# Gröbnerbasen

## SINGULAR

presented by Rudic Branislav

# Gröbnerbasen

F:

$$\begin{aligned} 2x - y - z &= 0 \\ x + 2y - 2z &= 1 \\ x - y + 2z &= 2 \end{aligned}$$

Gauss

G:

$$\begin{aligned} x + 2y - 2z &= 1 \\ y - 5z &= -4 \\ z &= 1 \end{aligned}$$

## Gröbnerbasen

F:

$$f = x^4 + x^3 - x - 1 = 0$$
$$g = x^4 + x^2 - 2 = 0$$

Euklid

G:

$$\gcd(f, g) = x^2 - 1 = 0$$

## Gröbnerbasen

F:

$$2x^4 - 3x^2y + y^4 - 2y^3 + y^2 = 0$$
$$4x^3 - 3xy = 0$$
$$4y^3 - 3x^2 - 6y^2 + 2y = 0$$

?

G:

$$3x^2 + 2y^2 - 2y = 0$$
$$xy = 0$$
$$y^3 - y^2 = 0$$

# Gröbnerbasen

Definition:

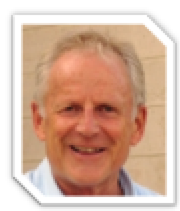
G is a **Gröbner basis** for <G>

:⇔

the reduction modulo G is unique

# Gröbnerbasen

Theorem (Buchberger 1965):



G is a **Gröbner basis** for <G>

:⇔

the reduction modulo G is unique

⇔

all S-polynomials reduce to 0

## Gröbnerbasen

F:

$$2x^4 - 3x^2y + y^4 - 2y^3 + y^2 = 0$$
$$4x^3 - 3xy = 0$$
$$4y^3 - 3x^2 - 6y^2 + 2y = 0$$

?

Gröbnerbasis:

$$3x^2 + 2y^2 - 2y = 0$$
$$xy = 0$$
$$y^3 - y^2 = 0$$

## Gröbnerbasen

Algorithmus (Buchberger 1965):

$G := F$
$C := \{\{g_1, g_2\} \mid g_1, g_2 \in G, g_1 \neq g_2\}$
**while** not all pairs $\{g_1, g_2\} \in C$ are marked **do**
  choose an unmarked pair $\{g_1, g_2\}$
  mark $\{g_1, g_2\}$
  $h :=$ normal form of spol$(g_1, g_2)$ w.r.t. $\longrightarrow_G$
  **if** $h \neq 0$ **then**
    $\{C := C \cup \{\{g, h\} \mid g \in G\}$
    $G := G \cup \{h\}$ }
**return** $G$ □

## Gröbnerbasen

F:

$$\begin{aligned} 2x^4 - 3x^2y + y^4 - 2y^3 + y^2 &= 0 \\ 4x^3 - 3xy &= 0 \\ 4y^3 - 3x^2 - 6y^2 + 2y &= 0 \end{aligned}$$

Buchberger

Gröbnerbasis:

$$\begin{aligned} 3x^2 + 2y^2 - 2y &= 0 \\ xy &= 0 \\ y^3 - y^2 &= 0 \end{aligned}$$

## Gröbnerbasen

Anwendungen:

- Gleichungssysteme
- Zugehörigkeitsproblem
- Gleichheit von Idealen

# SINGULAR

- entwickelt an der TU Kaiserslautern
- unter der Leitung von Gert-Martin Greuel, Gerhard Pfister und Hans Schönemann
- Schwerpunkte auf:
  - kommutative Algebra
  - algebraische Geometrie
  - Singulartätentheorie
- intuitive C-ähnliche Programmiersprache

# SINGULAR

- Wieso noch ein CAS?
  - Berechnungen in lokalen Ringen
- Was ist besonders an Singular?
  - Berechnungen in sehr allgemeinen Ringen
  - Große Anzahl verschiedener Körper für diese Ringe
  - C-ähnliche Programmiersprache
  - Schnittstellen für andere Programme / sich selbst
  - Hohe Rechengeschwindigkeit

# SINGULAR

- Baserings:
  - Polynomial Ring
  - Series Ring, i.e. a locaization of a polynomial ring
  - Factor Ring by an ideal of one of the above
  - Tensor products of one of the above, ...
- Ground fields:
  - Rational Numbers Q (char 0)
  - Finite Field Z/p (p a prime <= 2147483647), ...

# SINGULAR

- Global Orderings for polynomial rings:
  lp, dp, wp, Dp, Wp
- Local Orderings for series rings:
  ls, ds, ws, Ds, Ws
- Matrix Orderings
- ...

# SINGULAR

- Defining a Ring:

ring R=*coefficient field*, *variables*, *ordering*;

Examples:

ring R1=2147483647 , (x,y,z), dp;
ring R2=(0,a), x(1..3), ds;
ring R3=0, x, lp;

# SINGULAR

- Solving Equations:

$$\begin{aligned} 3x+ \quad y+ \quad z- \quad u &= a \\ 3x+ \quad 8y+ \quad 6z- \quad 7u &= b \\ 14x+ \quad 10y+ \quad 6z- \quad 7u &= c \\ 7x+ \quad 4y+ \quad 3z- \quad 3u &= c \end{aligned}$$

# SINGULAR

- Solving Equations:

```
ring R=(0,a,b,c,d), (x,y,z,u), dp;
poly f1=3x+y+z-u-a;
poly f2=3x+8y+6z-7u-b;
poly f3=14x+10y+6z-7u-c;
poly f4=7x+4y+3z-3u-d;
ideal i =f1, f2, f3, f4;
//option(redSB);
std(i); ...
```

# SINGULAR

- Solving ideal membership problem:

```
ring R=0, (x,y), dp;
poly g=(1-x)*(x2-y3);
poly h=y2-x2;
ideal i=g,h;
poly f=x2-x2y;
NF(f,std(i));
...
```

SINGULAR

# $