

University of Bath

**DEPARTMENT OF MATHEMATICAL SCIENCES
EXAMINATION**

CM30070: COMPUTER ALGEBRA

Some time in January 2003

You have used an obsolete rubric.

If you are preparing a new exam, please check `examdoc` for the correct arguments to `\papertype`.

1. Define what it means for a computer representation of mathematical objects to be *canonical*.

Assuming a canonical representation for \mathbf{Z} define *carefully* a canonical representation for \mathbf{Q} .

Explain, clearly the data structures needed to represent polynomials in a recursive manner. Give an example. To what extent is your representation canonical?

Explain, clearly the data structures needed to represent polynomials in a distributed manner. Give an example. To what extent is your representation canonical?

Answer.

The representation f of a set of mathematical objects O by a set of computer representations R , i.e. the relation “ $r \in R$ represents $o \in O$ ” is *canonical* if every element of O is represented by precisely one element of R . [2]

Represent the fraction a/b by the pair (a, b) *provided that*:

1. Any non-trivial gcd has been cancelled between a and b ;
2. b is positive.

Several possible representations. One option is, at each level, to represent a polynomial in x with n non-zero coefficients as a list of $2n + 1$ elements:

`(x exp coeff exp coeff ... exp coeff)`

where `exp` represents an exponent (sorted in decreasing order) and `coeff` the corresponding non-zero coefficient, which is either a polynomial in lesser variables, or a coefficient. If the language does not support tagged unions, we may need to “box” the coefficients in some way. The polynomial 0 ends up being represented by the empty list. So with the order $x > y > z$, $x^2(2z+3) + x(4y^2+5yz+6) + 7yz + 8z^2$ would be represented as

`(x 2 (z 1 2 0 3) 1 (y 2 4 1 (z 1 5) 0 6) 0 (y 1 (z 1 7) 0 (z 2 8)))`.

This is canonical provided that:

1. we have a fixed order on the variables;
2. The coefficients themselves are represented canonically. [7]

“Distributed” means regarding the polynomial as a sum of base coefficients times monomials. We need to fix an order on the monomials (many choices!). We also need to fix the number of variables — say x_1, \dots, x_n . Then a “coefficient times monomial”, i.e. “term” can be represented as a $n + 1$ -tuple: non-zero coefficient and the n exponents of the n variables. A polynomial is then a list of terms, sorted in decreasing order. The zero polynomial is then the empty list. Assuming lexicographic order, the example above would be

`((2 2 0 1) (3 2 0 0) (4 1 2 0) (5 1 1 1) (6 1 0 0) (7 0 1 1) (8 0 0 2))`

This is canonical provided that:

1. we have a fixed order on the monomials;
2. The coefficients themselves are represented canonically. [7]

Note to colleagues: this is somewhat similar to question 1 last year, but that wasn't very well answered, and the emphasis on canonical representations is new.

2. What is the main disadvantage of Euclid's algorithm, or its variants such as the sub-resultant algorithm, in computing gcds over $\mathbf{Z}[x]$?

Outline how working modulo a sufficiently large prime can obviate this difficulty. Denoting the Landau–Mignotte bound on the largest coefficient in the gcd of f and g by $\text{LM}(f, g)$, what does “sufficiently large” mean in this context?

Despite the existence of this “large prime” algorithm, what are the main reasons that computer algebra systems tend to use a “many small primes/Chinese Remainder” approach?

Answer.

Intermediate expression swell: one can be forced to work with integers much larger than those that could appear in the gcd. [2]

The following algorithm will work.

1. Choose a sufficiently large prime p .
2. Let h_p be the gcd of f_p and g_p , computed modulo p .
3. Write the coefficients of h_p as integers in the range $(-p/2, p/2)$, calling this polynomial h .
4. If h divides f and g (over \mathbf{Z}), return it as the answer. Care should be taken to do “early abort” if the division produces numbers bigger than $\text{LM}(f, g)$.
5. Otherwise return to step 1, picking a different p .

“Sufficiently large” means greater than $2 \text{LM}(f, g)$, since then all possible coefficients in the gcd are in the range $(-p/2, p/2)$. [10]

There are three main reasons.

1. If the gcd is 1, this should be detected by computations modulo one small prime, with much smaller numbers. Even if the first prime is unlucky, the second prime should detect it . . .
2. $\text{LM}(f, g)$ is generally an over-estimate, and we can often perform an “early detection” step, e.g. if one more prime doesn't change the answer.
3. It generalises (much more readily) to multivariate problems, where the “primes” are then of the form $y - v_i$. [8]

3. Give *three* alternative definitions of a Gröbner base of an ideal.
 Describe Buchberger’s algorithm for computing the Gröbner base of an ideal (you are *not* required to prove termination).
 If the ideal is zero-dimensional (finite number of solutions) what does a *lexicographical* Gröbner base look like?
 Given such a Gröbner base, explain how the Gianni–Kalkbrener theorem lets one write down the finite number of solutions.

Answer.

Possible ones are:

1. $S(g_i, g_j) \rightarrow_G 0$;
2. $Ideal(LM(G)) = Ideal(LM(I))$;
3. \rightarrow_G is confluent;
4. $f \in I \Leftrightarrow f \rightarrow_G = 0$. [3]

Bookwork. [7]

Bookwork. [4]

The following algorithm works, but see the gloss at end.

1. Compute all roots of $p_n(x_n)$.
2. For each such root, substitute into the $p_{n-1,i}(x_n, x_{n-1})$ (in order of increasing degree in x_{n-1}). The first one not to vanish gives you *precisely all* (this is the use of Gianni-Kalkbrener) the corresponding values of x_{n-1} .
3. For each such pair of values for (x_n, x_{n-1}) substitute into the $p_{n-2,i}(x_n, x_{n-1}, x_{n-2})$ (in order of increasing degree in x_{n-2}). The first one not to vanish gives you *precisely all* (this is the use of Gianni-Kalkbrener) the corresponding values of x_{n-2} .
4. ...

This can in fact be done with no “solving” — represent the roots of p_n via a `RootOf` construct, and take gcds with the coefficients of p_{n-1} to split as necessary. In fact Gianni–Kalkbrener implies that the leading coefficient is sufficient. [6]

4. If $p(x)/q(x)$ is a rational function in x with integer coefficients, what is the general shape of the integral (with respect to x) of $p(x)/q(x)$?

Give an algorithm which will compute the integral, given $p(x)/q(x)$, and which will not introduce any non-rational numbers if they are not needed.

Where does this algorithm break down when we have logarithms or exponentials involved?

Answer.

$\int \frac{p}{q} = \frac{r}{s} + \sum c_i \log v_i(x)$, where $\frac{r}{s}$ is another rational function with integer coefficients, the c_i are algebraic numbers, and the v_i are polynomials with coefficients in the field $\mathbf{Q}(c_i)$. [4]

Essentially bookwork.

1. Write p/q as “polynomial + proper rational function” and integrate the polynomial part. This is division and polynomial integration, so doesn’t leave \mathbf{Q} .
2. Square-free decompose q as $\prod q_i^{i_i}$. Gcd computations, so rational.
3. Perform a partial fraction decomposition, to get $\sum \frac{p_i}{q_i^{i_i}}$. Extended Euclidean, so again rational.
4. For each of these, write $aq_i + bq_i' = 1$ (Extended Euclid, so rational), then $\int \frac{p_i}{q_i^{i_i}} = \int \frac{p_i(aq_i + bq_i')}{q_i^{i_i}}$. This splits as $\int \frac{ap_i}{q_i^{i_i-1}}$ and $\int \frac{bp_i q_i'}{q_i^{i_i}}$. The second can be integrated by parts to give $\frac{-bp_i}{(i-1)q_i^{i_i-1}} + \int \frac{(bp_i)'}{(i-1)q_i^{i_i-1}}$. All this is rational, and we can continue reducing the exponent of q_i until it reaches 1.
5. We are now integrating a sum of expressions of the form $\frac{r_i}{q_i}$ where q_i is square free, and this is what gives the $\sum c_i \log v_i(x)$ term. Call the integrand r/q (in fact this step is not necessary, as algebraic numbers cannot cancel between different summands, but this is not proved in the course). In the integral, we can assume:
 - (a) the v_i are polynomials;
 - (b) the v_i are square-free;
 - (c) the v_i are relatively prime;
 - (d) the c_i are distinct;

none of which changes any rationality assumptions. So $\int \frac{r}{q} = \sum \frac{c_i v_i'}{v_i}$ and the assumptions above mean that no cancellation takes place in this expression. It is then possible to deduce that $v_i = \gcd(r - c_i q', q)$ and the c_i are *precisely* the values for which this gcd is non-trivial. The c_i are therefore the roots of the corresponding resultant, and, if they are all rational, we can compute the corresponding v_i by gcd computation, again purely rational.

Of course, Horowitz–Ostrowski would be equally feasible. [12]

The polynomial (or Laurent polynomial) parts gets more complicated, but that’s not the real breakdown. The problem is that the c_i roots of the resultant will be independent of the principal monomial, but may not be independent of x , and therefore are not the constants that Liouville’s principle requires. [4]