Factorization is representation-relative: an epistemic–algorithmic study

Orges Leka

November 11, 2025

Abstract

We formalize the idea that the practical difficulty of integer factorization depends on the *representation* of the input. Two agents know the *same* integer n, but in different forms: Agent A receives the binary string $\mathsf{bin}(n)$; Agent B receives a polynomial $f_n(x) \in \mathbb{Z}[x]$ whose degree is $\Theta(\log n)$ and that multiplicatively encodes the prime structure of n. Under standard, textbook bounds for univariate polynomial factorization over $\mathbb{Q}$ (polynomial in degree and coefficient bit-size), B can compute the prime factorization $\mathsf{Fact}(n)$ in time polynomial in $\log n$, while doing so from $\mathsf{bin}(n)$ is not known to be polynomial-time. We capture the advantage using resource-bounded epistemic operators, instance complexity, and conditional Kolmogorov complexity, and prove "representation-relative" theorems. The note is self-contained and conditional only on explicit assumptions stated herein.

Contents

1 Motivation and Overview

It is common wisdom that multiplication is easy whereas factorization seems hard. Less emphasized is that *difficulty is representation-relative.* We consider two representations of the same integer n:

- the standard *binary numeral* $\mathsf{bin}(n)$;
- a *polynomial code* $f_n(x) \in \mathbb{Z}[x]$ of small degree, designed so that factorizing f_n over $\mathbb{Q}$ reveals the prime factors of n by simple evaluation at $x = 2$.

We study how this difference of representation changes what agents can compute quickly (their *explicit knowledge*), quantify the advantage with epistemic and information-theoretic measures, and give precise conditional theorems.

High-level message. If the map $n \mapsto f_n$ satisfies natural axioms (irreducibility for primes, multiplicativity, small degree and height), then

$$\text{factorization from } f_n \text{ is in poly}(\log n)\text{-time,}$$

whereas factorization from $\mathsf{bin}(n)$ is *not known* to be in polynomial time. Thus, in a rigorous resource-bounded epistemic sense, Agent B *knows more* about n's factorization than Agent A, despite both "knowing n".

2 Representations as Epistemic Objects

Definition 2.1 (Representation scheme)**.** A *representation scheme* for natural numbers is a pair $\mathcal{R} = (\mathsf{Enc}, \mathsf{Dec})$ with

- $\mathsf{Enc} : \mathbb{N} \to \Sigma^*$ an injective encoding,
- $\mathsf{Dec} : \Sigma^* \to \mathbb{N}$ a partial decoding with $\mathsf{Dec}(\mathsf{Enc}(n)) = n$.

We identify the *representation* of n with $r = \mathsf{Enc}(n)$, and write $\mathsf{eval}_{\mathcal{R}}(r) := \mathsf{Dec}(r)$.

Example 2.2 (Two schemes)**.** (a) **Binary numerals:** $\mathcal{R}_{\mathsf{bin}}$ with $\mathsf{Enc} = \mathsf{bin}$ and Dec the usual base-2 interpretation.

(b) **Polynomial codes:** $\mathcal{R}_{\text{poly}}$ with $\mathsf{Enc}(n) = f_n(x) \in \mathbb{Z}[x]$ a univariate polynomial satisfying assumptions in §**??**, and $\mathsf{Dec}(f_n) = n$ given by evaluation at $x = 2$.

Definition 2.3 (Task and solution)**.** The *factorization task* maps n to

$$\mathsf{Fact}(n) = \{(p_i, v_{p_i}(n)) : p_i \in \mathcal{P},\ v_{p_i}(n) \geq 1\}.$$

A correct algorithm outputs $\mathsf{Fact}(n)$ up to ordering.

3 A Resource-Bounded Epistemic Lens

We adopt a minimal, operational notion of *explicit knowledge* indexed by time.

Definition 3.1 (Time-indexed explicit knowledge)**.** Let A be an algorithm and $\mathcal{R}$ a representation scheme. We write

$$\mathsf{K}^t_{\mathcal{R}}(\varphi(n)) \quad \text{iff} \quad \text{there exists an algorithm } A \text{ that, on input } \mathsf{Enc}_{\mathcal{R}}(n),\ \text{outputs a witness of } \varphi(n) \text{ in time } \leq t.$$

For the factorization task, $\varphi(n)$ is "outputs $\mathsf{Fact}(n)$".

Definition 3.2 (Time advantage and instance complexity)**.** For a fixed n, define the *time advantage* of $\mathcal{R}_2$ over $\mathcal{R}_1$ by

$$\Delta_T(n; \mathcal{R}_1 \to \mathcal{R}_2) := \inf\{t :\ \mathsf{K}^t_{\mathcal{R}_2}(\mathsf{Fact}(n))\} \ - \ \inf\{t :\ \mathsf{K}^t_{\mathcal{R}_1}(\mathsf{Fact}(n))\}.$$

We also write the *representation-relative instance complexity*

$$\mathrm{IC}_{\mathcal{R}}(n) := \inf\{t : \mathsf{K}^t_{\mathcal{R}}(\mathsf{Fact}(n))\}.$$

Definition 3.3 (Conditional description length advantage)**.** Let $K(\cdot \mid \cdot)$ be (prefix-free) Kolmogorov complexity. Define

$$\Delta_K(n; \mathcal{R}_1 \to \mathcal{R}_2) := K\big(\mathsf{Fact}(n) \mid \mathsf{Enc}_{\mathcal{R}_1}(n)\big) - K\big(\mathsf{Fact}(n) \mid \mathsf{Enc}_{\mathcal{R}_2}(n)\big).$$

4 The Polynomial Representation of Integers

We axiomatize the desired properties of the map $n \mapsto f_n(x) \in \mathbb{Z}[x]$.

Assumption 4.1 (Prime code)**.** For every prime p, $f_p(x) \in \mathbb{Z}[x]$ is *irreducible* over $\mathbb{Z}[x]$ and satisfies $f_p(2) = p$.

Assumption 4.2 (Multiplicativity)**.** For $n = \prod_i p_i^{e_i}$, we have

$$f_n(x) = \prod_i f_{p_i}(x)^{e_i}.$$

Assumption 4.3 (Small degree). There exist constants $c_1, c_2 > 0$ such that for all $n \geq 2$,

$$c_1 \log n \;\leq\; \deg f_n \;\leq\; c_2 \log n.$$

Assumption 4.4 (Moderate coefficient bit-size). Let $H = \mathrm{ht}(g)$ be the maximum absolute value of the coefficients of $g \in \mathbb{Z}[x]$. The bit-length of the height, $\log(1+H)$, is bounded by a polynomial in $\log n$. That is, there exists a polynomial q with

$$\log(1 + \mathrm{ht}(f_n)) \;\leq\; q(\log n).$$

Remark 4.5 (Decoding). By ?? and ??, $f_n(2) = n$, so decoding n from f_n is trivial: $\mathsf{Dec}(f_n) = f_n(2)$.

4.1 Correctness of factorization by factorization of f_n

Lemma 4.6 (Irreducible codes identify primes). *Under ??, if $g \in \mathbb{Z}[x]$ is an irreducible factor of some f_n, then there is a unique prime p with g associate to f_p (i.e. $g = \pm f_p$).*

Proof. By ??, every irreducible factor of f_n divides some f_p for a prime p dividing n. Since f_p is irreducible in the UFD $\mathbb{Z}[x]$, any irreducible divisor is an associate, hence $\pm f_p$. Uniqueness follows from unique factorization. □

Theorem 4.7 (Correctness of the polynomial route). *Assume ????. Let the factorization of f_n in $\mathbb{Z}[x]$ be $f_n = \prod_{j=1}^{t} g_j^{e_j}$ with irreducible g_j. Then there exist pairwise distinct primes $p_1, \dots, p_t$ with $g_j = \pm f_{p_j}$ and*

$$\mathsf{Fact}(n) = \{(p_j,\, e_j) \;:\; j = 1, \dots, t\}.$$

Moreover, each p_j can be read off *as $p_j = g_j(2)$.*

Proof. By ??, each g_j is associate to some f_{p_j}. Multiplicativity gives $f_n = \prod f_{p_j}^{e_j}$ and by uniqueness of factorization in $\mathbb{Z}[x]$ the exponents e_j must match the prime-power exponents of n. Finally $g_j(2) = \pm f_{p_j}(2) = \pm p_j$. Since $f_p(x)$ has non-negative coefficients (as shown in §6), $f_p(2) = p > 0$. We can choose the associate g_j such that $g_j(2) > 0$, fixing the sign to read $p_j = g_j(2)$. □

4.2 Complexity of the polynomial route

Theorem 4.8 (Polynomial-time factorization from f_n). *Assume ????????. Then*

$$\mathrm{IC}_{\mathcal{R}_{\mathrm{poly}}}(n) = \mathrm{poly}(\log n).$$

In particular, there exists an algorithm that on input f_n outputs $\mathsf{Fact}(n)$ in time polynomial in $\log n$.

Proof. By standard algorithms (e.g., LLL-based methods) the factorization of a univariate integer polynomial into irreducibles over $\mathbb{Q}$ can be computed in time polynomial in its bit-size. The bit-size of f_n is determined by its degree $d = \deg f_n$ and the bit-length of its largest coefficient (its height $H = \mathrm{ht}(f_n)$). The total bit-size is $O(d \cdot \log(1+H))$. By ??, $d = \Theta(\log n)$. By ??, $\log(1+H) \leq q(\log n)$ for some polynomial q. Therefore, the total bit-size is $O(\log n \cdot q(\log n))$, which is $\mathrm{poly}(\log n)$. Since the polynomial factoring algorithm runs in time polynomial in this bit-size, the factoring step is $\mathrm{poly}(\log n)$. By ??, reading off the primes as $p_j = g_j(2)$ (a polynomial-time evaluation) and pairing them with their exponents e_j yields $\mathsf{Fact}(n)$ with only polynomial overhead. Therefore the entire pipeline is $\mathrm{poly}(\log n)$. □

Remark 4.9 (Epistemic reading). ?? states that $\mathsf{K}_{\mathcal{R}_{\mathrm{poly}}}^{\mathrm{poly}(\log n)}(\mathsf{Fact}(n))$ holds under our assumptions.

5 Comparison with the Binary Representation

Let $\mathcal{R}_{\mathsf{bin}}$ be the usual binary numerals. For completeness:

Proposition 5.1 (Baseline decoding)**.** *Given* $\mathsf{bin}(n)$*, we can compute* n *in time* $O(\log n)$ *and* $|\mathsf{bin}(n)| = \Theta(\log n)$*. This does not, by itself, reveal* $\mathsf{Fact}(n)$*.*

Assumption 5.2 (Hardness hypothesis for the baseline)**.** There is no algorithm that, on input $\mathsf{bin}(n)$, computes $\mathsf{Fact}(n)$ in time polynomial in $\log n$.

Remark 5.3. **??** is consistent with the current state of knowledge: factoring is not known to be in $\mathbf{P}$, and the best general-purpose algorithms are subexponential but superpolynomial in $\log n$.

Theorem 5.4 (Representation-relative knowledge separation)**.** *Under* **??????????***, there exists a polynomial* p *such that for all sufficiently large* n*,*

$$\mathsf{K}^{p(\log n)}_{\mathcal{R}_{\text{poly}}}(\mathsf{Fact}(n)) \quad \textit{holds while} \quad \mathsf{K}^{p(\log n)}_{\mathcal{R}_{\mathsf{bin}}}(\mathsf{Fact}(n)) \quad \textit{fails.}$$

Equivalently, $\Delta_T(n; \mathcal{R}_{\mathsf{bin}} \to \mathcal{R}_{\text{poly}}) < 0$ *for infinitely many* n *(in fact, for all large* n*).*

Proof. The first claim is exactly **??**. The second is the negation supplied by **??**. The time-advantage statement is by definition of Δ_T. □

Proposition 5.5 (Description-length advantage)**.** *Under the same assumptions, for infinitely many* n *we have*

$$\Delta_K(n; \mathcal{R}_{\mathsf{bin}} \to \mathcal{R}_{\text{poly}}) > 0.$$

Proof sketch. Given f_n, a short universal program that (i) factors f_n over $\mathbb{Q}$ and (ii) evaluates each irreducible factor at $x = 2$ suffices to output $\mathsf{Fact}(n)$, so $K(\mathsf{Fact}(n) \mid f_n)$ is bounded by a constant plus the description of a polynomial-time factoring routine. Relative to $\mathsf{bin}(n)$, any program that outputs $\mathsf{Fact}(n)$ must (in effect) perform nontrivial search absent additional advice; hence for infinitely many n the conditional complexity cannot be uniformly bounded by the same constant. Formalizing yields the claim. □

6 A Concrete Polynomial Representation

We now exhibit a specific family of polynomials $(f_n)_{n \geq 1}$ and prove that it satisfies the four axioms required by our framework.

6.1 Definition

Let the family of polynomials $f_n(x) \in \mathbb{Z}[x]$ be defined recursively as follows:

$$\begin{aligned}
f_1(x) &= 1, \\
f_2(x) &= x, \\
\text{if } n \text{ is prime:} \quad f_n(x) &= 1 + f_{n-1}(x), \\
\text{if } n \text{ has the prime factorization } n = \prod_p p^{\nu_p(n)}: \quad f_n(x) &= \prod_p (f_p(x))^{\nu_p(n)}.
\end{aligned}$$

(All products are over primes p.)

6.2 Verification of Axioms

We prove that this family (f_n) satisfies all four assumptions from §**??**.

Theorem 6.1. *The polynomial family* $(f_n(x))_{n\geq 1}$ *defined above satisfies Assumptions* **??**, **??**, **??**, *and* **??**.

Proof.

- **Assumption ?? (Prime code):** We need to show that for any prime p, $f_p(x)$ is irreducible in $\mathbb{Z}[x]$ and $f_p(2) = p$.
  - **Evaluation at 2:** We prove $f_n(2) = n$ for all $n \geq 1$ by strong induction. It holds for $n = 1$ ($f_1(2) = 1$) and $n = 2$ ($f_2(2) = 2$). If $n = p$ is prime, $f_p(2) = 1 + f_{p-1}(2)$. By the inductive hypothesis, $f_{p-1}(2) = p - 1$. Thus $f_p(2) = 1 + (p - 1) = p$. If $n = \prod p_i^{e_i}$ is composite, $f_n(2) = \prod (f_{p_i}(2))^{e_i}$. By the inductive hypothesis, $f_{p_i}(2) = p_i$. Thus $f_n(2) = \prod p_i^{e_i} = n$. The claim holds. In particular, $f_p(2) = p$ for all primes p.
  - **Irreducibility:** It has been proven (e.g., in the paper "Counting primes with polynomials") that for every prime p, this $f_p(x)$ is irreducible in $\mathbb{Z}[x]$. The proof relies on showing that all zeros of f_p lie in the half-plane $\mathrm{Re}(z) < 3/2$ and then applying an irreducibility criterion (like Sury's lemma) using the evaluation $f_p(2) = p$.

- **Assumption ?? (Multiplicativity):** This is satisfied by definition. The rule for composite $n = \prod p_i^{e_i}$ is precisely $f_n(x) = \prod f_{p_i}(x)^{e_i}$.

- **Assumption ?? (Small degree):** Let $d_n = \deg f_n$. From the definition: $d_1 = 0$, $d_2 = 1$. If p is prime, $d_p = \deg(1 + f_{p-1}) = d_{p-1}$. If $n = \prod p_i^{e_i}$, then $d_n = \sum e_i d_{p_i}$. This implies $d_p = d_{p-1} = \sum_{r|(p-1)} \nu_r(p-1)\, d_r$. It is a known result for this family (and proven by induction) that there are absolute constants c_1, c_2 (namely $c_1 = 1/\log 3, c_2 = 1/\log 2$) such that for all $n \geq 2$,
$$\frac{\log n}{\log 3} \leq d_n \leq \frac{\log n}{\log 2}.$$
This establishes $d_n = \Theta(\log n)$, satisfying the assumption.

- **Assumption ?? (Moderate coefficient bit-size):** We need to show $\log(1 + H(f_n)) \leq q(\log n)$ for some polynomial q. First, we establish by induction that all f_n are monic and have non-negative integer coefficients. f_1, f_2 have this property. The operations $f \mapsto 1 + f$ and $(f, g) \mapsto fg$ preserve this property (since non-negative coefficients are closed under addition and multiplication). Because the coefficients are non-negative, the height $H(f_n)$ (the maximum coefficient) is bounded by the sum of all coefficients, which is $f_n(1)$.
$$H(f_n) \leq f_n(1).$$
We now bound $\log(f_n(1))$. $f_1(1) = 1$, $f_2(1) = 1$. If p is prime: $f_p(1) = 1 + f_{p-1}(1)$. If $n = \prod p_i^{e_i}$: $f_n(1) = \prod f_{p_i}(1)^{e_i}$. We claim $f_n(1) \leq n$ for all $n \geq 1$. We prove this by strong induction. It holds for $n = 1, 2$. If $n = p$ is prime: $f_p(1) = 1 + f_{p-1}(1)$. By hypothesis, $f_{p-1}(1) \leq p - 1$. So $f_p(1) \leq 1 + (p - 1) = p$. If $n = \prod p_i^{e_i}$ is composite: $f_n(1) = \prod f_{p_i}(1)^{e_i}$. By hypothesis, $f_{p_i}(1) \leq p_i$. So $f_n(1) \leq \prod p_i^{e_i} = n$. The claim holds.

  Therefore, we have $H(f_n) \leq f_n(1) \leq n$. This implies $\log(1 + H(f_n)) \leq \log(1 + n)$. Since $\log(1 + n) = O(\log n)$, we can choose $q(x) = c \cdot x$ for some constant c, which is a polynomial. The assumption is satisfied.

□

7 Algorithmic Pipeline for Agent B

Algorithm 1 Factorization-from-Polynomial for Agent **B**

Require: Input $f_n(x) \in \mathbb{Z}[x]$ satisfying **????????**
Ensure: Prime factorization $\mathsf{Fact}(n)$
1: Factor f_n over $\mathbb{Q}$: compute irreducible $g_1, \dots, g_t$ and exponents $e_1, \dots, e_t$ with $f_n = \prod g_j^{e_j}$.
2: **for** $j = 1$ to t **do**
3: Set $p_j := g_j(2)$ (choose sign so $p_j > 0$).
4: **return** $\{(p_j, e_j)\}_{j=1}^t$.

By **????**, the algorithm is correct and runs in time $\mathrm{poly}(\log n)$.

8 Epistemic and Evidence-Theoretic Measures

8.1 Resource-bounded knowledge operators

Definition 8.1 (Operators $K^t_{\mathcal{R}}$)**.** We write

$$K^t_{\mathcal{R}}[\mathsf{Fact}(n)] \quad \text{iff} \quad \exists \text{ algorithm } A : A(\mathsf{Enc}_{\mathcal{R}}(n)) = \mathsf{Fact}(n) \text{ in time } \leq t.$$

By **??**, $K^{\mathrm{poly}(\log n)}_{\mathcal{R}_{\mathrm{poly}}}$ holds under our assumptions; by **??**, $K^{\mathrm{poly}(\log n)}_{\mathcal{R}_{\mathsf{bin}}}$ fails.

These operators obey familiar modal axioms with resource caveats (monotonicity in t, conjunction for joint tasks, etc.).

8.2 Representation-relative instance complexity

For each n, $\mathrm{IC}_{\mathcal{R}}(n)$ is the least time bound that witnesses $K^t_{\mathcal{R}}[\mathsf{Fact}(n)]$. Then **??** gives

$$\mathrm{IC}_{\mathcal{R}_{\mathrm{poly}}}(n) = \mathrm{poly}(\log n), \qquad \mathrm{IC}_{\mathcal{R}_{\mathsf{bin}}}(n) \notin O\big(\mathrm{poly}(\log n)\big) \quad (\text{under } \textbf{??}).$$

8.3 Conditional Kolmogorov complexity

?? formalizes the intuitive statement that, given f_n, a *short* description suffices to reconstruct $\mathsf{Fact}(n)$, while given $\mathsf{bin}(n)$ such a short description typically does not exist uniformly across n.

8.4 Evidence as likelihood gains

Consider the hypothesis H_d : "$d \mid n$". Observation O_g : "g is an irreducible factor of f_n" implies $H_{g(2)}$ with certainty by **??**. Thus the (log) weight of evidence

$$w(H_{g(2)} : O_g) = \log \frac{\Pr[O_g \mid H_{g(2)}]}{\Pr[O_g \mid \neg H_{g(2)}]}$$

is $+\infty$ in the idealized model (or very large under noise), whereas from $\mathsf{bin}(n)$ no analogous immediate certificate arises without performing a hard computation. This captures B's *epistemic* edge as a gain in *decisive* evidence for divisibility claims.

9 Limits, Caveats, and Design Space

Coefficient height is crucial. **??** hinges on **??**. If $\log(\mathrm{ht}(f_n))$ grew super-polynomially in $\log n$ (e.g. exponentially in $\deg f_n$), factoring over $\mathbb{Q}$ could become expensive. Our **??** ensures the total bit-size of the polynomial is $\mathrm{poly}(\log n)$.

Who computes f_n? Our analysis treats f_n as *given*. If constructing f_n from $\mathsf{bin}(n)$ is itself hard (say factoring-hard), then the representation functions as a *compiled* or even *trapdoor* artifact. This does not invalidate B's advantage *given* the representation; it just moves work to an offline producer.

Robustness. The results remain valid if evaluation occurs at any fixed integer $x = x_0$ with $f_p(x_0) = p$. The choice $x_0 = 2$ is only for concreteness.

10 Related Perspectives (Brief)

Knowledge compilation. Representations that make certain queries (here, "factorization") polytime often become larger or more structured; there is a succinctness/tractability trade-off.

Algorithmic knowledge. Resource-bounded accounts of knowledge interpret "knows φ" as "has an algorithm to compute a witness of φ within resources". Our operators $K^t_{\mathcal{R}}$ are in this tradition.

11 Conclusions

Under clear axioms on a polynomial encoding $n \mapsto f_n$, factorization becomes easy relative to the polynomial representation and remains nontrivial relative to the binary representation. We demonstrated a concrete polynomial family that satisfies these axioms. The difference can be measured in running time, description length, and evidence-theoretic terms, providing a precise sense in which *representation is epistemology* for computational tasks.

Appendix A: Small Technical Details

Bit-size of polynomial input. If $d = \deg f_n$ and $H = \mathrm{ht}(f_n)$, the bit-size of the coefficient list is $O(d \log(1 + H))$. Our $\mathrm{poly}(\log n)$ bound implicitly refers to this encoding size.

Sign of irreducibles. If some $g_j(2) < 0$, the sign can be flipped by replacing g_j by $-g_j$ (which is an associate) without changing factorization or exponents. For the family in §6, all f_p have non-negative coefficients, so $f_p(2) = p > 0$ and this is not an issue.

Squarefreeness issues. If n is squarefree, the polynomial f_n is squarefree by **??**. For prime powers p^k, the exponent k is *exactly* the multiplicity of f_p in f_n.

Acknowledgments and Notes

The structure of the assumptions (irreducible codes for primes, multiplicativity, evaluation at a fixed point) is inspired by discussions about polynomial encodings of integers where irreducibility mirrors primality. Standard references for polynomial factorization over $\mathbb{Q}$ include textbooks on computer algebra and the original LLL method.

References

[1] A. K. Lenstra, H. W. Lenstra Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen* 261, 515–534 (1982).

[2] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*, 3rd ed. Cambridge University Press, 2013.

[3] J. Halpern and R. Pucella. A logic for reasoning about algorithmic knowledge. *Information and Computation* 206(9–10): 1302–1332, 2008.

[4] A. Darwiche and P. Marquis. A knowledge compilation map. *Journal of Artificial Intelligence Research* 17: 229–264, 2002.