Projects for Computer Algebra, WS2019

1 Division Algorithm

Similar to ordinary division in Euklidean domains there is a general concept of dividing a polynomial $f \in k[x_1, \ldots, x_n]$ by a finite **set of polynomials** $\{g_1, \ldots, g_r\} \subseteq k[x_1, \ldots, x_n]$.

1. Elaborate the necessary theory.

2. Implement an algorithm which computes the reduced Gröbner basis of a finite set of polynomials with respect to a term order together with the additional information of representation.
   Input: $F = [f_1, \ldots, f_r] \in \mathbb{C}[x_1, \ldots, x_n]^r$, $<$ a term order
   Output: G: reduced Gröbner basis of $\langle F \rangle$ wrto $<$
   A: matrix of polynomials such that $G = AF$.

References: [18] [1] [3]

2 Zero-dimensional Systems

Frequently in applications, a polynomial system has only finitely many solutions.

1. Write down criteria for determining whether a system of polynomial equations in several indeterminants with coefficients in $\mathbb{C}$ has only finitely many solutions. Present the necessary theory.

2. Implement an algorithm within a computer algebra system which decides whether a polynomial system has finitely many solutions and - in the affirmative case - determines all solutions to some specified precision.
   Input: $f_1, \ldots, f_r \in \mathbb{C}[x_1, \ldots, x_n]$, $d \in \mathbb{N}$
   Output: The list of all finitely many solutions of $f_1 = \cdots = f_r = 0$ in $\mathbb{C}^n$
   in n-tuples of complex numbers with precision d
   OR **INFINITE**.

References: [14] [18] [1] [17]

3 Basis Conversion

The concept of Gröbner basis depends heavily on the chosen term order. Besides that they may look quite disparate, Gröbner bases with respect to different

orders serve dissimilar needs.

Instead of computing a Gröbner basis of an ideal for several term orders separately, one may convert one into another.

1. Give the theory of converting a Gröbner basis of a zero-dimensional ideal with respect to a given term order into a lexicographic Gröbner basis.

2. Implement an algorithm which achieves this goal.
   Input: $<$ term order of $\{x_1, \ldots, x_n\}$
   $i\colon \{1, \ldots, n\} \longrightarrow \{1, \ldots, n\}$ permutation
   $F = \{f_1, \ldots, f_r\} \subset k[x_1, \ldots, x_n]$ Gröbner basis of the zero-dimensional ideal $\langle F \rangle$ wrto $<$
   Output: The reduced Gröbner basis of $\langle F \rangle$ wrto lex-order $x_{i_1} > \cdots > x_{i_n}$.

References: [15], [16] available online at

http : *//www.cs.purdue.edu/homes/cmh/distribution/books/geo.html*

4 Implicitization by Interpolation

Finding the implicit representation of a parametrized variety means finding the coefficients of a finite set of polynomials. So, if we know bounds for the degrees of the desired polynomials, we may evaluate the given parametrizing functions in some finite set of interpolation nodes, thereby obtaining a linear system L. A nontrivial solution of L yields an answer to the implicitization problem.

1. Work out the theory behind these remarks.

2. Implement an algorithm that does the job for curves.
   Input: $(r_1(t), r_2(t))$ proper parametrization of an irreducible plane curve $\mathcal{C}$
   Output: The implicit representation of $\mathcal{C}$.

3. Run the algorithm on some examples.

References: [12], [9]

5 Implicitization by Gröbner Bases

1. Work out the theory of implicitizing rationally parametrized algebraic varieties. Use Gröbner bases for computing the implicit representation out of a rational parametrization.

2. Demonstrate your results by computing several examples.

References: [14], [5], [12]

6 Geometric Theorem Proving

1. Develope the theory of proving theorems of plane geometry by using Gröbner bases: Translation of geometric statements into polynomial equations, definition of a strict/generic consequence from a set of hypotheses, sufficient conditions for being a strict/generic consequence.

2. Write an algorithm that detects whether a given geometric statement follows (strictly or generically) from a given system of geometric statements.

3. Prove a nontrivial geometric theorem of your choice along these lines.

References: [14]

7 Universal Gröbner Bases

Work out the theory of universal Gröbner bases. You can use the book [1] pages 514 ff and the references therein.

8 Comprehensive Gröbner Bases

Explain the theory of comprehensive Gröbner bases. Again [1] pages 515 ff together with its references is a good starting point.

9 Squarefree Factorization

1. Elaborate the theory of squarefree decomposition in unique factorization domains.

2. Write a program that computes the squarefree decomposition of multivariate rational polynomials.
   Input: (f, n) with $f \in \mathbb{Q}[x_1, \ldots, x_n]$
   Output: The squarefree decomposition of f.

References: [18] and [1] pages 99–109.

10 Ideal Arithmetic

Write programs for effective operations on ideals.

Input: $a, b \trianglelefteq \mathbb{Q}[x_1, \ldots, x_n]$
represented by polynomials $f_1, \ldots, f_r, g_1, \ldots, g_s \in \mathbb{Q}[x_1, \ldots, x_n]$
such that $a = \langle f_1, \ldots, f_r \rangle$, $b = \langle g_1, \ldots, g_s \rangle$.
Output: generators for $a + b$, $a \cdot b$, $a \cap b$, $a : b$, $\sqrt{a}$.
References: [19], [18], [1],

11 Linear Algebra over Polynomial Rings

1. Write out the theory of syzygies by using e.g. [1] section 6.1.

2. Implement an algorithm that computes solutions of linear equations over $\mathbb{Q}[x_1, \ldots, x_n]$.
   Input: $f_1, \ldots, f_r, g \in \mathbb{Q}[x_1, \ldots, x_n]$
   Output: The general solution of the equation.
   $f_1 y_1 + \cdots + f_r y_r = g$.

References: [1][20]

12 Hilbert Function

1. Develope the theory of Hilbert functions of graded modules over $\mathbf{k}[x_0, \ldots, x_n]$. Explain the concept of Hilbert series and its relation to the Hilbert function. Study methods for computing these objects.

2. Use your knowledge about Hilbert functions to determine essential data (e.g. degree, dimension,...) of some interesting algebraic varieties.

References: [20]

13 Modular GCD Computation

1. Examine in detail the theory at the basis of modular GCD computations.

2. Implement a modular GCD algorithm for integers.
   Input: $a, b, \in \mathbb{Z}$
   Output: $\mathrm{GCD}(a, b)$.

3. Adapt your program to modular GCD-computation of multivariate polynmials over $\mathbb{Q}$.

References: [18]

14 Robotics and Motion Planning

1. Work out the theory of planar robots (joint space, configuration space, forward/inverse kinematic problem).

2. Demonstrate the theory by means of a planar robot with a fixed segment 1 and with n revolute joints linking segments of length $l_2, \ldots, l_n$. The 'hand' is segment $n+1$, attached to segment n by joint n. Determine the position of the hand as a function of joint settings.

3. Consider a concrete planar robot with 3 revolute joints linking 4 segments of length 1, followed by one prismatic joint taking length values from the interval $[0, 1]$, linking the 4th segment to the hand. Solve the inverse kinematic problem for this robot. Describe possible kinematic singularities.

References: [14] [21]

15 Verifyable Algorithms in Computer Algebra

Proof Assistants like Isabelle [25] go towards *verfied* algorithms also in Computer Algebra, a respective Polynomial Package [23] is under construction. This project promotes first experiments with the package.

1. Implement some basic algorithms (division, conversion between polynomial representations, etc — to be defined at start of the project) using Isabelle's functional programming language [24].
2. Consider provability of the algorithms' properties within the given logical environment.
3. Use Isabelle's code generator [22] to make your algorithms executable (while proved properties are preserved by the system).

References

[1] T. Becker and V. Weisspfenning. *Gröbner Bases. A Computational Approach to Commutative Algebra.* Springer 1993.

[2] B. Buchberger. Gröbner basis: an algorithmic method in polynomial ideal theory. *Multidimensional Systems Theory* (1985), N. K. Bose ed., 184–232.

[3] B. Buchberger and F. Winkler. Gröbner Bases and Applications. *London Math. Soc. Lecture Notes 251* (1998).

[4] D. Cox, J. Little and D. O'Shea. *Ideals, Varieties, and Algorithms.* Springer Verlag, New York (1996).

[5] D. Cox, J. Little and D. O'Shea. Using Algebraic Geometry. Springer Verlag, New York (1998).

[6] G. Fix, Chih-Ping Hsu and Tie Luo. Implicitization of Rational Parametric Surfaces. *Journal of Symbolic Computation 21* (1996), 329–336.

[7] Xiao-Shan Gao and Shang-Ching Chou. Implicitization of Rational Parametric Equations. *Journal of Symbolic Computation 14* (1992), 459–470.

[8] D. Manocha and J.F. Canny. Algorithms for Implicitizing Rational Parametric Surfaces. *Computer Aided Geometric Design 9* (1992), 25–50.

[9] A. Marco and J. J. Martinez. Using polynomial interpolation for implicitizing algebraic curves. Computer Aided Geometric Design 18(4) (2001), 309–319.

[10] J. Schicho. Rational Parametrization of Real Algebraic Surfaces. *RISC-report, RISC-Linz 98-01* (1998).

[11] T.W. Sederberg. Improperly parametrized rational curves. *Computer Aided Geometric Design 3* (1986), 67–75.

[12] J.R. Sendra and F. Winkler. A symbolic algorithm for the parametrization of algebraic plane curves. *Tech. Rep. 89-41.1, RISC-Linz* (1998).

[13] J.R. Sendra and F. Winkler. Parametrization of Algebraic Curves over Optimal Field Extensions. *Journal of Symbolic Computation 23* (1997), 191–207.

[14] D. Cox, J. Little, and D. ÓShea. *Ideals, Varieties, and Algorithms.* Undergraduate Texts in Mathematics. Springer Verlag, 1992.

[15] J.C. Faugère, P. Gianni, D. Lazard, and T. Mora. *Efficient computation of zero-dimensional Groebner bases by change of ordering.* Journal of Symbolic Computation, 16:329–344, 1993.

[16] C. M. Hoffmann. *Geometric and Solid Modelling – an Introduction.* Morgan Kauffmann Publisher, San Mateo, California, 1989.

[17] Q.-N. Tran and F. Winkler, editors. *Applications of the Gröbner Bases Method*, volume 30 of *special issue of the J. of Symbolic Computation*, Oct. 2000.

[18] F. Winkler. *Polynomial algebrithms in computer algebra.* Springer-Verlag Wien New York, 1996.

[19] M. Atiyah and I.G. MacDonald. *Introduction to Commutative Algebra.* Addison Wesley, London, 1969

[20] D. Eisenbud. *Commutative Algebra with a View Toward Algebraic Geometry.* Graduate Texts in Mathematics 150, Springer 1995

[21] Tomas Lozano-Perez. *A simple motion-planning algorithm for general robot manipulators* IEEE Journal of Robotics and Automation, volume RA-3, NO. 3, June 1987

[22] Florian Haftmann. *Code generation from Isabelle/HOL theories.* Theorem Proving Group at TUM, Munich, 2015. Part of the Isabelle distribution, `http://isabelle.in.tum.de/dist/Isabelle2015/doc/codegen.pdf`.

[23] Florian Haftmann, Andreas Lochbihler, and Wolfgang Schreiner. Towards abstract and executable multivariate polynomials in isabelle. Isabelle Workshop 2014, `http://www.infsec.ethz.ch/people/andreloc/publications/haftmann14iw.pdf`, 2014.

[24] Alexander Krauss. *Defining Recursive Functions in Isabelle/HOL.* Theorem Proving Group TUM, Munich, 2015. Part of the Isabelle distribution, `http://isabelle.in.tum.de/dist/Isabelle2015/doc/functions.pdf`.

[25] Tobias Nipkow, Lawrence C. Paulson, and Markus Wenzel. *Isabelle/HOL — A Proof Assistant for Higher-Order Logic*, volume 2283 of *LNCS.* Springer, 2002.