

Φ -Gröbner Bases

After lectures by Professor Lawrence

James H. Davenport
J.H.Davenport@bath.ac.uk

March 15, 2009

1 Introduction

Notation 1 *In principle, we are interested in $R = \mathbf{Z}[t_1, \dots]$, but in order to make polynomials monic etc. we will generally work over \mathbf{Q} . If we need to be precise¹ we will use $R_{\mathbf{Z}}$ or $R_{\mathbf{Q}}$. Let \mathbf{M} be the monomials of R .*

Note that R is *not* noetherian.

Notation 2 *Let Φ be the monoid of all order-preserving maps from \mathbf{P} into itself ($\mathbf{P} =$ positive integers). For $\phi \in \Phi$, extend it by $\phi(t_i) = t_{\phi(i)}$, and hence to a monomorphism of R into itself.*

Definition 1 *A Φ -ideal of R is an ideal in the usual sense which is also closed under all elements of Φ , i.e.*

$$\forall a(t_1, \dots, t_n) \in I; \quad \phi(a) = a(t_{\phi(1)}, \dots, t_{\phi(n)}) \in I$$

Definition 2 *For monomials a and b we say that a Φ -divides b if there are $\phi \in \Phi$ and $c \in \mathbf{M}$ such that $b = \phi(a)c$. Write $a \phi b$.*

Example 1 *c may² not be unique: consider $a = t_1 t_2$ and $b = t_1 t_2 t_3$. Then c can be any of t_1, t_2 or t_3 .*

Theorem 1 *In \mathbf{M} , each of the following is true:*

1. $a \phi b$ and $b \phi c$ implies $a \phi c$;
2. $a \phi b$ and $b \phi a$ implies $a = b$;
3. $a \phi b$ implies $a \phi bc$

Example 2 $t_1 t_2^2 \phi t_2 t_4^2$ but $t_1 t_2^2 t_3$ does not $\phi t_2 t_4^2 t_3$; so in general $a \phi b$ does not imply $a c \phi b c$.

¹JHD: Do we ever?

²JHD:

Theorem 2 (Dickson's Lemma) *If a monoid M has a positive admissible partial order \leq , and M is generated by X , and $\leq|_X$ is Dickson, then so is \leq .*

The proof is essentially by divisibility.

Definition 3 *Let \leq^* be a total order on \mathbf{M} . We say that \leq^* is a Φ -monomial order if:*

1. $1 \leq^* m \forall m \in \mathbf{M}$;
2. $m_1 \leq^* m_2$ implies $m_1 m \leq^* m_2 m$, $mm_1 \leq^* mm_2$;
3. $m \leq^* \phi(m) \forall \phi$;
4. $m_1 \leq^* m_2$ implies $\phi(m_1) \leq^* \phi(m_2) \forall \phi$.

An example would be deglex with $t_1 < t_2 < \dots$.

Notation 3 *For a fixed \leq^* , we can define the leading monomial of a polynomial. Let $\Psi : R \mapsto \mathbf{M}$ be the leading monomial map.*

Suppose I is a non-zero ideal of R . Then $\Psi(I) \subset \mathbf{M}$ and has a Dickson basis under the order induced by Φ -divisibility, which we will call \leq .

Definition 4 *A finite set s of monic polynomials in I is a Φ -Gröbner basis for I if $\Psi(S) \supset$ the Dickson basis for $\Psi(I)$.*

It will turn out that S will Φ -generate I .³

Definition 5 (Φ -reduction) ⁴ *The aim is to reduce b by a , writing $b = \phi(a)q + r$.*

1. If $\Psi(a) \leq \Psi(b)$, write $\Psi(b) = \phi(\Psi(a))m$ with⁵ m minimal used \leq^* , and write $r = b - \phi(a)m$;
2. Otherwise write $b = 0a + b$.

Definition 6 *Let a_1, \dots, a_k be an ordered set of monic polynomials from R . Write*

$$\begin{aligned}
 b &= \phi_1(a_1) + r_1 \\
 r_1 &= \phi_2(a_2) + r_2 \\
 \dots &= \dots \\
 r_{k-1} &= \phi_k(a_k) + r_k \\
 r_k &= \phi_{k+1}(a_1) + r_{k+1} \\
 \dots &= \dots
 \end{aligned}$$

We have $b \geq^ r_1 \geq^* r_2 \dots$, so this sequence stabilises at some r_n , call this the Φ -reduction of b by $[a_1, \dots, a_k]$, denoted $R(b, [a_1, \dots, a_k])$.*

³JHD: In response to a question of JHD.

⁴JHD: Prof Lawrence called it Φ -division, but it's more akin to reduction.

⁵JHD: This is useful to avoid ambiguity, but is probably unnecessary.

Theorem 3 *Let I be a Φ -ideal with Φ -Gröbner base a_1, \dots, a_k . Then $b \in I$ if, and only if, $R(b, [a_1, \dots, a_k]) = 0$.*

\Rightarrow If $b \in I \setminus \{0\}$, we will use \leq^* -induction on $\psi(b)$. There is⁶ an a_j such that $\psi(a_j) \phi \psi(b)$ by definition of a Φ -Gröbner base. So we can write $b = \phi(a)q + r$. Then $r \leq^* b$ strictly, but $r \in I$, so we have the necessary contradiction.

\Leftarrow obvious — reverse the reduction.

Application 1 *Let S be any commutative ring. Let $I = \{q(t_1, \dots, t_n) \in \mathbf{Z}[t_1, \dots, t_n] \text{ such that } q = 0 \text{ is an identity on } S\}$ This is a Φ -ideal, hence the equivalence problem for S is decidable.*

Application 2 *Let G be a metabelian group, with $N \triangleleft G$ and $K \leq G$, both N and K abelian. So $NK = G$, $N \cap K = \{e\}$.*

Let $W(X_1, \dots, X_n)$ be a word over G . Replace X_i by $U_i V_i$ where, conceptually, the V_i “range over” N and the U_i “range over” K .

We recall that, of $A \triangleleft H$ with A abelian, then A is a $\mathbf{Z}[H/A]$ -module, with $\bar{h} \cdot a = hah^{-1}$, for $a \in A$, $\bar{h} \in H/A$, which we will write $a^{\bar{h}}$, and then

$$a^{\sum n_i \bar{h}_i} = \prod (a^{n_i})^{\bar{h}_i}.$$

Then

$$W(V_1 U_1, \dots, V_n U_n) = V_1^{P_1(U_1)} V_2^{P_2(U_2)} \dots V_n^{P_n(U_n)} \prod U_i,$$

where the last product “is in” K , and the rest “in” N . We need to decide if $P_j(U_j) \in I \forall j$, and this is decidable.

Hence the term equivalence problem for a split emtabelian group is decidable.

Example 3 *Consider the Φ -ideal $\langle t_1 t_2 - 1 \rangle$. Its minimal Φ -Gröbner base is $\{t_2 - t_1, t_1^2 - 1\}$, i.e. a two-generator set. This is demonstrated, using the methodology described below, in figure 1.*

To computed Φ -Gröbner bases, we need to consider, not just S -polynomials, but rather \mathcal{S} -polynomials. Suppose p, q involve only t_1, \dots, t_n . Consider all order-preserving maps from $1 \dots n$ to $1 \dots 2n$.

Claim 4 *t is sufficient to consider, for each pair of such $\langle \phi_1, \phi_2 \rangle$, the normal S -polynomial of $\phi_1(p), \phi_2(q)$, in other words the reduction of the unification of $\phi_1(\psi(p)), \phi_2(\psi(q))$.*

Example 4 *What happens if we consider a variant such as $\langle t_1 t_2 + 2 \rangle$. In fact we get $\{t_2 - t_1, t_1^2 - 2\}$, as in figure 2. In fact, in both cases t_4 is unnecessary. Similar behaviour can be observed in the case of three variables: figure 3.*

⁶JHD: We probably ought to choose j minimal, but it probably doesn't really matter.

Figure 1: Example 3 worked in Maple

```

> lphi:={t1=u1,t2=u2},{t1=u1,t2=u3},{t1=u2,t2=u3},{t1=u1,t2=u4},
      {t1=u2,t2=u4},{t1=u3,t2=u4};
lphi:=[{t1=u1,t2=u2},{t1=u1,t2=u3},{t1=u2,t2=u3},{t1=u1,t2=u4},{t1=u2,t2
      =u4},{t1=u3,t2=u4}] (1)

> orig:=t1*t2-1;
      orig := t1 t2 - 1 (2)

> new:=map(x->eval(orig,x),lphi);
      new := [u1 u2 - 1, u1 u3 - 1, u2 u3 - 1, u1 u4 - 1, u2 u4 - 1, u3 u4 - 1] (3)

> with(Groebner):
> Basis(new,grlex(u4,u3,u2,u1));
      [u2 - u1, u3 - u1, u4 - u1, -1 + u1^2] (4)

> # and we note that u2-u1 phi-divides the next two.
> [%[1],%[4]];
      [u2 - u1, -1 + u1^2] (5)

```

Figure 2: Example 4 worked in Maple

```

> lphi:=[{t1=u1,t2=u2},{t1=u1,t2=u3},{t1=u2,t2=u3},{t1=u1,t2=u4},
{t1=u2,t2=u4},{t1=u3,t2=u4}];
lphi:= [{t1 = u1, t2 = u2}, {t1 = u1, t2 = u3}, {t1 = u2, t2 = u3}, {t1 = u1, t2 = u4}, {t1 = u2, t2
= u4}, {t1 = u3, t2 = u4}] (1)

> orig:=t1*t2-2;
orig := t1 t2 - 2 (2)

> new:=map(x->eval(orig,x), lphi);
new := [u1 u2 - 2, u1 u3 - 2, u2 u3 - 2, u1 u4 - 2, u2 u4 - 2, u3 u4 - 2] (3)

> with(Groebner):
> Basis(new, grlex(u4, u3, u2, u1));
[u2 - u1, u3 - u1, u4 - u1, -2 + u12] (4)

> # and we note that u2-u1 phi-divides the next two.
> [%[1], %[4]];
[u2 - u1, -2 + u12] (5)

```

Figure 3: Example in more variables worked in Maple

```

> with(Groebner):
> llphi:={t1=u1,t2=u2,t3=u3},{t1=u1,t2=u2,t3=u4},{t1=u1,t2=u3,t3=
u4},{t1=u2,t2=u3,t3=u4},{t1=u1,t2=u2,t3=u5},{t1=u1,t2=u3,t3=u5},
{t1=u2,t2=u3,t3=u5},{t1=u1,t2=u4,t3=u5},{t1=u2,t2=u4,t3=u5},{t1=
u3,t2=u4,t3=u5},{t1=u1,t2=u2,t3=u6},{t1=u1,t2=u3,t3=u6},{t1=u2,
t2=u3,t3=u6},{t1=u1,t2=u4,t3=u6},{t1=u2,t2=u4,t3=u6},{t1=u3,t2=
u4,t3=u6},{t1=u1,t2=u5,t3=u6},{t1=u2,t2=u5,t3=u6},{t1=u3,t2=u5,
t3=u6},{t1=u4,t2=u5,t3=u6}};
llphi := [ {t1 = u1, t2 = u2, t3 = u3}, {t1 = u1, t2 = u2, t3 = u4}, {t1 = u1, t2 = u3, t3 = u4}, {t1 = u2, t2 = u3, t3 = u4}, {t1 = u1, t2 = u2, t3 = u5}, {t1 = u1, t2 = u3, t3 = u5}, {t1 = u2, t2 = u3, t3 = u5}, {t1 = u1, t2 = u4, t3 = u5}, {t1 = u2, t2 = u4, t3 = u5}, {t1 = u3, t2 = u4, t3 = u5}, {t1 = u1, t2 = u2, t3 = u6}, {t1 = u1, t2 = u3, t3 = u6}, {t1 = u2, t2 = u3, t3 = u6}, {t1 = u1, t2 = u4, t3 = u6}, {t1 = u2, t2 = u4, t3 = u6}, {t1 = u3, t2 = u4, t3 = u6}, {t1 = u1, t2 = u5, t3 = u6}, {t1 = u2, t2 = u5, t3 = u6}, {t1 = u3, t2 = u5, t3 = u6}, {t1 = u4, t2 = u5, t3 = u6} ] (1)
> origg:=t1*t2*t3-1;
origg := t1 t2 t3 - 1 (2)
> neww:=map(x->eval(origg,x),llphi);
neww := [ u1 u2 u3 - 1, u1 u2 u4 - 1, u1 u3 u4 - 1, u2 u3 u4 - 1, u1 u2 u5 - 1, u1 u3 u5 - 1, u2 u3 u5 - 1, u1 u4 u5 - 1, u2 u4 u5 - 1, u3 u4 u5 - 1, u1 u2 u6 - 1, u1 u3 u6 - 1, u2 u3 u6 - 1, u1 u4 u6 - 1, u2 u4 u6 - 1, u3 u4 u6 - 1, u1 u5 u6 - 1, u2 u5 u6 - 1, u3 u5 u6 - 1, u4 u5 u6 - 1 ] (3)
> nb:=Basis(neww,grlex(u6,u5,u4,u3,u2,u1));
nb := [ u2 - u1, u3 - u1, u4 - u1, u5 - u1, u6 - u1, -1 + u1^3 ] (4)
> # again, the first phi-equals most of the rest
> [nb[1],nb[6]];
[ u2 - u1, -1 + u1^3 ] (5)
> # do we really need all these extra variables?
> llphi[1..4];
[ {t1 = u1, t2 = u2, t3 = u3}, {t1 = u1, t2 = u2, t3 = u4}, {t1 = u1, t2 = u3, t3 = u4}, {t1 = u2, t2 = u3, t3 = u4} ] (6)
> newww:=map(x->eval(origg,x),%);
newww := [ u1 u2 u3 - 1, u1 u2 u4 - 1, u1 u3 u4 - 1, u2 u3 u4 - 1 ] (7)
> nbb:=Basis(newww,grlex(u6,u5,u4,u3,u2,u1));
nbb := [ u2 - u1, u3 - u1, u4 - u1, -1 + u1^3 ] (8)
> # so in fact one new variable is sufficient

```