Order-Invariant Cluster First-Order Logic on Graph Classes of Bounded Degree

Fatemeh Ghasemi
Univ Paris Est Creteil, LACL, F-94010 Creteil, France

Julien Grange
Univ Paris Est Creteil, LACL, F-94010 Creteil, France

Abstract

We introduce a new logic, called *cluster first-order logic*, a restricted fragment of first-order logic specifically designed to study order invariance. An order-invariant formula is one on a vocabulary that contains an order; however, whether a structure satisfies it or not is independent of the interpretation of the order.

We show that while order-invariant cluster first-order logic can define properties outside the scope of plain first-order logic in general, its expressive power is included in that of first-order logic when it comes to classes of bounded degree.

We establish this result by explicitly constructing linear orders such that similar structures remain similar when they are expanded with these orders. This similarity-preserving, local-to-global approach is technically involved and somewhat counterintuitive, since adding an order usually reveals distinctions that are otherwise hidden due to the locality of first-order logic. We believe that this work can be a stepping stone toward applying such techniques to plain first-order logic and toward settling the question of the expressive power of order-invariant plain first-order logic.

2012 ACM Subject Classification Theory of computation → Finite Model Theory; Theory of computation → Complexity theory and logic

Keywords and phrases Order invariance, cluster first-order logic, model checking

Digital Object Identifier 10.4230/LIPIcs.MFCS.2026.71

Related Version *Full Version*: https://arxiv.org/abs/2604.27693

1 Introduction

Logical languages like *first-order logic* (FO) are the main way to frame queries in finite model theory and database theory. By default, when writing a query in such a language, one only uses the relations appearing in the signature of the structures at hand. In practice however, structures do not exist in the void and are ordered (albeit in an arbitrary fashion) on tape or in memory, which only allow linear, one-dimensional representations of data. From this observation comes the natural question of whether our formal languages can use this additional information, provided the result of formula evaluation does not depend upon the precise order the input structure is endowed with.

To that end, we authorise the use of an extra binary symbol $<$ in the syntax of formulas, which is guaranteed to be interpreted as a linear order on the elements of the structure. However, insofar as we should only define properties of the underlying unordered structures, it is asked that whether such a formula holds in the ordered structure does not depend on the choice of the order. Such a formula is called an order-invariant formula. Order invariance stems from the fields of descriptive complexity, in particular the quest for a logic for PTIME, database theory, where one studies generic queries that should depend only on the underlying unordered database, and finite model theory, where one often allows access to additional relations as technical devices [1, 18]. Order-invariant first-order logic captures exactly this idea: formulas may use an auxiliary linear order, but their truth must be independent of which order is chosen [15].

From the perspective of descriptive complexity, this makes order-invariant logics particularly interesting. Many classical logical characterisations of complexity classes are obtained over ordered structures, where a built-in order provides a canonical way to organise computation [12]. Studying order-invariant fragments is therefore a way to understand whether access to order genuinely increases expressive power, or whether order merely serves as a technical aid. In this sense, order-invariant logics belong to the broader programme of understanding definability on unordered finite structures [15, 18].

A natural way to understand these logics is through their expressive power. Access to a linear order provides a global enumeration of the elements, and therefore it is not surprising that allowing formulas to access an order can increase expressive power. In the realm of finite structures, it has been shown that order-invariant first-order is more expressive than plain first-order logic [15]. Prominent separating examples are that of Rossman [14] (which only uses the successor relation induced by the order – this restriction of order-invariant FO is known as *successor-invariant* FO), and the one due to Gurevich (which can be found in [13, Theorem 5.3]). In all of the known examples proving the existence of order-invariant first-order formulas that have no first-order equivalent, the underlying structure is very complex from the eyes of graph structural theory. Hence, one may ask *whether on tamer structures we can see a collapse of order-invariant* FO *on plain* FO*.* This is the case on some specific classes of trees and on classes of graphs with bounded tree-depth for which first-order logic and its order-invariant extension have the same expressive power. For the former, the proof technique is to exhibit a chain of intermediate structures and intermediate orders that are pairwise similar (by which we informally mean that these structures agree on all the FO-formulas up to some quantifier depth) [3]. The latter is shown by proving that MSO[1] and FO have the same expressive power on classes of graphs of bounded tree-depth, and then proving the inclusion of order-invariant FO in MSO [5]. To get this inclusion, starting from two FO-similar structures, one constructs orders that maintain this similarity. The same method is used in [2] on *first-order logic with two variables* (FO^2) to show that order-invariant FO^2 is included in FO on bounded degree structures. While similar in terms of methodology, the two mentioned works ([5, 2]) construct orders in widely different fashions, which are influenced by the very different restrictions (bounded tree-depth and bounded degree). See Table 1, for a summary of known results on the expressive power of order-invariant FO.

To summarise, the methods showing the collapse of order-invariant FO to plain FO are:

- Showing the inclusion of order-invariant FO in MSO and an equivalence in the expressive power of MSO and FO, on a class $\mathcal{C}$.
- Showing that for any two FO-similar structures in $\mathcal{C}$, one can compute orders such that the expanded structures with said orders remain FO-similar; the proof of why this implies the collapse of order-invariant FO to plain FO in given in Theorem 3.1.

Now, if one wants to extend the current results, bounded-degree graphs seem like a good starting point. We know that on these structures, order-invariant FO is included in MSO and successor-invariant FO collapses to FO [9]. It is therefore natural to ask whether, on such classes, order-invariant FO collapse to FO as well. As MSO is strictly more expressive than FO on graphs of bounded degree[2], the first method will not work.

In this work, we take a step towards settling the question of *whether the expressive power of order-invariant first-order logic and plain first-order logic on graphs of bounded degree is the same.*

[1] Short for *monadic second-order logic*, which is an extension of first-order logic where quantification over sets of vertices is allowed. In general, it is strictly more expressive than FO.

[2] MSO can, for instance, express **connectivity**, see e.g. [13, Chapter 4].

Table 1 Known results on the expressive power of invariant first-order logics.

Setting	**Result**
Arbitrary finite structures	Every order-invariant FO formula is Gaifman-local [11].
Strings; trees	Collapse to FO [3].
Bounded-treewidth graphs; bounded-degree graphs	Contained in MSO [3];
Bounded-degree graphs	Successor-invariant FO collapses to FO [9].
Bounded-treedepth graphs	Collapse to FO [5].
Planar graphs; more generally, decomposable structures	Contained in MSO [8].
Hollow trees	Collapse to FO [10].
Classes of bounded degree structures	Order-invariant FO^2 is contained in FO; moreover, the same bounded-degree containment also holds for the two-variable counting extension considered there [2].

Although our main results concern the expressive power of order-invariant logics, the method used to prove them is also part of the contribution. The central step is to start from two bounded-degree structures that are sufficiently similar from the point of view of plain FO, and to construct linear orders on them so that the resulting ordered structures remain indistinguishable by CFO. In this sense, the proof shows how a global object, namely a linear order, can be added without destroying the relevant local similarity. Understanding when such orders can be constructed, and how they have to be organised, gives structural information about the graphs under consideration. We believe that this order-construction method may also be useful in future attempts to compare order-invariant FO with plain FO on broader classes of structures. Thus, beyond the specific results proved here for CFO, the paper develops a toolkit for studying order-invariant first-order logics.

Computing such good orders for FO on bounded-degree graphs and even trees (where we already know that order-invariant FO and plain FO collapse) seems very hard and out of reach. Therefore, we tackle this problem by introducing a new fragment of first-order logic: *cluster first-order logic* (CFO).

Cluster first-order logic operates on ordered graphs, and variables of this logic are grouped in clusters. Variables of the same cluster can be compared in an unrestricted fashion (both in terms of edges and precedence in the order), while elements from different clusters can (almost) not be compared. To prevent us from grouping all the variables in the same cluster, variables of a cluster are guarded by the edge-relation, meaning that when a new variable from a cluster is quantified, it has to be adjacent to a previously quantified variable of the same cluster. Up to this point, the logic is already as expressive as FO on bounded-diameter graphs, as shown in Remark 2.2. To grant more expressive power to CFO, we allow a single element of each cluster to be compared with the order to every element of one other cluster; the syntax of CFO prevents this dependency to be circular. Order-invariant CFO is the set of such sentences whose truth value do not depend on the order.

We show that order-invariant cluster first-order logic is included in FO *on classes of graphs of bounded degree (Theorem 3.1).* Let us emphasis again that we achieve this by computing good orders on similar bounded degree structures such that they remain similar after expanded by the orders.

We then prove that model-checking[3] of an order-invariant cluster first-order formula on classes of bounded degree can be done in fixed-parameter tractable time. Model-checking results for order-invariant and successor-invariant FO such as those in [7, 6, 17] proceed by computing nice orders or successor relations such that when a graph from those classes is expanded to a structure with those relations, the expanded class remains a class for which a fixed-parameter tractable algorithm exists. This way, model-checking an order- or successor-invariant FO formula is reduced to model-checking a plain FO formula to a class of structures with order or successor relations. This method shall not work for order-invariant FO on bounded degree graphs. Indeed, bounded-degree classes of graphs are incomparable with bounded twin-width classes of graphs, and it has been shown that model-checking of plain FO formulas on ordered structures is fixed-parameter tractable on a class $\mathcal{C}$ if and only if $\mathcal{C}$ has bounded twin-width [4]. Therefore, it is impossible to order every bounded-degree class of graphs so as to get a tame class of structures, in the sense that FO model-checking on this class is fixed-parameter tractable[4]. A concrete example of such a class is cubic graphs.

Accordingly, we had to approach the model-checking of order-invariant CFO in a totally different way. We know that on a class of bounded degree, every sentence of $<$-invariant CFO has an equivalent FO-sentence: an idea could be to translate $<$-invariant CFO to FO, and then to rely on the fixed-paramater tractability of FO on graphs of bounded degree [16]. However, we do not know such a translation to be computable. Instead, we use the structural tools developed for the expressive-power result. More precisely, we show that on bounded-degree graphs, the truth of an $<$-invariant CFO sentence depends only on finitely many local configurations. Since only boundedly many such configurations can occur, model-checking reduces to checking the formula on a finite summary of the ordered graph. We then show that this summary can itself be computed in fixed-parameter tractable time, which yields a fixed-parameter tractable model-checking algorithm on bounded-degree graphs.

As a last result, we adapt Gurevich's example separating order-invariant FO from FO to get an order-invariant CFO property that falls outside the scope of FO.

To summarise:

- $<$-invariant CFO is included in plain FO on bounded-degree classes of graphs (Theorem 3.1), and model-checking is fixed-parameter tractable on such classes (Theorem 4.1), and
- it is strictly more expressive than FO in general (Theorem 5.1).

2 Preliminaries

2.1 Graphs, general notations

In this setting, we consider *loopless, undirected, coloured simple graphs.* Such graphs can be formally represented as relational structures over the vocabulary $\sigma = \{E, P_1, \ldots, P_m\}$, where E is a binary predicate representing edges (whose interpretation is always a symmetric and loopless binary relation) and each P_i is a unary predicate representing a vertex colour. We will also consider ordered graphs, by which we mean structures over vocabulary $\sigma \cup \{<\}$ where $<$ is a binary predicate, interpreted as a linear order on the elements. Graphs and structures

[3] *Model-checking* is about determining whether a given formula φ holds in a given graph G. It is customary to distinguish the size of φ from the size of G as the latter is often much larger than the former. As such, the complexity of model-checking problems is often looked at through the lense of parametrized complexity, and a model-checking algorithm will be considered satisfactory when the dependency on G is polynomial. In case such an algorithm exists, the model-cheking problem is said to be fixed-parameter tractable.

[4] More precisely, there exists a class of bounded degree graphs such that expanded with any order relations, the expanded class will be monadically independent.

are denoted $\mathcal{A}, \mathcal{B}, \ldots$ while their sets of elements are denoted $A, B, \ldots$; if $P \subseteq A$, we write $\mathcal{A}_{|P}$ for the substructure of $\mathcal{A}$ induced by P. An isomorphism between two structures is a bijection between their element set that respects all predicates of σ in both directions.

For standard terminology and further background on the underlying structural properties, we refer to [13] for a more comprehensive treatment of finite model theory.

2.2 First-order logic

We use standard first-order logic FO over relational vocabularies. If τ is such a vocabulary, formulas over τ are built from atomic formulas $R(x_1, \ldots, x_n)$, for $R \in \tau$, and equalities $x = y$, using Boolean connectives and quantification over elements of the structure. When we want to make the vocabulary explicit, we write $\mathsf{FO}(\tau)$ for first-order logic over τ.

Given a formula $\varphi \in \mathsf{FO}(\tau)$ with free variables in X, a τ-structure $\mathcal{A}$, and a valuation $\nu : X \to A$, we write $(\mathcal{A}, \nu) \models \varphi$ if φ holds in $\mathcal{A}$ under ν. If $X = \emptyset$, then φ is a sentence and we simply write $\mathcal{A} \models \varphi$.

The *quantifier rank* of an FO formula is the maximum nesting depth of its quantifiers. We write $\mathsf{FO}[k]$ for the set of FO sentences of quantifier rank at most k.

2.3 Cluster first-order logic

We define *cluster first-order logic* (CFO), a syntactic fragment of FO over ordered, coloured graphs. Variables are organised into *clusters* indexed by words $w \in \Sigma^*$ over a fixed infinite alphabet Σ. Inside each cluster, variables are enumerated $x_w^0, x_w^1, x_w^2, \ldots$. Except for x_w^0, variables may only be introduced by *guarded* quantification, enforcing adjacency to a previously introduced variable of the same cluster. Moreover, E and $=$ may only relate variables from the same cluster. The order relation $<$ may additionally compare a variable from cluster w with the first variable $x_{w\alpha}^0$ of a child cluster $w\alpha$, where $\alpha \in \Sigma$.

Indices and variables. Fix an infinite alphabet Σ. For a finite set of indices $S \subseteq \Sigma^* \times \mathbb{N}$, define $X_S := \{x_w^i \ : \ (w, i) \in S\}$. We say that S is *valid* if for all $w \in \Sigma^*$, $\alpha \in \Sigma$, and $i \in \mathbb{N}$:

- if $(w, i) \in S$, then $(w, j) \in S$ for every $j < i$;
- if $(w\alpha, 0) \in S$, then $(w, 0) \in S$.

Syntax. Let $\sigma := \{E, P_1, \ldots, P_N\}$ where E is binary and each P_ℓ is unary, and consider the vocabulary $\sigma \cup \{<\}$ where $<$ will be interpreted as a linear order. For every valid set S, we define $\mathsf{CFO}_S[k]$ (formulas of quantifier rank at most k) inductively.

For $k = 0$, $\mathsf{CFO}_S[0]$ is the smallest set containing the following atomic formulas and closed under Boolean connectives:

- $P_\ell(x_w^i)$ for $x_w^i \in X_S$,
- $E(x_w^i, x_w^j)$ and $x_w^i \sim x_w^j$ for $x_w^i, x_w^j \in X_S$ and $\sim \in \{=, <\}$,
- $x_{w\alpha}^0 \sim x_w^i$ for $x_{w\alpha}^0, x_w^i \in X_S$ and $\sim \in \{=, <\}$.

For $k + 1$, $\mathsf{CFO}_S[k + 1]$ is the smallest set closed under Boolean connectives that contains all formulas of the following forms (where $\psi \in \mathsf{CFO}_{S'}[k]$):

- Root introduction (only when $S = \emptyset$): $\exists x_\varepsilon^0 \, \psi$ where $S' = \{(\varepsilon, 0)\}$.
- Opening a child cluster: $\exists x_{w\alpha}^0 \, \psi$ where $(w, 0) \in S$, $(w\alpha, 0) \notin S$, $\alpha \in \Sigma$, and $S' = S \cup \{(w\alpha, 0)\}$.
- Guarded continuation inside a cluster: $\exists x_w^i \big(E(x_w^i, x_w^j) \wedge \psi\big)$, where $(w, j) \in S$ and i is the smallest index such that $(w, i) \notin S$, and $S' = S \cup \{(w, i)\}$.

Finally, we let $\mathsf{CFO}_S := \bigcup_{k \in \mathbb{N}} \mathsf{CFO}_S[k]$, omit S when $S = \emptyset$, and write simply CFO.

Semantics. A formula $\psi \in \mathsf{CFO}_S$ is interpreted as a first-order formula with free variables in X_S. For an ordered σ-structure $(\mathcal{A}, <)$ and a valuation $\nu : X_S \to A$, satisfaction is defined as usual and denoted $(\mathcal{A}, <, \nu) \models \psi$.

▶ **Example 2.1.** Formula φ_1 detects a triangle within a specific cluster α ($\alpha \in \Sigma$).

$$\varphi_1 := \exists x^1_\alpha \Big(E(x^1_\alpha, x^0_\alpha) \wedge \exists x^2_\alpha \big(E(x^2_\alpha, x^1_\alpha) \wedge E(x^2_\alpha, x^0_\alpha) \big) \Big).$$

This formula belongs (among other) to $\mathsf{CFO}_{\{(\varepsilon,0),(\alpha,0)\}}$. In fact, φ_1 belongs to CFO_S for any valid set S of indices containing $(\alpha, 0)$ but not $(\alpha, 1)$. The sub-formula $\exists x^2_\alpha(E(x^2_\alpha, x^1_\alpha) \wedge E(x^2_\alpha, x^0_\alpha))$ belongs to $\mathsf{CFO}_{\{(\varepsilon,0),(\alpha,0),(\alpha,1)\}}$.

Formula $\varphi_2 \in \mathsf{CFO}_{\{(\varepsilon,0)\}}$ asserts the existence of a neighbour x^1_ε of x^0_ε such that the interval between x^0_ε and x^1_ε (by which we mean all the elements of the structure that are between x^0_ε and x^1_ε with respect to $<$) contains a vertex belonging to some triangle:

$$\varphi_2 := \exists x^1_\varepsilon \Big(E(x^1_\varepsilon, x^0_\varepsilon) \wedge \exists x^0_\alpha \Big((x^0_\varepsilon < x^0_\alpha \wedge x^0_\alpha < x^1_\varepsilon) \vee (x^1_\varepsilon < x^0_\alpha \wedge x^0_\alpha < x^0_\varepsilon) \Big) \wedge \varphi_1 \Big).$$

Crucially, note that it is not possible in CFO to check whether the two other vertices of the triangle are in the interval between x^0_ε and x^1_ε. In CFO, for two variables in the same cluster w, we can refer to at most one vertex at a time that lies in the interval between them and is not in the same cluster w, by using the first variable of a new cluster $w\alpha$ ($\alpha \in \Sigma$).

As can be expected, the obligation for new variables of a given cluster to be guarded via E to a previous variable does not hinder the expressive power of CFO on graphs of small diameters:

▶ Remark 2.2. Let $\mathcal{C}$ be a class of ordered graphs such that the underlying graph of each $(\mathcal{A}, <) \in \mathcal{C}$ has diameter at most δ.

Then $\mathsf{FO}(\sigma \cup \{<\})$ and CFO have the same expressive power on $\mathcal{C}$. In other words, every subclass of $\mathcal{C}$ definable by a first-order sentence over the ordered vocabulary $\sigma \cup \{<\}$ is definable in CFO, and conversely.

Proof sketch. One direction needs no proof, insofar as CFO is a fragment of $\mathsf{FO}(\sigma \cup \{<\})$. For the other direction, consider a sentence $\varphi \in \mathsf{FO}(\sigma \cup \{<\})$. Without loss of generality, using the equivalence between $\forall x \psi$ and $\neg \exists x \neg \psi$, we can thus assume φ contains only existential quantifiers.

We construct a sentence $\widehat{\varphi} \in \mathsf{CFO}$ equivalent to $\varphi \in \mathsf{FO}(\sigma \cup \{<\})$ as follows, only using cluster ε. First, we prepend φ with the existential quantification $\exists x^0_\varepsilon$. Then, moving from the outside in, and assuming the variables $x^0_\varepsilon, \ldots, x^i_\varepsilon$ have been used so far in the construction of $\widehat{\varphi}$, we replace subformula $\exists x \psi$ via

$$\bigvee_{j=0}^{\delta} \exists x^{i+1}_\varepsilon E(x^{i+1}_\varepsilon, x^0_\varepsilon) \wedge \exists x^{i+2}_\varepsilon E(x^{i+2}_\varepsilon, x^{i+1}_\varepsilon) \wedge \cdots \wedge \exists x^{i+j}_\varepsilon E(x^{i+j}_\varepsilon, x^{i+j-1}_\varepsilon) \wedge \psi\,.$$

All variables introduced in the translation belong to the same cluster, so all atomic relations of the ordered vocabulary appearing in φ, including order atoms, are allowed in the resulting CFO formula. Since in the underlying graphs of structures from $\mathcal{C}$, every possible witness to $\exists x \psi$ is at distance at most δ from the vertex interpreting x^0_ε, φ and $\widehat{\varphi}$ are equivalent on $\mathcal{C}$. ◀

A sentence $\varphi \in \mathsf{CFO}$ is *order-invariant* (or $<$-invariant) if its truth value is independent of the specific order chosen on a graph, i.e. if for any finite graph $\mathcal{A}$, and any two linear orders $<_1$ and $<_2$ on its vertex-set A,

$$(\mathcal{A}, <_1) \models \varphi \iff (\mathcal{A}, <_2) \models \varphi\,.$$

In this case, we write $\mathcal{A} \models \varphi$. We write $<$-invariant CFO for the set of all order-invariant CFO-sentences. This ensures the logic defines properties inherent to the underlying graph $\mathcal{A}$ rather than an arbitrary arrangement of its elements.

Given a logic $\mathcal{L} \in \{\mathsf{FO}, \mathsf{CFO}, <\text{-invariant } \mathsf{CFO}\}$ and a set X of variables, two structures $\mathcal{A}$ and $\mathcal{B}$ (specifically, ordered graphs in the case of CFO and graphs in the case of $<$-invariant CFO), together with valuations $\nu_A : X \to A$ and $\nu_B : X \to B$, are said to be *k-equivalent* (or *k-similar*) with respect to $\mathcal{L}$, denoted $(\mathcal{A}, \nu_A) \equiv^k_{\mathcal{L}} (\mathcal{B}, \nu_B)$, if they satisfy exactly the same formulas of $\mathcal{L}$ with quantifier rank at most k. If $X = \emptyset$, we omit the valuations and write $\mathcal{A} \equiv^k_{\mathcal{L}} \mathcal{B}$.

To reason about indistinguishability in CFO, we use a game characterisation called the *cluster Ehrenfeucht–Fraïssé game* (CEF), which captures exactly the expressive power of CFO. The game is played for k rounds between two players, *Spoiler* and *Duplicator*, on two ordered graphs. In each round, Spoiler selects an element in a CFO-consistent way, either opening a new cluster or introducing a new neighbour inside an existing cluster, and Duplicator must respond in the other graph with a matching choice that preserves adjacency, equality, and the restricted order comparisons allowed in CFO. Duplicator aims to maintain a partial isomorphism respecting the cluster discipline, while Spoiler tries to break this correspondence. Duplicator wins if they can maintain these conditions for all k rounds. The precise connection with CFO-equivalence is the following.

▶ **Theorem 2.3** (CEF captures CFO). *Let $(\mathcal{A}, <_A)$ and $(\mathcal{B}, <_B)$ be two ordered graphs. Then $(\mathcal{A}, <_A) \equiv^k_{\mathsf{CFO}} (\mathcal{B}, <_B)$ if and only if Duplicator has a winning strategy in the k-round cluster EF game between $(\mathcal{A}, <_A)$ and $(\mathcal{B}, <_B)$.*

Game characterisations of logical equivalence are classical in finite model theory, and typically follow a common template. We use such a game as a technical tool in our proofs to establish indistinguishability between ordered graphs. We then move to showing our main result.

3 Main result

Order invariance means that formulas are allowed to use an additional linear order $<$ on the vertices as a technical aid, but the property they define should not depend on which particular order is chosen. In other words, even though the formula may refer to $<$, its truth must be the same for every possible ordering of a given graph.

When comparing two logics on a class of graphs, we say that one is included in the other if every property definable on that class by the first logic can be expressed as well by the second one. Two graphs $\mathcal{A}$ and $\mathcal{B}$ are k-equivalent with respect to a logic $\mathcal{L}$, denoted $\mathcal{A} \equiv^k_{\mathcal{L}} \mathcal{B}$, if they satisfy exactly the same $\mathcal{L}$-formulas of quantifier rank at most k.

We can now formally state the main result of this work.

▶ **Theorem 3.1.** *Let $\mathcal{C}$ be a class of graphs of bounded degree. Order-invariant* CFO *is included in* FO *on $\mathcal{C}$.*

Proof sketch. To prove the inclusion $<$-invariant CFO $\subseteq$ FO on $\mathcal{C}$, we compare how well the two logics can distinguish graphs. For a fixed quantifier rank k, it suffices to find a function $f(k)$ such that whenever two graphs A and B cannot be distinguished by any FO sentence of rank $f(k)$, they also cannot be distinguished by any $<$-invariant CFO sentence of rank k. Intuitively, this means that FO is at least as expressive: if FO sees no difference, then CFO sees none either. Another way of looking at this is to notice that both logics partition $\mathcal{C}$ into finitely many indistinguishability classes, each one definable in that logic, and each $<$-invariant CFO$[k]$ class is a union of FO$[f(k)]$ classes. Since a formula of $<$-invariant CFO$[k]$ simply covers a union of $<$-invariant CFO$[k]$ classes, it can in turn be defined in FO$[f(k)]$.

Finally, because $<$-invariant formulas do not depend on the chosen order, it is enough to equip A and B with suitable orders $<_A$ and $<_B$ and show $(A, <_A) \equiv^k_{\mathsf{CFO}} (B, <_B)$ whenever $A \equiv^{f(k)}_{\mathsf{FO}} B$. It therefore remains to show that whenever two graphs are sufficiently similar, we can choose linear orders for each of them such that, after equipping the graphs with these orders, they become indistinguishable in CFO. In other words, by picking the right orders, we can reduce the problem to CFO equivalence on ordered graphs.

We prove this by introducing a special family of orders, called (k, F)-orders, which satisfy several useful structural properties. First, every graph that is sufficiently rich admits such an order. Next, any two graphs that are similar enough admit corresponding orders that are themselves similar. Finally, these orders allow us to establish CFO-equivalence: by equipping the two graphs with their respective orders, Duplicator obtains a winning strategy in the CEF game. Our construction relies on two complementary ways of describing the local structure of a bounded-degree graph: one tailored to FO and one to CFO. For FO, we use *neighbourhood types*. Fix a radius r. The r-neighbourhood type of a vertex v is the isomorphism type of its radius-r neighbourhood $N_r(v)$. Since FO formulas of bounded quantifier rank can only explore bounded neighbourhoods, vertices with the same r-neighbourhood type are indistinguishable to FO. Hence, on bounded-degree graphs, only finitely many such types can occur, and from the point of view of FO, a graph is essentially determined by how many vertices realise each type.

For CFO, we use *contexts*, which play an analogous role for the clustered, order-based setting. Contexts capture exactly the local configurations that CFO can distinguish. Thus, neighbourhood types describe what FO can see, while contexts describe what CFO can see, and on bounded-degree graphs both notions give rise to only finitely many possibilities and are functionally equivalent in their counts. Consequently, if two graphs are FO-equivalent for a sufficiently large quantifier rank, they realise the same neighbourhood types with the same multiplicities, and therefore also admit matching contexts. This correspondence allows us to align their vertices and construct matching (k, F)-orders, which is the key step toward proving CFO-equivalence.

Since contexts are central to the proof, we briefly describe them here. Informally, a k-context consists of the k-neighbourhood of a vertex equipped with a linear order that partitions it into intervals, together with the $(k-1)$-contexts occurring within these intervals, thereby recording how smaller local configurations are arranged along the order.

In bounded-degree graphs, if a k-neighbourhood type occurs sufficiently many times–say at least $g(k, r, d, m)$ times, where this means that there are $g(k, r, d, m)$ distinct vertices with that type, for a function depending only on k, the degree bound d, and the parameters r and m–then there exist m such vertices whose pairwise distances are greater than r. This property extends to several neighbourhood types simultaneously. In other words, if each of a fixed collection of neighbourhood types occurs sufficiently many times, where the required number depends only on the neighbourhood depth, the maximum degree, the desired pairwise distance, and the number of types under consideration, then we can select many vertices of these types whose pairwise distances are all greater than the prescribed bound. Thus, we can obtain many mutually far-apart occurrences across all the chosen types.

Now let F be a fixed collection of k-neighbourhood types. We say that a graph is *rich for* (k, F) if each type in F occurs sufficiently many times, namely at least a threshold M. This value M is precisely the bound discussed in the previous paragraph: it depends only on k and the degree bound and guarantees many well-separated occurrences of each type. Its exact numerical value is not important for this introductory discussion. What matters is the consequence that every type in F appears more than M times, which allows us to select many occurrences that are pairwise far apart and hence can be used independently to build a structured order.

We construct a (k, F)-order for a graph that is rich for (k, F) as follows.

1. We first place all vertices whose k-neighbourhood types do not belong to F at the beginning of the order. This initial segment is called the *extremity* and is denoted by X.
2. Next, we introduce $2k^2$ segments distributed along the order, called *universal segments*. In each such segment we place one copy of every context that can be constructed using only vertices whose k-neighbourhood types belong to F. These segments are called universal because each of them contains representatives of all such contexts. Recall that a context is simply a small ordered local configuration, consisting of a few vertices together with their bounded neighbourhoods arranged according to the order. Constructing a context therefore amounts to selecting appropriate vertices with the required neighbourhood types and arranging them in the prescribed order.

   The role of the threshold M becomes crucial here. Since every type in F occurs many times, we can choose occurrences that are pairwise far apart. This allows us to build the contexts inside different universal segments using disjoint sets of vertices and neighbourhoods, ensuring that they do not interfere with one another.
3. Between consecutive universal segments we insert *neighbouring segments*, which contain the vertices in the neighbourhood of the previous universal segment. Because the chosen contexts are well-separated, these neighbourhoods do not overlap and the construction can be carried out independently around each segment.
4. Finally, all remaining vertices are placed in a central segment called the *jungle*.

Thus the order has a fixed global structure consisting of the extremity, several universal segments containing all contexts one can construct with the neighbourhood types in F, their neighbouring segments, and the jungle. This global structure comes with an order $\prec$ which is the order between the segments. More precisely, when comparing elements a, b from different segments, b is greater than a if it belongs to a segment to the right of the segment of a. This order is depicted in Figure 1.

Within each universal segment we fix an arbitrary but consistent order on the contexts. Concretely, if $c_1, \ldots, c_\ell$ are all contexts that can be constructed using neighbourhood types from F, we place their chosen copies in the order $c_1 \prec' \cdots \prec' c_\ell$, meaning that every vertex belonging to c_i precedes every vertex belonging to c_j whenever $i < j$, in this short introduction on the order by $a \in c_i$ we mean that the element a belongs to the occurrence of the context c_i. On the extremity X, the neighbouring segments, and the jungle J, we choose arbitrary orders. For a graph $\mathcal{A}$, we denote the resulting order by $<_A$ and call it a (k, F)-order. The existence of such an order relies precisely on the richness assumption. Recall that each context c comes with a built-in order $<_{c,\mathcal{A}}$. Therefore, the order $<_A$ compares elements as follows.

$$a <_A b \iff \begin{cases} \text{segment of } a \prec \text{segment of } b, & \text{if segment of } a \neq \text{segment of } b, \\ c_i \prec' c_j, & \text{if } a, b \text{ belong to the same universal segment,} \\ & a \in c_i,\ b \in c_j, \text{ and } c_i \neq c_j, \\ a <_{c_i, \mathcal{A}} b, & \text{if } a, b \text{ belong to the same universal segment,} \\ & \text{and } a, b \text{ belong to the same context } c_i, \\ a < b & \text{if } a, b \text{ belong to } X \text{ or } J \text{ or a neighbouring segment} \\ & \text{and } < \text{is the order in that segment.} \end{cases}$$

Moreover, if two graphs $\mathcal{A}$ and $\mathcal{B}$ are rich for the same (k, F) and have matching counts for the neighbourhood types outside F, we first construct a (k, F)-order $<_A$ on $\mathcal{A}$ as above and then transfer this order to $\mathcal{B}$, obtaining an order $<_B$ with the same segment decomposition. Finally, we show that Duplicator has a winning strategy in the CEF game on $(\mathcal{A}, <_A)$ and $(\mathcal{B}, <_B)$, yielding $(\mathcal{A}, <_A) \equiv^k_{\mathsf{CFO}} (\mathcal{B}, <_B)$ and completing the proof of Theorem 3.1. ◀

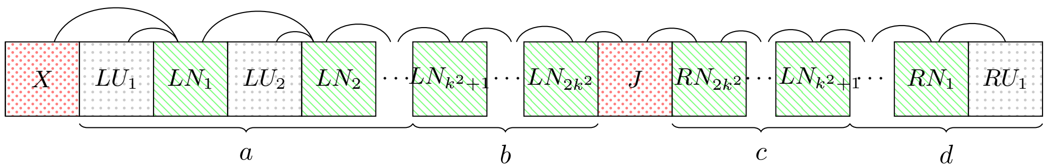

Figure 1 (a) k^2 universal and neighbouring left segments, (b) k^2 only left neighbouring segments, (c) k^2 only right neighbouring segments, and (d) k^2 universal and neighbouring right segments. On the left side, LN_i contains the neighbours of the vertices in $LU_i \cup LN_{i-1}$ for $2 \leqslant i \leqslant k^2$, only LN_{i-1} for $k^2 + 1 \leqslant i$, and $LN_1 := N(LU_1 \cup X)$. The same patterns appear on the right side. The extremity X contains all those vertices whose k-neighbourhood types does not belong to F. The Jungle J, is filled with the remaining vertices after all other segments are filled.

4 Model checking

In this section, we tackle the *model-checking* problem for $<$-invariant CFO. This problem amounts, given a graph and a sentence from $<$-invariant CFO, to deciding whether this sentence holds in the input graph. Since both input are asymetric in size – the graph is expected to be much larger than the formula – it is important for the time complexity to be polynomial in the size of the graph, but less crucial if the dependency in the length of the formula is expensive. This corresponds to deciding whether this model-checking problem is *fixed-parameter tractable* (FPT), where the size of the formula is taken as parameter.

We prove that when the degree is bounded, the model-checking problem for $<$-invariant CFO is indeed FPT. Altough FO is known to have an FPT model-checking problem when the degree is bounded [16], this result is not an immediate consequence of the inclusion of $<$-invariant CFO in FO as we do not know the translation to be computable.

Instead, we use again the tools that were developed in the previous section, namely the notion of contexts, to prove that whether a sentence holds in a given graph depends only on the set of contexts occurring in some ordering of this graph. Note that we must consider a slight extension of the notion of k-contexts, in the form of k*-outer contexts*, where the $(k-1)$-contexts occurring in the leftmost and rightmost intervals are considered, while they are disregarded in regular k-contexts. Given that the number of k-outer contexts is bounded, the model-checking can then take place on some data whose size is bounded independently of the size of the input graph.

We then show that the set of k-outer contexts realised in an ordered graph can be computed in FPT. When combined, these two results provide an FPT algorithm for the model-checking problem on classes of bounded degree.

▶ **Theorem 4.1.** *The model-checking problem for $<$-invariant* CFO *on a class $\mathcal{C}$ of bounded degree is fixed-parameter tractable.*

5 Separating example

We conclude with a result showing that the bounded-degree hypothesis is crucial for $<$-invariant CFO to be included in FO:

▶ **Theorem 5.1.** *There exist properties definable in $<$-invariant* CFO *that are not definable in* FO*.*

The separation follows from a classical example due to Gurevich. The property QEven states that a structure is isomorphic to a Boolean algebra $(2^X, \subseteq)$ where $|X|$ is even. This property is known not to be FO-definable by an Ehrenfeucht–Fraïssé argument, but it is definable in order-invariant first-order logic. We show that the same construction can already be carried out in order-invariant CFO: the order is used to isolate a single connected component, and the clustered quantification then simulates the original definition. Hence QEven is definable in order-invariant CFO but not in FO.

6 Conclusion

There remain many open problems and directions left for exploration. The gap between order-invariant cluster first-order logic and order-invariant first-order logic should be studied. One can also examine if the presented order can be tweaked and somehow lifted to plain FO instead of only CFO. The expressive power of order-invariant cluster FO on more general classes is also of interest; we conjecture that the same upper bound can be shown for *rooted trees* as well. On trees, it is already known that order-invariant FO collapses to plain FO with respect to expressive power. However, in [3], the authors indirectly use the second method and compute orders maintaining similarity of already FO-similar trees, but for a chain of unbounded length of intermediate structures between the two structures they began with. Their approach highly relies on the good properties of trees and does not seem to be generalisable to more general classes of structures. Nevertheless, we believe that this result could be recovered through a more direct approach, namely by constructing suitable orders on FO-similar trees so that their equivalence is preserved, and that establishing this result first for $<$-invariant CFO would be a stepping stone toward this goal. As stated above, we consider the development of such a method to be of independent interest, as it may be extendable to broader graph classes. We also conjecture that FO is contained in $<$-invariant CFO in general.

References

1. Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995. URL: `http://webdam.inria.fr/Alice/`.
2. Bartosz Bednarczyk and Julien Grange. About the expressive power and complexity of order-invariance with two variables. *Log. Methods Comput. Sci.*, 21(1), 2025. doi:10.46298/LMCS-21(1:30)2025.

3 Michael Benedikt and Luc Segoufin. Towards a characterization of order-invariant queries over tame graphs. *J. Symb. Log.*, 74(1):168–186, 2009. doi:10.2178/JSL/1231082307.

4 Édouard Bonnet, Ugo Giocanti, Patrice Ossona de Mendez, Pierre Simon, Stéphan Thomassé, and Szymon Toruńczyk. Twin-width IV: Ordered graphs and matrices. *J. ACM*, 71(3):21:1–21:42, 2024. doi:10.1145/3651151.

5 Kord Eickmeyer, Michael Elberfeld, and Frederik Harwath. Succinctness of order-invariant logics on depth-bounded structures. *ACM Trans. Comput. Log.*, 18(4):33:1–33:25, 2017. doi:10.1145/3152770.

6 Kord Eickmeyer and Ken-ichi Kawarabayashi. Successor-invariant first-order logic on graphs with excluded topological subgraphs. In Jean-Marc Talbot and Laurent Regnier, editors, *25th EACSL Annual Conference on Computer Science Logic, CSL 2016, Marseille, France, August 29 - September 1, 2016*, volume 62 of *LIPIcs*, pages 18:1–18:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.CSL.2016.18.

7 Kord Eickmeyer, Jan van den Heuvel, Ken-ichi Kawarabayashi, Stephan Kreutzer, Patrice Ossona de Mendez, Michal Pilipczuk, Daniel A. Quiroz, Roman Rabinovich, and Sebastian Siebertz. Model-checking on ordered structures. *ACM Trans. Comput. Log.*, 21(2):11:1–11:28, 2020. doi:10.1145/3360011.

8 Michael Elberfeld, Marlin Frickenschmidt, and Martin Grohe. Order invariance on decomposable structures. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2016, New York, NY, USA, July 5-8, 2016*, pages 397–406. ACM, 2016. doi:10.1145/2933575.2934517.

9 Julien Grange. Successor-invariant first-order logic on classes of bounded degree. *Log. Methods Comput. Sci.*, 17(3), 2021. doi:10.46298/LMCS-17(3:20)2021.

10 Julien Grange and Luc Segoufin. Order-invariant first-order logic over hollow trees. In *28th EACSL Annual Conference on Computer Science Logic, CSL 2020*, volume 152 of *LIPIcs*, pages 23:1–23:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.CSL.2020.23.

11 Martin Grohe and Thomas Schwentick. Locality of order-invariant first-order formulas. *ACM Trans. Comput. Log.*, 1(1):112–130, 2000. doi:10.1145/343369.343386.

12 Neil Immerman. *Descriptive Complexity*. Graduate Texts in Computer Science. Springer, 1999. doi:10.1007/978-1-4612-0539-5.

13 Leonid Libkin. *Elements of Finite Model Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2004. doi:10.1007/978-3-662-07003-1.

14 Benjamin Rossman. Successor-invariance in the finite. In *18th IEEE Symposium on Logic in Computer Science (LICS 2003), 22-25 June 2003, Ottawa, Canada, Proceedings*, page 148. IEEE Computer Society, 2003. doi:10.1109/LICS.2003.1210054.

15 Nicole Schweikardt. A short tutorial on order-invariant first-order logic. In Andrei A. Bulatov and Arseny M. Shur, editors, *Computer Science - Theory and Applications - 8th International Computer Science Symposium in Russia, CSR 2013, Ekaterinburg, Russia, June 25-29, 2013. Proceedings*, volume 7913 of *Lecture Notes in Computer Science*, pages 112–126. Springer, 2013. doi:10.1007/978-3-642-38536-0_10.

16 Detlef Seese. Linear time computable problems and first-order descriptions. *Math. Struct. Comput. Sci.*, 6(6):505–526, 1996. doi:10.1017/S0960129500070079.

17 Jan van den Heuvel, Stephan Kreutzer, Michal Pilipczuk, Daniel A. Quiroz, Roman Rabinovich, and Sebastian Siebertz. Model-checking for successor-invariant first-order formulas on graph classes of bounded expansion. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017*, pages 1–11. IEEE Computer Society, 2017. doi:10.1109/LICS.2017.8005115.

18 Victor Vianu. Databases and finite-model theory. In *Descriptive Complexity and Finite Models*, pages 97–148. American Mathematical Society, 1996. doi:10.1090/DIMACS/031/04.