Week 2 Exercises

Notes by James H. Davenport

10 September 2018

Exercise 1 is worth 5%, 2 and 3 10% each. 1 and 2 may be done in either Reduce or Singular: 3 needs Singular.

Submission is by e-mail to `J.H.Davenport@bath.ac.uk` by **23:59 on Thursday 13 September**, a single zip file, called 3160567890Ex2.zip (if your student ID is 3160567890 - use YOUR OWN student number) containing a worksheet (Result of saving in Reduce, but these are not very legible, so your PDF will need screenshots of key answers; Jupyter notebook in Sage) and a PDF with the answers, for each of the three questions.

So the PDF for question might look like Figure 1

Figure 1: Sample answer

Gröbner base is $a^2 + b * c + b^4 + c^8 = 0, b^3 + c^9 = 0, c^7 = 0$
6 S-polynomials computed of which 2 reduced to 0
The gcd criterion saved 3 and the lcm criterion 4
The minimal base is $a^2 + b * c = 0, b^3 = 0, c^7 = 0$
[The point is that b^3 reduces b^4 etc.]

1. Emulate the Buchberger algorithm on cyclic–3: viz: $a + b + c$; $ab + bc + ca$; $abc - 1$. By this I mean that each S-polynomial should be computed and reduced under human control, i.e. the most sophisticated algebraic operators you can use are of the form `S:=28x*f1-3*y*f2` or S:=S-7*z*f3: you may also use `expand` if necessary. You have to work out the leading monomials yourself (not using any built-in commands). You may use any order you wish, but should state it clearly in the worksheet. How many S-polynomials do you compute? How many of these reduce to zero? Use the criteria[1] to avoid computing polynomials, and you should also state (comment in your worksheet) how many were avoided.

1b. Based on these calculations, what is a *minimal* (i.e. fewest polynomials) Gröbner base for your chosen ordering? Tell me in your worksheet. This is Interreduce on slide 5/15 which is (2) on slide 3/13.

[1] See Slide 26 from Week 5; uploaded 10 September.

2. This question is about the cyclic–5 problem (Singular has `cyclic(5)`): viz.

$$L := \begin{cases} a+b+c+d+e \\ ab+bc+cd+de+ea \\ abc+bcd+cde+dea+eab \\ abcd+bcde+cdea+deab+eabc \\ abcde-1. \end{cases}$$

   (a) Compute (using any commands you want) a total degree reverse lexicographic (`tdeg`) for cyclic-5. Hence deduce, by working the Proposition (Lecture 5 Slide 19) yourself and **not** using a builtin) *how many* solutions this system has.

   (b) Convert this, via the Faugère–Gianni–Lazard–Mora algorithm (you may use the built-in one), into a purely lexicographical Gröbner basis.

   (c) Hence deduce, using the Gianni–Kalkbrener theorem, the number of solutions and a description of them. A description might say "e is a root of p_1 (a polynomial). When e is a root of p_2 (a smaller polynomial), then d is given in terms of e by $p_3, \ldots$". If Reduce/Sage has computed the p_i, you do not need to copy them out, just cross-reference the worksheet in Sage, but I fear you need a screenshot in Reduce..

   To give a complete example, Part 2c for $\{x^8-1, (x^4-1)(y^6-1), y^{12}-1\}$ would be

   - When x is a root of x^4-1, y is a root of $y^{12}-1$: $4\times 12 = 48$ solutions.
   - When x is a root of x^4+1, y is a root of y^6-1: $4\times 6 = 24$ solutions.
   - Total: 72 solutions.

3. The aim of this exercise is to understand modular Gröbner bases, on the lines of [IPS11, Algorithm 1].

   (a) Compute this was a degrevlex ($\prec_{\mathrm{tdeg}}$, `dp` in Singular) basis over the integers for the ideal in [Arn03, Example 1.1].

   (b) For the cyclic-5 example, look at small primes (at least $2,\ldots,13$) and see what the various leading monomial sets are. See the Sage notebook at `http://staff.bath.ac.uk/masjhd/Zhejiang/EX2-3demo.ipynb` for how to get started. What would you try combining to get a solution?

   (c) How far can you get (*do not spend more than one hour on this*) with cyclic-7.

1 Q&A

Q1 Am I right in thinking that these don't require programming.

A1 That's correct - you have to do the calculations, using Reduce/Sage as a calculator. At some points, you may find it *helpful* to write a small program, but that's your concern.

References

[Arn03] E.A. Arnold. Modular algorithms for computing Gröbner bases. *J. Symbolic Comp.*, 35:403–419, 2003.

[IPS11] I. Idrees, G. Pfister, and S. Steidel. Parallelization of Modular Algorithms. *J. Symbolic Comp.*, 46:672–684, 2011.