Gröbner Bases

This is just meant to be a thought dump on Gröbner bases. My goal is for the following discussion to be such that it would lead one to the definition of a Gröbner basis. I'm not going to talk about monomial ordering. This discussion is for someone who has already done division in $k[x]$, and is familiar with the division algorithm in $k[x_1, \dots, x_n]$.

Let's start with a simple example to show that there is a problem with the extended division algorithm. Let's try to see if $f = xy^2 - x$ is divisible by $(f_1 = xy - 1, f_2 = y^2 - 1)$[1] If we followed the order f_1, f_2, then we'd find that

$$xy^2 - x = y(xy - 1) + (-x + y),$$

giving us a remainder of $y - x$. However, if we tried to divide by (f_2, f_1), we get

$$xy^2 - x = x(y^2 - 1) + 0,$$

giving us a remainder of 0. Now, this non-uniqueness of remainder is clearly something we'd like to remedy. Trying to suggest that (f_2, f_1) is somehow a better order than (f_1, f_2) is a non-starter because we can easily construct an example where choosing (f_2, f_1) would give us a non-zero remainder, while choosing (f_1, f_2) would give us a remainder of 0.

Idea 1. *Suppose we are trying to divide by $\{f_i\}$. If we find $\{g_i\}$ such that, for each*

$$g_i = \sum h_j f_j,$$

and it turns out that having dividing by g_i is easier, then we are in better shape. This is because if we use g_i to divide, it is as good as dividing by $\{f_i\}$. ***This immediately suggests that we employ the language of ideals.***

In the language ideals, checking if $\{f_i\}|f$ is equivalent to checking if $f \in \langle\{f_i\}\rangle$. However, even with the new language, we haven't made any progress. $\{f_i\}$ is the generating set of the new ideal, and we are no better off than before. What we need is a better generating set/basis.

In the language of ideals, what was the problem with the example shown earlier? When we use the order (f_1, f_2), we obtain $-x + y$ as a remainder. The problem is that we couldn't go any further by using the division algorithm. Say we somehow knew that

$$-x + y = yf_1 - xf_2.$$

Then we'd have no trouble deducing that our f that we began with was indeed divisible by $\{f_1, f_2\}$. Our problem was that $LT(x + y)$ was not divisible by either $LT(f_1)$ or $LT(f_2)$.

Note: $-x + y$ was formed with multiplying both f_1 and f_2 with monomial so as to form the LCM of the leading terms of f_1 and f_2, and then cancelling it out. In fact, the notion of an S-polynomial comes from this exact observation, and in turn leads us to Buchberger's algorithm.

Idea 2. *This leads us to realize that we need a generating set $\{g_i\}_{i\in[s]}$ of our ideal $I = \langle f_1, f_2\rangle$ that is such that for any $f \in I$, we have that there exists an $i^* \in [s]$ such that $LT(g_{i^*}) \mid LT(f)$.*

[1] The $(\cdot, \cdot, \dots)$ notation is meant to denote ordered tuples.

Idea 2 is basically the definition of a Gröbner basis. A Gröbner basis is defined exactly as what is needed according to Idea 2. By ensuring that we have a g_{i^*} whose leading term divides $LT(f)$ for all $f \in I$, we are certain to always be able to get a remainder of 0 when $f \in I$. This is because anything that is leftover after subtracting a multiple of g_i from f is in the same congruence class of f, modulo I.

It is fairly reasonable to assume that one would get upto to this point, but how does one even hope that such a generating would exist? How can we even hope that it would be finite? Keep in mind that Buchberger's algorithm actually tells you *how to find it.* The transition from wondering if such a generating set could even exist to actually being able to always find it is possible because of an equivalent definition of a Gröbner basis.

Definition 1. *Define $LT(I) = \{LT(f) \mid f \in I\}$. The set $\{g_i\}_{i\in[s]} \subseteq I$ is a Gröbner basis if $I = \langle \{g_i\}_{i\in[s]}\rangle$ and*

$$\langle LT(I)\rangle = \langle \{LT(g_i) \mid i \in [s]\}\rangle.$$

That this definition is equivalent to what we expressed in Idea 2 is an exercise left to the reader. I am more interested in discussing how one would have arrived at this language. Apriori, I feel it is extremely unnatural to define $\langle LT(I)\rangle$ and $\langle \{LT(g_i) \mid i \in [s]\}\rangle$. How then?

Remember, we want for all $f \in I$, $LT(f) = LT(g_i) * m$ for some i and m a monomial. So naturally we would take a look at the set $LT(I)$. Let us form all possible $LT(g_i) * m$. It is just the set

$$\{LT(g_i)LT(f) \mid i \in [s], f \in k[x_1, \ldots, x_n]\}.$$

We want the above set to contain $LT(I)$, i.e. we want $\{LT(g_i)LT(f) \mid i \in [s], f \in k[x_1, \ldots, x_n]\} \supseteq LT(I)$. Since $g_i \in I$, we have that $g_iLT(f) \in I$ for all $f \in k[x_1, \ldots, x_n]$. This in turn means that $LT(g_i)LT(f)$ will be in $LT(I)$ thus proving that $\{LT(g_i)LT(f) \mid i \in [s], f \in k[x_1, \ldots, x_n]\} \subseteq LT(I)$. Thus we can say that the necessity expressed in Idea 2 is equivalent to needing

$$\{LT(g_i)LT(f) \mid i \in [s], f \in k[x_1, \ldots, x_n]\} = LT(I).$$

Since $\{LT(g_i)LT(f) \mid i \in [s], f \in k[x_1, \ldots, x_n]\}$ and $LT(I)$ are the same as sets, we have that

$$\langle \{LT(g_i)LT(f) \mid i \in [s], f \in k[x_1, \ldots, x_n]\}\rangle = \langle LT(I)\rangle.$$

Finally, $\langle \{LT(g_i)LT(f) \mid i \in [s], f \in k[x_1, \ldots, x_n]\}\rangle$ may as well be written as $\langle \{LT(g_i) \mid i \in [s]\}\rangle$, thus giving us the condition in Definition 1.

N.B. Actually the above discussion only proves that $\langle LT(I)\rangle = \langle \{LT(g_i) \mid i \in [s]\}\rangle$ is a sufficient condition for what is expressed in Idea 2. It is also a necessary condition, but that needs a proof. Proving it is a necessary condition requires the observation that if we have a monomial $m \in I = \langle \{m_i\}\rangle$, where m_i are also monomials, then there exists an i^* such that $m_{i^*} \mid m$.