

Math 701 Fall 2026, Lecture 3: The classical families e, h, p

website: <https://www.cip.ifi.lmu.de/~grinberg/t/26fs/>

We continue where we left off in Lecture 2.

Proof of the identity (1) from Lecture 2. Parts (a) and (c) of Proposition 1.2.3 yield

$$\prod_{i=1}^N (1 - tx_i) = \sum_{n \in \mathbb{N}} (-1)^n t^n e_n \quad \text{and} \\ \prod_{i=1}^N \frac{1}{1 - tx_i} = \sum_{n \in \mathbb{N}} t^n h_n$$

in the power series ring $\mathcal{P}[[t]]$. Multiplying these two equalities, we find¹

$$\begin{aligned} 1 &= \left(\sum_{n \in \mathbb{N}} (-1)^n t^n e_n \right) \left(\sum_{n \in \mathbb{N}} t^n h_n \right) \\ &= \left(\sum_{i \in \mathbb{N}} (-1)^i t^i e_i \right) \left(\sum_{j \in \mathbb{N}} t^j h_j \right) \\ &= \sum_{i \in \mathbb{N}} \sum_{j \in \mathbb{N}} (-1)^i \underbrace{t^i e_i t^j h_j}_{= e_i h_j t^{i+j}} \\ &= \sum_{n \in \mathbb{N}} \sum_{\substack{(i,j) \in \mathbb{N}^2; \\ i+j=n}} (-1)^i e_i h_j \\ &= \sum_{n \in \mathbb{N}} \sum_{\substack{(i,j) \in \mathbb{N}^2; \\ i+j=n}} (-1)^i e_i h_j \underbrace{t^{i+j}}_{\substack{= t^n \\ (\text{since } i+j=n)}} \\ &= \sum_{n \in \mathbb{N}} \left(\sum_{\substack{(i,j) \in \mathbb{N}^2; \\ i+j=n}} (-1)^i e_i h_j \right) t^n. \end{aligned}$$

¹All our manipulations with power series, such as expanding products of infinite sums or interchanging summation signs, are justified by the coefficientwise topology on the ring of formal power series. This is the topology with respect to which a sequence $(f_0, f_1, f_2, \dots)$ (or, more generally, a net $(f_d)_{d \in D}$) of power series converges to a power series f if and only if for each $n \in \mathbb{N}$, all sufficiently high $d \in \mathbb{N}$ (or, in the case of a net, all sufficiently high $d \in D$) satisfy $[t^n] f_d = [t^n] f$ (where $[t^n] g$ denotes the t^n -coefficient of a power series g). The “sufficiently high” here can depend on n . All our sums converge with respect to this topology. For a more elementary explanation of this, see [21s, Chapter 3] (for the univariate case).

Comparing t^n -coefficients, we see that each $n \in \mathbb{N}$ satisfies

$$\begin{cases} 1, & \text{if } n = 0; \\ 0, & \text{if } n > 0 \end{cases} = \sum_{\substack{(i,j) \in \mathbb{N}^2; \\ i+j=n}} (-1)^i e_i h_j = \sum_{i=0}^n (-1)^i e_i h_{n-i}$$

(since the pairs $(i, j) \in \mathbb{N}^2$ satisfying $i + j = n$ are simply the pairs $(i, n - i)$ for $i \in \{0, 1, \dots, n\}$). If n is a positive integer, then the left-hand side of this equality is simply 0, so the whole equality rewrites as

$$0 = \sum_{i=0}^n (-1)^i e_i h_{n-i} = \sum_{j=0}^n (-1)^j e_j h_{n-j}.$$

Thus, (1) is proved. $\square$

The proofs of the other two Newton–Girard identities (that is, (2) and (3)) rely on the notion of the **logarithmic derivative** of an FPS²:

Definition 1.2.4. For any FPS $f \in K[[t]]$ with constant term 1, define the **logarithmic derivative** $\text{loder } f$ of f by

$$\text{loder } f = \frac{f'}{f} \in K[[t]].$$

(As we recall, f' denotes the derivative of f with respect to t . Division by f is allowed since f has constant term 1.)

Proposition 1.2.5 (rules for logarithmic derivatives). Logarithmic derivatives satisfy the following rules:

(a) For any two FPSs f and g with constant terms 1, we have

$$\text{loder } (fg) = \text{loder } f + \text{loder } g.$$

(b) For any k FPSs $f_1, f_2, \dots, f_k$ with constant terms 1, we have

$$\text{loder} \left(\prod_{i=1}^k f_i \right) = \sum_{i=1}^k \text{loder } f_i.$$

(c) For any FPS f with constant term 1, we have

$$\text{loder} \frac{1}{f} = -\text{loder } f.$$

²The abbreviation “FPS” means “formal power series”.

- (d) If K is a commutative $\mathbb{Q}$ -algebra, then any FPS f with constant term 1 satisfies

$$\text{loder } f = (\log f)'.$$

Here, $\log f$ denotes the composition of the FPSs $\log(1+t) = \sum_{n \geq 1} \frac{(-1)^{n-1}}{n} t^n$ and $f-1$ (for details, see [21s, Definition 3.7.6], where $\log$ is denoted by Log).

Proof. (a) By the definition of a logarithmic derivative, we have

$$\begin{aligned} \text{loder}(fg) &= \frac{(fg)'}{fg} = \frac{f'g + fg'}{fg} && \text{(by the product rule)} \\ &= \underbrace{\frac{f'}{f}}_{=\text{loder } f} + \underbrace{\frac{g'}{g}}_{=\text{loder } g} = \text{loder } f + \text{loder } g. \end{aligned}$$

This proves part (a).

(b) Induct on k . The *base case* ($k = 0$) is just saying $\text{loder } 1 = 0$, which is clear (since $1' = 0$). The *induction step* uses part (a).

(c) Part (a) shows that loder is a group morphism from the multiplicative group of FPSs with constant term 1 to the additive group of all FPSs. But group morphisms respect inverses. So $\text{loder } \frac{1}{f} = -\text{loder } f$.

(d) Quick and dirty argument: the chain rule yields

$$(\log f)' = \underbrace{\log' f}_{=\frac{1}{f}} \cdot f' = \frac{1}{f} \cdot f' = \frac{f'}{f} = \text{loder } f.$$

“But wait”, you may say, “is this allowed when $\log$ is not an actual FPS?”. And indeed, this calculation is not quite kosher for this exact reason; instead, we have to decompose $\log f$ as the composition of the actual FPSs $\log(1+t)$ and $f-1$ and then apply the chain rule to these two FPSs. You are invited to make this change yourself, or look up the proof in [21s, Proposition 3.7.13]. $\square$

Proof of (3) from Lecture 2. We shall apply Proposition 1.2.5 with $\mathcal{P}$ in place of K .

Proposition 1.2.3 (c) in Lecture 2 says that

$$\prod_{i=1}^N \frac{1}{1 - tx_i} = \sum_{n \in \mathbb{N}} t^n h_n.$$

Applying the logarithmic derivative to this equality, we obtain

$$\text{loder} \prod_{i=1}^N \frac{1}{1-tx_i} = \text{loder} \left(\sum_{n \in \mathbb{N}} t^n h_n \right) = \frac{\left(\sum_{n \in \mathbb{N}} t^n h_n \right)'}{\sum_{n \in \mathbb{N}} t^n h_n} = \frac{\sum_{n>0} n t^{n-1} h_n}{\sum_{n \in \mathbb{N}} t^n h_n},$$

since the definition of a derivative yields $\left(\sum_{n \in \mathbb{N}} t^n h_n \right)' = \sum_{n>0} n t^{n-1} h_n$. Thus,

$$\begin{aligned} \frac{\sum_{n>0} n t^{n-1} h_n}{\sum_{n \in \mathbb{N}} t^n h_n} &= \text{loder} \prod_{i=1}^N \frac{1}{1-tx_i} \\ &= \sum_{i=1}^N \underbrace{\text{loder} \frac{1}{1-tx_i}}_{\substack{=\text{loder}((1-tx_i)^{-1}) \\ =-\text{loder}(1-tx_i) \\ \text{(by Proposition 1.2.5 (c))}}} &\quad \text{(by Proposition 1.2.5 (b))} \\ &= - \sum_{i=1}^N \underbrace{\text{loder}(1-tx_i)}_{\substack{=(1-tx_i)' \\ =\frac{-x_i}{1-tx_i}}} = - \sum_{i=1}^N \frac{-x_i}{1-tx_i} \\ &= \sum_{i=1}^N \underbrace{\frac{x_i}{1-tx_i}}_{\substack{=x_i(1+tx_i+(tx_i)^2+\dots) \\ \text{(by the geometric series formula)}}} = \sum_{i=1}^N x_i \underbrace{\left(1+tx_i+(tx_i)^2+\dots\right)}_{\substack{=x_i+tx_i^2+t^2x_i^3+\dots \\ =t^0x_i^1+t^1x_i^2+t^2x_i^3+t^3x_i^4+\dots}} \\ &= \sum_{i=1}^N \left(t^0 x_i^1 + t^1 x_i^2 + t^2 x_i^3 + t^3 x_i^4 + \dots \right) \\ &= t^0 \underbrace{\sum_{i=1}^N x_i^1}_{=p_1} + t^1 \underbrace{\sum_{i=1}^N x_i^2}_{=p_2} + t^2 \underbrace{\sum_{i=1}^N x_i^3}_{=p_3} + t^3 \underbrace{\sum_{i=1}^N x_i^4}_{=p_4} + \dots \\ &= t^0 p_1 + t^1 p_2 + t^2 p_3 + t^3 p_4 + \dots = \sum_{n>0} t^{n-1} p_n. \end{aligned}$$

That is,

$$\begin{aligned}
 \sum_{n>0} n t^{n-1} h_n &= \left(\sum_{n>0} t^{n-1} p_n \right) \left(\sum_{n \in \mathbb{N}} t^n h_n \right) = \left(\sum_{j>0} t^{j-1} p_j \right) \left(\sum_{i \in \mathbb{N}} t^i h_i \right) \\
 &= \sum_{j>0} \sum_{i \in \mathbb{N}} \underbrace{t^{j-1} p_j t^i h_i}_{=t^{i+j-1} p_j h_i} = \sum_{j>0} \sum_{i \in \mathbb{N}} t^{i+j-1} p_j h_i = \sum_{j>0} \sum_{n \geq j} \underbrace{t^{n-j+j-1} p_j h_{n-j}}_{=t^{n-1}} \\
 &= \sum_{n>0} \sum_{j=1}^n t^{n-1} p_j h_{n-j} = \sum_{n>0} \left(\sum_{j=1}^n p_j h_{n-j} \right) t^{n-1}.
 \end{aligned}$$

(here, we substituted $n - j$ for i in the inner sum)

Comparing t^{n-1} -coefficients, we get

$$n h_n = \sum_{j=1}^n p_j h_{n-j} = \sum_{j=1}^n h_{n-j} p_j.$$

This is precisely the 3rd Newton–Girard identity (3). □

The proof of the 2nd Newton–Girard identity (2) is similar: a bit easier since it does not require Proposition 1.2.5 (c), but a bit messier since signs appear. Note that all three Newton–Girard identities can also be proved combinatorially or by induction (this is a homework problem).

The Newton–Girard identities can be used to express the e 's, the h 's and the p 's in terms of each other; you just need to solve the identities recursively for each respective sequence. The only wrinkle is that to express the e 's or the h 's in terms of the p 's, you need to divide by positive integers, i.e., you need K to be a $\mathbb{Q}$ -algebra.

Example 1.2.6.

(a) Let us write the p 's in terms of the e 's. The Newton–Girard identity

$$e_{n-1} p_1 - e_{n-2} p_2 + e_{n-3} p_3 - e_{n-4} p_4 \pm \cdots + (-1)^{n-1} \underbrace{e_0}_{=1} p_n = n e_n$$

can be solved for p_n to give

$$\begin{aligned}
 (-1)^{n-1} p_n &= n e_n - e_{n-1} p_1 + e_{n-2} p_2 - e_{n-3} p_3 \\
 &\quad + e_{n-4} p_4 \pm \cdots - (-1)^{n-2} e_1 p_{n-1}.
 \end{aligned}$$

Thus, if we know $e_1, e_2, \dots, e_n$ and $p_1, p_2, \dots, p_{n-1}$, then this formula gives us p_n . Thus we find, recursively,

$$p_1 = e_1;$$

then

$$-p_2 = 2e_2 - e_1p_1 = 2e_2 - e_1e_1 = 2e_2 - e_1^2,$$

so that

$$p_2 = e_1^2 - 2e_2;$$

then

$$\begin{aligned} p_3 &= 3e_3 - e_2p_1 + e_1p_2 \\ &= 3e_3 - e_2e_1 + e_1(e_1^2 - 2e_2) \\ &= e_1^3 - 3e_1e_2 + 3e_3; \end{aligned}$$

and likewise

$$p_4 = e_1^4 - 4e_1^2e_2 + 2e_2^2 + 4e_1e_3 - 4e_4.$$

Thus, by recursion, we find a formula for each p_n in terms of $e_1, e_2, \dots, e_n$. Note that $e_i = 0$ for all $i > N$; thus, we will not have to deal with an ever-growing number of e 's as n increases.

- (b) Similarly, we can express the p 's in terms of the h 's:

$$\begin{aligned} p_1 &= h_1; \\ p_2 &= -h_1^2 + 2h_2; \\ p_3 &= h_1^3 - 3h_1h_2 + 3h_3; \\ p_4 &= -h_1^4 + 4h_1^2h_2 - 2h_2^2 - 4h_1h_3 + 4h_4. \end{aligned}$$

These are the same formulas as for the p 's in terms of the e 's, but with different signs! We will soon see that this is no coincidence.

Note, however, that this algorithm expresses each p_n in terms of $h_1, h_2, \dots, h_n$, not in terms of $h_1, h_2, \dots, h_N$. And since h_i does not become 0 when $i > N$, the formulas will keep increasing in complexity with rising n .

- (c) Assume that K is a $\mathbb{Q}$ -algebra. Then, recursively applying the second Newton–Girard formula in the opposite direction, we find

$$\begin{aligned} e_1 &= p_1; \\ e_2 &= \frac{1}{2}p_1^2 - \frac{1}{2}p_2; \\ e_3 &= \frac{1}{6}p_1^3 - \frac{1}{2}p_1p_2 + \frac{1}{3}p_3; \\ e_4 &= \frac{1}{24}p_1^4 - \frac{1}{4}p_1^2p_2 + \frac{1}{8}p_2^2 + \frac{1}{3}p_1p_3 - \frac{1}{4}p_4, \end{aligned}$$

and so on.

All the coefficients in the above expressions have combinatorial meanings, which we will see later.

The following theorem – at least its part **(a)** – was originally proved by Gauss, and has since become fundamental in many places in algebra. We will prove it in the next lecture.

Theorem 1.2.7 (Fundamental Theorem of Symmetric Polynomials).

- (a) The elementary symmetric polynomials $e_1, e_2, \dots, e_N$ are algebraically independent (over K) and generate the K -algebra $\mathcal{S}$. In other words, each $f \in \mathcal{S}$ can be uniquely written as a polynomial in $e_1, e_2, \dots, e_N$. In yet other words, the map

$$\underbrace{K[y_1, y_2, \dots, y_N]}_{\substack{\text{a polynomial ring} \\ \text{in } N \text{ variables}}} \rightarrow \mathcal{S},$$

$$g \mapsto g(e_1, e_2, \dots, e_N)$$

is a K -algebra isomorphism.

- (b) The complete homogeneous symmetric polynomials $h_1, h_2, \dots, h_N$ are algebraically independent (over K) and generate the K -algebra $\mathcal{S}$. In other words, each $f \in \mathcal{S}$ can be uniquely written as a polynomial in $h_1, h_2, \dots, h_N$. In yet other words, the map

$$\underbrace{K[y_1, y_2, \dots, y_N]}_{\substack{\text{a polynomial ring} \\ \text{in } N \text{ variables}}} \rightarrow \mathcal{S},$$

$$g \mapsto g(h_1, h_2, \dots, h_N)$$

is a K -algebra isomorphism.

- (c) Assume that K is a $\mathbb{Q}$ -algebra. Then, the power sums $p_1, p_2, \dots, p_N$ are algebraically independent (over K) and generate the K -algebra $\mathcal{S}$. In other words, each $f \in \mathcal{S}$ can be uniquely written as a polynomial in $p_1, p_2, \dots, p_N$. In yet other words, the map

$$\underbrace{K[y_1, y_2, \dots, y_N]}_{\substack{\text{a polynomial ring} \\ \text{in } N \text{ variables}}} \rightarrow \mathcal{S},$$

$$g \mapsto g(p_1, p_2, \dots, p_N)$$

is a K -algebra isomorphism.

Example 1.2.8. Theorem 1.2.7 **(b)** tells us that each h_n with $n > N$ is a polynomial in the $h_1, h_2, \dots, h_N$. For example, for $N = 3$, we have

$$h_4 = h_1^4 - 3h_1^2h_2 + h_2^2 + 2h_1h_3.$$

Theorem 1.2.7 (a) tells us that every symmetric polynomial in three variables x, y, z can be written as a polynomial in e_1, e_2, e_3 . For example,

$$((x - y)(y - z)(z - x))^2 = e_1^2 e_2^2 - 4e_2^3 - 4e_1^3 e_3 + 18e_1 e_2 e_3 - 27e_3^2.$$

As we said, Theorem 1.2.7 will be proved later. For now, let us notice a simple criterion for proving that a polynomial is symmetric:

Lemma 1.2.9. For each $i \in [N - 1]$, we let s_i be the simple transposition in the symmetric group S_N that swaps i with $i + 1$ (and leaves all remaining elements of $[N]$ unchanged). Let $f \in \mathcal{P}$. Assume that

$$s_k \cdot f = f \quad \text{for each } k \in [N - 1]. \quad (1)$$

Then, f is symmetric.

Proof. A classical result in permutation combinatorics (see, e.g., [21s, Theorem 5.3.17]) says that any permutation in S_N can be written as a composition of simple transpositions (i.e., of some of $s_1, s_2, \dots, s_{N-1}$). Thus, for any $\sigma \in S_N$, we can write σ as $\sigma = s_{i_1} s_{i_2} \cdots s_{i_{p-2}} s_{i_{p-1}} s_{i_p}$, and then we have

$$\begin{aligned} \sigma \cdot f &= (s_{i_1} s_{i_2} \cdots s_{i_{p-2}} s_{i_{p-1}} s_{i_p}) \cdot f \\ &= s_{i_1} \cdot s_{i_2} \cdot \cdots \cdot s_{i_{p-2}} \cdot s_{i_{p-1}} \cdot \underbrace{s_{i_p} \cdot f}_{\substack{=f \\ \text{(by (1))}}} \\ &\quad \text{(since we have a group action of } S_N \text{ on } \mathcal{P}) \\ &= s_{i_1} \cdot s_{i_2} \cdot \cdots \cdot s_{i_{p-2}} \cdot \underbrace{s_{i_{p-1}} \cdot f}_{\substack{=f \\ \text{(by (1))}}} \\ &= s_{i_1} \cdot s_{i_2} \cdot \cdots \cdot s_{i_{p-2}} \cdot f \\ &= \cdots \\ &= f. \end{aligned}$$

Thus, f is symmetric. $\square$

For example, a polynomial $f \in K[x, y, z]$ is symmetric if and only if $f(y, x, z) = f$ and $f(x, z, y) = f$.

References

- [21s] Darij Grinberg, *An Introduction to Algebraic Combinatorics (Math 531, Winter 2024 lecture notes)*, 22 July 2026.
<http://www.cip.ifi.lmu.de/~grinberg/t/21s/lecs.pdf>