Logical Labeling Schemes

Maurice Chandoo*

Abstract. A labeling scheme is a space-efficient data structure for encoding graphs from a particular graph class. The idea is to assign each vertex of a graph a short label s.t. adjacency of two vertices can be algorithmically determined from their labels. For instance, planar and interval graphs have labeling schemes. The algorithm used to determine adjacency—called label decoding algorithm—should be of low complexity since the time it takes to execute corresponds to the time to query an edge in that representation.

What graph classes have a labeling scheme if the label decoding algorithm must be very efficient, e.g. computable in constant time? In order to investigate this question we introduce logical labeling schemes where the label decoding algorithm is expressed as a first-order formula and consider their properties such as the relation to labeling schemes defined in terms of classical complexity classes. Additionally, we introduce a notion of reduction between graph classes in terms of boolean formulas and show completeness results.

Keywords: implicit representations, graph class reduction, structural graph theory

1 Introduction

Labeling schemes are a type of data structure that provide space-optimal representations for certain graph classes up to a constant factor. Let us consider interval graphs as an example. A graph is an interval graph if each of its vertices can be mapped to a closed interval on the real line such that two vertices are adjacent iff their corresponding intervals intersect. There are $2^{\mathcal{O}(n \log n)}$ different interval graphs on n vertices. Neither adjacency matrix nor adjacency list are optimal to represent an interval graph since both require more than $\mathcal{O}(n \log n)$ bits. Instead, the interval model of an interval graph can be used: given an interval graph G with n vertices, write down its interval model (the set of intervals that correspond to its vertices), enumerate the endpoints of the intervals from left to right and label each vertex with the two endpoints of its interval, see Figure 1. The set of vertex labels is a representation of the graph and adjacency of two vertices can be determined by comparing their four endpoints. Each endpoint is a number between 1 and $2n$ and therefore a vertex label requires $2 \log 2n$ bits. Thus, such a representation of an interval graph requires only $\mathcal{O}(n \log n)$ bits.

The idea behind this representation can be generalized. Let $\mathcal{C}$ be a graph class with $2^{\mathcal{O}(n \log n)}$ graphs on n vertices; we call such a graph class factorial[1]. We say $\mathcal{C}$ has a labeling scheme if the vertices of every graph in $\mathcal{C}$ can be assigned binary labels of length $\mathcal{O}(\log n)$ such that adjacency can be decided by an (efficient) algorithm A which gets two labels as input. The algorithm A may only depend on $\mathcal{C}$. By adjusting the label length it is also possible to find labeling schemes for

*Leibniz Universität Hannover, Institut für Theoretische Informatik; E-Mail: chandoo@thi.uni-hannover.de

[1] In the literature such graph classes are called *at most factorial* and the term factorial is reserved for graph classes with $2^{\Theta(n \log n)}$ graphs on n vertices see [BBW00; HWZ22]

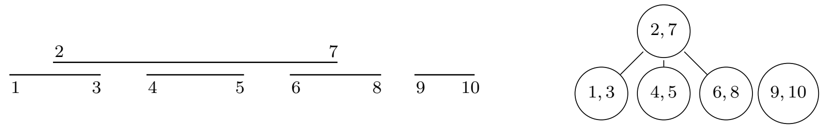

Figure 1: Interval model and the resulting labeling of the interval graph

non-factorial classes. However, many important graph classes are factorial and therefore we restrict our attention to them.

Labeling schemes were introduced by Muller [Mul88] and by Kannan, Naor and Rudich [KNR92]. One line of research in this area has aimed to minimize the label length (the constant in $\mathcal{O}(\log n)$) for certain graph classes such as forests and planar graphs since this also puts an upper bound on the number of graphs on n vertices of such a class [ADK17; Duj+21; Bon+21a]. Another fundamental question is what factorial and hereditary (= closed under vertex deletion) graph classes have labeling schemes. Until recently, it was not even known if there exist a factorial, hereditary graph class without a labeling scheme. This long open-standing problem, also known as the implicit graph conjecture [Spi03], has been recently solved by Hatami & Hatami [HH22]. They prove the existence of such graph classes using a counting argument. An additional interesting consequence of their argument is that any kind of representation for factorial graph classes will fail to represent all factorial, hereditary graph classes (see Lemma 2.4 and the subsequent paragraph).

Nonetheless, it remains a challenging problem to determine for certain natural graph classes whether they have a labeling scheme. Factorial, hereditary graph classes for which no labeling schemes are known include disk graphs, line segment graphs, k-dot product graphs [Spi03], graph classes with bounded functionality [AAL21], P_7-free bipartite graphs [LZ17] and T-free chordal bipartite graphs for every tree T [LZ15]. Also, for those graph classes for which a labeling scheme is already known, it is of interest to determine whether a labeling scheme with a label decoding algorithm in constant time is possible. For instance, such labeling schemes for graph classes with bounded clique-width are not known.

Labeling schemes for many graph classes have a label decoder that can be expressed as a first-order formula that only adds, multiplies and compares numbers. Such label decoders can be computed in constant time on a RAM [Cha17, Cor. 3.84]. This motivated us to investigate this class of labeling schemes, which we call logical. We show that certain fragments of logical labeling schemes admit various characterizations and have connections to semi-algebraic graph classes and communication complexity [HWZ22]. This suggests that they describe robust natural sets of graph classes. Additionally, proving lower bounds against logical labeling schemes seems more feasible than against labeling schemes with arbitrary efficiently computable label decoders due to the structure of first-order formulas.

Overview. In Section 2 we formally define labeling schemes and show how classes of labeling schemes can be defined in terms of sets of languages. In Section 3 we introduce logical labeling schemes where the label decoder is expressed as a first-order formula and relate them to classes of labeling schemes defined in terms of complexity classes. Moreover, we show that quantifiers do not increase the expressiveness of logical labeling schemes without addition and multiplication and that a certain fragment of logical labeling schemes coincides with equality-based labeling schemes introduced in [HWZ22]. In Section 4 we consider what happens when the size restriction on the labeling is omitted in quantifier-free logical labeling schemes. Graph classes that can be represented in such a way are a subset of semi-algebraic graph classes. Many factorial, hereditary graph classes not known to have a labeling scheme can be found there. In Section 5 we define a reduction notion between graph classes, which allows us to relate the difficulty of finding labeling schemes for different graph classes. We prove that two graph classes called dichotomic and linear neighborhood graphs are complete for certain fragments of logical labeling schemes. Additionally, we prove that no uniformly sparse graph class is complete for any such fragment. Figure 3 in the final section provides an overview of all the sets of graph classes discussed here and their relations.

2 Preliminaries

Notation. We write $\mathbb{N}, \mathbb{N}_0, \mathbb{Z}, \mathbb{Q}$ and $\mathbb{R}$ to denote the set of naturals excluding 0, naturals including 0, integers, rationals and reals. For $n \in \mathbb{N}$ let $[n] = \{1, \ldots, n\}$ and $[n]_0 = [n] \cup \{0\}$. Let $\log n = \lceil \log_2 n \rceil$. Given a set X whose elements are sets, let $[X]_{\subseteq} = \{x' \mid x \in X \text{ and } x' \subseteq x\}$ denote its closure under subsets. For a function f which maps to a k-tuple, we write $f_i(x)$ to denote the i-th component for $k \in \mathbb{N}$ and $i \in [k]$. We consider an undirected graph to be a directed graph with symmetric edge relation. Every graph $G = (V, E)$ is assumed to contain no self-loops ($(v, v) \notin E$ for all $v \in V$) unless explicitly stated otherwise; we say "a graph with self-loops" to indicate that it *may* contain self-loops. The in-neighborhood $N_{\text{in}}(v)$ of a vertex v in a graph G is $\{u \mid (u, v) \in E(G)\}$ and the out-neighborhood $N_{\text{out}}(v)$ is $\{u \mid (v, u) \in E(G)\}$. A graph class is a set of graphs closed under isomorphism. For a graph class $\mathcal{C}$ and $n \in \mathbb{N}$ let $\mathcal{C}_{=n}, \mathcal{C}_{>n}, \mathcal{C}_{<n}$ denote the set of graphs with n, more than n and less than n vertices in $\mathcal{C}$.

A boolean formula is an expression consisting of propositional variables and the boolean connectives $\neg, \wedge, \vee$. A first-order formula φ over signature σ is an expression consisting of boolean connectives $\neg, \wedge, \vee$, quantifiers $\exists, \forall$, the equality symbol '=', relation and function symbols from σ and variables. A variable in φ is called free if it is not quantified. A first-order formula is called atomic if it contains no quantifiers and boolean connectives. The structure $\mathcal{N}$ has $\mathbb{N}_0$ as universe and is equipped with order '<' and addition '+' and multiplication '×' as functions, i.e. $+(x, y) = x + y$ and $\times(x, y) = xy$. The structure $\mathcal{N}_n$ has $[n]_0$ as universe and is equipped with order '<', cut-off addition '+' ($+(x, y) := x + y$ if $x + y \leq n$ and 0 otherwise) and cut-off multiplication '×'. Given a first-order formula φ over $\{<, +, \times\}$ with k free variables and $a_1, \ldots, a_k \in \mathbb{N}$, we write $(\mathcal{M}, a_1, \ldots, a_k) \models \varphi$ to denote that φ is satisfied (modeled) when its free variables are replaced with $a_1, \ldots, a_k$ and it is interpreted over the structure $\mathcal{M}$.

Graph Theory. A graph class $\mathcal{C}$ is undirected if it contains only undirected graphs. A graph class $\mathcal{C}$ is hereditary if every graph that occurs as induced subgraph of a graph in $\mathcal{C}$ is in $\mathcal{C}$ as well. For example, forests and planar graphs are hereditary but trees are not. The hereditary closure $[\mathcal{C}]_{\text{hc}}$ of a graph class $\mathcal{C}$ is the set of graphs that occur as induced subgraph of a graph in $\mathcal{C}$. For example, the hereditary closure of trees are forests. A graph class is at most factorial if it has at most $2^{\mathcal{O}(n \log n)}$ different graphs on n vertices; for brevity we will omit the qualifier 'at most'. A graph class $\mathcal{C}$ is uniformly sparse if every graph G with n vertices in $[\mathcal{C}]_{\text{hc}}$ has at most cn edges for some fixed $c \in \mathbb{N}$ and all $n \in \mathbb{N}$. We write $\mathsf{Hereditary}$, $\mathsf{Factorial}$ and US to denote the set of hereditary, factorial and uniformly sparse graph classes. A graph class $\mathcal{C}$ is said to have a polynomial-size universal graph if there exists a sequence $G_1, G_2, \ldots$ of graphs such that every graph in $\mathcal{C}_{=n}$ is an induced subgraph of G_n for all $n \in \mathbb{N}$ and $n \mapsto |V(G_n)|$ is polynomially bounded.

An intersection graph class is a graph class where the vertices of every graph from that class can be mapped to some type of object (e.g. line segments in the plane) such that two vertices are adjacent iff their associated objects intersect. Line segment graphs, disk graphs and k-box graphs are the intersection graphs of line segments in $\mathbb{R}^2$, disks in $\mathbb{R}^2$ and k-dimensional axis-parallel boxes in $\mathbb{R}^k$ for $k \in \mathbb{N}$. A permutation graph is the intersection of line segments whose endpoints are placed on two paralells (the two endpoints of a line segment may not be placed on the same parallel). A graph is a k-interval graph if each of its vertices can be associated with k closed intervals on the real line such that two vertices are adjacent iff some of their intervals intersect. A graph $G = (V, E)$ is a k-dot product graph if there exists a mapping $f \colon V \to \mathbb{R}^k$ such that $(u, v) \in E$ iff

$\sum_{i=1}^{k} f_i(u)f_i(v) \geq 1$ for all $u \neq v \in V$. The interval number (resp. boxicity) of a graph G is the smallest $k \in \mathbb{N}$ such that G is a k-interval (resp. k-box) graph. The arboricity (resp. thickness) of a graph G is the smallest $k \in \mathbb{N}$ such that there exist k forests (resp. planar graphs) $G_1, \ldots, G_k$ on the same vertex set as G such that $E(G) = \cup_{i=1}^{k} E(G_i)$.

Complexity Theory. Let P, EXP and $\mathsf{2EXP}$ denote the set of languages that are decidable in polynomial time, exponential time and double exponential time. Let R denote the set of decidable languages and let PH denote the set of languages in the polynomial-time hierarchy. Additionally, we will refer to the two circuit complexity classes AC^0 and TC^0. The class AC^0 consists of the languages over $\{0, 1\}$ that can be decided by a family of boolean circuits of polynomial size and constant depth using negation gates and gates for conjunction and disjunction with unbounded fan-in. The class TC^0 is defined just as AC^0 but additionally majority gates (outputs 1 iff the majority of its inputs is 1) with unbounded fan-in may be used. Logspace-uniformity is assumed. It holds that $\mathsf{AC}^0 \subseteq \mathsf{TC}^0 \subseteq \mathsf{P}$. Moreover, order can be computed in AC^0 and multiplication can be computed TC^0. See [Vol99] for formal definitions and the mentioned results in circuit complexity.

Complexity classes can be interpreted as sets of labeling schemes by viewing a label decoder as decision problem. By extension, every complexity class A can be associated with the set of graph classes GA represented by its labeling schemes.

Definition 2.1. A labeling scheme is a tuple $S = (F, c)$ where $F \subseteq \{0, 1\}^* \times \{0, 1\}^*$ is called label decoder and $c \in \mathbb{N}$ is called label length. A graph G with n vertices is representable by S, in symbols $G \in \mathrm{gr}(S)$, if there exists a labeling $\ell \colon V(G) \to \{0, 1\}^{c \log n}$ such that for all $u \neq v \in V(G)$:

$$(u, v) \in E(G) \Leftrightarrow (\ell(u), \ell(v)) \in F$$

We call S a labeling scheme for a graph class $\mathcal{C}$ if every graph in $\mathcal{C}$ is representable by S ($\mathcal{C} \subseteq \mathrm{gr}(S)$). We also say S represents $\mathcal{C}$.

In a labeling scheme with label decoder F only queries '$(x, y) \in F$?' where x and y have equal length are ever made. Thus, we can assume w.l.o.g. that $(x, y) \in F$ implies $|x| = |y|$ for all label decoders F. A label decoder can be encoded as language over $\{0, 1\}$ by concatenating its entries.

Definition 2.2. Let $F \subseteq \{0, 1\}^* \times \{0, 1\}^*$ be a label decoder and let $L(F) = \{xy \mid (x, y) \in F\}$. Let A be a set of languages over $\{0, 1\}$. We say a graph class $\mathcal{C}$ is in GA if there exists a labeling scheme (F, c) for $\mathcal{C}$ with $L(F) \in \mathsf{A}$.

For example, GP is the complexity class of graph classes that have a labeling scheme with a polynomial-time computable label decoder. This means it takes polylogarithmic time to query an edge in a graph with n vertices since the labels have $\mathcal{O}(\log n)$ length. The classes GR (computable label decoder) and GP coincide with the ones defined by Muller [Mul88] and Kannan et al. [KNR92], respectively. The class GALL is the set of graph classes that have a labeling scheme without any restriction on the label decoder (ALL denotes the set of all languages). A graph class is in GALL iff it has a polynomial-size universal graph.

What graph classes have a labeling scheme? Due to the $\mathcal{O}(\log n)$-size restriction on the label length, only factorial graph classes qualify for having a labeling scheme. Not all factorial graph classes have a labeling scheme [Spi03, Thm. 2.1]. However, an arbitrary factorial graph class $\mathcal{C}$ may have no meaningful structure at all since there does not need to be any relation between the graphs on n vertices in $\mathcal{C}$ and the ones on m vertices for any $n \neq m$. To exclude such cases a common graph-theoretical uniformity requirement that can be imposed is that a graph class should be hereditary.

This leads to the question whether every factorial, hereditary graph class has a labeling scheme, which already has been posed in [KNR92]. Its affirmative statement became known as implicit graph conjecture. Recently, Hatami & Hatami proved that the answer to this question is negative. Their counting argument assumes no computational restriction on the label decoder and it also implies that any kind of finite representation for factorial graph classes will fail to capture all factorial, hereditary graph classes; see the paragraph after Lemma 2.4.

Theorem 2.3 ([HH22])**.** *There exists a factorial, hereditary graph class which has no polynomial-size universal graph, i.e.* $\mathsf{Factorial} \cap \mathsf{Hereditary} \not\subseteq \mathsf{GALL}$.

Lemma 2.4. *There exists no countable set of factorial graph classes* $\mathbb{X}$ *such that for every factorial, hereditary graph class* $\mathcal{C}$ *there exists a* $\mathcal{D} \in \mathbb{X}$ *with* $\mathcal{C} \subseteq \mathcal{D}$.

Proof. Let $k_n = 2^{\sqrt{n}}$ and $0 < \epsilon < 1$. In [HH22, Claim 3.1] it is shown that there exists a graph class $\mathcal{G}$ such that for every family of graph classes $(\mathcal{M}_n)_{n\in\mathbb{N}}$ with $\mathcal{M}_n \subseteq \mathcal{G}_{=n}$ and $|\mathcal{M}_n| = k_n$ the hereditary closure of $\cup_{n\in\mathbb{N}}\mathcal{M}_n$ is factorial. Moreover, it holds that $|\mathcal{G}_{=n}| \in 2^{\Omega(n^{2-\epsilon}\log n)}$ and therefore for each $\mathcal{M}_n$ there are $2^{\Omega(k_n n^{2-\epsilon}\log n)}$ choices.

Let $\mathbb{X} = \{\mathcal{C}_1, \mathcal{C}_2, \dots\}$ be a countable set of factorial graph classes. A factorial graph class $\mathcal{C}$ contains at most $2^{\mathcal{O}(k_n n\log n)}$ different sets of n-vertex graphs with k_n members, which is less than the number of choices for $\mathcal{M}_n$ for sufficiently large n. Therefore, for every $i \in \mathbb{N}$ one can choose a unique, sufficiently large n and $\mathcal{M}_n$ such that $\mathcal{M}_n \not\subseteq \mathcal{C}_i$. The hereditary closure of $\cup_{n\in\mathbb{N}}\mathcal{M}_n$ is factorial and not a subset of any class in $\mathbb{X}$. □

Let us call a function $r\colon \{0,1\}^* \to \mathsf{Factorial}$ a kind of (finite) representation (for factorial graph classes). We say a graph class $\mathcal{C}$ is representable in r if there exists a finite string x such that $\mathcal{C}$ is a subset of $r(x)$. The class of labeling schemes with computable label decoders is a kind of representation: $x \mapsto \mathrm{gr}(F, c)$ where x encodes a natural number c and a Turing machine which decides F. Lemma 2.4 implies that any kind of representation r will fail to represent all factorial, hereditary graph classes (the image of r corresponds to $\mathbb{X}$). A consequence of this is that not every factorial, hereditary graph class is semi-algebraic since polynomial-boolean systems (used to define semi-algebraic graph classes) are a kind of representation (see Lemma 4.2).

This implies that any kind of finite representation for factorial graph classes will fail to represent all factorial, hereditary graph classes since every representation being finite implies that there are only countably many and a representation is assumed to represent a factorial graph class and all of its subsets.

It remains an open problem to determine for natural graph classes such as disk graphs and others mentioned in the introduction whether they have a labeling scheme. The counting argument is based on constructing a set of factorial, hereditary graph classes of which there must be 'more' than what can be represented by polynomial-size universal graphs. This line of reasoning cannot be applied when the existence of a labeling scheme for a specific graph class is considered. Instead, it seems likely that restrictions on the label decoder must be presupposed in order to prove the absence of a labeling scheme for a specific graph class. We shall see the first such negative result at the end of the next section.

The following result shows via a diagonalization argument that restricting the computational complexity of the label decoder does indeed affect the set of graph classes that can be represented.

Theorem 2.5 ([Cha17, Corollary 3.4])**.** $\mathsf{GEXP} \subsetneq \mathsf{G2EXP} \subsetneq \cdots \subsetneq \mathsf{GR} \subsetneq \mathsf{GALL}$.

Many graph classes for which a labeling scheme is known can be trivially placed in GAC^0. This suggests that it is an interesting candidate to prove lower bounds against. For example, graph classes with bounded clique-width, interval number or boxicity are in GAC^0. A labeling scheme for graph classes with bounded clique-width is described in [Spi03, p. 165 f.] whose label decoder can be computed in AC^0. In fact, the only exception that we are aware of are graph classes with bounded twin-width and induced subgraphs of hypercubes for which labeling schemes are known but it is not clear whether they are in GAC^0. These graph classes are known to be in GP [Bon+21b] and GR [Har20; EHZ22], respectively.

To stay within the realm of graph theory, i.e. graph classes with some sort of graph-theoretical structure, we already mentioned that one should consider graph classes which are not only factorial but also hereditary. Additionally, one can also admit subsets of such classes, which leads to the set $[\mathsf{Factorial} \cap \mathsf{Hereditary}]_\subseteq$. If a factorial graph class is not in $[\mathsf{Factorial} \cap \mathsf{Hereditary}]_\subseteq$, this implies that its hereditary closure is not factorial.

Fact 2.6 ([Cha17, Theorem 4.5])**.** $\mathsf{GAC}^0 \not\subseteq [\mathsf{Factorial} \cap \mathsf{Hereditary}]_\subseteq$.

Informally, this may be interpreted as the fact that GAC^0 is not well-behaved from a graph-theoretical point of view. The same applies to all other complexity classes considered here as well since they are supersets of AC^0. In the next section, we shall see that if label decoders are defined in terms of quantifier-free first-order formulas then the representable graph classes stay within $[\mathsf{Factorial} \cap \mathsf{Hereditary}]_\subseteq$.

3 Logical Labeling Schemes

Establishing unconditional lower bounds in the context of computational complexity is difficult and usually involves considering weak models of computation. The setting of labeling schemes adds an additional layer of complexity to that task. Even though AC^0 is among the smallest complexity classes studied in complexity theory, it currently seems intractable to prove that a natural factorial, hereditary graph class is outside of GAC^0. Therefore a simpler class of labeling schemes that can still represent interesting graph classes would be helpful.

The following class of labeling schemes fits that bill. Suppose that each vertex of a graph G with n vertices is labeled with k integers between 0 and n^c for some fixed c and k. These k integers can be encoded using $\mathcal{O}(\log n)$ bits. The label decoder is allowed to add, multiply and compare these numbers and then determine adjacency from these comparisons. For example, the labeling scheme for interval graphs from the introduction falls into this class. Each vertex is labeled with two numbers between 1 and $2n$ to represent the endpoints of its interval and two vertices u, v with numbers u_1, u_2, v_1, v_2 are adjacent iff neither $u_2 < v_1$ nor $v_2 < u_1$ meaning no interval ends before the other starts; in this case $c, k = 2$. This type of label decoder can be formalized using first-order formulas. All uniformly sparse graph classes, k-interval graphs and graph classes with bounded boxicity can be represented with such labeling schemes.

By imposing syntactical restrictions on the formulas such as prohibiting quantifiers, different classes of labeling schemes can be obtained that admit various characterizations. An interesting instance of such a characterization is the following. Harms, Wild and Zamaraev have recently introduced and investigated a class of labeling schemes called equality-based whose definition is motivated from communication complexity [HWZ22]. They prove that graph classes which are unbounded w.r.t. a certain parameter cannot have such a labeling scheme. We show that this class of labeling schemes coincides with a certain fragment of logical labeling schemes.

In this section we formally define logical labeling schemes (Definition 3.1 & 3.2), describe their relation to classical complexity classes (Theorem 3.4), show that quantifiers do not affect the set of graph classes that can be represented if addition and multiplication are not allowed (Fact 3.6 and Theorem 3.7) and that equality-based labeling schemes coincide with the equality fragment of logical labeling schemes (Lemma 3.11). The last part implies that label decoders which can compare order '$x < y$' as opposed to just equality '$x = y$' are strictly more expressive (Corollary 3.12).

Definition 3.1. A logical labeling scheme is a tuple $S = (\varphi, c)$ where φ is a first-order formula over the signature $\{<, +, \times\}$ with $2k$ free variables and $c, k \in \mathbb{N}$. If φ contains no quantifiers we call

S quantifier-free. For a graph G with n vertices we define the following three interpretations of S:

$$G \in \mathrm{gr}(S) :\Leftrightarrow \exists \ell\colon V(G) \to [n^c]_0^k \;\; \forall u \neq v \in V(G)\colon (u,v) \in E(G) \Leftrightarrow (\mathcal{N}_{n^c}, \ell(u), \ell(v)) \models \varphi$$
$$G \in \mathrm{gr}_\infty(S) :\Leftrightarrow \exists \ell\colon V(G) \to [n^c]_0^k \;\; \forall u \neq v \in V(G)\colon (u,v) \in E(G) \Leftrightarrow (\mathcal{N}, \ell(u), \ell(v)) \models \varphi$$
$$G \in \mathrm{gr}_\mathrm{p}(S) :\Leftrightarrow \exists \ell\colon V(G) \to \mathbb{N}_0^k \quad \forall u \neq v \in V(G)\colon (u,v) \in E(G) \Leftrightarrow (\mathcal{N}, \ell(u), \ell(v)) \models \varphi$$

For a logical labeling scheme $S = (\varphi, c)$ and a signature $\sigma \subseteq \{<, +, \times\}$ we say S is over σ if φ is a formula over σ, i.e. φ uses only symbols from σ.

The definition of $\mathrm{gr}(S)$ essentially states that a graph can be represented by a logical labeling scheme $S = (\varphi, c)$ if each of its vertices can be labeled with k numbers from $[n^c]_0$ such that there is an edge (u, v) iff φ is satisfied when plugging in the numbers $\ell(u)$ and $\ell(v)$ and evaluating it over the finite structure $\mathcal{N}_{n^c}$. Due to the finiteness addition and multiplication are cut-off, e.g. a term $+(x, y)$ evaluates to 0 if $x + y > n^c$. The definition of $\mathrm{gr}_\infty(S)$ is identical to $\mathrm{gr}(S)$ except that φ is evaluated over $\mathcal{N}$ and therefore addition and multiplication are not cut-off (no overflow can occur). The definition of $\mathrm{gr}_\mathrm{p}(S)$ is identical to $\mathrm{gr}_\infty(S)$ except that the numbers used to label the vertices can be arbitrarily large instead of being at most n^c. Therefore this interpretation does not correspond to a labeling scheme. However, it does capture some factorial, hereditary graph classes for which no labeling scheme is known and which will be considered in the next section.

Definition 3.2. Let $\sigma \subseteq \{<, +, \times\}$. We define $\mathsf{GFO}(\sigma)$ (resp. $\mathsf{GFO}_\mathrm{qf}(\sigma)$) as the set of graph classes $\mathcal{C}$ for which there exists a (quantifier-free) logical labeling scheme S over σ such that $\mathcal{C} \subseteq \mathrm{gr}(S)$.

We omit the curly braces when referring to these classes, e.g. $\mathsf{GFO}(<, +) = \mathsf{GFO}(\{<, +\})$. Moreover, we write $\mathsf{GFO}_\mathrm{qf}/\mathsf{GFO}$ as shorthand for $\mathsf{GFO}_\mathrm{qf}(<, +, \times)/\mathsf{GFO}(<, +, \times)$ and $\mathsf{GFO}_{(\mathrm{qf})}(=)$ as synonym for $\mathsf{GFO}_{(\mathrm{qf})}(\emptyset)$ since equality is the only symbol available when $\sigma = \emptyset$. We say a graph class $\mathcal{C}$ is in $\mathsf{GFO}(\sigma)$ (resp. $\mathsf{GFO}_\mathrm{qf}(\sigma)$) via a logical labeling scheme S if S is over σ (and quantifier-free) and $\mathcal{C} \subseteq \mathrm{gr}(S)$. For example, interval graphs are in $\mathsf{GFO}_\mathrm{qf}(<)$ via $(\varphi, 2)$ with $\varphi \triangleq \neg(x_2 < y_1 \vee y_2 < x_1)$.

Suppose $\mathrm{gr}(S)$ in Definition 3.2 is replaced with $\mathrm{gr}_\infty(S)$. Does that affect the set of graph classes defined by $\mathsf{GFO}(\sigma)$ or $\mathsf{GFO}_\mathrm{qf}(\sigma)$? The following lemma shows that it does not make a difference for the class $\mathsf{GFO}_\mathrm{qf}(\sigma)$ if σ contains '$<$' or $\sigma = \emptyset$. This means that in the context of quantifier-free logical labeling schemes we can assume the natural interpretation of addition and multiplication instead of the cut-off version when order is present.

Lemma 3.3 (Overflow). *Let $\sigma \subseteq \{<, +, \times\}$ s.t. $\sigma = \emptyset$ or σ contains '$<$'. A graph class $\mathcal{C}$ is in $\mathsf{GFO}_\mathrm{qf}(\sigma)$ iff there exists a quantifier-free logical labeling scheme S over σ such that $\mathcal{C} \subseteq gr_\infty(S)$.*

Proof. If $\sigma = \emptyset$ then $\mathrm{gr}(S) = \mathrm{gr}_\infty(S)$ holds for every logical labeling scheme S over σ since no overflow can occur without using addition or multiplication. Therefore the statement trivially holds. Let us consider the other cases where σ contains '$<$'.

"$\Rightarrow$": Let $\mathcal{C}$ be a graph class in $\mathsf{GFO}_\mathrm{qf}(\sigma)$ via a logical labeling scheme $S = (\varphi, c)$. We construct a quantifier-free logical labeling scheme $S' = (\psi, c)$ over σ from S such that $\mathcal{C} \subseteq \mathrm{gr}_\infty(S')$. We assume w.l.o.g. that we have access to the constants $c_0 = 0$ and $c_1 = n^c$ in ψ. The constants can be realized by adding two variables to each vertex which are promised to receive the values 0 and n^c in every labeling; this means ψ has $2(k + 2)$ free variables if φ has $2k$ free variables. (Strictly speaking, c_1 is not a logical constant since its value depends on n, which is not constant w.r.t. $\mathcal{N}$; we ask the reader to think of it as a 'pseudo constant' instead.)

We build ψ from φ such that the overflow checks are incorporated into ψ. To do this, we replace each atomic subformula A of φ by a guarded one A'.

We demonstrate how to do this based on the following example. Let $A(x_1, x_2, y_1, y_2)$ be the atomic formula $\times(+(x_1, y_2), x_2) < +(x_2, y_1)$. We convert A into A' by checking whether an overflow occurs at each subterm bottom-up. A' is the following formula (order of operation is implied by indentation and reading a formula $\varphi \to \alpha \wedge \neg\varphi \to \beta$ as "if φ then α else β").

$$
\begin{aligned}
&c_1 < +(x_1, y_2) \rightarrow && (1)\\
&\quad c_1 < \times(c_0, x_2) \rightarrow && (2)\\
&\quad\quad c_1 < +(x_2, y_1) \rightarrow && (3)\\
&\quad\quad\quad c_0 < c_0 && (4)\\
&\quad\quad \wedge \neg c_1 < +(x_2, y_1) \rightarrow && (5)\\
&\quad\quad\quad c_0 < +(x_2, y_1) && (6)\\
&\quad \wedge \neg c_1 < \times(c_0, x_2) \rightarrow && (7)\\
&\quad\quad c_1 < +(x_2, y_1) \rightarrow && (8)\\
&\quad\quad\quad \times(c_0, x_2) < c_0 && (9)\\
&\quad\quad \wedge \neg c_1 < +(x_2, y_1) \rightarrow && (10)\\
&\quad\quad\quad \times(c_0, x_2) < +(x_2, y_1) && (11)\\
&\wedge \neg c_1 < +(x_1, y_2) \rightarrow && (12)\\
&\quad c_1 < \times(+(x_1, y_2), x_2) \rightarrow && (13)\\
&\quad\quad \vdots && (14)
\end{aligned}
$$

In line (1) it is checked whether an overflow occurs for $+(x_1, y_2)$ (if $x_1 + y_2 > n^c$ holds). In lines (2) to (11) it is assumed that $+(x_1, y_2)$ overflows and therefore it is replaced with c_0. For example, the overflow check for $\times(+(x_1, y_2), x_2)$ becomes $c_1 < \times(c_0, x_2)$ in line 2. In line (13) it is assumed that $+(x_1, y_2)$ does not overflow and thus $+(x_1, y_2)$ is not replaced with c_0.

"$\Leftarrow$": Let $\mathcal{C}$ be a graph class and $S = (\varphi, c)$ is a quantifier-free logical labeling scheme over σ such that $\mathcal{C} \subseteq \mathrm{gr}_\infty(S)$. The maximal value that results from evaluating any term in φ must be polynomially bounded since every term in φ is a polynomial. This means there exists a $d \in \mathbb{N}$ such that the largest value produced while evaluating φ for a graph with n vertices does not exceed n^{cd}. Therefore $\mathrm{gr}_\infty(\varphi, c) \subseteq \mathrm{gr}(\varphi, cd)$ and $\mathcal{C} \in \mathsf{GFO}_{\mathrm{qf}}(\sigma)$ via (φ, cd). $\square$

The following theorem describes the relation between the sets of graph classes defined in terms of logical labeling schemes and the ones defined in terms of classical complexity classes. The label decoder of a quantifier-free logical labeling scheme can be interpreted as a family of boolean circuits by replacing each atomic formula with a circuit that computes it (the size of the circuit depends on the number of vertices n). Quantifiers can be evaluated using non-determinism.

Theorem 3.4. $\mathsf{GFO}_{\mathrm{qf}}(<) \subsetneq \mathsf{GAC}^0$, $\mathsf{GFO}_{\mathrm{qf}} \subsetneq \mathsf{GTC}^0$ *and* $\mathsf{GFO} \subseteq \mathsf{GPH}$.

Proof. First, we show how to convert a logical labeling scheme S into regular labeling schemes S' and S'' such that $\mathrm{gr}(S) \subseteq \mathrm{gr}(S')$ and $\mathrm{gr}_\infty(S) \subseteq \mathrm{gr}(S'')$ (1). Then we argue that if $\mathcal{C}$ is in $\mathsf{GFO}_{\mathrm{qf}}(<)$ / $\mathsf{GFO}_{\mathrm{qf}}$ / GFO via a logical labeling scheme S then the label decoder of S' (or S'') can be computed in AC^0 / TC^0 / PH and therefore the inclusions hold (2).

Strictness of the first two inclusions follows from:

$$\mathsf{GFO}_{\mathrm{qf}}(<) \subseteq \mathsf{GFO}_{\mathrm{qf}} \subseteq \overset{\text{Corol. 4.5}}{\mathsf{PBS}} \subseteq [\mathsf{Factorial} \cap \mathsf{Hereditary}]_\subseteq \overset{\text{Fact 2.6}}{\not\supseteq} \mathsf{GAC}^0 \subseteq \mathsf{GTC}^0$$

(1) Let $S = (\varphi, c)$ be a logical labeling scheme with $2k$ free variables. We define $S' = (F_\varphi, d)$ as follows. There are two aspects that need to be considered when converting φ into F_φ. First, in order to express the overflow conditions in F_φ, the number of vertices n of a graph must be accessible somehow. However, since graphs with different numbers of vertices may receive vertex labels of equal length, n cannot be inferred from the label length alone. For example, the vertices in a graph with 9 vertices and the vertices in a graph with 16 vertices both receive labels whose length

is $d \log 9 = d \log 16 = 4d$ (reminder: by $\log n$ we mean $\lceil \log_2 n \rceil$). We encode n in the first $\log n$ bits of a vertex label. Secondly, a value in $[n^c]_0$ is encoded using $(c+1)\log n$ bits.

Let val: $\{0,1\}^+ \to \mathbb{N}_0$ be the function which maps a binary string (possibly with leading zeros) to its numeric value, e.g. $\mathrm{val}(0) = 0$, $\mathrm{val}(1100) = 12$ and so on. Let $d = 1 + k(c+1)$ and let F_φ be defined as:

$$(x_0 x_1 \dots x_k, y_0 y_1 \dots y_k) \in F_\varphi :\Leftrightarrow (\mathcal{N}_{z^c}, \mathrm{val'}(x_1), \dots, \mathrm{val'}(x_k), \mathrm{val'}(y_1), \dots, \mathrm{val'}(y_k)) \models \varphi$$

for all $x_0, y_0 \in \{0,1\}^m$, $x_i, y_i \in \{0,1\}^{(c+1)m}$, $m \in \mathbb{N}$ and $i \in [k]$ with $z := \mathrm{val}(x_0) + 1$ and $\mathrm{val'}(w) := \min\{z^c, \mathrm{val}(w)\}$.

Let $\mathrm{bin}_k \colon [2^k - 1]_0 \to \{0,1\}^k$ be the bijective function which maps a number between 0 and $2^k - 1$ to its binary representation padded with leading zeros, e.g. $\mathrm{bin}_4(2) = 0010$. Suppose a graph G with n vertices is in $\mathrm{gr}(S)$ via a labeling $\ell \colon V(G) \to [n^c]_0^k$. Then it holds that G is in $\mathrm{gr}(S')$ via the labeling

$$\ell'(v) := \mathrm{bin}_m(n-1)\mathrm{bin}_{(c+1)m}(\ell_1(v)) \dots \mathrm{bin}_{(c+1)m}(\ell_k(v))$$

where $m := \log n$ and ℓ_i is the i-th component of ℓ.

The labeling scheme S'' with $\mathrm{gr}_\infty(S) \subseteq \mathrm{gr}(S'')$ can be defined similarly to S'. The only two differences are that we can drop the first $\log n$ bits of a vertex label used to encode n since there is no need to check for overflows (therefore the label length of S'' is $k(c+1)$) and in the definition of the label decoder of S'' the formula φ is interpreted over $\mathcal{N}$ instead of $\mathcal{N}_{z^c}$.

(2) Suppose $\mathcal{C}$ is in $\mathsf{GFO}_{\mathrm{qf}}$. Due to Lemma 3.3 there exists a quantifier-free logical labeling scheme $S = (\varphi, c)$ such that $\mathcal{C} \subseteq \mathrm{gr}_\infty(S)$. The label decoder of S'' can be computed in TC^0 via the family of circuits that is described by φ itself since order, addition and multiplication can be computed in TC^0 and there is no need to consider overflows.

Similarly, if $\mathcal{C}$ is in $\mathsf{GFO}_{\mathrm{qf}}(<)$ there exists a quantifier-free logical labeling scheme S over $\{<\}$ such that $\mathcal{C} \subseteq \mathrm{gr}_\infty(S)$ due to Lemma 3.3. Since order can be computed in AC^0 it follows that the label decoder of S'' can be computed in AC^0 via the family of circuits described by φ.

Suppose $\mathcal{C}$ is in GFO via a logical labeling scheme $S = (\varphi, c)$. We can assume w.l.o.g. that φ is in prenex normal form. The label decoder of S' can be computed in PH due to the quantifier characterization of PH and the fact that the quantifier-free part of φ can be evaluated in polynomial time.

(To see that this proof is not circular despite its forward reference to Corollary 4.5, the reader can think of this theorem as appearing at the very end of the paper, which is not a problem since it is not used in any other proof.) $\square$

Next, we show that quantifiers do not increase the expressiveness in the absence of addition and multiplication. To do so, we explain how a logical labeling scheme with quantifiers but neither addition nor multiplication can be converted into an equivalent one without quantifiers.

Lemma 3.5. *Let $\sigma \subseteq \{<, +, \times\}$ s.t. $\sigma = \emptyset$ or σ contains '$<$'. $\mathsf{GFO}_{\mathrm{qf}}(\sigma)$ is closed under union.*

Proof. Let $\mathcal{C}, \mathcal{D} \in \mathsf{GFO}_{\mathrm{qf}}(\sigma)$. Due to Lemma 3.3 there exist quantifier-free logical labeling schemes (φ, c) and (ψ, d) over σ such that $\mathcal{C} \subseteq \mathrm{gr}_\infty(\varphi, c)$ and $\mathcal{D} \subseteq \mathrm{gr}_\infty(\psi, d)$. We assume w.l.o.g. that φ and ψ both have $2k$ free variables named $x_1, \dots, x_k, y_1, \dots, y_k$ and $c = d$ (assume $c < d$, then we could choose (φ, d) instead since $\mathrm{gr}_\infty(\varphi, c) \subseteq \mathrm{gr}_\infty(\varphi, d)$). We define a quantifier-free logical labeling scheme (ϕ, c) with $2(k+2)$ free variables over σ:

$$\phi(x_0^a, x_0^b, x_1, \dots, x_k, y_0^a, y_0^b, y_1, \dots, y_k) \triangleq ((x_0^a = x_0^b) \to \varphi) \land ((\neg x_0^a = x_0^b) \to \psi)$$

Assume a graph G with n vertices is in $\mathrm{gr}_\infty(\varphi, c)$ via a labeling $\ell \colon V(G) \to [n^c]_0^k$. Then G is in $\mathrm{gr}_\infty(\phi, c)$ via $\ell'(v) := (0, 0, \ell(v))$ for all $v \in V(G)$. Similarly, for a graph G in $\mathrm{gr}_\infty(\psi, c)$ one can choose $\ell'(v) := (0, 1, \ell(v))$. Therefore $\mathcal{C} \cup \mathcal{D} \subseteq \mathrm{gr}_\infty(\phi, c)$ and thus $\mathcal{C} \cup \mathcal{D} \in \mathsf{GFO}_{\mathrm{qf}}(\sigma)$. $\square$

Fact 3.6. $\mathsf{GFO}_{\mathrm{qf}}(=) = \mathsf{GFO}(=)$.

Proof. Let $\mathcal{C}$ be in $\mathsf{GFO}(=)$ via a labeling scheme $S = (\varphi, c)$ with $2k$ free variables and q quantified variables. We show that there exists a quantifier-free formula ψ with $2k$ free variables which only uses equality such that

$$(\mathcal{N}_{n^c}, \vec{a}) \models \varphi \Leftrightarrow (\mathcal{N}_{n^c}, \vec{a}) \models \psi$$

holds for all $n^c > 2k + q$ and $\vec{a} \in [n^c]_0^{2k}$. This implies that all graphs with more than $\alpha := \sqrt[c]{2k+q}$ vertices in $\mathrm{gr}(S)$ are in $\mathrm{gr}(\psi, c)$ as well and therefore $\mathcal{C}_{>\alpha} \in \mathsf{GFO}_{\mathrm{qf}}(=)$. Since $\mathsf{GFO}_{\mathrm{qf}}(=)$ is closed under union (Lemma 3.5) and contains every singleton graph class, it follows that $\mathcal{C}$ is in $\mathsf{GFO}_{\mathrm{qf}}(=)$.

To prove that for every φ there exists an equivalent quantifier-free ψ, it suffices to prove this for every φ of the form $\exists z \bigwedge_{i=1}^{l} L_i$ where every L_i is a literal and $l \in \mathbb{N}$ (see quantifier-elimination [Smo91, p. 310]). Suppose φ has this form. We assume that φ is neither a tautology nor unsatisfiable, otherwise we can define ψ as $x = x$ for some free variable x or the negation thereof. If z does not occur in any literal then we can simply remove the quantifier, i.e. $\psi \triangleq \bigwedge_{i=1}^{l} L_i$. Therefore we assume z occurs in at least one literal. Assume z occurs in at least one positive literal $L_i \triangleq z = x$ for some free variable x. Then we can obtain ψ by removing the literal L_i and replacing every occurrence of z with x. If z only occurs in negative literals, this means in order to satisfy φ one must assign z a value which, in the worst case, no other variable has. If the universe is sufficiently large ($n > \alpha$) then such a value always exists and therefore we can remove all literals containing z and the existential quantifier. $\square$

Theorem 3.7. $\mathsf{GFO}_{\mathrm{qf}}(<) = \mathsf{GFO}_{\mathrm{qf}}(<,+) = \mathsf{GFO}_{\mathrm{qf}}(<,\times) = \mathsf{GFO}(<)$.

Proof. Obviously, $\mathsf{GFO}_{\mathrm{qf}}(<)$ is a subset of the other three classes since it is more restrictive. We show that $\mathsf{GFO}_{\mathrm{qf}}(<,+)$ and $\mathsf{GFO}_{\mathrm{qf}}(<,\times)$ are subsets of $\mathsf{GFO}_{\mathrm{qf}}(<)$ in Lemma 5.12. Here, we prove that $\mathsf{GFO}(<) \subseteq \mathsf{GFO}_{\mathrm{qf}}(<,+)$ and therefore the theorem holds.

Let 'suc' be the unary function which increments its argument by one if the resulting number does not exceed the maximal element of the universe, otherwise it returns 0. Let $\mathsf{GFO}^{\infty}_{(\mathrm{qf})}(<,\mathrm{suc})$ be the set of graph classes $\mathcal{C}$ for which there exists a (quantifier-free) logical labeling scheme S over $\{<, \mathrm{suc}\}$ and $\mathcal{C} \subseteq \mathrm{gr}_\infty(S)$.

$$\mathsf{GFO}(<) \overset{(1)}{\subseteq} \mathsf{GFO}(<,\mathrm{suc}) \overset{(2)}{\subseteq} \mathsf{GFO}^{\infty}(<,\mathrm{suc}) \overset{(3)}{\subseteq} \mathsf{GFO}^{\infty}_{\mathrm{qf}}(<,\mathrm{suc}) \overset{(4)}{\subseteq} \mathsf{GFO}_{\mathrm{qf}}(<,+)$$

(1) Unlike in the case of $\mathsf{GFO}(=)$, quantifier elimination cannot be applied directly to $\mathsf{GFO}(<)$ since the formula $\exists z\, x < z \wedge z < y$ has no quantifier-free equivalent using only '$<$'. Instead, we consider the fragment that is additionally equipped with 'suc'.

(2) Suppose $\mathcal{C}$ is in $\mathsf{GFO}(<,\mathrm{suc})$ via the logical labeling scheme (φ, c) and φ has $2k$ free and q quantified variables. We assume w.l.o.g. that φ is in prenex normal form, i.e. it has the form $Q_1 z_1 \ldots Q_q z_q\, \psi$ with $Q_i \in \{\exists, \forall\}$ and ψ is quantifier-free. Let $V_{\mathrm{u}}/V_{\mathrm{e}}$ denote the set of universally/existentially quantified variables in φ and let

$$\phi \triangleq Q_1 z_1 \ldots Q_q z_q \left(\bigwedge_{z \in V_{\mathrm{u}}} \neg c_1 < z \right) \rightarrow \left(\psi' \wedge \bigwedge_{z \in V_{\mathrm{e}}} \neg c_1 < z \right)$$

where c_1 is a (pseudo) constant with the value n^c and ψ' is obtained by replacing every atomic subformula in ψ by a guarded one (see the "$\Rightarrow$"-direction in the proof of Lemma 3.3). It holds that

$$(\mathcal{N}_{n^c}, \vec{a}) \models \varphi \Leftrightarrow (\mathcal{N}, \vec{a}) \models \phi$$

for all $n \in \mathbb{N}$ and $\vec{a} \in [n^c]_0^{2k}$ and therefore $\mathcal{C}$ is in $\mathsf{GFO}^{\infty}(<,\mathrm{suc})$ via (ϕ, c). To see why this is the case, consider the game-theoretical semantics of first-order logic.

"⇒": Suppose Eloise (the verifier) has a winning strategy for the left-hand side. We argue that Eloise can also use this as a winning strategy for the right-hand side. If Abelard (the falsifier) chooses a value larger than n^c, Eloise immediately wins since the premise in ϕ becomes false. Thus, Abelard must only choose values $\leq n^c$. Eloise chooses only values $\leq n^c$ because the winning strategy for the left-hand side cannot use larger values. If only values $\leq n^c$ are 'assigned' to the quantified variables then φ and ϕ behave identically and therefore Eloise wins.

"⇐": Suppose Eloise has a winning strategy for the right-hand side. Eloise will never choose a value larger than n^c in this strategy since then Abelard could easily win because the conclusion in ϕ becomes false. Therefore the same strategy can be played for the left-hand side.

(3) Suppose $\mathcal{C}$ is in $\mathsf{GFO}^\infty(<, \mathrm{suc})$ via (φ, c) and φ has $2k$ free variables. Due to Lemma 3.8 there exist a quantifier-free formula ψ over $\{0, <, \mathrm{suc}\}$ with $2k$ free variables such that

$$(\mathcal{N}, \vec{a}) \models \varphi \Leftrightarrow (\mathcal{N}, \vec{a}) \models \psi$$

holds for all $\vec{a} \in \mathbb{N}_0^{2k}$. We can modify the labeling scheme (ψ, c) by adding an additional variable to each vertex which is promised to receive the value 0 in every labeling and use it to replace the constant 0. It follows that $\mathcal{C}$ is in $\mathsf{GFO}_{\mathrm{qf}}^\infty(<, \mathrm{suc})$ via (ψ, c).

(4) The expression $\mathrm{suc}(x)$ can be simulated by $x+1$ and the constant 1 can be realized by adding an additional variable to each vertex. To prevent overflow when translating a labeling scheme (φ, c) in $\mathsf{GFO}_{\mathrm{qf}}^\infty(<, \mathrm{suc})$ to $\mathsf{GFO}_{\mathrm{qf}}(<, +)$, choose $c + d$ as second component of the new logical labeling scheme where d is the number of times 'suc' appears in φ.

(To see that this proof is not circular despite its forward reference to Lemma 3.8 and 5.12, the reader can think of this theorem as appearing right before Theorem 5.16 since it is not used in any other proof before that.) □

Lemma 3.8. *For every formula φ over the signature $\{0, <, \mathrm{suc}\}$ there exists an equivalent quantifier-free formula ψ over the same signature, i.e. $(\mathcal{N}, \vec{a}) \models \varphi \Leftrightarrow (\mathcal{N}, \vec{a}) \models \psi$ holds for all $\vec{a} \in \mathbb{N}_0^k$ where k is the number of free variables in φ.*

Proof. To prove that for every φ there exists an equivalent quantifier-free ψ, it suffices to prove this for every φ of the form $\exists z \bigwedge_{i=1}^{l} L_i$ where every L_i is a literal and $l \in \mathbb{N}$ (see quantifier-elimination [Smo91, p. 310]). There are the following 4 types of literals:

1. $x + i < y + j$
2. $x + i = y + j$
3. $x + i \leq y + j$ (negation of $<$)
4. $x + i \neq y + j$ (negation of $=$)

where x, y are variables or the constant 0 and $i, j \in \mathbb{N}_0$ (note: $x + i$ means suc is applied i times to x). First, we argue why it suffices to consider only literals of the first two types (positive literals). The idea is to rewrite literals of type 3. and 4. using disjunction ($a \leq b \Leftrightarrow a < b \vee a = b$ and $a \neq b \Leftrightarrow a < b \vee b < a$), then rearrange the formula φ such that it becomes a disjunction of conjunctions and draw the existential quantifier inside the disjunction:

$$\varphi \equiv \exists z \bigvee_j \bigwedge_i L_i^j \equiv \bigvee_j \exists z \bigwedge_i L_i^j$$

where L_i^j are appropriately chosen positive literals. Therefore it suffices to rewrite $\exists z \bigwedge_i L_i^j$ into an equivalent quantifier-free formula for every j.

Due to the previous paragraph we can assume w.l.o.g. that $\varphi \triangleq \exists z \bigwedge_{i=1}^{l} L_i$ where every L_i is a positive literal. Next, we explain how to convert φ into an equivalent quantifier-free formula ψ. We assume that φ is neither unsatisfiable nor a tautology, otherwise it is trivial to write an equivalent quantifier-free formula. Moreover, we assume that no literal contains the same variable more than once, e.g. $x < x + 2$ does not occur. We assume that z occurs in at least one literal since otherwise we can simply remove the part '$\exists z$' from φ to obtain ψ. We distinguish the following two cases.

Case 1: there exists a literal $z + i = x + j$ for some $i, j \in \mathbb{N}_0$. In this case replace z with $x + j - i$ in every literal, rearrange each literal so that it contains no negative term and then remove the existential quantifier to obtain ψ. For example, a literal $z + q < y + p$ would become $x + j - i + q < y + p$ and then $x + j + q < y + p + i$.

Case 2: z only occurs in literals with '$<$'. Let X_{lt} denote the set that consists of all pairs (x, k) such that x is a free variable or the constant 0, $k \in \mathbb{Z}$ and there exists a literal $x + i < z + j$ with $k = j - i$ in φ. Analogously, let X_{gt} denote the set that consists of all pairs (x, k) such that there exists a literal $z + j < x + i$ with $k = j - i$. Observe that a literal corresponding to the pair $(x, k) \in X_{\text{lt}}$ is satisfied iff $x < z + k$ and a literal corresponding to $(x, k) \in X_{\text{gt}}$ is satisfied iff $z + k < x$. If X_{gt} is empty then we can simply remove every literal containing z from φ to obtain ψ because there always exists a sufficiently large value for z that satisfies all constraints implied by X_{lt}. Thus, we assume that X_{gt} is non-empty. If X_{lt} is empty then z can be replaced with the constant 0. Therefore we assume X_{lt} is non-empty as well. We define ψ as conjunction of the literals in the sets $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_3$:

- $\mathcal{L}_1 :=$ set of literals in φ that do not contain z
- $\mathcal{L}_2 :=$ literal equivalent to $y - m < x - k - 1$ for each $(x, k) \in X_{\text{gt}}$ and $(y, m) \in X_{\text{lt}}$
- $\mathcal{L}_3 :=$ literal equivalent to $k < x$ for each $(x, k) \in X_{\text{gt}}$

It remains to argue why φ and ψ are equivalent, i.e. for all $\vec{a} \in \mathbb{N}_0^k$ it holds that

$$(\mathcal{N}, \vec{a}) \models \varphi \Leftrightarrow (\mathcal{N}, \vec{a}) \models \psi$$

"$\Rightarrow$": Let $\vec{a} \in \mathbb{N}_0^k$ and let $(\mathcal{N}, \vec{a}) \models \varphi$. We need to argue that all literals in $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_3$ are satisfied. This implies $(\mathcal{N}, \vec{a}) \models \psi$. For $\mathcal{L}_1$ this holds because all its literals occur in φ as well. Let L be a literal in $\mathcal{L}_2$ via $(x, k) \in X_{\text{gt}}$ and $(y, m) \in X_{\text{lt}}$. This means L is equivalent to $y - m < x - k - 1$. The pair (x, k) implies $z + k < x$ and the pair (y, m) implies $y < z + m$ must hold in φ with respect to the assignment $\vec{a}$. This means $z < x - k$ and $y - m < z$ and therefore $y - m < z < x - k$, which implies $y - m < x - k - 1$. Let L be a literal in $\mathcal{L}_3$ via $(x, k) \in X_{\text{gt}}$. The pair (x, k) implies $z + k < x$ and therefore $k < x$ since $z \geq 0$.

"$\Leftarrow$": Let $\vec{a} \in \mathbb{N}_0^k$ and let $(\mathcal{N}, \vec{a}) \models \psi$. We argue that there exists a $b \in \mathbb{N}_0$ such that $(\mathcal{N}, \vec{a}) \models \varphi$ where z is assigned the value b. We define b as minimum over $\{a(x) - k - 1 \mid (x, k) \in X_{\text{gt}}\}$ where $a(x)$ denotes the value assigned to variable x in $\vec{a}$. It holds that $b \geq 0$: assume that this is not the case, i.e. $b < 0$. This would imply that there exists an $(x, k) \in X_{\text{gt}}$ such that $a(x) - k - 1 < 0$, which is equivalent to $a(x) \leq k$. Since $(\mathcal{N}, \vec{a})$ models ψ, the literal in $\mathcal{L}_3$ for $(x, k) \in X_{\text{gt}}$ implies $k < a(x)$, contradiction.

All literals of φ not containing z are satisfied due to $\mathcal{L}_1$. Each literal containing z in φ corresponds to either an element in X_{gt} or X_{lt}. Let $(x, k) \in X_{\text{gt}}$. This means $z + k < x \Leftrightarrow z \leq x - k - 1$ and therefore $b \leq a(x) - k - 1$. Our choice of b satisfies this. Let $(y, m) \in X_{\text{lt}}$. This means $y < z + m$ resp. $a(y) < b + m$ must hold. Due to $\mathcal{L}_2$ it holds that $a(y) - m < a(x) - k - 1$ for every $(x, k) \in X_{\text{gt}}$. Since $b = a(x) - k - 1$ for some (x, k) it follows that the literal for $(y, m) \in X_{\text{lt}}$ in φ is satisfied. $\square$

Lastly, we show how equality-based labeling schemes can be expressed as quantifier-free labeling schemes over $\{=\}$ and vice versa. As a consequence, the result that interval graphs have no equality-based labeling scheme from [HWZ22] implies that $\mathsf{GFO}(=)$ is a strict subset of $\mathsf{GFO}(<)$

Definition 3.9 ([HWZ22, Def. 2.4])**.** A graph class $\mathcal{C}$ is said to have a constant-size equality-based labeling scheme (EBLS) if there exists $s, k \in \mathbb{N}$ and a set of functions $D_{p_1,p_2} \colon \{0,1\}^{k \times k} \to \{0,1\}$ for each $p_1, p_2 \in \{0,1\}^s$ with the following property. For every graph $G \in \mathcal{C}$ there exists a labeling $\ell \colon V(G) \to \{0,1\}^s \times \mathbb{N}^k$ such that for all $u \neq v \in V(G)$ it holds that $(u,v) \in E(G)$ iff $D_{p_u,p_v}(Q_{u,v}) = 1$ with $\ell(u) = (p_u, u_1, \ldots, u_k), \ell(v) = (p_v, v_1, \ldots, v_k)$ and $Q_{u,v} \in \{0,1\}^{k \times k}$ is defined as $Q_{u,v}(i,j) = 1$ iff $u_i = v_j$. The s bit long string of the labeling is called the prefix of a vertex.

Theorem 3.10 ([HWZ22])**.** *Interval graphs and permutation graphs do not have constant-size EBLS.*

Proof. Any graph class with unbounded chain number ([HWZ22, Def. 1.10]) does not have a constant-size EBLS ([HWZ22, Prop. 2.8]); stable = bounded chain number. Interval graphs and permutation graphs have unbounded chain number. □

Lemma 3.11. *A graph class has a constant-size EBLS iff it is in* $\mathsf{GFO}_{\mathrm{qf}}(=)$*.*

Proof. "⇒": Let $\mathcal{C}$ be a graph class with an EBLS via $s, k \in \mathbb{N}$ and functions D_{p_1,p_2} for each $p_1, p_2 \in \{0,1\}^s$. Let $k' = 1 + s + k$. A logical labeling scheme (φ, k') with $2k'$ variables can be constructed which shows that $\mathcal{C}$ is in $\mathsf{GFO}_{\mathrm{qf}}(=)$. Suppose a graph G with n vertices is in $\mathcal{C}$ and thus representable by the EBLS via a labeling $\ell \colon V(G) \to \{0,1\}^s \times \mathbb{N}^k$. We can assume w.l.o.g. that the image of ℓ—more specifically the $\mathbb{N}^k$ part—contains only numbers from $[kn]$ since at most k different numbers can be picked per vertex and the magnitude of the numbers is irrelevant. Let $\ell' \colon V(G) \to [n^{k'}]_0^{k'}$ be defined as $\ell'(u) = (0, p_u, u_1, \ldots, u_k)$ for all $u \in V(G)$ with $\ell(u) = (p_u, u_1, \ldots, u_k)$. The formula φ can be defined such that it represents G via this labeling ℓ'. For example, for two vertices u, v the i-th bit of the prefix of u (or v) is 0 iff $x_0 = x_i$ (resp. $y_0 = y_i$), assuming the variables in φ are named $x_0, \ldots, x_{k'-1}, y_0, \ldots, y_{k'-1}$.

"⇐": Let $\mathcal{C}$ be in $\mathsf{GFO}_{\mathrm{qf}}(=)$ via a logical labeling scheme (φ, c) and φ has $2k$ variables $x_1, \ldots, x_k, y_1, \ldots, y_k$. We classify an atomic formula in φ as cross-comparison if it is of the form $x_i = y_j$ and as self-comparison if it is of the form $x_i = x_j$ or $y_i = y_j$ for some $i, j \in [k]$. It can be assumed w.l.o.g. that every variable either occurs only in cross-comparisons or only in self-comparisons (this can always be achieved by adding new variables). The self-comparisons correspond to the prefixes in an EBLS. To construct an EBLS for $\mathcal{C}$, replace all self-comparisons in φ with all combinations of true and false. This yields the different functions D_{p_1,p_2}. For the parameter s the maximum over s_x and s_y can be chosen where s_x (s_y) is the number of distinct self-comparisons with x_i's (resp. y_i's). □

Corollary 3.12. $\mathsf{GFO}(=) \subsetneq \mathsf{GFO}_{\mathrm{qf}}(<)$*.*

Proof. Interval graphs are in $\mathsf{GFO}_{\mathrm{qf}}(<)$ but not in $\mathsf{GFO}(=)$ due to Theorem 3.10, Lemma 3.11 and Fact 3.6. □

4 Polynomial-Boolean Systems

Line segment graphs, disk graphs and k-dot product graphs are factorial and hereditary but not known to have a labeling scheme. All three share in common that they can be defined as the set of induced subgraphs of some infinite graph $\mathcal{H}$ with vertex set $\mathbb{R}^k$ and two vertices in $\mathcal{H}$ are adjacent iff they satisfy a certain combination of polynomial (in)equations over $2k$ variables for some $k \in \mathbb{N}$. Given a graph G, a mapping $\ell \colon V(G) \to \mathbb{R}^k$ showing that G is an induced subgraph of $\mathcal{H}$ is called a realization of G. Graph classes that can be described in such a way are known as semi-algebraic graph classes (see e.g. [Fox+14]).

It can be shown that it suffices to use rationals instead of reals to define the three aforementioned graph classes by a perturbation argument, i.e. $V(\mathcal{H}) = \mathbb{Q}^k$; let us call such semi-algebraic graph classes rational. A natural question that arises is how many bits are required to represent each rational in some realization of a graph with n vertices from such a class. McDiarmid and Müller have shown that line segment and disk graphs require at least $2^{\Omega(n)}$ bits and that this also suffices for every such graph, i.e. the bound is tight [MM13]. Kang and Müller have shown that the same (upper and lower) bound holds for k-dot product graphs [KM12]. Therefore the labeling schemes induced by the definitions of these graph classes do not represent them since they allow only $\mathcal{O}(\log n)$ bits per rational. It is unclear whether every semi-algebraic graph class is rational. On a related note, we show that natural numbers suffice to represent rational semi-algebraic graph classes (Lemma 4.3).

This way of defining graph classes can be formalized by polynomial-boolean systems (PBS). Such a system consists of a sequence of $2k$-variate polynomials that are compared with each other and a boolean function that determines adjacency from these comparisons. Any graph class that can be defined in terms of a PBS is factorial and hereditary; however, not every factorial, hereditary graph class is semi-algebraic (Lemma 4.2). Therefore it provides a source of candidates for graph classes that could have a labeling scheme. It is not difficult to see that PBS are isomorphic to quantifier-free logical labeling schemes (Lemma 4.4). Every rational semi-algebraic graph class can be expressed as the hereditary closure of some graph class in $\mathsf{GFO}_{\mathrm{qf}}$ (Fact 4.6). This yields the following amplification result: if $\mathsf{GFO}(<) = \mathsf{GFO}_{\mathrm{qf}}$ then $\mathsf{GFO}(<)$ equals the set of rational semi-algebraic graph classes (Corollary 4.8).

In our context, we consider a polynomial to be a function that can be defined as an expression consisting of variables, addition and multiplication. This implies that the coefficients of a polynomial must be natural. This does not affect the set of graph classes that be represented by a PBS since coefficients outside of $\mathbb{N}_0$ can be mimicked by adding additional variables.

Definition 4.1. Let $\mathbb{X} \in \{\mathbb{N}_0, \mathbb{Z}, \mathbb{Q}, \mathbb{R}\}$. A polynomial-boolean system (PBS) is a tuple (P, f) where P is a sequence of q polynomials with signature $\mathbb{X}^{2k} \to \mathbb{X}$ and f is a q^2-ary boolean function for some $k, q \in \mathbb{N}$. We define $\mathrm{gr}(P, f)$ as the following set of graphs. A graph G with n vertices is in $\mathrm{gr}(P, f)$ iff there exists a labeling $\ell \colon V(G) \to \mathbb{X}^k$ such that for all $u \neq v \in V(G)$ it holds that

$$(u, v) \in E(G) \Leftrightarrow f(x_{1,1}, \ldots, x_{q,q}) = 1$$

where $x_{i,j} := [\![p_i(\ell(u), \ell(v)) < p_j(\ell(u), \ell(v))]\!]$ for $i, j \in [q]$ and p_i denotes the i-th polynomial in the sequence P.

A graph class $\mathcal{C}$ is in $\mathsf{PBS}(\mathbb{X})$ if there exists a PBS (P, f) with polynomials over $\mathbb{X}$ such that $\mathcal{C} \subseteq \mathrm{gr}(P, f)$. If $\mathbb{X} = \mathbb{N}_0$, we also write PBS instead of $\mathsf{PBS}(\mathbb{N}_0)$.

A graph class is called semi-algebraic if it is in $\mathsf{PBS}(\mathbb{R})$. A semi-algebraic graph class is called rational if it is in $\mathsf{PBS}(\mathbb{Q})$. It is easy to see that $\mathsf{PBS}(\mathbb{N}_0) \subseteq \mathsf{PBS}(\mathbb{Z}) \subseteq \mathsf{PBS}(\mathbb{Q}) \subseteq \mathsf{PBS}(\mathbb{R})$. Line segment graphs, disk graphs and k-dot product graphs are in $\mathsf{PBS}(\mathbb{Q})$ since their definitions can be expressed as PBS.

Lemma 4.2. $\mathsf{PBS}(\mathbb{R}) \subsetneq [\mathsf{Factorial} \cap \mathsf{Hereditary}]_{\subseteq}$.

Proof. It follows from Warren's theorem that semi-algebraic graph classes are at most factorial (see [Spi03, p. 54]). Therefore they are a subset of $[\mathsf{Factorial} \cap \mathsf{Hereditary}]_{\subseteq}$. The number of polynomial-boolean systems is countable since each polynomial and boolean function can be described by a finite string (for polynomials see the remark before Definition 4.1). From Lemma 2.4 it follows that there exists a factorial, hereditary graph class that is not semi-algebraic. □

The following theorem shows that choosing between polynomials over $\mathbb{N}_0$, $\mathbb{Z}$ or $\mathbb{Q}$ does not make a difference, i.e. they all lead to the same set of graphs classes. Therefore we simply write PBS to denote the set of rational semi-algebraic graph classes in the following.

Theorem 4.3. $\mathsf{PBS} := \mathsf{PBS}(\mathbb{N}_0) = \mathsf{PBS}(\mathbb{Z}) = \mathsf{PBS}(\mathbb{Q})$.

Proof. We argue that $\mathsf{PBS}(\mathbb{Q}) \subseteq \mathsf{PBS}(\mathbb{N}_0)$ in two steps. First, we show that $\mathsf{PBS}(\mathbb{Q}) \subseteq \mathsf{PBS}(\mathbb{Q}_+)$ where $\mathbb{Q}_+ = \{x \in \mathbb{Q} \mid x \geq 0\}$ (1). Secondly, we argue why $\mathsf{PBS}(\mathbb{Q}_+) \subseteq \mathsf{PBS}(\mathbb{N}_0)$ (2).

(1) Let $\mathcal{C} \in \mathsf{PBS}(\mathbb{Q})$ via a PBS (P, f) where P is a sequence of q $2k$-ary polynomials over $\mathbb{Q}$. We outline a PBS (P', f') over $\mathbb{Q}_+$ which shows that $\mathcal{C}$ is in $\mathsf{PBS}(\mathbb{Q}_+)$. This construction relies on the following observation. Given $a \in \mathbb{Q}$ let $|a|$ denote its absolute value and $\mathrm{sign}(a)$ equals -1 if a is negative and 1 otherwise. For $n \in \mathbb{N}$ and $\vec{a} \in \mathbb{Q}^n$ let $|\vec{a}| = (|a_1|, \ldots, |a_n|)$ and $\mathrm{sign}(\vec{a}) = (\mathrm{sign}(a_1), \ldots, \mathrm{sign}(a_n))$. For all polynomials $p, q \colon \mathbb{Q}^n \to \mathbb{Q}$ and sign patterns $\vec{s} \in \{-1, 1\}^n$ there exist polynomials $p', q' \colon \mathbb{Q}_+^n \to \mathbb{Q}_+$ such that for all $\vec{a} \in \mathbb{Q}^n$ with $\mathrm{sign}(\vec{a}) = \vec{s}$ it holds that $p(\vec{a}) < q(\vec{a})$ iff $p'(|\vec{a}|) < q'(|\vec{a}|)$. For example, consider the polynomials $p(x, y, z) = x^2y^3z + y$ and $q(x, y, z) = z$ and the sign pattern $(-1, 1, -1)$ for (x, y, z). If we only consider inputs with this sign pattern then it holds that $p(x, y, z) < q(x, y, z)$ iff $\underbrace{|y| + |z|}_{p'} < \underbrace{|x|^2|y|^3|z|}_{q'}$.

For each variable in (P, f) we have two variables in the new PBS (P', f'). The first one is used to store the absolute value of the original variable and the second one encodes the sign. Let G be a graph that is in $\mathrm{gr}(P, f)$ via a labeling $\ell \colon V(G) \to \mathbb{Q}^k$. We derive the following labeling $\ell' \colon V(G) \to \mathbb{Q}_+^{2k}$ from ℓ. Given $u \in V(G)$ let $\ell(u) = (u_1, \ldots, u_k)$. We set $\ell'(u) = (|u_1|, u'_1, \ldots, |u_k|, u'_k)$ where $u'_i = |u_i|$ if u_i is negative and any other non-negative value if u_i is positive. This allows us to infer the sign pattern and absolute values of the original labeling ℓ from ℓ'.

The PBS (P', f') is constructed such that $G \in \mathrm{gr}(P', f')$ via ℓ'. The adjacency of two vertices u and v depends on the results of $p_i(\ell(u), \ell(v)) < p_j(\ell(u), \ell(v))$ for $i, j \in [q]$. The result of these inequations is determined by checking $p'(|\ell(u)|, |\ell(v)|) < q'(|\ell(u)|, |\ell(v)|)$ in (P', f') where p' and q' depend on p_i, p_j and the sign pattern of $\ell(u), \ell(v)$. This means for every pair $i, j \in [q]$ and every sign pattern $s \in \{-1, 1\}^{2k}$ there is a pair of polynomials in P' and additionally P' has the polynomials $p(x_1, \ldots, x_{4k}) = x_i$ for $i \in [4k]$ to decode the signs.

(2) To see that $\mathsf{PBS}(\mathbb{Q}_+) \subseteq \mathsf{PBS}(\mathbb{N}_0)$ it suffices to make the following observation. For all polynomials $p, q \colon \mathbb{Q}_+^k \to \mathbb{Q}_+$ there exist polynomials $p', q' \colon \mathbb{N}_0^{2k} \to \mathbb{N}_0$ such that for all $\vec{a} = (\frac{a_1}{b_1}, \ldots, \frac{a_k}{b_k}) \in \mathbb{Q}_+^k$ it holds that $p(\vec{a}) < q(\vec{a})$ iff $p'(a_1, b_1, \ldots, a_k, b_k) < q'(a_1, b_1, \ldots, a_k, b_k)$. The functions p' and q' can be obtained from the inequation $p < q$ by multiplying with the denominators. Therefore a PBS (P, f) over $\mathbb{Q}_+$ with $2k$ variables can be translated into a PBS (P', f') over $\mathbb{N}_0$ with $4k$ variables such that $\mathrm{gr}(P, f) \subseteq \mathrm{gr}(P', f')$. ◻

Lemma 4.4. *A graph class $\mathcal{C}$ is in* PBS *iff there exists a quantifier-free logical labeling scheme S such that $\mathcal{C} \subseteq gr_{\mathrm{p}}(S)$.*

Proof. "⇒": Let (P, f) be a PBS where P is a sequence of q polynomials over $\mathbb{N}_0$. The PBS (P, f) can be directly encoded as quantifier-free logical labeling scheme $S = (\varphi, 1)$. Each of the q^2 inequations of (P, f) is an atomic formula in φ and the propositional part of φ must represent the boolean function f. It follows that $\mathrm{gr}(P, f) \subseteq \mathrm{gr}_{\mathrm{p}}(S)$.

"⇐": Let $S = (\varphi, c)$ be a quantifier-free logical labeling scheme. The value c is irrelevant since it does not affect $\mathrm{gr}_{\mathrm{p}}(S)$. Every atomic formula in φ is of the form $p < q$ or $p = q$ where p and q are expressions over addition and multiplication and therefore represent polynomials. Choose these as sequence of polynomials P and define f in terms of the boolean formula that is obtained by replacing every atomic formula in φ with a propositional variable. It follows that $\mathrm{gr}_{\mathrm{p}}(S) \subseteq \mathrm{gr}(P, f)$. ◻

Corollary 4.5. $\mathsf{GFO}_{\mathrm{qf}} \subseteq \mathsf{PBS} \subseteq \mathsf{PBS}(\mathbb{R}) \subsetneq [\mathsf{Factorial} \cap \mathsf{Hereditary}]_{\subseteq}$.

Proof. The inclusion $\mathsf{GFO}_{\mathrm{qf}} \subseteq \mathsf{PBS}$ holds for the following reason. Suppose $\mathcal{C}$ is in $\mathsf{GFO}_{\mathrm{qf}}$. Due to Lemma 3.3 there exists a quantifier-free logical labeling scheme S such that $\mathcal{C} \subseteq \mathrm{gr}_{\infty}(S)$. From that and Lemma 4.4 it directly follows that $\mathcal{C}$ is in PBS since $\mathrm{gr}_{\infty}(S) \subseteq \mathrm{gr}_{\mathrm{p}}(S)$. The third inclusion is Lemma 4.2. ◻

One can also characterize PBS as the set of graph classes that occur as subset of the hereditary closure of some graph class in $\mathsf{GFO}_{\mathrm{qf}}$ since the hereditary closure enables one to sidestep the size limitation of the labeling by choosing a sufficiently large graph to increase the maximal value allowed in the labeling and then taking the relevant subgraph. An interesting consequence of this is that $\mathsf{GFO}_{\mathrm{qf}} = \mathsf{PBS}$ if $\mathsf{GFO}_{\mathrm{qf}}$ is closed under hereditary closure.

Fact 4.6. *A graph class $\mathcal{C}$ is in PBS iff there exists a graph class $\mathcal{D}$ in $\mathsf{GFO}_{\mathrm{qf}}$ such that $\mathcal{C} \subseteq [\mathcal{D}]_{\mathrm{hc}}$.*

Proof. "$\Rightarrow$": Let $\mathcal{C} \in \mathsf{PBS}$. Due to Lemma 4.4 there exists a quantifier-free logical labeling scheme $(\varphi, 1)$ such that $\mathcal{C} \subseteq \mathrm{gr}_{\mathrm{p}}(\varphi, 1)$. We show that every graph in $\mathcal{C}$ occurs as induced subgraph of some graph in $\mathrm{gr}_{\infty}(\varphi, 1)$ and $\mathrm{gr}_{\infty}(\varphi, 1)$ is in $\mathsf{GFO}_{\mathrm{qf}}$ due to Lemma 3.3.

Let $G \in \mathcal{C}$. This means $G \in \mathrm{gr}_{\mathrm{p}}(\varphi, 1)$ via some labeling $\ell\colon V(G) \to \mathbb{N}_0^k$. Let r be the maximal value in the image of ℓ. Let H be a graph with r vertices whose vertex set is a superset of $V(G)$ and that is in $\mathrm{gr}_{\infty}(\varphi, 1)$ via the labeling $\ell'\colon V(H) \to [r]_0^k$ with $\ell'(v) = \ell(v)$ if $v \in V(G)$ and $(0, \ldots, 0)$ otherwise. Clearly, G is an induced subgraph of H.

"$\Leftarrow$": Let $\mathcal{C}$ and $\mathcal{D}$ be graph classes such that $\mathcal{D} \in \mathsf{GFO}_{\mathrm{qf}}$ and $\mathcal{C} \subseteq [\mathcal{D}]_{\mathrm{hc}}$. Since $\mathsf{GFO}_{\mathrm{qf}} \subseteq \mathsf{PBS}$ (Corollary 4.5) it follows that $\mathcal{D} \in \mathsf{PBS}$. And since PBS is trivially closed under hereditary closure and subsets it follows that $[\mathcal{D}]_{\mathrm{hc}}$ and therefore $\mathcal{C}$ is in PBS. $\square$

Lemma 4.7. $\mathsf{GFO}_{\mathrm{qf}}(<)$ *is closed under hereditary closure.*

Proof. We need to show that for every graph class $\mathcal{C} \in \mathsf{GFO}_{\mathrm{qf}}(<)$ its hereditary closure $[\mathcal{C}]_{\mathrm{hc}}$ is in $\mathsf{GFO}_{\mathrm{qf}}(<)$. Let $\mathcal{C}$ be in $\mathsf{GFO}_{\mathrm{qf}}(<)$ via a logical labeling scheme (φ, c) and φ has $2k$ free variables. We show that $[\mathcal{C}]_{\mathrm{hc}} \subseteq \mathrm{gr}(\varphi, k)$ and therefore $[\mathcal{C}]_{\mathrm{hc}}$ is in $\mathsf{GFO}_{\mathrm{qf}}(<)$.

Let G be a graph in $\mathcal{C}$ with n vertices. Since $G \in \mathcal{C}$ there exists a labeling $\ell\colon V(G) \to [n^c]_0^k$ which witnesses that G is in $\mathrm{gr}(\varphi, c)$. We convert ℓ into a 'normalized' labeling ℓ_0 such that the maximal value in the image of ℓ_0 is at most kn. Let $\{x_1, \ldots, x_r\}$ denote the subset of numbers from $[n^c]_0$ that occur in the image of ℓ, i.e. for every $i \in [r]$ there exists a $v \in V(G)$ and $j \in [k]$ such that $x_i = \ell_j(v)$. Assume the x_i's are ordered, i.e. $x_1 < x_2 < \cdots < x_r$. Replace the numbers in the image of ℓ with their index minus one, i.e. x_i becomes $i - 1$ and call the new labeling ℓ_0. Observe that ℓ_0 is a correct labeling for G since the order relation is maintained by the renumbering, i.e. $x_i < x_j$ iff $i - 1 < j - 1$. Moreover, the image of ℓ can contain at most kn different values ($r \leq kn$) which limits the maximal value in the image of ℓ_0.

Let H be an induced subgraph of G with $m > 1$ vertices. Take the labeling ℓ for G, restrict it to the vertices in H and normalize it as described above. The restricted labeling contains at most km different values and therefore the maximal value in the image of the normalized labeling is at most km, which does not exceed m^k. Thus, it witnesses that H is in $\mathrm{gr}(\varphi, k)$. $\square$

Corollary 4.8. *If* $\mathsf{GFO}_{\mathrm{qf}}(<) = \mathsf{GFO}_{\mathrm{qf}}$ *then* $\mathsf{GFO}_{\mathrm{qf}}(<) = \mathsf{PBS}$.

Proof. Let $\mathcal{C} \in \mathsf{PBS}$. From Fact 4.6 it follows that there exists a graph class $\mathcal{D} \in \mathsf{GFO}_{\mathrm{qf}}$ such that $\mathcal{C} \subseteq [\mathcal{D}]_{\mathrm{hc}}$. Assuming $\mathsf{GFO}_{\mathrm{qf}}(<) = \mathsf{GFO}_{\mathrm{qf}}$, this means $\mathcal{D}$ is in $\mathsf{GFO}_{\mathrm{qf}}(<)$ and therefore $[\mathcal{D}]_{\mathrm{hc}}$ is in it as well due to Lemma 4.7. Due to closure under subsets it follows that $\mathcal{C}$ is in $\mathsf{GFO}_{\mathrm{qf}}(<)$. $\square$

Therefore in order to prove that $\mathsf{GFO}_{\mathrm{qf}}(<) \neq \mathsf{GFO}_{\mathrm{qf}}$ it suffices to show that $\mathsf{GFO}_{\mathrm{qf}}(<) \neq \mathsf{PBS}$.

5 Algebraic Reductions

Consider the relation between interval and box graphs. Every box graph can be expressed by intersecting the edge relation of two interval graphs as depicted in Figure 2 since every box can be represented by two intervals. Also, every planar graph can be expressed by taking the union of the edge relation of three forests since planar graphs have arboricity at most 3. Additionally, every

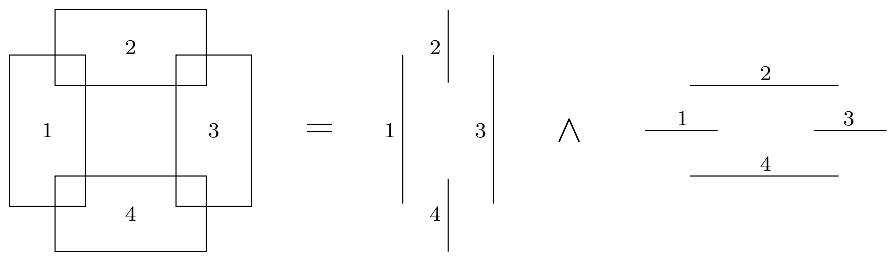

Figure 2: Box graph as conjunction of two interval graphs

forest is a box graph. It follows that every planar graph can be expressed in terms of 6 interval graphs as $\bigvee_{i=1}^{3}$ Interval $\wedge$ Interval. One could say that the adjacency structure of planar graphs is not more complex than that of interval graphs in a sense since the former can be expressed as boolean combination of the latter.

We would like to relate the difficulty of finding a labeling scheme with a label decoder of a particular complexity for one graph class to another. For instance, saying that $\mathcal{C}$ reduces to $\mathcal{D}$ ($\mathcal{C} \leq \mathcal{D}$) should mean that a labeling scheme for $\mathcal{D}$ can be translated to a labeling scheme for $\mathcal{C}$ with the same complexity. The crucial property required of such a reduction notion is that the different sets of graph classes that we consider must be closed under it, i.e. $\mathcal{C} \leq \mathcal{D}$ and $\mathcal{D} \in \mathsf{GA}$ implies $\mathcal{C} \in \mathsf{GA}$. It can also be used to relate graph classes without labeling schemes. For instance, if one could reduce two graph classes not known to have labeling schemes to each other, this would imply that there is a common obstacle that makes finding a labeling scheme for them difficult.

This section is structured as follows. We define an interpretation of conjunction, disjunction and negation on graph classes (Definition 5.1) and show that if two boolean formulas represent the same boolean function then their interpretation over graph classes coincides as well, provided every variable occurs at most once in each formula (Corollary 5.4). Definition 5.5 formalizes algebraic reductions $\leq_{\mathrm{BF}}$. All sets of graph classes considered here are closed under $\leq_{\mathrm{BF}}$-reductions (Fact 5.7 & 5.8). We call a graph class $\mathcal{C}$ $\leq_{\mathrm{BF}}$-complete for a set of graph classes $\mathbb{X}$ if $\mathcal{C}$ is in $\mathbb{X}$ (membership) and every graph class in $\mathbb{X}$ reduces to $\mathcal{C}$ (hardness). This means the adjacency structure of every graph class in $\mathbb{X}$ can be expressed as boolean combination of graphs from $\mathcal{C}$. Therefore we are interested in determining which sets of graph classes have a complete graph class.

The set $[\mathsf{Factorial} \cap \mathsf{Hereditary}]_{\subseteq}$ has no $\leq_{\mathrm{BF}}$-complete graph class and GAC^0 has no hereditary $\leq_{\mathrm{BF}}$-complete graph class (Fact 5.9 & 5.10). On the other hand, we show that so called dichotomic and linear neighborhood graphs are complete for $\mathsf{GFO}(=)$ and $\mathsf{GFO}(<)$ (Theorem 5.14 & 5.16). These completeness results follow from the tight correspondence between algebraic reductions and quantifier-free logical labeling schemes (Lemma 5.11).

Forests and interval graphs are well-studied graph classes that lie in $\mathsf{GFO}(=)$ and $\mathsf{GFO}(<)$, which begs the question whether they are $\leq_{\mathrm{BF}}$-complete for them. It trivially holds that they are not since they are undirected (see the paragraph preceding Theorem 5.17). If we restrict $\mathsf{GFO}(=)$ and $\mathsf{GFO}(<)$ to undirected graph classes, however, then the answers are not as obvious. We show that forests and, in fact, any uniformly sparse graph class fails to be $\leq_{\mathrm{BF}}$-complete for the set of undirected graph classes in $\mathsf{GFO}(=)$ (Theorem 5.17). For interval graphs it is not even clear if k-interval graphs reduce to them.

Definition 5.1. We define negation, conjunction and disjunction on graphs and graph classes as follows. Let G, H be graphs over the same vertex set V.

$$\begin{aligned}
\neg G &:= (V, \{(u,v) \mid u \neq v \in V\} \setminus E(G)) && \text{(edge-complement without self-loops)}\\
G \wedge H &:= (V, E(G) \cap E(H)) && \text{(intersection of edges in } G \text{ and } H\text{)}\\
G \vee H &:= (V, E(G) \cup E(H)) && \text{(union of edges in } G \text{ and } H\text{)}
\end{aligned}$$

Let $\mathcal{C}, \mathcal{D}$ be graph classes.

$$\neg\mathcal{C} := \{\neg G \mid G \in \mathcal{C}\} = \text{co-}\mathcal{C}$$
$$\mathcal{C} \circ \mathcal{D} := \{G \circ H \mid G \in \mathcal{C}, H \in \mathcal{D} \text{ and } V(G) = V(H)\} \text{ for } \circ \in \{\vee, \wedge\}$$

Let φ be a boolean formula with k variables. We write $\varphi(\mathcal{C}_1, \ldots, \mathcal{C}_k)$ to denote the graph class that results from evaluating φ for the graph classes $\mathcal{C}_1, \ldots, \mathcal{C}_k$.

A graph G has arboricity at most k iff $G \in \bigvee_{i=1}^k \text{Forest}$, it has thickness at most k iff $G \in \bigvee_{i=1}^k \text{Planar}$ and it has boxicity at most k iff $G \in \bigwedge_{i=1}^k \text{Interval}$.

This definition induces an algebra on graph classes, which satisfies some laws of boolean algebra. For instance, negation is an involution ($\neg\neg\mathcal{C} = \mathcal{C}$) and conjunction and disjunction are commutative and associative. But $\mathcal{C} \vee \mathcal{C} = \mathcal{C}$ does not hold for all graph classes $\mathcal{C}$ because Forest $\vee$ Forest is the class of graphs with arboricity at most two, which contains the complete graph on 3 vertices K_3. The next lemma implies that all laws of boolean algebra where each variable occurs only once on each side of the equation are satisfied.

Definition 5.2. Let f be a k-ary boolean function. We define the functions f' and f'' based on f as follows. Let $G_1, \ldots, G_k$ be graphs on the same vertex set V. Then $f'(G_1, \ldots, G_k)$ is defined as the graph $G = (V, E)$ with $(u, v) \in E$ iff $u \neq v$ and $f(x_1, \ldots, x_k) = 1$ where $x_i := [\![(u, v) \in E(G_i)]\!]$ for $i \in [k]$ and $u, v \in V$. Let $\mathcal{C}_1, \ldots, \mathcal{C}_k$ be graph classes. Then $f''(\mathcal{C}_1, \ldots, \mathcal{C}_k)$ is defined as the graph class:

$$\big\{G \mid \exists(G_1, \ldots, G_k) \in \mathcal{C}_1 \times \cdots \times \mathcal{C}_k \text{ on vertex set } V(G) \text{ s.t. } G = f'(G_1, \ldots, G_k)\big\}$$

Lemma 5.3. *Let φ be a boolean formula with k variables where each variable occurs at most once and let f_φ be the k-ary boolean function that is represented by φ. It holds that $\varphi(\mathcal{C}_1, \ldots, \mathcal{C}_k) = f''_\varphi(\mathcal{C}_1, \ldots, \mathcal{C}_k)$ for all graph classes $\mathcal{C}_1, \ldots, \mathcal{C}_k$.*

Proof. We write $\vec{\mathcal{C}}$ to abbreviate $(\mathcal{C}_1, \ldots, \mathcal{C}_k)$ and $\vec{\mathcal{C}}_\times$ for $\mathcal{C}_1 \times \cdots \times \mathcal{C}_k$.

We show this using structural induction over φ. Suppose φ uses the variables $x_1, \ldots, x_k$. The base case is projection, i.e. $\varphi \triangleq x_i$ for some $i \in [k]$. It holds that $\varphi(\vec{\mathcal{C}}) = \mathcal{C}_i$ by definition and $\mathcal{C}_i = f''_\varphi(\vec{\mathcal{C}})$ directly follows from the definition of f''_φ. For the induction step we have to consider $\neg$, $\wedge$ and $\vee$. Let us start with negation. Suppose $\varphi \triangleq \neg\psi$. Due to the induction hypothesis it holds that $\psi(\vec{\mathcal{C}}) = f''_\psi(\vec{\mathcal{C}})$. Therefore $\varphi(\vec{\mathcal{C}}) = \neg f''_\psi(\vec{\mathcal{C}})$. It remains to argue that $f''_\varphi(\vec{\mathcal{C}}) = \neg f''_\psi(\vec{\mathcal{C}})$, which holds iff:

$$G \in f''_\varphi(\vec{\mathcal{C}}) \Leftrightarrow \neg G \in f''_\psi(\vec{\mathcal{C}})$$

Let $G \in f''_\varphi(\vec{\mathcal{C}})$. This holds iff there exist $(G_1, \ldots, G_k) \in \vec{\mathcal{C}}_\times$ such that $G = f'_\varphi(G_1, \ldots, G_k)$. It holds that $f'_\psi(G_1, \ldots, G_k) = \neg G$ since $f_\varphi(x_1, \ldots, x_k) = 1 \Leftrightarrow f_\psi(x_1, \ldots, x_k) = 0$ and therefore $\neg G \in f''_\psi(\vec{\mathcal{C}})$.

Suppose that $\varphi \triangleq \psi_1 \wedge \psi_2$. Since every variable occurs at most once in φ we can assume w.l.o.g. that ψ_1 references only (at most) the first l variables of φ and ψ_2 the last $k - l$ variables for some $l \in [k-1]$. Due to the induction hypothesis it holds that $\psi_i(\vec{\mathcal{C}}) = f''_{\psi_i}(\vec{\mathcal{C}})$ for $i \in \{1, 2\}$. Therefore $\varphi(\vec{\mathcal{C}}) = f''_{\psi_1}(\vec{\mathcal{C}}) \wedge f''_{\psi_2}(\vec{\mathcal{C}})$. It remains to argue that $f''_\varphi(\vec{\mathcal{C}}) = f''_{\psi_1}(\vec{\mathcal{C}}) \wedge f''_{\psi_2}(\vec{\mathcal{C}})$.

$$\begin{aligned}
& G \in f''_\varphi(\vec{\mathcal{C}}) \\
\Leftrightarrow\ & \exists(G_1, \ldots, G_k) \in \vec{\mathcal{C}}_\times : G = f'_\varphi(G_1, \ldots, G_k) \\
\Leftrightarrow\ & \exists(G_1, \ldots, G_k) \in \vec{\mathcal{C}}_\times : G = f'_{\psi_1}(G_1, \ldots, G_k) \wedge f'_{\psi_2}(G_1, \ldots, G_k) \\
\Leftrightarrow\ & \exists(H_1, \ldots, H_k), (J_1, \ldots, J_k) \in \vec{\mathcal{C}}_\times : G = f'_{\psi_1}(H_1, \ldots, H_k) \wedge f'_{\psi_2}(J_1, \ldots, J_k) \\
\Leftrightarrow\ & G \in f''_{\psi_1}(\vec{\mathcal{C}}) \wedge f''_{\psi_2}(\vec{\mathcal{C}})
\end{aligned}$$

The second equivalence holds because $f_\varphi(x_1, \ldots, x_k)$ is true iff $f_{\psi_1}(x_1, \ldots, x_k)$ and $f_{\psi_2}(x_1, \ldots, x_k)$ are true. Let us explain why the fourth statement implies the third statement. Assume $G = f'_{\psi_1}(H_1, \ldots, H_k) \wedge f'_{\psi_2}(J_1, \ldots, J_k)$. Then

$$G = f'_{\psi_1}(H_1, \ldots, H_l, J_{l+1}, \ldots, J_k) \wedge f'_{\psi_2}(H_1, \ldots, H_l, J_{l+1}, \ldots, J_k)$$

because f'_{ψ_1} and f'_{ψ_2} only depend on the first l and last $k-l$ parameters, respectively. Stated differently, choose $(H_1, \ldots, H_l, J_{l+1}, \ldots, J_k) \in \vec{\mathcal{C}}_\times$ as witness for the third statement.

An analogous argument can be made for $\vee$. $\square$

Corollary 5.4. *Let φ, ψ be boolean formulas with k variables where every variable occurs at most once. If φ and ψ are logically equivalent then they are equivalent on graph classes as well, i.e. $\varphi(\mathcal{C}_1, \ldots, \mathcal{C}_k) = \psi(\mathcal{C}_1, \ldots, \mathcal{C}_k)$ holds for all graph classes $\mathcal{C}_1, \ldots, \mathcal{C}_k$.*

Proof. It holds that $\varphi(\mathcal{C}_1, \ldots, \mathcal{C}_k) = f''_\varphi(\mathcal{C}_1, \ldots, \mathcal{C}_k)$ and $\psi(\mathcal{C}_1, \ldots, \mathcal{C}_k) = f''_\psi(\mathcal{C}_1, \ldots, \mathcal{C}_k)$ where f_φ and f_ψ are the k-ary boolean functions represented by φ and ψ (Lemma 5.3). Since φ and ψ are logically equivalent $f_\varphi = f_\psi$ and therefore $f''_\varphi = f''_\psi$. $\square$

Definition 5.5 (Algebraic Reduction)**.** Let $\mathcal{C}, \mathcal{D}$ be graph classes. We say $\mathcal{C}$ reduces to $\mathcal{D}$, in symbols $\mathcal{C} \leq_{\mathrm{BF}} \mathcal{D}$, if there exists a boolean formula φ such that $\mathcal{C} \subseteq \varphi(\mathcal{D}, \ldots, \mathcal{D})$. A set of graph classes A is closed under $\leq_{\mathrm{BF}}$-reductions if $\mathcal{C} \leq_{\mathrm{BF}} \mathcal{D}$ and $\mathcal{D} \in \mathsf{A}$ implies $\mathcal{C} \in \mathsf{A}$. A graph class $\mathcal{C}$ is $\leq_{\mathrm{BF}}$-complete for a set of graph classes A if $\mathcal{C} \in \mathsf{A}$ and every graph class in A reduces to $\mathcal{C}$. We write $[\mathcal{C}]_{\mathrm{BF}}$ to denote the set of graph classes that reduce to $\mathcal{C}$.

It is easy to verify that $\leq_{\mathrm{BF}}$ is reflexive and transitive. Reflexivity follows from the fact that $\mathcal{C} \subseteq \mathcal{D}$ implies $\mathcal{C} \leq_{\mathrm{BF}} \mathcal{D}$.

The argument that planar graphs reduce to interval graphs which we made at the beginning of this section can be generalized to arbitrary uniformly sparse graph classes since every such graph class has bounded arboricity and therefore can be expressed as $\bigvee_{i=1}^{k} \mathrm{Interval} \wedge \mathrm{Interval}$ for some k.

In the following we show that all sets of graph classes considered here are closed under $\leq_{\mathrm{BF}}$.

Lemma 5.6. *A set of graph classes A is closed under $\leq_{\mathrm{BF}}$-reductions if it is closed under subsets, negation and conjunction, i.e. $\mathsf{A} = [\mathsf{A}]_\subseteq$ and for all graph classes $\mathcal{C}, \mathcal{D} \in \mathsf{A}$ it holds that $\neg\mathcal{C}, \mathcal{C} \wedge \mathcal{D} \in \mathsf{A}$.*

Proof. Assume A is closed under subsets, negation and conjunction. Let $\mathcal{C} \leq_{\mathrm{BF}} \mathcal{D}$ via a boolean formula φ ($\mathcal{C} \subseteq \varphi(\mathcal{D}, \ldots, \mathcal{D})$) and $\mathcal{D} \in \mathsf{A}$. If a variable occurs more than once in φ, rename it to make each variable occur at most once. Since $\mathcal{D}$ is inserted for each variable during evaluation this does not affect the resulting graph class. Due to Corollary 5.4 we can replace each occurrence $x \vee y$ in φ with $\neg(\neg x \wedge \neg y)$. Since A is closed under negation and conjunction it follows that $\varphi(\mathcal{D}, \ldots, \mathcal{D}) \in \mathsf{A}$ and therefore $\mathcal{C} \in \mathsf{A}$ due to closure under subsets. $\square$

Fact 5.7. $\mathsf{GAC}^0, \mathsf{GP}, \mathsf{GEXP}, \mathsf{GR}, \mathsf{GALL}$ *and* $[\mathsf{Factorial} \cap \mathsf{Hereditary}]_\subseteq$ *are closed under $\leq_{\mathrm{BF}}$-reductions.*

Proof. We show that all classes satisfy the premise of Lemma 5.6. All of them are closed under subsets by definition. For all $\mathsf{G}\cdot$ classes closure under negation follows from closure under complement of the sets of languages from which they are derived and closure under conjunction follows from combining two labeling schemes. Given two labeling schemes $S_1 = (F_1, c_1), S_2 = (F_2, c_2)$ let $S_3 = (F_3, c_1 + c_2)$ with $F_3 = \{(x_1 x_2, y_1 y_2) \mid \exists n \in \mathbb{N}\, \forall i \in \{1, 2\} \colon x_i, y_i \in \{0,1\}^{c_i n} \wedge (x_i, y_i) \in F_i\}$. It holds that $\mathrm{gr}(S_1) \wedge \mathrm{gr}(S_2) = \mathrm{gr}(S_3)$ and the computational complexity of F_3 is the same as of F_1 and F_2.

For $[\mathsf{Factorial} \cap \mathsf{Hereditary}]_{\subseteq}$ it suffices to consider only hereditary graph classes to prove that it is closed under negation and conjunction. Let $\mathcal{C} \in [\mathsf{Factorial} \cap \mathsf{Hereditary}]_{\subseteq}$. Then its hereditary closure $[\mathcal{C}]_{\text{hc}}$ is in $[\mathsf{Factorial} \cap \mathsf{Hereditary}]_{\subseteq}$ by definition and if $\neg[\mathcal{C}]_{\text{hc}}$ is in $[\mathsf{Factorial} \cap \mathsf{Hereditary}]_{\subseteq}$ then $\neg\mathcal{C}$ must be as well since it is a subset of $\neg[\mathcal{C}]_{\text{hc}}$. An analogous argument can be made for $\wedge$.

The complement of a factorial, hereditary graph class remains factorial and hereditary. Thus, $[\mathsf{Factorial} \cap \mathsf{Hereditary}]_{\subseteq}$ is closed under negation. Suppose $\mathcal{C}, \mathcal{D}$ are factorial, hereditary graph classes. We argue that $\mathcal{C} \wedge \mathcal{D}$ is factorial and hereditary as well. A graph in $\mathcal{C} \wedge \mathcal{D}$ on n vertices is determined by choosing a graph with n vertices from $\mathcal{C}$ and $\mathcal{D}$. Therefore $\mathcal{C} \wedge \mathcal{D}$ contains at most $n^{\mathcal{O}(n)} \cdot n^{\mathcal{O}(n)} = n^{\mathcal{O}(n)}$ graphs which makes it factorial. Assume $G \in \mathcal{C} \wedge \mathcal{D}$ via the graphs H_1, H_2, i.e. $G = H_1 \wedge H_2$. Then every induced subgraph of G is in $\mathcal{C} \wedge \mathcal{D}$ by choosing the corresponding induced subgraphs of H_1 and H_2. Therefore $\mathcal{C} \wedge \mathcal{D}$ is hereditary. $\square$

Fact 5.8. $\mathsf{GFO}_{\text{qf}}(\sigma)$*,* $\mathsf{GFO}(\sigma)$ *and* PBS *are closed under* $\leq_{\text{BF}}$*-reductions for all* $\sigma \subseteq \{<, +, \times\}$*.*

Proof. Suppose $\mathcal{C} \leq_{\text{BF}} \mathcal{D}$ via a boolean formula φ with l variables ($\mathcal{C} \subseteq \varphi(\mathcal{D}, \ldots, \mathcal{D})$) and $S = (\psi, c)$ is a logical labeling scheme with $\mathcal{D} \subseteq \text{gr}(S)$ and ψ has $2k$ free variables. We construct a logical labeling scheme $S' = (\phi, c)$ where ϕ has $2kl$ free variables such that $\mathcal{C} \subseteq \text{gr}(S')$. Let

$$\phi(\vec{x_1}, \ldots, \vec{x_l}, \vec{y_1}, \ldots, \vec{y_l}) \triangleq \varphi(\psi(\vec{x_1}, \vec{y_1}), \ldots, \psi(\vec{x_l}, \vec{y_l}))$$

where $\vec{x_i}$ and $\vec{y_i}$ represent k variables for each $i \in [l]$.

Now, we argue why $\mathcal{C} \subseteq \text{gr}(S')$ holds. Suppose $G \in \mathcal{C}$. This implies there exist $H_1, \ldots, H_l \in \mathcal{D}$ with the same vertex set as G such that for all $u \neq v \in V(G)$ it holds that

$$(u, v) \in E(G) \Leftrightarrow f_\varphi(x_1, \ldots, x_l) = 1 \text{ with } x_i := [\![(u, v) \in E(H_i)]\!] \text{ for } i \in [l]$$

due to Lemma 5.3. Since $H_i \in \mathcal{D}$ there exists a labeling $\ell_i \colon V(G) \to [n^c]_0^k$ for every $i \in [l]$ which witnesses that H_i is in $\text{gr}(S)$. It holds that G is in $\text{gr}(S')$ via the labeling $\ell(v) := (\ell_1(v), \ldots, \ell_l(v))$. Since ϕ does not contain any quantifiers or function/relation symbols that were not already present in ψ, it follows that S' shows that $\mathsf{GFO}_{\text{qf}}(\sigma)$ and $\mathsf{GFO}(\sigma)$ are closed under $\leq_{\text{BF}}$-reductions. The same construction works for PBS. $\square$

The fact that all these sets of graph classes are closed under $\leq_{\text{BF}}$-reductions suggests that algebraic reductions are a sensible notion of reduction for graph classes in the context of labeling schemes. Before we continue with treating completeness, let us give an example of a set of graph classes that is not closed under $\leq_{\text{BF}}$-reductions: the set of all graph classes with bounded clique-width. Since the closure of path graphs under disjoint union—let's call it $\mathcal{P}$—has bounded clique-width and every grid graph can be expressed as disjunction of two graphs from $\mathcal{P}$, it follows that grid graphs reduce to $\mathcal{P}$. Assuming closure, this would imply that grid graphs have bounded clique-width which is false.

Fact 5.9. *There exists no graph class that is* $\leq_{\text{BF}}$*-complete for* $[\mathsf{Factorial} \cap \mathsf{Hereditary}]_{\subseteq}$*.*

Proof. Suppose $\mathcal{C}$ is $\leq_{\text{BF}}$-complete for $[\mathsf{Factorial} \cap \mathsf{Hereditary}]_{\subseteq}$. This implies that every factorial, hereditary graph class is a subset of some graph class from $\{\varphi_1(\mathcal{C}, \ldots, \mathcal{C}), \varphi_2(\mathcal{C}, \ldots, \mathcal{C}), \ldots\}$ where $\varphi_1, \varphi_2, \ldots$ is the set of all boolean formulas. Fact 5.7 implies that each graph class in this set is factorial. This contradicts Lemma 2.4. $\square$

Fact 5.10. *There exists no hereditary graph class that is* $\leq_{\text{BF}}$*-complete for* GAC^0*.*

Proof. For the sake of contradiction, assume there exists a hereditary graph class $\mathcal{C}$ that is $\leq_{\text{BF}}$-complete for GAC^0. Since $\mathcal{C} \in \mathsf{GAC}^0$ it must hold that $\mathcal{C}$ is factorial. This implies $\mathcal{C} \in [\mathsf{Factorial} \cap \mathsf{Hereditary}]_{\subseteq}$ and since this set of graph classes is closed under $\leq_{\text{BF}}$-reductions (Fact 5.8) this implies $\mathsf{GAC}^0 \subseteq [\mathsf{Factorial} \cap \mathsf{Hereditary}]_{\subseteq}$ which contradicts Fact 2.6. $\square$

Perhaps not so surprisingly, quantifier-free logical labeling schemes and algebraic reductions are closely related. Replace the atomic formulas in such a labeling scheme with propositional variables. The resulting boolean formula yields the same graph class as the labeling scheme when the graph classes which are represented by the atomic formulas are plugged in for the propositional variables. We call a logical labeling scheme (φ, c) atomic if φ is an atomic formula, i.e. it contains no boolean connectives and quantifiers.

Lemma 5.11 (Algebraic Interpretation)**.** *Let $\sigma \subseteq \{<, +, \times\}$ s.t. $\sigma = \emptyset$ or σ contains '$<$'. A graph class $\mathcal{C}$ is in $\mathsf{GFO}_{\mathrm{qf}}(\sigma)$ iff there exist atomic labeling schemes $S_1, \ldots, S_a$ over σ and a boolean formula φ with a variables such that $\mathcal{C} \subseteq \varphi(gr_\infty(S_1), \ldots, gr_\infty(S_a))$.*

Proof. "$\Rightarrow$": Let $\mathcal{C}$ be in $\mathsf{GFO}_{\mathrm{qf}}(\sigma)$. Due to Lemma 3.3 there exists a quantifier-free logical labeling scheme $S = (\psi, c)$ over σ such that $\mathcal{C} \subseteq \mathrm{gr}_\infty(S)$. Let $A_1, \ldots, A_a$ be the atomic formulas of ψ and let φ be the boolean formula with a variables that results from replacing every atomic formula in ψ with a propositional variable. We assume w.l.o.g. that ψ has $2ak$ variables x^i_j, y^i_j for $i \in [a], j \in [k]$ and the variables used in every atomic formula A_i are a subset of $\left\{x^i_j, y^i_j \mid j \in [k]\right\}$. This implies that every variable of ψ occurs in at most one atomic formula.

We claim that $\mathcal{C} \subseteq \varphi(\mathrm{gr}_\infty(A_1, c), \ldots, \mathrm{gr}_\infty(A_a, c))$. Let f_φ be the a-ary boolean function represented by φ. Due to Lemma 5.3 it holds that $\varphi(\mathrm{gr}_\infty(A_1, c), \ldots, \mathrm{gr}_\infty(A_a, c)) = f''_\varphi(\mathrm{gr}_\infty(A_1, c), \ldots, \mathrm{gr}_\infty(A_a, c))$. Let G be a graph in $\mathcal{C}$. We need to show that:

$$G \in f''_\varphi(\mathrm{gr}_\infty(A_1, c), \ldots, \mathrm{gr}_\infty(A_a, c))$$

This requires us to show that there exist graphs $G_1, \ldots, G_a$ over the vertex set $V(G)$ such that $G = f'_\varphi(G_1, \ldots, G_a)$ and $G_i \in \mathrm{gr}_\infty(A_i, c)$ for all $i \in [a]$. Since G is in $\mathcal{C}$ there exist labelings $\ell_i \colon V(G) \to [n^c]^k_0$ for every $i \in [a]$ such that

$$(u, v) \in E(G) \Leftrightarrow f_\varphi(x_1, \ldots, x_a) = 1 \text{ with } x_i := [\![(\mathcal{N}, \ell_i(u), \ell_i(v)) \models A_i]\!]$$

holds for all $u \neq v \in V(G)$. Let G_i be the graph with the same vertex set as G and there is an edge $(u, v) \in E(G_i)$ iff $(\mathcal{N}, \ell_i(u), \ell_i(v)) \models A_i$ for all $i \in [a]$ and $u \neq v$. Then $G = f'_\varphi(G_1, \ldots, G_a)$ holds by definition and $G_i \in \mathrm{gr}_\infty(A_i, c)$ via ℓ_i for all $i \in [a]$.

"$\Leftarrow$": Suppose $\mathcal{C} \subseteq \varphi(\mathrm{gr}_\infty(S_1), \ldots, \mathrm{gr}_\infty(S_a))$. Since $\mathsf{GFO}_{\mathrm{qf}}(\sigma)$ is closed under union (Lemma 3.5) it holds that $\mathcal{D} = \bigcup_{i=1}^{a} \mathrm{gr}_\infty(S_i)$ is in $\mathsf{GFO}_{\mathrm{qf}}(\sigma)$. This implies $\mathcal{C} \leq_{\mathrm{BF}} \mathcal{D}$ via φ because $\mathcal{C} \subseteq \varphi(\mathcal{D}, \ldots, \mathcal{D})$. Therefore $\mathcal{C}$ is in $\mathsf{GFO}_{\mathrm{qf}}(\sigma)$ due to closure under $\leq_{\mathrm{BF}}$-reductions (Fact 5.8). ☐

Lemma 5.12. *$\mathsf{GFO}_{\mathrm{qf}}(<, +)$ and $\mathsf{GFO}_{\mathrm{qf}}(<, \times)$ are subsets of $\mathsf{GFO}_{\mathrm{qf}}(<)$.*

Proof. To prove that $\mathsf{GFO}_{\mathrm{qf}}(<, \alpha)$ is a subset of $\mathsf{GFO}_{\mathrm{qf}}(<)$ for $\alpha \in \{+, \times\}$ we argue that it suffices to show that $\mathrm{gr}_\infty(S) \in \mathsf{GFO}_{\mathrm{qf}}(<)$ for every atomic labeling scheme S over $\{<, \alpha\}$. Assume that is the case. Given a graph class $\mathcal{C} \in \mathsf{GFO}_{\mathrm{qf}}(<, \alpha)$, there exist atomic labeling schemes $S_1, \ldots, S_a$ over $\{<, \alpha\}$ and a boolean formula φ such that $\mathcal{C} \subseteq \varphi(\mathrm{gr}_\infty(S_1), \ldots, \mathrm{gr}_\infty(S_a))$ due to Lemma 5.11. By assumption it holds that $\mathrm{gr}_\infty(S_1), \ldots, \mathrm{gr}_\infty(S_a)$ are in $\mathsf{GFO}_{\mathrm{qf}}(<)$ and therefore $\mathcal{D} = \bigcup_{i=1}^{k} \mathrm{gr}_\infty(S_i)$ is in $\mathsf{GFO}_{\mathrm{qf}}(<)$ due to closure under union (Lemma 3.5). It holds that $\mathcal{C} \leq_{\mathrm{BF}} \mathcal{D}$ via φ and due to closure under $\leq_{\mathrm{BF}}$-reductions (Fact 5.8) it follows that $\mathcal{C} \in \mathsf{GFO}_{\mathrm{qf}}(<)$.

Let $S = (\varphi, c)$ be an atomic labeling scheme over $\{<, \alpha\}$. We argue that $\mathrm{gr}_\infty(S)$ is in $\mathsf{GFO}_{\mathrm{qf}}(<)$ via a logical labeling scheme S' that we will construct. Using $\mathrm{gr}_\infty(S)$ instead of $\mathrm{gr}(S)$ allows us to assume that addition and multiplication are associative. Assume φ has variables $x_1, \ldots, x_k, y_1, \ldots, y_k$. The idea is to rearrange the (in)equation such that the variables $x_1, \ldots, x_k$ are on one side of the (in)equation and $y_1, \ldots, y_k$ are on the other side. This allows us to precompute the required values in the labeling for S'. Let us show how this works in detail when α is '+' and φ uses '<'. In that case φ is a linear inequation and can be written as

$$\sum_{i=1}^{k} a_i x_i + b_i y_i < \sum_{i=1}^{k} c_i x_i + d_i y_i$$

for some $a_i, b_i, c_i, d_i \in \mathbb{N}_0$ for $i \in [k]$. This can be rewritten as:

$$\underbrace{\sum_{i=1}^{k} (a_i - c_i) x_i}_{l_n(x_1,\ldots,x_k)} < \underbrace{\sum_{i=1}^{k} (d_i - b_i) y_i}_{r_n(y_1,\ldots,y_k)}$$

For $n \in \mathbb{N}$ let l_n, r_n be the functions induced by the left-hand and right-hand expression with signature $[n^c]_0^k \to \mathbb{R}$. Let E_n be the union of the image of l_n and the image of r_n. Let $E_n = \{e_1, \ldots, e_{z_n}\}$ for some $z_n \in \mathbb{N}$ and $e_1 < e_2 < \cdots < e_{z_n}$. For all $n \in \mathbb{N}$, $\vec{a}, \vec{b} \in [n^c]_0^k$ and $e_i = l_n(\vec{a}), e_j = r_n(\vec{b})$ it holds that:

$$(\mathcal{N}, \vec{a}, \vec{b}) \models \varphi \Leftrightarrow l_n(\vec{a}) < r_n(\vec{b}) \Leftrightarrow e_i < e_j \Leftrightarrow i < j$$

Let $S' = (\psi, d)$ where $\psi(x_1, x_2, y_1, y_2) \triangleq x_1 < y_2$ and $d = 2k(c+1)$. We show that $\mathrm{gr}_\infty(S) \subseteq \mathrm{gr}(S')$. Consider a graph G on n vertices that is in $\mathrm{gr}_\infty(S)$ via a labeling $\ell\colon V(G) \to [n^c]_0^k$. We construct a labeling $\ell'\colon V(G) \to [n^d]_0^2$ which shows that G is in $\mathrm{gr}(S')$. For $u \in V(G)$ let $\ell'(u) = (i, j)$ with $e_i = l_n(\ell(u))$ and $e_j = r_n(\ell(u))$. For all $u \neq v \in V(G)$ it holds that

$$\begin{aligned}(u, v) \in E(G) &\Leftrightarrow (\mathcal{N}, \ell(u), \ell(v)) \models \varphi \\ &\Leftrightarrow l_n(\ell(u)) < r_n(\ell(v)) \\ &\Leftrightarrow \ell'_1(u) < \ell'_2(v) \\ &\Leftrightarrow (\mathcal{N}, \ell'(u), \ell'(v)) \models \psi\end{aligned}$$

where ℓ'_i denotes the i-th component of the tuple. Since $|E_n| \leq 2(n^c+1)^k \leq n^{2k(c+1)} = n^d$ it holds that no value in the image of ℓ' exceeds n^d. □

Definition 5.13. A directed graph G with self-loops is strictly dichotomic if for all $u, v \in V(G)$ and $\alpha \in \{\mathrm{in}, \mathrm{out}\}$ it holds that $N_\alpha(u) \cap N_\alpha(v) = \emptyset$ or $N_\alpha(u) = N_\alpha(v)$. A directed graph G is dichotomic if self-loops can be added to G such that it becomes strictly dichotomic.

The graph with vertices u, v, w and edges $(u, v), (v, u), (u, w), (v, w)$ is dichotomic but not strictly dichotomic since $N_{\mathrm{out}}(u)$ and $N_{\mathrm{out}}(v)$ are neither disjoint nor equal but if we add the self-loops (u, u) and (v, v) then it becomes strictly dichotomic. Every directed forest is strictly dichotomic. Every vertex in a forest has in-degree at most one and therefore $N_{\mathrm{in}}(u) = N_{\mathrm{in}}(v)$ or $N_{\mathrm{in}}(u) \cap N_{\mathrm{in}}(v) = \emptyset$ for all $u, v \in V(G)$. Additionally, the out-neighborhoods of every distinct pair of vertices are disjoint because every node has a unique parent.

Theorem 5.14. *Dichotomic graphs are $\leq_{\mathrm{BF}}$-complete for* $\mathsf{GFO}(=)$.

Proof. Since $\mathsf{GFO}(=) = \mathsf{GFO}_{\mathrm{qf}}(=)$ (Fact 3.6) it suffices to show that dichotomic graphs are $\leq_{\mathrm{BF}}$-complete for $\mathsf{GFO}_{\mathrm{qf}}(=)$. We show that (1) a graph is dichotomic iff it is in $\mathrm{gr}(S)$ where $S = (\varphi, 1)$ and $\varphi(x_1, x_2, y_1, y_2) \triangleq x_1 = y_2$ and (2) $\mathrm{gr}(S') \subseteq \mathrm{gr}(S)$ holds for every atomic labeling scheme S' over $\emptyset$ (i.e. using only equality). Membership of dichotomic graphs in $\mathsf{GFO}_{\mathrm{qf}}(=)$ directly follows from (1). To see that every graph class $\mathcal{C}$ in $\mathsf{GFO}_{\mathrm{qf}}(=)$ reduces to dichotomic graphs consider the following argument. Let $\mathcal{C} \in \mathsf{GFO}_{\mathrm{qf}}(=)$. Due to Lemma 5.11 there exist atomic labeling schemes $S_1, \ldots, S_a$ over $\emptyset$ and a boolean formula φ such that $\mathcal{C} \subseteq \varphi(\mathrm{gr}(S_1), \ldots, \mathrm{gr}(S_a))$ ($\mathrm{gr}_\infty(\cdot) = \mathrm{gr}(\cdot)$ since no overflow can occur without addition or multiplication). Due to (2) $\mathrm{gr}(S_i) \subseteq \mathrm{gr}(S)$ holds for all $i \in [a]$. This implies $\mathcal{C} \subseteq \varphi(\mathrm{gr}(S), \ldots, \mathrm{gr}(S))$ and therefore $\mathcal{C}$ reduces to dichotomic graphs via φ.

(1) "$\Rightarrow$": Let G be a dichotomic graph with n vertices and let $V(G) = [n]$. Let G' be a strictly dichotomic graph with the same vertex set as G such that after removing all self-loops from G' one obtains G. Let $\sim_\alpha$ denote the equivalence relation on $V(G)$ such that $u \sim_\alpha v$ iff $N_\alpha(u) = N_\alpha(v)$ for $\alpha \in \{\mathrm{in}, \mathrm{out}\}$ where N_α refers to the α-neighborhood of G'. For a vertex $v \in V(G)$ let $[v]_\alpha$ denote a representative of the equivalence class of v w.r.t. $\sim_\alpha$. For a vertex $v \in V(G)$ with in-degree at least one let $[v]_{\mathrm{pred}} = [u]_{\mathrm{out}}$ where u is some vertex in $N_{\mathrm{in}}(v)$. If v has in-degree zero let $[v]_{\mathrm{pred}} = 0$. Observe that for all $u, v \in V(G)$ it holds that $[u]_{\mathrm{pred}} = [v]_{\mathrm{pred}}$ whenever $u \sim_{\mathrm{in}} v$ and therefore $[u]_{\mathrm{pred}} = [[u]_{\mathrm{in}}]_{\mathrm{pred}}$. It holds that G is in $\mathrm{gr}(S)$ via the labeling $\ell(v) = ([v]_{\mathrm{out}}, [[v]_{\mathrm{in}}]_{\mathrm{pred}})$ because for all $u \neq v$:

$$(u, v) \in E(G) \Leftrightarrow ([u]_{\mathrm{out}}, v) \in E(G) \Leftrightarrow [u]_{\mathrm{out}} = [v]_{\mathrm{pred}} \Leftrightarrow [u]_{\mathrm{out}} = [[v]_{\mathrm{in}}]_{\mathrm{pred}} \Leftrightarrow \ell_1(u) = \ell_2(v)$$

"$\Leftarrow$": Let G be a graph with n vertices that is in $\mathrm{gr}(S)$ via a labeling $\ell \colon V(G) \to [n]_0^2$. Add a self-loop to every vertex u of G such that $\ell_1(u) = \ell_2(u)$ and call the resulting graph G'. We argue that G' is strictly dichotomic and therefore G is dichotomic. Given two vertices u, v it holds that either $\ell_1(u) = \ell_1(v)$ and therefore u and v must have the same out-neighborhood or $\ell_1(u) \neq \ell_1(v)$ and thus their out-neighborhoods must be disjoint. The same argument can be made for the in-neighborhoods. It follows that G' is strictly dichotomic.

(2) Let $S' = (\psi, c)$ be an atomic labeling scheme over $\emptyset$ and let ψ have $x_1, \ldots, x_k, y_1, \ldots, y_k$ as free variables. If ψ is $x_i = x_j$ or $y_i = y_j$ for some $i, j \in [k]$ then it is simple to see that every graph in $\mathrm{gr}(S')$ is dichotomic and therefore $\mathrm{gr}(S') \subseteq \mathrm{gr}(S)$. Suppose $\psi \triangleq x_i = y_j$ for some $i, j \in [k]$. Assume $G \in \mathrm{gr}(S')$ via $\ell \colon V(G) \to [n^c]_0^k$. Since only the i-th and j-th component of ℓ are considered when evaluating ψ, the other components can be ignored. Let $Z_n = \{\ell_i(v) \mid v \in V(G)\} = \{e_1, \ldots, e_{z_n}\}$. It holds that $z_n \leq n$ and G is in $\mathrm{gr}(S)$ via $\ell'(v) = (a, b)$ where $e_a = \ell_i(v)$ and b is chosen s.t. $\ell_j(v) = e_b$ if $\ell_j(v) \in Z_n$ and $b = 0$ otherwise. □

Definition 5.15. A directed graph G with self-loops is a strict linear neighborhood graph if for all $u, v \in V(G)$ and $\alpha \in \{\mathrm{in}, \mathrm{out}\}$ it holds that $N_\alpha(u) \subseteq N_\alpha(v)$ or $N_\alpha(v) \subseteq N_\alpha(u)$. A directed graph G is a linear neighborhood graph if self-loops can be added to G such that it becomes a strict linear neighborhood graph.

Theorem 5.16. *Linear neighborhood graphs are $\leq_{\mathrm{BF}}$-complete for $\mathsf{GFO}(<)$.*

Proof. Since $\mathsf{GFO}(<) = \mathsf{GFO}_{\mathrm{qf}}(<)$ (Theorem 3.7) it suffices to show that linear neighborhood graphs are $\leq_{\mathrm{BF}}$-complete for $\mathsf{GFO}_{\mathrm{qf}}(<)$. We show that (1) a graph is a linear neighborhood graph iff it is in $\mathrm{gr}(S)$ where $S = (\varphi, 1)$ and $\varphi(x_1, x_2, y_1, y_2) \triangleq x_1 < y_2$ and (2) $\mathrm{gr}(S')$ reduces to $\mathrm{gr}(S)$ for every atomic labeling scheme S' over $\{<\}$. Then the same argument as in the proof of Theorem 5.14 applies, except that $\mathrm{gr}(S_i)$ must be replaced with $\phi_i(\mathrm{gr}(S), \ldots, \mathrm{gr}(S))$ (with $\mathrm{gr}(S_i) \leq_{\mathrm{BF}} \mathrm{gr}(S)$ via ϕ_i) instead of $\mathrm{gr}(S)$ for all $i \in [a]$ since we only show reducibility in (2) here.

(1) "$\Rightarrow$": Let G be a linear neighborhood graph with n vertices. Let G' be a strict linear neighborhood graph with the same vertex set as G such that $G' = G$ after removing all self-loops from G'. Let $\sim_{\mathrm{in}}$ be the equivalence relation on $V(G)$ such that $u \sim_{\mathrm{in}} v$ if $N_{\mathrm{in}}(u) = N_{\mathrm{in}}(v)$ where N_{in} refers to the in-neighborhood of G'. Let V_0 be the set of vertices with in-degree zero. Let $V_1, \ldots, V_k$ be the equivalence classes of $\sim_{\mathrm{in}}$ except V_0 such that $N_{\mathrm{in}}(V_i) \subsetneq N_{\mathrm{in}}(V_j)$ for all $i, j \in [k]$ with $i < j$. The following labeling $\ell \colon V(G) \to [n]_0^2$ shows that G is in $\mathrm{gr}(S)$. For $u \in V(G)$ let $\ell(u) = (u_1, u_2)$ with $u \in V_{u_2}$ and u_1 is the minimal value such that $u \in N_{\mathrm{in}}(V_{u_1+1})$ (or $u_1 = k$ if this minimum does not exist) for $u_1, u_2 \in [k]_0$. To see that this is correct, consider an edge $(u, v) \in E(G)$ and $\ell(u) = (u_1, u_2), \ell(v) = (v_1, v_2)$. It holds that $u \in N_{\mathrm{in}}(v) = N_{\mathrm{in}}(V_{v_2})$. Since $u \in N_{\mathrm{in}}(V_{v_2})$ it follows that $u_1 + 1 \leq v_2$ and thus $u_1 < v_2$. For a non-edge $(u, v) \notin E(G)$ it holds that $u \notin N_{\mathrm{in}}(v) = N_{\mathrm{in}}(V_{v_2})$. Therefore $u_1 + 1 > v_2$ and thus $u_1 \not< v_2$.

"$\Leftarrow$": Let G be a graph that is in $\mathrm{gr}(S)$ via a labeling $\ell\colon V(G) \to [n]_0^2$. Add a self-loop to every vertex u of G such that $\ell_1(u) < \ell_2(u)$ and call the resulting graph G'. We argue that G' is a strict linear neighborhood graph and therefore G is a linear neighborhood graph. Let $u, v \in V(G)$ and $\ell(u) = (u_1, u_2), \ell(v) = (v_1, v_2)$. If $u_1 \leq v_1$ then $N_{\text{out}}(v) \subseteq N_{\text{out}}(u)$. If $u_1 \geq v_1$ then $N_{\text{out}}(u) \subseteq N_{\text{out}}(v)$. The same holds for u_2, v_2 and the in-neighborhoods of u and v. Therefore G' is a strict linear neighborhood graph.

(2) Let $S' = (\psi, c)$ be an atomic labeling scheme over $\{<\}$ and let ψ have $x_1, \dots, x_k, y_1, \dots, y_k$ as free variables. If ψ uses '=' then it can be rewritten using '<' since $x = y$ iff $\neg(x < y \vee y < x)$. Therefore it suffices to consider only atomic labeling schemes using '<' and show that they reduce to $\mathrm{gr}(S)$.

If ψ is $x_i < x_j$ or $y_i < y_j$ for some $i, j \in [k]$ then it is easy to see that $\mathrm{gr}(S')$ is dichotomic and therefore can be expressed as atomic labeling scheme using '='. Therefore we assume $\psi \triangleq x_i < y_j$ for some $i, j \in [k]$. Let G be a graph with n vertices in $\mathrm{gr}(S')$ via a labeling $\ell\colon V(G) \to [n^c]_0^k$. Let $Z_n = \{\ell_i(v) \mid v \in V(G)\}$ and $Z_n = \{e_0, \dots, e_{z_n - 1}\}$ such that $e_0 < e_1 < \dots < e_{z_n-1}$ (the order of the values is preserved by the indices). Additionally, for $x \in \mathbb{N}_0$ we define $\pi(x)$ as p such that e_p is the smallest value in Z_n with $x \leq e_p$; if such a value does not exist then $\pi(x) = z_n$. For example, if $Z_n = \{3, 7, 11\} = \{e_0, e_1, e_2\}$ then $\pi(x) = 0$ for $0 \leq x \leq 3$, $\pi(x) = 1$ for $4 \leq x \leq 7$, $\pi(x) = 2$ for $8 \leq x \leq 11$ and $\pi(x) = 3$ for $x > 11$. Then G is in $\mathrm{gr}(S)$ via $\ell(v) = (a, \pi(\ell_j(v)))$ with $e_a = \ell_i(v)$. $\square$

Observe that only undirected graph classes can reduce to an undirected graph class since conjunction, disjunction and negation preserve the symmetry of the edge relation (by undirected we mean a graph class that only contains graphs with symmetric edge relation). Therefore it trivially holds that forests or interval graphs cannot be complete for $\mathsf{GFO}(=)$ or $\mathsf{GFO}(<)$. However, we can consider the undirected version of these sets where all non-undirected graph classes are removed. For a set of graph classes A let undirected A denote the set of undirected graph classes in A.

Theorem 5.17. *No uniformly sparse graph class is $\leq_{\mathrm{BF}}$-complete for undirected* $\mathsf{GFO}(=)$.

Proof. We prove this by showing that (1) a graph class $\mathcal{C}$ reduces to forests iff $\mathcal{C}$ or $\neg\mathcal{C}$ is uniformly sparse and (2) the set of all complete and empty graphs $\mathcal{X}$ is in $\mathsf{GFO}(=)$ but neither uniformly sparse nor co-uniformly sparse. Suppose $\mathcal{C}$ is uniformly sparse. Due to (1) it holds that $\mathcal{C} \leq_{\mathrm{BF}}$ Forest and therefore the set of graph classes that reduce to $\mathcal{C}$ is a subset of the set of uniformly sparse graph classes and their complements since $\mathcal{D} \leq_{\mathrm{BF}} \mathcal{C}$ implies $\mathcal{D} \leq_{\mathrm{BF}}$ Forest. This implies $\mathcal{X}$ cannot be reduced to $\mathcal{C}$ but it is in $\mathsf{GFO}(=)$ due to (2). Therefore $\mathcal{C}$ is not complete for undirected $\mathsf{GFO}(=)$.

(1) We show that if $\mathcal{C} \leq_{\mathrm{BF}}$ Forest then $\mathcal{C}$ or $\neg\mathcal{C}$ is uniformly sparse. The other direction follows from the fact that every uniformly sparse graph class has bounded arboricity. First, observe that $\mathcal{C} \wedge \mathcal{D} \subseteq \mathcal{C}$ whenever $\mathcal{C}$ is closed under edge deletion since $E(G \wedge H) \subseteq E(G)$ for all graphs G, H. Analogously, $\mathcal{C} \vee \mathcal{D} \subseteq \mathcal{C}$ if $\mathcal{C}$ is closed under edge insertion. Therefore Forest $\wedge\, \mathcal{D} \subseteq$ Forest and $\neg$Forest $\vee\, \mathcal{D} \subseteq \neg$Forest for all graph classes $\mathcal{D}$.

Suppose $\mathcal{C} \leq_{\mathrm{BF}}$ Forest via a boolean formula φ, i.e. $\mathcal{C} \subseteq \varphi(\text{Forest}, \dots, \text{Forest})$. We can assume w.l.o.g. that φ is in DNF due to Lemma 5.3 . A clause of φ is a conjunction of literals and a literal can be either Forest or $\neg$Forest. If a clause C of φ contains at least one positive literal (Forest) then it evaluates to a subset of Forest since Forest $\wedge\, \mathcal{C} \subseteq$ Forest. If a clause C with k literals contains only negative literals, i.e. $C = \bigwedge_{i=1}^k \neg\text{Forest}$, then it evaluates to $\neg \bigvee_{i=1}^k \text{Forest}$ which is the complement of the class of graphs with arboricity at most k. Therefore each clause in φ either evaluates to Forest or $\neg \bigvee_{i=1}^k \text{Forest}$ for some $k \in \mathbb{N}$.

Assume every clause in φ evaluates to Forest and φ has k clauses. Then $\varphi(\text{Forest}, \dots, \text{Forest})$ evaluates to the class of graphs with arboricity at most k which is uniformly sparse and therefore $\mathcal{C}$, which is a subset of this class, is uniformly sparse as well. If this assumption does not hold then at least one clause evaluates to $\mathcal{A} := \neg \bigvee_{i=1}^k \text{Forest}$ for some $k \in \mathbb{N}$. Since $\mathcal{A}$ is closed under

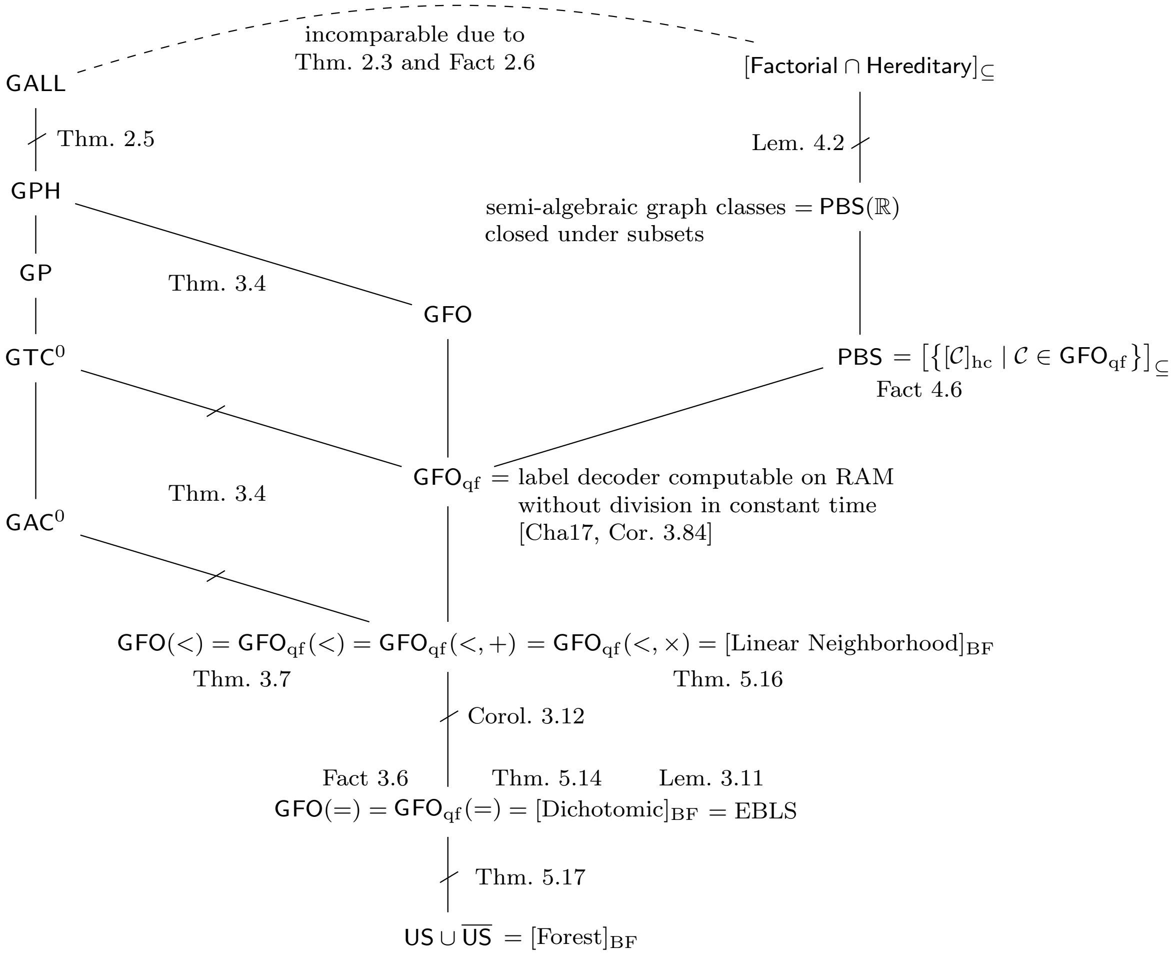

Figure 3: Overview of the sets of graph classes considered here

edge insertion it follows that $F(\text{Forest}, \ldots, \text{Forest})$ is a subset of $\mathcal{A}$ which is the complement of a uniformly sparse graph class and therefore this holds for $\mathcal{C}$ as well since it is a subset of $\mathcal{A}$.

(2) $\mathcal{X}$ is in $\mathsf{GFO}(=)$ via the logical labeling scheme $(\varphi, 1)$ with $\varphi(x_1, x_2, y_1, y_2) \triangleq x_1 = y_2 \vee y_1 = x_2$. For K_n label every vertex with $(1,1)$ and for $\neg K_n$ label every vertex with $(1,2)$. Neither $\mathcal{X}$ nor $\neg\mathcal{X}$ are uniformly sparse since both contain the set of complete graphs. □

6 Summary & Open Questions

There exist factorial, hereditary graph classes without a labeling scheme (Theorem 2.3). Albeit, the graph classes that witness this have been constructed for this very purpose. So the question remains for natural graph classes such as disk graphs, line segment graphs, k-dot product graphs or graph classes with bounded functionality whether they admit a labeling scheme. To refute the existence of a labeling scheme for a graph class one can either try to argue that it cannot have a polynomial-size universal graph or that such a labeling scheme cannot exist if the label decoder is confined to a particular level of complexity. The latter motivated us to introduce logical labeling schemes since they are highly structured compared to classical complexity classes while still capturing many of the graph classes known to have a labeling scheme.

As we considered various fragments of logical labeling schemes, three noteworthy sets of graph classes with various characterizations have emerged: $\mathsf{GFO}(=)$, $\mathsf{GFO}(<)$ and $\mathsf{GFO}_{\mathrm{qf}}$. The set $\mathsf{GFO}_{\mathrm{qf}}$ can be characterized as the set of graph classes with a labeling scheme whose label decoder can be computed on a RAM without division in constant time, making membership in this set also interesting from a more applied perspective. Moreover, its closure under hereditary closure coincides with PBS (Fact 4.6), which contains the first three aforementioned natural graph classes for which no labeling scheme is known. The sets $\mathsf{GFO}(=)$ and $\mathsf{GFO}(<)$ can be regarded as generalizations of the labeling schemes for uniformly sparse graphs and interval graphs. The set $\mathsf{GFO}(=)$ is equivalent to equality-based labeling schemes (EBLS), which cannot represent interval graphs among other graph classes (Theorem 3.10). Since disk graphs, line segment graphs and k-dot product graphs contain interval graphs for $k \geq 2$ ([Fid+98, Theorem 21]) this also rules out that these graph classes are in $\mathsf{GFO}(=)$. Is any of these graph classes in $\mathsf{GFO}(<)$? If one of them is not then this also implies that $\mathsf{GFO}(<) \neq \mathsf{GFO}_{\mathrm{qf}}$ due to Corollary 4.8.

Interestingly, all graph classes known to have a labeling scheme can be found in GAC^0 with the only exceptions that we are aware of being graph classes with bounded twin-width and induced subgraphs of hypercubes for which membership in GAC^0 is unclear. Therefore proving that a graph class is outside of GAC^0 would be a remarkable result since it shows that the most common techniques for constructing a labeling scheme fail. Proving a graph class to be outside of $\mathsf{GFO}(<)$ is a step towards such a result as it rules out a certain subset of labeling schemes from GAC^0. Note that no factorial, hereditary graph class is known which separates the two sets, i.e. it is theoretically possible that $\mathsf{GFO}(<) = \mathsf{GAC}^0 \cap [\mathsf{Factorial} \cap \mathsf{Hereditary}]_{\subseteq}$. Candidates to refute this are graph classes with bounded clique-width. Are graph classes with bounded clique-width even in PBS?

Algebraic reductions $\leq_{\mathrm{BF}}$ enable us to relate the adjacency structure of graph classes to one another in terms of boolean formulas. Showing that a graph class $\mathcal{C}$ reduces to another graph class $\mathcal{D}$ means that finding a labeling scheme for $\mathcal{C}$ is not harder than finding one for $\mathcal{D}$. The concept of completeness from complexity theory can also be applied in our context. We have shown that $\mathsf{GFO}(=)$ and $\mathsf{GFO}(<)$ have complete graph classes, namely dichotomic and linear neighborhood graphs, whereas $[\mathsf{Factorial} \cap \mathsf{Hereditary}]_{\subseteq}$ and GAC^0 do not (Fact 5.9 & 5.10). A line of inquiry that we find particularly interesting is to determine complete graph classes for $\mathsf{GFO}(=)$ and $\mathsf{GFO}(<)$ restricted to undirected graph classes. We have seen that no uniformly sparse graph class can be complete for undirected $\mathsf{GFO}(=)$ (Theorem 5.17). Is the graph class $\mathrm{gr}(\varphi, 1)$ with $\varphi \triangleq x_1 = y_2 \vee y_1 = x_2$—an undirected variant of dichotomic graphs—complete for undirected $\mathsf{GFO}(=)$? Another interesting problem is to establish a reduction among factorial, hereditary graph classes for which no labeling schemes are known.

Acknowledgements. I would like to thank Viktor Zamaraev for many stimulating discussions on the subject and for helping me understand the proof behind the refutation of the Implicit Graph Conjecture as well as the anonymous reviewer for giving me valuable feedback to improve the quality of this paper and pointers to relevant results in the literature that I missed.

References

[AAL21] Bogdan Alecu, Aistis Atminas, and Vadim V. Lozin. "Graph functionality". In: *J. Comb. Theory, Ser. B* 147 (2021), pp. 139–158. DOI: 10.1016/j.jctb.2020.11.002. URL: https://doi.org/10.1016/j.jctb.2020.11.002.

[ADK17] Stephen Alstrup, Søren Dahlgaard, and Mathias Bæk Tejs Knudsen. "Optimal Induced Universal Graphs and Adjacency Labeling for Trees". In: *J. ACM* 64.4 (Aug. 2017), 27:1–27:22. ISSN: 0004-5411. DOI: 10.1145/3088513. URL: http://doi.acm.org/10.1145/3088513.

[BBW00] József Balogh, Béla Bollobás, and David Weinreich. "The Speed of Hereditary Properties of Graphs". In: *Journal of Combinatorial Theory, Series B* 79.2 (2000), pp. 131–156. ISSN: 0095-8956. DOI: https://doi.org/10.1006/jctb.2000.1952. URL: https://www.sciencedirect.com/science/article/pii/S009589560091952X.

[Bon+21a] Marthe Bonamy, Louis Esperet, Carla Groenland, and Alex D. Scott. "Optimal labelling schemes for adjacency, comparability, and reachability". In: *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*. Ed. by Samir Khuller and Virginia Vassilevska Williams. ACM, 2021, pp. 1109–1117. DOI: 10.1145/3406325.3451102. URL: https://doi.org/10.1145/3406325.3451102.

[Bon+21b] Édouard Bonnet, Colin Geniet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. "Twin-Width II: Small Classes". In: *Proceedings of the Thirty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '21. Virtual Event, Virginia: Society for Industrial and Applied Mathematics, 2021, 1977–1996. ISBN: 9781611976465.

[Cha17] Maurice Chandoo. "Computational Complexity Aspects of Implicit Graph Representations". PhD thesis. Hannover, Germany, 2017. DOI: 10.15488/3569.

[Duj+21] Vida Dujmovic, Louis Esperet, Cyril Gavoille, Gwenaël Joret, Piotr Micek, and Pat Morin. "Adjacency Labelling for Planar Graphs (and Beyond)". In: *J. ACM* 68.6 (2021), 42:1–42:33. DOI: 10.1145/3477542. URL: https://doi.org/10.1145/3477542.

[EHZ22] Louis Esperet, Nathaniel Harms, and Viktor Zamaraev. "Optimal Adjacency Labels for Subgraphs of Cartesian Products". In: *CoRR* abs/2206.02872 (2022). DOI: 10.48550/arXiv.2206.02872. arXiv: 2206.02872. URL: https://doi.org/10.48550/arXiv.2206.02872.

[Fid+98] Charles M. Fiduccia, Edward R. Scheinerman, Ann Trenk, and Jennifer S. Zito. "Dot product representations of graphs". In: *Discrete Mathematics* 181.1 (1998), pp. 113 –138. ISSN: 0012-365X. DOI: http://dx.doi.org/10.1016/S0012-365X(97)00049-6. URL: http://www.sciencedirect.com/science/article/pii/S0012365X97000496.

[Fox+14] Jacob Fox, János Pach, Adam Sheffer, Andrew Suk, and Joshua Zahl. *A semi-algebraic version of Zarankiewicz's problem*. 2014. DOI: 10.48550/ARXIV.1407.5705. URL: https://arxiv.org/abs/1407.5705.

[Har20] Nathaniel Harms. "Universal Communication, Universal Graphs, and Graph Labeling". In: *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA*. Ed. by Thomas Vidick. Vol. 151. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, 33:1–33:27. DOI: 10.4230/LIPIcs.ITCS.2020.33. URL: https://doi.org/10.4230/LIPIcs.ITCS.2020.33.

[HH22] Hamed Hatami and Pooya Hatami. "The Implicit Graph Conjecture is False". In: *63nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022*. 2022. URL: https://arxiv.org/abs/2111.13198.

[HWZ22] Nathaniel Harms, Sebastian Wild, and Viktor Zamaraev. "Randomized communication and implicit graph representations". In: *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*. Ed. by Stefano Leonardi and Anupam Gupta. ACM, 2022, pp. 1220–1233. DOI: 10.1145/3519935.3519978. URL: https://doi.org/10.1145/3519935.3519978.

[KM12] Ross J. Kang and Tobias Müller. "Sphere and Dot Product Representations of Graphs". In: *Discrete & Computational Geometry* 47.3 (2012), pp. 548–568. ISSN: 1432-0444. DOI: 10.1007/s00454-012-9394-8. URL: https://doi.org/10.1007/s00454-012-9394-8.

[KNR92] Sampath Kannan, Moni Naor, and Steven Rudich. "Implicit Representation of Graphs". In: *SIAM Journal on Discrete Mathematics* 5.4 (Nov. 1992), pp. 596–603. ISSN: 0895-4801 (print), 1095-7146 (electronic).

[LZ15] Vadim V. Lozin and Victor Zamaraev. "Boundary Properties of Factorial Classes of Graphs". In: *Journal of Graph Theory* 78.3 (2015), pp. 207–218. DOI: https://doi.org/10.1002/jgt.21799. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jgt.21799. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/jgt.21799.

[LZ17] Vadim Lozin and Viktor Zamaraev. "The structure and the number of P7-free bipartite graphs". In: *Electronic Notes in Discrete Mathematics* 61 (2017). The European Conference on Combinatorics, Graph Theory and Applications (EUROCOMB'17), pp. 827–833. ISSN: 1571-0653. DOI: https://doi.org/10.1016/j.endm.2017.07.042. URL: https://www.sciencedirect.com/science/article/pii/S157106531730207X.

[MM13] Colin McDiarmid and Tobias Müller. "Integer realizations of disk and segment graphs". In: *Journal of Combinatorial Theory, Series B* 103.1 (2013), pp. 114 –143. ISSN: 0095-8956. DOI: http://dx.doi.org/10.1016/j.jctb.2012.09.004. URL: http://www.sciencedirect.com/science/article/pii/S0095895612000718.

[Mul88] John Harold Muller. "Local Structure in Graph Classes". Order No: GAX88-11342. PhD thesis. Atlanta, GA, USA, 1988.

[Smo91] Craig Smoryński. *Logical Number Theory I: An Introduction*. Logical Number Theory I: An Introduction Bd. 1. Springer, 1991. ISBN: 9780387522364.

[Spi03] Jeremy P. Spinrad. *Efficient Graph Representations.: The Fields Institute for Research in Mathematical Sciences*. Fields Institute monographs. American Mathematical Soc., 2003. ISBN: 9780821871775.

[Vol99] Heribert Vollmer. *Introduction to Circuit Complexity: A Uniform Approach*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1999. ISBN: 3540643109.