + i\frac{\partial u + iv}{\partial y} = (\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}) + i(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x})$$

Imponendo la condizione $\frac{\partial f}{\partial \overline{z}} = 0$ otteniamo allora il sistema

$$\begin{cases} \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} &= 0 \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} &= 0 \end{cases}$$

in cui riconosciamo le ben note condizioni di Cauchy-Riemann. È noto che tali condizioni sono equivalenti alla nozione classica di differenziabilità su $\mathbb{C}$ così come è stata introdotta nella definizione 2.1.2. Sembra quindi ragionevole utilizzare quest'idea per definire una nozione di "olomorfia" su $\mathbb{H}$, partendo da un operatore di derivazione rispetto alle variabili reali. Gli operatori di derivazione rispetto alle 4 variabili reali $x_0, \ldots, x_3$ saranno indicati con $\frac{\partial}{\partial x_0}, \ldots, \frac{\partial}{\partial x_3}$. È ora necessario spiegare che cosa si intende per derivazione rispetto ad una variabile quaternionica q.

Definizione 2.1.4. Si dice **operatore (sinistro) di Cauchy-Fueter** il seguente operatore di derivazione:

$$\frac{\partial}{\partial \overline{q}} = \frac{\partial}{\partial x_0} + i\frac{\partial}{\partial x_1} + j\frac{\partial}{\partial x_2} + k\frac{\partial}{\partial x_3}$$

viene detto "sinistro" per distinguerlo dall'operatore (destro)

$$\frac{\partial_r}{\partial \overline{q}} = \frac{\partial}{\partial x_0} + \frac{\partial}{\partial x_1}i + \frac{\partial}{\partial x_2}j + \frac{\partial}{\partial x_3}k$$

Ricordiamo infatti che i quaternioni formano un' algebra non commutativa e che dunque i due operatori appena definiti sono diversi.

Scrivendo quindi $\frac{\partial f}{\partial \overline{q}}$ intendiamo $\frac{\partial f(x_0,\ldots,x_3)}{\partial x_0} + \frac{\partial f(x_0,\ldots,x_3)}{\partial x_1}i + \frac{\partial f(x_0,\ldots,x_3)}{\partial x_2}j + \frac{\partial f(x_0,\ldots,x_3)}{\partial x_3}k$. La teoria che deriverebbe dallo studio dell'operatore destro al posto di quello sinistro è del tutto simmetrica e quindi non ce ne occuperemo. A questo punto siamo in grado di dire cosa intendiamo per funzione regolare su $\mathbb{H}$.

Definizione 2.1.5. Sia U un aperto di $\mathbb{H}$ e sia $f : U \longrightarrow \mathbb{H}$ una funzione differenziabile in senso reale (cioè come funzione su $\mathbb{R}^4$). Diciamo che f è **regolare** su U se vale

$$\frac{\partial f(q)}{\partial \overline{q}} = 0 \quad \forall q \in U$$

così come nel caso complesso, tale condizione da luogo a un sistema di equazioni differenziali alle derivate parziali, rispetto alle variabili reali, che scriviamo qui di seguito:

$$\begin{cases} \frac{\partial f_0}{\partial x_0} - \frac{\partial f_1}{\partial x_1} - \frac{\partial f_2}{\partial x_2} - \frac{\partial f_3}{\partial x_3} &= 0 \\ \frac{\partial f_0}{\partial x_1} + \frac{\partial f_1}{\partial x_0} - \frac{\partial f_2}{\partial x_3} + \frac{\partial f_3}{\partial x_2} &= 0 \\ \frac{\partial f_0}{\partial x_2} + \frac{\partial f_1}{\partial x_3} + \frac{\partial f_2}{\partial x_0} - \frac{\partial f_3}{\partial x_1} &= 0 \\ \frac{\partial f_0}{\partial x_3} - \frac{\partial f_1}{\partial x_2} + \frac{\partial f_2}{\partial x_1} + \frac{\partial f_3}{\partial x_0} &= 0 \end{cases} \tag{1}$$

Prima di indagare le proprietà del sistema (1), detto **sistema di Cauchy-Fueter**, osserviamo che vi sono strette analogie tra le funzioni olomorfe su $\mathbb{C}$ e le funzioni regolari su $\mathbb{H}$. Ad esempio, sembra lecito chiedersi se anche in $\mathbb{H}$ valgano delle formule integrali analoghe a quelle di Cauchy per le funzioni olomorfe, oppure se sia possibile uno sviluppo in serie di potenze. Alla prima domanda risponderemo dopo aver definito un opportuno nucleo:

$$G(q) := \frac{q^{-1}}{|q|^2}$$

detto **nucleo di Cauchy-Fueter**. Definiamo anche la seguente 3-forma differenziale:

$$D_q := dx_1 \wedge dx_2 \wedge dx_3 - i dx_0 \wedge dx_2 \wedge dx_3 + j dx_0 \wedge dx_1 \wedge dx_3 - k dx_0 \wedge dx_1 \wedge dx_2$$

Con le notazioni appena introdotte possiamo enunciare il seguente teorema che racchiude le formule integrali di Cauchy-Fueter:

Teorema 2.1.6 (Formule integrali di Cauchy-Fueter). *Sia U un aperto di $\mathbb{H}$ e sia $f : U \longrightarrow \mathbb{H}$ una funzione regolare su U. Sia V una varietà quadrimensionale orientata con bordo C^1 relativamente compatta in U, e sia $q_0 \in V$. Allora valgono le formule:*

$$\int_{\partial V} D_q f(q) = 0 \tag{I}$$

$$f(q_0) = \frac{1}{2\pi^2} \int_{\partial V} G(q - q_0) D_q f(q) \qquad \text{(II)}$$

Le formule (I) e (II) sono formalmente identiche a quelle di Cauchy nel caso di una variabile complessa z, basta effettuare le sostituzioni:

$$D_q \longrightarrow dz = dx + idy$$

$$G(q) \longrightarrow G(z) = \frac{1}{z}$$

Fino a questo punto quindi la teoria della variabile quaternionica e la teoria della variabile complessa sembrano simili. Tuttavia i problemi si presentano quando si tenta di studiare uno sviluppo in serie di potenze. Mentre infatti una funzione di variabile complessa è olomorfa se e solo se è analitica (cioè sviluppabile in serie di potenze), questo non è affatto vero per le funzioni su $\mathbb{H}$. Basti osservare che ad esempio la stessa funzione potenza

$$f(q) = q^t \quad , \quad dove \quad t \in \mathbb{N}$$

non è regolare per nessun t, nemmeno per $t = 1$! Esiste comunque la possibilità di espandere in serie una f regolare utilizzando come termini opportuni polinomi omogenei nelle x_i che nel caso $n = 1$ di una variabile sono

$$p_1 = x_1 - x_0 i \quad , \quad p_2 = x_2 - x_0 j \quad , \quad p_3 = x_3 - x_0 k,$$

ma noi non ce ne occuperemo in quanto non rientrerebbe nei nostri scopi. Piuttosto, vediamo come ultima osservazione come estendere il concetto di regolarità al caso di più variabili nel modo più naturale possibile.

Definizione 2.1.7. Sia $U \subseteq \mathbb{H}^n$ un aperto, $f : U \longrightarrow \mathbb{H}$ una funzione nelle n variabili quaternioniche $q_1, \ldots, q_n$ differenziabile come funzione su $\mathbb{R}^{4n}$. Diciamo che f è **regolare** su U se

$$\frac{\partial f(\underline{q})}{\partial \overline{q_1}} = \ldots = \frac{\partial f(\underline{q})}{\partial \overline{q_1}} = 0 \quad \forall \underline{q} \in U$$

dove $\underline{q} = (q_1, \ldots, q_n)$ e l'operatore $\frac{\partial}{\partial \overline{q_i}}$ è definito in modo ovvio come

$$\frac{\partial}{\partial \overline{q_i}} = \frac{\partial}{\partial x_{0i}} + i\frac{\partial}{\partial x_{1i}} + j\frac{\partial}{\partial x_{2i}} + k\frac{\partial}{\partial x_{3i}}$$

essendo $q_l = x_{0l} + ix_{1l} + jx_{2l} + kx_{3l} \quad \forall l = 1, \ldots, n$.

In pratica $f(q_1, \ldots, q_n)$ è regolare se lo è rispetto ad ognuna delle variabili quaternioniche in gioco. Ancora una volta questa nozione di regolarità si traduce in un sistema di equazioni differenziali alle derivate parziali del tipo

$$A \cdot \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{2}$$

dove la matrice A è una matrice a blocchi

$$A = \begin{bmatrix} A_1 \\ \vdots \\ A_n \end{bmatrix}$$

in cui ciascuno dei blocchi A_l è una matrice di operatori differenziali rispetto alle variabili reali, del tutto simile a quello che interviene nel caso $n = 1$, cioè

$$A_l = \begin{pmatrix} \frac{\partial}{\partial x_{0l}} & -\frac{\partial}{\partial x_{1l}} & -\frac{\partial}{\partial x_{2l}} & -\frac{\partial}{\partial x_{3l}} \\ \frac{\partial}{\partial x_{1l}} & \frac{\partial}{\partial x_{0l}} & -\frac{\partial}{\partial x_{3l}} & \frac{\partial}{\partial x_{2l}} \\ \frac{\partial}{\partial x_{2l}} & \frac{\partial}{\partial x_{3l}} & \frac{\partial}{\partial x_{0l}} & -\frac{\partial}{\partial x_{1l}} \\ \frac{\partial}{\partial x_{3l}} & -\frac{\partial}{\partial x_{2l}} & \frac{\partial}{\partial x_{1l}} & \frac{\partial}{\partial x_{0l}} \end{pmatrix} \quad l = 1, \ldots, n$$

Risolvere quindi il sistema (2) significa in realtà risolvere n sistemi "alla Cauchy-Feuter" del tipo (1). Il nostro prossimo passo sarà quello di trasformare tale problema analitico in un problema di tipo algebrico. Le proprietà delle soluzioni dei sistemi di Cauchy-Feuter, cioè le funzioni regolari, saranno collegate alle proprietà algebriche di opportuni moduli finitamente generati.

2.2 Complesso di Cauchy-Fueter

Fissiamo ora la notazione che useremo da qui fino alla fine del paragrafo. Ricordiamo che le funzioni su $\mathbb{H}^n$ a valori in $\mathbb{H}$ verranno sempre trattate come funzioni su $\mathbb{R}^{4n}$. Tale accorgimento ci permette di avvalerci degli strumenti dell'algebra commutativa, in quanto per le funzioni differenziabili su $\mathbb{R}^{4n}$ vale la relazione:

$$\frac{\partial}{\partial x_i}\frac{\partial}{\partial x_j} = \frac{\partial}{\partial x_j}\frac{\partial}{\partial x_i} \quad \forall i, j$$

cosa che invece non sarebbe vera se considerassimo le derivate rispetto alle varibili quaternioniche:

$$\frac{\partial}{\partial \overline{q_i}}\frac{\partial}{\partial \overline{q_j}} \neq \frac{\partial}{\partial \overline{q_j}}\frac{\partial}{\partial \overline{q_i}} \quad \forall i, j$$

Indichiamo quindi con $P := \mathbb{C}[x_0, \ldots, x_{4n-1}]$ l'anello dei polinomi nelle $4n$ indeterminate $x_0, \ldots, x_{4n-1}$ a coefficienti in $\mathbb{C}$ e con $A = (A_{ij})$ una matrice ad elementi in P di dimensione $r_1 \times r_0$ in cui ogni elemento è un polinomio omogeneo di grado 1. Tale matrice rappresenta un morfismo di P-moduli liberi:

$$A: \quad P^{r_0} \longrightarrow P^{r_1}$$

che continueremo a chiamare ancora A con un piccolo abuso. Tale notazione, che confonde una matrice con l'applicazione ad essa associata, è comunque coerente con il modo di definire una mappa lineare su CoCoA, dove infatti assegnare un morfismo significa assegnare la sua matrice associata rispetto alle basi canoniche. Indichiamo poi con

$$D := (-i\frac{\partial}{\partial x_0}, \ldots, -i\frac{\partial}{\partial x_{4n-1}})$$

l'operatore di derivazione rispetto alle $4n$ variabili reali. Tale operatore agisce sulle funzioni C^∞ su $\mathbb{R}^{4n}$. Con S indicheremo il fascio delle funzioni C^∞ di variabile reale, cioè se U è un aperto di $\mathbb{R}^{4n}$, si ha:

$$\Gamma(U, S) = S(U) := C^\infty(U, \mathbb{R}^4)$$

Abbiamo visto che le funzioni regolari su $\mathbb{H}^n$ possono essere studiate attraverso il sistema di Cauchy-Fueter, che è un sistema di equazioni differenziali per le funzioni $f_0, \ldots, f_3$ che di fatto sono funzioni differenziabili su $\mathbb{R}^{4n}$. Quindi ha senso limitarci a considerare D come operatore su S. In realtà ci limiteremo, per convenienza, alle funzioni appartenenti alla *classe di Schwarz* $\mathcal{S}$, cioè

C^∞ e a decrescenza rapida. Indichiamo ora con $A(D)$ l'operatore ottenuto applicando agli elementi della matrice A la sostituzione:

$$x_i \quad \longrightarrow \quad \frac{\partial}{\partial x_i}$$

in tal modo $A(D)$ diventa un operatore

$$A(D): \quad S(U)^{r_0} \longrightarrow S(U)^{r_1} \quad \text{dove U è un aperto di} \quad \mathbb{R}^{4n}$$

In pratica $A(D)$ è la trasformata di Fourier della matrice A intesa come operatore di moltiplicazione. Infatti coniugando l' operatore di moltiplicazione per x_j con la trasformata di Fourier, che è un isomorfismo sulla classe di Schwarz e quindi mantiene le proprietà algebriche dell'operatore, si ottiene l'operatore di derivazione rispetto a $\frac{\partial}{\partial x_j}$, e viceversa. Lo stesso dicasi per una matrice i cui elementi sono polinomi omogenei: essa rappresenta un operatore di moltiplicazione su $\mathcal{S}$ e quindi coniugando tale operatore con la trasformata di Fourier si ottiene il rispettivo operatore di derivazione. Riassumiamo quanto appena detto in una definizione, considerando in particolare il caso che ci interessa, cioè il caso in cui $A(D)$ genera il sistema di equazioni differenziali di Cauchy-Fueter. Per semplicità fissiamo $n = 1$.

Definizione 2.2.1. Sia $P := \mathbb{C}[x_0, \ldots, x_3]$, A la matrice data da:

$$A := \begin{pmatrix} x_0 & -x_1 & -x_2 & -x_3 \\ x_1 & x_0 & -x_3 & x_2 \\ x_2 & x_3 & x_0 & -x_1 \\ x_3 & -x_2 & x_1 & x_0 \end{pmatrix}$$

Sia $\mathcal{S}$ la classe di Schwarz, $D := (-i\frac{\partial}{\partial x_0}, \ldots, -i\frac{\partial}{\partial x_3})$. Allora, commettendo un piccolo abuso, chiameremo l'operatore $A(D)$ **operatore di Cauchy-Fueter**.

Tale definizione può essere estesa ovviamente al caso $n > 1$ e anche al caso in cui A è una qualunque matrice i cui elementi sono polinomi di primo grado nelle x_i, o più in generale polinomi omogenei dello stesso grado. Studiare la classe $\Re(U)$ delle funzioni regolari su U significa quindi studiare il nucleo dell'operatore $A(D)$ di Cauchy-Fueter, infatti tali funzioni soddisfano la relazione matriciale:

$$A(D) \cdot \underline{f} = \underline{0} \qquad (*)$$

dove $\underline{f}$ e $\underline{0}$ sono vettori colonna di $\mathbb{R}^4$. Il sistema di equazioni $(*)$ altro non è che il sistema di Cauchy-Fueter. In simboli possiamo scrivere quindi:

$$\text{Ker } (A(D)_{|_{\mathcal{S}(U)}}) = \Re(U)$$

dove $\Re$ è il fascio delle funzioni regolari su $\mathbb{H}^n$. Con una notazione un po' più sintetica, se indichiamo con $\mathcal{S}^A(U)$ il nucleo dell'operatore $A(D)$ su $\mathcal{S}(U)$ e con $\mathcal{S}^A$ il fascio associato, possiamo scrivere

$$\mathcal{S}^A = \Re$$

Studiare le proprietà algebriche di $A(D)$ quindi è utile per dedurre informazioni sulla classe delle funzioni regolari. Le proprietà algebriche di $A(D)$ sono le stesse di quelle della matrice A, data la commutatività delle indeterminate che riflette quella delle derivate parziali, quindi da qui in avanti cercheremo di studiare l'operatore $A : P^{r_0} \longrightarrow P^{r_1}$. Vale innanzitutto la seguente

Proposizione 2.2.2. *Sia* $P := \mathbb{C}[x_0, \dots, x_{4n-1}]$, $A : P^4 \longrightarrow P^{4n}$ *un'applicazione lineare. Indichiamo con* M_n *il conucleo della mappa trasposta:*

$$M_n := \text{Coker } ({}^tA)$$

Allora vale il seguente isomorfismo

$$\text{Ker } (A) \cong \text{Hom}(M_n, P)$$

Dimostrazione. Cominciamo a considerare la seguente sequenza esatta:

$$0 \longrightarrow \text{Ker } ({}^tA) \xrightarrow{\alpha} P^{4n} \xrightarrow{{}^tA} P^4 \xrightarrow{\beta} M_n \longrightarrow 0$$

dove α è l'inclusione naturale del nucleo di ${}^tA$ e β è la proiezione naturale sul quoziente $P^4/\text{Im } ({}^tA)$. Ora applichiamo alla sequenza il funtore controvariante $\text{Hom}(\cdot, P)$. Per una sua definizione precisa rimandiamo al capitolo successivo, osserviamo solo che grazie alla proposizione 3.2.8 la sequenza che ne otteniamo

$$0 \longrightarrow \text{Hom}(M_n, P) \xrightarrow{\beta^*} P^4 \xrightarrow{A} P^{4n} \xrightarrow{\alpha^*} \text{Ker } ({}^tA)$$

è ancora esatta. In particolare il morfismo $\beta^* = \text{Hom}(\beta, P)$ è iniettivo e quindi risulta

$$\text{Hom}(M_n, P) \cong \text{Im } (\beta^*) = \text{Ker } (A)$$

□

Da questa proposizione si evince che possiamo ottenere informazioni sul nucleo dell'operatore A studiando l'operatore trasposto ${}^tA$. In particolare ora ci occuperemo del modulo M_n, che è un modulo omogeneo finitamente generato sull'anello dei polinomi P. Ad esso applicheremo un procedimeno standard che permette di costruire un risoluzione libera di M_n. Tale procedimento verrà discusso in generale nel prossimo capitolo, per ora ci limitiamo ad affermare che esite una sequenza esatta di P-moduli liberi di lunghezza finita m:

$$0 \longrightarrow P^{r_m} \xrightarrow{{}^tA_m} P^{r_{m-1}} \xrightarrow{{}^tA_{m-1}} \dots \xrightarrow{{}^tA_2} P^{r_1} \xrightarrow{{}^tA} P^{r_0} \longrightarrow M_n \longrightarrow 0 \qquad (1)$$

dove gli r_i sono interi positivi e le mappe A_i sono definite da opportune matrici ad elementi omogenei in P. La finitezza di una tale sequenza esatta è garantita dal Teorema delle Sizigie di Hilbert (si veda 3.1.30), che afferma inoltre che la lunghezza m della risoluzione non supera il numero delle indeterminate (nel nostro caso $4n$ se A è la matrice che rappresenta l'operatore di Cauchy-Fueter). Ora applichiamo alla sequenza (1) il funtore $\mathrm{Hom}(\cdot, P)$, dualizzando tutte le mappe, e otteniamo un complesso:

$$0 \longrightarrow P^{r_0} \xrightarrow{A} P^{r_1} \xrightarrow{A_2} \dots \xrightarrow{A_{m-1}} P^{r_{m-1}} \xrightarrow{A_m} P^{r_m} \longrightarrow 0 \qquad (2)$$

che in generale non è esatto. La non esattezza di tale complesso si misura con i moduli di coomologia. Prima di parlare di tali moduli e della loro importanza riassumiamo questo processo in una definizione.

Definizione 2.2.3. Siano $P = \mathbb{C}[x_0, \dots, x_{4n-1}]$ l'anello dei polinomi,
$A = (A_{ij}) \in \mathrm{Mat}_{r_1 \times r_0}(P)$ una matrice di polinomi lineari omogenei,
$D = (-i\frac{\partial}{\partial x_0}, \dots, -i\frac{\partial}{\partial x_{4n-1}})$ l'operatore di derivazione rispetto alle x_i,
M_n il conucleo del P-morfismo definito da ${}^tA$,
$A(D)$ l'operatore di derivazione associato alla matrice A,
$f = (f_0, \dots, f_3)$ una funzione $f : \mathbb{R}^{4n} \longrightarrow \mathbb{R}^4$ apparetenente alla classe $\mathcal{S}$.
Diciamo che

$$A(D) \cdot \underline{f} = \underline{0}$$

è il **sistema associato all'operatore A(D)**, e il complesso

$$0 \longrightarrow P^{r_0} \xrightarrow{A} P^{r_1} \xrightarrow{A_2} \dots \xrightarrow{A_{m-1}} P^{r_{m-1}} \xrightarrow{A_m} P^{r_m} \longrightarrow 0 \qquad (2)$$

ottenuto mediante il procedimento appena illustrato, si dice **complesso associato all'operatore A(D)**. Nel caso in cui A(D) è l'operatore di Cauchy-Fueter, cioè la matrice è del tipo della definizione 2.2.1, il complesso (2) si chiama **complesso di Cauchy-Fueter**.

Ogni volta che abbiamo a disposizione un complesso come il (2) possiamo considerare i moduli di coomologia

$$H^i := \frac{\text{Ker } (A_{i+1})}{\text{Im } (A_i)} \quad i = 1, \dots, m-1$$

che misurano la "non esattezza" del complesso. Nel nostro caso, dato che il complesso è stato ottenuto applicando il funtore $\text{Hom}(\cdot, P)$, le coomologie in questione sono i moduli

$$\text{Ext}^{\text{i}}(M_n, P)$$

Per la definizione precisa del funtore $\text{Ext}^{\text{i}}(\cdot, P)$ rimandiamo al capitolo successivo. Ricordiamo solo che si pone per definizione:

$$\text{Ext}^0(M_n, P) := \text{Hom}(M_n, P)$$

Siamo quindi partiti da un operatore differenziale $A(D)$, abbiamo costruito una risoluzione libera del conucleo di ${}^tA$, M_n, per ottenere dei moduli finitamente generati $\text{Ext}^{\text{i}}(M_n, P)$. La cosa interessante di questi moduli è che essi non dipendono affatto dalla risoluzione libera da cui si parte, ma solo dai moduli M_n e P, quindi costituiscono degli invarianti importanti associati ai due moduli in gioco. Nel prossimo paragrafo vedremo come si legano le proprietà dell'operatore $A(D)$ e del suo nucleo $\mathcal{S}^A$ con lo studio $\text{Ext}^{\text{i}}(M_n, P)$.

2.3 Importanza di Ext per la regolarità

Ritorniamo per un attimo alle funzioni olomorfe su $\mathbb{C}^n$, con $n > 1$. Per tale classe di funzioni vale una interessante proprietà nota come "Fenomeno di Hartog" che impedisce alle funzioni di avere singolarità compatte "troppo grosse". In effetti questo non dovrebbe stupirci, poiché sappiamo dalla teoria della variabile complessa che le condizioni di Cauchy-Riemann hanno come conseguenza una particolare rigidità delle funzioni che le soddisfano. Non ci stupirebbe quindi scoprire che anche le funzioni regolari su $\mathbb{H}^n$ hanno certe caratteristiche di "rigidità". Cominciamo ad enunciare il seguente teorema:

Teorema 2.3.1 (Fenomeno di Hartog). *Sia U un aperto connesso di $\mathbb{C}^n$, $n > 1$, K un compatto di U tale che $V := U\backslash K$ sia connesso, e sia $f : V \longrightarrow \mathbb{C}$ una funzione olomorfa. Allora f si estende in modo unico ad una funzione olomorfa su tutto U.*

Il nostro scopo è ora capire per quali classi di funzoni valga il fenomeno di Hartog, in particolare se questo sia legato alle proprietà algebriche del sistema di equazioni differenziali che individua la classe di funzioni in esame. Con le notazioni dei paragrafi precedenti possiamo enunciare il seguente importante teorema:

Teorema 2.3.2. *Sia $A(D)$ un operatore differenziale associato ad una matrice di polinomi $A \in \mathrm{Mat}_{4n\times 4}(P)$, f una funzione di variabili reali soluzione del sitema associato*

$$A(D) \cdot \underline{f} = \underline{0}$$

Supponiamo che f sia definita su un aperto di R^{4n} del tipo $U\backslash K$ dove U è un aperto connesso e K è un compatto. Allora vale il fenomeno di Hartog (cioè esiste una funzione $\widetilde{f}$ che estende f su tutto U e che soddisfa il sistema $A(D) \cdot \widetilde{f} = 0$) se e solo se vale la relazione

$$\mathrm{Ext}^1(M_n, P) = 0$$

dove al solito $M_n = \mathrm{Coker}\ ({}^tA)$.

Dimostrazione. Di una versione anche più generale di questo teorema, in cui si considerano operatori differenziali del tipo $A(D)$ lineari omogenei a coefficienti costanti su fasci di funzioni $C^\infty(\mathbb{R}^k)$, si è occupato ad esempio Ehrenpreis, rimandiamo quindi a [LE] per la dimostrazione. Per una versione essenzialmente algebrica del teorema e della sua prova si può anche vedere [ABL]. □

Dunque l'annullarsi del primo modulo Ext è una condizione necessaria e sufficiente affinché valga il fenomeno di Hartog. Questo fatto è di importanza

fondamentale e quindi costituisce uno dei motivi per cui nel nostro lavoro affronteremo lo studio e la possibilità di rappresentare un modulo del tipo $\mathrm{Ext}^{\mathrm{i}}(M, N)$ anche dal punto di vista computazionale. Si potrà infatti dedurre direttamente un analogo del teorema 2.3.1 anche per le funzioni regolari, o per una qualunque altra classe di funzioni che nasca da un sistema di equazioni differenziali del tipo di quelli esaminati finora.

Soffermiamoci ad analizzare il caso di $\mathbb{C}^n$ e di $\mathbb{H}^n$. Nel caso $n = 1$ le matrici associate agli operatori di Cauchy-Riemann (per funzioni su $\mathbb{C}$) o di Cauchy-Fueter (per funzioni su $\mathbb{H}$) sono quadrate di ordine rispettivamente 2 e 4. Rimandiamo agli esempi 5.3.1 e 5.3.2 riportati nell'ultimo capitolo in cui verifichiamo che Ext^1 effettivamente non si annulla in nessuno dei due casi, mostrando così che l'ipotesi $n > 1$ dei teoremi appena enunciati non può essere rimossa. In effetti, ad esempio, una funzione olomorfa di una sola variabile complessa può presentare singolarità compatte senza necessariamente poter essere estesa su tutto l'aperto, si pensi ad una qualunque funzione con un polo in un punto (ad esempio $f(z) = \frac{1}{z}$). Possiamo generalizzare questi due casi, studiando una condizione sufficiente sull'operatore affinché il primo Ext non sia nullo, e cioè non valga il fenomeno di Hartog. La proposizione che enuncia questo fatto (si veda 5.3.3) verrà presentata nell'ultimo capitolo insieme ai due esempi di calcolo appena citati. Nel caso invece di $n > 1$ abbiamo già visto che per le funzioni olomorfe vale il 2.3.1, e un suo esatto analogo possiamo enunciarlo anche su $\mathbb{H}^n$. Rimandiamo all'esempio 5.3.4 per una verifica diretta del caso $n = 2$.

Teorema 2.3.3. *Sia U un aperto connesso di $\mathbb{H}^n$, $n > 1$, K un compatto di U tale che $V := U \backslash K$ sia connesso, e sia $f : V \longrightarrow \mathbb{H}$ una funzione regolare. Allora f si estende in modo unico ad una funzione regolare su tutto U.*

Le informazioni che derivano dall'annullarsi degli Ext non si limitano a questa, anche l'annullarsi degli altri moduli $\mathrm{Ext}^{\mathrm{i}}(M_n, P)$ per $i \neq 1$ fornisce informazioni utili sulle funzioni della classe $\mathcal{S}^A$, alcune delle quali sono ancora oggi oggetto di studio. Vediamo alcune di quelle finora studiate.

Caso i=0. L'annullarsi di $\mathrm{Ext}^0(M_n, P)$ equivale semplicemente a dire che $\mathrm{Hom}(M_n, P) = 0$ da cui, sapendo che vale l'isomorfismo

$$\mathrm{Hom}(M_n, P) \cong \mathrm{Ker}\ (A)$$

ne segue che l'operatore $A(D)$ è iniettivo sulla classe di Schwarz $\mathcal{S}$.

Caso i=1. Abbiamo già discusso questo caso osservando che il primo modulo Ext si annulla se e solo se vale il fenomeno di Hartog.

Caso $i > 1$. Le proprietà che derivano dall'annulamento degli Ext successivi al primo sono ancora oggi oggetto di studio da parte di alcuni matematici quali Palamodov ([VPP]), Malgrange ed Ehrenpreis. Faremo qui un breve cenno ad alcuni risultati recenti legati a questo problema. Cominciamo con una definizione di tipo geometrico.

Definizione 2.3.4. Sia $A(D)$ l'operatore differenziale associato alla matrice $A = (A_{ij})$ come nelle notazioni precedenti. Si dice **varietà caratteristica** associata all'operatore $A(D)$ la varietà data dalle soluzioni delle equazioni

$$m_1 = \ldots = m_s = 0 \quad ,$$

dove gli m_i sono i minori di rango massimo della matrice A.

Nel caso ad esempio dell'operatore di Cauchy-Fueter la varietà caratteristica è un sottoinsieme di $\mathbb{C}^4$, in quanto luogo di zeri di polinomi in quattro variabili a coefficienti in $\mathbb{C}$. Le proprietà geometriche di tale varietà sono strettamente legate ai moduli Ext come enuncia la seguente proposizione.

Proposizione 2.3.5. *Sia V la varietà caratteristica associata all'operatore $A(D)$, $4n$ il numero delle variabili reali. Allora si ha che* $\dim(V) = 4n-p-1$ *se e solo se*

$$\mathrm{Ext}^{\mathrm{i}}(M_n, P) = 0 \quad \forall i \leq p \quad e \quad \mathrm{Ext}^{\mathrm{p+1}}(M_n, P) \neq 0$$

Questa prima proposizione illustra una proprietà puramente geometrica dell' operatore $A(D)$ e sottolinea ancora una volta l'utilità del calcolo dei moduli Ext, tuttavia non ci fornisce alcuna informazione sulle funzioni della classe $\mathcal{S}^A$. Vediamo ora invece due risultati interessanti dal punto di vista dell'analisi.

Teorema 2.3.6. *Sia U un aperto connesso e convesso di $\mathbb{H}^n$, e sia K un sottoinsieme compatto di U. Siano $\Sigma_1, \ldots, \Sigma_{2n-2}$ dei semispazi chiusi di $\mathbb{R}^{4n}$ e sia $\Sigma = \Sigma_1 \cup \ldots \cup \Sigma_{2n-2}$. Allora ogni funzione regolare su $U \backslash (K \cup \Sigma)$ si estende in modo unico ad una funzione regolare su $U \backslash \Sigma$*

Teorema 2.3.7. *Sia L un sottospazio di $\mathbb{H}^n = \mathbb{R}^{4n}$ di dimensione $2n+2$. Allora per ogni compatto $K \subseteq L$ e per ogni aperto connesso U relativamente compatto in K, ogni funzione regolare definita in un intorno di $K \backslash U$ si può estendere ad una funzione regolare in un intorno di K.*

Dimostrazione. I due teoremi sono immediati corollari del teorema 4, a pagina 105, di [VPP]. □

Osservazione 2.3.8. Per poter dimostrare tali teoremi Palamodov si avvale del fatto che, nel caso del complesso di Cauchy-Fueter, la dimensione della varietà caratteristica associata è esattamente $2n+1$, quindi in base alla proposizione 2.3.4 (applicata nel caso $p = 2n-2$) ne segue che

$$\mathrm{Ext}^{\mathrm{i}}(M_n, P) = 0 \quad \forall i \leq 2n-2, \quad e \quad \mathrm{Ext}^{2n-1}(M_n, P) \neq 0$$

In base dunque agli studi di Palamodov, l'annullarsi degli Ext successivi al primo si riflette nella possibilità, per le funzioni regolari, di eliminare singolarità "complicate" come quelle considerate nei teoremi 2.3.5 e 2.3.6.

A questo punto abbiamo concluso questa breve rassegna dei fatti che ci hanno portato ad intuire l'importanza del calcolo di Ext. Nel prossimo capitolo vedremo con più precisione la definizione e la costruzione esplicita del funtore con l'utilizzo della teoria delle Basi di Gröbner .

3. Definizione formale di Ext

In questo capitolo giungeremo alla definizione formale del funtore Ext attraverso l'introduzione di alcuni importanti concetti algebrici. Come abbiamo visto nel capitolo precedente, per poter studiare le proprietà legate al modulo M_n era necessario prima di tutto costruire una risoluzione libera di tale modulo. Questo è reso possibile dalle sizigie, infatti è proprio grazie ad esse che saremo in grado di definire, uno a uno, i moduli e le opportune mappe della risoluzione, calcolando di volta in volta le sizigie dei generatori del modulo precedente. Cominceremo quindi a vedere come si possono calcolare esplicitamente le sizigie di una s-upla di vettori $(g_1, \ldots, g_s)$, e vedremo un esempio. In seguito enunceremo la definizione precisa di risoluzione libera e vedremo anche in questo caso che questa può essere effettivamente calcolata. L'ultimo ingrediente necessario alla definizione è il funtore Hom, che permette di dualizzare una risoluzione. Grazie alle Basi di Gröbner anche in questo caso è possibile trovare una rappresentazione esplicita di un modulo del tipo $\mathrm{Hom}(M, N)$. Il primo paragrafo e le sue sottosezioni saranno quindi dedicati a questi preliminari di tipo puramente algebrico.

3.1 Preliminari Algebrici

Fissiamo in linea di massima la notazione che utilizzeremo nel seguito, richiamando alcuni concetti di base. Ci avvarremo comunque della stessa notazione usata in [KR] quindi per ogni riferimento mancante rimandiamo alla sua consultazione. Salvo eventuali casi in cui le dimostrazioni possono essere presentate in modo diverso, ci riferiremo sempre a quelle riportate da [KR]. Ricordiamo che K indicherà sempre un corpo, $P = K[x_1, \ldots, x_n]$ l'anello dei polinomi, $M \subseteq P^r$ un P-sottomodulo generato da una s-upla $\mathcal{G} = (g_1, \ldots, g_s)$. La base canonica di P^r sarà sempre indicata con $\{e_1, \ldots, e_r\}$ e i termini di modulo con $\mathbb{T}^n\langle e_1, \ldots, e_r\rangle$, sui quali risulta definito un term-ordering σ. Su P considereremo fissata una $\mathbb{Z}$-graduazione, non necessariamente quella standard, che induce in modo naturale una struttura di modulo graduato su P^r. A meno che non venga detto esplicitamente, però, i P-sottomoduli che considereremo non saranno necessariamente omogenei. Considereremo spesso le due mappe:

$$\lambda : P^r \longrightarrow P^s, \quad \lambda(e_i) = g_i \tag{I}$$

$$\Lambda : P^r \longrightarrow P^s, \quad \Lambda(e_i) = \mathrm{LT}_\sigma(g_i) \tag{II}$$

e con $\{\varepsilon_1, \ldots, \varepsilon_s\}$ indicheremo la base canonica di P^s. Per quanto riguarda la definizione di Base di Gröbner ricordiamo che si tratta di un sistema di generatori del modulo M che soddisfi una delle condizioni equivalenti del seguente teorema: (si veda 2.4.1 di [KR])

Teorema 3.1.1. *Per un insieme di elementi $G = \{g_1, \ldots, g_s\} \subseteq P^r \backslash \{0\}$ che genera un sottomodulo $M = \langle g_1, \ldots, g_s\rangle \subseteq P^r$ sia $\xrightarrow{G}$ la regola di riscrittura definita da G, sia $\mathcal{G}$ la s-upla $(g_1, \ldots, g_s)$, siano λ e Λ le mappe definite come sopra in (I) e (II). Allora le seguenti condizioni sono equivalenti:*

A_1) Per ogni elemento non nullo $m \in M$ esistono $f_1, \ldots, f_s \in P$ tali che $m = \sum_{i=1}^s f_i g_i$ e $\mathrm{LT}_\sigma(m) \geq_\sigma \mathrm{LT}_\sigma(f_i g_i)$ per ogni $i = 1, \ldots, s$ tali che $f_i g_i \neq 0$, cioè tali che $\mathrm{LT}_\sigma(m) \geq_\sigma \deg_{\sigma,\mathcal{G}}\left(\sum_{i=1}^s f_i \varepsilon_i\right)$.

A_2) Per ogni elemento non nullo $m \in M$ esistono $f_1, \ldots, f_s \in P$ tali che $m = \sum_{i=1}^s f_i g_i$ e $\mathrm{LT}_\sigma(m) = \max_\sigma\{\mathrm{LT}_\sigma(f_i g_i) | i \in \{1, \ldots, s\}, f_i g_i \neq 0\}$, cioè tali che $\mathrm{LT}_\sigma(m) = \deg_{\sigma,\mathcal{G}}\left(\sum_{i=1}^s f_i \varepsilon_i\right)$.

B_1) L'insieme $\{\mathrm{LT}_\sigma(g_1), \ldots, \mathrm{LT}_\sigma(g_s)\}$ genera il $\mathbb{T}^n$-monomodulo $\mathrm{LT}_\sigma\{M\}$.

B_2) L'insieme $\{\mathrm{LT}_\sigma(g_1), \ldots, \mathrm{LT}_\sigma(g_s)\}$ genera il P-sottomodulo $\mathrm{LT}_\sigma(M)$ di P^r.

C_1) Dato un elemento $m \in P^r$, si ha che $m \xrightarrow{G} 0$ se e solo se $m \in M$.

C_2) *Se* $m \in M$ *è irriducibile rispetto a* $\xrightarrow{G}$, *allora si ha* $m = 0$.

C_3) *Per ogni elemento* $m_1 \in P^r$ *esiste un unico elemento* $m_2 \in P^r$ *tale che* $m_1 \xrightarrow{G} m_2$ *e* m_2 *è irriducibile rispetto alla regola* $\xrightarrow{G}$.

C_4) *Se* $m_1, m_2, m_3 \in P^r$ *soddisfano* $m_1 \xrightarrow{G} m_2$ *e* $m_1 \xrightarrow{G} m_3$ *allora esiste un unico elemento* $m_4 \in P^r$ *tale che* $m_2 \xrightarrow{G} m_4$ *e* $m_3 \xrightarrow{G} m_4$.

D_1) *Ogni elemento omogeneo di* $\mathrm{Syz}(\mathrm{LM}_\sigma(\mathcal{G}))$ *ha un sollevamento in* $\mathrm{Syz}(\mathcal{G})$.

D_2) *Esiste un sistema di generatori omogenei per* $\mathrm{Syz}(\mathrm{LM}_\sigma(\mathcal{G}))$ *che consiste unicamente di elementi che hanno un sollevamento in* $\mathrm{Syz}(\mathcal{G})$.

D_3) *Esiste un sistema finito di generatori omogenei per* $\mathrm{Syz}(\mathrm{LM}_\sigma(\mathcal{G}))$ *che consiste unicamente di elementi che hanno un sollevamento in* $\mathrm{Syz}(\mathcal{G})$.

Un sistema di generatori G *che soddisfi una di queste condizioni equivalenti si dice* **Base di Gröbner** *per il modulo* M *rispetto all'ordinamento* σ.

Per la spiegazione dei concetti che intervengono nella definizione di Base di Gröbner che noi non abbiamo definito (quali ad esempio le **regole di riscrittura** $\xrightarrow{G}$, i **leading-term module** $\mathrm{LT}_\sigma(M)$ ecc.) rimandiamo alla consultazione dei primi capitoli di [KR], in quanto per i nostri scopi sarebbe inutile soffermarvisi ora. Citiamo soltanto l'algoritmo che permette, dato un sottomodulo di P^r, di trovare una Base di Gröbner rispetto ad un dato ordinamento σ:

Teorema 3.1.2 (Algoritmo di Buchberger).
Sia $(g_1, \ldots, g_s) \in (P^r)^s$ *una* s*-upla di vettori non nulli che genera un* P*-sottomodulo* $M = \langle g_1, \ldots, g_s \rangle \subseteq P^r$. *Per ogni* $i = 1, \ldots, s$ *sia* $\mathrm{LM}_\sigma(g_i) = c_i t_i e_{\gamma_i}$, *dove* $c_i \in K \backslash \{0\}$, $t_i \in \mathbb{T}^n$, *e* $\gamma_i \in \{1, \ldots, r\}$. *Si consideri la seguente lista di istruzioni:*
1) Sia $s' = s$ *e* $B = \mathbb{B} = \{(i,j) | 1 \leq i < j \leq s', \gamma_i = \gamma_j\}$.
2) Se $B = \emptyset$, *il risutato è* $\mathcal{G}$, *altrimenti si scelga una coppia* $(i,j) \in B$ *e la si cancelli da* B.
3) Si calcolino $S_{ij} = \frac{t_j g_i}{c_i \gcd(t_i, t_j)} - \frac{t_i g_j}{c_j \gcd(t_i, t_j)}$ *e* $\mathrm{NR}_{\sigma,\mathcal{G}}(S_{ij})$. *Se* $\mathrm{NR}_{\sigma,\mathcal{G}}(S_{ij}) = 0$ *si continui con il punto 2).*
4) Si aumenti s' *di uno, si aggiunga* $g_{s'} = \mathrm{NR}_{\sigma\mathcal{G}}(S_{ij})$ *alla upla* $\mathcal{G}$ *e si aggiunga l'insieme di coppie* $\{(i,s') | 1 \leq i < s', \gamma_i = \gamma_{s'}\}$ *a* B, *e si continui con il punto 2).*
Questo è un algoritmo, cioè termina in un numero finito di passi, e fornisce come risultato una upla $\mathcal{G}$ *di vettori che formano una* σ-Base di Gröbner *per* M.

3.1.1 Sizigie

In questa sezione vedremo come sia possibile calcolare esplicitamente le sizigie attraverso la teoria delle Basi di Gröbner . Ricordiamo per ora la definizione.

Definizione 3.1.3. Sia M un P-modulo, e sia $\mathcal{G} = (g_1, \ldots, g_s)$ una s-upla di elementi di M.
a) Una **sizigia** di $\mathcal{G}$ è una s-upla $(f_1, \ldots, f_s) \in P^s$ tale che

$$f_1 g_1 + \cdots + f_s g_s = 0$$

b) L'insieme di tutte le sizigie di $\mathcal{G}$ forma un P-modulo che chiameremo (primo) **modulo delle sizigie** di $\mathcal{G}$ e indicheremo con $\mathrm{Syz}(\mathcal{G})$ o con $\mathrm{Syz}(g_1, \ldots, g_s)$.

Ovviamente la definizione dipende dall'ordine degli elementi di $\mathcal{G}$, e non solo dal modulo da essi generato. Per questo si parlerà sempre di sizigie di un s-upla ordinata di elementi e non di sizigie di un modulo. I passi della nostra trattazione saranno questi:

1) Riuscire a calcolare un sistema esplicito di generatori di $\mathrm{Syz}(\mathcal{G})$ nel caso in cui i vettori g_i sono monomi.
2) Sfruttare la proprietà di sollevamento delle sizigie relativa alle Basi di Gröbner per poter calcolare $\mathrm{Syz}(\mathcal{G})$ nel caso in cui $(g_1, \ldots, g_s)$ è una Base di Gröbner per il modulo $\langle g_1, \ldots, g_s \rangle$.
3) Dedurre da questo caso il calcolo delle sizigie nel caso generale.

È necessario innanzitutto definire una graduazione fine su P^s utilizzando come "gradi" i termini di modulo $\mathbb{T}^n\langle e_1, \ldots, e_r \rangle$.

Definizione 3.1.4. Sia σ un term-ordering su P, sia $\mathcal{G} = (g_1, \ldots, g_s)$ una s-upla di elementi di P^r. Definiamo su P^s una $\mathbb{T}^n\langle e_1, \ldots, e_r \rangle$-graduazione in questo modo:

$$\deg_{\sigma,\mathcal{G}}(m) = \max_{\sigma}\{\mathrm{LT}_\sigma(f_i g_i), \quad i = 1..s\}$$

dove $m = \sum_{i=1}^{s} f_i \varepsilon_i$ è il generico elemento di P^r. Tale grado appena definito verrà chiamato anche **σ-grado**.

Con questa graduazione sul modulo P^s possiamo trovare un sistema di generatori omogenei per il modulo delle sizigie di $\mathrm{LM}_\sigma(\mathcal{G})$, risolvendo così il problema relativo al punto 1).

Teorema 3.1.5 (Sizigie di elementi di Moduli Monomiali).
Per ogni $j = 1..s$ scriviamo $\mathrm{LM}_\sigma(g_j)$ nella forma

$$\mathrm{LM}_\sigma(g_j) = c_j t_j e_{\gamma_j},$$

dove $c_j \in K$, $t_j \in \mathbb{T}^n$ *e* $\gamma_j \in \{1, \ldots, r\}$. *Poniamo poi,*

$$t_{ij} = \frac{\mathrm{lcm}\,(t_i, t_j)}{t_i} \quad \forall i, j \in \{1, \ldots, s\}$$

Allora

a) per ogni $i, j \in \{1, \ldots, s\}$ *tale che* $i < j$ *e* $\gamma_i = \gamma_j$, *l'elemento*

$$\sigma_{ij} = \frac{1}{c_i} t_{ij} \varepsilon_i - \frac{1}{c_j} t_{ji} \varepsilon_j$$

è una sizigia di $\mathrm{LM}_\sigma(\mathcal{G})$ *ed è omogeneo di* σ*-grado* $\deg_{\sigma,\mathcal{G}}(\sigma_{ij}) = \mathrm{lcm}\,(t_i, t_j) e_{\gamma_j}$

b) Si ha $\mathrm{Syz}\,(\mathrm{LM}_\sigma(\mathcal{G})) = \langle \sigma_{ij} \mid 1 \leq i < j \leq s, \gamma_i = \gamma_j \rangle$

Dimostrazione. Si veda [KR], teorema 2.3.7 □

Da qui in poi indicheremo con $\mathbb{B} = \{(i,j) | 1 \leq i < j \leq s, \gamma_i = \gamma_j\}$ l'insieme delle coppie di indici (i, j) che intervengono nei generatori σ_{ij}. Prima di andare avanti a calcolare le sizigie nel caso non monomiale introduciamo un particolare ordinamento τ su P^s che sia compatibile con la graduazione appena introdotta, nel senso che vedremo, e rispetto al quale gli elementi σ_{ij} formino una Base di Gröbner per il modulo $\mathrm{Syz}\,(\mathrm{LM}_\sigma(\mathcal{G})) = \mathrm{Ker}\,(\Lambda)$.

Definizione 3.1.6. Dati due termini di modulo $t\varepsilon_i$ e $t'\varepsilon_j \in \mathbb{T}^n \langle \varepsilon_1, \ldots, \varepsilon_s \rangle$ definiamo la seguente relazione:

$$t\varepsilon_i \geq_\tau t'\varepsilon_j$$

se $\mathrm{LT}_\sigma(tg_i) >_\sigma \mathrm{LT}_\sigma(t'g_j)$, oppure $\mathrm{LT}_\sigma(tg_i) = \mathrm{LT}_\sigma(t'g_j)$ e $i \leq j$.

In base alla definizione di σ-grado data in precedenza, questo ordinamento τ ordina i termini guardando prima il σ-grado e poi la posizione. Infatti vale che $\deg_{\sigma,\mathcal{G}}(t\varepsilon_i) = \mathrm{LT}_\sigma(tg_i)$, ed è in questo senso quindi che tale ordinamento è compatibile con il σ-grado.

Proposizione 3.1.7. *L'insieme* $\{\sigma_{ij} | (i,j) \in \mathbb{B}\}$ *è una* τ-Base di Gröbner *per* $\mathrm{Syz}\,(\mathrm{LM}_\sigma(\mathcal{G}))$.

Dimostrazione. Se $\mathrm{Syz}\,(\mathrm{LM}_\sigma(\mathcal{G})) = (0)$ non c'è nulla da dimostrare. Consideriamo quindi un elemento $z \in \mathrm{Syz}\,(\mathrm{LM}_\sigma(\mathcal{G}))$, $z \neq 0$ e sia $z' = \mathrm{LF}(z)$ la sua leading form rispetto a σ e $\mathcal{G}$. In base alle condizioni B) dobbiamo dimostrare che $\mathrm{LT}_\tau(z) \in \langle \mathrm{LT}_\tau(\sigma_{ij}) | (i,j) \in \mathbb{B} \rangle$. Ricordando la definizione di τ notiamo che $\mathrm{LT}_\tau(z) = \mathrm{LT}_\tau(z')$ quindi non è restrittivo supporre che l'elemento z sia omogeneo rispetto al σ-grado, cioè supponiamo $z = z'$. Scriviamo z nella

forma

$$z = \sum_{i=1}^{s} c'_i t'_i \varepsilon_i \in P^s$$

dove $c'_i \in K$ e $t'_i \in \mathbb{T}^n$. Sia $\mu := \min\{i \mid c'_i \neq 0\}$. Allora z è in realtà una sizigia relativa ai monimi di testa solo per i $g_\mu, \ldots, g_s$ cioè

$$z \in \mathrm{Syz}\,(\mathrm{LM}_\sigma(g_\mu), \ldots, \mathrm{LM}_\sigma(g_s)),$$

e il suo leading term rispetto a τ si trova semplicemente guardando la posizione, poiché il σ-grado è lo stesso per tutti i termini di z: $\mathrm{LT}_\tau(z) = t'_\mu \varepsilon_\mu$. Da ciò ne segue, in base alla proposizione 3.1.5 che z si scrive in questo modo:

$$z = \sum_{\mu \leq i < j \leq s} a_{ij}\sigma_{ij} = \sum_{\mu \leq i < j \leq s} a_{ij}\frac{1}{c_i}t_{ij}\varepsilon_i - a_{ij}\frac{1}{c_j}t_{ji}\varepsilon_j \qquad (1)$$

per opportuni $a_{ij} \in P$. I termini in posizione ε_μ compaiono nell scrittura (1) solo per $i = \mu$ dunque t'_μ è un multiplo di uno dei termini $\{t_{\mu j} | \mu < j \leq s\}$. Esiste allora un indice k tale che $t'_\mu = t'' t_{\mu k}$. Ma d'altra parte $\mathrm{LT}_\tau(\sigma_{\mu k}) = t_{\mu k}\varepsilon_\mu$ quindi abbiamo scoperto che $\mathrm{LT}_\tau(z) = t''\,\mathrm{LT}_\tau(\sigma_{\mu k})$ come volevamo. □

La prossima proposizione invece studierà il caso in cui $\mathcal{G} = (g_1, \ldots, g_s)$ è una Base di Gröbner , risolvendo così il problema del punto 2). Ricordiamo che data la s-upla $\mathcal{G}$ risulta definito un morfismo $\lambda : P^s \longrightarrow P^r$ così : $\lambda(\varepsilon_i) = g_i$. Dire che un certo elemento $\sigma \in P^s$ è una sizigia di $\mathcal{G}$ equivale a dire che $\lambda(\sigma) = 0$, cioè

$$\mathrm{Syz}\,(\mathcal{G}) = \mathrm{Ker}\,(\lambda)$$

Proposizione 3.1.8. *Sia $\mathcal{G} = (g_1, \ldots, g_s)$ una* σ-Base di Gröbner *per il modulo $M = \langle g_1, \ldots, g_s \rangle$, siano $\{\sigma_{ij} | (i,j) \in \mathbb{B}\}$ i generatori di* $\mathrm{Syz}\,(\mathrm{LM}_\sigma(\mathcal{G}))$ *come nel teorema 3.1.5. Scriviamo $\lambda(\sigma_{ij}) = \sum_{k=1}^{s} f_{ijk} g_k$ con opportuni $f_{ijk} \in P$. Chiamiamo*

$$s_{ij} := \begin{cases} \sigma_{ij} & se \lambda(\sigma_{ij}) = 0 \\ \sigma_{ij} - \sum_{k=1}^{s} f_{ijk}\varepsilon_k & se \lambda(\sigma_{ij}) \neq 0 \end{cases}$$

Allora l'insieme $\{s_{ij} | (i,j) \in \mathbb{B}\}$ è una τ-Base di Gröbner *per il modulo* $\mathrm{Syz}\,(\mathcal{G})$.

Dimostrazione. In base al teorema 3.1.5 l'elemento σ_{ij} è omogeneo di σ-grado $\deg_{\sigma,\mathcal{G}}(\sigma_{ij}) = \mathrm{lcm}\,(t_i, t_j) e_{\gamma_i}$. Dalla proposizione 2.3.6.b di [KR] segue che $\mathrm{LT}_\sigma(\lambda(\sigma_{ij})) <_\sigma \deg_{\sigma,\mathcal{G}}(\sigma_{ij})$. Prendiamo ora un elemento non nullo $z \in \mathrm{Syz}\,(\mathcal{G})$ (se infatti avessimo che $\mathrm{Syz}\,(\mathcal{G}) = 0$ non ci sarebbe nulla da dimostrare). In base alla definizione di τ deduciamo che $\mathrm{LT}_\tau(s_{ij}) = t_{ij}\varepsilon_i =$

$\mathrm{LT}_\tau(\sigma_{ij})$ e dobbiamo dimostrare che $\mathrm{LT}_\tau(z)$ è un multiplo degli elementi di $\{\mathrm{LT}_\tau(s_{ij})|(i,j) \in \mathbb{B}\}$. Grazie alle condizioni B) infatti questo significa che gli s_{ij} sono un τ-Base di Gröbner . Ma osservando che $\mathrm{LT}_\tau(z) = \mathrm{LT}_\tau(\mathrm{LF}(z))$ e sapendo che $\mathrm{LF}(z) \in (\mathrm{LM}_\sigma(\mathcal{G}))$ grazie alla proposizione 2.3.6.d di [KR], ne segue che $\mathrm{LT}_\tau(z)$ è un multiplo proprio dei $\mathrm{LT}_\tau(s_{ij}) = \mathrm{LT}_\tau(\sigma_{ij})$. □

La cosa interessante della precedente proposizione è che gli elementi $S_{ij} := \lambda(\sigma_{ij})$ possono essere rappresentati nella forma $S_{ij} = \sum_{k=1}^{s} f_{ijk} g_k$ proprio perché $\mathcal{G}$ è una Base di Gröbner e quindi gli f_{ijk} sono forniti dall'Algoritmo di Divisione (si veda ad esempio [KR] 1.6.4). Dunque il modulo $\mathrm{Syz}(\mathcal{G}) = \langle s_{ij}|(i,j) \in \mathbb{B}\rangle$ può essere calcolato esplicitamente. Ci apprestiamo ora quindi a calcolare le sizigie nel caso generale. Nel seguente teorema useremo una notazione matriciale in cui $\mathcal{A}$ e $\mathcal{B}$ sono due matrici, $\mathcal{G}$ e $\mathcal{H}$ sono due vettori riga e quindi ad esempio $\mathcal{GA}$ indica un prodotto righe per colonne.

Teorema 3.1.9. *Sia $\mathcal{H} = (h_1, \ldots, h_t)$ un sistema di generatori del P-sottomodulo M di P^r, sia $\mathcal{G} = (g_1, \ldots, g_s)$ una σ-Base di Gröbner per M. Supponiamo di avere due matrici $\mathcal{A} = (a_{ij}) \in \mathrm{Mat}_{t\times s}(P)$ e $\mathcal{B} = (b_{ij}) \in \mathrm{Mat}_{s\times t}(P)$ tali che $\mathcal{G} = \mathcal{HA}$ e $\mathcal{H} = \mathcal{GB}$. Sia poi $\mathcal{M}$ una matrice le cui colonne generino $\mathrm{Syz}(\mathcal{G})$, e indichiamo con $\mathcal{I}_t$ la matrice identica di ordine t. Allora le colonne della matrice*

$$\mathcal{N} = (\mathcal{AM}|\mathcal{I}_t - \mathcal{AB})$$

generano il modulo $\mathrm{Syz}(\mathcal{H})$.

Dimostrazione. Vale la seguente catena di uguaglianze:

$$\mathcal{HN} = (\mathcal{HAM}|\mathcal{H} - \mathcal{HAB}) = (\mathcal{GM}|\mathcal{H} - \mathcal{GB}) = (0|\mathcal{H} - \mathcal{H}) = 0$$

quindi le colonne di $\mathcal{N}$ sono sizigie per $\mathcal{H}$. Viceversa, se v è un vettore colonna che rappresenta una sizigia di $\mathcal{H}$, abbiamo $0 = \mathcal{H} \cdot v = \mathcal{GB} \cdot v$ da cui $\mathcal{B} \cdot v \in \mathrm{Syz}(\mathcal{G})$, e quindi il vettore $\beta \cdot v$ è nello spazio generato dalle colonne di $\mathcal{M}$. osservando l'identità

$$v = \mathcal{A}(\mathcal{B} \cdot v) + (\mathcal{I}_t - \mathcal{AB}) \cdot v$$

si conclude che v è nello spazio generato dalle colonne di $\mathcal{N}$. □

Osservando l'enunciato del teorema vediamo che nelle ipotesi è richiesta la conoscenza di due matrici $\mathcal{A}$ e $\mathcal{B}$ che leghino fra di loro le due uple di generatori $\mathcal{G}$ e $\mathcal{H}$. La matrice $\mathcal{A}$ può essere calcolata esplicitamente usando l'algoritmo di Buchberger 3.1.2 per ottenere una Base di Gröbner $\mathcal{G}$ a partire da $\mathcal{H}$, in modo che, tenendo conto dei passi fatti nell'algoritmo, si possa avere

una rappresentazione di $\mathcal{G}$ in termini di $\mathcal{H}$ del tipo:

$$g_j = a_{1j}h_1 + \cdots + a_{tj}h_t, \quad \forall j = 1..s$$

mentre una relazione inversa può essere calcolata con l'Algoritmo di Divisione già citato. Sfruttando il fatto che $\mathcal{G}$ è una Base di Gröbner per M troviamo infatti una rappresentazione esplicita

$$h_i = b_{1i}g_1 + \cdots + b_{si}g_s, \quad \forall i = 1..t$$

in tal modo basta usare gli a_{ij} e i b_{ij} per definire le matrici cercate. La matrice $\mathcal{M}$ nelle cui colonne mettiamo i generatori di $\mathrm{Syz}(\mathcal{G})$ può essere ovviamente costruita usando la proposizione 3.1.8. Facciamo fin da ora la seguente importante ossevazione che riguarda ancora il calcolo esplicito delle sizigie.

Osservazione 3.1.10. Supponiamo che gli elementi di $\mathcal{G}$ siano omogenei rispetto ad una certa graduazione su P^r. Grazie al Lemma di Nakayama e alle sue conseguenze (si veda ad es [KR] teorema 1.7.15 e corollario) possiamo sempre estrarre da $\mathcal{G}$ un sistema minimale di generatori. Ugualmente, una volta calcolate le sizigie del sistema ottenuto possiamo ancora ridurre il sistema di generatori delle sizigie stesse ad uno minimale. Questo è garantito dal fatto che le sizigie di elementi omogenei tra di loro sono ancora elementi omogenei, e quindi si può riapplicare Nakayama.

Dimostrazione. Dobbiamo solo dimostrare che le sizigie di elementi omogenei sono elementi omogenei. Basta ricordare gli elementi s_{ij} della proposizione 3.1.6:

$$s_{ij} := \begin{cases} \sigma_{ij} & se \lambda(\sigma_{ij}) = 0 \\ \sigma_{ij} - \sum_{k=1}^{s} f_{ijk}\varepsilon_k & se \lambda(\sigma_{ij}) \neq 0 \end{cases}$$

Essi hanno lo stesso σ-grado dei σ_{ij}, cioè $\mathrm{lcm}\,(t_i, t_j)e_{\gamma_i}$. □

L'osservazione si rivelerà particolarmente utile quando vorremo costruire una risoluzione libera "minimale" per un modulo finitamente generato. Di fatto, su CoCoA l'algoritmo sviluppato per il calcolo delle risoluzioni si applica solo al caso omogeneo, proprio perché questo genera una risoluzione che in qualche modo è "speciale" rispetto a tutte quelle possibili. Prima di passare alla prossima sezione però vediamo almeno un esempio di calcolo delle sizigie.

Esempio 3.1.11. consideriamo l'ideale $I := (h_1, h_2) = (x^2, xy + y^2)$. Vogliamo calcolare $\mathrm{Syz}(\mathcal{H})$ dove $\mathcal{H} = (h_1, h_2)$. In base al teorema 3.1.7 dobbiamo innanzitutto trovare una Base di Gröbner per l'ideale I, quindi fissiamo ad esempio l'ordinamento $\sigma = \mathrm{lex}$ e con l'aiuto di CoCoAtroviamo una Base di Gröbner . Si ottiene $\mathcal{G} = (g_1, g_2, g_3) = (x^2, xy + y^2, y^3)$. Come

si vede chiaramente, il fatto che CoCoA utilizzi l'Algoritmo di Buchberger (o meglio, una sua ottimizzazione) per calcolare $\mathcal{G}$, ha come effetto quello di aggiungere dei nuovi generatori all'ideale I, quindi il calcolo della matrice $\mathcal{B}$ è estremamente semplice:

$$\mathcal{B} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}$$

in tal modo infatti $\mathcal{H} = \mathcal{G}\mathcal{B}$. Per trovare la matrice $\mathcal{A}$ invece osserviamo che $g_3 = yh_1 + (y-x)h_2$ quindi mi basta definire

$$\mathcal{A} = \begin{pmatrix} 1 & 0 & y \\ 0 & 1 & y-x \end{pmatrix}$$

Ora dobbiamo trovare le sizigie di $\mathcal{G}$. Seguendo la proposizione 3.1.6 otteniamo gli elementi:

$$\begin{array}{rcl} s_{12} & = & y\varepsilon_1 + (y-x)\varepsilon_2 - \varepsilon_3 \\ s_{13} & = & y^3\varepsilon_1 - x^2\varepsilon_3 \\ s_{23} & = & y^2\varepsilon_2 - (x+y)\varepsilon_3 \end{array}$$

quindi la matrice $\mathcal{M}$ richiesta dal teorema 3.1.7 è la seguente

$$\mathcal{M} = \begin{pmatrix} y & y^3 & 0 \\ y-x & 0 & y^2 \\ -1 & -x^2 & -x-y \end{pmatrix}$$

da cui si ottiene la matrice cercata $\mathcal{N} = (\mathcal{A}\mathcal{M}|\mathcal{I}_t - \mathcal{A}\mathcal{B})$

$$\mathcal{N} = \begin{pmatrix} 0 & -y(x^2-y^20 & -y(x+y) & 0 & 0 \\ 0 & x^2(x-y) & x^2 & 0 & 0 \end{pmatrix}$$

Chiaramente le colonne della matrice $\mathcal{N}$ sono ridondanti, in realtà basta infatti la terza per generare il modulo:

$$\mathrm{Syz}\,(\mathcal{H}) = \langle -y(x+y)\varepsilon_1 + x^2\varepsilon_2 \rangle$$

Ora che sappiamo come calcolare le sizigie, vediamo come intervengono nelle risoluzioni libere.

3.1.2 Risoluzioni libere di un modulo finitamente generato

Ricordiamo per ora la definizione di complesso, concetto che era già intervenuto nel primo capitolo. Per le dimostrazioni dei questa sezione riguardanti l'algebra omologica ci si può riferire ad esempio a [T] e a [G].

Definizione 3.1.12. Un **complesso** $\mathbb{C}$ di P-moduli è una collezione, non necessariamente finita, di P-moduli e di morfismi $\{(C_i, d_i),\, d_i : C_i \longrightarrow C_{i-1}\}_{i\in\mathbb{N}}$, tale che:

$$d_i \circ d_{i+1} = 0$$

cioè vale la relazione $\mathrm{Im}\,(d_{i+1}) \subseteq \mathrm{Ker}\,(d_i)\ \forall i \in \mathbb{N}$. $\mathbb{C}$ si indica spesso con la sequenza:

$$\mathbb{C}: \quad \ldots \longrightarrow C_i \xrightarrow{d_i} C_{i-1} \xrightarrow{d_{i-1}} \ldots \xrightarrow{d_2} C_1 \xrightarrow{d_1} C_0 \xrightarrow{d_0=0} 0$$

Di particolare importanza sono i moduli di omologia associati al complesso, che come già osservato nel primo capitolo, ne misurano in un certo senso la non esattezza.

Definizione 3.1.13. Sia $\mathbb{C}$ un complesso, n un numero naturale. Sono definiti i seguenti P-sottomoduli di C_n:

$\mathbf{Z}_n(\mathbb{C}) = \mathrm{Ker}\,(d_n)$, detto modulo degli n-**cicli**,

$\mathbf{B}_n(\mathbb{C}) = \mathrm{Im}\,(d_{n+1})$, detto modulo degli n-**bordi**,

$\mathbf{H}_n(\mathbb{C}) = \frac{Z_n(\mathbb{C})}{B_n(\mathbb{C})}$ detto n-esimo **modulo di omologia** del complesso.

Siamo ora in grado di spiegare cosa intendiamo per risoluzione libera di un modulo. Le definizioni che stiamo dando sono ovviamente del tutto generali, e si applicano al caso di R-moduli qualunque, se R è un anello commutativo con identità. Dati i nostri scopi, comunque, avremo sempre a che fare con moduli finitamente generati sull'anello dei polinomi.

Definizione 3.1.14. Sia M un P-modulo. Una **risoluzione libera** di M è un complesso di P-moduli $\mathbb{L}$:

$$\mathbb{L}: \quad \ldots \longrightarrow L_i \xrightarrow{d_i} L_{i-1} \xrightarrow{d_{i-1}} \ldots \longrightarrow L_1 \xrightarrow{d_1} L_0 \xrightarrow{d_0=0} 0$$

con le seguenti proprietà:

1) L_i è libero per ogni $i \in \mathbb{N}$

2) $H_0(\mathbb{L}) = \mathrm{Coker}\,(d_1) = \frac{L_0}{\mathrm{Im}\,(d_1)} \cong M$

3) $H_i(\mathbb{L}) = 0 \quad \forall i \geq 1$ (esattezza)

Le proprietà 2) e 3) dicono che con il complesso $\mathbb{L}$ possiamo in realtà costruire una sequenza esatta:

$$\mathbb{L} \xrightarrow{p} M \longrightarrow 0 : \ldots \longrightarrow L_i \xrightarrow{d_i} L_{i-1} \xrightarrow{d_{i-1}} \ldots \xrightarrow{d_2} L_1 \xrightarrow{d_1} L_0 \xrightarrow{p} M \longrightarrow 0 \quad (1)$$

dove la mappa p indica la proiezione canonica sul quoziente che realizza l'isomorfismo della proprietà 2).
Il nostro problema ora è cercare di capire se dato un modulo esiste sempre una sequenza esatta come la (1), e soprattutto se è possibile costruirla esplicitamente, magari con un algoritmo finito. Sfrutteremo il fatto che stiamo lavorando in un anello graduato, e in particolare utilizzeremo l'osservazione 3.1.10 per concludere alcuni fatti importanti sull'esistenza di una risoluzione "minimale".

Definizione 3.1.15. Sia M un modulo graduato finitamente generato sull'anello dei polinomi P munito della graduzione standard. Diciamo che una risoluzione libera $\mathbb{L}$ di M è **graduata** se i morfismi d_i sono tutti omogenei di grado zero, cioè se portano elementi omogenei di un dato grado in elementi omogenei dello stesso grado.

Definizione 3.1.16. Indichiamo con $I_{max} = (x_1, \ldots, x_n)$ l'ideale massimale irrilevante di P. Diciamo allora che una risoluzione libera graduata $\mathbb{L}$ di M è **minimale** se vale

$$\mathrm{Im}\,(d_i) \subseteq I_{max} L_{i-1}, \quad \forall i \geq 1$$

Nel nostro caso, in cui consideriamo P-moduli finitamente generati, i morfismi d_i possono essere pensati come matrici ad elementi in P. La definizione appena data, allora, si traduce nel fatto che le matrici che rappresentano i morfismi della risoluzione hanno come elementi polinomi omogenei di grado maggiore o uguale a 1 (oppure zeri), cioè non compaiono costanti non nulle.

Possiamo allora enunciare e dimostrare un'importante proposizione, nella cui dimostrazione è racchiuso un algoritmo che permette di calcolare una risoluzione libera minimale per un modulo finitamente generato.

Proposizione 3.1.17. *Ogni P-modulo graduato M finitamente generato ammette una risoluzione libera graduata e minimale.*

Dimostrazione. Sia $M = \langle m_1, \ldots, m_r \rangle$ dove $\deg(m_i) = p_i \quad \forall i = 1..r$. Cominciamo la risoluzione costruendo il morfismo d_0 in questo modo:

$$\begin{array}{ccccc} \bigoplus_{i=1}^{r} P[-p_i] & \xrightarrow{d_0} & M & \longrightarrow & 0 \\ e_i & \longmapsto & m_i & & \end{array}$$

dove la scrittura $P[-p_i]$ indica l'usuale shift della graduazione (alza di p_i il grado di ciascun elemento omogeneo). Chiaramente d_0 è surgettiva. Per andare avanti nella costruzione dobbiamo poter trovare $\mathrm{Ker}\,(d_0)$ in modo da definire poi d_1 tale da soddisfare la relazione $\mathrm{Ker}\,(d_0) = \mathrm{Im}\,(d_1)$. Ma per come è definita d_0 il suo nucleo altro non è che il modulo delle sizigie $\mathrm{Syz}\,(m_1, \ldots, m_r)$ quindi possiamo ricorrere al teorema 3.1.9 per calcolcare esplicitamente un sistema di generatori $m'_1, \ldots, m'_s$ per $\mathrm{Syz}\,(m_1, \ldots, m_r)$, omogenei di gradi $p'_1, \ldots, p'_s$. Inoltre in base all'osservazione 3.1.10 tale sistema di generatori lo possiamo supporre minimale. Basta quindi ora definire:

$$\begin{array}{ccc} \bigoplus_{i=1}^{s} P[-p'_i] & \xrightarrow{d_1} & \bigoplus_{i=1}^{r} P[-p_i] \\ e_i & \longmapsto & m'_i \end{array}$$

Iterando questo procedimento otteniamo una sequenza esatta costituita da moduli liberi graduati e da morfismi omogenei (grazie agli shift). Dobbiamo solo verificare che si tratta di una risoluzione minimale. Controlliamo ad esempio il primo passo della costruzione. Se per assurdo risultasse $\mathrm{Ker}\,(d_0) \nsubseteq I_{max}L_0$ ci sarebbe un elemento $v \in \mathrm{Ker}\,(d_0) \backslash I_{max}L_0$. Non è restrittivo supporre che $v = \sum a_i e_i$ con $a_1 \in K\backslash\{0\}$. Ma $d_0(v) = 0$ quindi $\sum a_i m_i = 0$. Questo va contro il fatto che abbiamo scelto gli m_i minimali. Un'argomentazione analoga vale nei passi successivi. □

La proposizione appena enunciata non ci dice però se l'algoritmo per la costruzione della risoluzione termini o meno, nè ci da informazioni sui ranghi dei moduli liberi definiti. A priori, tali ranghi potrebbero dipendere dalla scelta dei generatori di M e delle sizigie calcolate di volta in volta. Per poter risolvere questi due problemi, che dal punto di vista computazionale hanno un'importanza cruciale come si può facilmente capire, dobbiamo fare un passo indietro e definire in modo preciso cosa si intende per funtore, concetto che ci sarà utile anche nei paragrafi successivi.

Definizione 3.1.18. Dati due complessi

$$\mathbb{C} = \{(C_i, d_i)\}_{i\in\mathbb{N}} \quad e \quad \mathbb{C}' = \{(C'_i, d'_i)\}_{i\in\mathbb{N}}$$

si dice **morfismo di complessi** una collezione $\{f_i\}_{i\in\mathbb{N}}$ di mappe P-lineari $f_i : C_i \longrightarrow C'_i$ tali che

$$d'_{i+1} \circ f_{i+1} = f_i \circ d_i, \quad \forall i \in \mathbb{N}$$

Scriveremo brevemente $f : \mathbb{C} \longrightarrow \mathbb{C}'$ per indicare il morfismo.

Si nota subito che un morfismo di complessi f manda cicli in cicli e bordi in bordi, quindi induce in modo naturale una collezione di morfismi:

$$H_i(f) : H_i(\mathbb{C}) \longrightarrow H_i(\mathbb{C}'), \quad i \in \mathbb{N}$$

tra i moduli di omologia.

Definizione 3.1.19. Due morfismi di complessi $f, g : \mathbb{C} \longrightarrow \mathbb{C}'$ si dicono **omotopicamente equivalenti** ($f \simeq g$) se esiste una collezione di mappe P-lineari $\{s_i : C_i \longrightarrow C'_{i+1}\}_{i\in\mathbb{N}}$ tali che

$$d'_{i+1}s_i + s_{i-1}d_i = f_i - g_i, \quad \forall i \in \mathbb{N}$$

I due complessi $\mathbb{C}$ e $\mathbb{C}'$ si dicono poi omotopicamente equivalenti se esistono due morfismi $f : \mathbb{C} \longrightarrow \mathbb{C}'$ e $g : \mathbb{C}' \longrightarrow \mathbb{C}$ tali che $gf \simeq Id_{\mathbb{C}}$ e $fg \simeq Id_{\mathbb{C}'}$, dove $Id_{\mathbb{C}} := \{Id_{C_i}\}_{i\in\mathbb{N}}$.

Proposizione 3.1.20. *Siano $f, g : \mathbb{C} \longrightarrow \mathbb{C}'$ due morfismi omotopicamente equivalenti. Allora $H_i(f)$ e $H_i(g)$ coincidono per ogni $i \in \mathbb{N}$. Se invece $\mathbb{C}$ e $\mathbb{C}'$ sono complessi omotopicamente equivalenti allora $H_i(\mathbb{C})$ e $H_i(\mathbb{C}')$ sono isomorfi per ogni $i \in \mathbb{N}$.*

Proposizione 3.1.21. *Sia $f : M \longrightarrow M'$ un morfismo di P-moduli, e siano $\mathbb{L} \xrightarrow{\varepsilon} M \longrightarrow 0$ ed $\mathbb{L}' \xrightarrow{\varepsilon'} M' \longrightarrow 0$ due risoluzioni libere di M e M' rispettivamente. Allora esiste un morfismo di complessi, unico a meno di omotopia, $F : \mathbb{L} \longrightarrow \mathbb{L}'$ tale che $f \circ \varepsilon = \varepsilon' \circ F_0$.*

Corollario 3.1.22. *Due risoluzioni libere sono sempre omotopicamente equivalenti e le omologie non dipendono dalla risoluzione scelta.*

Dimostrazione. Basta applicare la 3.1.21 due volte con $f = Id_M$. Il morfismo F che viene indotto tra le risoluzioni libere costituisce una equivalenza di omotopia. Quindi per la 3.1.20 le omologie associate sono isomorfe. □

Definizione 3.1.23. Un **funtore additivo covariante** (rispettivamente **controvariante**) è una corrispondenza che associa ad ogni P-modulo M un P-modulo $T(M)$ e ad ogni morfismo $f : M \longrightarrow M'$ un morfismo

$$T(f) : T(M) \longrightarrow T(M') \quad (\textit{rispettivamente} \quad T(f) : T(M') \longrightarrow T(M))$$

tale che

1) $T(Id_M) = Id_{T(M)}$

2) $T(g \circ f) = T(g) \circ T(f)$ $\quad (\textit{rispettivamente} \quad T(g \circ f) = T(f) \circ T(g))$

3) $T(f + g) = T(f) + T(g)$

per ogni $f : M \longrightarrow M'$ e $g : M' \longrightarrow M''$

Applicando un funtore covariante additivo ad un complesso $\mathbb{C}$ si ottiene ancora un complesso, precisamente:

$$T(\mathbb{C}) : \ldots \longrightarrow T(C_i) \overset{T(d_i)}{\longrightarrow} T(C_{i-1}) \longrightarrow \ldots \longrightarrow T(C_1) \overset{T(d_1)}{\longrightarrow} T(C_0) \longrightarrow 0$$

Se applichiamo invece un funtore controvariante T' otteniamo ancora un complesso, ma data la proprietà 2) tutte le mappe saranno dirette nel verso opposto:

$$T'(\mathbb{C}) : \quad 0 \longrightarrow T'(C_0) \overset{T'(d_i)}{\longrightarrow} T'(C_1) \longrightarrow \ldots \longrightarrow T'(C_{i-1}) \overset{T'(d_i)}{\longrightarrow} T'(C_i) \longrightarrow \ldots$$

Tale complesso viene anche detto **complesso di coomologia**.

Esempio 3.1.24. Tra i funtori più importanti che useremo ricordiamo ad esempio il funtore covariante dato dal prodotto tensore $\otimes_P N$:

$$T(M) = M \otimes_P N, \quad T(f) = f \otimes_P Id_N, \quad \textit{dove } f : M \longrightarrow M'$$

e il funtore controvariante $\mathrm{Hom}(\cdot, N)$

$$T'(M) = \mathrm{Hom}(M, N), \quad T'(f) = \mathrm{Hom}(f, Id_N)$$

che avremo modo di illustrare meglio nel prossimo paragrafo.

Diamo ancora la seguente definizione che introduce il concetto di funtore derivato sinistro (rispettivamente destro):

Definizione 3.1.25. Dato un funtore additivo covariante T definiamo il **funtore derivato sinistro** LT in questo modo: sia M un P-modulo e sia $\mathbb{L}$ una risoluzione libera di M. Applichiamo il funtore T alla risoluzione $\mathbb{L}$ e consideriamo le omologie $H_i(T(\mathbb{L}))$ del complesso ottenuto. Poniamo allora

$$L_iT(M) := H_i(T(\mathbb{L})) = \frac{\mathrm{Ker}\,(T(d_i))}{\mathrm{Im}\,(T(d_{i+1}))}$$

Sia invece $f : M \longrightarrow M'$ un morfismo e $\mathbb{L}$ ed $\mathbb{L}'$ due risoluzioni libere di M ed M' rispettivamente. Grazie alla proposizione 3.1.21, f si può sollevare ad un morfismo $F : \mathbb{L} \longrightarrow \mathbb{L}'$ che induce un morfismo di complessi $T(F) : T(\mathbb{L}) \longrightarrow T(\mathbb{L}')$ e di conseguenza abbiamo le seguenti mappe tra i moduli di omologia:

$$H_i(T(F)) : H_i(T(\mathbb{L})) \longrightarrow H_i(T(\mathbb{L}'))$$

basta quindi definire $L_iT(f) := H_i(T(F))$ per avere a disposizione un funtore.

Dato invece un funtore additivo controvariante T', definiamo il **funtore derivato destro** RT' in modo simmetrico. Applichiamo alla risoluzione $\mathbb{L}$ di M il funtore T', ottenendo un complesso:

$$T'(\mathbb{L}) : 0 \longrightarrow T'(L_0) \overset{T'(d_1)}{\longrightarrow} T'(L_1) \longrightarrow \ldots \longrightarrow T'(L_{i-1}) \overset{T'(d_i)}{\longrightarrow} T'(C_i) \to \ldots$$

Poniamo allora, simmetricamente al caso precedente,

$$R^iT'(M) := H_i(T'(\mathbb{L})) = \frac{\operatorname{Ker}(T'(d_{i+1}))}{\operatorname{Im}(T'(d_i))}$$

I moduli appena definiti si dicono anche **moduli di coomologia** di $\mathbb{L}$ rispetto al funtore controvariante T'. Si usa generalmente una notazione analoga a quella dei moduli di omologia, ma con gli indici scritti in alto:

$\mathbf{Z}^i(\mathbb{L}) = \operatorname{Ker}(T'(d_{i+1}))$, detto modulo degli i-**cocicli**

$\mathbf{B}^i(\mathbb{L}) = \operatorname{Im}(T'(d_i))$, detto modulo degli i-**cobordi**

$\mathbf{H}^i(\mathbb{L}) = \frac{Z^i(\mathbb{L})}{B^i(\mathbb{L})}$, detto i-**esimo modulo di coomologia**.

L'azione di R^iT' sulle mappe è del tutto simile al caso covariante. Con le stesse notazione poniamo infatti $R^iT'(f) := H_i(T'(F))$. La corrispondenza che ne otteniamo è dunque un funtore.

Vale la seguente proprietà per i funtori derivati sinistri e destri:

Proposizione 3.1.26. *I moduli $L_iT(M)$ (rispettivamente $R^iT'(M)$) e i morfismi $L_i(T(f))$ (rispettivamente $R^iT'(f)$) sono ben definiti e non dipendono dalla risoluzione scelta. Inoltre L_iT (rispett. R^iT') è un funtore additivo covariante (rispett. controvariante) per ogni $i \in \mathbb{N}$.*

Applicando la definizione di funtore derivato sinistro nel caso $T = \otimes_P N$ si ottiene un importante funtore detto **Tor**.

Definizione 3.1.27. Indichiamo con $\operatorname{Tor}_i(\cdot, N)$ il funtore L_iT dove $T = \otimes_P N$. In particolare se M è un P modulo e $\mathbb{L}$ una sua risoluzione libera abbiamo

$$\operatorname{Tor}_i(M, N) = H_i(\mathbb{L} \otimes_P N)$$

dove il complesso $\mathbb{L} \otimes_P N$ è il seguente

$$\mathbb{L} \otimes_P N : \ldots L_i \otimes_P N \longrightarrow L_{i-1} \otimes_P N \longrightarrow \ldots \longrightarrow L_0 \otimes_P N \longrightarrow 0$$

ottenuto tensorizzando con N, a destra, la risoluzione libera di M.

Il funtore Tor riveste una particolare importanza per rispondere alle domande che ci eravamo posti riguardo alla finitezza della risoluzione minimale di un modulo finitamente generato e alla buona definizione dei ranghi dei moduli liberi che intervengono nella risoluzione. Vediamo in che senso:

Proposizione 3.1.28. *Sia* $\mathbb{L}$ *una risoluzione libera, minimale e graduata di un* P*-modulo* M

$$\mathbb{L} \longrightarrow M \longrightarrow 0$$

dove $L_i = \bigoplus_{j\in\mathbb{N}} P^{\beta_{ij}}[-j]$*. Allora per ogni* $i, j \in \mathbb{N}$ *si ha che*

$$\beta_{ij} = \dim_K[\mathrm{Tor}_i(K, M)]_j$$

Dal momento che i moduli $\mathrm{Tor}_i(K, M)$ sono omologie e sono quindi indipendenti dalla risoluzione scelta (grazie al corollario 3.1.22) i numeri β_{ij} sono invarianti associati al modulo M.

Definizione 3.1.29. Con le notazioni precedenti β_{ij} si dice i-esimo **numero di Betti graduato** (in grado j) e il numero

$$\beta_i := \sum_{j\geq 0} \beta_{ij} = \dim_K(\mathrm{Tor}_i(K, M))$$

si dice i-esimo **numeri di Betti**.

Il problema invece della finitezza della risoluzione libera di M è risolto dal seguente

Teorema 3.1.30 (Teorema delle Sizigie di Hilbert).
Sia $P = K[x_1, \ldots, x_n]$ *e sia* M *un* P*-modulo graduato finitamente generato. Allora ogni risoluzione libera graduata di* M *ha lunghezza al più* n*.*

Il problema del calcolo di una risoluzione, almeno nel caso omogeneo, è stato quindi risolto. Vediamo un esempio facile in cui si vede come funzionano le cose.

Esempio 3.1.31. Consideriamo il P-modulo $M := \frac{K[x,y]}{(x^2,xy^2)}$. Ovviamente risulta $M = \langle \ \overline{1} \ \rangle$ quindi cominciamo a definire la mappa d_0 in questo modo:

$$\begin{array}{ccccc} P & \xrightarrow{d_o} & M & \longrightarrow & 0 \\ 1 & \longmapsto & \overline{1} & & \end{array}$$

Chiaramente vale che $\mathrm{Ker}\,(d_0) = (x^2, xy^2)$ e allora definiamo d_1 nel modo

seguente:

$$\begin{array}{ccc} P[-2]\bigoplus P[-3] & \xrightarrow{d_1} & P \\ \\ e_1 & \longmapsto & x^2 \\ e_2 & \longmapsto & xy^2 \end{array}$$

e infine calcolando con il teorema 3.1.5 le sizigie $Syz(x^2, xy^2) = \langle y^2e_1 - xe_2 \rangle$ definiamo

$$d_2 : P[-4] \longrightarrow P[-2]\bigoplus P[-3], \quad d_2(1) = y^2e_1 - xe_2$$

La mappa d_2 che abbiamo ottenuto è chiaramente iniettiva, vale infatti che $\mathrm{Ker}\,(d_2) = \mathrm{Syz}\,(y^2e_1 - xe_2) = 0$, dunque abbiamo terminato la risoluzione. In effetti il Teorema delle Sizigie di Hilbert già ci assicurava che tale risoluzione libera sarebbe stata lunga al più 2:

$$0 \longrightarrow P[-4] \xrightarrow{d_2} P[-2]\bigoplus P[-3] \xrightarrow{d_1} P \xrightarrow{d_0} M \longrightarrow 0$$

3.2 Presentazione del modulo Hom

Ora che abbiamo capito come si calcolano le risoluzioni, dobbiamo approfondire lo studio del funtore controvariante Hom. Applicando infatti tale funtore ad una risoluzione libera si ottiene il complesso necessario alla definizione di Ext, quindi il nostro scopo è quello di riuscire a rappresentare un modulo del tipo $\mathrm{Hom}(U, V)$ dove U e V sono due P-moduli finitamente generati. Coerentemente con le scelte di [KR] e con le esigenze computazionali di CoCoA, ogni volta che avremo a che fare con un modulo finitamente generato lavoreremo in realtà con una sua presentazione, cioè lo penseremo della forma P^r/M. Dunque in realtà questo paragrafo è dedicato a fornire una presentazione di un modulo $\mathrm{Hom}(P^r/M, P^s/N)$ come quoziente di un modulo libero di rango finito. Ricordiamo intanto la definizione.

Definizione 3.2.1. Siano $M \subseteq P^r$ e $N \subseteq P^s$ due sottomoduli. L'insieme

$$\mathrm{Hom}(P^r/M, P^s/N) = \{\text{morfismi } \varphi : P^r/M \longrightarrow P^s/N\}$$

è dotato di una struttura di P-modulo nel modo seguente:

$$(\varphi + \psi)(v + M) = \varphi(v + M) + \psi(v + M) \quad e \quad (f \cdot \varphi)(v + M) = f \cdot \varphi(v + M)$$

dove $v + M$ rappresenta la classe di $v \in P^r$ nel quoziente P^r/M.

Come primo passo studieremo il caso semplice in cui $M = 0$ e $N = 0$. Di fatto il modulo $\mathrm{Hom}(P^r, P^s)$ è isomorfo ad un modulo di matrici a elementi in P come descritto dalla seguente definizione e successiva proposizione.

Definizione 3.2.2. Dato un P-morfismo $\varphi \in \mathrm{Hom}(P^r, P^s)$ sia

$$\varphi(e_j) = (a_{1j}, \ldots, a_{sj}), \quad \forall j = 1..r\,,$$

e sia $\mathcal{A}_\varphi = (a_{ij}) \in \mathrm{Mat}_{sr}(P)$. Tale matrice si dice **matrice associata** a φ. La costruzione di A_φ induce una mappa

$$\begin{array}{rccc} \Lambda_{rs}: & \mathrm{Hom}(P^r, P^s) & \longrightarrow & \mathrm{Mat}_{sr}(P) \\ & \varphi & \longmapsto & A_\varphi \end{array}$$

Tale mappa è chiaramente un morfismo di P-moduli. Definiamo anche il morfismo

$$\mathrm{Fl}_{sr} : \mathrm{Mat}_{sr}(P) \longrightarrow P^{rs}$$

che manda una matrice $\mathcal{A} = (a_{ij})$ in un vettore $(a_{11}, \ldots, a_{s1}, \ldots, a_{1r}, \ldots, a_{sr})$. Esso è banalmente un isomorfismo di P-moduli detto **isomorfismo di flattering**.

Proposizione 3.2.3. *La mappa* $\Psi_{rs} = \mathrm{Fl}_{sr} \circ \Lambda_{rs} : \mathrm{Hom}(P^r, P^s) \longrightarrow P^{rs}$ *è un isomorfismo.*

Dimostrazione. Il flattering è ovviamente un isomorfismo, dobbiamo quindi solo verificare che lo sia anche Λ. Tale morfismo è iniettivo perché se $\Lambda(\varphi) = 0$ allora $A_\varphi = 0$ e quindi $\varphi(e_j) = 0 \, \forall j = 1..r$. Questo implica che φ è il morfismo nullo. Per la surgettività basta osservare che data una matrice $A = (a_{ij}) \in \mathrm{Mat}_{sr}(P)$ possiamo sempre definire una φ tale che $A_\varphi = A$ in questo modo: $\varphi(e_j) = (a_{1j}, \ldots, a_{sj})$, $\forall j = 1..r$. □

Dunque il modulo $\mathrm{Hom}(U, V)$ nel caso di moduli liberi è anch'esso un modulo libero, di rango il prodotto dei due ranghi dei moduli in gioco. Ora vogliamo studiare il caso generale. Per fare questo dobbiamo prima capire quali sono le proprietà funtoriali di Hom, cioè come questo agisca sui morfismi.

Definizione 3.2.4. Siano U, V e W tre P-moduli. Dato un P-morfismo $\varphi : U \longrightarrow V$, risulta definito un altro P-morfismo

$$\varphi^* : \mathrm{Hom}(V, W) \longrightarrow \mathrm{Hom}(U, W), \quad \varphi^*(\lambda) = \lambda \circ \varphi, \quad \forall \lambda \in \mathrm{Hom}(V, W)$$

Scriveremo $\varphi^* = \mathrm{Hom}(\varphi, W)$.

In tal modo $\mathrm{Hom}(\cdot, W)$ ha una struttura di funtore controvariante. Ricordando infatti la definizione 3.1.23 vediamo che vale la proprietà 2), in questa forma:

$$\mathrm{Hom}(\varphi \circ \psi, W) = \mathrm{Hom}(\psi, W) \circ \mathrm{Hom}(\varphi, W)$$

Dato che le mappe come φ^* sono P-lineari, vogliamo cercare di capire se anche esse posso essere rappresentate con una matrice associata. Questo ci verrà molto utile non tanto per il calcolo esplicito del modulo $\mathrm{Hom}(U, V)$ ma soprattutto nel calcolo di Ext in cui intervengono le mappe del complesso ottenuto dalla risoluzione libera. Sfruttiamo gli isomorfismi Ψ_{rs}, definiti in 3.2.3, per costruire il diagramma commutativo:

$$\begin{array}{ccc} \mathrm{Hom}(P^{r'}, P^s) & \xrightarrow{\varphi^*} & \mathrm{Hom}(P^r, P^s) \\ \downarrow{\scriptstyle \Psi_{r's}} & & \downarrow{\scriptstyle \Psi_{rs}} \\ P^{r's} & \xrightarrow{\widetilde{\varphi}} & P^{rs} \end{array}$$

dove $\widetilde{\varphi} = \Psi_{rs} \circ \varphi^* (\Psi_{r's})^{-1}$. Il nostro scopo è quindi trovare la matrice associata a $\widetilde{\varphi}$. Intuitivamente, se applicare φ^* significa comporre a destra con φ, applicare $\widetilde{\varphi}$ avrà a che fare, a livello di matrici associate, con la moltiplicazione a destra per la matrice $\mathcal{A}_\varphi$.

Definizione 3.2.5. Dato un P-morfismo $\varphi : P^r \longrightarrow P^{r'}$ definiamo un P-morfismo

$$\overline{\varphi} : \text{Mat }_{sr'}(P) \longrightarrow \text{Mat }_{sr}(P), \quad \overline{\varphi}(\mathcal{B}) = \mathcal{B} \cdot \mathcal{A}_\varphi, \quad \forall \mathcal{B} \in \text{Mat }_{sr'}(P)$$

Abbiamo così la possibilità di scrivere il seguente diagramma commutativo:

$$\begin{array}{ccc} \text{Hom}(P^{r'}, P^s) & \xrightarrow{\varphi^*} & \text{Hom}(P^r, P^s) \\ \downarrow{\scriptstyle \Lambda_{r's}} & & \downarrow{\scriptstyle \Lambda_{rs}} \\ \text{Mat }_{sr'}(P) & \xrightarrow{\overline{\varphi}} & \text{Mat }_{sr}(P) \end{array} \tag{1}$$

Definiamo ora un'operazione tra matrici detta **prodotto tensore di matrici** o **prodotto di Kronecker**.

Definizione 3.2.6. Siano $\mathcal{A} = (a_{ij}) \in \text{Mat }_{rr'}(P)$ e $\mathcal{B} = (b_{ij}) \in \text{Mat }_{ss'}(P)$ due matrici. Allora la matrice a blocchi

$$\begin{pmatrix} a_{11}\mathcal{B} & \dots & a_{ir'}\mathcal{B} \\ \vdots & \ddots & \vdots \\ a_{r1}\mathcal{B} & \dots & a_{rr'}\mathcal{B} \end{pmatrix} \quad \in \text{Mat }_{rs \times r's'}(P)$$

si dice prodotto tensore delle matrici $\mathcal{A}$ e $\mathcal{B}$ e si indica con $\mathcal{A} \otimes \mathcal{B}$.

Proposizione 3.2.7. *Sia $\varphi : P^r \longrightarrow P^{r'}$ un P-morfismo, $\varphi^* = \text{Hom}(\varphi, P^s)$, $\overline{\varphi}$ la mappa definita come moltiplicazione a destra per $\mathcal{A}_\varphi$ e $\widetilde{\varphi}$ la mappa la cui matrice associata è ${}^t\mathcal{A}_\varphi \otimes \mathcal{I}_s$. Allora il seguente diagramma è commutativo:*

$$\begin{array}{ccc} \text{Mat }_{sr}(P) & \xrightarrow{\overline{\varphi}} & \text{Mat }_{sr}(P) \\ \downarrow{\scriptstyle \text{Fl}_{sr'}} & & \downarrow{\scriptstyle \text{Fl}_{sr}} \\ P^{r's} & \xrightarrow{\widetilde{\varphi}} & P^{rs} \end{array} \tag{2}$$

Dimostrazione. Per la verifica di questo fatto, così come per tutte le dimostrazioni di questo paragrafo che non riportiamo, si veda [KR]. □

Combinando i diagrammi commutativi (1) e (2), siccome le frecce verticali sono tutti isomorfismi, ne segue che la matrice ${}^t\mathcal{A}_\varphi \otimes \mathcal{I}_s$ è adatta a rappresentare il P-morfismo φ^*, almeno nel caso in cui i due moduli U e V sono

liberi. Prima di fare il passo successivo vediamo una proprietà di esattezza del funtore Hom. Tale proprietà si esprime dicendo che il funtore $\mathrm{Hom}(\cdot, V)$ è "esatto a sinistra".

Proposizione 3.2.8. *Sia $U_1 \xrightarrow{\varphi} U_2 \xrightarrow{\psi} U_3 \longrightarrow 0$ una sequenza esatta di P-moduli, sia V un P-modulo, $\varphi^* = \mathrm{Hom}(\varphi, V)$ e $\psi^* = \mathrm{Hom}(\psi, V)$. Allora*

$$0 \longrightarrow \mathrm{Hom}(U_3, V) \xrightarrow{\psi^*} \mathrm{Hom}(U_2, V) \xrightarrow{\varphi^*} \mathrm{Hom}(U_1, V)$$

è una sequenza esatta di P-moduli.

Con questi ingredienti siamo ora pronti per fornire il teorema in cui è racchiuso l'algoritmo per la rappresentazione di Hom, a meno di enunciare un lemma di tipo tecnico in cui si vede come le sizigie siano fondamentali per il calcolo e la presentazione di nuclei. Il modulo Hom cercato infatti sarà ottenuto proprio come nucleo di una opportuna mappa P-lineare.

Lemma 3.2.9. *Siano $M = \langle g_1, \ldots, g_\alpha \rangle \subseteq P^r$ e $N = \langle h_1, \ldots, h_\beta \rangle \subseteq P^s$ due sottomoduli, sia $\varphi : P^r/M \longrightarrow P^s/N$ un P-morfismo. Scriveremo*

$$\varphi(e_j + M) = w_j + M$$

dove gli e_j formano la base canonica di P^r e i $w_j = (f_{1j}, \ldots, f_{sj})$ sono rappresentanti delle immagini in P^s/N. Consideriamo $\{v_1, \ldots, v_u\}$ un sistema di generatori per $\mathrm{Syz}(w_1, \ldots, w_r, h_1, \ldots, h_\beta)$. Per ogni $j = 1..u$ scriviamo tali generatori nella forma $v_j = (k_{1j}, \ldots, k_{r+\beta\, j})$. Allora

$$\mathrm{Ker}(\varphi) = \langle (k_{1j}, \ldots, k_{rj}) + M \mid j = 1..u \rangle$$

Detta poi ψ la mappa definita da $\psi(e_j) = (k_{1j}, \ldots, k_{rj})$, $j = 1..u$, π la proiezione canonica di P^r sul quoziente P^r/M, calcoliamo un sistema di generatori $\{(l_{1j}, \ldots, l_{u+\alpha\, j}) \mid j = 1..u'\}$ per il modulo

$$\mathrm{Syz}(\psi(e_1), \ldots, \psi(e_u), g_1, \ldots, g_\alpha)$$

allora una presentazione del nucleo di φ si ottiene dalla seguente sequenza esatta:

$$P^{u'} \xrightarrow{\psi'} P^u \xrightarrow{\pi \circ \psi} \mathrm{Ker}(\varphi) \longrightarrow 0, \quad \textit{ovvero} \quad \mathrm{Ker}(\varphi) \cong P^u / \mathrm{Im}(\psi')$$

dove ψ' è data da $\psi'(e_j) = (l_{1j}, \ldots, l_{uj})$, $\quad j = 1..u'$.

Teorema 3.2.10 (Calcolo esplicito dei moduli Hom). *Siano $M = \langle g_1, \ldots, g_\alpha \rangle \subseteq P^r$ e $N = \langle h_1, \ldots, h_\beta \rangle \subseteq P^s$ due sottomoduli. Sia $\mathcal{G}$ la matrice $r \times \alpha$ le cui colonne sono i vettori $g_1, \ldots, g_\alpha$, e sia $\mathcal{H}$ la matrice $s \times \beta$*

le cui colonne sono i vettori $h_1, \ldots, h_\beta$. *Chiamiamo* U *il sottomodulo di* P^{rs} *generato dalle colonne della matrice* $\mathcal{I}_r \otimes \mathcal{H}$ *e* V *il sottomodulo di* $P^{\alpha s}$ *generato dalle colonne della matrice* $\mathcal{I}_\alpha \otimes \mathcal{H}$. *Indichiamo poi con* $\lambda : P^{rs} \longrightarrow P^{\alpha s}$ *la mappa* P*-lineare che ha come matrice associata* ${}^t\mathcal{G} \otimes \mathcal{I}_s$. *Allora valgono i seguenti fatti:*

1) la mappa λ *soddisfa* $\lambda(U) \subseteq V$ *e induce quindi un* P*-morfismo* $\overline{\lambda} : P^{rs}/U \longrightarrow P^{\alpha s}/V$.

2) $\mathrm{Hom}(P^r/M, P^s/N) \cong \mathrm{Ker}\,(\overline{\lambda})$ *e quindi una presentazione di* Hom *può essere calcolata esplicitamente usando il lemma 3.2.9.*

3) sia $\vartheta \in \mathrm{Hom}(P^r/M, P^s/N)$ *rappresentato, come elemento di* $\mathrm{Ker}\,(\overline{\lambda})$, *dalla classe* $(a_{11}, a_{21}, \ldots, a_{s1}, \ldots\ldots, a_{r1}, a_{r2}, \ldots, a_{rs}) + U$. *Allora la mappa* ϑ *è indotta dal* P*-morfismo* $\Theta : P^r \longrightarrow P^s$ *la cui matrice associata è* $\mathcal{A}_\Theta = (a_{ij})$.

Il teorema 3.2.10 risolve il problema che ci eravamo posti. Basta infatti calcolare il prodotto tensore di alcune matrici e definire la mappa λ. Per il punto 1) tale mappa induce un morfismo tra i quozienti di opportuni moduli liberi e se ne può calcolare quindi il Ker come illustrato nel lemma 3.2.9. Dunque abbiamo ottenuto una presentazione del modulo Hom richiesto. Non solo, ma il punto 3) ci dice anche come reinterpretare gli elementi del $\mathrm{Ker}\,(\overline{\lambda})$ come mappe. Basta di fatto convertire un vettore rappresentante di ϑ in P^{rs} in una matrice, iniziando a "riempire" la prima colonna con i primi s elementi del vettore e proseguendo colonna per colonna. Tale matrice che si ottiene è la matrice associata alla mappa Θ che solleva il morfismo $\vartheta \in \mathrm{Hom}(P^r/M, P^s/N)$.

Giunti a questo punto, gli ingredienti formali per la definizione di Ext ci sono tutti. Nel prossimo paragrafo daremo solo la sua definizione astratta, rimandando poi al capitolo successivo l'algoritmo che permette di calcolarlo esplicitamente.

3.3 Definizione del funtore Ext

Sappiamo già cosa si intende per funtore e per funtore derivato. Quindi dare una definizione precisa di Ext risulta semplice:

Definizione 3.3.1. Il funtore **Ext** è il funtore derivato destro del funtore Hom controvariante.

Avevamo già di fatto anticipato questa definizione, almeno in modo implicito. Osserviamo che tale definizione, essendo di natura puramente categoriale, puo essere applicata al funtore $Hom(\cdot, V)$ inteso come funtore tra R-moduli qualunque, dove R è un anello qualunque, oppure a strutture algebriche diverse per le quali abbia senso considerare il funtore Hom. Tuttavia noi ci limitiamo a considerarlo come funtore tra P-moduli finitamente generati, in modo che valga il Teorema delle Sizigie di Hilbert. Ci limitiamo qui di seguito ad elencare alcune proprietà di Ext che derivano facilmente dalla sua definizione e dalla teoria generale dei funtori derivati (si veda ad esempio [G]).

Proposizione 3.3.2. *Siano U, V e W tre P-moduli finitamente generati. Valgono le seguenti proprietà:*

1) $\mathrm{Ext}^{\mathrm{i}}(U, V)$ *è ben definito per ogni i, cioè non dipende dalla risoluzione scelta per U, ed è a sua volta un P-modulo finitamente generato.*

2) $\mathrm{Ext}^{\mathrm{i}}(\cdot, V)$ *è un funtore additivo controvariante per ogni i.*

3) $\mathrm{Ext}^0(U, V) = \mathrm{Hom}(U, V)$

4) Se $0 \longrightarrow U \longrightarrow V \longrightarrow W \longrightarrow 0$ *è una sequenza esatta di P-moduli e M un ulteriore P-modulo, allora risulta definita una sequenza esatta lunga:*

$$0 \longrightarrow \mathrm{Hom}(W, M) \longrightarrow \mathrm{Hom}(V, M) \longrightarrow \mathrm{Hom}(U, M) \longrightarrow \mathrm{Ext}^1(W, M) \longrightarrow$$
$$\longrightarrow \mathrm{Ext}^1(V, M) \longrightarrow \mathrm{Ext}^1(U, M) \longrightarrow \mathrm{Ext}^2(W, M) \longrightarrow \dots$$

5) Se U è libero allora $\mathrm{Ext}^{\mathrm{i}}(U, V) = 0$ *per ogni $i > 0$.*

Vediamo anche alcune definizioni di carattere puramente algebrico che mettono in luce lo stretto legame tra Ext e le proprietà algebriche di un P-modulo M.

Definizione 3.3.3. Sia $I \subseteq P$ un ideale omogeneo, M un P-modulo. Diciamo che una r-upla $J = (F_1, \dots, F_r)$ di elementi omogenei di I è una M-**successione regolare** in I se:

$$F_1 \nmid 0 \in M, \quad F_2 \nmid \overline{0} \in M/(F_1)M, \quad \dots \quad F_r \nmid \overline{0} \in M/(F_1, \dots, F_{r-1})M.$$

La massima lunghezza di una M-successione regolare in I si dice **profondità** di M rispetto a I e si indica con $\mathrm{depth}_I(M)$. Se come I si considera l'ideale irrilevante di P, $I_{max} = (x_1, \dots, x_n)$, si può parlare allora semplicemente di profondità del modulo M che si indica con $\mathrm{depth}(M)$.

Teorema 3.3.4 (Décalage). *Sia $I \subseteq P$ un ideale omogeneo, M un P-modulo e $J = (F_1, \dots, F_r)$ una M-successione regolare in I. Allora*

$$\mathrm{Ext}^{\mathrm{r}}(P/I, M) \cong \mathrm{Hom}(P/I, M/JM)$$

dove con JM abbiamo indicato il sottomodulo $(F_1, \dots, F_r)M$.

Proposizione 3.3.5. *Esiste una M-succesione regolare in I di lunghezza r se e solo se va la relazione $\mathrm{Ext}^{\mathrm{i}}(P/I, M) = 0$ per ogni $i < r$.*

Come corollario della proposizione otteniamo un modo di calcolare la profondità del modulo M: basta calcolare gli $\mathrm{Ext}^{\mathrm{i}}(P/I, M)$ e vedere qual è il primo indice i per cui Ext non si annulla.

Corollario 3.3.6. *Sia M un P modulo, I un ideale omogeneo di P. Allora*

$$\mathrm{depth}_I(M) = \min\{i \in \mathbb{N} \mid \mathrm{Ext}^{\mathrm{i}}(P/I, M) \neq 0\}$$

e in particolare

$$\mathrm{depth}(M) = \min\{i \in \mathbb{N} \mid \mathrm{Ext}^{\mathrm{i}}(K, M) \neq 0\}$$

4. Algoritmo per il calcolo di Ext

Dopo aver dato la definizione formale è giunto il momento di vedere come calcolare i moduli del tipo $\mathrm{Ext}^{\mathrm{i}}(P^r/M, P^s/N)$. Nel primo paragrafo ci occuperemo dell'algoritmo che abbiamo creato, della sua correttezza e della sua finitezza. Nel secondo paragrafo ci occuperemo di trasformare tale sequenza di istruzioni in un opportuno codice CoCoA, spiegandone la sintassi e l'utilizzo pratico. Nel terzo vedremo come simulare al calcolatore la gestione degli shift nel caso in cui la graduazione indotta da P su P^r non sia quella canonica.

4.1 Come calcolare Ext

Le notazioni utilizzate sono le stesse del capitolo precedente. Prima di presentare l'algoritmo, però, introduciamo alcuni lemmi che saranno necessari per la dimostrazione. Il primo di questi ci servirà per poter rappresentare le mappe d_i^* che intervengono nel complesso di coomologia da cui si ricava Ext.

Lemma 4.1.1. *Nelle ipotesi del Teorema 3.2.10, indichiamo con*

$$0 \longrightarrow P^{r_m} \xrightarrow{d_m} \dots \xrightarrow{d_{i+1}} P^{r_i} \xrightarrow{d_i} P^{r_{i-1}} \xrightarrow{d_{i-1}} \dots \xrightarrow{d_1} P^{r_0} \xrightarrow{d_o} P^r/M \longrightarrow 0$$

una risoluzione libera del modulo P^r/M, di lunghezza m. Applichiamo il funtore $\mathrm{Hom}(\cdot, P^s/N)$ *e otteniamo il complesso*

$$0 \longrightarrow \mathrm{Hom}(P^{r_0}, P^s/N) \xrightarrow{d_1^*} \dots \xrightarrow{d_{i-1}^*} \mathrm{Hom}(P^{r_{i-1}}, P^s/N) \xrightarrow{d_i^*} \dots$$

Sia U_i, $i = 1..m$ il modulo generato dalle colonne della matrice $\mathcal{I}_{r_i} \otimes \mathcal{H}$, e siano $\Phi_{r_i,s}$ gli isomorfismi

$$\Phi_{r_i,s} : \mathrm{Hom}(P^{r_i}, P^s/N) \longrightarrow \mathrm{Ker}\,(\overline{\lambda_i}) \subseteq P^{r_i s}/U_i$$

che nascono dall'applicazione del teorema 3.2.10. Allora possiamo costruire il seguente diagramma commutativo:

$$\begin{array}{ccccc}
\mathrm{Hom}(P^{r_{i-1}}, P^s/N) & \xrightarrow{d_i^*} & \mathrm{Hom}(P^{r_i}, P^s/N) & \xrightarrow{d_{i+1}^*} & \mathrm{Hom}(P^{r_{i+1}}, P^s/N) \\
\downarrow{\scriptstyle \Phi_{r_{i-1},s}} & & \downarrow{\scriptstyle \Phi_{r_i,s}} & & \downarrow{\scriptstyle \Phi_{r_{i+1},s}} \\
\mathrm{Ker}\,(\overline{\lambda_{i-1}}) & \xrightarrow{\delta_i} & \mathrm{Ker}\,(\overline{\lambda_i}) & \xrightarrow{\delta_{i+1}} & \mathrm{Ker}\,(\overline{\lambda_{i+1}}) \\
\cap & & \cap & & \cap \\
P^{r_{i-1}}/U_{i-1} & & P^{r_i}/U_i & & P^{r_{i+1}}/U_{i+1}
\end{array}$$

dove le mappe δ_i sono definite da $\delta_i := \Phi_{r_i,s} \circ d_i^* \circ \Phi_{r_{i-1},s}^{-1}$. Vale allora l'isomorfismo

$$\mathrm{Ext}^{\mathrm{i}}(P^r/M, P^s/N) \cong \frac{\mathrm{Ker}\,(\delta_{i+1})}{\mathrm{Im}\,(\delta_i)}$$

Dimostrazione. La commutatività del diagramma segue dalla definizione dei morfismi δ_i. L'isomorfismo invece nasce dall'osservazione che, essendo i $\Phi_{r_i,s}$ isomorfismi, risulta $\mathrm{Ker}\,(d_i^*) \cong \mathrm{Ker}\,(\delta_i)$ e $\mathrm{Im}\,(d_i^*) \cong \mathrm{Im}\,(\delta_i)$. La tesi segue allora per definizione di Ext. □

Il secondo lemma invece ci dice come rappresentare un quoziente del tipo

$$M/(M \cap N)$$

e noi lo applicheremo nel caso particolare $N \subseteq M$ per poter rappresentare un quoziente del tipo tipo M/N.

Lemma 4.1.2. *Siano $M = \langle g_1, \ldots, g_s \rangle$ $N = \langle h_1, \ldots, h_t \rangle$ due sottomoduli di P^r e sia $\lambda : P^s \longrightarrow P^r$ il P-morfismo dato da $\lambda(\varepsilon_i) = g_i$ per $i = 1..s$. Sia $\{v_1, \ldots, v_u\}$ un insieme di generatori per il P-modulo $\mathrm{Syz}(g_1, \ldots, g_s, h_1, \ldots, h_t) \subseteq P^{s+t}$, e sia $v_j = (f_{1j}, \ldots, f_{s+t\,j})$ dove $f_{ij} \in P$. Definiamo i vettori di P^s:*

$$w_j = (f_{1j}, \ldots, f_{sj}), \quad j = 1..u$$

allora il morfismo λ induce un epimorfismo $\overline{\lambda} : P^s \longrightarrow M/(M \cap N)$. Inoltre, definendo $\psi : P^u \longrightarrow P^s$ come $\psi(\varepsilon_j) = w_j$, $j = 1..u$, si ottiene la presentazione

$$P^u \xrightarrow{\psi} P^s \xrightarrow{\overline{\lambda}} M/(M \cap N) \longrightarrow 0, \quad \textit{ovvero} \quad M/(M \cap N) \cong P^s/\langle w_1, \ldots, w_u \rangle$$

Dimostrazione. Si veda il corollario 3.2.6 di [KR]. □

Teorema 4.1.3 (Algorimo per il calcolo di Ext).
Siano $M = \langle g_1, \ldots, g_\alpha \rangle \subseteq P^r$ e $N = \langle h_1, \ldots, h_\beta \rangle \subseteq P^s$ due P-moduli finitamente generati, di cui M omogeneo. Consideriamo la seguente lista di istruzioni:

1) Si ponga $r_0 := r$ e $r_1 := \alpha$, e si definisca il morfismo $d_1 : P^{r_1} \longrightarrow P^{r_0}$ in questo modo: $d_1(e_j) = g_j$, $j = 1..\alpha$.

2) Si calcoli $\mathrm{Syz}(g_1, \ldots, g_\alpha) = \langle v_1, \ldots, v_{r_2} \rangle$ con il teorema 3.1.7, utilizzando generatori minimali per il modulo delle sizigie come discusso nell'osservazione 3.1.10.

3) Si definisca il morfismo $d_2 : P^{r_2} \longrightarrow P^{r_1}$ dato da $d_2(e_j) = v_j$.

4) Si ripetano i punti 2 e 3 calcolando di volta in volta il modulo delle sizigie con generatori minimali, definendo le mappe $d_j : P^{r_j} \longrightarrow P^{r_{j-1}}$ in modo analogo, fino a quando il modulo delle sizigie trovato non si annulla. Chiamiamo d_m l'ultima mappa non nulla definita.

5) Si costruisca in tal modo una risoluzione libera per P^r/M:

$$0 \longrightarrow P^{r_m} \xrightarrow{d_m} P^{r_{m-1}} \xrightarrow{d_{m-1}} \ldots \xrightarrow{d_2} P^{r_1} \xrightarrow{d_1} P^r \longrightarrow 0$$

6) Si definiscano le matrici A_i associate alle mappe d_i.

7) Si applichi il funtore $\mathrm{Hom}(\cdot, P^s/N)$ *alla risoluzione del punto 5 per ottenere il complesso*

$$0 \longrightarrow \mathrm{Hom}(P^r, P^s/N) \xrightarrow{d_1^*} \dots \xrightarrow{d_m^*} \mathrm{Hom}(P^{r_m}, P^s/N) \longrightarrow 0$$

8) Si trovi una presentazione dei moduli $\mathrm{Hom}(P^{r_i}, P^s/N)$, $i = 1..m$ *come illustrato nel teorema 3.2.10, e si definiscano le mappe* δ_i *come nel lemma 4.1.1.*

9) Si costruiscano le matrici associate alle mappe δ_i, $D_i := ({}^tA_i) \otimes \mathcal{I}_s$ *in base alla definizione 3.2.6.*

10) Si calcolino, esplicitandone i generatori, i moduli

$$Z^i := Ker(\delta_{i+1}) \subseteq P^{sr_{i+1}}$$

come indicato nella prima parte del lemma 3.2.9, e i moduli

$$B^i = \mathrm{Im}\,(\delta_i) \subseteq P^{sr_{i+1}}$$

generati dalle colonne delle matrici D_i *definite nel punto 9.*

11) Si trovi una presentazione per i moduli $H^i := \frac{Z^i}{B^i}$, $i = 1..m$, *come indicato nel lemma 4.1.2.*

12) Si calcoli $H^0 = \mathrm{Hom}(P^r/M, P^s/N)$ *usando ancora il teorema 3.2.10, e si ponga invece* $H^i = 0$ *per ogni* $i > m$.

Tale procedimento è un algoritmo e i moduli H^i *che si ottengono forniscono delle presentazioni per i moduli* $\mathrm{Ext^i}(P^r/M, P^s/N)$, *per ogni* $i \in \mathbb{N}$.

Dimostrazione. I passi da 1) a 5) non sono altro che la costruzione di una risoluzione libera graduata per il modulo M così come mostrato nella proposizione 3.1.16. Al solito, la finitezza della risoluzione è garantita dal teorema 3.1.30. Nei punti 6), 7) e 8) si costruisce il complesso di coomologia associato alla risoluzione. Soffermiamoci sul punto 9). Osservando il diagramma commutativo del lemma 4.1.1 possiamo concludere che le mappe δ_j sono indotte dalle mappe d_j^*, le quali, a meno di considerare opportune proiezioni sul quoziente, possono essere rappresentate come mappe tra moduli liberi e quindi hanno come matrici associate proprio $D_j = ({}^tA_j) \otimes \mathcal{I}_s$, così come indicato nella proposizione 3.2.7. I punti 10) e 11) calcolano gli Ext utilizzando l'isomorfismo del lemma 4.1.1 e la presentazione di un quoziente

fornita dal lemma 4.1.2. Osserviamo solo che tale lemma può essere applicato perché la sequenza costruita al punto 7) è un complesso, e quindi risulta $\mathrm{Im}\,(\delta_i) \subseteq \mathrm{Ker}\,(\delta_{i+1})$. Il punto 12) calcola invece $\mathrm{Ext}^0(P^r/M, P^s/N)$ in base alla proprietà 3) della proposizione 3.3.2, mentre gli $\mathrm{Ext}^{\mathrm{i}}(P^r/M, P^s/N)$, con $i > m$, sono nulli tenendo conto del fatto che la risoluzione è lunga m e quindi le coomologie sono nulle dal punto $m+1$ in poi. □

Osservazione 4.1.4. I primi 5 passi dell'algoritmo ripropongono il procedimento di calcolo di una risoluzione libera graduata minimale che avevamo già visto nella dimostrazione della proposizione 3.1.16. In realtà può essere utilizzato un qualunque procedimento il cui risultato sia una risoluzione libera per il modulo P^r/M, anche non minimale, perché comunque il calcolo di Ext non dipende dalla risoluzione scelta, come afferma la proposizione 3.1.26.

4.2 Presentazione del pacchetto CoCoA per il calcolo di Ext

In questo paragrafo spiegheremo nei dettagli la sintassi e il funzionamento del pacchetto di funzioni messo a punto per il calcolo di Ext. Per ciascuna funzione definita illustreremo la scelta dei parametri, il contenuto e l'output, precisando di volta in volta a quale passo dell'algoritmo 4.1.3 ci si sta riferendo. Verranno presentate anche le funzioni del pacchetto per il calcolo di Hom creato da Massimo Caboara, alle quali sono state aggiunte alcuni funzioni che permettono di presentare i moduli tipo Hom e Ker sotto forma di quoziente e non solo con generatori. Non sono state utilizzate variabili globali. La versione di CoCoA di cui ci è serviti è la versione 4.1 per sistema Linux.

Prima di introdurre il pacchetto vero e proprio, vediamo le funzioni che fanno parte del pacchetto Hom, al quale abbiamo apportato qualche aggiunta. La funzione **FCMod** prende in input una lista **E** di vettori ed elimina da ogni vettore le ultime componenti, lasciando solo le prime **ToLive** componenti.

```
Define FCMod(E,ToLive)
 L:=[];
 Foreach V In E Do
  Append(L,First(V,ToLive));
End;
Return L;
End;
```

GensToMat crea la matrice che ha come colonne i generatori di **M**

```
Define GensToMat(M)
 Return Transposed(Cast(Gens(M),MAT));
EndDefine;
```

MatToModule è il modulo generato dalle colonne della matrice **A**

```
Define MatToModule(A)
 Return Module(Transposed(A));
EndDefine;
```

La funzione **TensorMat** calcola il prodotto tensore di due matrici **A** e **B**

```
Define TensorMat(A,B)
 RowNum:=Len(A);
 ColNum:=Len(A[1]);
 T1:=[[A[I,J]*B|J In 1..ColNum]|I In 1..RowNum];
```

```
 Return $cocoa/mat.BlockMatrix(T1);
End;
```

Ora vediamo le funzioni che calcolano i nuclei. **KerR_MR_N** calcola il nucleo di una mappa **Phi** tra i due moduli $P^r/\mathbf{M}$ e $P^s/\mathbf{N}$ utilizzando la prima parte del lemma 3.2.9, fornendo come output i geneatori del modulo Ker (**Phi**)

```
Define KerR_MR_N(M,N,Phi)
 L:=Concat(Phi,Gens(N));
 S:=Syz(L);
 K:=Module(FCMod(Gens(S),NumComps(M)));
 Return Module(Interreduced(Gens(K)));
End;
```

Abbiamo aggiunto la funzione **PresentKerR_MR_N** che fa la stessa cosa ma fornisce in output una presentazione del nucleo, come indicato nella seconda parte del lemma 3.2.9.

```
Define PresentKerR_MR_N(M,N,Phi);
 K:=KerR_MR_N(M,N,Phi);
 If K=Module([0]) Then
   Psi:=[Vector(0)];
 Else
   Psi:=Gens(K);
 EndIf;
 U:=Len(Psi);
 T:=Syz(Concat(Psi,Gens(M)));
 Den:=Module(FCMod(Gens(T),U));
 Den:=Module(Interreduced(Gens(Den)));
 If Den=CurrentRing()^U Then
   Return ZeroModule(Den);
 Elsif Den=Module([0]) Then
   Den:=ZeroModule(CurrentRing()^U);
   Return CurrentRing()^U/Den;
 EndIf;
 Return CurrentRing()^U/Den;
EndDefine;
```

HomR_MR_N calcola il modulo $\mathrm{Hom}(P^r/\mathbf{M}, P^s/\mathbf{N})$ seguendo il teorema 3.2.10 e fornendo dei generatori

```
Define HomR_MR_N(M,N)
 U:=Module(TensorMat(Identity(NumComps(M)),Cast(Gens(N),MAT)));
 V:=Module(TensorMat(Identity(Len(M)),Cast(Gens(N),MAT)));
 Phi:=TensorMat(Cast(Gens(M),MAT),Identity(NumComps(N)));
```

```
 Phi:=Transposed(Phi);
 Phi:=[Vector(V)|V In Phi];
 Return KerR_MR_N(U,V,Phi);
End;
```

Anche in questo caso abbiamo aggiunto la funzione che invece fornisce una presentazione

```
Define PresentHomR_MR_N(M,N)
 U:=Module(TensorMat(Identity(NumComps(M)),Cast(Gens(N),MAT)));
 V:=Module(TensorMat(Identity(Len(M)),Cast(Gens(N),MAT)));
 Phi:=TensorMat(Cast(Gens(M),MAT),Identity(NumComps(N)));
 Phi:=Transposed(Phi);
 Phi:=[Vector(V)|V In Phi];
 Return PresentKerR_MR_N(U,V,Phi);
End;
```

Di seguito riportiamo le due funzioni **Ker** e **Hom** che calcolano un nucleo e un modulo Hom rispettivamente. Esse servono a distinguere i casi in cui l'input è libero oppure un quoziente, e forniscono come output una lista di generatori.

```
Define Ker(Var R_M, Var R_N, Phi)
 If Type(R_M) = TAGGED("Quotient") Then
   M := @R_M;
    If Type(M)=IDEAL Then
      M:=Module(M);
    EndIf;
 Elsif Type(R_M) = MODULE Then
   M := Module(Vector([0|I In 1..NumComps(R_M)]));
 Else
   Return ERR.BAD_PARAMS;
 EndIf;
 If Type(R_N) = TAGGED("Quotient") Then
   N := @R_N;
    If Type(N)=IDEAL Then
      N:=Module(N);
    EndIf;
 Elsif Type(R_N) = MODULE Then
   N := Module(Vector([0|I In 1..NumComps(R_N)]));
 Else
   Return ERR.BAD_PARAMS;
 EndIf;
 Return KerR_MR_N(M,N,Phi)
EndDefine;
```

```
Define Hom(Var R_M, Var R_N)
 If Type(R_M) = TAGGED("Quotient") Then
   M := @R_M;
    If Type(M)=IDEAL Then
     M:=Module(M);
    EndIf;
 Elsif Type(R_M) = MODULE Then
   M := Module(Vector([0|I In 1..NumComps(R_M)]));
 Else
   Return ERR.BAD_PARAMS;
 EndIf;
 If Type(R_N) = TAGGED("Quotient") Then
   N :=@R_N;
    If Type(N)=IDEAL Then
      N:=Module(N);
    EndIf;
 Elsif Type(R_N) = MODULE Then
   N := Module(Vector([0|I In 1..NumComps(R_N)]));
 Else
   Return ERR.BAD_PARAMS;
 EndIf;
 Return HomR_MR_N(M,N)
EndDefine;
```

Abbiamo aggiunto a queste due funzioni le versioni che calcolano le presentazioni:

```
Define PresentKer(Var R_M, Var R_N, Phi)
 If Type(R_M) = TAGGED("Quotient") Then
   M := @R_M;
    If Type(M)=IDEAL Then
      M:=Module(M);
    EndIf;
 Elsif Type(R_M) = MODULE Then
   M := Module(Vector([0|I In 1..NumComps(R_M)]));
 Else
   Return ERR.BAD_PARAMS;
 EndIf;
 If Type(R_N) = TAGGED("Quotient") Then
   N := @R_N;
    If Type(N)=IDEAL Then
      N:=Module(N);
    EndIf;
 Elsif Type(R_N) = MODULE Then
```

```
    N := Module(Vector([0|I In 1..NumComps(R_N)]));
  Else
    Return ERR.BAD_PARAMS;
  EndIf;
    Return PresentKerR_MR_N(M,N,Phi)
EndDefine;

Define PresentHom(Var R_M, Var R_N)
 If Type(R_M) = TAGGED("Quotient") Then
   M := @R_M;
    If Type(M)=IDEAL Then
      M:=Module(M);
    EndIf;
 Elsif Type(R_M) = MODULE Then
   M := Module(Vector([0|I In 1..NumComps(R_M)]));
 Else
   Return ERR.BAD_PARAMS;
 EndIf;
 If Type(R_N) = TAGGED("Quotient") Then
   N :=@R_N;
    If Type(N)=IDEAL Then
      N:=Module(N);
    EndIf;
 Elsif Type(R_N) = MODULE Then
   N := Module(Vector([0|I In 1..NumComps(R_N)]));
 Else
   Return ERR.BAD_PARAMS;
EndIf;
  Return PresentHomR_MR_N(M,N)
EndDefine;
```

Questo conclude la presentazione del pacchetto Hom di CoCoA nella sua ultima versione. Passiamo ora al pacchetto Ext.

Cominciamo a definire due funzioni che calcolino, a partire da una risoluzione, l' **I**-esimo numero di Betti. Una risoluzione in CoCoA è una lista in cui compaiono numerosi oggetti. Se consideriamo ad esempio il modulo omogeneo

$$M := R/(x^2, x+3z, y^3) \quad dove\ R = \mathbb{Q}[x,y,z]$$

CoCoA fornisce la seguente risoluzione, con il comando **Res(M)**:

$$0 \to R(-6) \to R(-3) \bigoplus R(-4) \bigoplus R(-5) \to R(-1) \bigoplus R(-2) \bigoplus R(-3) \to R$$

Se chiediamo a CoCoA di descrivere l'oggetto appena calcolato, otteniamo una lista il cui primo elemento è l'anello R in cui si sta lavorando, mentre il secondo elemento è una lista di questa forma:

```
[
 [Module([x^2], [x + 3z], [y^3 + xyz]), [1, 0]],
 [Module([x^2], [x + 3z], [y^3 + xyz]),
    [1, -1], [1, -2], [1, -3]],
 [Module(Shifts([x, z^2, y^3]], [z^2, -1/9x - 1/3z, 0],
 [y^3, 0, -x-3z], [0, 1/9y^3, -z^2]),
    [1, -3], [1, -4], [1, -5]],
 [Module(Shifts([xz^2, xy^3, y^3z^2]), [y^3, -z^2, x + 3z]),
    [1, -6]]
]
```

Si tratta di una lista di liste, ciascuna delle quali contiene l'immagine di una mappa della risoluzione, a partire da quella più a destra, e i numeri di Betti graduati con il rispettivo shift. Ad esempio nel nostro caso il primo modulo è libero di rango 1 e senza shift, come indica nella prima riga la coppia $[1, 0]$, mentre il secondo modulo è la somma diretta di un modulo libero di rango 1 shiftato di -1 (coppia $[1, -1]$ nella seconda riga), un altro modulo libero di rango 1 e shiftato di -2 (coppia $[1, -2]$) e un terzo di rango 1 e shiftato di -3 (coppia $[1, -3]$).
Dunque nella nostra funzione prendiamo in input una risoluzione **RES** e l'indice **I** del numero di betti (corrispondente all'indice i della definizione 3.1.29) da calcolare, e sommando i vari ranghi dei moduli liberi otteniamo ciò che vogliamo.

```
Define BettiExt(RES,I);
 M:=RES[2][1][1]; -- il primo modulo nella RES
 If I=0 Then Return
   NumComps(M);
 EndIf;
 MAPS:=RES[2];
 If I>Len(MAPS) Then
   Return 0;
 EndIf;
 MyMap:=MAPS[I];
   Return Sum([MyMap[J][1]|J In 2..Len(MyMap)]);
   --sommo gli esponenti degli shift
 EndDefine;
```

Scriviamo anche una funzione che ci permetta di calcolare l'ultimo numero di Betti non nullo, andando a vedere la lunghezza della risoluzione, m, e

calcolando l'm-esimo numero di Betti con la funzione percedente.

```
Define LastBetti(RES);
 MAPS:=RES[2];
 LastIndex:=Len(MAPS);
 Return BettiExt(RES,LastIndex)
EndDefine;
```

Ora voglio costruire le mappe della risoluzione, come al punto 6) dell'algoritmo. Una mappa lineare per CoCoA non è altro che una lista di vettori: le immagini della base canonica. Di fatto quindi è la lista delle colonne della matrice associata. L'input **P** può essere sia il modulo di cui si calcola la risoluzione, sia la risoluzione stessa. Le funzioni principali infatti saranno fatte tali per cui verrà calcolata la risoluzione una volta sola, essendo questo un processo piuttosto laborioso dal punto di vista computazionale che peserebbe eccessivamente sulla velocità dell'algoritmo.

```
Define ResMap(P,I);
If Type(P)=TAGGED("Quotient") Then
  M:=@P;
  RES:=Res(M);
Elsif Type(P)=TAGGED("$cocoa/gb.Res") Then
  M:=P[2][1][1];
  RES:=P;
Else Error('Cannot compute ResMap');
EndIf;
MAPS:=RES[2];
NMaps:=Len(MAPS)+1;
If I<=0 Then
  Error('Index must be positive ');
Elsif I=NMaps Then
  Return [NewVector(LastBetti(RES),0)];-- mappa nulla
Elsif I>NMaps Then
  Return [Vector(0)];
Elsif I=1 Then
  Return Gens(Module(Interreduced(Gens(M))));
Else
  S:=Module(Interreduced(Gens(M)));
  For J:=2 To I Do
     S:=Module(Interreduced(Gens(SyzOfGens(S))));
  EndFor;
   Return Gens(S);
EndIf;
EndDefine;
```

La funzione **HomResMap** definisce invece le mappe del complesso di coomologia ottenuto applicando alla risoluzione il funtore Hom. La sintassi è simile a **ResMap**, anche qui dunque sono previsti i casi in cui l'input è il modulo o la sua risoluzione. Come indicato dal punto 9) dell'algoritmo, le matrici associate si costruiscono con un prodotto tensore, quindi ci si avvale della funzione **TensorMat** già definita nel pacchetto Hom.

```
Define HomResMap(P,Q,I);
 Phi:=ResMap(P,I);
 N:=Untagged(Q);
 QGens:=Gens(Q);
 QDim:=Len(QGens[1]);
 MatPhi:=Mat(Phi);
 MatPhiStar:=TensorMat(MatPhi,Identity(QDim));
 PhiStar:=Transposed(MatPhiStar); -- trasposta
   Return [Vector(PhiStar[J])|J In 1..Len(PhiStar)];
EndDefine;
```

Ci servirà anche la funzione **ZeroModule** che prende in input un modulo e ritorna il modulo nullo della stessa dimensione del modulo di partenza, cioè se **M** è un sottomodulo di P^s, ZeroModule(**M**) è il sottomodulo nullo di P^s.

```
Define ZeroModule(M);
 G:=Gens(M);
 V:=G[1];
 V:=0*V;
 Return Module(V);
EndDefine;
```

Ora dobbiamo creare una funzione che fornisca il Ker delle mappe del modulo di coomologia, cioè i moduli K_i definiti nel passo 10). Per fare questo ci avvarremo delle funzioni del pachetto Hom, **PresentHom** che calcola una presentazione per i moduli Hom come indicato nel punto 8), e la funzione **Ker** che fornisce il nucleo di una mappa esibendone i generatori.

```
Define CoCicles(RES,Q,Phi,I);
 Ranks:=[BettiExt(RES,I),BettiExt(RES,I+1)];
 Arguments:=PresentHom(CurrentRing()^(Ranks[1]),Q);
 Values:=PresentHom(CurrentRing()^(Ranks[2]),Q);
  Return Ker(Arguments,Values,Phi);
EndDefine;
```

Con la funzione precedente siamo in grado quindi di calcolare i moduli K_i definiti nel punto 10) dell'algoritmo, mentre gli I_i sono le immagini delle mappe del complesso di coomologia e quindi risultano semplicemente dall'applicazione della funzione **HomResMap**. Dobbiamo perciò introdurre una funzione **PresentQuotient** che esegua il passo 11) avvalendosi del lemma 4.1.2. Dati in input due moduli **M** ed **N** tale funzione calcola una presentazione di $\mathbf{M}/\mathbf{M}\cap\mathbf{N}$.

```
Define PresentQuotient(M,N);
 G:=Gens(M);
 S:=Len(G);
 H:=Gens(N);
 W:=Gens(Syz(Concat(G,H)));
 Den:=Module([First(W[J],S)|J In 1..Len(W)]);
 Den:=Module(Interreduced(Gens(Den)));
 If Den=CurrentRing()^S Then
   Return ZeroModule(Den);
 Elsif Den=Module([0]) Then
 Den:=ZeroModule(CurrentRing()^S);
   Return CurrentRing()^S/Den;
 EndIf;
 Return CurrentRing()^S/Den;
EndDefine;
```

Finalmente possiamo definire la funzione **NoShiftExt** che prende in input due moduli **P** e **Q**, nella forma P^r/M e P^s/N coerentemente con la teoria presentata nei capitoli precedenti, e un indice **I**, e calcola il modulo $\mathrm{Ext}^{\mathrm{I}}(P,Q)$. Il nome di questa funzione è dovuto al fatto che non stiamo simulando la presenza di shift nel modulo P^r/M, cosa che sarà invece gestita a parte dalle funzioni successive. **NoShiftExt** controlla dapprima se si ricade in uno dei casi banali indicati dal punto 12) dell'algoritmo, calcola quindi una risoluzione del modulo **P**, le mappe necessarie con la funzione **HomResMap** e i moduli K_i e I_i utilizzando la funzione **CoCicles**. Infine presenta il quoziente K_i/I_i utilizzando **PresentQuotient**.

```
Define NoShiftExt(P,Q,I);
 If I=0 Then
   Return PresentHom(P,Q);
 Elsif I<0 Then
   Error('Index must be positive');
 EndIf;
 M:=Untagged(P);
 RES:=Res(M);
 If I>Len(RES[2]) Then
```

```
    Return Module([0]);
  EndIf;
 PhiUp:=HomResMap(RES,Q,I+1);
 PhiDown:=HomResMap(RES,Q,I);
  If I=Len(RES[2]) Then
    NumExt:=PresentHom(CurrentRing()^(LastBetti(RES)),Q);
    Dim:=NumComps(NumExt);
    DenExt:=Module(PhiDown)+Untagged(NumExt);
    DenExt:=Module(Interreduced(Gens(DenExt)));
    Return (CurrentRing()^(Dim))/DenExt;
  EndIf;
 NumExt:=CoCicles(RES,Q,PhiUp,I); -- Cocicli della coomologia
  If NumExt=Module([0]) Then
    Return NumExt;
  EndIf;
 DenExt:=Module(PhiDown); --Cobordi della coomologia
    Return PresentQuotient(NumExt,DenExt);
 EndDefine;
```

Nel prossimo paragrafo tratteremo la gestione degli shift e definiremo le funzioni necessarie.

4.3 Gestione degli shift

Fino ad ora abbiamo sempre supposto che la graduazione sui moduli liberi P^r e sui quozienti P^r/M fosse quella indotta naturalmente dalla graduazione di P. Ad esempio, se P risulta graduato standard, il vettore (x^2, y^2) di P^2 è omogeneo di grado 2, mentre (x, y^2) non risulta essere omogeneo. Risulterebbe omogeneo se fosse considerato come vettore di $P(-1) \bigoplus P$, in tal modo infatti i gradi delle prime componenti vengono shiftati di 1, e il vettore considerato risulta omogeneo di grado 2. Vogliamo quindi provare ad applicare il nostro algoritmo anche nel caso in cui il modulo omogeneo di cui si calcola una risoluzione si presenti nella forma:

$$\bigoplus_{j=1}^{r} P(-\gamma_j)/M, \quad M = \langle g_1, \ldots, g_\alpha \rangle, \quad \gamma_j \in \mathbb{N} \; \forall j = 1, \ldots, r \tag{1}$$

La versione di CoCoA in nostro possesso non prevede che l'input possa essere della forma (1), quindi dovremo ricorrere ad un isomorfismo con un opportuno modulo in cui non siano presenti gli shift. Di fatto, basta moltiplicare ciascun generatore del modulo M, componente per componente, con delle opportune potenze di una nuova indeterminata x_{n+1}, in modo da omogeneizzare i gradi rispetto alla graduazione indotta da quella di $P[x_{n+1}]$. Consideriamo ad esempio il generatore

$$g_1 = (g_{11}, \ldots, g_{1r}), \quad con \quad g_{1k} \in P \quad \forall k = 1, \ldots, r$$

Definiamo allora l'omogeneizzato:

$$\widetilde{g_1} := (x_{n+1}^{\gamma_1} g_{11}, \ldots, x_{n+1}^{\gamma_r} g_{1r}) \in (x_{n+1} P)^r$$

In tal modo il vettore $\widetilde{g_1}$ risulta essere omogeneo in $(x_{n+1}P)^r$, con la graduazione naturale priva degli shift. Quello che abbiamo costruito è in pratica un isomorfismo omogeneo (assegnando grado 1 a x_{n+1}):

$$\begin{array}{ccc} \bigoplus_{j=1}^{r} P(-\gamma_j)/\langle g_1, \ldots, g_\alpha \rangle & \longrightarrow & \bigoplus_{j=1}^{r} (x_{n+1}^{\gamma_j} P)/\langle \widetilde{g_1}, \ldots, \widetilde{g_\alpha} \rangle \\ \overline{e_j} & \longmapsto & x_{n+1}^{\gamma_j} \overline{e_j} \end{array} \tag{2}$$

Con questo isomorfismo siamo in grado di immettere nelle funzioni di CoCoA un input corretto, senza perdere alcuna informazione sugli shift. Basterà prima trasformare il modulo aggiungendo una nuova indeterminata, fare i calcoli esattamente come nel caso omogeneo del paragrafo precedente e poi ritornare indietro con l'isomorfismo ponendo $x_{n+1} = 1$. Vediamo come, commentando le funzioni del pacchetto.

La funzione **Ext** è quella principale, e permette di distinguere i casi in cui i moduli prevedano gli shift o meno. Possiamo prevedere per tale funzione

3, 4 o 5 argomenti. Nel primo caso si tratterà dei due moduli **P** e **Q** e dell'indice **I**. In questo caso la funzione si avvale di **NoShiftExt** per calcolare $\mathrm{Ext}^{\mathbf{I}}(\mathbf{P}, \mathbf{Q})$. Negli altri casi invece si potranno inserire gli shift, sotto forma di liste di interi positivi **V** che rappresentino gli shift $(\gamma_1, \ldots, \gamma_r)$. La funzione allora lavorerà in un nuovo anello con una o due indeterminate in più, costruendo l'isomorfismo (2) con la funzione **AddShifts**. Una volta costruiti i nuovi moduli verrà utilizzata ancora la funzione **NoShiftExt** per il calcolo di Ext e quindi si tornerà indietro con l'isomorfismo utilizzando la funzione **RemoveShifts**.

```
Define Ext(...);
If Len(ARGV)<3 Then
  Error('Expected arguments: 2 modules and 1 integer');
Elsif Len(ARGV)=3 Then
  Return NoShiftExt(ARGV[1],ARGV[2],ARGV[3]);
Elsif Len(ARGV)>5 Then
  Error('Too many arguments');
Elsif Len(ARGV)=4 Then
  P:=ARGV[1];
  Q:=ARGV[2];
  I:=ARGV[3];
  SHIFT1:=ARGV[4];
  SHIFT2:=NewList(NumComps(Q),0);
Else
  P:=ARGV[1];
  Q:=ARGV[2];
  I:=ARGV[3];
  SHIFT1:=ARGV[4];
  SHIFT2:=ARGV[5];
EndIf;
N:=NumIndets();
ExtRing::=CoeffRing[x[1..(N+2)]];
Using ExtRing Do
-- simulazione di shift dei moduli gestita a parte
    ShiftForP:=[x[N+1]^J|J In SHIFT1];
    ShiftedP:=AddShifts(P,ShiftForP,N);
    ShiftForQ:=[x[N+2]^J|J In SHIFT2];
    ShiftedQ:=AddShifts(Q,ShiftForQ,N);
-- ora posso calcolare l'Ext
   EXT:=NoShiftExt(ShiftedP,ShiftedQ,I);
EndUsing;
EXT:=RemoveShifts(EXT);
DimExt:=NumComps(EXT);
 If (@EXT)=(CurrentRing())^(DimExt) Then
```

```
    Return ZeroModule(@EXT);
  EndIf;
    Return EXT;
EndDefine;
```

Per poter realizzare l'isomorfismo (2) usiamo la funzione **PunctualProduct** che moltiplica un generatore **W** per il vettore di shift **V**, componente per componente.

```
Define PunctualProduct(V,W);
  Return [V[J]*W[J]|J In 1..Len(V)];
EndDefine;
```

La funzione **Ext** usa le due seguenti funzioni per simulare lo shift sui moduli in gioco. Si noti che nelle ipotesi dell'algoritmo 4.1.3 era richiesta l'omogeneità soltanto del modulo P^r/M, quindi non è necessario rendere omogeneo il secondo modulo P^s/N, infatti la funzione **AddShifts** può agire anche solo sul primo modulo, nel caso in cui vengano immessi solo quattro argomenti nella funzione **Ext**. In tal caso il vettore di shift per **Q** definito nella funzione **Ext** corrisponde infatti agli shift nulli.

```
Define AddShifts(P,V,N);  -- N sono le indeterminate
 M := @P;
 If Len(V)<>NumComps(P) Then
   Error('Bad Shifts Vector');
 EndIf;
 M:=Image(M,RMap(x[1]..x[N]));
 G:=Gens(M);
 NewGens:=[PunctualProduct(V,W)|W In G];
 M:=Module(NewGens);
 Return CurrentRing()^(NumComps(M))/M
EndDefine;
```

La funzione **RemoveShifts** invece ritorna nell'anello di partenza con n indeterminate ponendo $x_{n+1} = x_{n+2} = 1$.

```
Define RemoveShifts(EXT);
  Return Image(EXT,RMap(Concat(Indets(),[1,1])));
EndDefine;
```

Osservazione 4.3.1. Le funzioni che abbiamo presentato in questo paragrafo hanno permesso di poter simulare una graduazione, in input, diversa da quella naturalmente indotta da P sui moduli tipo P^r, tuttavia anche una volta costruito l'isomorfismo (2), durante il processo di calcolo di Ext

l'informazione sugli shift non viene mantenuta, in quanto le presentazioni e gli isomorfismi utilizzati nel pacchetto per il calcolo di Hom e di Ext non rispettano in generale la struttura graduata, ma forniscono semplicemente isomorfismi algebrici. Dunque l'output della funzione **Ext**, anche nel caso di simulazione degli shift, è soltanto un modulo algebricamente isomorfo al modulo Ext che vogliamo conoscere, cioè in generale l'isomorfismo non è di grado zero.

Abbiamo dunque terminato la presentazione delle funzioni che permettono di simulare la presenza degli shift, concludiamo il capitolo con una funzione che permette di calcolare tutti gli $\mathrm{Ext}^{\mathrm{i}}(P^r/M, P^s/N)$ che potrebbero non essere nulli, cioè quelli per $i = 0, \ldots, m$ dove m è la lunghezza della risoluzione calcolata da CoCoA . Il Teorema delle Sizigie di Hilbert assicura come sappiamo che esiste una risoluzione libera lunga al più n, e quindi le coomologie dalla $n+1$-esima in poi sono sicuramente nulle, come del resto abbiamo ribadito nel teorema 4.1.3 al punto 12). I primi due argomenti della funzione **EveryExt** devono essere i due moduli, e gli eventuali altri due argomenti devono contenere le liste degli shift.

```
Define EveryExt(...);
N:=NumIndets();
If Len(ARGV)<2 And Len(ARGV)>4 Then
  Error('Bad Number of Parameters');
EndIf;
M:=Untagged(ARGV[1]);
If IsHomog(M) Then
  RES:=Res(M);
  MaxIndex:=Len(RES[2]);
Else
  MaxIndex:=NumIndets();
EndIf;
PrintLn(NewLine);
If Len(ARGV)=2 Then
 For I:=0 To MaxIndex Do
   PrintLn('Ext^',I,' =');
   Ext(ARGV[1],ARGV[2],I);
   PrintLn(NewLine);
 EndFor;
Elsif Len(ARGV)=3 Then
 For I:=0 To MaxIndex Do
   PrintLn('Ext^',I,' =');
   Ext(ARGV[1],ARGV[2],I,ARGV[3]);
   PrintLn(NewLine);
 EndFor;
Else
```

```
 For I:=0 To MaxIndex Do
   PrintLn('Ext^',I,' =');
   Ext(ARGV[1],ARGV[2],I,ARGV[3],ARGV[4]);
   PrintLn(NewLine);
 EndFor;
EndIf;
EndDefine;
```

5. Esempi e applicazioni

Presenteremo ora alcuni esempi di calcolo di Ext elaborati con CoCoA. Prima di fare i calcoli, però, è necessario riflettere sul corpo di coefficienti in cui stiamo lavorando, dato che utilizziamo il calcolo simbolico attraverso il computer. Nel primo paragrafo affronteremo e risolveremo questo problema grazie ad un teorema di unicità del corpo di definizione che si avvale ancora una volta della teoria delle Basi di Gröbner . Nei paragrafi successivi invece elencheremo alcuni esempi sia di tipo puramente algebrico, sia legati agli operatori differenziali come quelli presentati nel secondo capitolo.

5.1 Corpo dei coefficienti

Prima di presentare gli esempi di applicazione delle funzioni definite nel capitolo precedente è necessaria un'osservazione. Sappiamo infatti che gli anelli con cui possiamo lavorare su CoCoA sono soltanto il corpo dei numeri razionali $\mathbb{Q}$ e gli anelli del tipo $\mathbb{Z}_p$ con p primo, mentre noi vorremmo poter dedurre delle informazioni su P-moduli finitamente generati dove $P = K[x_1, \ldots, x_n]$. Ad esempio, riferendosi ai casi introdotti nel capitolo 2, i moduli M_n sono $\mathbb{C}[x_0, \ldots, x_{4n-1}]$-moduli, ma il corpo dei coefficienti $\mathbb{C}$ non è computabile. Tuttavia i generatori dei moduli M_n sono tutti vettori con coefficienti in $\mathbb{Q}$, dunque è possibile rappresentarli su CoCoA. Dobbiamo però essere sicuri che i moduli prodotti in output siano effettivamente quelli voluti, cioè che forniscano una effettiva rappresentazione di Ext come $\mathbb{C}[x_0, \ldots, x_{4n-1}]$-modulo e non solo come $\mathbb{Q}[x_0, \ldots, x_{4n-1}]$-modulo. Il problema è risolto dalla seguente proposizione, per la cui dimostrazione rimandiamo a [KR]:

Proposizione 5.1.1. *Sia $K' \subseteq K$ un'estensione di corpi, siano $P' = K'[x_1, \ldots, x_n]$ e sia $M' \subseteq (P')^r$ un P'-sottomodulo finitamente generato. Sia M il P-sottomodulo di P^r generato dagli elementi di M'. Allora una* σ-Base di Gröbner *per M' è anche una* σ-Base di Gröbner *per M.*

In tal modo, grazie a questo risultato, ogni volta che calcoliamo una Base di Gröbner per un $\mathbb{Q}[x_0, \ldots, x_{4n-1}]$-modulo, in realtà stiamo anche calcolando una Base di Gröbner dello stesso inteso come $\mathbb{C}[x_0, \ldots, x_{4n-1}]$-modulo. Siccome di fatto i nostri algoritmi per Hom ed Ext utilizzano più volte il calcolo di moduli di sizigie, il risultato che otteniamo non dipende dal corpo dei coefficienti ma vale per una qualunque sua estensione.

Vediamo ora alcuni esempi di tipo puramente algebrico in cui vogliamo calcolare la profondità di $P = \mathbb{Q}[x, y, z]$ rispetto ad un ideale I. Questo sarà possibile utilizzando il corollario 3.3.6, in base al quale la $\mathrm{depth}_I(M)$ è l'indice del primo $\mathrm{Ext}^{\mathrm{i}}(P/I, M)$ che non si annulla.

5.2 Esempi algebrici

Tutti gli esempi di questo paragrafo e del successivo sono stati elaborati con la versione 4.1 di CoCoA per sistema operativo Linux.

Esempio 5.2.1. Consideriamo l'anello $P = \mathbb{Q}[x, y, z]$ e il suo ideale massimale $I = (x, y, z)$. Ovviamente P ha profondità 3 come P-modulo, essendo (x, y, z) una successione regolare in I. Verifichiamo dunque che il primo Ext che non si annulla è il terzo:

```
Use R::=Q[x,y,z];
------------------------------
I:=Ideal(x,y,z);
I:=Module(I);
P:=R^1;

EveryExt(R^1/I,P);

Ext^0 =
Module([0])

Ext^1 =
Module([0])

Ext^2 =
Module([0, 0, 0])

Ext^3 =
R^1/Module([-x], [y], [-z])
------------------------------
```

Se invece avessimo voluto calcolarne la profondità in $J = (x, y)$ avremmo ottenuto 2:

```
J:=Ideal(x,y);
J:=Module(J);
P:=R^1;

EveryExt(R^1/J,P);

Ext^0 =
Module([0])

Ext^1 =
Module([0])
```

```
Ext^2 =
R^1/Module([y], [-x])

------------------------------
```

Esempio 5.2.2. Vediamo un esempio leggermente più complicato in cui i due moduli sono del tipo P/I e P/J con $I = (x^2, 2xz + 3yz)$ e $J = (2y^2 + 3x, -5x^3 + 2z)$.

```
Use R::=Q[xyz];

M:=R^1/Module([x^2], [2xz + 3yz]);
N:=R^1/Module([2y^2 + 3x], [-5x^3 + 2z]);


EveryExt(M,N);
Ext^0 =
Module([0, 0])

Ext^1 =
R^5/Module([-4/45y, -2/15, -2/9x, -4/45, 0],
           [0, 0, -4/9z, -9/4y, -x + 3/2y],
           [-4/45x, 4/45y, 0, 0, -4/45],
           [0, 0, 0, -9/4x - 27/8y, 9/4y])

Ext^2 =
R^1/Module([2/9x^2], [y^2 + 3/2x], [-2/5z])


------------------------------
```

Abbiamo potuto calcolare gli Ext anche se il secondo modulo non era omogeneo (rispetto alla graduazione standard che su CoCoA è di default). Non vi è comunque modo di renderlo omogeneo utilizzando opportuni shift, dato che i polinomi già si presentano in forma non omogenea e abbiamo una sola componente a disposizione. Possiamo eventualmente shiftare la graduazione sul primo modulo, ma il risultato ottenuto è ovviamente isomorfo come mostra CoCoA:

```
Ext(M,N,1,[1]);
R^5/Module([-4/45y, -2/15, -2/9x, -4/45, 0],
           [-4/45x, 4/45y, 0, 0, -4/45],
           [0, 0, 0, -4/9x - 2/3y, 4/9y],
[0, 0, -4/9z, -4/9x^2 + y^2 + 3/2x, 4/9xy - 2/3y^2 - x])
------------------------------
```

```
Ext(M,N,1)=Ext(M,N,1,[1]);
TRUE
--------------------------------
```

Dunque nel caso di sottomoduli di P, cioè ideali, lo shift della graduazione non produce alcun effetto. Nel prossimo paragrafo vedremo invece alcuni esempi legati allo studio degli operatori differenziali, e in uno di questi sarà necessario utilizzare gli shift per poter rendere omogeneo il modulo di cui serve la risoluzione.

5.3 Esempi legati agli Operatori Differenziali

Esempio 5.3.1. Consideriamo il sistema di equazioni differenziali dato dalle condizioni di **Cauchy-Riemann** per una funzione di variabile complessa:

$$\begin{cases} \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} &= 0 \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} &= 0 \end{cases}$$

La matrice A che fornisce l'operatore di derivazione associato è

$$A = \begin{pmatrix} x & -y \\ y & x \end{pmatrix}$$

dunque il conucleo della sua trasposta è il modulo

$$M = \mathbb{Q}[x,y]^2/\langle (x,-y),(y,x)\rangle$$

In base alla teoria vista nel capitolo 2, calcoliamo il modulo $\mathrm{Ext}^1(M,P)$ dove $P = \mathbb{Q}[x,y]$ e verifichiamo che NON vale il Fenomeno di Hartog del teorema 2.3.1. Ecco come fare con CoCoA.

```
Use R::=Q[x,y];

M:=R^2/Module([y,x],[x,-y]);
-------------------------------
P:=R^1;
-------------------------------
EveryExt(M,P);

Ext^0 =
Module([0])

Ext^1 =
R^2/Module([y, x], [x, -y])


-------------------------------
```

Coerentemente con quello che ci si aspettava, Ext^1 non si annulla.

Esempio 5.3.2. Verifichiamo la stessa cosa per le funzioni regolari su $\mathbb{H}$, nucleo dell'operatore di **Cauchy-Fueter** che ricordiamo è associato alla matrice

$$A = \begin{pmatrix} x_0 & -x_1 & -x_2 & -x_3 \\ x_1 & x_0 & -x_3 & x_2 \\ x_2 & x_3 & x_0 & -x_1 \\ x_3 & -x_2 & x_1 & x_0 \end{pmatrix}$$

Facciamo calcolare gli $\mathrm{Ext}^i(M_1, P)$ a CoCoA, dove M_1 è il conucleo della trasposta di A.

```
Use R::=Q[x[0..3]];

M1:=R^4/Module([x[0],-x[1],-x[2],-x[3]],
   [x[1],x[0],-x[3],x[2]],
   [x[2],x[3],x[0],-x[1]],
   [x[3],-x[2],x[1],x[0]]);

P:=R^1;

EveryExt(M1,P);

Ext^0 =
Module([0])

Ext^1 =
R^4/Module([x[0], x[1], x[2], x[3]],
                [-x[1], x[0], x[3], -x[2]],
                [-x[2], -x[3], x[0], x[1]],
                [-x[3], x[2], -x[1], x[0]])

-------------------------------
```

Questi due esempi suggeriscono che nel caso in cui le matrici associate agli operatori siano quadrate, ci sia la possibilità che il primo Ext non si annulli, cosa che è riflessa dal fatto che non ci sono sizigie per i generatori del modulo che costituisce il denominatore di M_n, cioè non ci sono sizigie per le righe di A. Enunciamo e dimostriamo quindi una proposizione che generalizza questo caso. Le notazioni sono le stesse del secondo capitolo.

Proposizione 5.3.3. *Sia n un intero positivo, $P = \mathbb{C}[x_1, \ldots, x_n]$ l'anello dei polinomi, sia $A \in \mathrm{Mat}_{n\times n}(P)$ una matrice quadrata di ordine n, ad elementi omogenei di grado 1, tale che $\mathrm{Det}(A) \neq 0$. Allora per la classe di funzioni $\mathcal{S}^A$, nucleo dell'operatore associato $A(D)$, non vale il Fenomeno di Hartog.*

Dimostrazione. Posto $M = \mathrm{Coker}\,({}^tA)$, dobbiamo verificare che con tali ipotesi il modulo $\mathrm{Ext}^1(M,P)$ non si annulla. Costruiamo dunque una risoluzione minimale graduata per M, modulo che si presenta nella forma

$$M \cong P^n/\langle r_1,\dots,r_n\rangle$$

dove $r_1,\dots,r_n$ sono i generatori di $\mathrm{Im}\,({}^tA)$, cioè le righe di A. Mostriamo che non vi sono sizigie per tali vettori. Sia infatti $s = (s_1,\dots,s_n)$ una sizigia di $(r_1,\dots,r_n)$, allora utilizzando una notazione matriciale, vale l'equazione

$${}^tA \cdot \underline{s} = \underline{0}$$

dove al solito $\underline{s}$ indica il vettore colonna. La matrice A ha determinante diverso da zero quindi risulta $s = (0,\dots,0)$. Dunque, in base all'algoritmo fornito dalla dimostrazione della proposizione 3.1.17, abbiamo la seguente risoluzione libera del tipo $\mathbb{L} \xrightarrow{\pi} M \longrightarrow 0$:

$$0 \xrightarrow{d_2} P^n \xrightarrow{d_1} P^n \xrightarrow{\pi} M \longrightarrow 0$$

dove la mappa d_1 è definita da $d_1(e_j) = r_j$, $i = 1,\dots,n$, mentre la mappa d_2 è la mappa nulla perché non vi sono sizigie non banali. Dualizziamo $\mathbb{L}$ applicando il funtore $\mathrm{Hom}(\cdot,P)$:

$$0 \longrightarrow P^n \xrightarrow{{}^td_1} P^n \xrightarrow{{}^td_2} 0$$

Quello che dobbiamo valutare è il quoziente $E_1 = \frac{\mathrm{Ker}\,({}^td_2)}{\mathrm{Im}\,({}^td_1)}$. Il nucleo di ${}^td_2$ è tutto P^n perché d_2 è la mappa nulla, mentre $\mathrm{Im}\,({}^td_1)$ è generato dalle colonne $c_1,\dots,c_n$ di A. Quindi risulta

$$\mathrm{Ext}^1(M,P) = P^n/\langle c_1,\dots,c_n\rangle \neq 0$$

□

Esempio 5.3.4. Supponiamo invece di voler verificare le proprietà delle funzioni regolari di due varibili quaternioniche. Lavoreremo nell'anello di polinomi $P = \mathbb{Q}[x_0,\dots,x_3,y_0,\dots,y_3]$. La matrice associata all'operatore di Cauchy-Fueter in due variabili $q_1 = x_0 + x_1i + x_2j + x_3k$ e $q_2 = y_0 + y_1i + y_2j + y_3k$ è la seguente:

$$A = \begin{pmatrix} x_0 & -x_1 & -x_2 & -x_3 \\ x_1 & x_0 & -x_3 & x_2 \\ x_2 & x_3 & x_0 & -x_1 \\ x_3 & -x_2 & x_1 & x_0 \\ y_0 & -y_1 & -y_2 & -y_3 \\ y_1 & y_0 & -y_3 & y_2 \\ y_2 & y_3 & y_0 & -y_1 \\ y_3 & -y_2 & y_1 & y_0 \end{pmatrix}$$

In questo caso il modulo M_2 che devo considerare è il conucleo di ${}^tA$ quindi è il quoziente di P^4 con il modulo generato dalle righe di A.

```
Use R::=Q[x[0..3],y[0..3]];

M2:=R^4/Module([x[0],-x[1],-x[2],-x[3]],
   [x[1],x[0],-x[3],x[2]],
   [x[2],x[3],x[0],-x[1]],
   [x[3],-x[2],x[1],x[0]],
   [y[0],-y[1],-y[2],-y[3]],
   [y[1],y[0],-y[3],y[2]],
   [y[2],y[3],y[0],-y[1]],
   [y[3],-y[2],y[1],y[0]]);

P:=R^1;

EveryExt(M2,P);

Ext^0 =
Module([0])

Ext^1 =
Module([0, 0, 0, 0])

Ext^2 =
Module([0, 0, 0, 0, 0, 0, 0, 0])

Ext^3 =
R^4/Module([-2x[0], 2x[1], 2x[2], 2x[3]],
         [-2x[1], -2x[0], 2x[3], -2x[2]],
           [2x[2], 2x[3], 2x[0], -2x[1]],
           [2x[3], -2x[2], 2x[1], 2x[0]],
         [2y[0], -2y[1], -2y[2], -2y[3]],
           [2y[1], 2y[0], -2y[3], 2y[2]],
         [-2y[2], -2y[3], -2y[0], 2y[1]],
         [-2y[3], 2y[2], -2y[1], -2y[0]])
-------------------------------
```

Con riferimento all proposizione 2.3.5 deduciamo che la varietà caratteristica associata all'operatore $A(D)$ ha dimensione $4n - p - 1 = 7 - p$, dove p è l'indice dell'ultimo Ext nullo, cioè $p = 2$. La varietà caratteristica ha dunque dimensione $5 = 2n + 1$, in accordo con i risultati di Palamodov, in particolare si veda l'osservazione 2.3.8.

Esempio 5.3.5. Vediamo ancora un esempio di operatore differenziale su funzioni di variabili reali. In questo caso si tratta di classi di funzioni $\underline{f} \in C^\infty(\mathbb{R}^3)$ che soddisfano il seguente sistema di equazioni differenziali:

$$\begin{cases} \frac{\partial f_1}{\partial x} + \frac{\partial f_2}{\partial z} &= 0 \\ \frac{\partial f_1}{\partial y} + \frac{\partial f_3}{\partial z} &= 0 \\ -\frac{\partial f_2}{\partial y} + \frac{\partial f_3}{\partial x} &= 0 \end{cases} \tag{1}$$

L'operatore di moltiplicazione associato è quindi rappresentato dalla matrice

$$A_1 = \begin{pmatrix} x & z & 0 \\ y & 0 & z \\ 0 & -y & x \end{pmatrix}$$

CoCoA permette di verificare che anche per questa classe di funzioni vale il fenomeno di Hartog. In effetti la matrice A ha determinante nullo e quindi non rientria nelle ipotesi della proposizione 5.3.3.

```
Use R::=Q[xyz];

M1:=R^3/Module([x,z,0],[y,0,z],[0,-y,x]);
P:=R^1;

EveryExt(M1,P);

Ext^0 =
R^1/Module([0])

Ext^1 =
Module([0, 0, 0])

Ext^2 =
R^1/Module([y], [-x], [z])

-------------------------------
```

Esempio 5.3.6. Il sistema di equazioni può anche prevedere derivate seconde:

$$\begin{cases} \frac{\partial^2 f_1}{x^2} + \frac{\partial f_2}{\partial z} &= 0 \\ \frac{\partial^2 f_1}{y^2} + \frac{\partial f_3}{\partial z} &= 0 \\ -\frac{\partial f_2}{\partial y} + \frac{\partial f_3}{\partial x} &= 0 \end{cases} \tag{2}$$

anche in tal caso la matrice associata ha come elementi polinomi omogenei:

$$A_2 = \begin{pmatrix} x^2 & z & 0 \\ y^2 & 0 & z \\ 0 & -y & x \end{pmatrix}$$

Per poter utilizzare CoCoA ricordiamo che il conucleo deve essere opportunamente modificato shiftando la graduazione, alzando di 1 i gradi delle ultime due componenti.

```
Use R::=Q[xyz];

M2:=R^3/Module([x^2,z,0],[y^2,0,z],[0,-y,x]);
P:=R^1;

V:=[0,1,1];  -- vettore di shift

EveryExt(M2,P,V);

Ext^0 =
Module([0, 0, 0])


Ext^1 =
R^3/Module([x^2, y^2, 0], [z, 0, -y], [0, z, x])

Ext^2 =
Module([0])

Ext^3 =
Module([0])

------------------------------
```

Per la classe di funzioni soluzione del sistema (2) dunque non si può dedurre nessun fenomeno di regolarità come quelli studiati nel secondo capitolo.

Bibliografia

[ABL] W.W. Adams, C.A. Berenstein, P. Loustanau, I. Sabadini,D.C. Struppa: Regular functions of several quaternionic variables and the Cauchy-Fueter complex. J.Geom.Anal. (n. 1, pp. 1-15), 1999.

[AM] M.F. Atiyah, I.G.Macdonald: Introduction to Commutative Algebra. Addison-Wesley, Reading, Mass. , 1969.

[DS] D.C. Struppa: Grobner Bases in partial differential equation. London Math. Soc. Lecture Note Ser. 251 (Cambridge University Press), Cambridge, 1998.

[FU1] R. Fueter: Die Funktionetheorie der Differentialgleichungen $\Delta = 0$ und $\Delta^2 = 0$ mit vier reelen Variablen. Comm.Math.Helv (n. 7, pp. 307-330), 1935.

[FU2] R. Fueter: Über eine Hartogs'schen Satz. Comm.Math.Helv (n. 12, pp. 394-400), 1942.

[G] N.S.Gopalakrishnan: Commutative Algebra. Oxonian Press, PVT. LTD. New Delhi, 1984.

[KR] M. Kreuzer, L.Robbiano: Computational Commutative Algebra 1. Springer-Verlag, 2000.

[LE] L. Ehrenpreis: A New proof and an extension of Hartog's Theorem. Bull. A.M.S. (n. 67, pp. 507-509), 1961.

[SS] I. Sabadini, D.C. Struppa: Some open problems on the analysis of the Cauchy-Fueter system in several variables. (survey) Kyoto, 1997.

[T] J.Tate: Homology of local and Noetherian rings. Illiois J. Math. (n. 1 pp. 14-24), 1957.

[VPP] V.P. Palamodov: Linear Differential Operators with Constant Coefficients Springer-Verlag, New York, 1970.