Interpolation

In this supplementary section we will discuss an important polynomial technique called *interpolation* and see some of its uses.

Proposition 1 (Lagrange Interpolation)**.** *Let R be a domain with field of fractions F. Let (a_i, b_i), $0 \leq i \leq d$, be a list of $d+1$ pairs of elements of R such that the a_i are all distinct. Then there is a unique polynomial $q(x) \in F[x]$ of degree at most d such that $q(a_i) = b_i$ for each i.*

Proof. Define the polynomial $q(x) \in F[x]$ as follows:

$$q(x) = \sum_{i=0}^{d} b_i \prod_{j \neq i} \frac{x - a_j}{a_i - a_j}.$$

Certainly q has degree at most d, since it is a sum of polynomials of degree exactly d. Now consider the value of q at a particular a_k. In the product $\prod_{j \neq i}(a_k - a_j)/(a_i - a_j)$, if j is ever equal to k then we have 0 as a factor. Thus the only term of this sum which contributes to $q(a_k)$ is $i = k$. So

$$q(a_k) = b_k \prod_{j \neq k} \frac{a_k - a_j}{a_k - a_j} = b_k$$

as claimed.

To see that q is unique, use Corollary (blah). □

1. **A Factorization Algorithm.** In this exercise we will use interpolation to construct an algorithm capable of factoring polynomials over certain UFDs. (This algorithm will not be very efficient in practice, but it is correct and simple. Although much faster factorization algorithms exist, they are also more complicated. Given two algorithms to solve the same problem, one slow and simple and one fast and complicated, we typically prefer the fast to the slow. But slow, simple algorithms do have an important role to play: they can be used as an easy-to-verify "reference implementation" for the more complicated algorithm, which may be harder to verify directly. We can implement both algorithms in software and use both to solve lots of problems; if the results always agree, we can have more confidence that the fast algorithm is correct on larger problems for which the slow implementation is not practical. This is a useful engineering technique.)

2. **Secret Sharing.** A *secret sharing scheme* is a method of dividing information into incomplete parts, called *shares*, so that the information can only be retrieved if one is in possession of all (or enough) of the shares. Polynomial interpolation can be used to design a secret sharing scheme.