ECE/CS 5745/6745: Testing and Verification of Digital Circuits
Programming with Singular

Fall 2014, Homework # 3
Due Date: Wed, October 22, 2014

In this assignment, you will implement, within the Singular tool, the symbolic computing algorithms that we have been studying over the past two weeks. Download and install the Singular tool, and *read the manual*. In the manual, you only need to glance through the following sections for now:

1) Section 2 on Introduction, of course.
2) Section 3.1 (usage), 3.3 (rings and orderings), 3.7 and 4.16 (procedures), and section 4.15 for all polynomial related operations and functions.
3) There are some standard procedures (or functions) already implemented in Singular that may be hard to find. So I am providing you with a list of procedures that you may find useful in learning the material.
4) Also uploaded on the class website is a `demo.sing` file with many examples that will be helpful for you to start programming this assignment.

On a unix terminal, the way to load a singular script file is as follows:

```
prompt>> Singular
                     SINGULAR                                 /
 A Computer Algebra System for Polynomial Computations   /   version 3-1-1
                                                       0<
     by: G.-M. Greuel, G. Pfister, H. Schoenemann        \   Feb 2010
FB Mathematik der Universitaet, D-67653 Kaiserslautern    \
> < "demo.sing";
```

In Singular, you can declare rings, term orderings, polynomials, ideals and compute their Gröbner bases. There are many algorithms implemented to compute a Gröbner basis: `std`, `groebner`, `slimgb` are three such commands. There is also a `reduce` command which reduces a polynomial modulo and ideal. See the "demo.sing" file on the class website.

Now, let us get to the assignment. In the class, we have studied division and reduction operations on univariate and multivariate polynomials, and the Buchberger's algorithm, which are mostly based on operations on leading terms. This exercise will provide you with some experience with the Gröbner basis computations. Show your work, and submit the code and the executed run of the experiments.

1) **Univariate Division: (25 points)**
   - First, let us take the case of univariate polynomials in $\mathbb{F}[x]$. Given two polynomials $f, g \in \mathbb{F}[x], g \neq 0$, there exist unique $q, r \in \mathbb{F}[x]$, such that $f = q \cdot g + r$, where either $r = 0$ or $\deg(r) < \deg(g)$. This is the basic division operation. In class, we have seen how q and r are computed by canceling leading terms of f as $f - \frac{lt(f)}{lt(g)} \cdot g$, and corresponding updates to q.
   - Implement the algorithm in the Singular tool, and test it on $f = 2x^5 - 4x^3 + x^2 - x + 2$, and $g = x^2 + x + 1$ in $\mathbb{Q}[x]$, resulting in q, r. Implement this algorithm as a procedure in Singular.
2) **The Euclidean Algorithm: (25 points)**
   - Using the above division algorithm, now design and implement an algorithm to compute $GCD(f_1, f_2)$ in $\mathbb{Q}[x]$. [You will have to find out yourself about the Euclidean algorithm and how it applies to polynomials.]
   - If $f = GCD(f_1, f_2)$, then $\langle f \rangle = \langle f_1, f_2 \rangle$, so there exist $u_1, u_2 \in \mathbb{Q}[x]$ such that $f = u_1 f_1 + u_2 f_2$. Design your algorithm so as to output f, u_1, u_2.
   - Test your algorithm on $f_1 = x^6 - 1, f_2 = x^4 + 2x^3 + 2x^2 - 2x - 3$ in $\mathbb{Q}[x]$.
3) **Multivariate Division: (50 points)**
   - Implement the multi-variate division algorithm that performs the reduction $f \xrightarrow{f_1,\ldots,f_s}_{+} r$.
   - Let $f = x^3 - x^2y - x^2z + x,\ f_1 = x^2y - z,\ f_2 = xy - 1$. Impose a deglex order, $x > y > z$ on $\mathbb{Q}[x, y, z]$. Using your algorithm, compute $r_1 =$ remainder of division by (f_1, f_2), and $r_2 =$ remainder of division by (f_2, f_1). What do you notice?
   - Repeat the above using degrevlex term order.
4) **The Gröbner Basis Algorithm:(100 points)**
   - Now you are asked to implement the Buchberger's algorithm to compute a Gröbner basis G for a given ideal $J = \langle f_1, \ldots, f_s \rangle$. Then you will further compute a reduced, minimal, unique Gröbner basis.
   - Your algorithm should take a set of polynomials as input ($F = \{f_1, \ldots, f_s\}$) and produce the Gröbner basis ($G = \{g_1, \ldots, g_t\}$) as output. Your program should also produce as output:

i) the total number of S-polynomials computed (or the number of iterations); and ii) the number of S-polynomials that result in a non-zero remainder after reduction.

- From the resulting Gröbner basis G, further compute a reduced, canonical Gröbner basis.
- You have to use the multi-variate division (reduction) subroutine that you implemented in the above assignment.
- From Exercise 2.1.1 in the book [1]: Let $f_1 = x^2y - y + x,\ f_2 = xy^2 - x$. Determine the reduced Gröbner basis G of $I = \langle f_1, f_2 \rangle \subset \mathbb{Q}[x, y]$ using your program. First use the lex order with $x > y$, and then repeat the experiment using deglex and degrevlex orders.
- Using your programs, determine whether or not $f \in \langle f_1, f_2 \rangle$, where $f = x^4y - 2x^5 + 2x^2y^2 - 2x^3y - 2x^4 - 2y^3 + 4xy^2 - 3x^2y + 2x^3 - y + 2x$; where f_1, f_2 are as given above. Use any monomial order.

Please write your code modularly, making use of procedural constructs of Singular. Preserve your code, don't lose it. In the next few assignments, we will further build/improve upon your implementation of the Gröbner basis algorithm. We will also apply your algorithms to circuit verification and equivalence checking using Nullstellensatz.

Submission instructions: I will provide you with submission instructions in the next couple of days. I am still trying to find out how source code could be uploaded on Canvas. As soon as I resolve this issue, I will send you an email with submission instructions.

Happy programming!

References

[1] W. W. Adams and P. Loustaunau, *An Introduction to Gröbner Bases.* American Mathematical Society, 1994.