Electronic Colloquium on Computational Complexity, Report No. 114 (2026)

The Complexity Landscape of Boolean Formula Isomorphism: From Graph Isomorphism to the Second Level

Francesco Cristiano[*]

Abstract

This paper studies the isomorphism problem for Boolean formulas and places it precisely in the polynomial hierarchy. Two of its results are new. The first sharpens the relationship between Boolean and graph isomorphism. Chang's reduction shows only that the unrestricted Boolean isomorphism problem is GI-hard, in one direction; restricting both inputs to canonical form, where equivalence is free, sharpens this to a both-directions many-one equivalence. Encoding each input as a weight-two-minterm canonical DNF rigid enough that every renaming witnessing isomorphism is forced to be a graph isomorphism, we obtain Canonical Formula Isomorphism CFI $\equiv_m$ GI, together with CFNI $\equiv_m$ GNI for the separation problem. This is completeness in both directions, not the one-directional hardness Chang established. The second result is a classification instrument: writing formula isomorphism as $\mathrm{FI} = \exists\lambda \cdot \mathrm{EQ}$ exhibits the renaming as a single existential quantifier over a polynomial-size witness applied to the equivalence problem of the inputs, so the renaming is the only complexity the isomorphism question adds beyond equivalence, and the equivalence cost caps the level and fixes its semantic component, while the renaming search supplies the residual structural complexity, which on canonical inputs is exactly the graph-isomorphism core above. Applying the instrument to compact representations, where equivalence is coNP-complete, places the mixed case DT-FI (an arbitrary formula matched against a DNF target) and the fully compact case CNF-FI at the second level: both lie in Σ_2^P, are coNP-hard and GI-hard, and are complete for no level unless the polynomial hierarchy collapses. The structural facts that frame this level (Schöning's lowness $\Sigma_k^P[\mathrm{GI}] = \Sigma_k^P$ for k $\geq$ 2, under which a graph-isomorphism oracle adds nothing at the second level and above; a first-level boundary at which the graph-isomorphism level lies in NP $\cap$ coNP exactly when GI is low for NP; and a conditional non-completeness from GI $\in$ coAM) are not new, and the paper's role for them is to combine them into a four-case analysis of when the graph-isomorphism core can affect the hierarchy: only when it is complete for its level, and then only at the second. In every case the residue that remains once the equivalence test is made free is graph isomorphism itself.

Keywords: Boolean isomorphism, graph isomorphism, polynomial hierarchy, lowness, Arthur–Merlin classes, normal forms, compactness, representation.

1 Introduction

A Boolean function of arity n is a mapping f : $\{0,1\}^n \to \{0,1\}$; a Boolean formula is one of the infinitely many syntactic objects denoting it. The gap between a function and its expressions is the source of every question studied here. Two relational problems sit across that gap. Formula Equivalence (FE) asks whether two formulas denote the same function, F $\equiv$ G, a purely semantic question [HURY04, TH00]. Formula Isomorphism (FI) asks whether the functions become equal after a renaming of variables, F $\equiv G \circ \lambda$ [AGTH96, AGTH00]; for example x $\wedge \neg y$ and $\neg x \wedge$ y are isomorphic via the transposition of x and y. FI mixes the registers: the renaming is syntactic, the equality it must achieve is semantic. Agrawal and Thierauf showed FI $\in \Sigma_2^P$ and FI not Σ_2^P-complete unless the polynomial hierarchy (PH) collapses [AGTH96, AGTH00].

---

[*] fra.cristiano@gmail.com

ISSN 1433-8092

A more rigid variant requires the renaming to make G string-identical to F; for DNF inputs this syntactic isomorphism, DSFI, is GI-complete [ACL12]. Equivalence and syntactic isomorphism are the two extremes (the semantic component and the syntactic component) and ordinary isomorphism is what a single permutation achieves when asked to satisfy both. The thesis of this paper is that formula isomorphism has two independent sources of complexity, a semantic component (the cost of deciding equivalence in the chosen representation) and a structural component (the search for a variable renaming), separated by the factorisation $\mathrm{FI} = \exists\lambda \cdot \mathrm{EQ}$: the equivalence cost caps the level and fixes the semantic component, while the renaming search supplies the residual structural complexity, which on canonical inputs is precisely graph isomorphism.

1.1 The factorisation principle

Fix a representation class R (Definition 1, §2.1) closed under variable renaming, with equivalence problem EQ_R and isomorphism problem FI_R. The defining identity is

$$\langle A, B\rangle \in \mathrm{FI}_R \iff \exists\lambda : A \equiv B \circ \lambda \iff \exists\lambda : \langle A, B \circ \lambda\rangle \in \mathrm{EQ}_R$$

The permutation λ is a polynomial-size object, so the renaming contributes exactly one existential quantifier over a small witness, and the only complexity beyond guessing it is the cost of EQ_R. In a representation with canonical forms $\mathrm{EQ}_R \in \mathrm{P}$, i.e. the equivalence check contributes no complexity beyond polynomial time, and FI reduces to testing whether a single renaming makes the two inputs equal; in a compact representation EQ_R is coNP-complete and the prefix is $\exists \cdot \mathrm{coNP} = \Sigma_2^P$. The cost of equivalence therefore caps the level and fixes its semantic component, while the search for a renaming contributes the residual structural complexity isolated in Section 6.1. Because that cost is governed by whichever input is harder to test, a single input in compact form is enough to push the problem to the second level.

1.2 Positive and negative isomorphism

Whether the graph-isomorphism structure contributes to the hardness depends on the quantifier (existential or universal) applied to the renaming. A positive question, namely whether a renaming exists, lies in NP whenever the renamed inputs can be tested for equivalence in polynomial time — as they can for canonical forms: the renaming is its own certificate, so a nondeterministic machine can simply guess it and verify, with no oracle required. When equivalence is itself costly the guess sits atop a coNP test and the positive question rises with it (Section 4). The hardness of GI resides in its negative form. Graph non-isomorphism, i.e. no renaming makes two structures match, has no known short certificate, so a nondeterministic machine cannot settle it simply by guessing. To make graph isomorphism a genuine source of hardness, one must therefore use its negative form rather than its positive form; this is the role of the complement problem CFNI below, and it is also why no oracle is needed to place the positive problems.

1.3 Three Cases across the Representation Hierarchy

This paper applies the principle to three problems that differ only in how their inputs are represented:

- CFI : both inputs in canonical form. Equivalence is free, and the problem reduces to graph isomorphism (Section 3).
- DT-FI : an arbitrary formula matched against a DNF target. As the DNF-targeted restriction of FI, its compact arguments make equivalence coNP-complete, placing it at the second level (Section 4.1).

- CNF-FI : both inputs compact (CNF formulas). It too lies at the second level (Section 4.2).

In every case the level's semantic component is fixed by the cost of equivalence, the renaming layer being the common graph-isomorphism core: the three placements are one classification applied across the representation hierarchy, not three independent arguments (Figure 3, Section 4.3).

1.4 Contributions

The main results are:

Theorem 1: FI factorises as $\exists\lambda\cdot\mathrm{EQ}$ over any renaming-closed representation. The identity itself is elementary; its contribution is as the classification tool that organises the paper, separating the semantic component of each FI_R, the cost of its EQ_R that caps the level, from the renaming search, the residual structural component, so that the placements below follow by applying it rather than by separate ad hoc arguments.

Theorem 2: CFI is GI-complete, and its complement CFNI is GNI-complete (Theorem 3). This sharpens Chang's one-directional GI-hardness to a many-one equivalence in both directions: the reverse reduction uses a weight-two encoding rigid enough to force every renaming witnessing isomorphism to be a graph isomorphism. Both reductions are given in full.

Theorems 4–6: DT-FI coincides with FI restricted to a DNF target (no minimisation occurs), lies in Σ_2^P with no oracle, and is coNP-hard and GI-hard.

Proposition 3: CNF-FI lies in Σ_2^P and is coNP-hard and GI-hard — the fully compact instance contrasting with CFI.

Theorem 7, Proposition 4, and Theorem 8: The problems are complete for no level unless PH collapses (the interactive-proof structure of GNI); the GI content is absorbed at the second level and above (Schöning lowness, k $\geq$ 2); and the GI level descends into NP $\cap$ coNP exactly when GI is low for NP (open at $k = 1$).

Corollary 1: A four-case classification of the graph-isomorphism level, identifying the unique case in which it can collapse PH — and then only to the second level.

Two contributions are new and carry the paper. The first is the both-directions completeness CFI $\equiv_m$ GI, with CFNI $\equiv_m$ GNI for the separation problem: Chang's reduction [BRS98, Prop. 12] gives only that the unrestricted Boolean isomorphism problem is GI-hard, and restricting to canonical inputs, where equivalence is free, sharpens this to a many-one equivalence, the hardness direction GI $\leq_m$ CFI supplied by a weight-two encoding rigid enough that every renaming witnessing isomorphism is a graph isomorphism, and membership CFI $\leq_m$ GI by the structural equivalence of [LUKS99]. The second is the use of the factorisation $\mathrm{FI}_R \in \exists\cdot\mathrm{EQ}_R$ as a classification tool, placing each problem by the cost of its equivalence test and exhibiting graph isomorphism as the residue that remains once that test is free. The structural bounds that frame these results are established facts the paper situates rather than reproves: the absorption of the graph-isomorphism oracle at the second level and above is Schöning's lowness [SCH88], and the conditional non-completeness rests on GI $\in$ coAM [GMW91] with the collapse barrier of Boppana, Håstad and Zachos [BHZ87].

2 Preliminaries and Definitions

For foundational material this paper follows [ARBA09, BOCRE06, RO05]. The polynomial hierarchy is generated by $\Sigma_0^P = \Pi_0^P = P$ and $\Sigma_{k+1}^P = \mathrm{NP}^{\Sigma_k^P}, \Pi_{k+1}^P = \mathrm{coNP}^{\Sigma_k^P}$, with $\Sigma_2^P = \mathrm{NP}^{\mathrm{NP}}$ the second level; in operator form $\Sigma_{k+1}^P = \mathrm{NP} \cdot \Pi_k^P$. It is standard that $\Sigma_k^P \neq \Sigma_{k+1}^P \implies P \neq \mathrm{NP}$ and P $\neq$ NP $\implies P \neq$ PH [BOCRE06]. None of the results below implies any collapse.

Known facts about GI. This paper uses four classical facts. (i) GI $\in$ NP. (ii) GNI $\in$ AM, equivalently GI $\in$ coAM [GMW91]. (iii) coNP $\subseteq$ AM $\Longrightarrow$ PH $= \Sigma_2^P$ [BHZ87]; with (ii) this gives the corollary that GI is not NP-complete unless PH collapses. (iv) $\Sigma_2^P[\text{GI}] = \Sigma_2^P$: GI is low for the second level [SCH88]. Babai's quasi-polynomial-time algorithm for GI [BAB16] is the major recent advance, but as a worst-case running-time result it leaves facts (i)–(iv) and the completeness and lowness structure unchanged, and none of the conclusions here depend on it. Fact (iii) gives an upper bound: GI is not NP-complete unless the hierarchy collapses. Fact (iv) gives a lower bound on where GI is absorbed: any problem solvable with a GI oracle at the second level is already solvable without one. The formula isomorphism problem on canonical inputs sits precisely in the territory these two facts leave open.

2.1 Representation classes, normal forms, and isomorphisms

Definition 1 (Representation class)**.** A representation class R is a set of syntactic objects — strings, formulas, circuits, truth tables, or the like — each of which denotes a Boolean function over some variable set $X = \{x_1, ..., x_n\}$, where n may depend on the object, subject to two requirements. First, R is closed under variable renaming: if A $\in$ R and λ is a permutation of X, then $A \circ \lambda \in$ R, so that applying a renaming to an input keeps it in R. Second, R has a well-defined equivalence problem EQ_R: given A, B $\in$ R, decide whether they denote the same function. These are the only properties used below; the cost of EQ_R is the single quantity that, by Theorem 1, fixes the level of the associated isomorphism problem. The instances studied here are the class of canonical DNFs (equivalently full disjunctive normal forms), for which EQ_R is decidable in polynomial time; the class of arbitrary Boolean formulas and the class of CNF formulas, for which EQ_R is coNP-complete; and the mixed setting in which the two arguments are drawn from different classes (Section 4.1). We call R canonical in the first case and compact (equivalently succinct) in the second. A compact Boolean formula is a general formula (an arbitrary formula, or a CNF or DNF whose clauses or terms need not be maxterms or minterms), not a canonical full-expansion form of Definition 2 (canonical DNF, equivalently the truth table / on-set ON(F), and canonical CNF). The dividing line is canonicity, not size: a canonical form is its function's unique representative up to reordering, so EQ_R is a polynomial syntactic comparison; a compact formula is one of many representatives of the same function, possibly exponentially smaller than its canonical DNF/CNF, so EQ_R is coNP-complete. Theorem 1 turns precisely this coNP-completeness into the second level, so throughout, 'compact' denotes such a general formula and never a canonical DNF or CNF. The rise of EQ_R from polynomial to coNP-complete as the input passes from canonical to compact is an instance of the classical phenomenon that a succinct input representation raises complexity [WAG86].

Definition 2 (Canonical DNF and CNF)**.** For F over $X = \{x_1, ..., x_n\}$, a minterm is a conjunction of n literals using each variable once. The canonical DNF F^c is the disjunction of the minterms satisfied by F, unique up to reordering [CH11]; this paper identifies it with its on-set ON(F) $\subseteq \{0,1\}^n$, the set of assignments on which F evaluates to 1 (its satisfying assignments), on which a renaming acts by permuting coordinates. Dually, the canonical (full) CNF is the conjunction of the maxterms ruled out by ON(F), also unique [CH11]; two canonical CNFs over the same variables are equivalent iff identical, exactly as for canonical DNFs.

Proposition 1. *Every non-contradictory formula has a unique canonical* DNF*, and every non-tautological formula a unique canonical* CNF*, modulo the reorderings of Definition 2 [CH11].*

Definition 3 (Formula Equivalence: FE, Formula Isomorphism: FI)**.** $\langle F_1, F_2 \rangle \in$ FE iff $F_1(\alpha) = F_2(\alpha)$ for all α; $\langle F_1, F_2 \rangle \in$ FI iff $\exists \lambda$ with $F_1 \equiv F_2 \circ \lambda$ [TH00]. For DNF inputs the equivalence and isomorphism problems are written DFE and DFI.

Definition 4 (DNF Syntactic Formula Isomorphism: DSFI)**.** $\langle A, B \rangle \in$ DSFI iff some renaming makes A and B identical as DNF formulas (up to reordering). DSFI is GI-complete (the DNF dual of the CNF result of [ACL12], immediate since complementation exchanges DNF and CNF without changing the renaming); DSFI $\in$ NP follows immediately: an NP machine guesses the renaming λ and verifies in polynomial time that it makes the two DNF formulas syntactically identical.

2.2 The equivalence layer

Definition 5 (Tautology Problem: TAUT)**.** $\langle F\rangle \in \mathrm{TAUT}$ iff $F(\alpha) = 1$ for all α. TAUT is coNP-complete [COOK71].

Proposition 2 (DFE is coNP-complete)**.** DFE $\in$ coNP *by universal evaluation. For hardness, a* DNF *F over $\{x_1, ..., x_n\}$ is a tautology iff $F \equiv T_n$, where $T_n = \bigvee_i (x_i \vee \neg x_i)$ is a* DNF *tautology over the same variables;* DNF*-tautology is* coNP*-complete, so* TAUT $\leq^p_m$ DFE. ■

The equivalence cost across the range of representations. For inputs in canonical form EQ is in P (comparison of canonical forms); for compact inputs (formulas, CNF, arbitrary DNF) EQ is coNP-complete. By Theorem 1 the cost of equivalence caps the level and fixes its semantic component, and that cost is dictated by the harder of the two inputs.

2.3 The four problems

Definition 6 (Canonical Formula Isomorphism: CFI, Canonical Formula Non-Isomorphism: CFNI)**.** CFI is FI restricted to inputs given in canonical form (canonical DNF, identified throughout with the on-set $\mathrm{ON}(F) \subseteq \{0,1\}^n$, equivalently the minterm list), where EQ is polynomial: $\langle F_1, F_2\rangle \in \mathrm{CFI}$ iff some coordinate permutation λ satisfies $\lambda \cdot \mathrm{ON}(F_2) = \mathrm{ON}(F_1)$. CFNI is its complement on the same inputs: the claim that no permutation of variables makes F_1 and F_2 equivalent.

Remark (the on-set encoding is the operative one)**.** CFI is defined on the on-set / minterm-list encoding, on which equivalence is a comparison of minterm sets and the reductions of Theorem 2 are polynomial. This is not interchangeable with the dense truth-table encoding: a truth table has size 2^n, so on truth-table inputs equivalence under a renaming is decidable in NC [LUKS99], and the GI-hardness reduction below (whose image f_G has an exponential truth table for sparse G) is super-polynomial there. The GI-completeness of CFI is therefore a statement about the succinct on-set encoding; under the dense encoding the same isomorphism question is strictly easier unless GI $\in$ NC.

Definition 7 (DNF-Targeted Formula Isomorphism: DT-FI)**.** Given an arbitrary formula F_1 and a target DNF F_2 with $|F_2| = l$, $\langle F_1, F_2\rangle \in \mathrm{DT\text{-}FI}$ iff there is a DNF G with (i) $|G| = l$, (ii) G $\equiv F_1$, and (iii) $\langle G, F_2\rangle \in \mathrm{DSFI}$. Equivalently: given a Boolean formula and a target DNF, can the first be rewritten as a DNF of the same size that is a variable renaming of the target?

Definition 8 (CNF Formula Isomorphism: CNF-FI)**.** $\langle \varphi_1, \varphi_2\rangle \in \mathrm{CNF\text{-}FI}$ iff φ_1, φ_2 are CNF formulas and some renaming λ makes $\varphi_2 \circ \lambda \equiv \varphi_1$ (logically, not string-identically).

3 Isomorphism on Canonical Inputs: The Graph Isomorphism Level

3.1 The factorisation

Theorem 1 (Factorisation)**.** *For any R closed under variable renaming, $\langle A, B\rangle \in \mathrm{FI}_R$ iff $\exists\lambda : \langle A, B \circ \lambda\rangle \in \mathrm{EQ}_R$. Hence $\mathrm{FI}_R \in \exists \cdot \mathrm{EQ}_R$.*

Proof. By Definition 3, $\langle A, B\rangle \in \mathrm{FI}_R$ means there is a permutation λ with $\mathrm{A} \equiv B \circ \lambda$. Closure under renaming (Definition 1) guarantees $B \circ \lambda \in \mathrm{R}$, so the equivalence $\mathrm{A} \equiv B \circ \lambda$ is exactly the statement $\langle A, B \circ \lambda\rangle \in \mathrm{EQ}_R$. Quantifying over λ gives $\langle A, B\rangle \in \mathrm{FI}_R \iff \exists\lambda : \langle A, B \circ \lambda\rangle \in \mathrm{EQ}_R$. ■

Remark (canonical and compact as endpoints)**.** The canonical and compact classes instantiated above are the extremes of the cost of EQ_R, not an exhaustive partition. An arbitrary Boolean formula is therefore not a separate, intermediate case: its equivalence test is coNP-complete (TAUT is the $\langle F, 1\rangle$ instance), so by Theorem 1 it sits squarely at the compact end, $\mathrm{FI} = \exists\lambda \cdot \mathrm{EQ} = \exists \cdot \mathrm{coNP} = \Sigma_2^P$, which is the general problem FI of Definition 3 (of which CNF-FI and DT-FI are restrictions). Since $\mathrm{FI}_R = \exists\lambda \cdot \mathrm{EQ}_R$ transfers the cost of EQ_R wherever it lies, classes of intermediate equivalence also have a definite place: free (read-once) branching programs, whose equivalence is in coRP, give $\mathrm{FI}_R \in \exists \cdot coRP = \mathrm{NP} \cdot coRP$ [TH00], between the canonical endpoint (NP) and the compact endpoint (Σ_2^P). Fixed-order reduced OBDDs are canonical yet not renaming-closed in polynomial size, and so fall outside Definition 1; admitting per-OBDD orders restores closure but makes EQ_R cross-order OBDD equivalence, a read-once-BP equivalence instance in coRP [BCW80] and not known to be in P, whence OBDD-FI $\in \exists \cdot coRP = \mathrm{NP} \cdot coRP$ alongside the read-once case, not at the NP endpoint.

Remark (provenance of the two-stage form)**.** The factorisation is not new as a proof technique: the Σ_2^P-membership arguments of Borchert, Ranjan and Stephan [BRS98, Prop. 5] and of Agrawal and Thierauf [AGTH00] both proceed by guessing a renaming and then verifying agreement on all assignments, that is $\exists \cdot \mathrm{coNP}$. What Theorem 1 adds is the use of this form as a representation-indexed classification across the whole spectrum, reading the level of any FI_R off the cost of EQ_R.

This theorem presents formula isomorphism as two stages applied in turn. The outer stage is a single existential quantifier: guess a renaming λ of the variables, a polynomial-size object that a nondeterministic machine can produce directly. The inner stage is an equivalence test: check whether the renamed formula $B \circ \lambda$ denotes the same function as A. Crucially, the renaming adds only that one quantifier to the prefix, the same combinatorial step regardless of how the inputs are written, so the equivalence cost caps the level and fixes its semantic component. The renaming search supplies the residual structural component within that level, which on canonical inputs is exactly the graph-isomorphism core and, as the Boolean-constraint classification of Section 6.3 shows, need not be trivial. Figure 1 displays this two-stage structure and traces its two outcomes: when the representation admits canonical forms the equivalence test runs in polynomial time and the problem reduces to deciding the renaming alone, which is graph isomorphism (the left branch, $\mathrm{CFI} \equiv_m \mathrm{GI}$); when the representation is compact the equivalence test is coNP-complete and the problem rises to the second level (the right branch, $\mathrm{FI} \in \exists{\cdot}\mathrm{coNP} = \Sigma_2^P$).

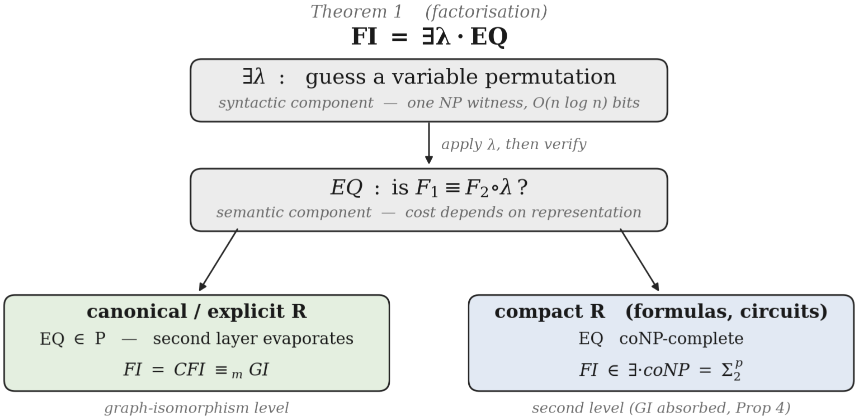


Figure 1. The factorisation FI = $\exists\lambda \cdot$ EQ as a two-layer quantifier prefix. The existential renaming (syntactic component) feeds the equivalence check (semantic component); the cost of the lower layer, set by the representation, caps the level and fixes its semantic component, while the renaming layer carries the residual graph-isomorphism core. When both inputs are canonical, EQ is polynomial-time decidable and the problem is the graph-isomorphism core (CFI $\equiv_m$ GI); when an input is compact, EQ is coNP-complete and the problem lies in Σ_2^P, where the graph-isomorphism oracle is absorbed (Proposition 4).

For a Boolean formula F write $\mathcal{D}(F) := \{$ D : D is a DNF formula and D $\equiv$ F $\}$ for its DNF representation class. Every F has at least one DNF representative (the canonical DNF F^c when F is non-contradictory, and a DNF for the constant 0 otherwise), so $\mathcal{D}(F) \neq \emptyset$.

Lemma 1 (Syntactic kernel of FI)**.** *For arbitrary Boolean formulas* $F_1, F_2, \langle F_1, F_2\rangle \in$ FI $\iff \forall\ D_2 \in \mathcal{D}(F_2)\ \exists\ D_1 \in \mathcal{D}(F_1) : \langle D_1, D_2\rangle \in$ DSFI. *Because* $\mathcal{D}(F_2) \neq \emptyset$ *and a single renaming witnesses the right-hand side, the universal and existential forms coincide:* $\langle F_1, F_2\rangle \in$ FI $\iff \exists\ D_2 \in \mathcal{D}(F_2)\ \exists\ D_1 \in \mathcal{D}(F_1) : \langle D_1, D_2\rangle \in$ DSFI.

Proof. ($\Leftarrow$) Suppose $D_2 \in \mathcal{D}(F_2)$ and $D_1 \in \mathcal{D}(F_1)$ satisfy $\langle D_1, D_2\rangle \in$ DSFI; by Definition 4 there is a renaming λ with $D_1 = D_2 \circ \lambda$, identical as DNF formulas up to reordering. Since $D_1 \equiv F_1$ and $D_2 \equiv F_2$, and renaming respects equivalence, $F_1 \equiv D_1 = D_2 \circ \lambda \equiv F_2 \circ \lambda$, so $F_1 \equiv F_2 \circ \lambda$ and $\langle F_1, F_2\rangle \in$ FI. A single witnessing pair suffices, which already gives the existential form. ($\Rightarrow$) Suppose $\langle F_1, F_2\rangle \in$ FI and fix λ with $F_1 \equiv F_2 \circ \lambda$. Let $D_2 \in \mathcal{D}(F_2)$ be arbitrary and set $D_1 := D_2 \circ \lambda$. Renaming preserves DNF form, so D_1 is a DNF; and $D_1 = D_2 \circ \lambda \equiv F_2 \circ \lambda \equiv F_1$, so $D_1 \in \mathcal{D}(F_1)$. By construction $D_1 = D_2 \circ \lambda$, hence $\langle D_1, D_2\rangle \in$ DSFI. The renaming λ was fixed before D_2 was chosen, so the same λ serves every $D_2 \in \mathcal{D}(F_2)$, giving the universal form. ■

Lemma 1 is the syntactic-kernel counterpart of Theorem 1. Theorem 1 isolates the semantic cost (equivalence) that any isomorphism test must pay; Lemma 1 isolates the syntactic object (the GI-complete kernel DSFI) that this cost gates. Together they say that FI carries DSFI inside it at every representation, reachable only through an equivalence test whose price the representation fixes.

Remark (a characterisation, not a reduction)**.** Lemma 1 does not give FI $\leq_m^p$ DSFI, equivalently FI $\leq_m^p$ GI. The quantifier ranges over $\mathcal{D}(F_2)$, an infinite family of unbounded-size DNFs, and certifying $D_1 \in \mathcal{D}(F_1)$ hides the coNP-complete test $D_1 \equiv F_1$. Were the equivalence a polynomial-time many-one reduction, then DSFI $\in$ NP (Definition 4) would force FI $\in$ NP; but FI is coNP-hard ($\langle F, 1\rangle \in$ FI iff F $\in$ TAUT, and TAUT is coNP-complete [COOK71]), so

NP = coNP and PH collapses. The kernel DSFI is therefore logically inside FI but not polynomially accessible there; what makes it accessible is canonicity, as the next theorem shows.

Remark (Theorem 2 as the canonical collapse of Lemma 1)**.** Take the inputs in canonical form, so F_2 is presented as F_2^c and F_1 as F_1^c (Definition 2). Two reductions of Lemma 1 then occur at once: the existential form supplies the representative $D_2 = F_2^c$ with no search, and the side-condition $D_1 \in \mathcal{D}(F_1)$ (in general a coNP test) becomes the polynomial check that D_1 and F_1^c agree as minterm sets. The quantified statement collapses to the single instance $\langle F_1^c, F_2^c \rangle \in$ DSFI, which holds iff some λ carries ON(F_2) onto ON(F_1), i.e. iff $\langle F_1, F_2 \rangle \in$ CFI. This is exactly the equivalence CFI $\equiv_m$ GI of Theorem 2: Lemma 1 is the general identity, and Theorem 2 is its specialisation to inputs on which the representation class is a single, polynomially comparable point.

3.2 CFI **is graph-isomorphism-complete**

Theorem 2. CFI *is* GI*-complete under polynomial-time many-one reductions.*

Membership (CFI $\leq_m^p$ GI)**.** By Definition 6 the task is to decide whether a coordinate permutation maps ON(F_2) onto ON(F_1). We may assume F_1 and F_2 are over the same number of variables and that $|\text{ON}(F_1)| = |\text{ON}(F_2)|$; if either fails, the two graphs built below differ in their vertex counts and the instance is mapped to a fixed no-instance of GI. Encode each on-set as a two-coloured bipartite graph $\Gamma(F)$: n vertices coloured "coordinate", one vertex per point of ON(F) coloured "point", and an edge from coordinate i to point α iff $\alpha_i = 1$. By construction every edge runs between the coordinate side and the point side, so $\Gamma(F)$ is bipartite, hence triangle-free. A colour-preserving isomorphism $\Gamma(F_2) \cong \Gamma(F_1)$ restricts on the coordinate side to a permutation λ with $\lambda \cdot \text{ON}(F_2) = \text{ON}(F_1)$, and conversely any such λ induces the point bijection. It remains to discharge the two colours into plain graph isomorphism. (Identifying each on-point with its support, CFI is precisely the isomorphism problem for the hypergraph whose vertices are the coordinates and whose hyperedges are the on-points; Theorem 2 thus recasts the classical equivalence of hypergraph isomorphism and GI in Boolean form [LUKS99].)

The colour gadget. Fix two clique sizes a < b, both larger than the maximum degree occurring in either $\Gamma(F_1)$ or $\Gamma(F_2)$ (e.g. $a = n + |\text{ON}| + 1, b = a + 1$, where $|\text{ON}| = |\text{ON}(F_1)| = |\text{ON}(F_2)|$; the same a, b are used for both graphs). Attach to every "coordinate" vertex a fresh clique K_a by a single edge, and to every "point" vertex a fresh clique K_b; write Γ(F) for the resulting uncoloured graph, polynomial in the input. ($\Rightarrow$) A colour-preserving isomorphism extends to the tails, preserving colours hence tail sizes. ($\Leftarrow$) Any isomorphism h of $\Gamma(F_2), \Gamma(F_1)$ carries tail vertices to tail vertices and original vertices to original vertices, since original vertices lie in no clique of size ≥ 3 ($\Gamma(F)$ is triangle-free and each tail is attached by one edge), while tail vertices lie in a clique of size a or b ≥ 3; the distinct sizes a $\neq$ b then force h to respect the two colour classes. So h restricts to a colour-preserving isomorphism, and CFI on $\langle F_1, F_2 \rangle$ holds iff $\Gamma(F_2) \cong \Gamma(F_1)$. Hence CFI $\leq_m^p$ GI.

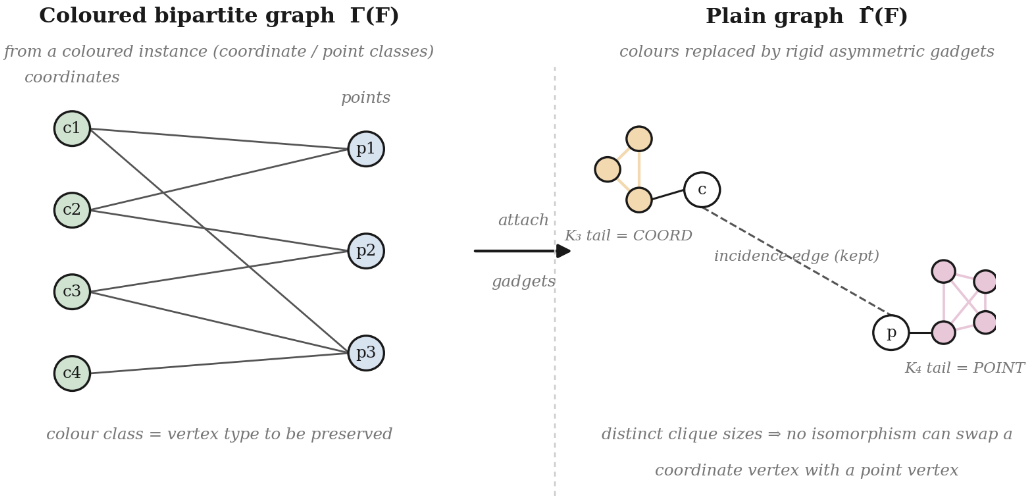


Figure 2. Discharging the two colours of $\Gamma(F)$ into plain graph isomorphism. Each "coordinate" vertex receives a K_a tail and each "point" vertex a K_b tail, with a $\neq$ b both exceeding every original degree. The distinct, rigid clique sizes force any isomorphism of the uncoloured graph Γ̂(F) to preserve the colour classes; the original incidence edges (dashed) are untouched.

Hardness (GI $\leq_m^p$ CFI)**.** Let the variables be vertices and the satisfying assignments edges. Take G a simple graph on [n] with at least one edge (edgeless graphs are decided directly, mapping to the always-false function). For a pair {i, j} let χ_{ij} be the assignment with ones exactly in coordinates i and j, and m_{ij} its minterm; set $f_G = \bigvee_{\{}\{i, j\} \in E\}\ m_{ij}$, a canonical DNF of $n \cdot |E| \leq n^3$ literals with ON(f_G) = { χ_{ij} : {i,j} ∈ E }, all of Hamming weight two. A coordinate permutation preserves weight and acts by $\chi_{ij} \mapsto \chi_{\{}\lambda(i), \lambda(j)\}$, the action of a vertex permutation on potential edges. If σ is an isomorphism G → H it carries ON(f_G) onto ON(f_H), giving $\langle f_G, f_H \rangle \in$ CFI; conversely a witnessing λ carries the weight-two on-sets onto one another and so induces the edge bijection, an isomorphism G → H. Thus G ≅ H iff f_G ≅ f_H, and GI $\leq_m^p$ CFI. The weight-two on-set encoding plays, for canonical inputs, the role that Chang's reduction GI $\leq_m^p$ BOOLE-ISO plays for formula inputs [BRS98, Prop. 12]; it is a distinct construction, not Chang's rendered canonically. Chang encodes G by $h_G = \bigvee_{\{}\{i, j\} \in E\}\ (v_i \wedge v_j)$, whose two-variable terms give a function with an exponential canonical DNF in general, whereas f_G is the weight-two-minterm function with ON(f_G) = { χ_{ij} : {i,j} ∈ E }. Theorem 2 sharpens Chang's one-directional hardness to a both-directions equivalence on canonical inputs. ■

Observation. The set of isomorphisms G → H and the set of permutations witnessing f_G ≅ f_H are the same set of permutations of [n]; for $G = H$ the automorphism groups coincide. The weight-two confinement is the rigidity, supplied by the encoding.

Example (the triangle K_3)**.** For $G = K_3$ on {1,2,3} the three edges give $f_{\{}K_3\} = (x_1 \wedge x_2 \wedge \neg x_3) \vee (x_1 \wedge \neg x_2 \wedge x_3) \vee (\neg x_1 \wedge x_2 \wedge x_3)$, with ON $= \{110, 101, 011\}$, the weight-two points of the cube. Its automorphism group is S_3, matching that of K_3, so all 3! permutations witness self-isomorphism as the comment requires. Dropping the edge {2,3} deletes $\neg x_1 \wedge x_2 \wedge x_3$, leaving ON $= \{110, 101\}$; the resulting $f_{\{}P_3\}$ is non-isomorphic to $f_{\{}K_3\}$, as $P_3 \ncong K_3$.

Relationship with Theorem 1. On canonical inputs EQ$_R$ ∈ P, so by Theorem 1 the problem is $\exists{\cdot}P$ = NP at most, and the explicit reduction CFI $\leq^p$ GI then pins it exactly at the

GI level. Without Theorem 1 one would have to argue from scratch why the semantic check does not add an extra quantifier layer.

3.3 The separation problem

Theorem 3. CFNI *is* GNI-*complete under polynomial-time many-one reductions.*

Proof. CFNI is the complement of CFI on the same inputs. Let f be the reduction with GI $\leq_m^p$ CFI (Theorem 2), so x $\in$ GI $\iff f(x) \in$ CFI; negating both sides, x $\in$ GNI $\iff f(x) \in$ CFNI, witnessing GNI $\leq_m^p$ CFNI. Symmetrically CFI $\leq_m^p$ GI gives CFNI $\leq_m^p$ GNI. ∎

Note. CFNI admits two equivalent readings: semantically it asserts the inequivalence of two functions under every renaming; syntactically, by Theorem 2, it is graph non-isomorphism. This is the universal-quantifier instance of the correspondence of §1.2, and it places the graph-isomorphism content in a form that cannot be absorbed into an existential quantifier.

Corollary (a zero-knowledge proof for CFI)**.** Since CFI is many-one equivalent to GI (Theorem 2), the zero-knowledge protocol of Goldreich, Micali and Wigderson for graph isomorphism [GMW91] transfers directly: a prover can convince a verifier that two formulas in canonical form are isomorphic (that some renaming makes them equivalent) without revealing the witnessing permutation λ. Dually, the constant-round interactive proof for graph non-isomorphism gives an Arthur–Merlin proof of separation for CFNI. These are immediate corollaries of the equivalences in Theorems 2 and 3 and require no further construction.

4 The Compact Form and the Mixed Case: The Second Level

4.1 DT-FI: an arbitrary formula and a DNF target

Theorem 4 (Reparametrisation)**.** $\langle F_1, F_2\rangle \in$ DT-FI *iff* $\exists\lambda : F_2 \circ \lambda \equiv F_1$. *Equivalently,* DT-FI *is* FI *restricted to a* DNF *target, and conditions (i), (iii) of Definition 7 are redundant; no minimisation occurs.*

Proof. ($\Leftarrow$) Given λ with $F_2 \circ \lambda \equiv F_1$, set G := $F_2 \circ \lambda$; renaming preserves the literal count so $|G| = |F_2| = l, G \equiv F_1$, and λ^{-1} witnesses $\langle G, F_2\rangle \in$ DSFI. ($\Rightarrow$) From DSFI, $G = F_2 \circ \mu$ up to reordering, so $F_2 \circ \mu \equiv G \equiv F_1$; take $\lambda = \mu$. ∎

Theorem 5. DT-FI $\in \Sigma_2^P$.

Proof. By Theorem 4, $\langle F_1, F_2\rangle \in$ DT-FI iff $\exists\lambda\ \forall\alpha\ [\ F_1(\alpha) = (F_2 \circ \lambda)(\alpha)\ ]$, the canonical $\exists \cdot \forall$ form, so DT-FI $\in \Sigma_2^P$. The DSFI side-condition is in NP (Definition 4), so its renaming witness is guessed in the existential phase; equivalently, this is Theorem 1 with EQ the equivalence of an arbitrary formula with a DNF, a coNP predicate, giving $\exists \cdot$ coNP $= \Sigma_2^P$. ∎

Theorem 6. DT-FI *is* coNP-*hard and* GI-*hard; moreover* DFI $\leq_m^p$ DT-FI.

Proof. DFI $\leq_m^p$ DT-FI is the identity map (Theorem 4 with DNF inputs). For coNP-hardness, map F over $\{x_1, ..., x_n\}$ to $\langle F, T_n\rangle$ with $T_n = \bigvee_i (x_i \vee \neg x_i)$; since $T_n \circ \lambda \equiv T_n \equiv 1, \langle F, T_n\rangle \in$ DT-FI iff F is a tautology, and TAUT is coNP-complete [COOK71]. For GI-hardness, map a GI instance $\langle G, H\rangle$ to $\langle f_G, f_H\rangle$ with f_G, f_H the weight-two encodings of Theorem 2 (here f_G is the arbitrary first argument, f_H the DNF target); by Theorem 4 this is in DT-FI iff $\exists\lambda :$ $f_H \circ \lambda \equiv$ f_G iff G $\cong$ H. ∎

Remark. Because the renaming is queried only positively, it certifies itself and is absorbed into the existential guess; no graph-isomorphism oracle is needed, and by Schöning's theorem (Proposition 4) one would in any case be inert at Σ_2^P. DT-FI is the mixed-representation point of the classification: its compact first argument forces the coNP equivalence cost, and hence the second level.

4.2 CNF-FI: both arguments compact

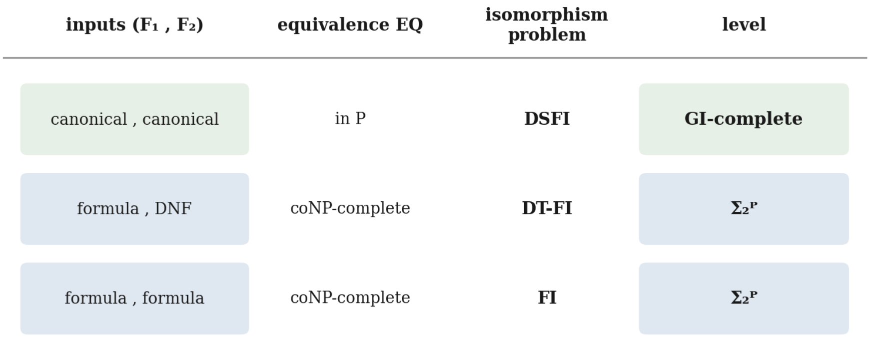
Proposition 3. CNF-FI $\in \Sigma_2^P$*, and* CNF-FI *is both* coNP*-hard and* GI*-hard.*

Proof. Membership is Theorem 1 with R = CNF: CNF equivalence is coNP-complete, so CNF-FI $= \exists\lambda \cdot \mathrm{EQ}_{\mathrm{CNF}} \in \exists \cdot \mathrm{coNP} = \Sigma_2^P$. For coNP-hardness, reduce from CNF-UNSAT, which is coNP-complete. Fix the contradictory CNF$\perp_n = (x_1) \wedge (\neg x_1)$ over the n variables; every renaming of a contradiction is a contradiction, so $\perp_n \circ \lambda \equiv \perp_n \equiv 0$ for every λ, whence $\langle \varphi, \perp_n \rangle \in$ CNF-FI iff $\varphi \equiv 0$ iff φ is unsatisfiable. (The dual route through a tautology target fails for CNF: a CNF is a tautology iff every clause contains a complementary literal pair, so CNF-tautology is decidable in linear time. Tautology is the hard direction for DNF — as used in Proposition 2 — while unsatisfiability is the hard direction for CNF; the choice between them must follow the representation in hand.) For GI-hardness, encode G not by the function f_G of Theorem 2 but by its complement $g_G = \neg f_G$, whose off-set is exactly the weight-two edge vectors. Its canonical CNF carries one maxterm per off-point, hence $\psi_G = \bigwedge_{\{} \{i,j\} \in E\}$ $(\neg x_i \vee \neg x_j \vee \bigvee_{\{} k \neq i$,j} $x_k)$: one clause per edge, so |E| $\leq n^2$ clauses, polynomial in the input. (Encoding the on-set f_G directly, as written, would give a canonical CNF with 2^n- |E| maxterms — exponential — so passing to the dual is essential, not cosmetic.) By Proposition 1 equivalence of canonical CNFs is identity, so $\psi_H \circ \lambda \equiv \psi_G$ iff λ carries the weight-two off-sets onto one another iff $g_H \circ \lambda = g_G$ iff G ≅ H, exactly the argument of Theorem 2. GI-hardness therefore holds already on canonical CNF inputs, consistent with the syntactic GI-completeness of [ACL12] (the restriction of CNF-FI to syntactically canonical inputs), to which it specialises. ■

Note. CNF-FI sits at the same second level as DT-FI, not at the GI level: in Σ_2^P, coNP-hard and GI-hard, and, being a restriction of FI, not Σ_2^P-complete unless PH collapses [AGTH00]. What separates these problems from CFI is the absence of a unique normal form for the inputs, not the choice of conjunctive versus disjunctive syntax: for canonical DNF the semantic and syntactic notions coincide and CFI rests at GI; for CNF the two diverge, equivalence costs coNP, and the same isomorphism core is lifted to Σ_2^P.

4.3 The Complexity Classification by Representation

The three problems differ only in the compactness of their inputs, so they vary along the semantic axis alone: the cost of equivalence is the maximum over the two arguments, polynomial only when both are canonical and coNP-complete as soon as one is compact, while the renaming layer (the graph-isomorphism core) is common to all three. This is the semantic axis of the factorisation traced across representations (Figure 3).



inputs (F_1 , F_2)	equivalence EQ	isomorphism problem	level
canonical , canonical	in P	**DSFI**	**GI-complete**
formula , DNF	coNP-complete	**DT-FI**	**Σ_2^P**
formula , formula	coNP-complete	**FI**	**Σ_2^P**

The level tracks the maximum equivalence cost across the two inputs: a single compact argument forces coNP, hence Σ_2^P.

Figure 3. Three isomorphism problems by input representation. Equivalence is free only when both inputs are canonical, where the problem is GI-complete (DSFI/CFI); any compact argument makes equivalence coNP-complete and lifts the problem to Σ_2^P. DT-FI is the mixed case (one formula, one DNF) and CNF-FI the fully compact case; both sit with FI at the second level.

5 Position in the Polynomial Hierarchy: Bounds on the Graph-Isomorphism Level

5.1 Conditional non-completeness

Theorem 7. CFI *is not* NP*-complete and* CFNI *is not* coNP*-complete unless* PH *collapses to its second level.* DT-FI *and* CNF-FI *are not* Σ_2^P*-complete unless* PH *collapses to its third level.*

Proof. If CFI were NP-complete then GI would be NP-complete (Theorem 2), so GNI would be coNP-complete; but GNI $\in$ AM [GMW91], giving coNP $\subseteq$ AM, which collapses PH to Σ_2^P [BHZ87]. The CFNI statement is the same argument through Theorem 3. For the second-level problems, DT-FI $\leq_m^p$ FI (Theorem 4) and CNF-FI is a restriction of FI. The identity map is a many-one reduction CNF-FI $\leq_m^p$ FI, since a pair of CNF formulas is already an FI instance on which isomorphism is the same relation. Agrawal and Thierauf show that formula non-isomorphism has a two-round interactive proof with an NP-oracle verifier, so FNI $\in BP \cdot \Sigma_2^P$; since a Π_2^P-complete set cannot lie in $BP \cdot \Sigma_2^P$ unless the polynomial hierarchy collapses, FI is not Σ_2^P-complete unless PH $= \Sigma_3^P$ [AGTH00]. If DT-FI or CNF-FI were Σ_2^P-complete, then FI — into which DT-FI reduces and of which CNF-FI is a restriction — would be Σ_2^P-complete as well, forcing the same collapse to the third level. ■

5.2 Lowness of GI at the second level and above

Proposition 4 (lowness of GI above the first level)**.** *For every* $k \geq 2, \Sigma_k^P[\mathrm{GI}] = \Sigma_k^P$.

Proof. The statement is standard: the base case is Schöning's [SCH88], and its propagation to all higher levels is the increasing low hierarchy [KST93]; we record the short argument. The inclusion $\Sigma_k^P \subseteq \Sigma_k^P[\mathrm{GI}]$ is trivial, since a Σ_k^P computation may ignore the oracle, so we prove $\Sigma_k^P[\mathrm{GI}] \subseteq \Sigma_k^P$ by induction on k. Base case ($k = 2$): $\Sigma_2^P[\mathrm{GI}] = \Sigma_2^P$ is Schöning's lowness of GI for the second level [SCH88], resting on GI $\in$ NP $\cap$ coAM (GNI $\in$ AM [GMW91]). Inductive step ($k \geq 3$): what is relativised is the defining recurrence of the hierarchy, $\Sigma_m^P = \mathrm{NP}^{\Sigma_{m-1}^P}$, which holds relative to any oracle A as $\Sigma_m^P[A] = \mathrm{NP}^{\Sigma_{m-1}^P}[A]$ (the outer NP machine needs no separate A-gate, since A $\in \Sigma_1^P[A] \subseteq \Sigma_{m-1}^P[A]$). Taking $A =$ GI and the induction hypothesis $\Sigma_{k-1}^P[\mathrm{GI}] = \Sigma_{k-1}^P$ gives $\Sigma_k^P[\mathrm{GI}] = \mathrm{NP}^{\Sigma_{k-1}^P}[\mathrm{GI}] = \mathrm{NP}^{\Sigma_{k-1}^P} = \Sigma_k^P$. The recurrence removes one level from the top at each step, so every k $\geq$ 2 descends to the base at level two, never to level one. ■

Note (the case $k = 1$ is open). The first-level statement $\Sigma_1^P[\mathrm{GI}] = \Sigma_1^P$, i.e. $\mathrm{NP}^{\mathrm{GI}} = \mathrm{NP}$, is neither given by [SCH88] nor reached by the relativisation, which propagates only upward; it is precisely the open problem characterised in Theorem 8 as equivalent to GI $\in$ coNP. Lowness is established for level two and above; whether it extends to the first level is open, and the two statements must not be conflated.

Discussion (heuristic). The construction cannot be lifted past its level by the obvious strengthening: quantifying existentially over a guessed template with a separation condition "no λ works" gives a $\exists \cdot \forall$ prefix, hence Σ_2^P, where Proposition 4 renders the graph-isomorphism oracle inert. For the graph-isomorphism content to remain essential, the separation condition must be evaluated on the fixed input, as in CFNI, which is what places the problem at the level of GI. This is offered as motivation, not as a proof that no problem strictly between GI and Σ_2^P exists.

5.3 The first-level question $\mathrm{NP}^{\mathrm{GI}} = \mathrm{NP}$

Theorem 8. CFI $\in$ NP $\cap$ coNP *iff* GI $\in$ coNP*, equivalently iff* GI *is low for* NP ($\mathrm{NP}^{\mathrm{GI}} = \mathrm{NP}$)*. The same holds for* CFNI.

Proof. CFI $\in$ NP always, being many-one equivalent to GI $\in$ NP (Theorem 2); so CFI $\in$ NP $\cap$ coNP iff CFI $\in$ coNP iff GI $\in$ coNP iff GNI $\in$ NP; the BOOLE-ISO form of this equivalence, that BOOLE-ISO is not in coNP unless GI $\in$ coNP, is Corollary 13 of [BRS98]. If GI $\in$ NP $\cap$ coNP an NP machine resolves any GI query by guessing the relevant certificate, so $\mathrm{NP}^{\mathrm{GI}} = \mathrm{NP}$; conversely if $\mathrm{NP}^{\mathrm{GI}} = \mathrm{NP}$ then GNI $\in P^{\mathrm{GI}} \subseteq \mathrm{NP}^{\mathrm{GI}} = \mathrm{NP}$, whence GI $\in$ coNP. The argument is unchanged for CFNI. ■

Note. Whether GI is low for NP, i.e. whether non-isomorphism has short certificates, is a long-standing open problem. Theorem 8 restates it in semantic–syntactic terms: the graph-isomorphism level lies in NP $\cap$ coNP exactly when separation under all renamings admits a polynomial certificate. The construction reformulates this open question; it does not resolve it.

Proposition 5 (Representation and the equivalence obstruction)**.** *Canonical Formula Isomorphism lies in* P^{GI}*, hence in* $\mathrm{coNP}^{\mathrm{GI}}$*. For unrestricted* FI*, membership in* $\mathrm{coNP}^{\mathrm{GI}}$ *is open, and is equivalent to certifying formula non-isomorphism by a deterministic* $\mathrm{NP}^{\mathrm{GI}}$ *procedure. The obstruction is the cost of equivalence in the compact representation, not the renaming layer: any route that uses a graph-isomorphism oracle to decide succinct formula equivalence collapses the polynomial hierarchy to its second level.*

Proof. By Theorem 2, CFI $\equiv_m^p$ GI; with the input in canonical form a graph-isomorphism oracle decides it in polynomial time, so CFI $\in P^{\mathrm{GI}} \subseteq \mathrm{coNP}^{\mathrm{GI}}$. For the conditional claim, suppose formula equivalence were decidable in P^{GI} — the natural way to discharge the equivalence test of Theorem 1 by graph-isomorphism queries. Formula equivalence is coNP-complete [COOK71], so $\mathrm{coNP} \subseteq P^{\mathrm{GI}} \subseteq \mathrm{NP}^{\mathrm{GI}}$. To place $\mathrm{NP}^{\mathrm{GI}}$ in AM, simulate the graph-isomorphism oracle directly. An accepting computation is a nondeterministic path together with the answers to its graph-isomorphism queries, and the path and answers fix the queries, so adaptivity is immaterial. Merlin sends the path and the claimed answers; Arthur certifies each YES-answer by the supplied isomorphism, using only GI $\in$ NP, and each NO-answer by the protocol for graph non-isomorphism, using only GNI $\in$ AM [GMW91]. The polynomially many non-isomorphism protocols run in parallel, each amplified so that the union bound keeps the total soundness error below 1/3; completeness is perfect, and any accepting transcript on a rejected input must misreport a query, a false YES lacking an isomorphism and a false NO failing the non-isomorphism protocol. As constant-round Arthur–Merlin collapses to two rounds, the simulation is an AM computation, so $\mathrm{NP}^{\mathrm{GI}} \subseteq \mathrm{AM}$. Hence coNP $\subseteq$ AM, which collapses PH to its second level [BHZ87]. Derandomisation is no escape: derandomising the interactive proof for FNI [AGTH00] would place FI in Σ_2^P, which already holds, with no descent into $\mathrm{coNP}^{\mathrm{GI}}$. ■

Note. The obstruction is the cost of equivalence, not the renaming. Theorem 2 discharges the graph-isomorphism layer in P^{GI}; what remains is recognising equivalence of a function presented compactly, exactly the coNP-complete step a graph-isomorphism oracle cannot supply without collapse. Should FI $\in \mathrm{coNP}^{\mathrm{GI}}$ hold, any witnessing procedure must therefore certify non-isomorphism through global structural invariants that do not decide formula equivalence, since truth-table canonicalisation is exponential in the input. The question is thus adjacent to the open question of Theorem 8: both concern how far the graph-isomorphism level extends when compact equivalence is required, and both are settled only conditionally.

5.4 A classification of the graph-isomorphism level

The lower bounds above, combined with the classical position of GI, yield a complete case analysis of the possible positions of the shared graph-isomorphism core and of the conditions under which it forces a collapse of the polynomial hierarchy. The statement synthesises [BHZ87, GMW91, SCH88] with Theorems 2–6; it establishes no new collapse, which is precisely its content.

Corollary 1 (Classification of the GI level)**.** *The following four cases are exhaustive: every possible location of* GI *within* NP *falls under one of them. They are pairwise mutually exclusive except in worlds where the polynomial hierarchy already collapses, and only the last affects it.*

(1) GI $\in$ **P.** The entire graph-isomorphism level descends into P: CFI, CFNI $\in$ P, and the GI-hardness of DT-FI and CNF-FI becomes vacuous, though both remain coNP-hard and so stay outside P unless $P =$ NP. No collapse; consistent with P $\neq$ NP and PH infinite.

(2) GI $\in$ (NP$\cap$coNP)$\setminus$ **P.** This is the non-trivial low-for-NP case: GI is low for NP but GI $\notin$ P. CFI and CFNI rest in NP$\cap$coNP and the second-level problems keep both lower bounds, the GI core no longer NP-complete. No collapse; consistent with the derandomisation hypothesis AM = NP, which forces GI $\in$ NP $\cap$ coNP and hence case (1) or (2).

(3) GI $\in$ (NP $\setminus$ coNP) **and** GI **not** NP**-complete.** Graph non-isomorphism has no short certificates, yet GI is not complete for NP; CFI and CFNI then lie strictly between P and the NP-complete sets. This case is consistent with all present knowledge — GI $\in$ coAM does not place GI in coNP — and is precisely the case left open below the NP-completeness of case (4) and by Theorem 8. No collapse.

(4) GI **is** NP**-complete.** Then GNI is coNP-complete and in AM [GMW91], so coNP $\subseteq$ AM, and PH collapses to its second level [BHZ87]: the least k with $\Sigma_k^P = \Sigma_{k+1}^P$ is at most 2. (Cases (1) and (4) can hold together only if $P =$ NP, and (2) and (4) only if NP = coNP; under the hypothesis that PH is infinite the four cases form a genuine partition.)

Observation (on the smallest collapse index). The graph-isomorphism level can trigger a collapse only in case (4), and then only to k $\leq$ 2; locations (1)–(3) imply no collapse, since none of GI $\in P$, GI $\in$ NP $\cap$ coNP, or GI $\in$ NP $\setminus$ coNP without completeness yields any consequence for PH. DT-FI and CNF-FI are not Σ_2^P-complete unless PH collapses (Theorem 7), so neither can collapse PH on its own; hardness alone, without completeness, provides no such mechanism.

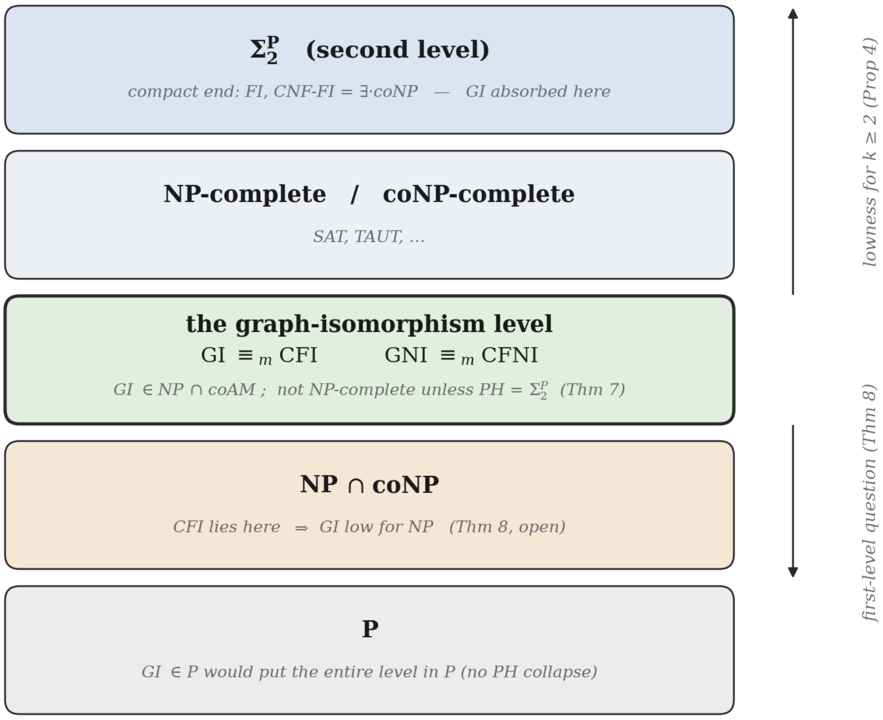


Figure 4. The graph-isomorphism level in the polynomial hierarchy. CFI and CFNI are many-one equivalent to GI and GNI and lie at the level of GI: below Σ_2^P, in which the compact-input problems FI, DT-FI and CNF-FI lie and in which a graph-isomorphism oracle adds no power, $\Sigma_2^P[\mathrm{GI}] = \Sigma_2^P$ (Proposition 4, k $\geq$ 2); and above NP $\cap$ coNP, into which the level descends precisely if GI is low for NP (Theorem 8; open). The level is not NP- or coNP-complete unless PH collapses (Theorem 7). If GI $\in$ P, the entire level falls into P with no collapse.

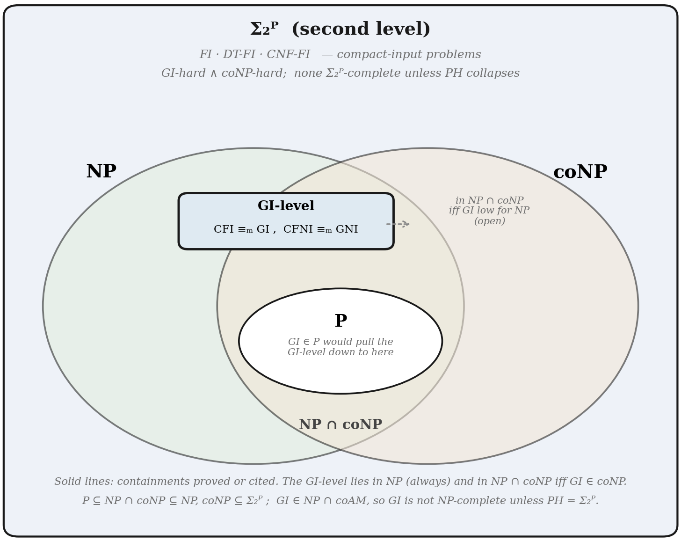


Figure 5. A set-theoretic companion to Figure 4. The classes are nested as proved or cited: P $\subseteq$ NP $\cap$ coNP, with NP and coNP overlapping inside Σ_2^P. The GI-level (CFI $\equiv_m$ GI, CFNI $\equiv_m$ GNI) lies in NP, and whether it reaches the NP $\cap$ coNP lens is the open question of whether GI is low for NP

(Theorem 8); the compact-input problems FI, DT-FI and CNF-FI populate Σ_2^P above the level. Since GI $\in$ NP $\cap$ coAM, the level cannot be NP-complete unless PH collapses (Theorem 7), and GI $\in$ P would draw it down into P with no collapse. Conjectural separations are left undrawn.

6 Discussion

6.1 The localisation principle

The factorisation of Theorem 1 splits formula isomorphism into two parts: the existence of a renaming and the test of equivalence. The first part is always graph isomorphism: Theorem 2 shows this exactly when equivalence is decidable in polynomial time, and Theorem 6 and Proposition 3 show that a graph-isomorphism lower bound survives in every representation. The second part, the cost of testing equivalence, is the only thing that changes from one representation to another. So, every formula-isomorphism problem combines the same graph-isomorphism core with a representation-dependent equivalence cost, and the three problems studied here are simply three settings of that cost. This makes Theorem 1 a **classification tool**: it localises the semantic component, the equivalence cost that caps the level, while Theorem 2 localises the residual structural component, the renaming search, which on canonical inputs is graph isomorphism, so the placement of each problem follows from these two readings rather than a separate analysis of each.

6.2 DT-FI is isomorphism testing, not optimisation

The Minimum DNF problem (MIN-DNF) is the following optimisation question: given a Boolean formula F and an integer k, decide whether there exists a DNF formula with at most k terms that is equivalent to F. Umans showed MIN-DNF is Σ_2^P-complete [UMA98], the hardness being intrinsic to the optimisation over all sufficiently short DNFs. DT-FI is a different problem despite the similar name: by Theorem 4 its DNF target fixes the candidate up to renaming, so there is no minimisation over sizes and no relaxation of MIN-DNF. Its complexity is that of isomorphism testing with a compact argument, not of optimisation, and nothing here lowers the complexity of MIN-DNF.

6.3 Canonical versus Compact Inputs: Connection to Agrawal–Thierauf

Agrawal and Thierauf studied FI on general formulas and showed FI $\in \Sigma_2^P$, not Σ_2^P-complete unless PH collapses [AGTH96, AGTH00]. In the language of Theorem 1 this is the case EQ = coNP-complete, FI = $\exists$·coNP = Σ_2^P; CFI and CFNI are the same isomorphism core, studied here on canonical inputs, with the coNP equivalence not needed, and DT-FI and CNF-FI are intermediate and fully compact instances of the same family. Their non-completeness of FI for Σ_2^P is, one level up, the analogue of the present Theorem 7 for CFI at NP; both are interactive-proof results. In particular, the canonical restriction sharpens the collapse consequence from the third level to the second: CFI $\equiv_m$ GI lies in NP with GNI $\in$ AM, so its NP-completeness yields coNP $\subseteq$ AM and hence only PH = Σ_2^P (Theorem 7), whereas the coNP-complete equivalence test in general FI bars this route and leaves their third-level bound in force. This is a sharper collapse for the canonical problem, not a strengthening of the FI result itself. Proposition 4 explains why GI plays no visible role in their analysis: at Σ_2^P the GI core is absorbed, leaving the coNP burden as the sole difficulty.

Relation to prior work. Two of the components used here are already present in the literature and are not claimed as new. Borchert, Ranjan, and Stephan [BRS98] record Chang's reduction GI $\leq_m^p$ BOOLE-ISO and prove (their Corollary 13) that BOOLE-ISO and its relatives are not in coNP unless GI $\in$ coNP; these are, respectively, the GI-hardness gadget of Theorem 2 and the open question of Theorem 8, recovered and relocated within the factorisation framework.

Goldsmith, Hagen, and Mundhenk [GHM08] show BoolIso_mon $\equiv_m^p$ BoolIso, so the reducibility of formula isomorphism to its monotone case is likewise established; the present paper does not rest any claim on the monotone reductions being novel. What remains specific to this paper is the canonical-input completeness (CFI $\equiv_m$ GI in both directions and CFNI as a GNI-complete problem stated without reference to graphs) together with the factorisation FI $= \exists\lambda \cdot$ EQ that organises the whole spectrum by the cost of the underlying equivalence test.

Boolean constraint isomorphism. Böhler, Hemaspaandra, Reith, and Vollmer [BHRV04] give a complete classification of the isomorphism problem for Boolean constraints: for a finite constraint set C, ISO(C) is coNP-hard and GI-hard when C is not Schaefer, polynomial-time many-one equivalent to GI when C is Schaefer but not 2-affine, and in P when C is 2-affine. Read through Theorem 1, this trichotomy is the factorisation principle instantiated on the constraint-language axis. The Schaefer boundary is the equivalence boundary: constraint equivalence is polynomial-time decidable for Schaefer C and coNP-complete otherwise, so by FI $= \exists\lambda \cdot$ EQ the isomorphism problem inherits $\exists \cdot P$, hence the GI level, when C is Schaefer, and $\exists \cdot$ coNP $= \Sigma_2^P$, with coNP-hardness propagating from equivalence, when it is not. Their further split inside the Schaefer case (2-affine falling into P while the remaining Schaefer classes are GI-complete) is a distinction at the renaming layer, below the resolution of Theorem 1: it records whether the canonical form is rigid enough to make the residual permutation test trivial or as hard as graph isomorphism. In particular, fixing the equivalence cost does not fix the isomorphism complexity: the Schaefer classes all share polynomial-time equivalence, yet their isomorphism problems split between graph-isomorphism-complete and polynomial-time, a split invisible to the equivalence cost and so located entirely in the renaming layer. The present paper varies a different parameter. Where [BHRV04] fix arbitrary functions and vary the logical expressiveness of the constraint language, here the function is arbitrary and what varies is the representation in which it is presented (canonical, DNF, CNF) so that the same equivalence-cost law is exhibited along the axis of representational succinctness rather than constraint type. The two classifications are the same phenomenon seen along orthogonal axes; the constraint classification of [BHRV04], obtained independently, is the constraint-language instance of the factorisation that organises this paper.

6.4 Consequences for the polynomial hierarchy

- GI $\in$ NP $\cap$ coAM, and GNI $\in$ AM [GMW91]; coNP $\subseteq$ AM $\Longrightarrow$ PH $= \Sigma_2^P$ [BHZ87].
- CFI $\equiv_m$ GI and CFNI $\equiv_m$ GNI (Theorems 2–3); DT-FI and CNF-FI lie in Σ_2^P, coNP-hard and GI-hard (Theorems 5–6, Proposition 3).
- None of the four problems is complete for its level unless PH collapses (Theorem 7).
- $\Sigma_k^P[\mathrm{GI}] = \Sigma_k^P$ for k $\geq$ 2 [SCH88] (Proposition 4); the case $k = 1$, i.e. $\mathrm{NP}^{\mathrm{GI}} = \mathrm{NP}$, is open and equivalent to GI $\in$ coNP (Theorem 8).
- The GI level can collapse PH only when GI is NP-complete, and then only to k $\leq$ 2 (Corollary 1).
- $\Sigma_k^P \neq \Sigma_{k+1}^P \Longrightarrow P \neq$ NP, and P $\neq$ NP $\Longrightarrow P \neq$ PH [BOCRE06].

All statements are conditional and consistent with PH being infinite. The contribution is a classification: the polynomial-hierarchy level of any formula-isomorphism problem is governed by the cost of testing equivalence in its representation, graph isomorphism is the component common to every case, and three concrete problems (at the GI level, the mixed second level, and the fully compact second level) exhibit this dependence together with the upper and lower bounds on the graph-isomorphism level.

References

[ACL12] G. Ausiello, F. Cristiano, and L. Laura. Syntactic isomorphism of CNF Boolean formulas is graph isomorphism complete. ECCC, 19:122, 2012.

[AGTH96] M. Agrawal and T. Thierauf. The Boolean isomorphism problem. FOCS 1996, p. 422.

[AGTH00] M. Agrawal and T. Thierauf. The formula isomorphism problem. SIAM Journal on Computing, 30(3):990–1009, 2000.

[ARBA09] S. Arora and B. Barak. Computational Complexity: A Modern Approach. Cambridge University Press, 2009.

[BAB16] L. Babai. Graph isomorphism in quasipolynomial time. STOC 2016, pp. 684–697.

[BCW80] M. Blum, A. K. Chandra, and M. N. Wegman. Equivalence of free Boolean graphs can be decided probabilistically in polynomial time. Information Processing Letters, 10(2):80–82, 1980.

[BHRV04] E. Böhler, E. Hemaspaandra, S. Reith, and H. Vollmer. The complexity of Boolean constraint isomorphism. STACS 2004, LNCS 2996, pp. 164–175, Springer, 2004.

[BHZ87] R. Boppana, J. Håstad, and S. Zachos. Does co-NP have short interactive proofs? Information Processing Letters, 25(2):127–132, 1987.

[BOCRE06] D. P. Bovet and P. Crescenzi. Introduction to the Theory of Complexity. Electronic Edition, 2006.

[BRS98] B. Borchert, D. Ranjan, and F. Stephan. On the computational complexity of some classical equivalence relations on Boolean functions. Theory of Computing Systems, 31(6):679–693, 1998.

[CH11] Y. Crama and P. L. Hammer. Boolean Functions: Theory, Algorithms, and Applications. Cambridge University Press, 2011.

[COOK71] S. Cook. The complexity of theorem-proving procedures. STOC 1971, pp. 151–158.

[GHM08] J. Goldsmith, M. Hagen, and M. Mundhenk. Complexity of DNF and isomorphism of monotone formulas. Information and Computation, 206(6):760–775, 2008.

[GMW91] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity. Journal of the ACM, 38(3):690–728, 1991.

[HURY04] M. Huth and M. Ryan. Logic in Computer Science. 2nd ed., Cambridge University Press, 2004.

[KST93] J. Köbler, U. Schöning, and J. Torán. The Graph Isomorphism Problem: Its Structural Complexity. Progress in Theoretical Computer Science. Birkhäuser, Boston, 1993.

[LUKS99] E. M. Luks. Hypergraph isomorphism and structural equivalence of Boolean functions. In Proceedings of the 31st Annual ACM Symposium on Theory of Computing (STOC), pp. 652–658, 1999.

[RO05] J. Rothe. Complexity Theory and Cryptology. Springer, 2005.

[SCH88] U. Schöning. Graph isomorphism is in the low hierarchy. JCSS, 37(3):312–323, 1988.

[TH00] T. Thierauf. The Computational Complexity of Equivalence and Isomorphism Problems. LNCS 1852, Springer, 2000.

[UMA98] C. Umans. The minimum equivalent DNF problem and shortest implicants. FOCS 1998, pp. 556–563.

[WAG86] K. Wagner. The complexity of combinatorial problems with succinct input representation. Acta Informatica, 23:131–147, 1986.

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il