ALGUNAS APLICACIONES DE LAS BASES DE GROEBNER

Eugenio Roanes Lozano
eroanes@mat.ucm.es

Universidad Complutense de Madrid
Facultad de Educación y Facultad de CC. Matemáticas
Departamento de Álgebra

Talleres Divulgativos "Matemáticas en Acción 2009"
Universidad de Cantabria, 25-XI-2009

Parte I: Porqué trabajar con un sistema de cómputo algebraico

¿Qué es el cálculo simbólico?

- álgebra computacional (Computer Algebra)
- computación simbólica
- computación algebraica
- manipulación simbólica
- cálculo formal (Calcul Formel)

• Una definición: R. Loos: "Computer Algebra is that part of Computer Science which designs, analyzes, implements and applies algebraic algorithms".

• Los sistemas de cómputo algebraico (Computer Algebra Systems, CAS) tienen dos características diferenciadoras:

- trabajan por defecto en aritmética exacta,

- pueden manejar variables no asignadas.

• Los CAS usan por defecto su propia aritmética exacta, mientras que los lenguajes habituales usan la aritmética en coma flotante interna (del procesador).

• En coma flotante (CF) sólo se considera un número fijo de dígitos. En CF el tamaño del valor absoluto de los números tiene cota superior e inferior.

• Sorprendentemente, suele ser una aritmética modular de módulo grande.

• En arítmética exacta (AE) los racionales se tratan como fracciones.

• En AE irracionales como π, e, $\sqrt{2}$,... Se tratan como símbolos con ciertas propiedades (como $\sqrt{2}^{\,2} = 2$).

• Ventajas AE: exactitud.

• Desventajas AE: lentitud. Por:

- no "built-in"

- crecimiento de expresiones: pensemos por ejemplo en tratar de simplificar $(2+\pi+e)^2 - \pi^2$; es más cómodo "hacer cuentas" con 4 decimales.

- Problemas fundamentales de la CF:

  - Redondeo de números muy grandes (incluso en $\mathbb{Z}$):

Número	Representación	Resultado
234	234	Correcto
3450000000	$3.45 * 10^9$	Correcto
3450000001	$3.45 * 10^9$	Inexacto

Deja de ser una biyección: $\mathbb{Z} \longleftrightarrow \{representaciones\ de\ enteros\}$!!!

- Truncamiento de la representación decimal (en $\mathbb{Q}$):

Número	Representación	Resultado
0.000000003	$3 \cdot 10^{-9}$	Correcto
1.000000003	1.0	Inexacto

- Aproximación de fracciones (en $\mathbb{Q}$):

Número	Representación	Resultado
$1/3$	0.333333333	Inexacto

- Aproximación de irracionales: $\sqrt{2}$, π, e,... (en $\mathbb{R}$):

Número	Representación	Resultado
$\sqrt{2}$	1.414213562	Inexacto

- Consecuencias a que puede dar lugar una inexactitud:

1) a un error al tomar una decisión

2) a un error muy grande

   i) al realizar una iteración de aproximaciones

   ii) al realizar futuras operaciones.

• Observación: En la representación interna, como ocurre con las calculadoras, puede haber más dígitos significativos que los que ve el usuario y además se hacen "arreglos" como:

$$(\sqrt{2})^2 = 2.0$$

mientras que el resultado que muestra en pantalla, al cuadrado, es:

$$1.999999999$$

Se puede descubrir el engaño con expresiones como:

$$\sqrt{3} \cdot \sqrt{2} - \sqrt{6}$$

Veamos, por ejemplo, con la calculadora de MS-Windows,

$$13^2 - (12^2 + 5^2)$$

o bien

$$\sqrt{6} - (\sqrt{3} * \sqrt{2})$$

• Algo sorprendente: En un lenguaje usual (que usa aritmética en CF), un programa "matemáticamente" correcto y correctamente escrito puede dar lugar por un redondeo o aproximación a un resultado totalmente erróneo, ni siquiera aproximado.

• Veremos a continuación algunos ejemplos de situaciones en las que, si atendemos ciegamente a lo que responde el ordenador, llegaremos a conclusiones erróneas.

Ejemplo 1: Existen números consecutivos y estrictamente mayores que 1 que son uno múltiplo del otro (!!!)

Programa en BASIC que estudia si un entero es múltiplo de otro.

```
10 REM "PROG. QUE VERIF. SI UN NUM. ES MULTIPLO DE OTRO"
20 PRINT "ESCRIBE EL PRIMER NUMERO"
30 INPUT A
40 PRINT "ESCRIBE EL SEGUNDO NUMERO"
50 INPUT B
55 PRINT ""
60 IF INT(A/B) = (A/B) THEN GOTO 90
70 PRINT "NO ES MULTIPLO"
80 GOTO 100
90 PRINT "ES MULTIPLO"
100 END
```

Veamos sus respuestas en varios ejemplos:

$$200{,}20 \longrightarrow \text{ES MULTIPLO}$$
$$23{,}34 \longrightarrow \text{NO ES MULTIPLO}$$

Hasta aquí todo bien...

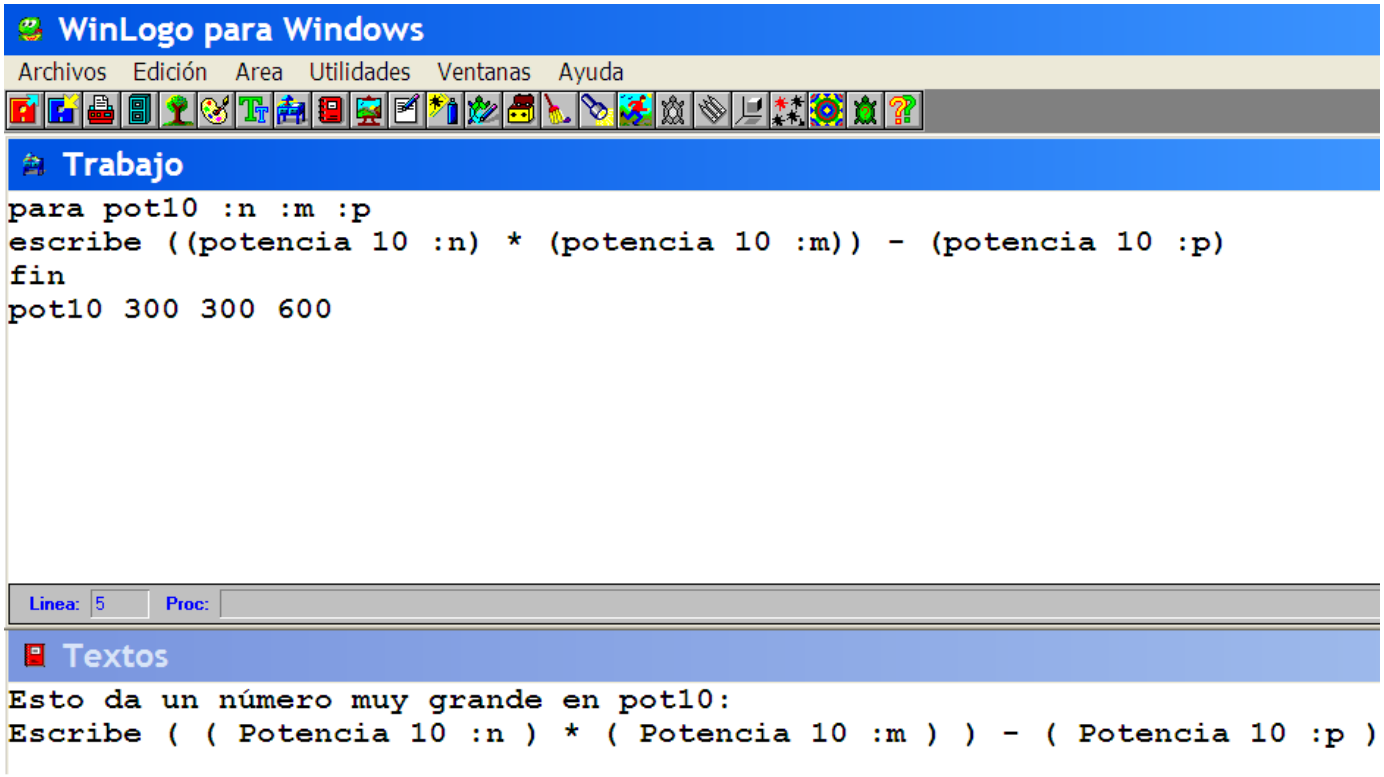
Veamos sus respuestas en varios ejemplos:

$$200{,}20 \longrightarrow \text{ES MULTIPLO}$$
$$23{,}34 \longrightarrow \text{NO ES MULTIPLO}$$
$$300000001{,}300000000 \longrightarrow \text{ES MULTIPLO (!!!)}$$

El algoritmo era correcto y estaba bien escrito en GW-BASIC. La razón es que los dos últimos números han sido representados como 3E+08 (y no ha habido ningún aviso del paso a notación científica).

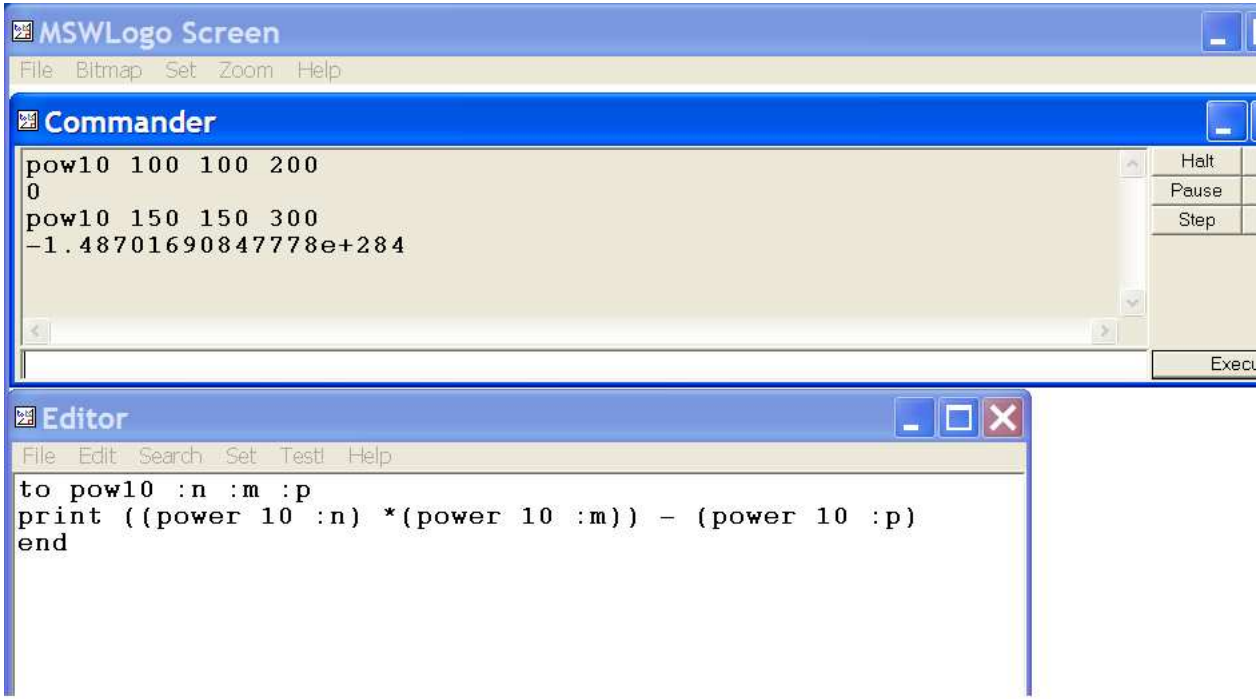
Ejemplo 2: $a^n \cdot a^m \neq a^{n+m}$ (!!!)

Según WinLogo para Windows, $10^{300} \cdot 10^{300} - 10^{600}$ es un número muy grande...



Ejemplo 3: $a^n \cdot a^m \neq a^{n+m}$ (!!!)

Pero MSWLogo incluso precisa el valor de $10^{150} \cdot 10^{150} - 10^{300}$



Ejemplo 4: Un sistema homogéneo de dos ecuaciones, una de ellas "múltiplo" de la otra, que tiene solución única (!!!)

Programa en LCSI-LOGO que estudia el numero de soluciones de un sistema homogéneo de dos ecuaciones y dos incógnitas:

```
TO SIST :A11 :A12 :A21 :A22
PRINT [DISCUSION (SISTEMA HOMOGENEO]
PRINT [DE DOS ECUACIONES LINEALES)]
PRINT []
MAKE "DET (:A11 * :A22) - (:A21 * :A12)
IF :DET = 0
   [PRINT [EL SISTEMA TIENE INFINITAS SOLUCIONES]]
   [PRINT [EL SISTEMA TIENE SOLUCION UNICA]]
END
```

Apliquémoslo:

$$\begin{aligned} x + y &= 0 \\ 2 \cdot x + 2 \cdot y &= 0 \end{aligned}$$

Responde: tiene infinitas soluciones (correcto).

Apliquémoslo:

$$\begin{aligned} x + y &= 0 \\ 2 \cdot x + 2 \cdot y &= 0 \end{aligned}$$

Contesta: tiene infinitas soluciones (correcto).

Pero para:

$$\begin{aligned} \frac{1}{3} \cdot x + y &= 0 \\ x + 3 \cdot y &= 0 \end{aligned}$$

Contesta: tiene solución única (!!!):

(la razón es que $3 \cdot \frac{1}{3} - 1$ no ha resultado ser 0 ...)

Ejemplo 5: Un polinomio de segundo grado con coeficientes racionales que factoriza como $3 \cdot (x - \frac{1}{3})^2$ tiene dos raíces reales distintas (!!!)

Programa en Turbo-Pascal 7.0 que estudia el numero de raíces de una ecuación de segundo grado con coeficientes racionales.

```
uses
    Crt;
var
   an,ad,bn,bd,cn,cd : integer;
   a,b,c,disc : Real;
begin
     ClrScr;
     writeln;
     writeln;
```

```
write('Escribe el numerador del coef. de x cuadrado: ');
readln(an);
write('Escribe el denominador del coef. de x cuadr.: ');
readln(ad);
a := an / ad;
write('Escribe el numerador del coef. de x: ');
readln(bn);
write('Escribe el denominador del coef. de x: ');
readln(bd);
b := bn / bd;
write('Escribe el numerador del termino independiente: ');
readln(cn);
write('Escribe el denominador del termino independiente: ');
readln(cd);
c := cn / cd;
writeln;
disc := b*b - 4*a*c;
```

```
      if disc = 0
         then write('  Tiene una raiz doble')
         else if disc < 0
              then write('No tiene raices reales')
              else write('Tiene dos raices reales distintas');
      readln;
      writeln;
      write('El discriminante que ha utilizado ha sido: ');
      writeln(disc);
      readln
end.
```

Para: $x^2 - 2x + 1 = (x-1)^2$ contesta correctamente:

Tiene una raíz doble.

Pero para: $3x^2 - 2x + \frac{1}{3} = 3 \cdot (x - \frac{1}{3})^2$ contesta:

Tiene dos raíces reales distintas (!)

(utiliza como discriminante $7.27595761E-12$, no 0).

Ejemplo 6: Una matriz 2×2 con determinante no nulo y rango 1 (!!!)

Comprobemos en una versión obsoleta (1987) de MatLab el determinante y rango de cierta matriz 2×2...

```
>> A = [1/7,1;1,7]
A =
    0.1429    1.0000
    1.0000    7.0000

>> rank(A)
ans=
    1

>> det(A)
ans=
    -5.5511e-017
```

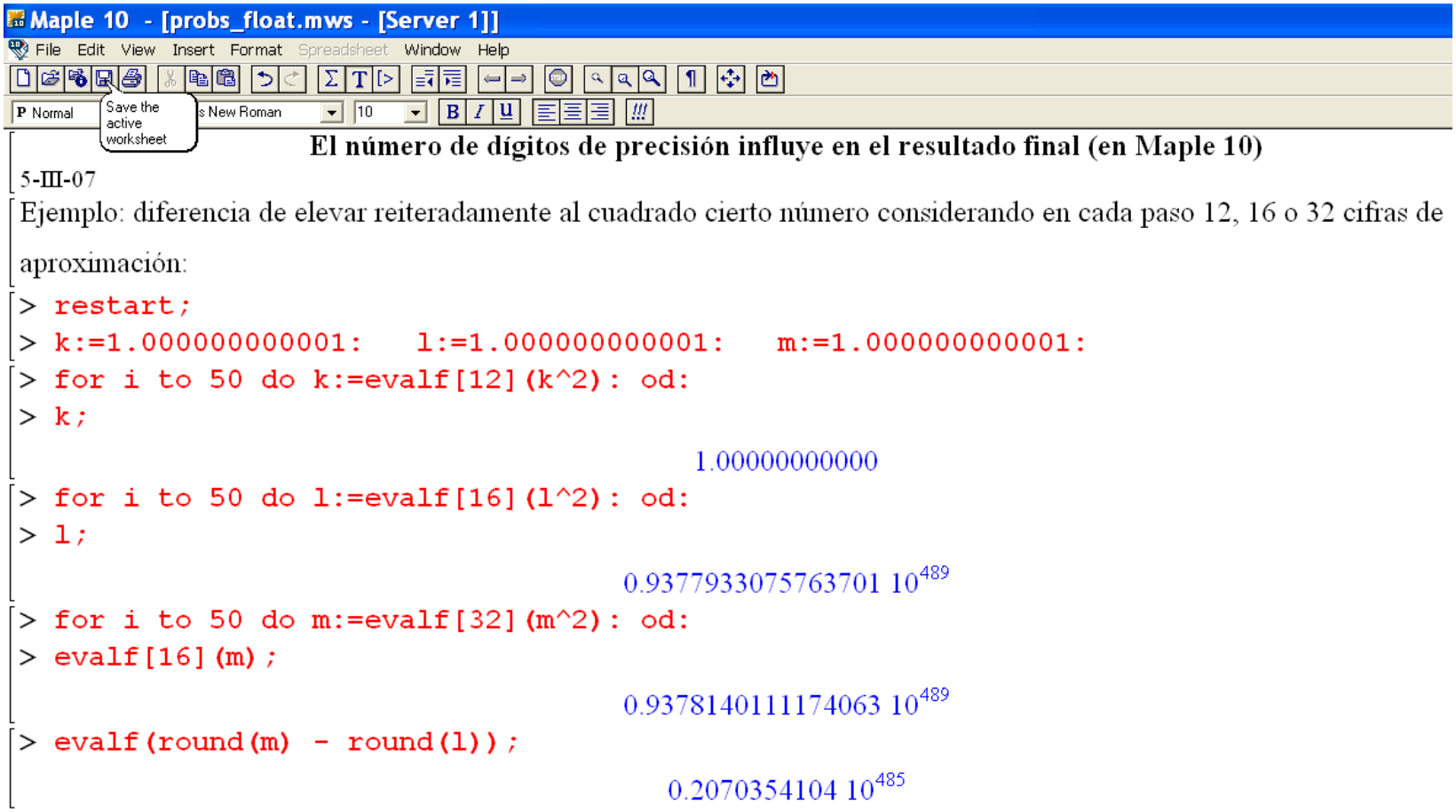
Por último, si tratamos de calcular la inversa de A: `inv(A)`, la calcula (!) aunque A no es invertible, pero al menos avisa de que la matriz es muy próxima a una singular, y el resultado podría ser inexacto.

Esto se ha corregido en las versiones modernas de Matlab.

De hecho las versiones modernas de Matlab pueden llamar al núcleo ("kernel") de Maple para realizar cálculos simbólicos o exactos (para ello hay que adquirir la "Symbolic Toolbox").

Ejemplo 7: Sorprendentemente, el resultado de un cálculo puede variar mucho dependiendo del número de dígitos de aproximación que se tomen.

Cuando se habla de una precisión de n dígitos nos referimos a la precisión en CADA PASO, no al número de dígitos exactos en el número final (sobre el que, directamente, no se puede afirmar nada).


Maple 10 - [probs_float.mws - [Server 1]]
File Edit View Insert Format Spreadsheet Window Help
Save the active worksheet
P Normal
s New Roman
10
El número de dígitos de precisión influye en el resultado final (en Maple 10)
5-III-07
Ejemplo: diferencia de elevar reiteradamente al cuadrado cierto número considerando en cada paso 12, 16 o 32 cifras de aproximación:
> restart;
> k:=1.000000000001: l:=1.000000000001: m:=1.000000000001:
> for i to 50 do k:=evalf[12](k^2): od:
> k;
1.00000000000
> for i to 50 do l:=evalf[16](l^2): od:
> l;
0.9377933075763701 10^489
> for i to 50 do m:=evalf[32](m^2): od:
> evalf[16](m);
0.9378140111174063 10^489
> evalf(round(m) - round(l));
0.2070354104 10^485

MANIPULACIÓN DE VARIABLES NO ASIGNADAS

• Los CAS pueden trabajar de un modo cualitativamente distinto a los lenguajes usuales. En cualquier lenguaje podemos calcular $x + y$ para cualesquiera valores de x e y, pero un CAS puede hacer cálculos como:

$$(x+y)^2 - (x-y)^2 = 4xy$$

sin especificar valores para x e y, esto es, sin asignarles valores numéricos.

• Ello permite:

- inmediatamente, manipular polinomios
- muy sencillamente, calcular la función derivada de una función dada
- calcular primitivas
- ...

Bibliografía:

E. Roanes Lozano: "Precisión indefinida" y matemática elemental. *Bol. Soc. "Puig Adam"* **31** (1992) 33-52.

Algunas aplicaciones de las Bases de Groebner

Eugenio Roanes Lozano
Universidad de Cantabria, 25-XI-2009

--- o o O o o ---

PARTE II: Ideales y Bases de Groebner

Ideales y GB

Subanilllo tal que el producto de un elemento del ideal por un elemento del anillo pertenece al ideal.

Subanillo no ideal: Z de Q, Q de R, R de C

Subanillo y además ideal: 3Z de Z

Los ideales de A[x,y,z,..,t] son las combinaciones lineales algebraicas de un conjunto de polinomios (al que se denomina "una base del ideal").

Las operaciones que se pueden llevar a cabo en un sistema algebraico (algo ánalogo al método de Gauss para sistemas lineales) son exactamente las que se pueden realizar con los elementos de un ideal (combinaciones lineales algebraicas).

Fijado como ordenamos los monomios (grado total, lexicográfico puro,...) y el orden entre las variables, la GB permiten identificar exactamente al ideal.

Una variedad algebraica está constituida por los "ceros" del ideal (conjunto de raíces comunes a todos los polinomios).

```
> with(Groebner):   with(plots):
```

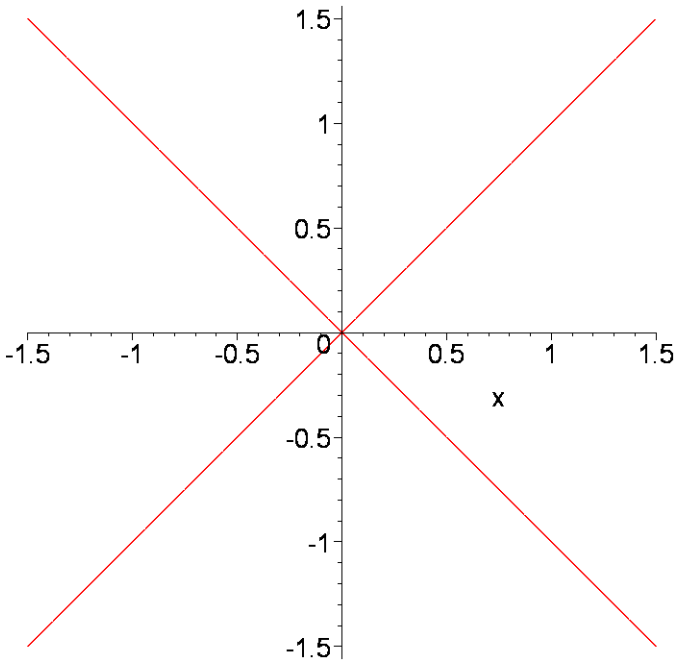
Algunos ejemplos de GB (caso lineal, no haría falta realmente usar GB)

Ejemplo 1: (en estos primeros son ecuaciones lineales, no haría falta usar GB)

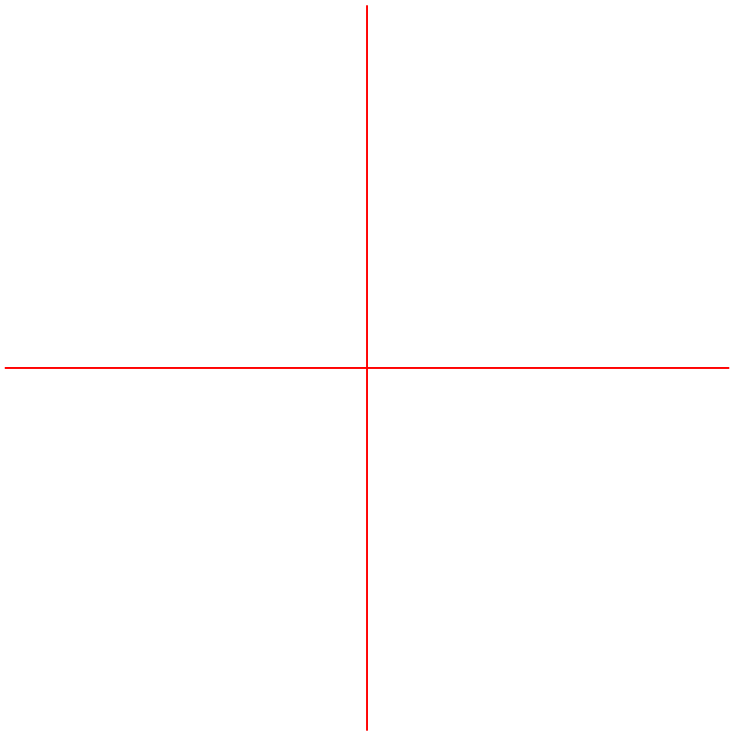
```
> Basis( [x+y,x-y] , plex(y,x,z));
```

$$[x, y]$$

```
> plot({-x,x},x=-1.5..1.5,color=red,scaling=constrained,thickne
  ss=2);
```




```
> implicitplot({x=0,y=0},x=-1.5..1.5,y=-1.5..1.5,scaling=constr
  ained,axes=none,thickness=2);
```

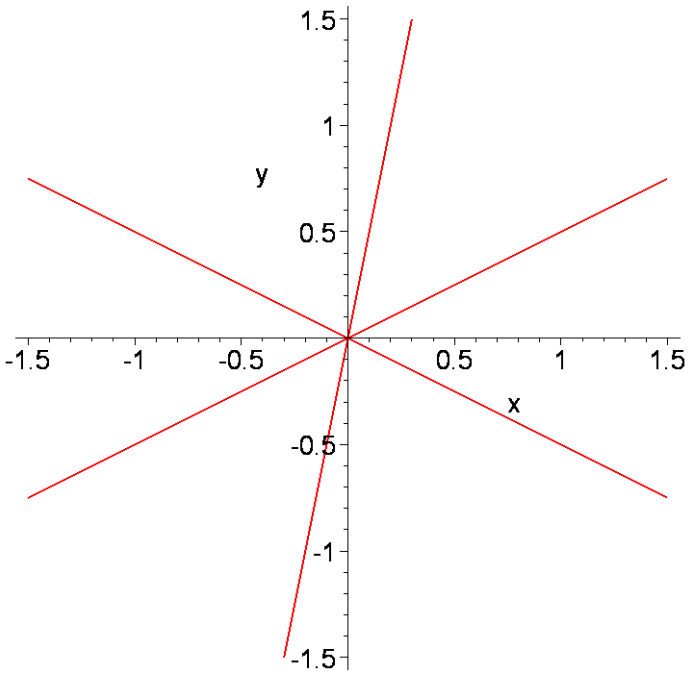


Ejemplo 2:

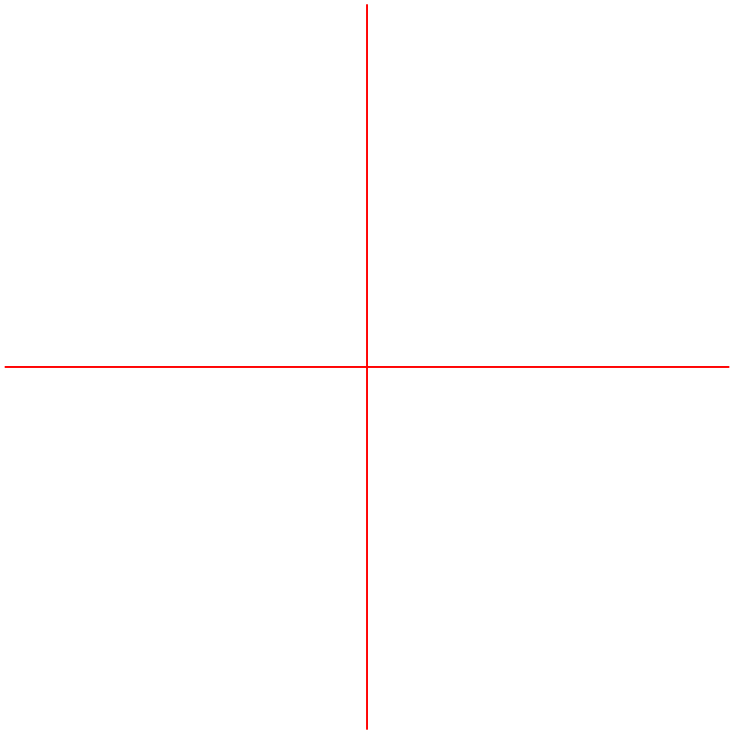
```
> Basis( [x+2*y,3*x-6*y,5*x-y] , plex(y,x,z) );
```

$$[x, y]$$

```
> implicitplot({x+2*y=0,3*x-6*y=0,5*x-y=0},x=-1.5..1.5,y=-1.5..
  1.5,scaling=constrained,thickness=2);
```




```
> implicitplot({x=0,y=0},x=-1.5..1.5,y=-1.5..1.5,scaling=constr
  ained,axes=none,thickness=2);
```



Otros ejemplos de GB

Ejemplo 3: ideales degenerados

```
> Basis( [0] , plex(y,x,z));
```

$$[\,]$$

```
> Basis( [x+1,x+y,y] , plex(y,x,z));
```

$$[1]$$

Ejemplo 4: Distintas bases pueden generar el mismo ideal

```
> Basis( [z-x^2-y^2,z-1] , plex(y,x,z));
```

$$[z-1, y^2-1+x^2]$$

```
> Basis( [z-x^2-y^2,z+x^2+y^2-2] , plex(y,x,z));
```

$$[z-1, y^2-1+x^2]$$

```
> Basis( [x^2+y^2+z^2-2,z-1] , plex(y,x,z));
```

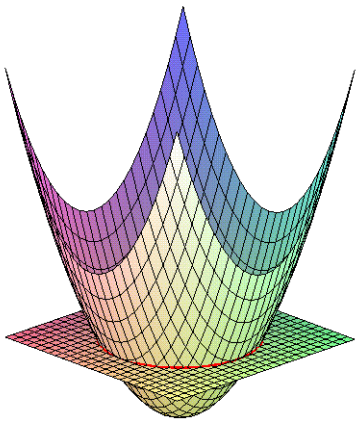
$$[z-1, y^2-1+x^2]$$

```
> Basis( [x^2+y^2+z^2-2,z-x^2-y^2,z+x^2+y^2-2] , plex(y,x,z));
```

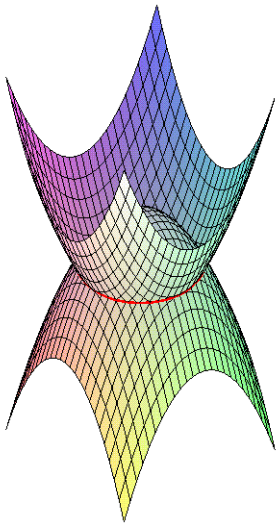
$$[z-1, y^2-1+x^2]$$

Veamos su interpretación geométrica:

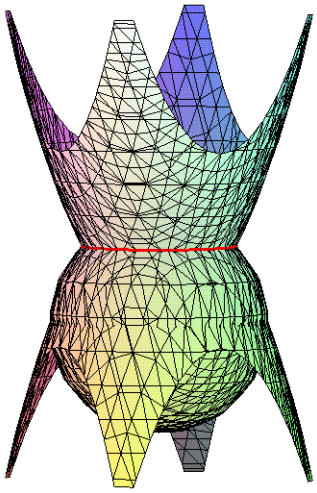
```
> d1:=plot3d([x^2+y^2,1],x=-1.5..1.5,y=-1.5..1.5,scaling=constr
  ained):
> c1:=spacecurve([t,sqrt(1-t^2),1],t=-1..1,color=red,thickness=
  3):
> c2:=spacecurve([t,-sqrt(1-t^2),1],t=-1..1,color=red,thickness
  =3):
> display(d1,c1,c2);
```



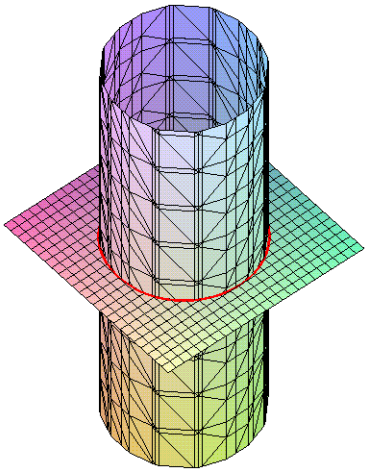
```
> d2:=plot3d([x^2+y^2,-x^2-y^2+2],x=-1.5..1.5,y=-1.5..1.5,scali
  ng=constrained):
> display(d2,c1,c2);
```



```
> d3:=implicitplot3d(x^2+y^2+z^2-2,x=-1.5..1.5,y=-1.5..1.5,z=-2
  ..4,numpoints=2000):
> d4:=implicitplot3d(z-x^2-y^2,x=-1.5..1.5,y=-1.5..1.5,z=-2..4,
  numpoints=2000):
> d5:=implicitplot3d(z+x^2+y^2-2,x=-1.5..1.5,y=-1.5..1.5,z=-2..
  4,numpoints=2000):
> display(d3,d4,d5,c1,c2,scaling=constrained);
```



```
> d1:=implicitplot3d(y^2+x^2-1,x=-1.5..1.5,y=-1.5..1.5,z=-2..4)
  :
> d2:=plot3d(1,x=-1.5..1.5,y=-1.5..1.5):
> display(d1,d2,c1,c2,scaling=constrained);
```



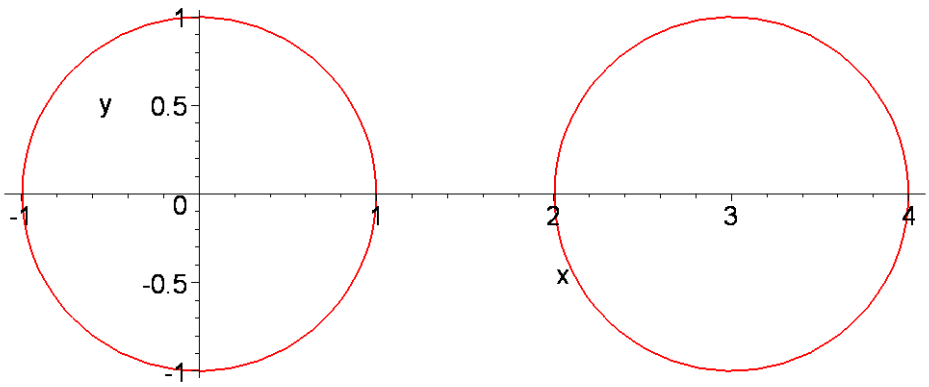
Algunas observaciones sobre GB

Ejemplo 5: El que haya o no soluciones es en C (cuerpo algebraicamente cerrrado):

```
> Basis( [x^2+y^2-1,(x-3)^2+y^2-1], plex(x,y));
```

$$[4\,y^2+5,\,2\,x-3]$$

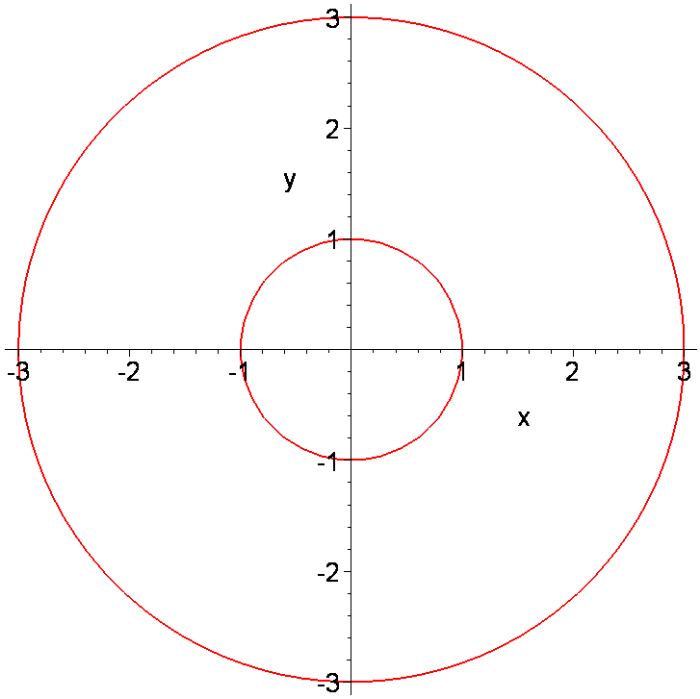
```
> implicitplot({x^2+y^2-1,(x-3)^2+y^2-1},x=-2..5,y=-1.5..1.5,sc
  aling=constrained,thickness=2,numpoints=2000);
```




```
> Basis( [x^2+y^2-1,x^2+y^2-9], plex(x,y));
```

$$[1]$$

```
> implicitplot({x^2+y^2-1,x^2+y^2-9},x=-4..4,y=-4..4,scaling=co
  nstrained,thickness=2,numpoints=2000);
```



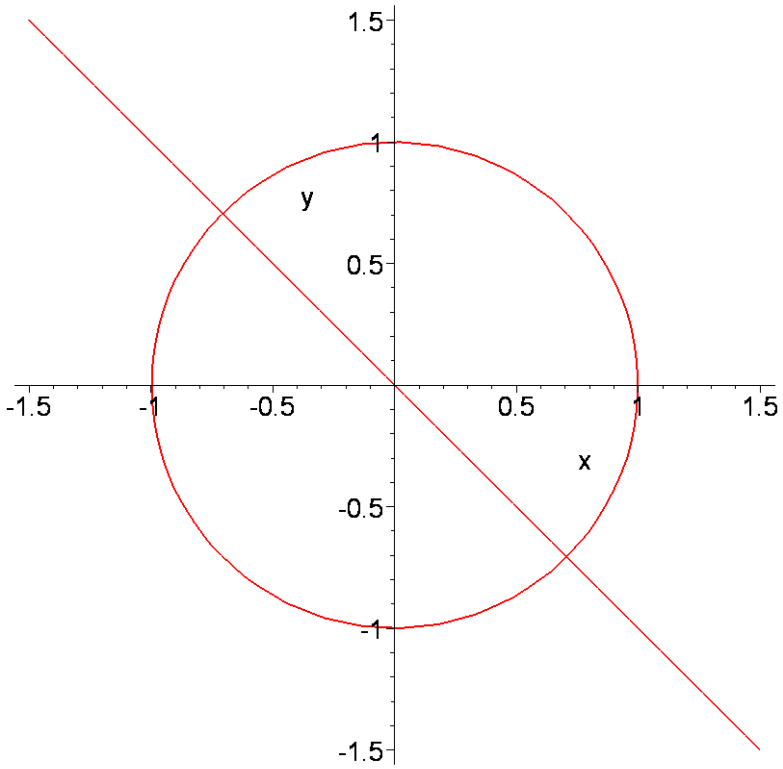
Ejemplo 6: el orden de las variables importa:

```
> Basis( [x^2+y^2-1,x+y] , plex(y,x));
```

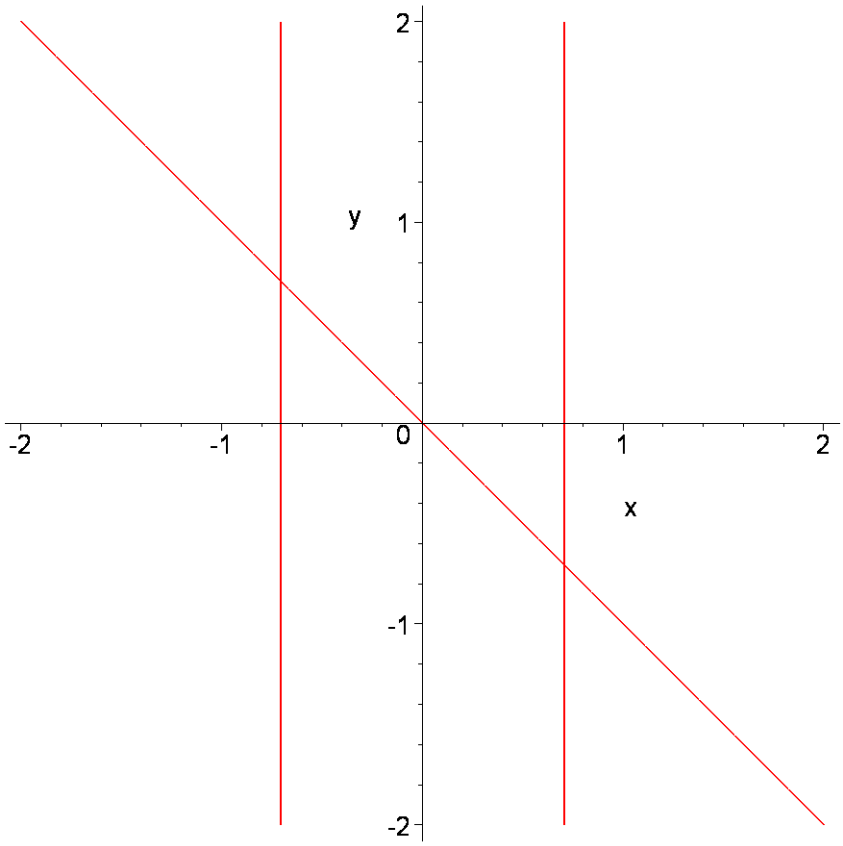
$$[2x^2 - 1, x + y]$$

```
> implicitplot({x^2+y^2-1,x+y},x=-2..5,y=-1.5..1.5,scaling=cons
```

```
trained,thickness=2,numpoints=2000);
```



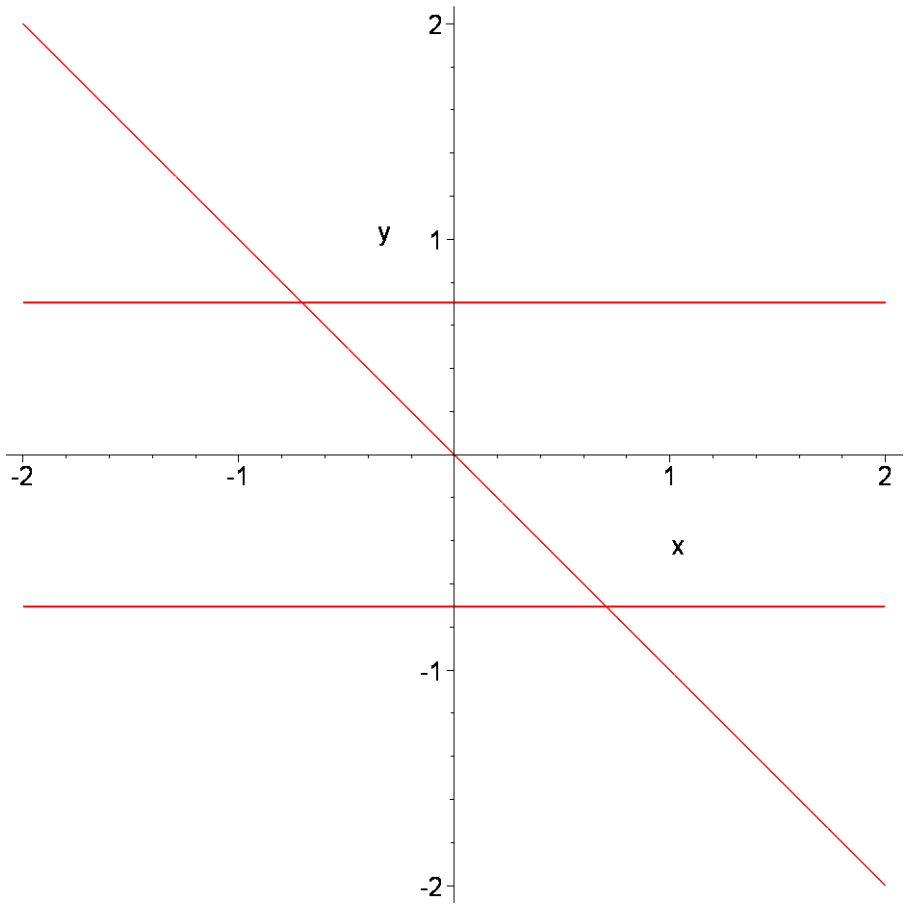

```
> implicitplot({2*x^2-1,
x+y},x=-2..2,y=-2..2,scaling=constrained,thickness=2,numpoint
s=2000);
```




```
> Basis( [x^2+y^2-1,x+y] ,plex(x,y));
```

$$[-1+2y^2, x+y]$$

```
> implicitplot({-1+2*y^2,
  x+y},x=-2..2,y=-2..2,scaling=constrained,thickness=2,numpoint
  s=2000);
```



Ejemplo 7: el orden de los monomios importa (y las GB pueden tener incluso distinto número de elementos)

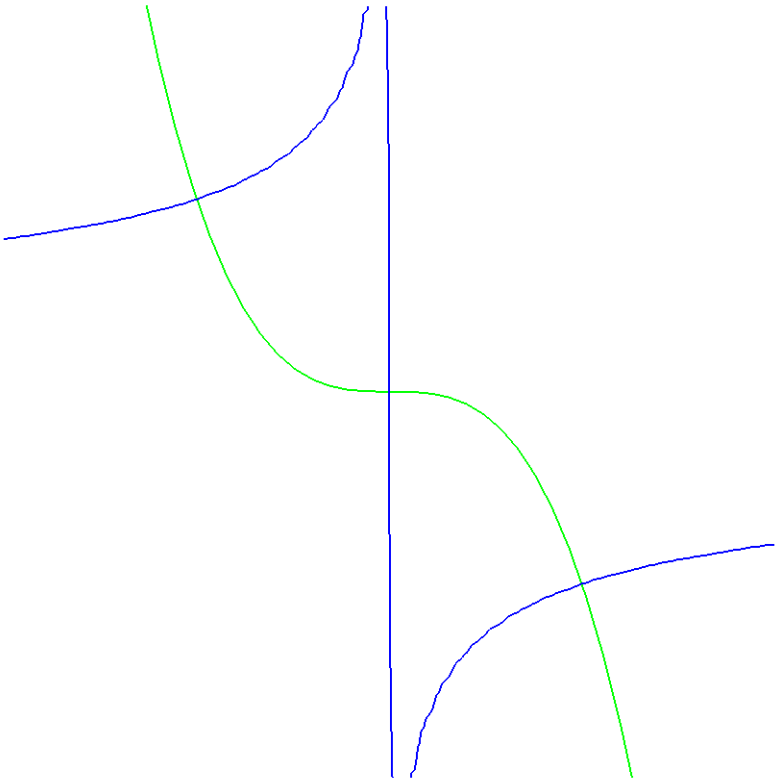
```
> Basis( [x^2*y^3+x,x^3+y], plex(y,x));
```

$$[-x+x^{11}, x^3+y]$$

```
> Basis( [x^2*y^3+x,x^3+y], tdeg(y,x));
```

$$[x^3+y, y^4-x^2, x^2y^3+x]$$

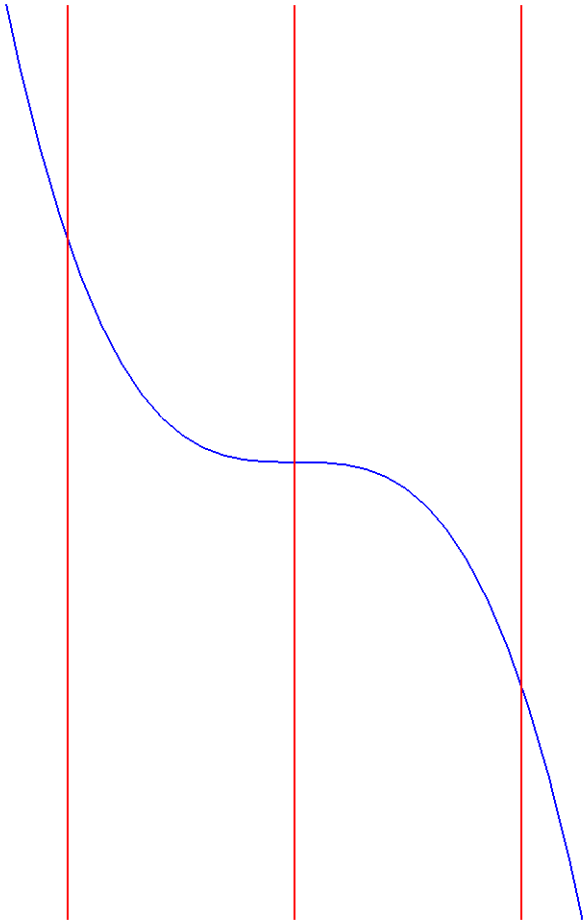
```
> d1:=implicitplot(x^2*y^3+x,x=-2..2,y=-2..2,signchange=true,sc
  aling=constrained,axes=none,thickness=2,numpoints=2000,color=
  blue):
> d2:=implicitplot(x^3+y,x=-2..2,y=-2..2,signchange=true,scalin
  g=constrained,axes=none,thickness=2,numpoints=2000,color=gree
  n):
> display(d1,d2);
```



```
> factor(x^2*y^3+x);
```

$$x\left(x\,y^3+1\right)$$

```
> d1:=implicitplot(-x+x^11,x=-2..2,y=-2..2,scaling=constrained,
  thickness=2,axes=none,color=red,numpoints=2000):
> d2:=implicitplot(x^3+y,x=-2..2,y=-2..2,scaling=constrained,th
  ickness=2,axes=none,color=blue,numpoints=2000):
> display(d1,d2);
```



```
> factor(-x+x^11);
```

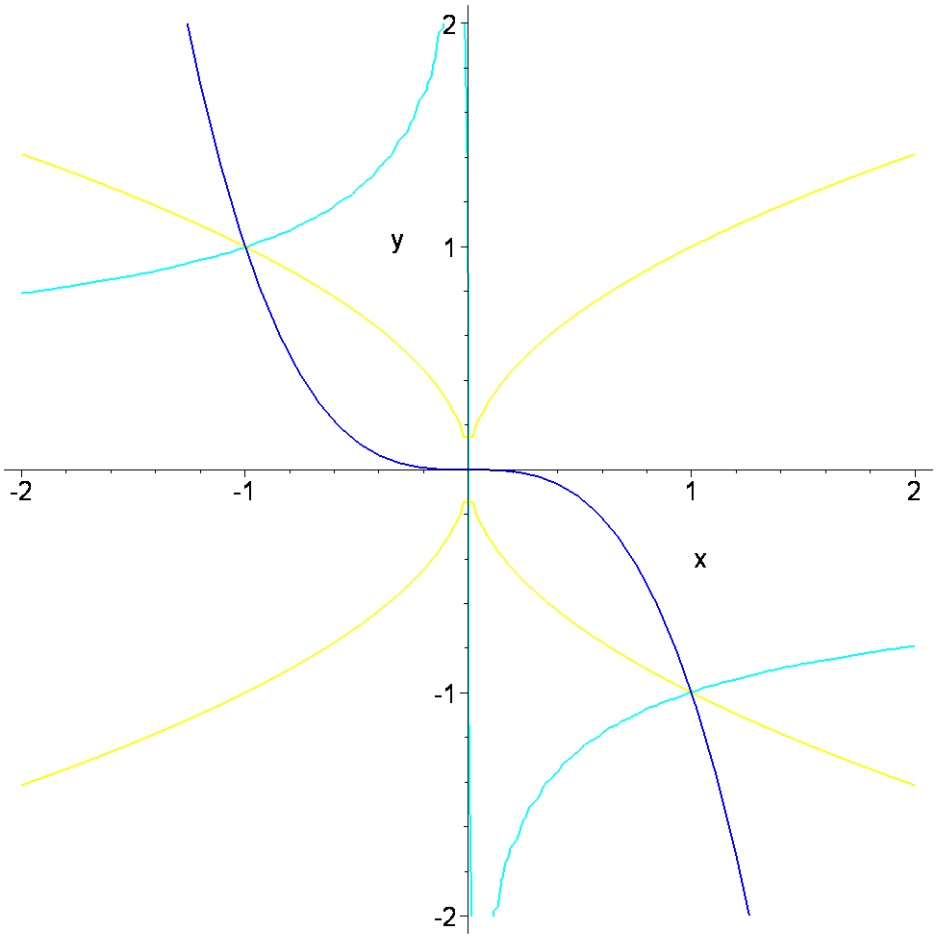
$$x\,(x-1)\,(x+1)\,(x^4+x^3+x^2+x+1)\,(x^4-x^3+x^2-x+1)$$

```
> solve(x^4-x^3+x^2-x+1,x);
```

$$\frac{1}{4}+\frac{\sqrt{5}}{4}+\frac{1}{4}I\sqrt{2}\,\sqrt{5-\sqrt{5}},\ -\frac{\sqrt{5}}{4}+\frac{1}{4}+\frac{1}{4}I\sqrt{2}\,\sqrt{5+\sqrt{5}},\ -\frac{\sqrt{5}}{4}+\frac{1}{4}-\frac{1}{4}I\sqrt{2}\,\sqrt{5+\sqrt{5}},$$

$$\frac{1}{4}+\frac{\sqrt{5}}{4}-\frac{1}{4}I\sqrt{2}\,\sqrt{5-\sqrt{5}}$$

```
> d1:=implicitplot(x^3+y,x=-2..2,y=-2..2,color=blue,scaling=con
  strained,thickness=2,numpoints=2000):
> d2:=implicitplot(y^4-x^2,x=-2..2,y=-2..2,color=yellow,scaling
  =constrained,thickness=2,numpoints=8000):
> d3:=implicitplot(x^2*y^3+x,x=-2..2,y=-2..2,color=cyan,scaling
  =constrained,thickness=2,numpoints=2000):
> display(d1,d2,d3);
```



Ejemplo de aplicación: decidir si está una variedad algebraica contenida en otra o no

Ejemplo: Comprobar que la curva dada por las ecuaciones {z-x^3=0, y-x^2=0} está contenida en la superficie xz-y^2=0:

```
> Basis( [z-x^3, y-x^2] , plex(y,x,z));
```

$$[-z+x^3, y-x^2]$$

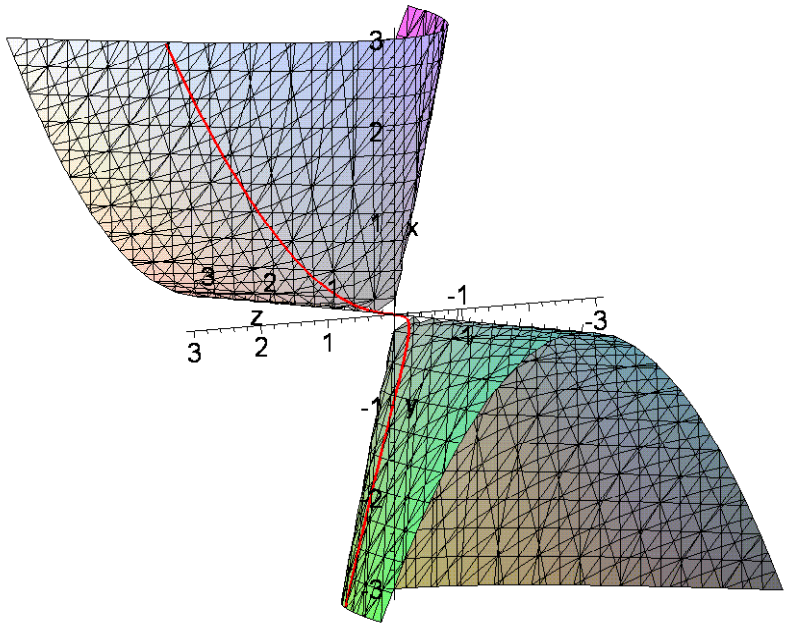
```
> Basis( [z-x^3, y-x^2, x*z-y^2] , plex(y,x,z));
```

$$[-z+x^3, y-x^2]$$

(como las GB son iguales, la superficie no "añade nada" a la curva, luego la curva estácontenida en la superficie).

Interpretación geométrica:

```
> c:=implicitplot3d(x*z-y^2,x=-3..3,y=-3..3,z=-3..3,axes=normal
  ,scaling=constrained,orientation=[128,72],numpoints=5000):
> d:=spacecurve([t,t^2,t^3],t=-1.44..1.44,color=black,axes=norm
  al,orientation=[128,72],color=red,thickness=3):
> display(c,d);
```



Ejemplo: También el eje "z" está contenido en esa superficie (las GB siguientes son iguales):

```
> Basis( [x, y] , plex(y,x,z));
```

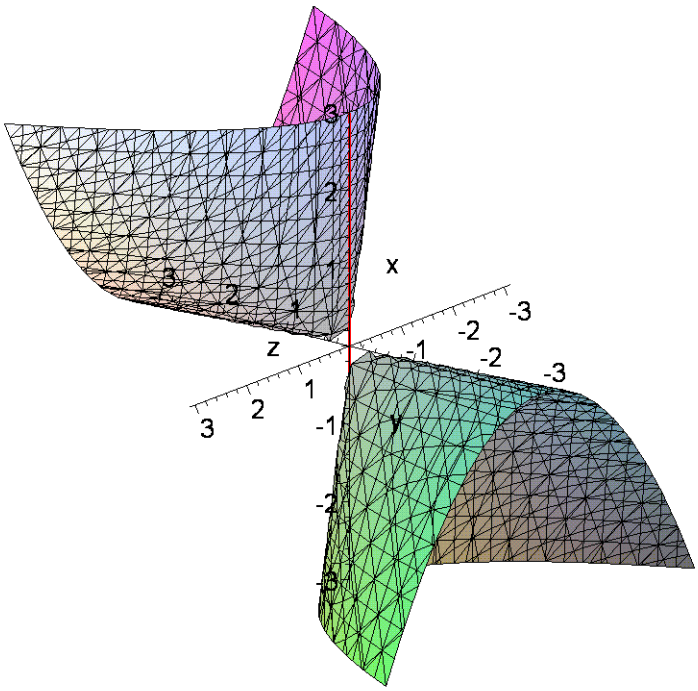
$$[x, y]$$

```
> Basis( [x, y, x*z-y^2] , plex(y,x,z));
```

$$[x, y]$$

Interpretación geométrica:

```
> c:=implicitplot3d(x*z-y^2,x=-3..3,y=-3..3,z=-3..3,axes=normal
  ,scaling=constrained,orientation=[128,72],numpoints=5000):
> d:=spacecurve([0,0,t],t=-3..3,color=black,axes=normal,orienta
  tion=[128,72],color=red,thickness=3):
> display(c,d);
```



Ejemplo: También lo está la recta x=y=z

```
> Basis( [x-y, x-z] , plex(y,x,z));
```

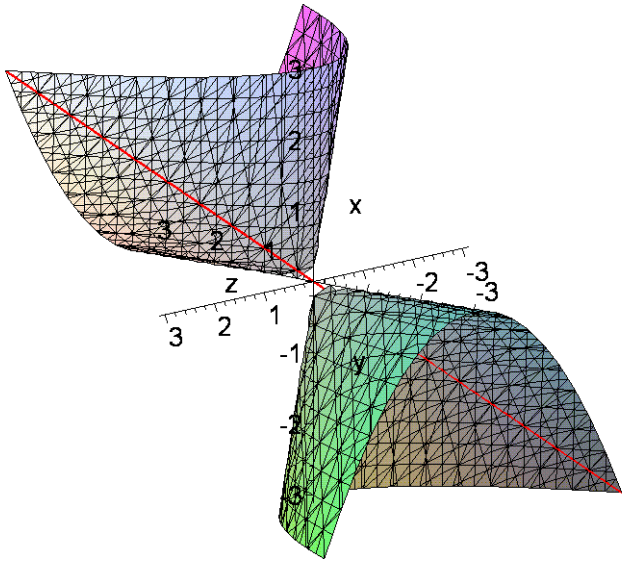
$$[x-z, -z+y]$$

```
> Basis( [x-y, x-z, x*z-y^2] , plex(y,x,z));
```

$$[x-z, -z+y]$$

Interpretación geométrica:

```
> c:=implicitplot3d(x*z-y^2,x=-3..3,y=-3..3,z=-3..3,axes=normal
  ,scaling=constrained,orientation=[128,72],numpoints=5000):
> d:=spacecurve([t,t,t],t=-3..3,color=black,axes=normal,orienta
  tion=[128,72],color=red,thickness=3):
> display(c,d);
```



Ejemplo: Pero no lo está la recta x=y=1:

```
> Basis( [x-1, y-1] , plex(y,x,z));
```

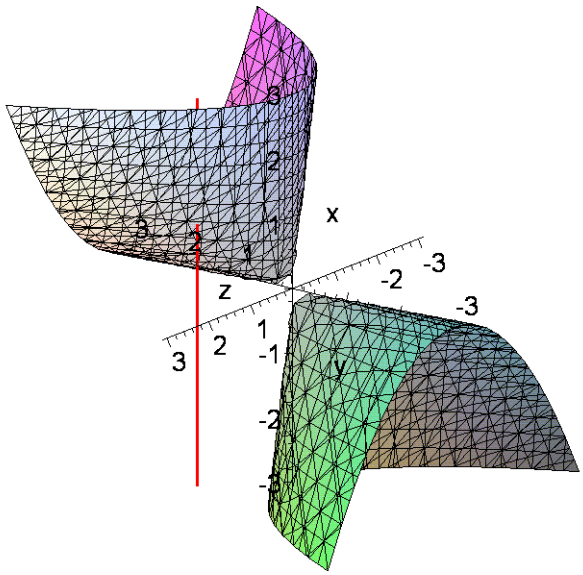
$$[x-1, y-1]$$

```
> Basis( [x-1, y-1, x*z-y^2] , plex(y,x,z));
```

$$[z-1, x-1, y-1]$$

Interpretación geométrica:

```
> c:=implicitplot3d(x*z-y^2,x=-3..3,y=-3..3,z=-3..3,axes=normal
  ,scaling=constrained,orientation=[128,72],numpoints=5000):
> d:=spacecurve([1,1,t],t=-3..3,color=black,axes=normal,orienta
  tion=[128,72],color=red,thickness=3):
> display(c,d);
```



>

Bibliografía:

E. Roanes Macías, Interpretación Geométrica de la Teoría de Ideales. Instituto Jorge Juan (CSIC), 1974.

E. Roanes Lozano, E. Roanes Macías, L.M. Laita. *The Geometry of Algebraic Systems and Their Exact Solving Using Groebner Bases*. Computing in Science and Engineering 6/2 (2004) 76-79.

FIN

Algunas aplicaciones de las Bases de Groebner

Eugenio Roanes Lozano
Universidad de Cantabria, 25-XI-2009

--- o o O o o ---

PARTE III: 3-Coloración de grafos con GB (debida a D. Bayer)

Idea:

1) Representamos los nodos con variables.
2) Identificamos cada color con una raíz cúbica de la unidad.
3) Creamos un sistema algebraico formado por las siguientes ecuaciones:
   i) todas las variables correspondientes a nodos, al cubo, igual a cero,
   ii) si hay una arista entre dos nodos, incluimos la suma de sus cuadrados con su producto.

El grafo es 3-coloreable syss este sistema es compatible.

Justicación: para todas las aristas, p.ej. una entre x e y: x^3-y^3 = 1-1 = 0 , pero:

```
> factor(x^3-y^3);
```

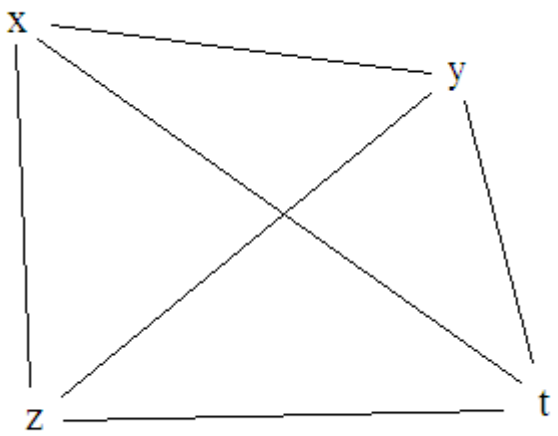
$$(x-y)\,(x^2+x\,y+y^2)$$

Implementación:

```
> restart;
> with(Groebner):
```

Ejemplos de grafos:

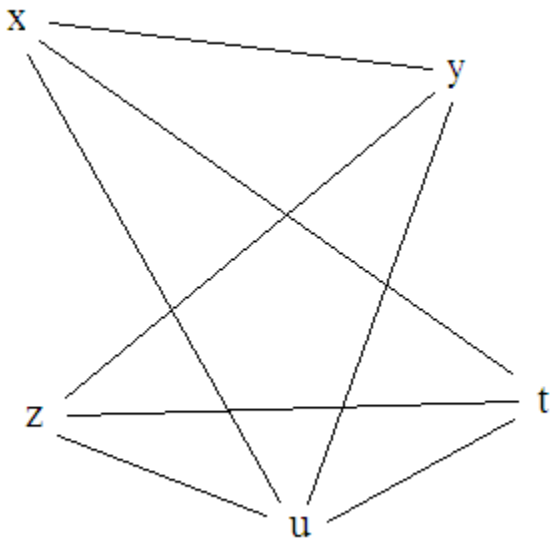
Claramente G1 no es coloreable (relacionados los 4 todos con todos)




```
> G1:=[[x,y],[y,z],[z,x],[t,x],[t,y],[t,z]]:
```

La cosa no es tan evidente para G2:

```
> G2:=[[x,y],[y,z],[z,t],[t,x],[u,x],[u,y],[u,z],[u,t]]:
```



Funciones

```
> pol_vertice:=x->x^3-1:
> pol_arista:= L -> L[1]^2+L[1]*L[2]+L[2]^2:
```

Procedimiento que calcula la GB del grafo:

```
> trecol:=proc(G::list)
>    {op(map(pol_arista,G))} union map(pol_vertice,indets(G)):
>    Basis( % , tdeg(op(indets(G))) );
>  end:
> trecol(G1);    #no 3-coloreable
```

$$[1]$$

```
> trecol(G2);    #3-coloreable
```

$$[x-z,\ u+y+z,\ t-y,\ y^2+yz+z^2,\ z^3-1]$$

Ejemplo de utilización de la salida del procedimiento trecol para determinar todas las 3-coloraciones posibles del grafo:

```
> trecol(G2):
> solve( % , indets(G2) );
```

$$\{t=\text{RootOf}(\_Z^2+\_Z+1),\ u=-\text{RootOf}(\_Z^2+\_Z+1)-1,\ x=1,\ y=\text{RootOf}(\_Z^2+\_Z+1),\ z=1\},\ \{t=1,\ u=-\text{RootOf}(\_Z^2+\_Z+1)-1,\ x=\text{RootOf}(\_Z^2+\_Z+1),\ y=1,\ z=\text{RootOf}(\_Z^2+\_Z+1)\},\ \{t=-\text{RootOf}(\_Z^2+\_Z+1)-1,\ u=1,\ x=\text{RootOf}(\_Z^2+\_Z+1),\ y=-\text{RootOf}(\_Z^2+\_Z+1)-1,\ z=\text{RootOf}(\_Z^2+\_Z+1)\}$$

En el output anterior, los tres colores son: 1, RootOf(_Z^2+_Z+1) y -RootOf(_Z^2+_Z+1)+1, esto es, (-1/2)+(1/2) I sqrt(3), (-1/2)-(1/2) I sqrt(3)

```
> allvalues([%]);
```

$$\left[\{t=-\frac{1}{2}+\frac{1}{2}I\sqrt{3},\ u=-\frac{1}{2}-\frac{1}{2}I\sqrt{3},\ x=1,\ y=-\frac{1}{2}+\frac{1}{2}I\sqrt{3},\ z=1\},\right.$$

$$\{t=1, u=-\frac{1}{2}-\frac{1}{2}I\sqrt{3}, x=-\frac{1}{2}+\frac{1}{2}I\sqrt{3}, y=1, z=-\frac{1}{2}+\frac{1}{2}I\sqrt{3}\},$$

$$\{t=-\frac{1}{2}-\frac{1}{2}I\sqrt{3}, u=1, x=-\frac{1}{2}+\frac{1}{2}I\sqrt{3}, y=-\frac{1}{2}-\frac{1}{2}I\sqrt{3}, z=-\frac{1}{2}+\frac{1}{2}I\sqrt{3}\}\Big],\Big[$$

$$\{t=-\frac{1}{2}-\frac{1}{2}I\sqrt{3}, u=-\frac{1}{2}+\frac{1}{2}I\sqrt{3}, x=1, y=-\frac{1}{2}-\frac{1}{2}I\sqrt{3}, z=1\},$$

$$\{t=1, u=-\frac{1}{2}+\frac{1}{2}I\sqrt{3}, x=-\frac{1}{2}-\frac{1}{2}I\sqrt{3}, y=1, z=-\frac{1}{2}-\frac{1}{2}I\sqrt{3}\},$$

$$\{t=-\frac{1}{2}+\frac{1}{2}I\sqrt{3}, u=1, x=-\frac{1}{2}-\frac{1}{2}I\sqrt{3}, y=-\frac{1}{2}+\frac{1}{2}I\sqrt{3}, z=-\frac{1}{2}-\frac{1}{2}I\sqrt{3}\}\Big]$$

En el output anterior, la diferencia entre las coloraciones en el primer corchete y en el segundo es la permutación de (-1/2)+(1/2) I sqrt(3) con (-1/2)-(1/2) I sqrt(3), luego es mejor la forma de presentar la solución anterior a esta última, preguntando en todo caso después:

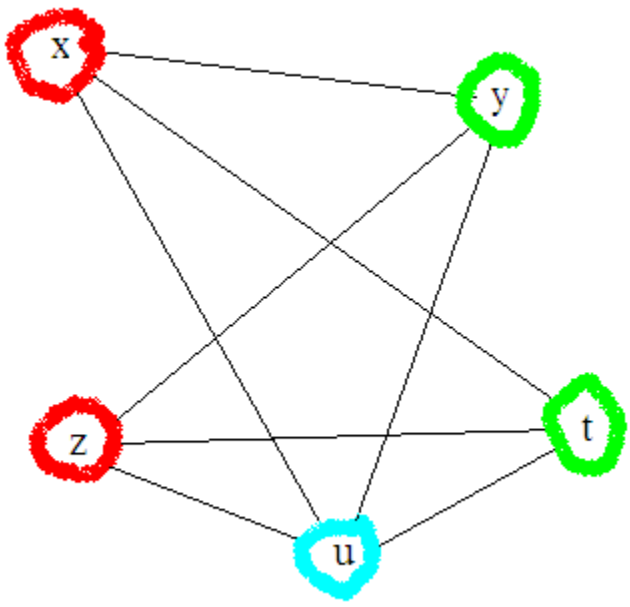
```
> allvalues(RootOf(_Z^2+_Z+1));
```

$$-\frac{1}{2}+\frac{1}{2}I\sqrt{3}, -\frac{1}{2}-\frac{1}{2}I\sqrt{3}$$

Una posible 3-coloración sería pues

$$\left\{t=-\frac{1}{2}+\frac{1\,I\,3^{\left(\frac{1}{2}\right)}}{2}, u=-\frac{1}{2}-\frac{1\,I\,3^{\left(\frac{1}{2}\right)}}{2}, x=1, y=-\frac{1}{2}+\frac{1\,I\,3^{\left(\frac{1}{2}\right)}}{2}, z=1\right\}$$

o sea, si, por ejemplo, 1=rojo , -1/2+1/2*I*3^(1/2) = verde , -1/2-1/2*I*3^(1/2) = azul



Bibliografía:
William W. Adams, Philippe Loustaunau: *An Introduction to Grobner Bases*. AMS Graduate Studies in Mathematics, 1994 (págs. 102-105).

FIN

Algunas aplicaciones de las Bases de Groebner

Eugenio Roanes Lozano
Universidad de Cantabria, 25-XI-2009

--- o o O o o ---

PARTE IV: aplicación de GB en enclavamientos ferroviarios

- El ferrocarril

(RENFE serie 100)

es un medio de transporte guiado:

• Un tren puede dirigirse hacia una vía u otra en los “desvíos”, mediante la colocación de las “agujas” en una de las dos posiciones posibles (vía directa / vía desviada).

Desvío mostrando las agujas en posición "vía directa" desde el lado de la punta (lado opuesto: talón).

Talonar: el tren llega al desvío desde el lado del talón con las agujas colocadas en al posición incorrecta.

¿Qué es un enclavamiento (railway interlocking)?

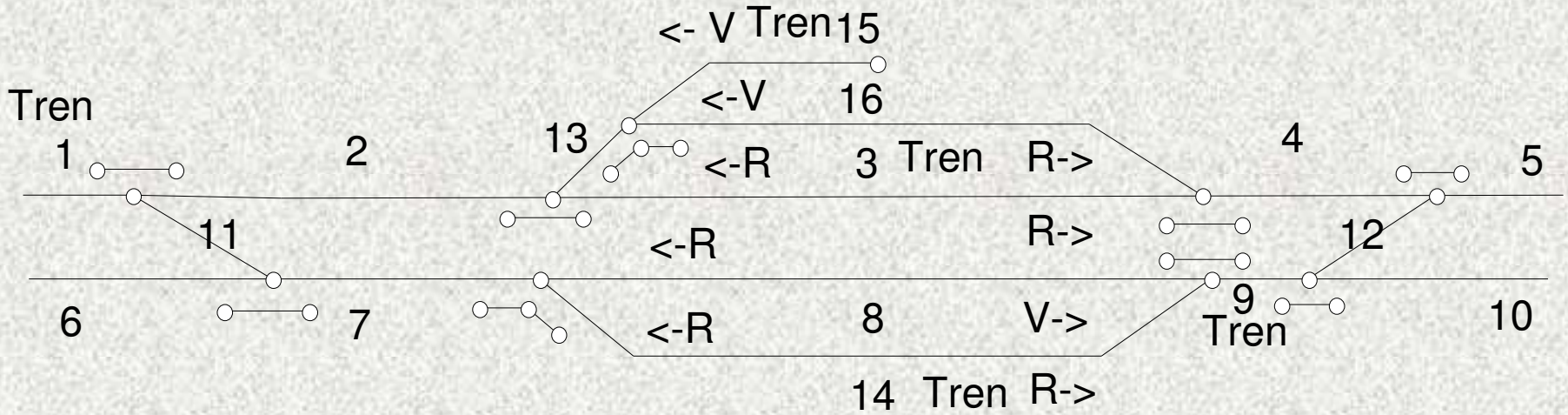
- Un enclavamiento ferroviario se instala para impedir una combinación de posiciones de señales y agujas que pudiera llevar a una colisión o al talonamiento de un desvío.

- El problema no es trivial. Pensemos por ejemplo en el caso siguiente:

Ejemplo de estación pequeña (circulación por la derecha, señalización en lado derecho, vía doble banalizada, desvios talonables -contemplamos retrocesos-)

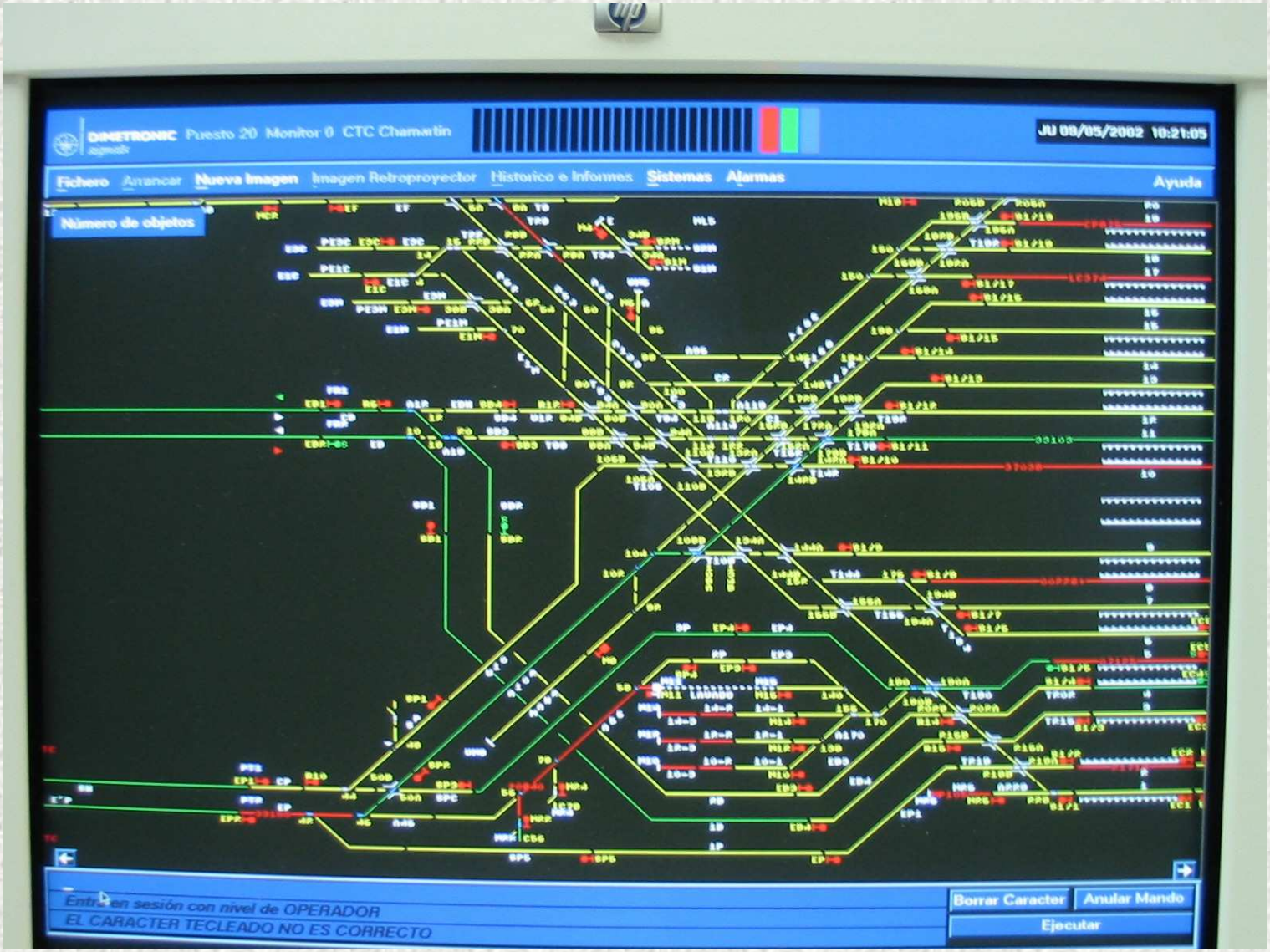
Posibles movimientos de trenes:

1 -> 1,2,3 ; 14 ->14 ; 9 -> 9,10,8 ; 3 -> 3 ; 15 ->15 (y talonando: 13,16,2,1,3)



- Pensemos en la complejidad del problema en una estación grande como Madrid-Chamartín:

Enclavamiento Estación de Chamartín en 2002 (tecnología mixta: mando desde ordenador de un enclavamiento de relés --antes de las obras para doble ancho de vía--, sutituido recientemente)



CTC Chamartín controlando la estación de Nuevos Ministerios (2002)

CTC Chamartín (parte de la sala de relés)

Evolución histórica de los enclavamientos:

- Primero: Saxby, 1859, en las cercanías de Londres.

- Inicialmente eran sistemas sencillos, relacionando dos palancas (como las cerraduras Bouré).

- Posteriormente se desarrollaron para palancas concentradas (mecánicos y más tarde relés).

- Desde los ‘80: enclavamientos controlados por ordenador (Chiaso, Rotterdam, Hilversum,...).

- RENFE: primeros enclavamientos electrónicos (1992): estaciones de Francia (Barcelona), Madrid-Atocha y AVE. RENFE en 2002: 150 enclavamientos electrónicos.

CTC y enclavamiento con tecnología de relés (ex-Renfe, puerto Pajares)

Enclavamientos controlados por ordenador:

- Un programa de ordenador es capaz de decidir sobre la seguridad de una situación, esto es, poder asegurar que no se podría darse una colisión, frontal o por alcance, entre trenes que respetaran las señales.

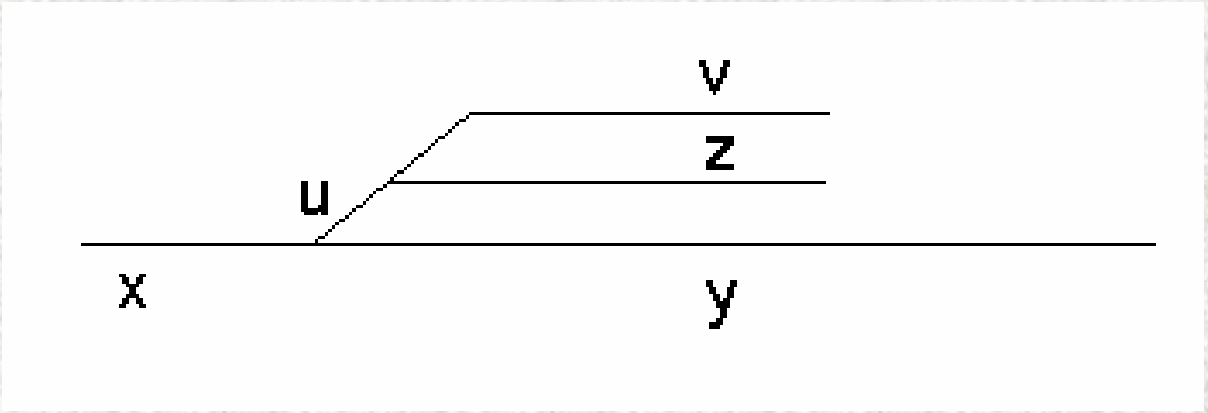
Enclavamientos independientes de la topología de la estación:

- Sería deseable que este "sistema de decisión" no tuviera que desarrollarse "ad hoc" para cada situación específica (esto es, que fuera independiente de la topología de la estación).

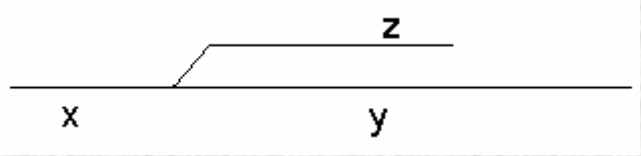
Nuestra aproximación al problema:

Interpretación como un grafo:

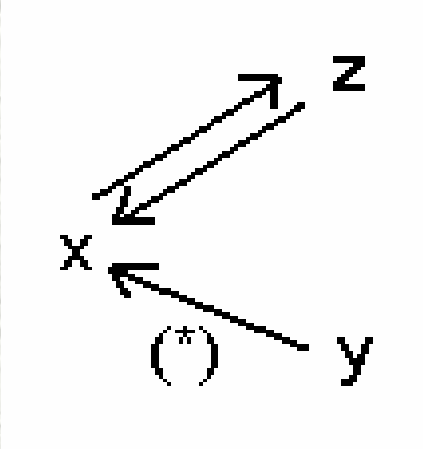
Nota: Denominaremos “cantones” no sólo a los denominados usualmente cantones en un sistema bloqueo sino también a los distintos tramos conectados por un desvío (p.ej. “u” en la fig.).



Pero, interesa considerar grafos dirigidos. Así, si las agujas están en la posición vía desviada en



el grafo dirigido asociado será:



(*) talonando, si se permiten talonamientos.
Y es obvio que hace falta considerar digrafos para semáforos.

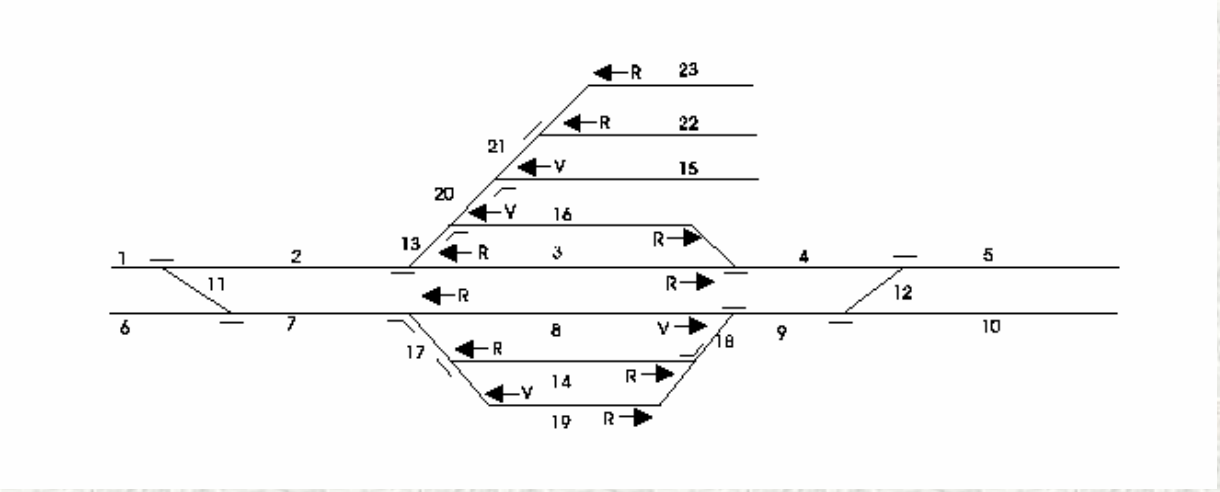
- Para estudiar si el paso a un cantón adyacente es posible, habría que comprobar:

- que topológicamente es posible

- que la posición de las agujas lo permite

- que el color de los semáforos lo permite.

• Ahora bien, cuando consideremos una instalación completa, no sólo un desvío, podremos realizar movimientos más complejos, no sólo al cantón adyacente, luego tendremos que considerar el cierre transitivo del grafo.



• Un convoy puede mantenerse en donde está, luego hay que considerar el cierre reflexivo del grafo.

Nuestras soluciones al problema:

Hemos desarrollado e implementado dos formas distintas de atacar el problema:

Modelo i) Algorítmico matricial.
(II Simposio Ing. Transporte 1996,
mejorada: IMACS-ACA 96 & MatCom 45/1 1998)

Modelo ii) Basado en técnicas algebraicas (bases de Groebner) y relativamente similar a otros trabajos de los mismos autores sobre verificación de sistemas expertos basados en reglas
(III Simposio Ing. Transp. 1998 e IEA-98 & Springer LNAI;
mejorada: IMACS-ACA'98 & MatCom 51/5 2000).

MODELO i)

- Se traduce la situación propuesta al modo usual en teoría de grafos:

- como una matriz cuadrada booleana, cuya dimensión es el número de cantones

- se calcula su cierre reflexivo-transitivo

- se calculan los conjuntos de cantones accesibles por cada tren

- se comprueba que esos conjuntos sean disjuntos dos a dos.

Ejemplo:

- Tratemos un ejemplo con nuestra implementación en Maple (recordemos que el paquete es independiente de la topología de la estación).

(Ej. en MAPLE 13 Classic)

Nota:

- Además de tratarse los dos casos (desvios talonables y no-talonables):

- se estudia si las agujas bajo un convoy que ocupe más de un cantón están bien situadas

- en el caso de que los desvíos sean no-talonables se advierte de los posibles talonamientos

(pero no entraremos en estos detalles aquí).

- Al usar MAPLE, el código es MUY BREVE.

MODELO ii)

- Se traduce la situación propuesta en un sistema de ecuaciones no lineales (ecuaciones de grado total ≤ 2). Se muestra como, curiosamente, la seguridad de la situación propuesta es equivalente a la compatibilidad de dicho sistema.

- Idea inspiradora (para grafos, no para digrafos): como decidir si un grafo es o no 3-coloreable usando GB (D. Bayer).

Codificación.- La idea (original de los autores) en que se basa el modelo es sencilla: un cierto sistema de ecuaciones polinomiales que denominaremos *SIST* recoge la información del grafo dirigido de semáforos y desvíos.

Los cantones se representan por enteros y los trenes por números enteros positivos.

a) La ecuación:

variable-núm=0

se incluye en *SIST* syss el tren número *núm* está en el cantón *variable.*

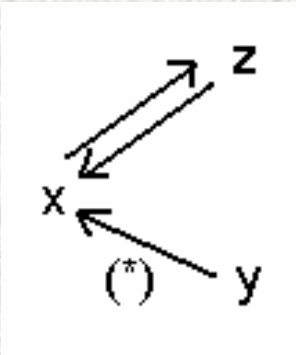
- Por ejemplo, la ecuación *x-11=0* se incluye en *SIST* syss el tren *11* está en el cantón *x*.

b) La ecuación

$$variable1 \cdot (variable1 - variable2) = 0$$

se incluye en *SIST* syss los cantones *variable1* y *variable2* son adyacentes y además es posible pasar del cantón *variable1* al *variable2* de acuerdo con la posición de las agujas y respetando los semáforos.

- Por ejemplo,



se representa incluyendo en *SIST* las ecuaciones:

$$x \cdot (x - z) = 0 \quad ; \quad z \cdot (z - x) = 0 \quad ; \quad y \cdot (y - x) = 0 \ .$$

Observación.- Suponemos que en cada cantón hay, a lo más, un tren. Por tanto, en *SIST* no puede haber dos ecuaciones de la forma:

variable-núm1=0 ; variable-núm2=0

(donde *núm1≠núm2)*.

Proposición.- Algún cantón será accesible por más de un tren syss *SIST* es un sistema incompatible.

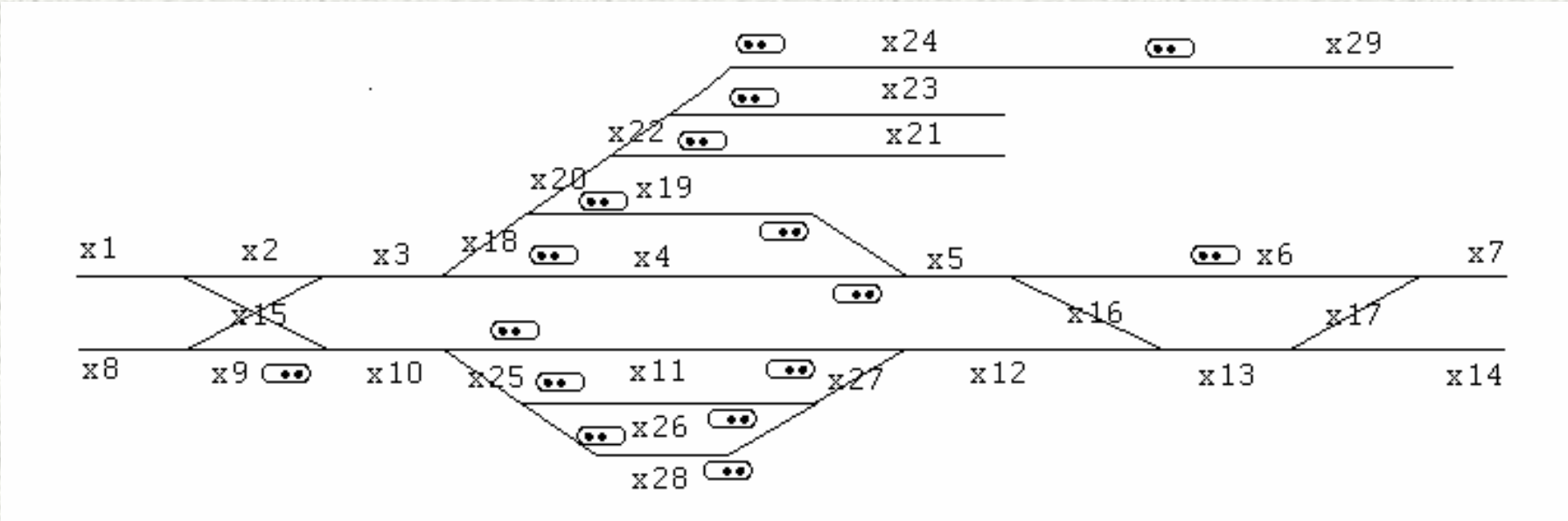
Justificación.- Tren *2* en cantón *x*: $x-2=0$ está en *SIST*.
*S*e puede pasar del cantón *x* al *y*: $x\cdot(x-y)=0$ está en *SIST*.
Entonces: $x=2 \Rightarrow y=2$.
Si se puede pasar del cantón *y* al *m*: $y\cdot(y-m)=0$ está en *SIST*.
En consecuencia: $m=2$.
Intuitivamente, el valor 2 se propaga por todos los cantones accesibles desde el *x*, y no sólo a los adyacentes.

Justificación algebraica usando Teoría de Ideales en (Roanes et al., 2000).

Consecuencia.- Se puede autorizar la posición de agujas y semáforos syss *SIST* es compatible.

Ejemplo:

- Tratemos el siguiente caso con nuestra implementación del modelo algebraico en Maple.



(Ej. en MAPLE 13 Classic)

Sobre el problema tratado:

- El problema es complejo: por ejemplo un modelo similar, independiente de la topología de la estación, pero que usa otra aproximación, fue el tema de la tesis de Markus Montigel en el ETH de Zurich.

- Utilizar un paquete como Maple (que incorpore Bases de Gröbner) hace que se simplifique increíblemente el código: por ejemplo nuestro paquete ocupa unas 80 líneas comparado con unas 20.000 de cierto paquete comercial.

FIN