Math 370 Computational Algebra - Homework 8

due April 30th

Part I (15 pts) The following are all sufficient conditions for a set of (nonzero) polynomials $G = \{g_1, \ldots, g_s\}$ to be a Gröbner basis for the ideal I they generate. Choose *one of them* and prove that it is sufficient (the last one is worth 20 points). However, they are not necessary conditions. For `each one of them`, find a counterexample to show that they are not necessary. Without loss of generality, you can always assume that the g_i's are monic. Note also that the first two conditions do not depend on the term order you choose, so if G satisfies one of them, it is automatically a σ-Gröbner basis for every possible choice of σ. The third one, however, depends on σ. Why?

1) $\quad s = 1$

That is, if $I = \langle g \rangle$ is a principal ideal, then the set $\{g\}$ is a Gröbner basis for I. We called this fact "Amy's theorem"!

2) $\quad$ For all i, we have $g_i \in \mathbb{T}^n$.

That is, when I is a monomial ideal, its monomial generators are automatically a G. basis.

3) $\quad$ For all $i \neq j$, we have $GCD(LT_\sigma(g_i), LT_\sigma(g_j)) = 1$.

That is, when the leading terms of the generators are relatively prime, then G is automatically a Gröbner basis. Notice that this allows you to recognize many Gröbner bases "at a glance", for example if the leading terms contain different variables. Hint for the proof: apply Buchberger's criterion and the following property: $GCD(s,t) * LCM(s,t) = st$ for all terms s, t.

Part II: Choose two problems. (**15 points each**)

4) Calculate the S-polynomials, with respect to `DegLex`, of all pairs of elements of the following set:

$$G = \{x^2y^2z + 2xy + z, y^3z^2 - x^4, 2xyz + yz.\}.$$

Conclude that G is *not* a `DegLex`-Gröbner basis for the ideal it generates. Use CoCoA to generate a Gröbner basis for the ideal $\langle G \rangle$ using the command `GBasis`.

5) Construct a `Lex`-Gröbner basis for the ideal $I = \langle x^3 - y^3, x^2 + 2y^2 \rangle \subset \mathbb{Q}[x, y]$ using Buchberger's algorithm. Then calculate the leading term ideal $LT(I) = \langle LT(f) \mid f \in I \rangle$ and check your result with the CoCoA command `LT`. Draw a picture of $LT(I)$ in $\mathbb{N}^2$ that illustrates what it is. How many terms are left out? [Question for those who are taking Modern Algebra: what does the "left out" region represent?]

6) Let $I = \langle g_1, g_2, g_3 \rangle = \langle y^2 - x, z^2 - y, w^2 - z \rangle \subset \mathbb{Q}[x, y, z, w]$. Prove that its generators are a Gröbner basis with respect to all three term orders we have defined in class. Note that you are not allowed to use 3) unless you have proved it! Read the definition of *reduced* Gröbner basis from the book and show that this set of generators actually is *the* reduced Gröbner basis of I. Test it using the CoCoA command `ReducedGBasis`. What happens if we add the polynomial $g_4 = w^4 - y$? Is $\{g_1, \ldots, g_4\}$ still a Gröbner basis? Can you make this into a more general statement?

Part III: Choose one problem. (**15 points**)

7) Write a CoCoA function `SPoly` which takes two polynomials, checks that they are both nonzero (if one of them is zero it should return an error message), and calculates their S-polynomial. Test it on the three generators for the ideal in problem 4). Then write a function `IsGBasis` which takes an ideal (again returning an error message in case one of the generators is zero[1]), and checks whether its generators form a Gröbner basis with respect to the active term order, using your function `SPoly` and the built-in function `NR` for the normal remainder. Test your function against all the ideals from Part II. The output of `IsGBasis` should be a boolean value.

8) Write a CoCoA program `CompareIdeals` which takes two nonzero ideals I and J and returns a message saying what their relative position is within the ring of polynomials, e.g. "*I is a proper subset of J*", and so on. Make sure you exhaust all *possible* cases. Do not make use of the CoCoA function `IsIn`. You are allowed, however, to use either `NR` or `NF`. Produce at least one example to illustrate all possible cases and test your program against them. For 10 points of extra credit, make CoCoA generate 1000 random pairs of ideals of $\mathbb{Q}[x, y, z]$ (choose a bound on the degree of the generators, the number of generators, and use the function `randomized`) and test their relative positions. What can you observe? How do you think you could influence the outcome (less/more variables, higher/lower degree, different field)?

[1] there are at least 2 "quick tricks" to check whether at least one of the polynomials is zero, and they do not require you to use a loop, but only some built-in CoCoA functions: find them both!