Symmetric Conjunctions of NP-Complete Relations: A Complexity Collapse Phenomenon and Its Limits

Domenico Cantone[1], Stefano Savino[1]

[1]Dept. of Mathematics and Computer Science, University of Catania, Italy

Abstract

We study symmetric conjunctions of NP-complete relations under type-preserving permutations of their input arguments. Given a language L over a typed instance scheme and a type-preserving permutation π, we consider

$$L^{\pi} = L \cap \pi^{-1}(L),$$

whose instances must satisfy both L and its permuted copy. We ask whether imposing such a symmetry constraint can lower the complexity of an NP-complete problem.

We focus on binary relations with two inputs of the same type, where π swaps the two arguments. For proper subgraph and proper induced subgraph embeddings, the symmetry requirement turns out to be inconsistent: the underlying problems are NP-complete, yet their symmetric conjunctions are empty. For several non-strict relations, the collapses are non-degenerate: induced subgraph isomorphism and ordinary subgraph isomorphism collapse to graph isomorphism, while Hamming Subcode Equivalence, viewed as a subcode-inclusion problem, collapses to permutation code equivalence. Over every fixed finite field, the latter target admits a perfect zero-knowledge proof and hence belongs to Statistical Zero Knowledge (SZK).

The phenomenon is not universal, however. For graph homomorphism, the symmetric conjunction is homomorphic equivalence, equivalently isomorphism of graph cores, and it remains NP-complete. We use these examples to compare structural mechanisms behind distinct complexity behaviors: symmetrizing a preorder yields its induced equivalence relation, symmetrizing the strict part of a preorder yields the empty language, and an anchored-hardness criterion gives a sufficient condition under which an induced equivalence relation remains NP-hard. Thus type-preserving symmetric conjunctions can collapse NP-complete relations to simpler targets, but they need not do so in general. The resulting picture is a comparative taxonomy rather than a complete classification.

Keywords

NP-completeness, complexity collapse, symmetric conjunctions, preorders, subgraph isomorphism, graph isomorphism, graph homomorphism, permutation code equivalence, zero-knowledge proofs

1. Introduction

Many natural computational relations are directional. They express that one object embeds into, is contained in, simulates, or maps to another. In such settings, requiring the relation in both directions is a natural reciprocity condition:

$$R(x, y) \wedge R(y, x).$$

This condition asks whether x and y are mutually realizable with respect to R. Its complexity may differ substantially from that of the original directional relation.

We study the complexity of this reciprocal feasibility condition when R is NP-complete. More generally, for a language L over a typed input scheme and a type-preserving permutation π of its arguments, we consider

$$L^{\pi} = L \cap \pi^{-1}(L).$$

For the transposition of two arguments, this is precisely the language defined by $R(x, y) \wedge R(y, x)$. The general notation accommodates arbitrary typed input schemes and type-preserving permutations, but all results in this paper concern binary relations with same-type arguments and the transposition of those arguments. More general permutations are discussed only as a direction for future work.

ICTCS 2026: Italian Conference on Theoretical Computer Science, September 7–9, 2026, Udine, Italy
domenico.cantone@unict.it (D. Cantone); stefano.savino@phd.unict.it (S. Savino)
0000-0002-1306-1166 (D. Cantone); 0009-0004-6023-9045 (S. Savino)

The question is motivated in part by the classical theory of NP-completeness [7, 11] and by the long-standing search for natural problems of intermediate complexity. Assuming $\mathsf{P} \neq \mathsf{NP}$, NP-intermediate problems exist by Ladner's theorem [12], but natural examples remain elusive. We therefore ask whether reciprocal constraints extracted from NP-complete directional relations can collapse to targets of substantially different and, in some cases, apparently lower complexity.

Our examples show that no uniform answer is possible: reciprocal constraints can either simplify the original relation dramatically or preserve its computational hardness.

Summary of results. Table 1 summarizes the examples studied in this paper. They exhibit four distinct behaviors of symmetric conjunction: inconsistency for strict relations, non-empty polynomial-time tractability, collapse to natural equivalence problems, and persistence of NP-completeness. Later sections identify the structural mechanisms behind these outcomes.

On the nature of the contribution. This paper takes a structural and comparative approach to reciprocal constraints on typed relations. Its goal is not to introduce new equivalence relations—those arising in the examples are classical—but to explain *systematically* why the same syntactic operation

$$R(x, y) \wedge R(y, x)$$

produces qualitatively different complexity outcomes.

While several individual equivalences used in the examples (e.g., mutual embeddability implying isomorphism over finite structures) are standard or implicit in the literature, the contribution of this paper is threefold:

1. we introduce *type-preserving symmetric conjunctions* as a uniform transformation on NP-complete relations;
2. we isolate three preorder-based mechanisms—strict profile growth, monotone-profile rigidity, and anchored hardness—and distinguish them from the global input restriction responsible for the polynomial-time example;
3. we provide a compact taxonomy showing that four qualitatively distinct outcomes (empty, polynomial-time, equivalence-type, NP-complete) can arise from the same transformation.

Thus the novelty lies not in the individual collapse results themselves, but in their unification under a common structural perspective, made explicit in Section 9.

Problem L	Type-preserving π	Characterization of L^π	Target status
PSubgraphIso	swap $\langle G, H\rangle$	$\emptyset$	P
PIndSubgraphIso	swap $\langle G, H\rangle$	$\emptyset$	P
IndSubgraphIso	swap $\langle G, H\rangle$	GraphIso	equivalent to GraphIso
SubcodeEq	swap $\langle C, D\rangle$	PermCodeEquiv	in PZK $\subseteq$ SZK for fixed $\mathbb{F}_q$
SubgraphIso	swap $\langle G, H\rangle$	GraphIso	equivalent to GraphIso
CliqueSubgraphIso	swap $\langle G, H\rangle$	pairs of complete graphs of equal order	P
GraphHom	swap $\langle G, H\rangle$	homomorphic equivalence	NP-complete

Table 1
Summary of type-preserving symmetric-conjunction results.

The non-empty equivalence targets GraphIso and PermCodeEquiv are not known to be in P or NP-complete. In addition, for every fixed finite field, PermCodeEquiv belongs to PZK $\subseteq$ SZK.

Related work. Each of the relations studied below is well established in its own area. GraphIso is a classical candidate for NP-intermediate behavior [8, 15, 3]. Graph homomorphisms and graph cores provide canonical preorder and induced-equivalence frameworks [10]. Permutation code equivalence is a classical equivalence relation in coding theory, with known connections to graph isomorphism [14], and is a special case of the Subspace Transporter problem studied by Arvind and Das [1], whose results imply PermCodeEquiv $\in$ SZK for every fixed finite field.

Here, however, we study all these problems through a single uniform transformation: symmetric conjunction under type-preserving permutations. To our knowledge, the operation $L^\pi = L \cap \pi^{-1}(L)$ has not been systematically investigated as a uniform complexity-theoretic transformation on NP-complete relations.

The contribution is not the introduction of new equivalence problems, but the identification of structural conditions under which symmetrization yields inconsistency, tractability, equivalence, or persistent NP-hardness.

Organization. Sections 2–8 present the main examples. Section 9 reorganizes them by their resulting complexity behavior and isolates the structural mechanisms behind collapse and non-collapse. Section 10 collects open problems.

2. Preliminaries

We assume familiarity with standard complexity theory [2, 13]. All graphs are finite, simple, and undirected unless otherwise stated. We write $G \cong H$ for graph isomorphism, and $V(G)$ and $E(G)$ for the vertex and edge sets of a graph G.

Definition 2.1 (Typed instances and symmetric conjunctions)**.** *A typed instance scheme is a tuple $\tau = (\tau_1, \dots, \tau_k)$ of sorts. A language over τ is a set $L \subseteq D_{\tau_1} \times \cdots \times D_{\tau_k}$. A permutation π of the input positions is* type-preserving *if $\tau_i = \tau_{\pi(i)}$ for every i. For such a permutation, the π-symmetric conjunction of L is*

$$L^\pi = \{x \colon x \in L \ \text{ and } \ \pi(x) \in L\} = L \cap \pi^{-1}(L),$$

where $\pi(\langle x_1, \dots, x_k\rangle) = \langle x_{\pi(1)}, \dots, x_{\pi(k)}\rangle$.

Remark 2.2. If $L \in \mathsf{NP}$ and π is type-preserving, then $L^\pi \in \mathsf{NP}$.

Lemma 2.3 (Strict profile growth)**.** *Let X be a class of finite objects, let R be a binary relation on X, and let $(P, \leq_P)$ be a partially ordered set. Suppose that there exists a map $\mu \colon X \to P$ such that*

$$R(x, y) \quad \Longrightarrow \quad \mu(x) <_P \mu(y),$$

where $<_P$ is the strict order induced by $\leq_P$. Setting $L_R = \{\langle x, y\rangle \colon R(x, y)\}$ and letting $\pi(\langle x, y\rangle) = \langle y, x\rangle$, we have $L_R^\pi = \emptyset$.

Proof. If $\langle x, y\rangle \in L_R^\pi$, then both $R(x, y)$ and $R(y, x)$ hold, whence $\mu(x) <_P \mu(y)$ and $\mu(y) <_P \mu(x)$, which is impossible. □

3. Proper subgraph isomorphisms

Definition 3.1 (ProperSubgraphIsomorphism (PSubgraphIso))**.** *Let G and H be finite simple graphs. We write $G <_{\text{sub}} H$ if G is isomorphic to a proper subgraph of H. Equivalently, there exists an injective map $f \colon V(G) \to V(H)$ such that*

$$\{u, v\} \in E(G) \quad \Longrightarrow \quad \{f(u), f(v)\} \in E(H)$$

for all $u, v \in V(G)$*, and the image of* G *under* f *is proper in the following sense: either*

$$f(V(G)) \subsetneq V(H),$$

or

$$f(V(G)) = V(H) \quad \textit{and} \quad \big\{\{f(u), f(v)\} \colon \{u, v\} \in E(G)\big\} \subsetneq E(H).$$

Define

$$\textsc{PSubgraphIso} = \big\{\langle G, H\rangle \colon G <_{\text{sub}} H\big\}.$$

Proposition 3.2. PSubgraphIso *is* NP*-complete.*

Proof. Membership is immediate. The hardness proof is a standard reduction from Clique. □

Theorem 3.3 (Strict subgraph collapse)**.** *Let* π *be the type-preserving involution* $\pi(\langle G, H\rangle) = \langle H, G\rangle$*. Then* $\textsc{PSubgraphIso}^{\pi} = \emptyset$*. In particular,* $\textsc{PSubgraphIso}^{\pi} \in \mathsf{P}$.

Proof. We apply Lemma 2.3 with $\mu(G) = (|V(G)|, |E(G)|)$. It remains to verify that $G <_{\text{sub}} H$ implies $\mu(G) <_{\text{lex}} \mu(H)$.

By Definition 3.1, let f be an injective map witnessing $G <_{\text{sub}} H$. If $f(V(G)) \subsetneq V(H)$, then $|V(G)| < |V(H)|$. Otherwise, $f(V(G)) = V(H)$, so f is bijective on vertices. By properness, the edge image $\big\{\{f(u), f(v)\} \colon \{u, v\} \in E(G)\big\}$ is a proper subset of $E(H)$. Since f is injective, this edge image has cardinality $|E(G)|$. Hence $|E(G)| < |E(H)|$. Thus, in either case,

$$\mu(G) = (|V(G)|, |E(G)|) <_{\text{lex}} (|V(H)|, |E(H)|) = \mu(H).$$

Lemma 2.3 then gives $\textsc{PSubgraphIso}^{\pi} = \emptyset$, so $\textsc{PSubgraphIso}^{\pi} \in \mathsf{P}$. □

The induced case.

Definition 3.4 (ProperInducedSubgraphIsomorphism (PIndSubgraphIso))**.** *For finite simple graphs* G *and* H*, write* $G <_{\text{ind}} H$ *if* G *is isomorphic to a proper induced subgraph of* H*. Define*

$$\textsc{PIndSubgraphIso} = \big\{\langle G, H\rangle \colon G <_{\text{ind}} H\big\}.$$

The problem PIndSubgraphIso is NP-complete. Membership in NP is immediate, and NP-hardness follows from the standard reduction from Clique.

Corollary 3.5 (Strict induced-subgraph collapse)**.** *Let* $\pi(\langle G, H\rangle) = \langle H, G\rangle$*. Then*

$$\textsc{PIndSubgraphIso}^{\pi} = \emptyset.$$

In particular, $\textsc{PIndSubgraphIso}^{\pi} \in \mathsf{P}$.

Proof. If $G <_{\text{ind}} H$, then $|V(G)| < |V(H)|$. Lemma 2.3, applied to the profile $\mu(G) = |V(G)|$, yields the result. □

4. Induced subgraph isomorphism and graph isomorphism

The strict examples above yield empty symmetric conjunctions. Dropping strictness gives a non-degenerate collapse: the symmetric conjunction of induced-subgraph embeddability is exactly graph isomorphism.

Definition 4.1 (InducedSubgraphIsomorphism (IndSubgraphIso))**.** *Let* G *and* H *be finite simple graphs. We write* $G \leq_{\text{ind}} H$ *if* G *is isomorphic to an induced subgraph of* H*, i.e., there exists an injective map* $f \colon V(G) \to V(H)$ *such that, for all* $u, v \in V(G)$*,*

$$\{u, v\} \in E(G) \iff \{f(u), f(v)\} \in E(H).$$

Define

$$\textsc{IndSubgraphIso} = \big\{\langle G, H\rangle \colon G \leq_{\text{ind}} H\big\}.$$

Proposition 4.2. $\textsc{IndSubgraphIso}$ *is* NP*-complete.*

Proof. Membership in NP is immediate. For NP-hardness, reduce from $\textsc{Clique}$: given $\langle G, k\rangle$, let K_k be the complete graph on k vertices. Then

$$\langle G, k\rangle \in \textsc{Clique} \quad \Longleftrightarrow \quad \langle \mathsf{K}_k, G\rangle \in \textsc{IndSubgraphIso},$$

since an induced copy of K_k in G is exactly a k-clique. The construction is polynomial-time, so $\textsc{IndSubgraphIso}$ is NP-complete. □

Theorem 4.3 (Induced-subgraph collapse)**.** *Let* $\pi(\langle G, H\rangle) = \langle H, G\rangle$*. Then*

$$\langle G, H\rangle \in \textsc{IndSubgraphIso}^{\pi} \quad \Longleftrightarrow \quad G \cong H,$$

and therefore $\textsc{IndSubgraphIso}^{\pi} = \textsc{GraphIso}$.

Proof. If $G \cong H$, then each graph is an induced subgraph of the other. Conversely, suppose that $G \leq_{\mathrm{ind}} H$ and $H \leq_{\mathrm{ind}} G$. The corresponding injections between the finite vertex sets imply

$$|V(G)| = |V(H)|.$$

Hence an induced embedding of G into H is bijective on vertices and therefore is a graph isomorphism. □

The proof can be viewed as a finite Cantor–Schröder–Bernstein argument for induced graph embeddings.

5. Subcode equivalence and permutation code equivalence

We present a second non-degenerate example, now in the setting of linear codes. The collapse is structurally analogous to the one in Section 4, but the target equivalence problem is permutation code equivalence rather than graph isomorphism. Placing this example before ordinary subgraph isomorphism makes clear that the collapse mechanism is not specific to graph relations.

Fix a finite field $\mathbb{F}_q$. A linear code of length n over $\mathbb{F}_q$ is a linear subspace of the vector space $\mathbb{F}_q^n$. Thus, when we write $C \leq \mathbb{F}_q^n$, we mean that C is a linear subspace of $\mathbb{F}_q^n$. Throughout this section, the symbol $\leq$ denotes subspace inclusion. The dimension of a code is its dimension as a vector space over $\mathbb{F}_q$.

Throughout this section, a linear code is represented by the unique reduced row-echelon generator matrix of its row space over the fixed field $\mathbb{F}_q$. Thus the chosen representation is canonical, and equality, dimension, and subspace inclusion of codes can be checked in polynomial time by standard row-reduction procedures.

A coordinate permutation is a permutation $\sigma \in S_n$, where S_n denotes the symmetric group on $\{1, \ldots, n\}$. It acts on a vector $c = (c_1, \ldots, c_n) \in \mathbb{F}_q^n$ by permuting its coordinates, and it acts on a code C by $\sigma(C) = \{\sigma(c) \colon c \in C\}$.

Definition 5.1 ($\textsc{PermutationCodeEquivalence}$ ($\textsc{PermCodeEquiv}$))**.** *Let* $C, D \leq \mathbb{F}_q^n$ *be linear codes. We write* $C \equiv_{\mathrm{perm}} D$ *if there exists a coordinate permutation* $\sigma \in S_n$ *with* $\sigma(C) = D$*. Define*

$$\textsc{PermCodeEquiv} = \big\{\langle C, D\rangle \colon C \equiv_{\mathrm{perm}} D\big\}.$$

This is the standard code equivalence problem in which equivalence is restricted to coordinate permutations.

Definition 5.2 ($\textsc{SubcodeEquivalence}$ ($\textsc{SubcodeEq}$))**.** *Let* $C, D \leq \mathbb{F}_q^n$ *be linear codes. We write* $C \leq_{\mathrm{subcode}} D$ *if there exists a coordinate permutation* $\sigma \in S_n$ *such that* $\sigma(C) \leq D$*. Define*

$$\textsc{SubcodeEq} = \big\{\langle C, D\rangle \colon C \leq_{\mathrm{subcode}} D\big\}.$$

This is a non-strict version of the standard Hamming Subcode Equivalence problem: for codes of the same length, it allows arbitrary dimensions, whereas the standard NP-complete formulation imposes $\dim C < \dim D$. We allow arbitrary dimensions because this slightly broader setting is the natural one for the symmetric conjunction below. Subcode-equivalence variants also arise in code-based cryptography [5, 4].

Proposition 5.3. $\textsc{SubcodeEq}$ *is* NP*-complete.*

Proof. Membership in NP is immediate: a certificate is a coordinate permutation $\sigma \in S_n$, and checking $\sigma(C) \leq D$ is polynomial-time. Hardness follows from the known NP-completeness of the standard Hamming Subcode Equivalence problem [5], which is the restriction of $\textsc{SubcodeEq}$ to instances satisfying $\dim C < \dim D$. □

The type scheme of $\textsc{SubcodeEq}$ is $(\mathrm{LinearCode}_q, \mathrm{LinearCode}_q)$, with the well-formedness condition that both codes have the same length, so the transposition $\pi(\langle C, D\rangle) = \langle D, C\rangle$ is type-preserving.

Theorem 5.4 (Subcode-equivalence collapse)**.** *Let* $\pi(\langle C, D\rangle) = \langle D, C\rangle$*. Then*

$$\langle C, D\rangle \in \textsc{SubcodeEq}^{\pi} \quad \Longleftrightarrow \quad \langle C, D\rangle \in \textsc{PermCodeEquiv},$$

and therefore $\textsc{SubcodeEq}^{\pi} = \textsc{PermCodeEquiv}$.

Proof. ($\Leftarrow$) If $C \equiv_{\mathrm{perm}} D$, then there exists $\sigma \in S_n$ with $\sigma(C) = D$. In particular $\sigma(C) \leq D$, so $C \leq_{\mathrm{subcode}} D$. Applying the inverse permutation gives $\sigma^{-1}(D) = C$, hence $D \leq_{\mathrm{subcode}} C$. Therefore $\langle C, D\rangle \in \textsc{SubcodeEq}^{\pi}$.

($\Rightarrow$) Suppose $\langle C, D\rangle \in \textsc{SubcodeEq}^{\pi}$. Then there exist $\sigma, \tau \in S_n$ with $\sigma(C) \leq D$ and $\tau(D) \leq C$. Since coordinate permutations preserve dimension, the first inclusion gives $\dim C \leq \dim D$ and the second gives $\dim D \leq \dim C$, so $\dim C = \dim D$. Since $\sigma(C) \leq D$ and $\dim \sigma(C) = \dim C = \dim D$, we conclude $\sigma(C) = D$, i.e., $C \equiv_{\mathrm{perm}} D$. □

Corollary 5.5. *For every fixed finite field* $\mathbb{F}_q$*,*

$$\textsc{SubcodeEq}^{\pi} \in \mathsf{PZK} \subseteq \mathsf{SZK}.$$

Proof. By Theorem 5.4, it is enough to show that $\textsc{PermCodeEquiv} \in \mathsf{PZK}$.

Let $C_0 = C$ and $C_1 = D$. The prover chooses uniformly a bit $b \in \{0, 1\}$ and a coordinate permutation $\rho \in S_n$, and sends the unique reduced row-echelon generator matrix of the code $M = \rho(C_b)$ to the verifier. The verifier replies with a challenge $c \in \{0, 1\}$. The prover then returns a coordinate permutation τ such that $\tau(C_c) = M$. If $c = b$, the prover may take $\tau = \rho$. If $c \neq b$, such a permutation exists because $C_0 \equiv_{\mathrm{perm}} C_1$.

The verifier computes the reduced row-echelon generator matrix of $\tau(C_c)$ and checks that it is equal to the received matrix M. This verification is polynomial-time. If $C \not\equiv_{\mathrm{perm}} D$, the two permutation orbits are disjoint, so a prover can answer at most one of the two challenges. Sequential repetition therefore gives negligible soundness error.

For a yes-instance, a uniformly chosen coordinate permutation induces the uniform distribution on the common coordinate-permutation orbit of C and D. Since the reduced row-echelon representation is canonical, the first message has the same distribution whether it is generated from C or from D. The usual rewinding simulator for the Goldreich–Micali–Wigderson protocol [9] therefore produces an identical transcript distribution. Hence $\textsc{PermCodeEquiv} \in \mathsf{PZK}$, and Theorem 5.4 gives

$$\textsc{SubcodeEq}^{\pi} \in \mathsf{PZK} \subseteq \mathsf{SZK}.$$

□

6. Subgraph isomorphism and graph isomorphism

We return to the embeddability relation underlying ordinary Subgraph Isomorphism. This case is slightly more delicate: a subgraph embedding need only preserve edges, not non-edges. Nevertheless, mutual subgraph embeddability collapses exactly to graph isomorphism.

Definition 6.1 (SubgraphIsomorphism (SubgraphIso) and GraphIsomorphism (GraphIso))**.** *For finite simple graphs G and H, define*

$$\textsc{SubgraphIso} = \big\{\langle G, H\rangle :\, G \textit{ is isomorphic to a subgraph of } H\big\}$$

and

$$\textsc{GraphIso} = \big\{\langle G, H\rangle :\, G \cong H\big\}.$$

The problem SubgraphIso is NP-complete [8]. The exact complexity of GraphIso is a longstanding open problem: GraphIso is in NP, but is not known to lie in P or to be NP-complete.

Since the type scheme of SubgraphIso is $(\mathrm{Graph}, \mathrm{Graph})$, the transposition $\pi(\langle G, H\rangle) = \langle H, G\rangle$ is type-preserving.

Theorem 6.2 (SubgraphIso collapses to GraphIso)**.** *For all finite simple graphs G and H,*

$$\langle G, H\rangle \in \textsc{SubgraphIso}^{\pi} \quad\Longleftrightarrow\quad \langle G, H\rangle \in \textsc{GraphIso},$$

and therefore $\textsc{SubgraphIso}^{\pi} = \textsc{GraphIso}$.

Proof. If $G \cong H$, then each graph is a subgraph of the other. Conversely, suppose that G and H embed as subgraphs of one another. The two vertex injections imply $|V(G)| = |V(H)|$, so both embeddings are bijective on vertices. They therefore yield $|E(G)| \le |E(H)|$ and $|E(H)| \le |E(G)|$. Thus the edge image of either bijective embedding has the same cardinality as its target edge set, and hence equals that edge set. The embedding is consequently a graph isomorphism. □

This is again a finite Cantor–Schröder–Bernstein argument, but the ordinary subgraph case also requires an edge-counting step.

7. Clique subgraph isomorphism: a non-empty collapse

In the next example, the symmetric conjunction is non-empty and polynomial-time decidable. Unlike the preceding examples, however, the collapse does not arise from mutual comparability in a preorder. It is instead forced by a global structural constraint on one input: the second graph is required to be complete.

Definition 7.1 (CliqueSubgraphIso)**.**

$$\textsc{CliqueSubgraphIso} = \big\{\langle G, H\rangle :\, H \textit{ is a complete graph isomorphic to a subgraph of } G\big\}.$$

The language asks whether G contains a clique whose size is specified by the complete input graph H.

Proposition 7.2. CliqueSubgraphIso *is* NP*-complete.*

Proof. Membership in NP is immediate: a certificate is an injective map $f\colon V(H) \to V(G)$ witnessing that H embeds as a subgraph of G, together with a check that H is complete.

For NP-hardness, reduce from Clique. Given $\langle G, k\rangle$, map to $\langle G, \mathsf{K}_k\rangle$. The second component is trivially complete. If G contains a k-clique $\{v_1, \ldots, v_k\}$, the map $i \mapsto v_i$ embeds K_k as a subgraph of G. Conversely, any embedding of K_k as a subgraph of G yields k pairwise adjacent vertices, hence a k-clique in G. □

Theorem 7.3 (Non-empty polynomial collapse). *Let $\pi(\langle G, H\rangle) = \langle H, G\rangle$. Then*

$$\textsc{CliqueSubgraphIso}^{\pi} = \big\{\langle G, H\rangle : G \ \textit{and} \ H \ \textit{are complete graphs with} \ |V(G)| = |V(H)|\big\}.$$

In particular, $\textsc{CliqueSubgraphIso}^{\pi} \in \mathsf{P}$.

Proof. Suppose $\langle G, H\rangle \in \textsc{CliqueSubgraphIso}^{\pi}$. Then both $\langle G, H\rangle \in \textsc{CliqueSubgraphIso}$ and $\langle H, G\rangle \in \textsc{CliqueSubgraphIso}$, so G and H are both complete. Since both graphs are complete, any injection between vertex sets automatically maps edges to edges. The two embedding conditions therefore reduce to the existence of injections $V(H) \hookrightarrow V(G)$ and $V(G) \hookrightarrow V(H)$, giving $|V(G)| = |V(H)|$.

Conversely, if G and H are both complete with $|V(G)| = |V(H)|$, then $G \cong H$, so each is isomorphic to a subgraph of the other.

Membership in P: check completeness of G and H and equality of their orders. □

Remark 7.4. Theorem 7.3 lies outside the preorder mechanism: $\textsc{CliqueSubgraphIso}$ is not a preorder on the class of all finite graphs, since it is not reflexive on non-complete graphs. It shows that polynomial-time non-empty collapses can arise from structural reasons other than symmetrization of a preorder.

8. Graph homomorphism: a natural non-collapse

The preceding examples show several forms of collapse under type-preserving symmetric conjunctions. We now turn to a contrasting case: for graph homomorphism, the symmetric conjunction remains NP-complete. This shows that the collapse phenomenon is not universal, even for natural preorder-like relations.

Definition 8.1 ($\textsc{GraphHomomorphism}$ ($\textsc{GraphHom}$)). *Let G and H be finite simple graphs. A* graph homomorphism *from G to H is a map $h : V(G) \to V(H)$ such that, for all $u, v \in V(G)$,*

$$\{u, v\} \in E(G) \quad \Longrightarrow \quad \{h(u), h(v)\} \in E(H).$$

We write $G \to H$ if there exists a graph homomorphism from G to H, and define

$$\textsc{GraphHom} = \big\{\langle G, H\rangle : G \to H\big\}.$$

The relation $\to$ is reflexive and transitive, hence a preorder on finite graphs. Its induced equivalence relation is homomorphic equivalence: two graphs G and H are homomorphically equivalent if $G \to H$ and $H \to G$.

A finite graph C is a *core* if every homomorphism $C \to C$ is an automorphism. Equivalently, C has no homomorphism onto a proper subgraph of itself. Every finite graph G is homomorphically equivalent to a core, unique up to isomorphism, called the *core of* G. Two finite graphs are homomorphically equivalent if and only if their cores are isomorphic.

The language $\textsc{GraphHom}$ belongs to NP, since a graph homomorphism is a polynomial-size certificate that can be checked in polynomial time. It is NP-hard because $G \to \mathsf{K}_3$ is exactly 3-colorability [8].

Since the type scheme of $\textsc{GraphHom}$ is $(\mathrm{Graph}, \mathrm{Graph})$, the transposition $\pi(\langle G, H\rangle) = \langle H, G\rangle$ is type-preserving. The symmetric conjunction is therefore

$$\textsc{GraphHom}^{\pi} = \big\{\langle G, H\rangle : G \to H \ \text{ and } \ H \to G\big\},$$

the language of homomorphically equivalent pairs of graphs.

Its NP-completeness will follow in Section 9 from the anchored-hardness criterion.

Thus $\textsc{GraphHom}^{\pi}$ is the language of homomorphically equivalent pairs of graphs, equivalently characterized by isomorphism of their cores.

The next section explains this non-collapse through the preorder structure of graph homomorphism.

9. A common pattern: collapses and non-collapses

All the examples above concern relations between objects of the same type, with the relevant permutation being the transposition $\pi(\langle x,y\rangle) = \langle y,x\rangle$. Thus the symmetric conjunction has the form $R(x,y) \wedge R(y,x)$. We now isolate the mechanisms responsible for the different outcomes displayed in Table 1. The novelty lies not in the underlying order-theoretic notions themselves, but in recognizing that they arise uniformly from a single transformation on NP-complete relations, namely symmetric conjunction under type-preserving permutations.

Proposition 9.1 (Preorder symmetrization and strictification). *Let $\preceq$ be a preorder on a class X. Define the equivalence relation induced by $\preceq$ by*

$$x \equiv_{\preceq} y \quad\Longleftrightarrow\quad x \preceq y \ \text{ and } \ y \preceq x,$$

and define the strict part of $\preceq$ by

$$x \prec y \quad\Longleftrightarrow\quad x \preceq y \ \text{ and not } \ y \preceq x.$$

Let $L_{\preceq} = \{\langle x,y\rangle : x \preceq y\}$, $L_{\prec} = \{\langle x,y\rangle : x \prec y\}$, and let $L_{\equiv_{\preceq}} = \{\langle x,y\rangle : x \equiv_{\preceq} y\}$. Finally, let $\pi(\langle x,y\rangle) = \langle y,x\rangle$. Then:

(i) $L_{\preceq}^{\pi} = L_{\equiv_{\preceq}}$;
(ii) $L_{\prec}^{\pi} = \emptyset$;
(iii) $L_{\preceq} = L_{\prec} \,\dot\cup\, L_{\equiv_{\preceq}}$, *where $\dot\cup$ denotes disjoint union, i.e., the two languages have empty intersection.*

Proof. All three parts follow directly from the definitions. □

The strict graph examples instantiate mechanism (ii): PSubgraphIso $= L_{<_{\mathrm{sub}}}$ is the strict part of the preorder SubgraphIso $= L_{\leq_{\mathrm{sub}}}$, so Proposition 9.1(ii) applies directly; Lemma 2.3 provides a concrete sufficient condition via a monotone finite invariant. The relations IndSubgraphIso, SubgraphIso, SubcodeEq, and GraphHom are full preorders and instantiate mechanism (i): their symmetric conjunctions equal their respective induced equivalence relations.

Whether those equivalence relations are tractable or NP-complete is determined by Proposition 9.3 and Theorem 9.7 below.

Definition 9.2 (Antisymmetry up to structural equivalence). *Let $\preceq$ be a preorder on a class X of finite structures, and let $\simeq$ be an equivalence relation on X. We say that $\preceq$ is* antisymmetric up to $\simeq$ *if*

$$x \preceq y \ \text{ and } \ y \preceq x \quad\Longrightarrow\quad x \simeq y.$$

Proposition 9.3 (Antisymmetry criterion). *Let $\preceq$ be a preorder on a class X of finite structures, and let $\simeq$ be an equivalence relation on X. Assume that $\preceq$ is invariant under $\simeq$, that is,*

$$x \simeq x' \ \text{ and } \ y \simeq y' \quad\Longrightarrow\quad \big(x \preceq y \iff x' \preceq y'\big).$$

Let $L_{\preceq} = \{\langle x,y\rangle : x \preceq y\}$ and $E_{\simeq} = \{\langle x,y\rangle : x \simeq y\}$. Let $\pi(\langle x,y\rangle) = \langle y,x\rangle$. Then $L_{\preceq}^{\pi} = E_{\simeq}$ if and only if $\preceq$ is antisymmetric up to $\simeq$. Moreover, if $\preceq$ is not antisymmetric up to $\simeq$, then $L_{\preceq}^{\pi} \supsetneq E_{\simeq}$.

Proof. Assume first that $\preceq$ is antisymmetric up to $\simeq$. If $\langle x,y\rangle \in L_{\preceq}^{\pi}$, then $x \preceq y$ and $y \preceq x$, so $x \simeq y$. Hence $\langle x,y\rangle \in E_{\simeq}$.

Conversely, let $x \simeq y$. By reflexivity, $x \preceq x$ and $y \preceq y$. Since $x \simeq y$, $\simeq$-invariance applied respectively to $x \preceq x$ and $y \preceq y$ yields $x \preceq y$ and $y \preceq x$. Thus $\langle x,y\rangle \in L_{\preceq}^{\pi}$. Therefore $L_{\preceq}^{\pi} = E_{\simeq}$.

The converse implication is immediate. If $L_{\preceq}^{\pi} = E_{\simeq}$ and $x \preceq y$, $y \preceq x$, then $\langle x,y\rangle \in L_{\preceq}^{\pi}$, hence $\langle x,y\rangle \in E_{\simeq}$, so $x \simeq y$.

Finally, if antisymmetry up to $\simeq$ fails, then there exist x,y such that $x \preceq y,\ y \preceq x,\ x \not\simeq y$. Thus $\langle x,y\rangle \in L_{\preceq}^{\pi} \setminus E_{\simeq}$. Since $E_{\simeq} \subseteq L_{\preceq}^{\pi}$ by reflexivity and $\simeq$-invariance, the containment is proper. □

Proposition 9.4 (Monotone-profile rigidity criterion)**.** *Let $\preceq$ be a preorder on a class X of finite structures, let $\simeq$ be an equivalence relation on X, and assume that $\preceq$ is invariant under $\simeq$. Let $(P, \le_P)$ be a partially ordered set and let $\mu\colon X \to P$ satisfy:*

(i) *$x \preceq y$ implies $\mu(x) \le_P \mu(y)$;*
(ii) *$x \preceq y$ and $\mu(x) = \mu(y)$ imply $x \simeq y$.*

Then $L^{\pi}_{\preceq} = E_{\simeq}$.

Proposition 9.4 provides a concrete sufficient condition for the antisymmetry hypothesis of Proposition 9.3. Accordingly, the proof below verifies that hypothesis and then applies Proposition 9.3.

Proof. If $x \preceq y$ and $y \preceq x$, monotonicity gives $\mu(x) \le_P \mu(y)$ and $\mu(y) \le_P \mu(x)$. Hence $\mu(x) = \mu(y)$, and condition (ii) gives $x \simeq y$. Thus $\preceq$ is antisymmetric up to $\simeq$, so the conclusion follows from Proposition 9.3. □

Remark 9.5 (Profiles in the collapse examples). Proposition 9.4 applies to induced-subgraph embedding with profile $|V(G)|$, to ordinary subgraph embedding with the lexicographic profile $(|V(G)|, |E(G)|)$, and to subcode inclusion with profile $\dim C$. In each case, an embedding or inclusion at equal profile is rigid up to the relevant structural equivalence.

Remark 9.6 (Interpretation). Proposition 9.3 gives a structural characterization of when symmetric conjunction collapses a preorder to a specific equivalence relation. Under the invariance assumption, antisymmetry up to $\simeq$ is both necessary and sufficient for obtaining $L^{\pi}_{\preceq} = E_{\simeq}$.

For graphs, take $\simeq$ to be graph isomorphism; for linear codes of fixed length, take $\simeq$ to be equivalence under coordinate permutations. The profiles in Remark 9.5 give a concrete verification of the antisymmetry required above. Hence the symmetric conjunctions of induced-subgraph embedding, ordinary subgraph embedding, and subcode inclusion are respectively GraphIso, GraphIso, and PermCodeEquiv.

The graph-homomorphism preorder is not antisymmetric up to isomorphism, since non-isomorphic graphs may be homomorphically equivalent. The *moreover* clause of Proposition 9.3 therefore applies: GraphHom$^{\pi}$ properly contains GraphIso, so the induced equivalence relation—homomorphic equivalence—is strictly coarser than isomorphism.

Theorem 9.7 (Anchored hardness principle)**.** *Let X be a class of finite objects with polynomial-size representations, and let $\preceq$ be a preorder on X. Define $L_{\preceq} = \{\langle x, y\rangle : x \preceq y\}$, and let $\pi(\langle x, y\rangle) = \langle y, x\rangle$. For a language A, suppose there exist a fixed object $t \in X$, called an* anchor, *and a polynomial-time computable map F from instances of A to objects of X such that, for every instance u,*

(i) $t \preceq F(u)$;
(ii) $F(u) \preceq t$ if and only if $u \in A$.

Then: $u \in A \iff \langle F(u), t\rangle \in L^{\pi}_{\preceq}$.
In particular, if A is NP*-hard, then $L^{\pi}_{\preceq}$ is* NP*-hard. If, in addition, $L_{\preceq} \in$ NP, then $L^{\pi}_{\preceq}$ is* NP*-complete.*

Proof. By definition,

$$\langle F(u), t\rangle \in L^{\pi}_{\preceq} \quad \iff \quad F(u) \preceq t \text{ and } t \preceq F(u).$$

Since $t \preceq F(u)$ holds for every u, assumption (ii) yields

$$\langle F(u), t\rangle \in L^{\pi}_{\preceq} \quad \iff \quad u \in A.$$

Thus $u \mapsto \langle F(u), t\rangle$ is a polynomial-time many-one reduction from A to $L^{\pi}_{\preceq}$, proving NP-hardness whenever A is NP-hard. If also $L_{\preceq} \in$ NP, then $L^{\pi}_{\preceq} \in$ NP, since both directions can be certified simultaneously. Hence $L^{\pi}_{\preceq}$ is NP-complete. □

Remark 9.8 (Scope of anchored hardness). The anchored-hardness construction shows that symmetric conjunction does not reduce complexity when one comparison can be forced uniformly by a fixed anchor, while the opposite comparison retains the full hardness of the original problem. This mechanism explains the non-collapse of GraphHom and extends to fixed-template *constraint satisfaction problems* (CSPs), where one asks whether an input relational structure admits a homomorphism into a fixed finite template.

The following theorem specializes the anchored construction to fixed-template CSPs.

Theorem 9.9 (CSP-generated non-collapse)**.** *Let τ be a finite relational signature of positive arity, and let $\mathbf{T}$ be a finite τ-structure. For finite τ-structures $\mathbf{A}$ and $\mathbf{B}$, write $\mathbf{A} \to \mathbf{B}$ when there is a map from the domain of $\mathbf{A}$ to that of $\mathbf{B}$ preserving every relation of τ coordinatewise. Define*

$$\mathrm{CSP}(\mathbf{T}) = \big\{\mathbf{A} \colon \mathbf{A} \to \mathbf{T}\big\} \quad \textit{and} \quad \mathrm{HomEq}_\tau = \big\{\langle \mathbf{A}, \mathbf{B}\rangle \colon \mathbf{A} \to \mathbf{B} \ \textit{ and } \ \mathbf{B} \to \mathbf{A}\big\}.$$

If $\mathrm{CSP}(\mathbf{T})$ is NP-hard, then HomEq_τ is NP-complete.

Proof. For every finite τ-structure $\mathbf{A}$, let $F_{\mathbf{T}}(\mathbf{A}) = \mathbf{A} \sqcup \mathbf{T}$, where the disjoint union interprets each relation componentwise. The canonical inclusion gives $\mathbf{T} \to F_{\mathbf{T}}(\mathbf{A})$, whereas

$$F_{\mathbf{T}}(\mathbf{A}) \to \mathbf{T} \quad \Longleftrightarrow \quad \mathbf{A} \to \mathbf{T}.$$

Hence

$$\mathbf{A} \in \mathrm{CSP}(\mathbf{T}) \quad \Longleftrightarrow \quad \langle F_{\mathbf{T}}(\mathbf{A}), \mathbf{T}\rangle \in \mathrm{HomEq}_\tau\,.$$

This is a polynomial-time reduction. Membership in NP follows by guessing and checking homomorphisms in both directions. □

Corollary 9.10 (Graph homomorphism remains NP-complete)**.** *The symmetric conjunction* $\textsc{GraphHom}^\pi$ *is* NP*-complete.*

Proof. Apply the construction of Theorem 9.9 to the class of finite simple graphs and to $\mathbf{T} = \mathsf{K}_3$. For every graph G,

$$G \to \mathsf{K}_3 \quad \Longleftrightarrow \quad \langle G \sqcup \mathsf{K}_3, \mathsf{K}_3\rangle \in \textsc{GraphHom}^\pi.$$

Since $G \to \mathsf{K}_3$ is 3-colorability, the result follows. □

Remark 9.11 (Fixed-template scope). Theorem 9.9 makes the graph example a special case of a general fixed-template phenomenon. The CSP dichotomy theorem, proved independently by Bulatov and Zhuk [6, 16], ensures that every finite-template CSP is either in P or NP-complete.

Corollary 9.12 (Four behaviors of type-preserving symmetrization)**.** *There exist* NP*-complete binary relations L and type-preserving involutions π such that L^π is respectively:*

(i) *empty;*
(ii) *non-empty and decidable in polynomial time;*
(iii) *equivalent to a natural candidate for* NP*-intermediate behavior, such as* GraphIso *or* PermCodeEquiv*;*
(iv) NP*-complete.*

Proof. The witnesses are: PSubgraphIso or PIndSubgraphIso for (i); CliqueSubgraphIso for (ii) by Theorem 7.3; IndSubgraphIso, SubgraphIso, or SubcodeEq for (iii); and GraphHom via Corollary 9.10 for (iv). □

Thus the relevant question is not whether symmetric conjunctions always lower complexity, but which structural features of the underlying relation determine the outcome. Cases (i), (iii), and (iv) of Corollary 9.12 arise from preorder mechanisms: strictification forces inconsistency, antisymmetry up to isomorphism yields natural equivalence problems, and anchored constructions preserve NP-hardness.

By contrast, CliqueSubgraphIso is not a preorder on the class of all finite graphs; its polynomial-time collapse stems from the completeness constraint on the inputs. Whether there exists a natural NP-complete preorder $\preceq$ on a class of finite structures such that its induced equivalence relation

$$x \equiv_{\preceq} y \quad \Longleftrightarrow \quad x \preceq y \text{ and } y \preceq x$$

is non-empty and polynomial-time decidable remains open.

10. Open problems

Open Problem 10.1. Can one characterize natural classes of NP-complete binary relations $R(x, y)$, possibly arising from different categories of finite mathematical structures, according to the complexity of $R(x, y) \wedge R(y, x)$? Can the mechanisms isolated in Section 9 be extended to a useful characterization theorem, or to broader sufficient and necessary conditions for the different complexity outcomes?

Open Problem 10.2. What structural properties distinguish the non-collapse of GraphHom from the collapses of IndSubgraphIso, SubgraphIso, and SubcodeEq? More generally, beyond the anchored-hardness criterion of Theorem 9.7, can one identify necessary conditions, or broader sufficient conditions, under which the induced equivalence relation of an NP-complete preorder-like relation remains NP-complete?

Further directions include the study of non-involutive type-preserving permutations. If π has order m, the corresponding orbit-symmetric conjunction is

$$L^{\langle\pi\rangle} = \bigcap_{i=0}^{m-1} \pi^{-i}(L).$$

Thus an instance is retained only when all its images along the π-orbit satisfy the original relation. The transposition studied in this paper is the special case $m = 2$. It would be interesting to determine whether longer permutation orbits yield collapse mechanisms that are not visible in the reciprocal binary setting. Possible analogues of the collapse phenomenon at higher levels of the polynomial hierarchy are another natural direction.

11. Conclusions

We have studied symmetric conjunctions of NP-complete relations under type-preserving permutations of their input arguments. For a transposition, the construction requires $R(x, y) \wedge R(y, x)$.

The examples realize the four behaviors in Table 1: reciprocal constraints may be inconsistent, tractable, equivalence-like, or remain NP-complete. Strict profile growth explains the empty collapses, while monotone-profile rigidity accounts for the collapses to structural equivalence. Anchored hardness explains the graph-homomorphism non-collapse, and Theorem 9.9 extends this to fixed-template CSPs. The CliqueSubgraphIso example falls outside the preorder framework, its collapse arising from a global input restriction.

Thus the paper provides a comparative taxonomy rather than a complete classification. A systematic classification of reciprocal constraints remains open.

Acknowledgments The authors wish to thank Eugenio Omodeo and Alberto Policriti for valuable comments on this work.

The first author was partially supported by PIAno di inCEntivi per la Ricerca di Ateneo 2024/2026 – Linea di Intervento I "Progetti di ricerca collaborativa" – Università di Catania – Progetto "Semantic Web of EveryThing through Ontological Protocols" (SWETOP).

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT, powered by GPT-5.5, to paraphrase and reword text, improve writing style, and check grammar and spelling. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References

[1] V. Arvind and B. Das. SZK proofs for black-box group problems. *Theory of Computing Systems*, 43(2):100–117, 2008.

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

[3] L. Babai. Graph isomorphism in quasipolynomial time [extended abstract]. In *Proc. 48th ACM STOC*, pages 684–697, 2016.

[4] M. Bardet, C. Brion, P. Gaborit, M. Haiech, and R. Neveu. The matrix subcode equivalence problem and its application to signature with MPC-in-the-head. *Designs, Codes and Cryptography*, 94(4), Article 76, 2026.

[5] T. P. Berger, C. T. Gueye, and J. B. Klamti. A NP-complete problem in coding theory with application to code based cryptography. In *C2SI 2017*, pages 230–237, 2017.

[6] A. A. Bulatov. A dichotomy theorem for nonuniform CSPs. In *Proc. 58th IEEE Symposium on Foundations of Computer Science*, pages 319–330, 2017.

[7] S. A. Cook. The complexity of theorem-proving procedures. In *Proc. 3rd ACM STOC*, pages 151–158, 1971.

[8] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of* NP-*Completeness.* W. H. Freeman, 1979.

[9] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, 38(3):690–728, 1991.

[10] P. Hell and J. Nešetřil. *Graphs and Homomorphisms.* Oxford Lecture Series in Mathematics and Its Applications, Vol. 28. Oxford University Press, 2004.

[11] R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[12] R. E. Ladner. On the structure of polynomial time reducibility. *J. ACM*, 22(1):155–171, 1975.

[13] C. H. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.

[14] E. Petrank and R. M. Roth. Is code equivalence easy to decide? *IEEE Transactions on Information Theory*, 43(5):1602–1604, 1997.

[15] U. Schöning. Graph isomorphism is in the low hierarchy. *J. Comput. System Sci.*, 37(3):312–323, 1988.

[16] D. Zhuk. A proof of the CSP dichotomy conjecture. *Journal of the ACM*, 67(5):Article 30, 2020.