Transforming Matrices and Semialgebraic Sets into a Jordan basis

Julian D'Costa
University of Oxford
2 Dec 2024

Abstract

In the study of linear dynamical systems, one often needs to compute matrix powers. This is made much easier by diagonalizing the matrices, or writing them in the almost-diagonal Jordan normal form (JNF). One also often needs to work with associated semialgebraic sets, which are sets defined by polynomial inequalities, and transform them as well into the new basis. In this note, we explain how to transform a matrix into a Jordan basis, and how such a basis change affects the description size of the associated semialgebraic sets.

1 The Jordan normal form

Analysing the powers of a linear map is easiest if it can be written as a diagonal matrix in some basis. Unfortunately, this is not always possible. The Jordan normal form (JNF) is a generalization of diagonalisation that applies to any square matrix, even if it is non-diagonalisable, at the cost of needing to work in the splitting field of the characteristic polynomial. For a rational matrix, we usually work in the field of algebraic numbers.

Jordan Decomposition. For a given rational square matrix A one can compute *change of basis matrix* Q and *Jordan normal form* J so that $A = QJQ^{-1}$ and $J = \mathrm{diag}(J_1, J_2, \cdots, J_z)$ with J_i representing the i^{th} Jordan block taking the following form

$$J_i = \begin{bmatrix} \Lambda_i & 1 & 0 & \dots & 0 & 0 \\ 0 & \Lambda_i & 1 & \dots & 0 & 0 \\ 0 & 0 & \Lambda_i & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \Lambda_i & 1 \\ 0 & 0 & 0 & \dots & 0 & \Lambda_i \end{bmatrix}, \tag{1}$$

where Λ_i denotes the i^{th} eigenvalue of A. The size of J_i is equal to the multiplicity of the eigenvalue Λ_i and is denoted by $\kappa(\Lambda_i)$.

1.1 Real Jordan form.

For any $A \in \mathbb{R}^{n\times n}$ having complex eigenvalues, matrices Q and J in the Jordan normal form could have complex entries. In this case, the complex eigenvalues form complex conjugate pairs and give a *real Jordan form*: there are *real* matrices Q and J such that $A = QJQ^{-1}$ and $J = \mathrm{diag}(J_1, J_2, \cdots, J_z)$. The matrix J_i represents the i^{th} real Jordan block corresponding to either a real eigenvalue Λ_i or a complex pair $\Lambda_i = a_i \pm jb_i$. It is equal to (1) for real Λ_i and

has the following form for the complex pair $\Lambda_i = a_i \pm jb_i$,

$$J_i = \begin{bmatrix} \Lambda_i & I_{2\times 2} & 0_{2\times 2} & \dots & 0_{2\times 2} & 0_{2\times 2} \\ 0_{2\times 2} & \Lambda_i & I_{2\times 2} & \dots & 0_{2\times 2} & 0_{2\times 2} \\ 0_{2\times 2} & 0_{2\times 2} & \Lambda_i & \dots & 0_{2\times 2} & 0_{2\times 2} \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0_{2\times 2} & 0_{2\times 2} & 0_{2\times 2} & \dots & \Lambda_i & I_{2\times 2} \\ 0_{2\times 2} & 0_{2\times 2} & 0_{2\times 2} & \dots & 0_{2\times 2} & \Lambda_i \end{bmatrix}, \tag{2}$$

where with abuse of notation, we have indicated $\Lambda_i = \begin{bmatrix} a_i & -b_i \\ b_i & a_i \end{bmatrix}$. $I_{2\times 2}$ and $0_{2\times 2}$ denote identity and fully zero matrices of size 2 by 2.

1.2 Matrix powers.

If $A = QJQ^{-1}$, then we have $A^n = QJ^nQ^{-1}$ for $n \in \mathbb{N}$, where $J^n = \operatorname{diag}(J_1^n, J_2^n, \dots, J_z^n)$ and

$$J_i^n = \begin{bmatrix} \Lambda_i^n & n\Lambda_i^{n-1} & \binom{n}{2}\Lambda_i^{n-1} & \cdots & \binom{n}{k-1}\Lambda_i^{n-k+1} \\ 0 & \Lambda_i^n & n\Lambda_i^{n-1} & \cdots & \binom{n}{k-2}\Lambda_i^{n-k+2} \\ 0 & 0 & \Lambda_i^n & \cdots & \binom{n}{k-3}\Lambda_i^{n-k+3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & n\Lambda_i^{n-1} \\ 0 & 0 & 0 & \cdots & \Lambda_i^n \end{bmatrix}.$$

Here $k = \kappa(\Lambda_i)$ is the size of the Jordan block J_i.

For a square matrix $A \in \mathbb{R}^{m\times m}$, its corresponding exponential matrix is denoted by e^A and is defined as $e^A := \sum_{k=0}^{\infty} \frac{A^k}{k!}$.

2 Algorithmically computing Jordan normal form.

Computing the (complex) Jordan normal form of a given rational matrix has historically been seen as a difficult problem, because it is highly numerically unstable. The map from matrices to their Jordan forms is discontinuous - an infinitesimal perturbation that alters one occurrence of a duplicate eigenvalue to create a distinct eigenvalue will change the structure of the Jordan blocks. It is still possible to compute the JNF symbolically representing the eigenvalues in their canonical forms as algebraic numbers. This gives an approximation to the JNF of any desired precision in time polynomial in the digits of precision, the dimension, and the bitsize of the entries of the original matrix (assumed rational). Until recently, the best known method was the algorithm of Cai in 1996 [1] where the polynomial was unspecified but "of degree at least 12 in the dimension" [2].

However there is now a more explicit bound:

▶ **Theorem 1** (Computing JNF in poly time [2]). *Let A be an $n \times n$ matrix of τ-bit integers. Let b be a desired precision parameter. Then there exists an algorithm that outputs complex matrices J' and V' (with rational real and imaginary parts of entries of bitsize $\tau n^3 + b$) with the following properties.*

- *$\|J - J'\| \leqslant 2^{-b}\|J\|$, $\|V - V'\| \leqslant 2^{-b}\|V\|$ for some exact JNF $A = VJV^{-1}$.*
- *V' is relatively well conditioned: $\|V'\| \cdot \left\|V'^{-1}\right\| \leqslant 2^{O^*(\tau n^3)}$.*
- *The algorithm runs in expected $O^*(n^{\omega+3}\tau + n^4\tau^2 + n^\omega b)$ bit operations.*

Here the notation O^* denotes suppression of factors polylogarithmic in n, τ and b. The symbol ω is the complexity constant of matrix multiplication, which is known to be less than 3.

We can apply this to a rational matrix of bounded bitsize by factoring out a common denominator (of size at most 2^τ) and increasing precision by τ.

Theorem 1 does not specify how the matrix inverse V'^{-1} is computed. However, the matrix V' is well-conditioned, so we can compute V'^{-1} in polynomial time using standard techniques. Alternatively, we can use the algorithm of Cai which essentially uses a "row vector" version of the algorithm for V' to compute the inverse of V' in polynomial time.

2.1 Computing the real Jordan normal form.

We now discuss how to compute the *real* Jordan normal form QJQ^{-1} of A in polynomial time. First compute, in polynomial time, the (complex) Jordan normal form J' and matrices V, V^{-1} such that $A = VJ'V^{-1}$ using the algorithm from [2] or [1].

Computing J: Suppose, without loss of generality, that

$$J' = \operatorname{diag}(J'_1, J'_2, \ldots, J'_{2k-1}, J'_{2k}, J'_{2k+1}, \ldots, J'_{2k+z})$$

where for $1 \leqslant j \leqslant k$, the Jordan blocks J_{2j-1} and J_{2j} have the same dimension and have conjugate eigenvalues $\lambda_j = a_j + b_j i$ and $\overline{\lambda} = a_j - b_j i$, respectively. The blocks $J'_{2k+1}, \ldots, J'_{2k+z}$, on the other hand, have real eigenvalues. J is obtained by replacing, for each $1 \leqslant j \leqslant k$, $\operatorname{diag}(J'_{2j-1}, J'_{2j})$ with a real Jordan block of the same dimension with $\Lambda = \begin{bmatrix} a & -b \\ b & a \end{bmatrix}$ and keeping the blocks $J'_{2k+1}, \ldots, J'_{2k+z}$ unchanged.

Computing Q: Let $\kappa(j)$ denote the multiplicity of the Jordan block J'_i for $1 \leqslant i \leqslant 2k+z$, and $v_1^1, \ldots, v_{\kappa(1)}^1, \ldots, v_1^{2k}, \ldots, v_{\kappa(2k)}^{2k}, \ldots, v_1^{2k+z}, \ldots, v_{\kappa(2k+z)}^{2k+z} \in \overline{\mathbb{Q}}^m$ be the columns of V. It will be the case that for all $1 \leqslant j \leqslant k$ and l, $v_l^{2j-1} = \overline{v_l^{2j}}$ in the sense that $v_l^{2j-1} = x_l^j + y_l^j i$ and $v_l^{2j} = x_l^j - y_l^j i$ for vectors $x_l^j, y_l^j \in \mathbb{R}^m$. Moreover, for $j > 2k$, $v_l^{2j} \in \mathbb{R}^m$. Finally, columns of Q are obtained from columns of V as follows. For $1 \leqslant j \leqslant k$ and all l, replace v_l^{2j-1} with x_l^j and v_l^{2j} with y_l^j and keep v_l^{2k+z} for all l and $m > 0$ unchanged, in the same way the proof of existence of real Jordan normal form proceeds.

Computing Q^{-1}: Summarizing the construction above, Q is obtained from V by replacing columns $x + yi$ and $x - yi$, $x, y \in \mathbb{R}^m$ by x and y, respectively. Since $x = \frac{1}{2}(x+yi) + \frac{1}{2}(x-yi)$ and $y = -\frac{1}{2}i(x+yi) + \frac{1}{2}i(x-yi)$, this construction is linear and we can write $Q = VT$ for some $T \in \mathbb{C}^{m \times m}$ with entries in $\{\frac{1}{2}, -\frac{1}{2}, \frac{1}{2}i, -\frac{1}{2}i, 1, 0\}$. Moreover, the linear transformation T is clearly invertible: $x + yi = 1 \cdot x + iy$ and $x - yi = 1 \cdot x - (-i)y$, and hence $T^{-1} \in \mathbb{C}^{m \times m}$ with entries in $\{1, i, -i, 0\}$. Finally, compute Q^{-1} via $Q = VT \implies Q^{-1} = T^{-1}V^{-1}$, observing that we already know how to compute V^{-1} in polynomial time.

3 Description size bounds on transforming a semialgebraic set to a new basis

When working with iterated matrix maps it is often useful to transform the system under consideration so that the matrix is in Jordan form. We now combine results from the previous sections to control the description size increase that may occur when transforming a semialgebraic set to a new basis.

3.1 Semialgebraic Sets

A *semialgebraic* set is a subset of $\mathbb{R}^n$ that is the solution set of a boolean combination of polynomial equalities and inequalities

We say that a quantifier-free semialgebraic set S has *description size*[1] at most (n, d, τ) if it can be expressed by a boolean combination of polynomial equations and inequalities $P(x_1, \ldots, x_n) \bowtie 0$ with $\bowtie \in \{\leqslant, =\}$, involving polynomials $P \in \mathbb{Z}[x_1, \ldots, x_n]$ in at most n variables of total degree at most d with integer coefficients bounded in bitsize by τ.

3.2 Example

The following example illustrates the kind of blowup that may occur: Consider the matrix

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 + 10^{-2} \end{bmatrix},$$

its associated Jordan decomposition

$$A = QJQ^{-1} = \begin{bmatrix} 1 & 0 & 10000 \\ 0 & 1 & 100 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1.01 \end{bmatrix} \begin{bmatrix} 1 & 0 & -10000 \\ 0 & 1 & -100 \\ 0 & 0 & 1 \end{bmatrix}$$

and the (singleton) set $S = \{(0, 0, 1) \in \mathbb{R}^3\}$. In the Jordan basis, this set becomes $\{(-10^4, -10^2, 1) \in \mathbb{R}^3\}$, increasing the bitsize of the coefficients defining the set by a factor polynomial in the dimension and the bitsize of the matrix entries.

3.3 Transforming Semialgebraic Sets

Let A and A' be matrices (not necessarily of the same dimensions) and K and K' be semialgebraic sets. We say (A, K) and (A', K') are *equivalent systems* if there exists a bijective map B such that $A^k x \in K \iff A'^k B(x) \in K'$ for all $k \in \mathbb{N}$.

Normally if A' is the Jordan normal form of A via a change of basis Q (i.e. $A = QA'Q^{-1}$) we would expect Q^{-1} to serve as the map B. However a naive approach to defining the set $Q^{-1}K$ via quantifier elimination may result in significantly greater description size than K itself, so in fact we make a more general definition. This lets us introduce additional dimensions to safely manipulate the algebraic numbers that arise in the Jordan normal form.

▶ **Lemma 2.** *Let $A \in \mathbb{Q}^{n \times n}$ be a matrix with entries of bitsize a and K be a semialgebraic set of description size (n, d, τ).*

Then there exists $m \leqslant n$ and a system (J', K') equivalent to (A, K) such that the matrix $J' \in (\overline{\mathbb{Q}} \cap \mathbb{R})^{(n+m)\times(n+m)}$ is in real Jordan normal form and K' has description size $(n + m, n^2 d, \tau + d \log 2n^3 + d\sigma)$, where $\sigma = O^*(an^3)$.

Proof. By Section 2 we can compute in polynomial time real algebraic numbers $\gamma_1, \ldots, \gamma_m$ and a matrix $Q \in \mathbb{Q}(\gamma_1, \ldots, \gamma_m)^{n \times n}$ such that $A = QJQ^{-1}$, where J is in real Jordan normal form. Note that $m \leqslant n$ since all complex eigenvalues occur in conjugate pairs. Let δ be a

[1] The number of inequalities and equations is often relevant to the complexity of algorithmic operations on semialgebraic sets, but we will compute only size bounds on the set, so it is not relevant here.

bound on the degrees of $\gamma_1, \ldots, \gamma_m$. Since the real and imaginary parts may be obtained by adding or subtracting conjugates, we have $\delta \leqslant n^2$ by elementary Galois theory.

More precisely, we can compute in polynomial time:

1. Univariate polynomials $f_1, \ldots, f_m$ with integer coefficients such that $f_j(\gamma_j) = 0$ for all $j = 1, \ldots, m$.
2. Rational numbers $a_1, b_1, \ldots, a_m, b_m$, such that γ_j is the unique root of f_j in the interval $[a_j, b_j]$..
3. For $i = 1, \ldots, n$ and $j = 1, \ldots, n$ polynomials of degree at most δ $Q_{i,j} \in \mathbb{Q}[x]$ and indexes $\ell_{i,j}$ such that the matrix Q at row i and column j is given by the algebraic number $Q_{i,j}(\gamma_{\ell_{i,j}})$.

Let $\sigma \in \mathbb{N}$ be a common bound on the following quantities:

1. The bitsize of the coefficients of $f_1, \ldots, f_m$.
2. The bitsize of the endpoints of the isolating intervals $[a_j, b_j]$.
3. The bitsize of the coefficients of the polynomials $Q_{j,k}$.

Then σ is computable in polynomial time from A, so it depends polynomially on n and on a (the bitsize of the entries of A). From the proof of Theorem 1 we can derive that in fact $\sigma = O^*(an^3)$ (suppressing logarithmic factors in n).

We fix $K' = (Q^{-1}K) \times \{\gamma_1, \ldots, \gamma_m\}$ and note that $A^k x \in K$ if and only if

$$(J \times I_m)^k (Q^{-1}x, (\gamma_1, \ldots, \gamma_m)) \in (Q^{-1}K) \times \{(\gamma_1, \ldots, \gamma_m)\}.$$

Here the map $x \mapsto (Q^{-1}x, (\gamma_1, \ldots, \gamma_m))$ is the map B required for equivalence. The last m coordinates are added to allow us to manipulate these algebraic numbers within the description of K' symbolically through a formula. (If we did not do this, and tried to apply quantifier elimination to K' to work directly with the set $Q^{-1}K$, we would end up with a set that has description size $(n, (dn)^{O(n)}, (\tau + d\sigma)(dn)^{O(n^2)})$ which is exponential in n.)

We now show there exists a description of the set K' with the claimed size.

Let $\Phi(x_1, \ldots, x_n)$ be the formula that describes K. We introduce fresh variables $z_1, \ldots, z_m$ and consider the formula

$$\Psi(z_1, \ldots, z_m) \wedge \widehat{\Phi}(x_1, \ldots, x_n, z_1, \ldots, z_m)$$

where $\Psi(z_1, \ldots, z_m)$ is the conjunction of the terms

$$f_j(z_j) = 0 \wedge z_j \geqslant a_j \wedge z_j \leqslant b_j$$

which ensures $z_j = \gamma_j$ for $j = 1, \ldots, m$, and $\widehat{\Phi}$ is obtained from Φ by replacing each atom $P(x_1, \ldots, x_n) \bowtie 0$ in Φ by the atom

$$P\left(\sum_{k=1}^{n} Q_{1,k}(z_{\ell_{1,k}})x_k, \ldots, \sum_{k=1}^{n} Q_{n,k}(z_{\ell_{n,k}})x_k,\right) \bowtie 0.$$

It is not hard to see that this new formula describes the set K'. Evidently, the number of variables in this description is $n + m$. The formula Ψ involves $3m$ polynomials of degree at most δ whose coefficients are bounded in bitsize by σ.

It remains to determine the description size of the formula $\widehat{\Phi}$. We claim that the degrees of the polynomials in $\widehat{\Phi}$ are bounded by $\delta \cdot d$ and that the bitsize of their coefficients is bounded by $\tau + d(\log(n) + \log(\delta + 1) + \sigma)$. This is established by a straightforward but cumbersome calculation, which we relegate to the appendix.

◀

References

1 J.-Y. Cai. Computing Jordan normal forms exactly for commuting matrices in polynomial time. *Int. J. Found. Comput. Sci.*, 5(3/4):293–302, 1994.

2 Papri Dey, Ravi Kannan, Nick Ryder, and Nikhil Srivastava. Bit Complexity of Jordan Normal Form and Polynomial Spectral Factorization. In *ITCS*, 2023. URL: `https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ITCS.2023.42`, `doi:10.4230/LIPIcs.ITCS.2023.42`.

A Bounding the degree and coeffcients of $\widehat{\Phi}$

We now show that the degrees of the polynomials in $\widehat{\Phi}$ are bounded by $\delta \cdot d$ and that the bitsize of their coefficients is bounded by $\tau + d(\log(n) + \log(\delta + 1) + \sigma)$.

We recall the multinomial theorem:

▶ **Lemma 3** (Multinomial theorem). *Let* R *be a ring. Let* N *be a positive integer. Let* $z_1, \ldots, z_N \in R$. *Then*

$$\left(\sum_{k=1}^{N} z_k\right)^e = \sum_{j_1+\cdots+j_N=e} \binom{e}{j_1, \ldots, j_N} \prod_{t=1}^{N} z_t^{j_t},$$

where

$$\binom{e}{j_1, \ldots, j_N} = \frac{e!}{j_1! \cdot \cdots \cdot j_N!}.$$

It will be convenient to make use of the following straightforward application of the distributivity law of multiplication over addition. Here $\coprod$ denotes the disjoint union.

▶ **Lemma 4.** *Let* I *be a finite set. Let* J *be a set-valued function that sends each* $i \in I$ *to a finite set* $J(i)$. *Let* R *be a ring. For all* $(i, j) \in I \times \coprod_{i \in I} J(i)$, *let* $a_{i,j}$ *be an element of* R. *Then we have*

$$\prod_{i \in I} \sum_{j \in J(i)} a_{i,j} = \sum_{f} \prod_{i \in I} a_{i,f(i)}$$

where f *ranges over all functions*

$$f \colon I \to \coprod_{i \in I} J(i)$$

satisfying $f(i) \in J(i)$ *for all* $i \in I$.

Now, let $P(x_1, \ldots, x_n) \bowtie 0$ be an atom in Φ. This atom is a sum of monomials

$$C \cdot x_1^{e_1} \cdot \cdots \cdot x_n^{e_n}$$

with $\log |C| \leqslant \tau$ and $e_1 + \cdots + e_n \leqslant d$. It suffices to bound the degrees and the bitsize of the coefficients of the polynomials that are obtained by applying our substitution of variables to monomials of this form.

Under our substitution such a monomial becomes:

$$C \cdot \left(\sum_{j=1}^{n} Q_{1,j}(z_{\ell_{1,j}}) x_j\right)^{e_1} \cdot \cdots \cdot \left(\sum_{j=1}^{n} Q_{n,j}(z_{\ell_{n,j}}) x_j\right)^{e_n} = C \prod_{k=1}^{n} \left(\sum_{j=1}^{n} Q_{k,j}(z_{\ell_{k,j}}) x_j\right)^{e_k}.$$

Apply the multinomial theorem to the expressions $\left(\sum_{j=1}^{n} Q_{k,j}(z_{\ell_{k,j}})x_j\right)^{e_k}$ to obtain:

$$C \cdot \prod_{k=1}^{n} \left(\sum_{j_{k,1}+\cdots+j_{k,n}=e_k} \binom{e_k}{j_{k,1},\ldots,j_{k,n}} \prod_{t=1}^{n} (Q_{k,t}(z_{\ell_{k,t}})x_t)^{j_{k,t}} \right).$$

Write

$$Q_{k,t}(z_{\ell_{k,t}}) = \sum_{p=0}^{\delta} \alpha_{k,t,p} z_{\ell_{k,t}}^{p}.$$

Applying the multinomial theorem to the terms

$$(Q_{k,t}(z_{\ell_{k,t}})x_t)^{j_{k,t}} = \left(\sum_{p=0}^{\delta} \alpha_{k,t,p} z_{\ell_{k,t}}^{p} x_t \right)^{j_{k,t}}$$

we obtain

$$(Q_{k,t}(z_{\ell_{k,t}})x_t)^{j_{k,t}} = \sum_{r_0+\cdots+r_\delta=j_{k,t}} \binom{j_{k,t}}{r_0,\ldots,r_\delta} \prod_{s=0}^{\delta} \alpha_{k,t,s}^{r_s} z_{\ell_{k,t,s}}^{sr_s} x_t^{r_s}.$$

The full expression is hence:

$$C \cdot \prod_{k=1}^{n} \left(\sum_{j_{k,1}+\cdots+j_{k,n}=e_k} \binom{e_k}{j_{k,1},\ldots,j_{k,n}} \prod_{t=1}^{n} \sum_{r_0+\cdots+r_\delta=j_{k,t}} \binom{j_{k,t}}{r_0,\ldots,r_\delta} \prod_{s=0}^{\delta} \alpha_{k,t,s}^{r_s} z_{\ell_{k,t,s}}^{sr_s} x_t^{r_s} \right).$$

Write this as:

$$C \cdot \prod_{k=1}^{n} \sum_{j_{k,1}+\cdots+j_{k,n}=e_k} \binom{e_k}{j_{k,1},\ldots,j_{k,n}} \prod_{t=1}^{n} \sum_{r_0+\cdots+r_\delta=j_{k,t}} c_{k,j_{k,1},\ldots,j_{k,n},t,r_0,\ldots,r_\delta}.$$

Apply Lemma 4 to move out the innermost sum, thus obtaining an equal expression:

$$C \cdot \prod_{k=1}^{n} \left(\sum_{j_{k,1}+\cdots+j_{k,n}=e_k} \sum_{f} \binom{e_k}{j_{k,1},\ldots,j_{k,n}} \prod_{t=1}^{n} c_{k,j_{k,1},\ldots,j_{k,n},t,f(t)} \right).$$

where the sum $\sum_f$ ranges over all functions $f\colon \{1,\ldots,n\} \to \mathbb{N}^{\delta+1}$ with $f(t) = (r_0,\ldots,r_\delta)$ satisfying $r_0 + \cdots + r_\delta = j_{k,t}$.

Write the result as:

$$C \cdot \prod_{k=1}^{n} \sum_{j_{k,1}+\cdots+j_{k,n}=e_k} \sum_{f} d_{k,j_{k,1},\ldots,j_{k,n},f}.$$

Apply Lemma 4 again to obtain that this is equal to:

$$\sum_{g} C \cdot \prod_{k=1}^{n} d_{k,g(k)},$$

where g ranges over all functions $g\colon \{1,\ldots,n\} \to \mathbb{N}^n \times (\mathbb{N}^{\delta+1})^{\{1,\ldots,n\}}$ with $g(k) = (j_{k,1},\ldots,j_{k,n},f)$ satisfying $j_{k,1} + \cdots + j_{k,n} = e_k$ and f as above.

Thus, the final result is a sum of monomials of the form

$$C \cdot \prod_{k=1}^{n} d_{k,g(k)} = C \cdot \prod_{k=1}^{n} \binom{e_k}{j_{k,1},\ldots,j_{k,n}} \prod_{t=1}^{n} c_{k,j_{k,1},\ldots,j_{k,n},t,f(t)}$$
$$= C \cdot \prod_{k=1}^{n} \binom{e_k}{j_{k,1},\ldots,j_{k,n}} \prod_{t=1}^{n} \binom{j_{k,t}}{r_0(t),\ldots,r_\delta(t)} \prod_{s=0}^{\delta} \alpha_{k,t,s}^{r_s(t)} z_{\ell_{k,t,s}}^{s r_s(t)} x_t^{r_s(t)},$$

Where $j_{k,1}+\cdots+j_{k,n} = e_k$ and $r_0(t),\ldots,r_\delta(t)$ are functions of t satisfying $r_0(t)+\cdots+r_\delta(t) = j_{k,t}$.

Since $e_1 + \cdots + e_n \leqslant d$, the degrees of these monomials are bounded by $\delta \cdot d$.

Let us compute a bound on the bitsize of the coefficients. We have:

$$\log\left(|C| \cdot \prod_{k=1}^{n} \binom{e_k}{j_{k,1},\ldots,j_{k,n}} \prod_{t=1}^{n} \binom{j_{k,t}}{r_0(t),\ldots,r_\delta(t)} \prod_{s=0}^{\delta} |\alpha_{k,t,s}|^{r_s(t)}\right)$$
$$\leqslant \tau + \sum_{k=1}^{n} \log\binom{e_k}{j_{k,1},\ldots,j_{k,n}} + \sum_{k=1}^{n}\sum_{t=1}^{n} \log\binom{j_{k,t}}{r_0(t),\ldots,r_\delta(t)} + \sum_{k=1}^{n}\sum_{t=1}^{n}\sum_{s=0}^{\delta} r_s(t)\sigma.$$

Use the estimate $\binom{f}{k_1,\ldots,k_m} \leqslant m^f$ to obtain:

$$\log\left(|C| \cdot \prod_{k=1}^{n} \binom{e_k}{j_{k,1},\ldots,j_{k,n}} \prod_{t=1}^{n} \binom{j_{k,t}}{r_0(t),\ldots,r_\delta(t)} \prod_{s=0}^{\delta} |\alpha_{k,t,s}|^{r_s(t)}\right)$$
$$\leqslant \tau + \sum_{k=1}^{n} e_k \log(n) + \sum_{k=1}^{n}\sum_{t=1}^{n} j_{k,t} \log(\delta+1) + \sum_{k=1}^{n}\sum_{t=1}^{n}\sum_{s=0}^{\delta} r_s(t)\sigma.$$
$$\leqslant \tau + d\log(n) + d\log(\delta+1) + d\sigma.$$
$$= \tau + d(\log(n) + \log(\delta+1) + \sigma).$$

Thus, everything is shown.