

Multivariate polynomials in Julia

Benoît Legat

July, 2022

Joint work with Robin Deits, Chris Elrod and Yingbo Ma

JuliaCon-dev 2022



Where are multivariate polynomials used in Julia ?

Used in variety of applications:

- HomotopyContinuation: Solve polynomial systems with homotopy continuation
- SumOfSquares: Sum-of-Squares programming
- Symbolics: Manipulate and simplify symbolic expressions
- Groebner: Gröbner bases computations based on Faugère's F4 algorithm
- SemialgebraicSets: Reduction modulo ideal with Gröbner bases + solve polynomial systems with Schur decomposition
- Nemo/AbstractAlgebra: Computer algebra system
- CondensedMatterSOS: Condensed Matter with Sum-of-Squares
- ...

How are multivariate polynomials represented?

Sorted vector of terms:

```
struct Term{C,M}  
    coefficient::C  
    monomial::M  
end  
struct Polynomial{C,M}  
    terms::Vector{Term{T,M}}  
end
```

How are multivariate polynomials represented?

Sorted vector of terms:

```
struct Term{C,M}  
    coefficient::C  
    monomial::M  
end  
struct Polynomial{C,M}  
    terms::Vector{Term{T,M}}  
end
```

Sorted how ? Any ordering ?

How are multivariate polynomials represented?

Sorted vector of terms:

```
struct Term{C,M}
    coefficient::C
    monomial::M
end
struct Polynomial{C,M}
    terms::Vector{Term{T,M}}
end
```

Sorted how ? Any ordering ?

If $m_1 > m_2$ and $m_3 > m_4$, we need $m_1 m_3 > m_2 m_4$.

See 2.§2.Definition 1 of Cox Little O'Shea *"Ideal, Varieties, and Algorithms"*.

How are monomials represented?

DynamicPolynomials

```
struct Variable
  id::Int
  name::String
end
struct Monomial
  variables::Vector{Variable}
  exponents::Vector{Int}
end
```

TypedPolynomials (Robin Deits)

```
struct Variable{Name} end
struct Monomial{V<:Tuple{Vararg{Variable,N}},N}
  exponents::NTuple{N,Int}
end
```

How are monomials represented? (part 2)

SIMDPolynomials (Chris Elrod and Yingbo Ma)

Bit packed with maximum L variables and E bits of exponents (so $64K > LE$).

```
struct Monomial{L,E,K}  
    bits::NTuple{K,UInt64}  
end
```

CondensedMatterSOS (Jacopo Surace)

```
struct Monomial  
    variables::SortedDict{Int,SpinVariable}  
end
```

MultivariatePolynomials API

- `AbstractVariable`, `AbstractMonomial`, `AbstractTerm`, `AbstractPolynomials`
- Implementation-independent API: `degree`, `coefficient`, `monomial`, `isless`, `+`, ...
- Implements `MutableArithmetics` to make **generic** code **fast**
- Pretty printing
- Differentiation
- Substitution (partial) and evaluation (i.e., full substitution)
- Euclidean division and **gcd/gcdx**

Get all that for free for any new fancy monomial representation!