Το υπολογιστικό πακέτο axiom

Άγγελος Μαντζαφλάρης, amantzaf@math.uoa.gr

16 Απριλίου 2008

Σε αυτό το σύντομο κείμενο επιχειρούμε να δώσουμε τις βασικές οδηγίες για το Axiom καθώς και τις κυριότερες εντολές του, τις οποίες θα χρειαστούμε στο μάθημα της *Υπολογιστικής Άλγεβρας*[*]. Δεν πρόκειται για πλήρεις σημειώσεις ή αναλυτική παρουσίαση του πακέτου, αλλά για μια εισαγωγή με απλά λόγια προς το χρήστη που δεν έχει χρησιμοποιήσει κάποιο ανάλογο υπολογιστικό πακέτο στο παρελθόν.

Το Axiom είναι ένα πακέτο *συμβολικών υπολογισμών*. Αυτό σημαίνει πως δε χρησιμοποιεί τη γνωστή αριθμητική κινητής υποδιαστολής της *Αριθμητικής Ανάλυσης*. Αντίθετα, έχει τη δυνατότητα να πραγματοποιεί πράξεις με σύμβολα και μεταβλητές, ακριβώς όπως θα κάνατε στο χαρτί σας, π.χ. για να αναπτύξετε την παράσταση $(a+b)^2$. Επίσης μπορεί να χειρίζεται κλάσματα και ριζικά, και γενικότερα *αλγεβρικούς αριθμούς*, δηλαδή αριθμούς οι οποίοι μπορούν να παρασταθούν ως ρίζες κάποιου πολυωνύμου με ακέραιους συντελεστές. Με την έννοια αυτή, οι υπολογισμοί στο Axiom χαρακτηρίζονται από άπειρη ακρίβεια! Βέβαια υπάρχουν αρκετοί περιορισμοί σε αυτούς τους θεωρητικά ακριβείς υπολογισμούς, και θα πρέπει να είμαστε πάντα επιφυλακτικοί απέναντι σε όσα μας επιστρέφει ο υπολογιστής. Τέλος, όταν το ζητήσουμε με τις κατάλληλες εντολές, το Axiom μπορεί να υπολογίσει από μια συμβολική παράσταση μια αριθμητική προσέγγιση, στην ακρίβεια που ορίζουμε.

1 Εγκατάσταση στα Windows

Η εγκατάσταση είναι εξαιρετικά απλή: Απλώς κατεβάστε και τρέξτε το αρχείο(48MB)

`http://axiom-wiki.newsynthesis.org/public/axiom-windows-0.1.4.exe`

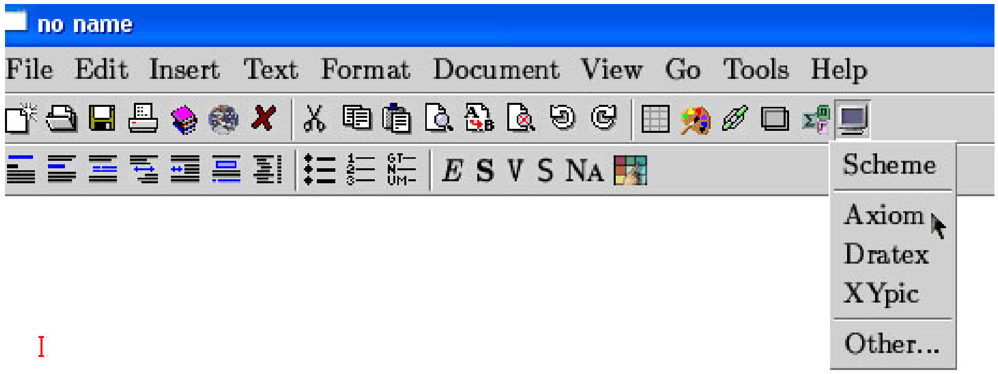
Αφού εγκατασταθεί το Axiom μπορείτε να το εκκινήσετε από το εικονίδιο που δημιουργείται στην επιφάνεια εργασίας. Με τον τρόπο αυτό θα σας ανοίξει ένα παράθυρο του DOS, για να δώσετε εντολές στο Axiom.

Στο περιβάλλον του DOS η εμφάνιση μαθηματικών τύπων δεν είναι καθόλου κομψή. Για το λόγο αυτό προτείνεται να κατεβάσετε και το πρόγραμμα WinTeXmacs από τη διεύθυνση(12MB)

`http://www.texmacs.org/Download/ftp/windows/stable/wintexmacs-1.0.5.exe`

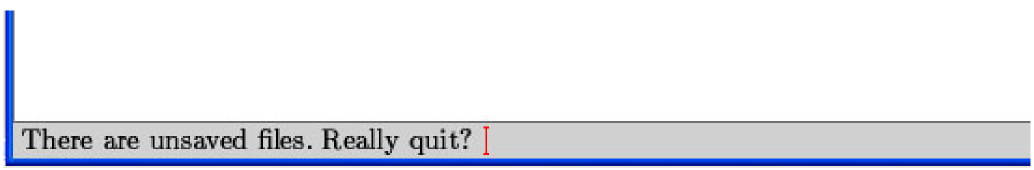
για να έχετε ένα πολύ καλύτερο περιβάλλον εργασίας για το Axiom. Αφού το εγκαταστήσετε κι αυτό, μπορείτε να το τρέξετε (κατά την πρώτη εκκίνηση πατήστε «όχι» στο παράθυρο που αναδύεται) και να επιλέξετε το Axiom όπως φαίνεται παρακάτω(κάνετε κλικ στο εικονίδιο-οθόνη και επιλέγετε Axiom):

[*]Οι σημειώσεις για την ώρα είναι πρόχειρες και ατελείς! Περιμένω τα σχόλιά σας στην *κουβέντα* για προσθήκες - διορθώσεις - βελτιώσεις.



Κατόπιν γράφετε τις εντολές σας στο ευχάριστο αυτό περιβάλλον (το οποίο χρησιμοποιούμε και στα παρακάτω). Με τον τρόπο αυτό το Axiom εκκινείται μέσω του WinTeXmacs και δε χρειαζόμαστε πλέον το εικονίδιο *Axiom* που υπάρχει στην επιφάνεια εργασίας, παρά μόνο το *WinTeXmacs*.

Σημειώνουμε ότι στο WinTeXmacs διάφορες «ερωτήσεις» που σας κάνει το πρόγραμμα εμφανίζονται, αντίθετα με όσα έχετε συνηθίσει στα windows, στη γραμμή κατάστασης του προγράμματος. Για παράδειγμα, αν προσπαθήσετε να κλείσετε το πρόγραμμα χωρίς να έχετε σώσει το έγγραφο που έχετε ανοικτό, θα εμφανιστεί στη γραμμή κατάστασης η ερώτηση:



και θα πρέπει να πατήσετε ένα από τα πλήκτρα `y`, `n` για yes και no αντίστοιχα.

Σημείωση: Ο συνδυασμός Axiom/WinTeXmacs δουλεύει και σε Windows Vista(εκτέλεση ως διαχειριστής). Όμως οι εκδόσεις αυτών των προγραμμάτων είναι σχετικά παλιές (ενός ή δυο ετών). Μια πιο πρόσφατη έκδοση του Axiom συμβατή με Windows είναι η OpenAxiom, διαθέσιμη από εδώ (19MB):

`http://downloads.sourceforge.net/open-axiom/OpenAxiom-1.1.0.exe`

Αν υπάρχει κάποιο πρόβλημα με τον συνδυασμό Axiom/WinTeXmacs προτείνεται να εγκαταστήσετε το OpenAxiom (τελευταία έκδοση 15 Φεβρουαρίου 2008). Το μειονέκτημα είναι ότι αυτό δουλεύει μόνο στο περιβάλλον του DOS, όπου η εμφάνιση μαθηματικών τύπων δεν είναι ευκρινής. Είναι πιθανό όμως να είναι ταχύτερο και να διορθώνει τυχόν σφάλματα του συνδυασμού Axiom/WinTeXmacs.

2 Βασικές εντολές

Έχοντας ξεκινήσει το Axiom μας εμφανίζεται στην οθόνη του WinTeXmacs ένα βελάκι ($\rightarrow$), πράγμα που σημαίνει πως το πρόγραμμα είναι έτοιμο και περιμένει τις εντολές μας(αν δουλεύουμε στο DOS εμφανίζονται οι χαρακτήρες `->`). Το πρώτο που μπορούμε να κάνουμε είναι να ορίσουμε μια μεταβλητή.

Η ανάθεση γίνεται δίνοντας ένα όνομα, ακολουθεί := και τέλος η τιμή που θέλουμε να δώσουμε στη μεταβλητή (φυσικά κάθε εντολή που γράφουμε, και φαίνεται με μπλε χρώμα, πρέπει να ακολουθείται από το πλήκτρο Enter):

```
→ i:=1
```

1 (1) Type: PositiveInteger

Το Axiom μας «απαντάει» με μαύρα γράμματα την τιμή που δώσαμε στη μεταβλητή. Επίσης στα δεξιά μας δείχνει με καφέ γράμματα τον τύπο του αντικειμένου που επέστρεψε. Στην περίπτωσή μας γράφει ότι μόλις του δώσαμε έναν θετικό ακέραιο! Η αρίθμηση σε παρενθέσεις είναι απλώς ένας αύξων μετρητής των εντολών που δίνουμε στο σύστημα.

Αν θέλουμε να εισάγουμε μια μεταβλητή, η οποία θα περιέχει μια συμβολοσειρά, ορίζουμε ότι ο τύπος της θα είναι String ως εξής:

```
→ s: String := "mia-seira-apo-xaraktires"
```

"mia-seira-apo-xaraktires" (2) Type: String

Το Axiom μπορεί να κάνει γρήγορα συμβολικές πράξεις. Για παράδειγμα αν θέλουμε να αναπτύξουμε το $(a+b)^2$ δίνουμε:

```
→ expand( (a+b)^2 )
```

$b^2 + 2ab + a^2$ (3) Type: Polynomial Integer

Παρατηρήστε τον τύπο που κατάλαβε το Axiom: πολυώνυμο με ακέραιους συντελεστές. Στην αμέσως επόμενη εντολή μπορούμε να χρησιμοποιήσουμε το χαρακτήρα % για να δηλώσουμε την τελευταία έξοδο που μας έδωσε το Axiom. Δηλαδή αν θέλουμε να εκτιμήσουμε την παράσταση $b^2 + 2ab + a^2$ για $a = 1$ και $b = 2$ δίνουμε:

```
→ eval(%,[a=1,b=2])
```

9 (4) Type: Polynomial Integer

Το δεύτερο όρισμα, δηλαδή το $[a = 1, b = 2]$ είναι μια *λίστα* με δυο στοιχεία.

Από μια συμβολική παράσταση μπορούμε να λάβουμε μια αριθμητική προσέγγιση μετατρέποντας την σε τύπο float. Για παράδειγμα, παρακάτω υπολογίζουμε μια αριθμητική προσέγγιση του $\sqrt[3]{2}$:

```
→ nthRoot(2,3)
```

$\sqrt[3]{2}$ (5) Type: AlgebraicNumber

```
→ %::Float
```

1.2599210498948731648 (6) Type: Float

Το :: σημαίνει ότι θέλουμε να μετατρέψουμε την παράσταση $\sqrt[3]{2}$, η οποία είναι *τύπου* αλγεβρικού αριθμού (AlgebraicNumber) σε τύπο αριθμού κινητής υποδιαστολής (Float), δηλαδή μια αριθμητική προσέγγιση.

Μπορούμε να εισάγουμε σχόλια, π.χ. για να επεξηγήσουμε τις εντολές που ακολουθούν, ακόμη και ολόκληρο κείμενο, αν ξεκινήσουμε μια γραμμή με δυο παύλες, όπως φαίνεται παρακάτω:

→ `-- Se autes tis grammes`

→ `-- pros8etoume ena mikro sxolio`

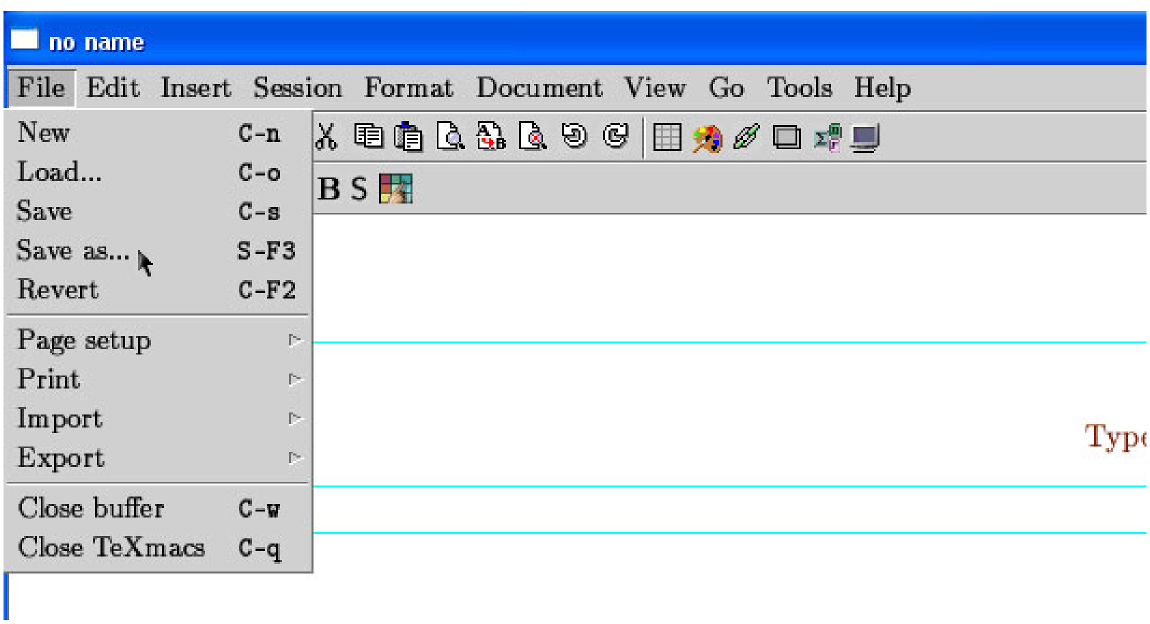
Το Axiom κρατάει στη μνήμη όλες τις μεταβλητές που έχουμε ορίσει. Αν θέλουμε να σβήσουμε τη μνήμη και να ξεκινήσουμε ξανά από την αρχή δίνουμε:

→ `)clear all`

```
All user variables and function definitions have been cleared.
```

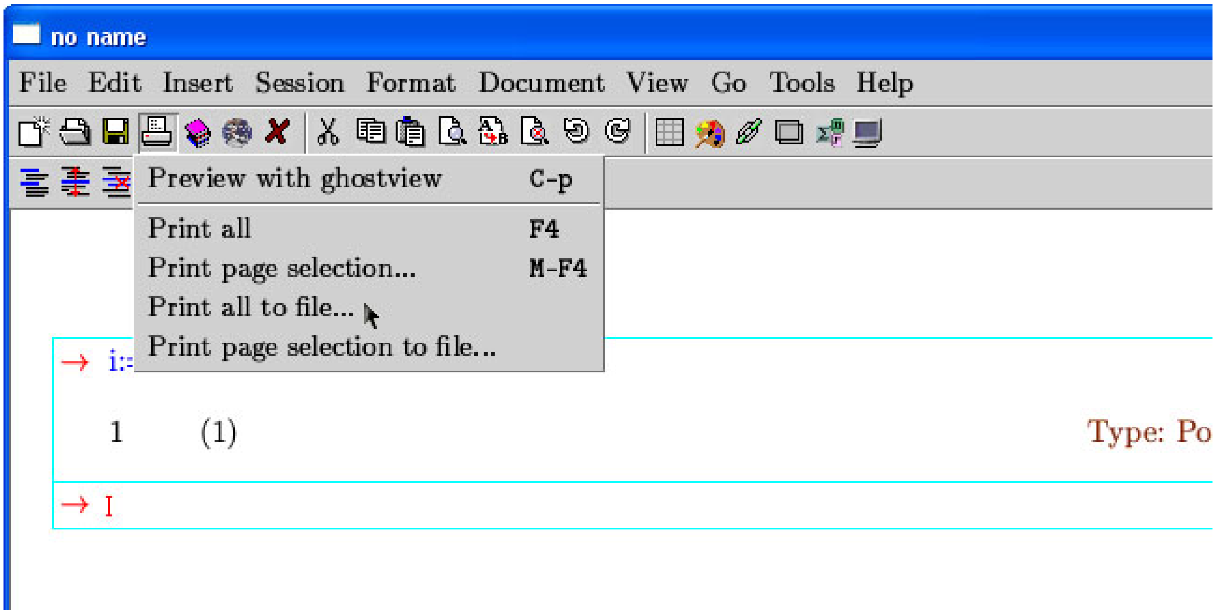
Παρατηρήστε ότι αυτή η εντολή είναι *εντολή συστήματος*, γι΄ αυτό το λόγο εισάγεται ξεκινώντας με). Μετά την εκτέλεση της εντολής η αρίθμηση των εξόδων αρχίζει και πάλι από το (1).

Μπορείτε να κρατάτε την εργασία σας σε αρχεία .tm, ώστε να τα ανοίγετε αργότερα:



Το WinTeXmacs προτείνει να αποθηκεύετε τα αρχεία σας στο φάκελο `C:\Program Files\WinTeXmacs\bin` και, εφόσον συμφωνήσετε και δεν αλλάξετε φάκελο, θα βρίσκονται εκεί. Για λόγους συμβατότητας συνίσταται να αποθηκεύετε τα αρχεία σας σε έναν φάκελο του οποίου το όνομα δεν περιέχει κενά ή ελληνικούς χαρακτήρες, πχ `C:\axiom`.

Μπορείτε επίσης να αποθηκεύετε τα περιεχόμενα της εργασίας σας «τυπώνοντας τα» σε ένα αρχείο .ps, με τον τρόπο που φαίνεται παρακάτω:



Τα αρχεία .ps ανοίγουν με το πρόγραμμα GSview, το οποίο μπορείτε να χρησιμοποιήσετε αφού εγκαταστήσετε το Ghostscript από τη διεύθυνση

`http://mirror.cs.wisc.edu/pub/mirrors/ghost/GPL/gs860/gs860w32.exe`

και κατόπιν το GSview από τη διεύθυνση:

`http://mirror.cs.wisc.edu/pub/mirrors/ghost/ghostgum/gsv49w32.exe`

3 Ρίζες πολυωνύμου

Ας εισάγουμε τώρα ένα πολυώνυμο, το οποίο θα ονομάσουμε f, και ας υπολογίσουμε την τιμή του π.χ. για $x = 3$ ως εξής:

```
→ f:= x^3 - 2*x^2 - 2*x + 4
```

$$x^3 - 2x^2 - 2x + 4 \qquad (7)$$ Type: Polynomial Integer

```
→ eval(f,x=3)
```

$$7 \qquad (8)$$ Type: Polynomial Integer

Μπορούμε να βρούμε την ανάλυση του f σε ανάγωγα πολυώνυμα στο $\mathbb{Q}[x]$ με την εντολή(παρατηρήστε ότι δηλώνουμε πως η μεταβλητή μας είναι το x):

```
→ solve( f , x )
```

$$[x = 2, \;\; x^2 - 2 = 0] \qquad (9)$$ Type: List Equation Fraction Polynomial Integer

Η έξοδος του Axiom είναι μια λίστα με τα δυο πολυώνυμα. Δίνοντας %(1) θα λάβουμε $x = 2$, δηλαδή το πρώτο στοιχείο της λίστας, ενώ με %(2) το δεύτερο.

Αν οι ρίζες του f μπορούν να εκφραστούν με ριζικά, τότε αυτές βρίσκονται με την εντολή:

→ `radicalSolve(f,x)`

$$\left[x=\sqrt{2},\ \ x=-\sqrt{2},\ \ x=2\right] \qquad (10)$$ Type: `List Equation Expression Integer`

Αν σαν δεύτερο όρισμα στην εντολή `solve` δώσουμε μια ακρίβεια, π.χ. 0.0001, θα πάρουμε αριθμητικές προσεγγίσεις των πραγματικών ριζών, με την ακρίβεια αυτή:

→ `solve(f,0.0001)`

$$[x=-1.414215087890625,\ \ x=2.0,\ \ x=1.414215087890625] \qquad (11)$$ Type: `List Equation Polynomial Float`

Για αριθμητικές προσεγγίσεις των μιγαδικών ριζών χρησιμοποιούμε την εντολή:

→ `complexSolve(x^3-2,.0001)`

$$[x=1.259918212890625,\ \ x=-0.62989432795395613131-1.091094970703125i,$$

$$x=-0.62989432795395613131+1.091094970703125i] \qquad (12)$$ Type: `List Equation Polynomial Complex Float`

4 Διαίρεση πολυωνύμων - ΜΚΔ και ΕΚΠ

Θεωρούμε δυο πολυώνυμα με τρεις μεταβλητές:

→ `f:=(y-1)^2*x*z;`

`Polynomial Integer`

→ `g:=(y-1)*x*(z+5);`

`Polynomial Integer`

Παρατηρείστε ότι χρησιμοποιούμε `;` στο τέλος μιας εντολής αν δε θέλουμε να δούμε την αντίστοιχη έξοδο. Παρακάτω βρίσκουμε το Μέγιστο Κοινό Διαιρέτη (GCD) και το Ελάχιστο Κοινό Πολλαπλάσιο(LCM) των f, g και κατόπιν τα παραγοντοποιούμε με χρήση της εντολής factor:

→ `gcd(f,g)`

$$xy-x \qquad (15)$$ Type: `Polynomial Integer`

→ `factor %`

$$x\,(y-1) \qquad (16)$$ Type: `Factored Polynomial Integer`

→ `lcm(f,g)`

$$\left(xy^2-2xy+x\right)z^2+\left(5xy^2-10xy+5x\right)z \qquad (17)$$ Type: `Polynomial Integer`

→ `factor %`

$$x\,(y-1)^2\,z\,(z+5) \qquad (18)$$ Type: `Factored Polynomial Integer`

Το Axiom παρέχει εντολή για τη διαίρεση δυο πολυωνύμων μιας μεταβλητής και με ρητούς συντελεστές, εφόσον ο διαιρέτης είναι *μονικό* πολυώνυμο (δηλαδή έχει μεγιστοβάθμιο όρο 1):

```
→ f:= 5*x^6 + 2*x^2 + 4/17*x + 5/2
```

$$5x^6 + 2x^2 + \frac{4}{17}x + \frac{5}{2} \qquad (19)$$

Type: Polynomial Fraction Integer

```
→ g:=x^2 + 11/45*x - 25/2
```

$$x^2 + \frac{11}{45}x - \frac{25}{2} \qquad (20)$$

Type: Polynomial Fraction Integer

```
→ monicDivide(f,g)
```

$$\left[quotient = 5x^4 - \frac{11}{9}x^3 + \frac{50867}{810}x^2 - \frac{558206}{18225}x + \frac{2606263939}{3280500},\ remainder = -\frac{1447592944093}{2509582500}x + \frac{2606920039}{262440}\right] \qquad (21)$$

Type: Record(quotient: UnivariatePolynomial(x,Fraction Integer),remainder: UnivariatePolynomial(x,Fraction Integer))

Κατόπιν μπορούμε να αναφερθούμε στο πηλίκο ή στο υπόλοιπο όπως φαίνεται παρακάτω, όπου επαληθεύουμε το αποτέλεσμα:

```
→ %.quotient*g + %.remainder
```

$$5x^6 + 2x^2 + \frac{4}{17}x + \frac{5}{2} \qquad (22)$$

Type: Polynomial Fraction Integer

Παρακάτω θα δούμε μια (γενικότερη) εντολή η οποία επιστρέφει το υπόλοιπο της διαίρεσης δυο πολυωνύμων, χωρίς περιορισμό για το διαιρέτη(normalForm).

5 Διατάξεις μονωνύμων και υπόλοιπο διαίρεσης

Για να δουλέψουμε με τη *λεξικογραφική* διάταξη στο Axiom πρέπει να ορίσουμε τον εξής τύπο:

```
→ lex:=DMP([x,y,z], FRAC INT)
```

DistributedMultivariatePolynomial([x,y,z],Fraction Integer) (23) Type: Domain

Αυτό σημαίνει πως ορίζουμε lex να είναι ένας τύπος(ένα «είδος») πολυωνύμου με τα εξής χαρακτηριστικά: έχει ρητούς συντελεστές και οι όροι του είναι διατεταγμένοι με τη λεξικογραφική διάταξη $x > y > z$. Από εδώ και πέρα μπορούμε να ορίσουμε πολυώνυμα με τις ιδιότητες αυτές (δηλαδή αυτού του τύπου) ως εξής:

```
→ f:lex:=y^2 + 2*x*y*z^3 + x^2 + x*y + y^2 +x*z + y*z + y + 12 - x^4 +x*y*z
```

$$-x^4 + x^2 + 2xyz^3 + xyz + xy + xz + 2y^2 + yz + y + 12 \qquad (24)$$

Type: DistributedMultivariatePolynomial([x,y,z],Fraction Integer)

Το κομμάτι :lex σημαίνει πως στο πολυώνυμο αυτό θέλουμε να διατάξουμε τους όρους όπως ορίζει ο τύπος lex, δηλαδή με τη λεξικογραφική διάταξη. Βλέπουμε πως το Axiom επέστρεψε το πολυώνυμο με τους όρους στη «σωστή» σειρά.

Μια άλλη διάταξη που μπορούμε να εφαρμόσουμε είναι η *βαθμωτή αντίστροφη λεξικογραφική*, η οποία σαν πρώτο κριτήριο έχει το συνολικό βαθμό του πολυωνύμου.

```
→ glex:= HDMP([x,y,z], FRAC INT)
```

HomogeneousDistributedMultivariatePolynomial([x,y,z],Fraction Integer) (25) Type: Domain

Έχει παρατηρηθεί πως όταν οι υπολογισμοί με πολυώνυμα (π.χ. η διαίρεση) πραγματοποιούνται με τη συγκεκριμένη διάταξη ο υπολογιστικός χρόνος είναι μικρότερος, σε σχέση με τη λεξικογραφική.

Μπορούμε να αλλάξουμε τη διάταξη που έχουμε χρησιμοποιήσει στο πολυώνυμο f, ως εξής:

```
→ f::glex
```

$$2xyz^3 - x^4 + xyz + x^2 + xy + 2y^2 + xz + yz + y + 12 \quad (26)$$

Type: HomogeneousDistributedMultivariatePolynomial([x,y,z],Fraction Integer)

Το :: σημαίνει, όπως έχει αναφερθεί, ότι θέλουμε να μετατρέψουμε το f σε έναν διαφορετικό τύπο. Πράγματι, τώρα οι όροι είναι διατεταγμένοι σύμφωνα με τη νέα διάταξη. Σημειώστε πως με την εντολή αυτή δεν τροποποιήσαμε τη μεταβλητή f, απλώς εμφανίσαμε το f στην έξοδο, με διαφορετική διάταξη. Αν είχαμε δώσει `f:= f::glex` τότε το f θα αποθηκευόταν στη μνήμη με τη νέα διάταξη.

Έστω δυο πολυώνυμα με τη λεξικογραφική διάταξη:

```
→ f1: lex := x + 2*x*y
```

$$2xy + x \quad (27)$$

Type: DistributedMultivariatePolynomial([x,y,z],Fraction Integer)

```
→ f2: lex := x^2 + y
```

$$x^2 + y \quad (28)$$

Type: DistributedMultivariatePolynomial([x,y,z],Fraction Integer)

Μπορούμε να λάβουμε τη λίστα των μεταβλητών που εμφανίζονται στο f_1 δίνοντας:

```
→ variables f1
```

$$[x,\ y] \quad (29)$$

Type: List OrderedVariableList [x,y,z]

Το διάνυσμα των βαθμών (σύμφωνα με τη διάταξη x, y των μεταβλητών) είναι:

```
→ degree(f2,[x,y])
```

$$[2,\ 1] \quad (30)$$

Type: NonNegativeInteger

Αν θέλουμε να δούμε το συνολικό βαθμό του f_1 δίνουμε:

```
→ totalDegree(f1)
```

$$2 \quad (31)$$

Type: PositiveInteger

Ο μεγιστοβάθμιος όρος του f_1 δίνεται πληκτρολογώντας

```
→ leadingMonomial f1
```

$2xy$ (32) Type: DistributedMultivariatePolynomial([x,y,z],Fraction Integer)

ενώ ο συντελεστής του μεγιστοβαθμίου όρου με

→ `leadingCoefficient f1`

2 (33) Type: Fraction Integer

Έχοντας ορίσει ένα πολυώνυμο πολλών μεταβλητών, μαζί με τη διάταξη που θέλουμε να έχουμε στους όρους του, μπορούμε να λάβουμε το υπόλοιπο της διαίρεσης με μια λίστα από πολυώνυμα. Το υπόλοιπο της διαίρεσης του παρακάτω πολυωνύμου δια (f_1, f_2) δίνεται με την εντολή†:

→ `normalForm(x^5*y^8+x^4*y^2-6*x^3*y^3-8*x^2*y+25::lex, [f1,f2] )`

$$\frac{385}{1024}\,x - \frac{33}{8}\,y + 25 \qquad (34)$$

Type: DistributedMultivariatePolynomial([x,y,z],Fraction Polynomial Fraction Integer

Παρατηρήστε ότι στο τέλος του διαιρετέου προσάπτουμε `::lex` για να δηλώσουμε πως θέλουμε να είναι και ο διαιρετέος πολυώνυμο με την ίδια διάταξη όπως τα $f_1,\ f_2$.

Αν δώσουμε την εντολή:

→ `normalForm(x^5*y^8+x^4*y^2-6*x^3*y^3-8*x^2*y+25::lex, [f2,f1] )`

$$\frac{385}{1024}\,x + y^4 + 8y^2 + 25 \qquad (35)$$

Type: DistributedMultivariatePolynomial([x,y,z],Fraction Polynomial Fraction Integer βλέπουμε ότι στη διαίρεση με πολλές μεταβλητές το υπόλοιπο δε μένει γενικά το ίδιο, όταν αλλάζει η σειρά των διαιρετών.

6 $S-$πολυώνυμα

Συνεχίζουμε με τα ίδια $f_1,\ f_2$ όπως πιο πάνω.

Αν θέλουμε να λάβουμε τα μεγιστοβάθμια μονώνυμα των $f_1,\ f_2$ χωρίς το συντελεστή τους, δεν έχουμε παρά να διαιρέσουμε τους μεγιστοβάθμιους όρους με το συντελεστή τους:

→ `M1:= leadingMonomial f1 / leadingCoefficient f1`

xy (36) Type: DistributedMultivariatePolynomial([x,y,z],Fraction Integer)

→ `M2:= leadingMonomial f2 / leadingCoefficient f2`

x^2 (37) Type: DistributedMultivariatePolynomial([x,y,z],Fraction Integer)

Γνωρίζουμε τώρα πως το $S-$πολυώνυμο $S(f_1, f_2)$ δίνεται ως εξής:

→ `Spoly:=( lcm(M1,M2)/leadingMonomial f1)*f1 - ( lcm(M1,M2)/leadingMonomial f2)*f2`

†Στην ενότητα 11 υλοποιούμε τον αλγόριθμο της διαίρεσης, ώστε να μας επιστρέφει και το πηλίκο.

$\frac{1}{2}x^2 - y^2$ (38) Type: Fraction DistributedMultivariatePolynomial([x,y,z],Fraction Integer)

Στο Axiom μπορούμε να δηλώσουμε μια συνάρτηση, π.χ. τη συνάρτηση $S-$πολυώνυμο, χρησιμοποιώντας τη διπλή ισότητα (==). Σε παρενθέσεις βάζουμε τα ορίσματα της και μετά το διπλό ίσον την τιμή που θέλουμε να αντιστοιχίζει η συνάρτησή μας σε αυτά. Παρακάτω ορίζουμε το $S-$πολυώνυμο δυο πολυωνύμων f, g:

```
→ S(f,g) == ( lcm(leadingMonomial f/leadingCoefficient f, leadingMonomial g/
leadingCoefficient g)/leadingMonomial f)*f - ( lcm(leadingMonomial f/
leadingCoefficient f, leadingMonomial g/ leadingCoefficient g)/leadingMonomial g)*g
```

Void

Έτσι μπορούμε πολύ εύκολα να υπολογίσουμε όποιο $S-$πολυώνυμο θέλουμε, π.χ.:

```
→ S(f1,f2)
```

```
Compiling function S with type (DistributedMultivariatePolynomial([x
,y,z],Fraction Integer),DistributedMultivariatePolynomial([x,y,z]
,Fraction Integer)) -> Fraction DistributedMultivariatePolynomial
([x,y,z],Fraction Integer)
```

$\frac{1}{2}x^2 - y^2$ (40) Type: Fraction DistributedMultivariatePolynomial([x,y,z],Fraction Integer)

```
→ S(f2,f1)
```

$-\frac{1}{2}x^2 + y^2$ (41) Type: Fraction DistributedMultivariatePolynomial([x,y,z],Fraction Integer)

Παρατηρήστε ότι επαληθεύεται η γνωστή σχέση $S(f_1, f_2) = -S(f_2, f_1)$.

Ας βρούμε τώρα μια βάση Groebner του $I = \langle f_1, f_2 \rangle$ ακολουθώντας τον αλγόριθμο Buchberger: Υπολογίζουμε $f_3 = \overline{S(f_1, f_2)}^{\{f_1, f_2\}}$:

```
→ f3:= normalForm( S(f1,f2)::lex, [f1,f2])
```

$-y^2 - \frac{1}{2}y$ (42) Type: DistributedMultivariatePolynomial([x,y,z],Fraction Integer)

Καθώς είναι μη μηδενικό το προσθέτουμε στη βάση που φτιάχνουμε και συνεχίζουμε να ελέγχουμε τα καινούρια $S-$πολυώνυμα:

```
→ normalForm(S(f1,f3)::lex, [f1,f2,f3])
```

0 (43) Type: DistributedMultivariatePolynomial([x,y,z],Fraction Integer)

```
→ normalForm(S(f2,f3)::lex, [f1,f2,f3])
```

0 (44) Type: DistributedMultivariatePolynomial([x,y,z],Fraction

Integer)

Βλέπουμε ότι όλα τα υπόλοιπα είναι μηδενικά, άρα σύμφωνα με το κριτήριο Buchberger το σύνολο $\{f_1, f_2, f_3\}$ είναι βάση Groebner του ιδεώδους I. Στα επόμενα θα δούμε πως το Axiom έχει έτοιμες εντολές για να υπολογίζει βάσεις Groebner.

7 Βάσεις Groebner

Ας θεωρήσουμε τη λεξικογραφική διάταξη

```
→ lex:=DMP([x,y,z], FRAC INT)
```

DistributedMultivariatePolynomial([x,y,z],Fraction Integer) (45) Type: Domain

και τρία πολυώνυμα:

```
→ g: lex := x^3 + 3*y^2;
```

Type: DistributedMultivariatePolynomial([x,y,z],Fraction Integer)

```
→ f1 : lex := x + 2*x*y;
```

Type: DistributedMultivariatePolynomial([x,y,z],Fraction Integer)

```
→ f2 : lex := x^2 + y;
```

Type: DistributedMultivariatePolynomial([x,y,z],Fraction Integer)

Θα υπολογίσουμε την ανηγμένη βάση Groebner του ιδεώδους που παράγεται από το

```
→ F:= [ f1,f2 ]
```

$$\left[2xy + x,\ \ x^2 + y\right] \tag{49}$$

Type: DistributedMultivariatePolynomial([x,y,z],Fraction Integer)

με την εντολή:

```
→ G:= groebner(F)
```

$$\left[x^2 + y,\ \ xy + \frac{1}{2}x,\ \ y^2 + \frac{1}{2}y\right] \tag{50}$$

Type: List DistributedMultivariatePolynomial([x,y,z],Fraction Integer)

Αν και το υπόλοιπο

```
→ normalForm(g,F)
```

$$\frac{1}{2}x + 3y^2 \tag{51}$$

Type: DistributedMultivariatePolynomial([x,y,z],Fraction Integer)

δε μένει το ίδιο αν αλλάξουμε τη σειρά των διαιρετών, μπορείτε να ελέγξετε ότι το υπόλοιπο

```
→ normalForm(g,G)
```

$\frac{1}{2}x - \frac{3}{2}y$ (52) Type: DistributedMultivariatePolynomial([x,y,z],Fraction Integer)

παραμένει το ίδιο αν διαιρέσουμε με μια μετάθεση του G.

Χρησιμοποιώντας σαν δεύτερο όρισμα την επιλογή "redcrit" μπορούμε να δούμε διαδοχικά τα πολυώνυμα που το Axiom προσθέτει στη βάση. Επειδή η εντολή groebner επιστρέφει την ανηγμένη βάση Groebner, δεν υπολογίζει απλώς το $\overline{S(f_i, f_j)}^F$, αλλά το επεξεργάζεται κατάλληλα πριν το εισάγει στη βάση (δείτε ότι αυτό το πολυώνυμο καλείται reduced Critpair). Για παράδειγμα, παρακάτω το $\overline{S(f_1, f_2)}^F$ έχει πολλαπλασιαστεί με -1, έτσι ώστε να γίνει μονικό, πρωτού προστεθεί στη βάση:

```
→ groebner(F,"redcrit")
```

```
reduced Critpair - Polynom :
 2   1
y  + - y
     2
 reduced Critpair - Polynom :
0
    THE GROEBNER BASIS POLYNOMIALS
```

$$\left[x^2 + y,\ \ xy + \frac{1}{2}x,\ \ y^2 + \frac{1}{2}y\right] \qquad (53)$$

Type: List DistributedMultivariatePolynomial([x,y,z],Fraction Integer)

Τέλος, για να βρούμε γρήγορα μια βάση Groebner με τη λεξικογραφική διάταξη, μπορούμε να δώσουμε την εντολή σε μια γραμμή ως εξής:

```
→ lexGroebner( [x + 2*x*y, x^2 + y], [x,y] )
```

$$\left[y + x^2,\ \ 2xy + x,\ \ 2y^2 + y\right] \qquad (54)$$

Type: List Polynomial Integer

Εδώ το Axiom παίρνει του τα πολυώνυμα που δώσαμε χωρίς κάποια διάταξη, τα διατάσσει με τη λεξικογραφική διάταξη $x > y$ και εκτελεί τον αλγόριθμο Buchberger. Παρατηρήστε ότι η εντολή αυτή επιστρέφει μια βάση με ακέραιους συντελεστές: Καθώς τα πολυώνυμα που δώσαμε έχουν ακέραιους συντελεστές, το Axiom θεώρησε ότι δουλεύουμε σε πολυώνυμα στο $\mathbb{Z}[x, y]$. Όμοια δουλεύει και η εντολή totalGroebner αν θέλουμε να χρησιμοποιήσουμε τη βαθμωτή αντίστροφη λεξικογραφική διάταξη.

8 Επίλυση συστήματος με χρήση βάσεων Groebner

Στο Axiom μπορούμε να θεωρήσουμε το ιδεώδες που παράγεται από το $F = \{f_1, f_2\}$. Η *διάσταση* είναι μια σημαντική ιδιότητα των ιδεωδών: όταν ένα ιδεώδες έχει μηδενική διάσταση, το σύνολο των λύσεων είναι πεπερασμένο. Μπορούμε να ελέγξουμε αν το σύστημα $f_1 = f_2 = 0$ έχει πεπερασμένου πλήθους λύσεις, ρωτώντας αν το ιδεώδες έχει μηδενική διάσταση:

```
→ ideal F
```

$$\left[2xy + x,\ \ x^2 + y\right] \qquad (55)$$

Type: PolynomialIdeals(Fraction Integer, DirectProduct(3,NonNegativeInteger), OrderedVariableList [x,y,z],

```
DistributedMultivariatePolynomial([x,y,z], Fraction Integer))
```

→ `zeroDim? %`

$$true \qquad (56)$$ Type: Boolean

Βλέπουμε πως αυτό είναι αληθές, άρα το σύστημα έχει πεπερασμένου πλήθους λύσεις.

Παρατηρήστε τώρα ότι στη βάση Groebner G του ιδεώδους, υπάρχει το πολυώνυμο

→ `G(3)`

$$y^2 + \frac{1}{2}y \qquad (57)$$ Type: DistributedMultivariatePolynomial([x,y,z],Fraction Integer

το οποίο είναι μιας μεταβλητής. Οι ρίζες του είναι οι μόνες τιμές που μπορεί να πάρει η μεταβλητή y, ώστε να φτάσουμε σε λύση του συστήματος. Με το σκεπτικό αυτό ακολουθούμε την εξής διαδικασία:

1. Βρίσκουμε της ρίζες του $y^2 + \frac{1}{2}y$

   → `s1:= radicalSolve( G(3) )`

   $$\left[y = -\frac{1}{2}, \ \ y = 0\right] \qquad (58)$$ Type: List Equation Expression Fraction Integer

2. Αντικαθιστούμε την τιμή $y = -\frac{1}{2}$ στα υπόλοιπα πολυώνυμα της βάσης και (εφόσον τώρα υπάρχει μόνο η x μεταβλητή) αναζητούμε τις λύσεις ως προς x. Για το $G(2)$ είναι:

   → `radicalSolve( eval( G(2), y=-1/2) )`

   ```
     >> Error detected within library code:
   equation is always satisfied
   ```

   δηλαδή ικανοποιείται για κάθε x. Για το $G(1)$:

   → `radicalSolve( eval( G(1), y=-1/2) )`

   $$\left[x = \frac{1}{\sqrt{2}}, \ \ x = -\frac{1}{\sqrt{2}}\right] \qquad (59)$$ Type: List Equation Expression Integer

   δηλαδή για $y = -\frac{1}{2}$ παίρνουμε τις τιμές $x = \pm\frac{1}{\sqrt{2}}$. Άρα δυο λύσεις του συστήματος είναι οι:

   $$\left(\frac{1}{\sqrt{2}}, -\frac{1}{2}\right) \quad \text{και} \quad \left(-\frac{1}{\sqrt{2}}, -\frac{1}{2}\right)$$

3. Επαναλαμβάνουμε για την τιμή $y = 0$:

   → `radicalSolve( eval( G(2), y=0) )`

   $$[x = 0] \qquad (60)$$ Type: List Equation Expression Fraction Integer

   → `radicalSolve( eval( G(1), y=0) )`

   $$[x = 0, \ \ x = 0] \qquad (61)$$ Type: List Equation Expression Fraction Integer

   Άρα στις λύσεις του συστήματος προστίθεται και η $(0, 0)$.

Τελικά το σύστημα $f_1 = f_2 = 0$ έχει ακριβώς τρεις λύσεις.

9 Πολυώνυμα με συντελεστές από άλλα σώματα

Πολλές φορές χρειάζεται να δουλεύουμε με συντελεστές από άλλα αλγεβρικά σώματα αντί των ρητών, όπως έχουμε κάνει ως τώρα. Όλες οι εντολές δουλεύουν το ίδιο εφόσον δηλώσουμε στον *τύπο* το σώμα στο οποίο βρίσκονται οι συντελεστές των πολυωνύμων μας. Θα δώσουμε τις απαραίτητες εντολές για τρεις χρήσιμες κατηγορίες σωμάτων.

1. Ρητές επεκτάσεις: Αν θέλουμε να έχουμε μια (ή περισσότερες) παράμετρο στους συντελεστές των πολυωνύμων, η οποία δεν πρέπει να εκλαμβάνεται σαν μεταβλητή, τότε είναι απαραίτητο να δουλέψουμε σε μια επέκταση των ρητών. Για παράδειγμα, αν έχουμε παραμέτρους u, v τότε οι συντελεστές των πολυωνύμων βρίσκονται στο $\mathbb{Q}(u, v)$. Ας ορίσουμε τον ανάλογο τύπο:

```
→ lex1 := DMP( [x,y,z], FRAC POLY INT )
```

DistributedMultivariatePolynomial([x,y,z],Fraction Polynomial Integer) (62) Type: Domain

Δείτε πως το μόνο που αλλάζει είναι η λέξη POLY στη δήλωση μετά το FRAC. Έτσι οι συντελεστές μας είναι λόγοι πολυωνύμων ως προς τις παραμέτρους u, v. Ορίζουμε δυο πολυώνυμα αυτού του τύπου:

```
→ h1: lex1 := v*x^2 + y
```

$$vx^2 + y \qquad (63)$$

Type: DistributedMultivariatePolynomial([x,y,z],Fraction Polynomial Integer)

```
→ h2: lex1 := u*x*y + y^2
```

$$uxy + y^2 \qquad (64)$$

Type: DistributedMultivariatePolynomial([x,y,z],Fraction Polynomial Integer)

και υπολογίζουμε τη βάση Groebner του αντίστοιχου ιδεώδους:

```
→ H := groebner( [h1,h2] )
```

$$\left[x^2 + \frac{1}{v}y,\ \ xy + \frac{1}{u}y^2,\ \ y^3 + \frac{u^2}{v}y^2\right] \qquad (65)$$

Type: List DistributedMultivariatePolynomial([x,y,z],Fraction Integer)

Η παραπάνω διαδικασία είναι χρήσιμη όταν θέλουμε να εντοπίσουμε κατάλληλες τιμές, ώστε δυο σύνολα πολυωνύμων να παράγουν το ίδιο ιδεώδες. Για παράδειγμα, αν έχουμε δυο ακόμα πολυώνυμα, με ρητούς τώρα συντελεστές

```
→ lex:=DMP([x,y,z], FRAC INT)
```

DistributedMultivariatePolynomial([x,y,z],Fraction Integer) (66) Type: Domain

```
→ f1:lex:=y^2+2*x*y;  f2:lex:=y^2+x^2+2*x*y+y;
```

DistributedMultivariatePolynomial([x,y,z],Fraction Integer)

υπολογίσουμε τη βάση Groebner:

```
→ F:= groebner( [f1,f2] )
```

$$\left[x^2+y,\ \ xy+\frac{1}{2}y^2,\ \ y^3+4y^2\right] \qquad (68)$$

Type: List DistributedMultivariatePolynomial([x,y,z],Fraction Integer)

και ψάχνουμε τις τιμές των u, v ώστε $\langle h_1, h_2\rangle = \langle f_1, f_2\rangle$, τότε αρκεί να εξισώσουμε τα πολυώνυμα των βάσεων H και F:

```
→ solve( H(1)=F(1), v)
```

$$[v=1] \qquad (69)$$

Type: List Equation Fraction Polynomial Integer

```
→ solve( H(2)=F(2), u )
```

$$[u=2] \qquad (70)$$

Type: List Equation Fraction Polynomial Integer

άρα οι ζητούμενες τιμές είναι $v = 1$ και $u = 2$.

2. Πεπερασμένα σώματα: Με την επόμενη εντολή ορίζουμε έναν τύπο πολυωνύμου με τη λεξικογραφική διάταξη και συντελεστές από το πεπερασμένο σώμα $\mathbb{Z}_p$ των ακεραίων modulo p, όπου p πρώτος αριθμός. Στο παράδειγμά μας $p = 17$:

```
→ lex3:= DMP( [x,y,z], PrimeField(17) )
```

DistributedMultivariatePolynomial([x,y,z],PrimeField 17) (71) Type: Domain

Ορίζουμε ένα $f \in \mathbb{Z}_{17}[x, y]$:

```
→ f: lex3 := 23*x^3 + 8*y^2 + 17*y + 45
```

$$6x^3+8y^2+11 \qquad (72)$$

Type: DistributedMultivariatePolynomial([x,y,z],PrimeField 17)

Παρατηρήστε ότι οι συντελεστές που επιστρέφει το Axiom ανήκουν στο $\{0, 1, 2, \ldots, 16\}$.

3. Ρητοί του Gauss: Αν θέλουμε να επιτρέψουμε και μιγαδικούς συντελεστές στα πολυώνυμά μας, το κατάλληλο σώμα συντελεστών είναι το $\mathbb{Q}(i) = \{\alpha + \beta i \mid \alpha, \beta \in \mathbb{Q}\}$ των *ρητών του Gauss*:

```
→ lex4 := DMP( [x,y,z], FRAC COMPLEX INT)
```

DistributedMultivariatePolynomial([x,y,z],Fraction Complex Integer) (73) Type: Domain

```
→ q:= (2+%i)*x^2 + 5*%i*y + 4+2*%i
```

$$5iy+(2+i)\,x^2+4+2i \qquad (74)$$

Type: Polynomial Complex Integer

```
→ q := q::lex4
```

$$(2+i)\,x^2+5iy+4+2i \qquad (75)$$

Type: DistributedMultivariatePolynomial([x,y,z],Fraction Complex Integer)

Δείτε ότι η *φανταστική μονάδα* $i = \sqrt{-1}$ εισάγεται στο Axiom με %i.

10 Προγραμματισμός στο Axiom

Σε αυτήν την ενότητα θα δώσουμε τα απαραίτητα για τη γραφή ενός προγράμματος στο Axiom. Τα προγράμματα στο Axiom γράφονται σε αρχεία με επέκταση .input, και μπορούν να συνταχθούν με κάποιο κειμενογράφο (πχ το Σημειωματάριο των Windows). Ένα αρχείο .input μπορεί να περιέχει περισσότερες από μια συναρτήσεις και όταν με την κατάλληλη εντολή το φορτώσουμε στο Axiom, όλες οι συναρτήσεις που περιέχονται σε αυτό είναι στη διάθεσή μας για να τις καλέσουμε.

Μια συνάρτηση αποτελείται από δυο μέρη. Το πρώτο είναι η *δήλωση*, η οποία καταλαμβάνει μια γραμμή και δηλώνει το όνομα, τους τύπους των ορισμάτων και τον τύπο του αποτελέσματος. Για παράδειγμα δηλώνουμε τη συνάρτηση Metro1:

```
Metro1 : (Integer,Integer) -> AlgebraicNumber
```

Δηλαδή δηλώνεται μια συνάρτηση με όνομα Metro1 και είσοδο ένα ζεύγος ακεραίων, έξοδο έναν αλγεβρικό αριθμό.

Αμέσως μετά ακολουθεί το *σώμα* της συνάρτησης. Το σώμα ξεκινά με μια γραμμή `Metro1(x,,y)==` και όλες οι εντολές που βρίσκονται παρακάτω ξεκινούν ένα TAB (στηλοθέτη) πιο δεξιά. Με αυτόν τον τρόπο το Axiom «καταλαβαίνει» ποιες εντολές υπάγονται σε ποια συνάρτηση. Αυτό είναι απαραίτητο, καθώς ένα αρχείο .input μπορεί να περιέχει πολλές διαφορετικές συναρτήσεις. Έτσι πρέπει να ξέρουμε που ξεκινά και που τελειώνει μια συνάρτηση.

Μέσα σε μια συνάρτηση μπορούμε να δώσουμε οποιαδήποτε εντολή του Axiom καθώς και να δηλώσουμε νέες μεταβλητές (αυτές είναι *τοπικές μεταβλητές*, δηλαδή χρησιμοποιούνται μόνο εσωτερικά από τη συνάρτηση και διαγράφονται από τη μνήμη μόλις τελειώσει η εκτέλεσή της). Όταν με μια σειρά εντολών υπολογίσουμε το επιθυμητό αποτέλεσμα, χρησιμοποιούμε την εντολή `return` για να δηλώσουμε το τέλος της εκτέλεσης και να επιστρέψουμε το αποτέλεσμα στη γραμμή εντολών του Axiom.

Για να γίνουν όλα αυτά κατανοητά, δίνουμε δυο συναρτήσεις που υπολογίζουν δυο διαφορετικά μέτρα διανυσμάτων με ακέραιες συντεταγμένες, τα $\|(x,y)\|_1 = |x| + |y|$ και $\|(x,y)\|_2 = \sqrt{x^2+y^2}$.

```
Metro1 : (Integer,Integer) -> AlgebraicNumber
Metro1(x,y)==
        t:= abs(x) + abs(y)
        return t

Metro2 : (Integer,Integer) -> AlgebraicNumber
Metro2(x,y)==
        t:= x^2+y^2
        t:= sqrt(t)
        return t
```

Παρατηρήστε ότι η είσοδος είναι ένα ζεύγος ακεραίων, η `t` είναι μια τοπική μεταβλητή και το αποτέλεσμα που επιστρέφεται στο χρήστη είναι το περιεχόμενο της `t` με την εντολή `return`. Το σώμα της συνάρτησης ξεκινά με το όνομα, δυο σύμβολα `(x,y)` για την είσοδο και όλες οι υπόλοιπες γραμμές βρίσκονται μια στήλη δεξιότερα (με χρήση TAB).

Δίνουμε βήμα βήμα τον τρόπο αποθήκευσης και εκτέλεσης των δυο παραπάνω συναρτήσεων, τις οποίες θα εναποθέσουμε στο αρχείο metro.input.

1. Ανοίγουμε έναν κειμενογράφο (κατά προτίμηση το Σημειωματάριο(notepad) των Windows) και γράφουμε τις συναρτήσεις ακριβώς όπως φαίνονται παραπάνω στο καφέ κουτί.

2. Αποθηκεύουμε το αρχείο σαν metro.input, πχ στο φάκελο `C:\axiom` (*Αρχείο* → *Αποθήκευση*). Το Σημειωματάριο συνήθως προσθέτει αυτόματα μια κατάληξη .txt. Για να μη γίνει αυτό επιλέξτε «Αποθήκευση ως: Όλα τα αρχεία» στο παράθυρο αποθήκευσης πριν πατήσετε το κουμπί *Αποθήκευση*.

3. Ανοίγουμε το WinTeXmacs και ξεκινούμε το Axiom. Για να φορτώσουμε το αρχείο δίνουμε:

   → `)read "C:/axiom/metro.input"`

   Αν υπάρχουν λάθη στις εντολές κάποιας συνάρτησης το Axiom θα υποδείξει τη γραμμή που υπάρχει το λάθος και δε θα φορτώσει τη συνάρτηση. Μπορούμε να διορθώσουμε το λάθος στο αρχείο metro.input και να εκτελέσουμε πάλι την εντολή `read` για να φορτώσουμε εκ νέου το αρχείο στο Axiom (παρατηρήστε ότι πρόκειται για εντολή συστήματος). Διαφορετικά μπορούμε να χρησιμοποιήσουμε τις νέες συναρτήσεις μας:

   → `Metro1(5,-4)`

   9 (76) `Type: AlgebraicNumber`

   → `Metro2(5,-4)`

   $\sqrt{41}$ (77) `Type: AlgebraicNumber`

   Σημειώστε ότι την πρώτη φορά που θα καλέσετε τη συνάρτηση θα γίνει μια απαραίτητη διαδικασία (μεταγλώτιση - compilation) ώστε να μπορεί να γίνει χρήση της συνάρτησης. Έτσι θα δείτε το μήνυμα πχ

   ```
   Compiling function Metro1 with type (Integer,Integer)   AlgebraicNumber
   ```

   πριν πάρετε το αποτέλεσμα της εκτέλεσης.

10.1 Τύποι στο Axiom

Οι τύποι στο Axiom έχουν μακροσκελή ονόματα. Μπορούμε για ευκολία να χρησιμοποιήσουμε συντμήσεις για διάφορους τύπους. Για παράδειγμα, το INT είναι σύντμηση του Integer. Οι συντμήσεις ξεχωρίζουν επειδή γράφονται πάντα με κεφαλαία. Παρακάτω δίνεται ένας πίνακας με τους βασικότερους τύπους και τις συντμήσεις τους.

Τύπος	Πλήρες όνομα	Σύντμηση
ακέραιος	`Integer`	`INT`
θετικός ακέραιος	`PositiveInteger`	`PI`
μη αρνητικός ακέραιος	`NonNegativeInteger`	`NNI`
ακέραιος Gauss	`Complex Integer`	`COMPLEX INT`
ρητός	`Fraction Integer`	`FRAC INT`
ρητός Gauss	`Complex Fraction Integer`	`COMPLEX FRAC INT`
λογική μεταβλητή	`Boolean`	`BOOLEAN`
πραγματικός αριθμός κινητής υποδιαστολής	`Float`	`FLOAT`
μιγαδικός αριθμός κινητής υποδιαστολής	`Complex Float`	`COMPLEX FLOAT`
αλγεβρικός αριθμός	`AlgebraicNumber`	`AN`
πολυώνυμο	`Polynomial X`	`POLY X`
πολυώνυμο πολλών μεταβλητών	`Distr...Mult...Poly...(V,X)`	`DMP(V,X)`
ομογενές πολυώνυμο πολλών μεταβλητών	`Hom...Distr...(V,X)`	`HDMP(V,X)`
πίνακας	`Matrix`	`MATRIX`
διάνυσμα	`Vector`	`VECTOR`
λίστα	`List`	`LIST`
εγγραφή	`Record`	-

Έχουμε ήδη χρησιμοποιήσει συντμήσεις στον ορισμό του τύπου για τη λεξικογραφική διάταξη: δώσαμε `DMP([x,y,z], FRAC INT)` αντί `DistributedMultivariatePolynomial([x,y,z],Fraction Integer)`. Ας δούμε πιο προσεκτικά τι σημαίνει αυτή η δήλωση. Ορίζουμε έναν τύπο «κατανεμημένου πολυωνύμου πολλών μεταβλητών». Αυτός για να οριστεί πλήρως χρειάζεται δυο ακόμη ορίσματα. Το πρώτο είναι μια λίστα με τις μεταβλητές `V`, εδώ `[x,y,z]` και το δεύτερο το σώμα `X` στο οποίο βρίσκοται οι συντελεστές. Εδώ οι συντελεστές είναι τύπου `FRAC INT` δηλαδή ρητοί αριθμοί. Παρατηρούμε ότι κάποιοι τύποι μπορεί να παίρνουν σαν ορίσματα άλλους τύπους, πχ οι ρητοί είναι κλάσματα ακεραίων, οπότε το Axiom προσφέρει έναν τύπο `FRAC` ο οποίος παίρνει ένα όρισμα, τον τύπο των αριθμών που θέλουμε να έχει το κλάσμα. Αυτή είναι μια γενικότερη φιλοσοφία στο Axiom η οποία του προσδίδει μεγάλη ευελιξία.

10.2 Εργασία με λίστες

Μια πολύ χρήσιμη δομή στο Axiom είναι η λίστα. Αρκετές από τις εντολές που έχουμε δει επιστρέφουν λίστες με περιεχόμενα συγκεκριμένου τύπου. Όταν γράφουμε ένα πρόγραμμα εμφανίζεται η ανάγκη να δουλεύουμε με λίστες. Θα δώσουμε (μέσα από παραδείγματα) τις βασικές εργασίες που μπορούμε να κάνουμε με αυτές.

- Πλήθος στοιχείων σε λίστα. Αρκεί να βάλουμε στην αρχή το χαρακτήρα `#`:

  → `#[1,2,1,2]`

  $$4 \qquad (78)$$

  Type: `PositiveInteger`

- Έλεγχος αν ένα στοιχείο ανήκει σε μια λίστα.

  → `member?(1,[1,2,3])`

  true (79)

  Type: `Boolean`

→ `member?(4,[1,2,3])`

$$false \quad (80)$$ Type: Boolean

- Αντιστροφή μιας λίστας.

  → `reverse([1,2,3,4])`

  $$[4,3,2,1] \quad (81)$$ Type: List PositiveInteger

- Προσθήκη στοιχείου στο τέλος μιας λίστας

  → `concat([1,2,3],4)`

  $$[1,2,3,4] \quad (82)$$ Type: List PositiveInteger

  ή ακόμα και στην αρχή μιας λίστας:

  → `concat(4,[1,2,3])`

  $$[4,1,2,3] \quad (83)$$ Type: List PositiveInteger

- Παράθεση δυο λιστών σε μια νέα λίστα

  → `append([1,2],[3,4])`

  $$[1,2,3,4] \quad (84)$$ Type: List PositiveInteger

- Ταξινόμηση μιας λίστας (χρησιμοποιώντας τη διάταξη που είναι ορισμένη στα στοιχεία της):

  → `sort([3,1,4,2])`

  $$[1,2,3,4] \quad (85)$$ Type: List PositiveInteger

- Διαγραφή διπλών στοιχείων.

  → `removeDuplicates([1,2,1,2,3,3,4])`

  $$[1,2,3,4] \quad (86)$$ Type: List PositiveInteger

Αν θελουμε να φτιάξουμε μια λίστα με όλους τους αριθμούς από 1 ως 10 και από 20 ως 30 δίνουμε:

→ `u:= [ 1..10 , 20..30 ]`

$$[1..10, 20..30] \quad (87)$$ Type: List Segment PositiveInteger

Βλέπουμε ότι οι τελίτσες δηλώνουν όλους τους ενδιάμεσους αριθμούς. Για να δούμε τη λίστα σε πλήρη ανάπτυξη δίνουμε:

→ `expand(u)`

$$[1,2,3,4,5,6,7,8,9,10,20,21,22,23,24,25,26,27,28,29,30] \quad (88)$$ Type: List Integer

Μπορούμε γρήγορα να δημιουργήσουμε λίστες με τη χρήση for-in:

→ `u:= [i^2 for i in 1..10]`

$$[1,4,9,16,25,36,49,64,81,100] \quad (89)$$ Type: List PositiveInteger

Εδώ το Axiom φτιάχνει μια λίστα u με τα τετράγωνα των αριθμών $1,2,\ldots,10$. Θα δούμε τη δομή for και στα επόμενα.

Επίσης με χρήση του χαρακτήρα | μπορούμε να εισάγουμε περιορισμούς για τη μεταβλητή i, για παράδειγμα αν θέλουμε μόνο τα τετράγωνα των άρτιων αριθμών στο $\{1, 2, \ldots, 10\}$:

→ `u:= [i^2 for i in 1..10 | rem(i,2)=0]`

$$[4, 16, 36, 64, 100] \tag{90}$$

Type: List PositiveInteger

Αν θέλουμε να λάβουμε πχ το άθροισμα όλων των στοιχείων της λίστας ή γενικότερα να εφαρμόσουμε έναν τελεστή σε όλα τα στοιχεία της λίστας χρησιμοποιούμε την εντολή `reduce`:

→ `reduce(+, [3,1,4,2])`

$$10 \tag{91}$$

Type: PositiveInteger

Παρόμοια μπορούμε να βρούμε το μέγιστο ή το ελάχιστο στοιχείο στη λίστα:

→ `reduce(max, [3,1,4,2])`

$$4 \tag{92}$$

Type: PositiveInteger

→ `reduce(min, [3,1,4,2])`

$$1 \tag{93}$$

Type: PositiveInteger

10.3 Δομή ελέγχου συνθήκης

Ο έλεγχος συνθήκης γίνεται με χρήση του γνωστού if-then-else. Όπως και με το σώμα της συνάρτησης, ξεκινούμε τις εντολές μέσα στο if μια στήλη δεξιότερα, το else στην ίδια στήλη με το if και οτιδήποτε θέλουμε να εκτελεστεί στην περίπτωση «else» πάλι μια στήλη δεξιότερα.

Η παρακάτω συνάρτηση παίρνει είσοδο ένα πολυώνυμο και έναν αριθμό και επιστρέφει $true$ αν ο αριθμός είναι ρίζα του πολυωνύμου, $false$ διαφορετικά.

```
isRoot: (POLY FRAC INT,AN) -> BOOLEAN
isRoot(f,a)==
        if eval(f,x=a)=0 then
                return true
        else
                return false
```

Αν φορτώσουμε τη συνάρτηση στο Axiom θα έχουμε:

→ `isRoot( x^2+2 , sqrt(2) )`

$$false \tag{94}$$

Type: Boolean

→ `isRoot( x^2-2 , sqrt(2) )`

$$true \tag{95}$$

Type: Boolean

Δίνουμε έναν πίνακα με τους βασικότερους τελεστές (αριθμητικούς, λογικούς) που μπορούμε να χρησιμοποιήσουμε για να συντάξουμε μια συνθήκη.

Σύμβολο	Λειτουργία	Σύμβολο	Λειτουργία
+	πρόσθεση	-	αφαίρεση
*	πολλ/σμός	/	διαίρεση
=	ισότητα	~=	μη ισότητα
<	μικρότερο	>	μεγαλύτερο
<=	μικρότερο ή ίσο	>=	μεγαλύτερο ή ίσο
max	μέγιστο	min	ελάχιστο
and	λογική σύζευξη	/\	λογική σύζευξη
or	λογική διάζευξη	\/	λογική διάζευξη
not	λογική άρνηση	~	λογική άρνηση

Παρατηρήστε ότι οι λογικοί τελεστές μπορούν να εισαχθούν με δυο τρόπους, πχ `and` ή `/\`.

10.4 Δομές επανάληψης

Οι κυριότερες δομές επανάληψης είναι δυο. Η επανάληψη για όλα τα στοιχεία μιας λίστας με χρήση for-in-repeat και η επανάληψη υπό συνθήκη με χρήση while-repeat. Θα δώσουμε τη χρήση των δομών με παραδείγματα.

Η παρακάτω ρουτίνα δέχεται σαν είσοδο ένα πολυώνυμο $f(x)$ και μια λίστα `list` από αλγεβρικούς αριθμούς και επιστρέφει τους εκείνους τους αριθμούς t για τους οποίους $f(t) > 0$. Αυτό το κάνει ως εξής: για κάθε αριθμό t στη λίστα, αν είναι θετική η τιμή του $f(t)$ τον προσθέτει στη λίστα `p` , η οποία αρχικά είναι κενή. Αφού δοκιμάσει όλα τα στοιχεία της `list` επιστρέφει τη λίστα `p`.

```
PosVal: (POLY FRAC INT, LIST AN) -> LIST AN
PosVal(f,list)==
        p: LIST AN :=[]
        for t in list repeat
                if eval(f,x=t)>0 then
                        p:=concat(p,t)
        return p
```

Παρατηρήστε ότι είναι σημαντική η διατήρηση των σωστών στηλοθετών ώστε να γράφουμε σωστά τη διαδικασία.

Για να επιδείξουμε τη χρήση της while δίνουμε τον Ευκλείδειο αλγόριθμο εύρεσης του ΜΚΔ. Αυτός έχει ως εξής: όσο $g \neq 0$, αντικατέστησε το g με το υπόλοιπο του f δια g και το f με το g. Αυτή η διαδικασία επαναλαμβάνεται έως ότου $g = 0$.

```
Euclid: (POLY FRAC INT, POLY FRAC INT) -> POLY FRAC INT
Euclid(f,g)==
        while g ~=0 repeat
                t:=g
                g:=normalForm(f,[g])
                f:=t
        return f
```

10.5 Αναδρομικές συναρτήσεις

Μια συνάρτηση λέγεται αναδρομική αν μέσα στο σώμα της υπάρχει κλήση της ίδιας συνάρτησης (με διαφορετικά ορίσματα). Στο Axiom δίνεται η δυνατότητα να χρησιμοποιήσουμε αναδρομή. Για παράδειγμα, μπορούμε να γράψουμε ξανά τον αλγόριθμο του Ευκλείδη για την εύρεση του ΜΚΔ με χρήση αναδρομής:

```
EuclidRec: (POLY FRAC INT, POLY FRAC INT)-> POLY FRAC INT
EuclidRec(f,g)==
        if g=0 then
                return f
        return EuclidRec(g, normalForm(f,[g]) )
```

Η ρουτίνα είναι ακριβώς η γνωστή μας σχέση ΜΚΔ(f,g)=ΜΚΔ$(g, f \mod g)$. Παρατηρήστε ότι ο αλγόριθμος τερματίζει όταν $g = 0$ οπότε και επιστρέφει το f. Αυτό είναι το *βήμα διακοπής* του αναδρομικού αλγορίθμου.

11 Ο αλγόριθμος της διαίρεσης

Η παρακάτω συνάρτηση εκτελεί τη διαίρεση σε πολλές μεταβλητές ενός πολυωνύμου f δια μιας λίστας διαιρετών $[d_1, \ldots, d_m]$. Παρατηρήστε ότι δε δηλώνουμε τους τύπους εισόδου/εξόδου της συνάρτησης, επειδή θέλουμε η συνάρτηση να δουλεύει για πολλούς διαφορετικούς τύπους, πχ για πολυώνυμα διατεταγμένα με διάφορες διατάξεις όρων.

```
-- Multivariate Polynomial Division
mpolydiv(f,D)==
        lex:= typeOf(f)
        Q:=[ 0::lex for j in D ]
        r:= 0::lex
        m:= #D
        lms:=[leadingMonomial(D.j) for j in 1..m]
        j:= 1
        flag:= false
        while f ~= 0 repeat
                if j>m then
                        j:= 1
                        if flag=false then
                                t:= leadingMonomial f
                                r:= r + t
                                f:= f - t
                        flag:= false
                if normalForm( leadingMonomial(f),[lms.j])=0 then
                        t   := (leadingMonomial f/lms.j)::lex
                        Q.j := Q.j + t
                        f   := f   - t*D.j
                        flag:= true
                else
                        j:= j+1
        rec:= Record( quotient:LIST lex, remainder: lex)
        return [Q,r]::rec
```

Μπορούμε τώρα να εκτελέσουμε μια διαίρεση (αφού φορτώσουμε τη συνάρτηση στο Axiom). Έστω ότι η συνάρτηση βρίσκεται στο αρχείο `C:\axiom\mpolydiv.input`:

```
→ )read "C:/axiom/mpolydiv.input"
```

```
-- Multivariate Polynomial Division
```

```
→ lex:=DMP([x,y,z], FRAC INT)
```

DistributedMultivariatePolynomial([x,y,z],Fraction Integer) (96) `Type: Domain`

```
→ d1: lex := -2* x^12 + 4* x^2 * y^4
```

$$-2x^{12} + 4x^2y^4 \qquad (97)$$

`Type: DistributedMultivariatePolynomial([x,y,z],Fraction Integer)`

```
→ d2: lex := x^2 - y^11
```

$$x^2 - y^{11} \qquad (98)$$

`Type: DistributedMultivariatePolynomial([x,y,z],Fraction Integer)`

```
→ f :lex := 3*x^14*y^10 + 5*x*y^3 -6*y^7 + 1
```

$$3x^{14}y^{10} + 5xy^3 - 6y^7 + 1 \qquad (99)$$

`Type: DistributedMultivariatePolynomial([x,y,z],Fraction Integer)`

```
→ mpolydiv(f, [d1,d2])
```

```
  Cannot compile conversion for types involving local variables.  In
particular, could not compile the expression involving ::  lex
AXIOM will attempt to step through and interpret the code.
```

$$\left[quotient = \left[-\frac{3}{2}x^2y^{10},\ 6x^2y^{14} + 6y^{25}\right],\ remainder = 5xy^3 + 6y^{36} - 6y^7 + 1\right] \qquad (100)$$

`Type: Record(quotient: List DistributedMultivariatePolynomial([x,y,z],Fraction Integer),remainder: DistributedMultivariatePolynomial([x,y,z],Fraction Integer))`

```
→ mpolydiv(f, [d2,d1])
```

$$\left[quotient = \left[3x^{12}y^{10} + 3x^{10}y^{21} + 3x^8y^{32} + 3x^6y^{43} + 3x^4y^{54} + 3x^2y^{65} + 3y^{76},\ 0\right],\right.$$
$$\left. remainder = 5xy^3 + 3y^{87} - 6y^7 + 1\right] \qquad (101)$$

`Type: Record(quotient: List DistributedMultivariatePolynomial([x,y,z],Fraction Integer), remainder: DistributedMultivariatePolynomial([x,y,z],Fraction Integer))`

Παρατηρήστε ότι το πηλίκο και το υπόλοιπο αλλάζουν όταν αλλάξουμε τη σειρά των διαιρετών. Επαληθεύουμε το αποτέλεσμα της τελευταίας διαίρεσης:

```
→ %.quotient 1 * d2 + %.quotient 2 * d1 + %.remainder
```

$$3x^{14}y^{10} + 5xy^3 - 6y^7 + 1 \qquad (102)$$

`Type: DistributedMultivariatePolynomial([x,y,z],Fraction Integer)`

12 Ο Αλγόριθμος Buchberger

Σημείωση: Οι παρακάτω ρουτίνες δίνονται για εκπαιδευτικούς σκοπούς και ενδέχεται να είναι αρκετά πιο αργές από τις υπάρχουσες εντολές του Axiom για την εύρεση βάσεων Groebner (οι οποίες ενσωματώνουν διάφορες βελτιώσεις στον αλγόριθμο Buchberger).

Η παρακάτω ρουτίνα ακολουθεί τον αλγόριθμο Buchberger και επιστρέφει μια βάση Groebner του F ως εξής: Αρχικά παράγει όλα τα ζεύγη πολυωνύμων του F. Για κάθε ζεύγος, βρίσκει το υπόλοιπο r του $S-$πολυωνύμου δια του F. Αν το υπόλοιπο αυτό δεν είναι μηδενικό, το προσθέτει στο F. Σημειώστε πως όταν συμβεί αυτό στα προς εξέταση ζεύγη προστίθενται όλα τα νέα ζεύγη που περιέχουν το r. Η ρουτίνα τερματίζει όταν όλα τα ζεύγη έχουν ελεγχθεί.

```
buchberger(F)==
   pairs:=[]
   for i in 1..#F-1 repeat
      pairs:=append(pairs, [[F.i,F.j] for j in i+1..#F] )
   while pairs ~= [] repeat
      p     := pairs.1
      pairs:= rest(pairs)
      r     := normalForm( S(p.1,p.2) , F )
      if r ~=0 then
         F     := concat( F, r )
         pairs:= append( pairs, [[F.i, r] for i in 1..#F-1] )
   return F

S(f,g)==
   lex:= typeOf(f)
   ekp:= lcm(leadingMonomial(f)/leadingCoefficient(f),_
             leadingMonomial(g)/leadingCoefficient(g))
   return ((ekp/leadingMonomial f)*f - (ekp/leadingMonomial g)*g )::lex
```

Η βάση που επιστρέφει η παραπάνω ρουτίνα είναι απαραίτητα ένα υπερσύνολο του αρχικού F.

Μπορούμε από μια οποιαδήποτε βάση να φτάσουμε σε μια *ελάχιστη* βάση ακολουθώντας τον ορισμό. Ένας αλγόριθμος είναι: Ξεκινάμε μια ένα κενό σύνολο M. Αφαιρούμε από την αρχική βάση G ένα ένα τα πολυώνυμα. Αν ο ΜΟ του πολυωνύμου δε διαιρείται με τους μεγιστοβάθμιους όρους στο G και στο M, τότε προσθέτουμε το πολυώνυμο στο M (διαφορετικά το «πετάμε»). Στο τέλος (όταν το G μείνει κενό) το M είναι μια ελάχιστη βάση, εφόσον τα πολυώνυμά της μετατραπούν σε μονικά.

```
minimal(G)==
   lmG:= [leadingMonomial G.i for i in 1..#G]
   M   := []
   while lmG~=[] repeat
      if normalForm(lmG.1, rest(lmG) )~=0 and _
      normalForm(lmG.1,[leadingMonomial m for m in M])~=0 then
          M:=concat(M,G.1)
      G:= rest( G )
      lmG:= rest(lmG)
   M:= [M.i/leadingCoefficient M.i for i in 1..#M]
   return M
```

Για να φτάσουμε στην *ανηγμένη* βάση Groebner αντικαθιστούμε κάθε πολυώνυμο μιας ελάχιστης βάσης με το υπόλοιπο της διαίρεσης αυτού δια όλων των άλλων πολυωνύμων της βάσης. Η παρακάτω ρουτίνα απαιτεί το G να είναι μια ελάχιστη βάση:

```
reduced(G)==
   for i in 1..#G repeat
      G.i := normalForm(G.i, [f for f in G|f~=G.i])
   return G
```

Έστω ότι όλες οι παραπάνω ρουτίνες είναι στο αρχείο `C:\axiom\grobner.input`:

```
→ )read "C:/axiom/grobner.input"
```

```
-- Buchberger's Algorithm (input:  any finite set of polynomials)
-- S-polynomial
-- Minimal Basis (input:  any Grobner basis)
-- Reduced Basis (input:  a minimal Grobner basis)
```

Θα εφαρμόσουμε τα παραπάνω στο σύνολο πολυωνύμων:

```
→ F:= [ x^2*y - 1, x*y^2 - x , x^2+y^2 -y-1 ] ::LIST DMP( [x,y], FRAC INT )
```

$$\left[x^2y-1,\ \ xy^2-x,\ \ x^2+y^2-y-1\right] \tag{103}$$

Type: List DistributedMultivariatePolynomial([x,y],Fraction Integer)

τα οποία έχουμε διατάξει με τη λεξικογραφική διάταξη $x > y$. Ξεκινάμε με τον αλγόριθμο Buchberger:

```
→ G:= buchberger( F )
```

$$\left[x^2y-1,\ \ xy^2-x,\ \ x^2+y^2-y-1,\ \ -y^2+1\right] \tag{104}$$

Type: List DistributedMultivariatePolynomial([x,y],Fraction Integer)

Άρα η G είναι μια βάση Groebner του $\langle F\rangle$ (η βάση αποτελείται από τέσσερα πολυώνυμα, όμως δεν είναι όλα μονικά και ο μεγιστοβάθμιος του τρίτου διαιρεί το μεγιστοβάθμιο του πρώτου). Αντικαθιστούμε αυτή τη βάση με μια ελάχιστη βάση Groebner:

```
→ G:= minimal( G )
```

$$\left[x^2+y^2-y-1,\ \ y^2-1\right] \tag{105}$$

Type: List DistributedMultivariatePolynomial([x,y],Fraction Integer)

Άρα μια ελάχιστη βάση αποτελείται από δυο πολυώνυμα (σε αυτή τη βάση όμως υπάρχει όρος του πρώτου πολυωνύμου που διαιρείται με το μεγιστοβάθμιο του δεύτερου). Τέλος αντικαθιστούμε την ελάχιστη βάση με την ανηγμένη βάση Groebner:

```
→ G:= reduced( G )
```

$$\left[x^2-y,\ \ y^2-1\right] \tag{106}$$

Type: List DistributedMultivariatePolynomial([x,y],Fraction Integer)

Ακολουθώντας τώρα τη διαδικασία της ενότητας 8 φτάνουμε στο σύνολο λύσεων του αρχικού συστήματος, $\Lambda = \{\ (1,1),\ (-1,1),\ (i,-1),\ (-i,-1)\ \}$.

13 Παραδείγματα

- Επίλυση πολυωνύμου δευτέρου και τρίτου βαθμού

```
→ radicalSolve( a*x^2 + b*x + c , x)
```

$$\left[x=\frac{-\sqrt{-4ac+b^2}-b}{2a},\ x=\frac{\sqrt{-4ac+b^2}-b}{2a}\right] \tag{107}$$

Type: List Equation Expression Integer

```
→ radicalSolve( a*x^3+b*x^2+c*x+d , x)
```

$$\left[x=\frac{\left(-9a^2\sqrt{-3}+9a^2\right)\sqrt[3]{\frac{54a^3\sqrt{\frac{27a^2d^2+\left(-18abc+4b^3\right)d+4ac^3-b^2c^2}{108a^4}}-27a^2d+9abc-2b^3}{54a^3}}^{2}}{\left(9a^2\sqrt{-3}+9a^2\right)\sqrt[3]{\frac{54a^3\sqrt{\frac{27a^2d^2+\left(-18abc+4b^3\right)d+4ac^3-b^2c^2}{108a^4}}-27a^2d+9abc-2b^3}{54a^3}}}\right.$$

$$+\frac{\left(-3ab\sqrt{-3}-3ab\right)\sqrt[3]{\frac{54a^3\sqrt{\frac{27a^2d^2+\left(-18abc+4b^3\right)d+4ac^3-b^2c^2}{108a^4}}-27a^2d+9abc-2b^3}{54a^3}}+6ac-2b^2}{\left(9a^2\sqrt{-3}+9a^2\right)\sqrt[3]{\frac{54a^3\sqrt{\frac{27a^2d^2+\left(-18abc+4b^3\right)d+4ac^3-b^2c^2}{108a^4}}-27a^2d+9abc-2b^3}{54a^3}}},$$

$$x=\frac{\left(-9a^2\sqrt{-3}-9a^2\right)\sqrt[3]{\frac{54a^3\sqrt{\frac{27a^2d^2+\left(-18abc+4b^3\right)d+4ac^3-b^2c^2}{108a^4}}-27a^2d+9abc-2b^3}{54a^3}}^{2}}{\left(9a^2\sqrt{-3}-9a^2\right)\sqrt[3]{\frac{54a^3\sqrt{\frac{27a^2d^2+\left(-18abc+4b^3\right)d+4ac^3-b^2c^2}{108a^4}}-27a^2d+9abc-2b^3}{54a^3}}}$$

$$+\frac{\left(-3ab\sqrt{-3}+3ab\right)\sqrt[3]{\frac{54a^3\sqrt{\frac{27a^2d^2+\left(-18abc+4b^3\right)d+4ac^3-b^2c^2}{108a^4}}-27a^2d+9abc-2b^3}{54a^3}}-6ac+2b^2}{\left(9a^2\sqrt{-3}-9a^2\right)\sqrt[3]{\frac{54a^3\sqrt{\frac{27a^2d^2+\left(-18abc+4b^3\right)d+4ac^3-b^2c^2}{108a^4}}-27a^2d+9abc-2b^3}{54a^3}}},$$

$$x=\frac{9a^2\sqrt[3]{\frac{54a^3\sqrt{\frac{27a^2d^2+\left(-18abc+4b^3\right)d+4ac^3-b^2c^2}{108a^4}}-27a^2d+9abc-2b^3}{54a^3}}^{2}}{9a^2\sqrt[3]{\frac{54a^3\sqrt{\frac{27a^2d^2+\left(-18abc+4b^3\right)d+4ac^3-b^2c^2}{108a^4}}-27a^2d+9abc-2b^3}{54a^3}}}$$

$$\left.-\frac{3ab\sqrt[3]{\frac{54a^3\sqrt{\frac{27a^2d^2+\left(-18abc+4b^3\right)d+4ac^3-b^2c^2}{108a^4}}-27a^2d+9abc-2b^3}{54a^3}}-3ac+b^2}{9a^2\sqrt[3]{\frac{54a^3\sqrt{\frac{27a^2d^2+\left(-18abc+4b^3\right)d+4ac^3-b^2c^2}{108a^4}}-27a^2d+9abc-2b^3}{54a^3}}}\right] \tag{108}$$

Type: List Equation Expression Integer

• Εύρεση σημείων τομής του κύκλου και της παραβολής

```
→ complexSolve( [x^2+y^2-1, y-x^2], .00001 )
```

$$[[y = 0.61803178891204879619, \;\; x = 0.78614997863769531 25],$$
$$[y = 0.61803178891204879619, \;\; x = -0.7861499786376953125],$$
$$[y = -1.6180405976629117504, \;\; x = -1.272022247314453125i],$$
$$[y = -1.6180405976629117504, \;\; x = 1.272022247314453125i]] \qquad (109)$$

Type: List List Equation Polynomial Complex Float