Definability in First Order Theories of Graph Orderings

R. Ramanujam and R.S. Thinniyam(✉)

The Institute of Mathematical Sciences, Chennai 600113, India
{jam,thinniyam}@imsc.res.in

Abstract. We study definability in the first order theory of graph order: that is, the set of all simple finite graphs ordered by either the minor, subgraph or induced subgraph relation. We show that natural graph families like cycles and trees are definable, as also notions like connectivity, maximum degree etc. This naturally comes with a price: bi-interpretability with arithmetic. We discuss implications for formalizing statements of graph theory in such theories of order.

Keywords: Graphs · Partial order · Logical theory · Definability

1 Introduction

Reasoning about graphs is a central occupation in computing science, since graphs are used to model many computational problems such as those in social networks, communication etc. In many cases, a single fixed graph is considered and some property has to be verified (e.g. bipartiteness) or some numerical parameter computed(e.g. independence number). However, as the complexity of the query increases, it can often be naturally recast as a question of relationships between graphs. For instance, asking if a graph is Hamiltonian is the same as looking for a cycle of the same order as the graph which occurs as a subgraph; asking for a k-colouring is the same as asking for a homomorphism of the graph to the k-clique. Studying the nature of relations on the set of all graphs has led to results such as the Graph Minor Theorem [18], whose algorithmic implications and influence on computer science cannot be overstated [1,2].

Consider the natural relations on graphs given by subgraph, induced subgraph and minor: these form partial orders over the set of all (simple, finite) graphs with interesting properties (see Fig. 1). Logical statements about these partial orders refer to graph families, and typically those given by some 'first order' closure condition, such as including/avoiding specific characteristics. Such statements are of immense interest to the theory of algorithms, motivating the logical study of graph order, and first order theories are the natural candidates for such a study.

Model theorists have taken up such studies. In a series of papers, Jezek and McKenzie [8–11] study the first order definability in substructure orderings on various finite ordered structures such as lattices, semilattices etc. Such a

S. Artemov and A. Nerode (Eds.): LFCS 2016, LNCS 9537, pp. 331–348, 2016.
DOI: 10.1007/978-3-319-27683-0_23

study is indeed foundational, and yet, it is of interest to study specific order structures on graphs to exploit their additional properties (if any). Indeed, the substructure order over graphs corresponds to the induced subgraph order, and this was investigated by Wires [19]. However, subgraph and minor orders are less amenable as substructure and hence deserve a closer look, which is the attempt initiated here. In the setting of directed graphs, the subdigraph order has been investigated by Kunos [16] recently. Work on word orders has been carried out by Kuske [17] as well as Kudinov et al. [14,15]. Other recent work on theories of classes of structures such as boolean algebras, linear orders and groups by Kach and Montalban [12] are different in spirit to ours, since they consider additive operations and the underlying structures may be infinite.

Our attempt here is not to study one graph order but rather to highlight the subtle differences in definability between different graph orders even while showing that they are all powerful enough to encode first order arithmetic. In fact, the subgraph and induced subgraph order are shown to be bi-interpretable with first order arithmetic. Many predicates which are interesting from a graph theoretic perspective such as connectivity, regularity, etc. are found to be first order definable, enabling us to articulate classical theorems of graph theory in such order theories.

We suggest that this paper as well as the related work mentioned are merely first steps of a larger programme of research, since we lack the tools as yet to address many related questions regarding indefinability, succinctness, algorithmic solutions, and so on.

The paper is organised as follows. After setting up the preliminaries, we study the subgraph order and show that certain numerical parameters such as order of a graph, commonly encountered graph families such as paths, cycles etc. and interesting graph predicates such as connectivity can be defined. We then show how such results can be lifted to the minor order. The machinery developed is used to show the bi-interpretability with arithmetic of the induced subgraph and subgraph orders and to interpret arithmetic in the minor order. Finally we display some interesting graph theoretical statements which can be stated using graph orders and discuss the research programme ahead.

2 Preliminaries

For the standard syntax and semantics of first order logic, we refer the reader to Enderton [4].

Definition 1 (Definability of Constants). *Fix a first order language $\mathcal{L}$. Let a be an element of the domain of an $\mathcal{L}$ structure $\mathcal{A}$. We say that a is definable in $\mathcal{A}$ if there exists an $\mathcal{L}$ formula $\phi_a(x)$ in one free variable such that $\mathcal{A}, a \vDash \phi_a(x)$ and for any $a' \neq a$ in the domain of $\mathcal{A}$, $\mathcal{A}, a' \nvDash \phi_a(x)$.*

We use a as a constant symbol representing the domain element a with the understanding that an equivalent formula can be written without the use of this constant symbol.

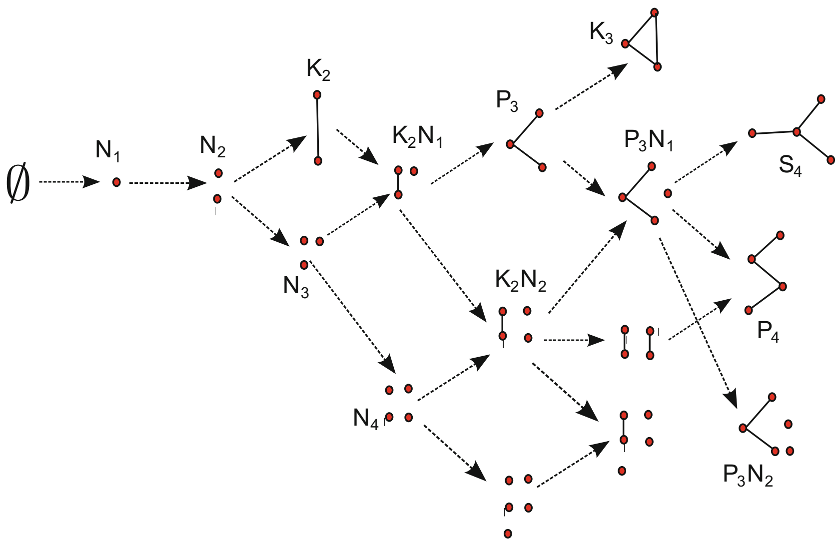

Fig. 1. The first few layers of the subgraph order.

Definition 2 (Covering Relation of a Poset). *Given an element x of a poset $(E, \leq)$, y is called an* upper cover *of x iff $x < y$ and there exists no element z of E such that $x < z < y$.*
Similarly y is called a lower cover *of x iff $y < x$ and there exists no element z of E such that $y < z < x$.*

Definition 3 (Graph Partial Orders). *Consider the following operations on graphs:*

- *A1. Deletion of a vertex (and all the edges incident on that vertex).*
- *A2. Deletion of an edge.*
- *A3. Contraction of an edge (given an edge $e = uv$, delete both u and v and introduce a new vertex w not in $V(g)$; connect all vertices which were adjacent to either u or v to w).*

For graphs g and g', g can be obtained from g' by any finite sequence of the operations:

1. *A1, A2 and A3 iff $g \leq_m g'$ (g is a minor of g').*
2. *A1 and A2 iff $g \leq_s g'$ (g is a subgraph of g').*
3. *A1 iff $g \leq_i g'$ (g is an induced subgraph of g').*

Let $\mathcal{G}$ denote the set of all simple graphs. We consider the base first order language $\mathcal{L}_0$ which has only the binary predicate symbol $\leq$ and an extension $\mathcal{L}_1$ that extends $\mathcal{L}_0$ with a constant symbol P_3 which stands for the path on three vertices. The latter is used in the case of the induced subgraph order in order to break the symmetry imposed by the automorphism which takes every graph to its complement.

Definition 4 (Graph Structures). *We denote the first order theories of the subgraph and minor orders by $\mathcal{L}_0$ structures $(\mathcal{G}, \leq_s)$ and$(\mathcal{G}, \leq_m)$ respectively; and the induced subgraph order by the $\mathcal{L}_1$ structure $(\mathcal{G}, \leq_i, P_3)$.*

Notation: We use the letters x, y, z to denote variables representing graphs in formulas, u, v to represent nodes of a graph, e to represent the edge of a graph, g, h to represent graphs, $\mathcal{F}, \mathcal{G}$ to represent families of graphs. We write uv to denote the edge joining nodes u and v. We will denote by N_i, K_i, C_i, S_i, P_i the graph consisting of i isolated vertices, the i-clique, the cycle on i vertices, the star on i vertices and the path on i vertices respectively (Fig. 2); and by $\mathcal{N}, \mathcal{K}, \mathcal{C}, \mathcal{S}, \mathcal{P}$ the corresponding families of isolated vertices, cliques, cycles, stars and paths. $\mathcal{F}, \mathcal{T}$ represent forests and trees respectively. We will also on occasion, refer to certain fixed graphs or graph families by descriptive names (see Fig. 4).

k, l, m, n are used for natural numbers (also on occasion, members of the $\mathcal{N}$ family). All subscript or superscript variants such as x', x_i, etc. will be used to denote the same kind of object.

Given a graph g, $V(g)$ stands for the vertex set of g, $E(g)$ stands for the edge set of g, $|g|$ stands for the number of vertices of g (also called the order of g) and $|g|_{gr}$ stands for the graph consisting of only isolated vertices which has the same number of vertices as g. $||g||$ stands for the number of edges of g, also called the size of g. Given graphs g and h, $g \cup h$ stands for the disjoint union of g and h.

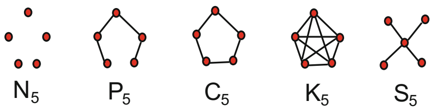

Fig. 2. Isolated points, path, cycle, clique and star of order 5 from left to right.

3 Definability in the Subgraph Order

We will take up definability in the subgraph order first. The defining formulae have been chosen such that most of them carry over in a straightforward way to the minor order. For a few predicates, significant modifications are required.

Constants, Covers and Cardinality

Lemma 1. *The upper and lower covering relations, the order of a graph, the family $\mathcal{N}$ and the graphs N_1, K_2, K_3, S_4, P_4 are definable in subgraph.*

The upper and lower covering relations for subgraph can immediately be defined:
$uc_s(x, y)$ iff x is an upper cover of y: $uc_s(x, y) := y <_s x \ \wedge \ \neg\exists z \ y <_s z <_s x$
$lc_s(x, y)$ iff x is a lower cover of y: $lc_s(x, y) := x <_s y \ \wedge \ \neg\exists z \ x <_s z <_s y$
Next we show that certain graphs in the first few layers of the subgraph order are definable. Refering to Fig. 1, the following formulae can easily be verified:

1. $\emptyset(x) := \forall y\ x \leq_s y$
2. $N_1(x) := uc_s(x, \emptyset)$; $N_2(x) := uc_s(x, N_1)$
3. $K_2(x) := uc_s(x, N_2) \wedge \exists y\ uc_s(y, x)\ \wedge\ \forall z\ uc_s(z, x) \supset z = y$
4. $N_3(x) := uc_s(x, N_2)\ \wedge\ x \neq K_2$
5. $K_2N_1(x) := uc_s(x, K_2)$; $K_2N_2(x) := uc_s(x, K_2N_1)\ \wedge\ uc_s(x, N_3)$
6. $P_3(x) := \exists! y\ uc_s(x, y) \wedge y = K_2N_1$ (where $\exists!$ is short for there exists unique)
7. $P_3N_1(x) := uc_s(x, P_3) \wedge uc_s(x, K_2N_2) \wedge \forall y\ uc_s(x, y) \supset (y = P_3 \vee y = K_2N_2)$
8. $S_4(x) := uc_s(x, P_3N_1)\ \wedge\ \forall y\ uc_s(x, y) \supset y = P_3N_1$
9. $K_3(x) := \exists! y\ lc_s(y, x)\ \wedge\ y = P_3$
10. $P_4(x) := uc_s(x, P_3N_1)\ \wedge\ x \neq S_4$

We note that if a family of totally ordered graphs is definable, then every member is definable as a constant by repeated use of the covering relation.
The family of isolated points is now easily seen to be definable via: $\mathcal{N}(x) := K_2 \not\leq_s x$. In addition, using the family $\mathcal{N}$ as a "yardstick", we can capture the cardinality (order) of a graph.
$order(n, x)$ iff $n \in \mathcal{N}$ and $|x| = |n|$:
$order(n, x) := \mathcal{N}(n)\ \wedge\ \forall m\ (\mathcal{N}(m)\ \wedge\ m \leq_s x) \supset m \leq_s n$.

For definable numerical predicates such as cardinality, we will simply use them as functions instead of predicates to simplify notation from here on i.e. $|x|_{gr}$ will denote the member of $\mathcal{N}$ whose order is the same as that of x.

Graph Families

Theorem 1. *The families $\mathcal{K}, \mathcal{P}, \mathcal{C}, \mathcal{F}, \mathcal{T}, \mathcal{S}$ are definable using subgraph.*

Cliques: Any graph to which an edge can be added contains at least two upper covers. The unique upper cover of a clique is formed by adding an isolated point to it. $\mathcal{K}(x) := \exists! y\ uc_s(y, x)$.

Paths
In order to define paths, we need to define a few additional families :

1. Disjoint unions of paths and cycles (denoted pac)
2. Disjoint unions of cycles i.e. sums of cycles (denoted soc)
3. Disjoint unions of paths i.e. forest of paths(denoted fop)

Assuming these, we can define paths :
$\mathcal{P}(x) := fop(x)\ \wedge\ \forall y\ |x|_{gr} = |y|_{gr}\ \wedge\ fop(y) \supset y \leq_s x$.

Out of all the fops of the same order n, the P_n forms the maximum element. Clearly by adding appropriate edges to a fop of the same order, one can form P_n. Adding any more edges to P_n gives a non-fop.

A graph is a disjoint union of paths and cycles iff it has maximum degree at most two: $pac(x) := S_4 \not\leq_s x$.

Assuming soc, fop can be defined: $fop(x) := pac(x) \wedge (\forall y\ soc(y) \supset y \not\leq_s x)$.
if: x is clearly a pac. Since x does not have any cycles as subgraph, it cannot have any soc as a subgraph.

only if: Let $x = c_1 \cup c_2 \cup ... \cup c_n$ where c_i is either a path or a cycle for all i. Suppose there is an i with c_i cycle. Then clearly $c_i \leq_s x$ but c_i is also a soc, which is a contradiction. Hence all components are paths and x is a fop.

It is only left to define disjoint unions (sums) of cycles i.e. soc (Fig. 3):

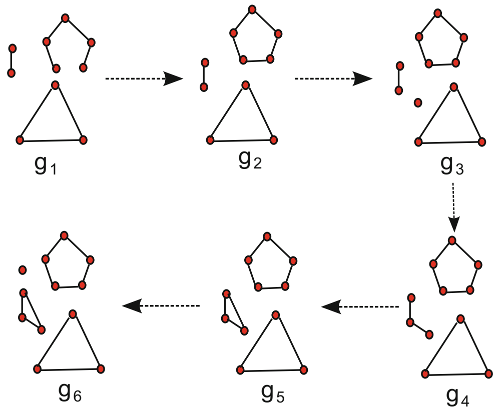

Fig. 3. $g_1, g_2, g_3, g_4, g_5, g_6$ all pac, only g_2, g_5, g_6 soc', only g_5 soc.

$$soc(x) = soc'(x) \wedge \forall y\,(uc_s(y,x) \wedge pac(y)) \supset soc'(y)$$

where

$$soc'(x) := x \neq \emptyset \ \wedge \ pac(x) \wedge \ \forall y\,(|y|_{gr} = |x|_{gr} \wedge pac(y)) \supset \neg x <_s y$$

Claim 1. *$soc'(x)$ iff every component of x is a cycle, N_1 or K_2 and x contains at most one copy of N_1 or one copy of K_2 but not both and x is not the empty graph.*

Proof. if: Clearly x is a pac. Suppose there exists a pac y of the same order as x and $x <_s y$. We can obtain y from x by addition of edges. But addition of any edge would introduce a degree three node, thus such a y cannot exist.

only if: Let $x = c_1 \cup c_2 \cup ... \cup c_n$ where c_i is either a cycle or a path. Suppose there is an i such that c_i is a path of order at least three. Let c_i' be the cycle formed by joining the ends of c_i. Now $x' = c_1 \cup ...c_{i-1} \cup c_i' \cup c_{i+1} ... \cup c_n$ is also a pac, $|y| = |x|$ and x can obtained from y by deleting the newly added edge to get c_i from c_i'. Thus no path of length more than one can exist. Similarly, we can obtain a contradiction in the following cases by appropriately constructing x':

1. There are two copies of K_2 in x. Join the two copies end to end to form a path of length three, to get x'.

2. There are two copies of N_1 in x. Join the copies by an edge to get x'.
3. There is a K_2 and an N_1 as components in x. Join N_1 by an edge to K_2 to get a path of length two, to get x'. □

Now we show the correctness of $soc(x)$.

if: Clearly x is a soc'. The only upper cover of x which is a pac is $x \cup N_1$ since adding any more edges would lead to a degree three node. $x \cup N_1$ is a soc'.

only if: Let $x = c_1 \cup c_2 \cup ... \cup c_n$ and x is a soc'. Suppose there is i such that c_i is K_2. Let $x' = x \cup N_1$. x' is an upper cover of x, is a pac but is not a soc' because it has an N_1 and a K_2 as components. Similarly we can rule out N_1 as a component of x.

Cycles, Forests, Trees, Stars

$$\begin{aligned}
\mathcal{C}(x) :=& pac(x) \;\wedge\; \exists y\, \mathcal{P}(y) \;\wedge |x|_{gr} = |y|_{gr} \;\wedge\; uc_s(x,y) \\
forest(x) :=& \forall y\, \mathcal{C}(y) \supset y \not\leq_s x \\
\mathcal{T}(x) :=& forest(x) \;\wedge\; \forall y\, (forest(y) \;\wedge\; |x|_{gr} = |y|_{gr}) \supset \neg x <_s y \\
\mathcal{S}(x) :=& \mathcal{T}(x) \;\wedge\; P_4 \not\leq_s x
\end{aligned}$$

It is clear that by deleting any edge from a cycle, we get a path which is a lower cover of the same order.

Conversely, consider any upper cover of a path with the same order. Adding an edge which joins the degree one vertices of the path gives a cycle, but adding an edge any where else creates a degree three vertex, which violates the condition that x is a pac. Thus only a cycle fulfills all the conditions.

A forest is a graph which contains no cycles. Of all forests with the same order, a tree is a maximal element since adding another edge gives a cycle. A non-tree forest can be made into a tree of same order by adding appropriate edges. A star is a tree which does not contain a path on four vertices as subgraph. Conversely, consider any tree with longest path on at most three vertices. Any other vertex must be connected to the midpoint of this longest path, thus it is a star.

Graph Predicates

Theorem 2. *Connectivity, maximum degree and maximum path length are definable in subgraph.*

Connectivity

$$conn(x) := \exists y\, \mathcal{T}(y) \;\wedge\; y \leq_s x \;\wedge\; |x|_{gr} = |y|_{gr}$$

A graph is connected iff it has a spanning tree.

Maximum path

$maxPath(n, x)$ iff $n \in \mathcal{N}$ and the largest path which is a subgraph of x is P_n.

$$\begin{aligned}
maxPath(n, x) :=& \mathcal{N}(n) \;\wedge\; \exists y\, \mathcal{P}(y) \;\wedge\; y \leq_s x \;\wedge |y|_{gr} = n \;\wedge \\
& \forall z\, (\mathcal{P}(z) \;\wedge\; z \leq_s x) \supset z \leq_s y
\end{aligned}$$

Maximum degree
$maxDeg(n,x)$ iff $n \in \mathcal{N}$ and the maximum degree of x is $|n|$.

$$maxDeg(n,x) := \mathcal{N}(n) \ \wedge \ \exists y\, \mathcal{S}(y) \ \wedge \ y \leq_s x \ \wedge \ uc_s(|y|_{gr}, n) \wedge \\ \forall z\, (\mathcal{S}(z) \ \wedge \ z \leq_s x) \supset z \leq_s y$$

The maximum degree of x is one less than the order of the largest star which is a subgraph of x.

4 Definability in the Minor Order

Note that the minor order is identical to subgraph in an initial segment; the first additional relation which occurs in cycles and forests is shown in Fig. 4. This observation helps us reuse some of the machinery already developed for subgraph.

Observation 1. *The downclosure of S_5 and downclosure of K_3 are identical under subgraph and minor.*

Observation 2. *If $|x| = |y|$ then $x \leq_s y$ iff $x \leq_m y$ and $uc_s(x,y)$ iff $uc_m(x,y)$. Since the contraction operation reduces the number of vertices, restricting the orders to tuples of the same cardinality makes minor and subgraph equivalent.*

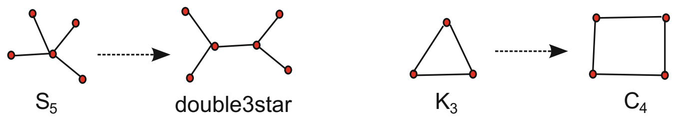

Fig. 4. First difference between subgraph and minor.

We also have the following lemma on when the two orders can be taken to be equivalent.

Lemma 2. *Let x_n be an tree with at most one degree 3 node and no node of degree 4 or more. Then for any other graph x_0, $x_n \leq_m x_0$ iff $x_n \leq_s x_0$.*

Proof. It suffices to prove the only if direction since any subgraph is also a minor.

We observe that there is a normal form for any sequence of minor operations. Let $x_n \leq_m x_0$ via a sequence of minor operations $o_1, o_2, ..., o_n$, then there exists a series of minor operations $o'_1, ..., o'_m$ on x_0 resulting in x_n such that no deletion operation occurs after a contraction operation and the number of contraction operations in the sequence $o'_1, ..., o'_m$ is at most the number of contractions in the original sequence $o_1, ..., o_n$.

The result is proved by induction on the number of contraction operations in transforming x_0 to x_n. The details are given in the appendix. $\square$

The lemma and observations above help us transfer some results on definability from subgraph and minor order, simply by replacing the subgraph order by the minor order in the defining formulae.

Lemma 3. *The upper and lower covering relations, the order of a graph, the family $\mathcal{N}$ and the graphs N_1, K_2, K_3, S_4 are definable in minor order.*

Theorem 3. *The families $\mathcal{K}, \mathcal{P}, \mathcal{C}, \mathcal{F}, \mathcal{T}, \mathcal{S}$ are definable using minor.*

Firstly note that a graph contains a cycle as subgraph iff it contains K_3 as a minor (by contraction along the cycle). Hence forests are defined by: $\mathcal{F}(x) := K_3 \not\leq_m x$.

Observe that disjoint unions of paths and cycles (pac) can be defined by: $pac(x) := S_4 \not\leq_m x$. (By Lemma 2, we replace subgraph by minor in the defining formula). For forest of paths (fop), note that we can restrict a pac to be a forest, giving a fop: $fop(x) := \mathcal{F}(x) \;\wedge\; pac(x)$. Now, using Observation 2, we can immediately get paths, cliques, cycles and trees; and stars by Lemma 2:

$$\begin{aligned}
\mathcal{P}(x) :=& fop(x) \;\wedge\; \forall y\; |x|_{gr} = |y|_{gr} \;\wedge\; fop(y) \supset y \leq_m x \\
\mathcal{K}(x) :=& \forall y\; |y|_{gr} = |x|_{gr} \supset y \leq_m x \\
\mathcal{C}(x) :=& pac(x) \;\wedge\; \exists y\, \mathcal{P}(y) \;\wedge |x|_{gr} = |y|_{gr} \;\wedge\; uc_m(x, y) \\
\mathcal{T}(x) :=& forest(x) \;\wedge\; \forall y\; (forest(y) \;\wedge\; |x|_{gr} = |y|_{gr}) \supset \neg x <_m y \\
\mathcal{S}(x) :=& \mathcal{T}(x) \;\wedge\; P_4 \not\leq_m x
\end{aligned}$$

Theorem 4. *Connectivity, maximum degree and maximum path length are definable in minor.*

$$\begin{aligned}
conn(x) :=& \exists y\; \mathcal{T}(y) \;\wedge\; y \leq_m x \;\wedge\; |x|_{gr} = |y|_{gr} \\
maxPath(n, x) :=& \mathcal{N}(n) \;\wedge\; \exists y\; \mathcal{P}(y) \;\wedge\; y \leq_m x \;\wedge |y|_{gr} = n \;\wedge \\
& \forall z\; (\mathcal{P}(z) \;\wedge\; z \leq_m x) \supset z \leq_m y
\end{aligned}$$

$maxDeg(n, x)$ iff the maximum degree of x is $|n|$.

Here we need to do some more work since the largest star which is a minor of x may be much larger than the maximum degree of the graph. The slightly involved construction is given in the appendix.

5 Arithmetic in Graph Orders

We define the ternary predicate version of arithmetic $(\mathbb{N}, plus, times)$ in the subgraph and minor orders. In order to do so, we need the following formulae:
$N(g)$ iff g is a graph representing a number in our chosen representation. Let us denote by n_g the number denoted by g.
$plus(x, y, z)$ iff $N(x), N(y), N(z)$ hold and $n_x + n_y = n_z$ is true.
$times(x, y, z)$ iff $N(x), N(y), N(z)$ hold and $n_x \times n_y = n_z$ is true.

As can be gathered from the notation, our choice of (the unique) representation for natural number i is N_i, and from Lemmas 1 and 3, this family is definable in subgraph and minor.

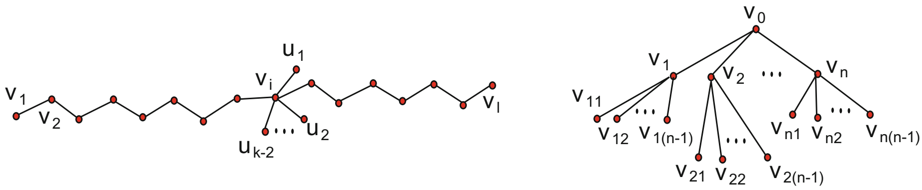

Fig. 5. Gadget graphs for addition and squaring.

To show the definability of the *plus* and *times* predicates, we will write out formulae using subgraph, but by Lemma 2 they can be transferred to minor.

Addition

$$\begin{aligned}
plus(k,l,m) :=& \mathcal{N}(k) \wedge \mathcal{N}(l) \wedge \mathcal{N}(m) \wedge \\
& (initial(k,l,m) \vee (N_3 \leq_s k \wedge N_3 \leq_s l \wedge \\
& \exists x\ starTail(k,l,x) \wedge plus2(m,x))); \text{ where} \\
starTail(k,l,x) :=& starTail'(k,l,x) \wedge \forall x'\ starTail'(k,l,x') \supset |x|_{gr} \leq_s |x'|_{gr} \\
starTail'(k,l,x) :=& \mathcal{T}(x) \wedge maxDeg(x) = k \wedge maxPath(x) = l \\
plus2(m,x) :=& \exists m'\ uc_s(|m'|_{gr}, x) \wedge uc_s(m, |m'|_{gr}) \\
initial(k,l,m) :=& (k = \emptyset \wedge m = l) \vee (l = \emptyset \wedge m = k) \vee \\
& (k = N_1 \wedge uc_s(m,l)) \vee (l = N_1 \wedge uc_s(m,k)) \vee \\
& (k = N_2 \wedge \exists m'\ uc_s(|m'|_{gr}, l) \wedge uc_s(m, |m'|_{gr})) \vee \\
& (l = N_2 \wedge \exists m'\ uc_s(|m'|_{gr}, k) \wedge uc_s(m, |m'|_{gr}))
\end{aligned}$$

When either k or l are strictly less than two, we hardcode the function value using *initial*.

When both are at least three, consider a tree with maximum degree k and maximum path l. A tree of least order with these properties is formed from a path by choosing some degree two vertex of the path v_i, adding $k-2$ new vertices $u_1, u_2, ..., u_{k-2}$ and adding the edges $u_1v_i, u_2v_i, ..., u_{k-2}v_i$ (see Fig. 5). The order of this tree is $k + l - 2$. This is captured in the formula *starTail* and in *plus*2 we add two to its cardinality to get $k + l$.

Multiplication

We will show instead that squaring is definable, multiplication is easily obtained via the formula

$$(n_1 + n_2)^2 = n_1^2 + n_2^2 + 2 \times n_1 \times n_2$$

square(n, m) iff $n, m \in \mathcal{N}$ and $|m| = |n|^2$

$$\begin{aligned}
square(n,m) :=& \mathcal{N}(n) \wedge \mathcal{N}(m) \wedge \exists z\ ntree(n,z) \wedge uc_s(|z|_{gr}, m) \wedge \\
& \forall y\ ntree(n,y) \supset |y|_{gr} \leq_s |z|_{gr}; \text{ where} \\
ntree(n,z) :=& tree(z) \wedge maxDeg(z) = n \wedge P_6 \not\leq_s z
\end{aligned}$$

There exists a tree t of maximum order whose maximum degree is n and maximum path is P_5. To see that t has order n^2+1, observe that the tree has depth three (Fig. 5) and the total number of vertices is $1+n+n\times(n-1)=1+n^2$.

Thus by Lemmas 1, 3 and the definability of addition and multiplication shown above we have :

Theorem 5. *First order arithmetic is definable in the subgraph and minor orders.*

6 Encoding Graph Orders in Arithmetic

We will show that the structures $(\mathcal{G}, \leq_s)$ and $(\mathcal{G}, \leq_i, P_3)$ can be interpreted in first order arithmetic. In order to do this, we define the following formulae:

1. $isGraph(x)$ iff x is a number which represents a graph.
2. $sameGraph(x, y)$ iff x and y are numbers which represent the same graph.
3. $subGraph(x, y)$ iff the graph represented by x is a subgraph of the graph represented by y.
4. $inSubGraph(x, y)$ iff the graph represented by x is an induced subgraph of the graph represented by y.
5. $P_3(x)$ iff x represents the graph P_3.

Lemma 4 (Definable Arithmetical Predicates). *The following predicates are definable in first order arithmetic (defining formulae in appendix):*

1. *$bit(i, x)$ iff the i^{th} bit of the binary representation of x is a 1.*
2. *$length(n, x)$ iff the length of the binary representation of x is n. We will denote this unique n by $|x|$.*
3. *$pow2(i, x)$ iff $x = 2^i$.*
4. *$rem(n, x, y)$ iff n is the remainder when x is divided by y; denoted $n = rem(x, y)$.*
5. *$div(n, x, y)$ iff n is the quotient when x is divided by y; denoted $n = x/y$.*
6. *$nchoose2(n, x)$ iff $x = \binom{n}{2}$ where $\binom{n}{2} = n \times (n-1)/2$.*

6.1 Encoding Graphs

Any graph on n vertices has $\binom{n}{2}$ possible pairs of vertices. By fixing an appropriate order on these pairs, we may interpret any number whose binary representation has $\binom{n}{2}+1$ bits as a graph on n vertices (we ignore the leading 1 since every binary representation has to start with a 1 except for the number 0). Let g be a graph on vertices $\{v_1, v_2, ..., v_n\}$. We define the ordering $\leq_e$ on tuples of vertices:
For $i < j, i' < j'$, $v_i v_j \leq_e v_{i'} v_{j'}$ iff $i < i'$, or $i = i'$ and $j \leq j'$.
Writing down the tuples in descending order, we get
$v_n v_{n-1}, v_n v_{n-2}, v_{n-1} v_{n-2}, v_n v_{n-3}, ..., v_3 v_1, v_2 v_1$. If we now replace the tuples by 0's for non-edges of g and 1's for edges and prefix a 1 to this string, we get a number m with bit length $\binom{n}{2}+1$ which we say represents the (isomorphism

class of) graph g. Note that as presented, there are multiple numbers which represent the same graph (upto isomorphism). We could choose the smallest of these to make the representation unique, instead we will stop at showing that there is a formula which identifies isomorphic graphs. We choose for the sake of completeness 0 as the unique number representing the empty graph. We denote the graph represented by x by g_x.

$$isGraph(x) := x = 0 \vee \exists n\ x = (1 + \binom{n}{2})$$

Since we have arithmetical predicates available, we can define a formula for the order of a graph. Then to assert that there is an edge from v_i to v_j in x (where $i < j$), note that the tuple $\{v_i, v_j\}$ occurs at bit position $(2ni - 2n - i^2 - i + 2j)/2 = n(n-1)/2 - (n-i)(n-i+1)/2 + j - i$. We can further use arithmetical predicates to define formulas such as:
$perm(x, n)$ iff x represents a permutation on $[n]$.
$applyperm(x, i, j, n)$ iff x is a permutation on $[n]$ and sends i to j for $i, j \in [n]$.

We can then define the isomorphism of graphs:

$$sameGraph(x, y) := \exists n\ |x| = |y| = 1 + \binom{n}{2}\ \wedge\ \exists z\ perm(z, n) \wedge$$
$$\forall i\ \forall j\ 1 \leq i < j \leq n \supset\ (edgeExists(x, i, j) \iff (\exists i' \exists j'\ applyPerm(z, i, i', n)$$
$$\wedge\ applyPerm(z, j, j', n)\ \wedge edgeExists(y, i', j')\ ))$$

The formula states that for x and y to represent the same graph, there must exist a permutation z such that for any tuple $\{v_i, v_j\}$ of vertices of x, $v_i v_j \in E(g_x)$ iff $v_{z(i)} v_{z(j)} \in E(g_y)$. Details are given in the appendix.

6.2 Subgraph and Induced Subgraph

$subGraph(x, y)$ iff x, y represent graphs and g_x is a subgraph of g_y.

$$subGraph(x, y) := isGraph(x)\ \wedge\ isGraph(y)\ \wedge\ |g_x| \leq |g_y| \wedge$$
$$\exists z\ sameGraph(y, z)\ \wedge\ \forall k\ 1 \leq k \leq |x|\ \supset\ (bit(k, x) \supset bit(k + |y| - |z|, z))$$

If x on vertices $u_1, u_2, ..., u_n$ is a subgraph of y on vertices $v_1, v_2, ..., v_m$ without regard for vertices, then there is a map $f : V(x) \to V(y)$ which witnesses it. Rename the vertices of y to get z by the map which sends $f(u_i)$ for any $u \in V(x)$ to v_{m-n+i} and fixes the other vertices. Then x is a subgraph of the graph on $v_m, v_{m-1} ..., v_{m-n+1}$ when considered with the labels.

Conversely, if the formula is true, the *sameGraph* predicate gives us a witnessing permutation using which we can define the map witnessing that x is a subgraph of y.

We can define induced subgraph by a small modification in the subgraph formula as follows:

$$inSubGraph(x, y) := isGraph(x)\ \wedge\ isGraph(y)\ \wedge\ |g_x| \leq |g_y| \wedge$$
$$\exists z\ sameGraph(y, z)\ \wedge\ \forall k\ 1 \leq k \leq |x|\ \supset\ (bit(k, x) \iff bit(k, z))$$

Defining the constant P_3
The formula $P_3(x)$ can be easily defined as the disjunction over the formulae $x = c$ where c is a number representing P_3 since there are only finitely many of them. Thus we have:

Theorem 6. *The structures $(\mathcal{G}, \leq_s)$ and $(\mathcal{G}, \leq_i, P_3)$ are definable in $(\mathbb{N}, +, \times)$.*

Theorem 7 (Wires [19]). *Arithmetic is definable in $(\mathcal{G}, \leq_i, P_3)$.*

Combining the above result with Theorems 5 and 6 we have:

Theorem 8. *The structures $(\mathcal{G}, \leq_s)$ and $(\mathcal{G}, \leq_i, P_3)$ are bi-interpretable with first order arithmetic. The structure $(\mathcal{G}, \leq_m)$ can encode $(\mathbb{N}, plus, times)$.*

7 Discussion and Future Work

7.1 Decidability and Descriptive Complexity

An obvious corollary of our results is that the theories of the orders considered are undecidable, but it is natural to ask what the decidable fragments are. One may consider various restrictions: syntactic ones such as the $\forall^*\exists^*$ fragment, subclasses of graphs such as trees $(\mathcal{T}, \leq_s)$ or restrict the order e.g. theory of the covering relation $\mathrm{Th}(\mathcal{G}, uc_s)$. There is much work on general frameworks for graph theory, especially extremal graph theory, whose focus is on homomorphisms. In particular Hatami's paper [6] on the undecidability of inequalities over homomorphism densities underlines the difficulty of answering general questions about graphs. If our interest is in only obtaining undecidability results, ideas of recursive inseparability and other techniques (see [5]) may be more apt.

We also note that there is a large body of work on descriptive complexity [7], which takes the graph-as-a-model point of view. How definable families in our approach compare with the above is a matter of interest. In particular, every constant can be defined in the subgraph order using the methods of Wires [19], just as they can in descriptive complexity.

7.2 Extensions, Interdefinability and Graph Theory

We do not know if subgraph is definable using minor or vice versa. However, if we add the predicate $sameSize(x, y)$ which stands for x and y have the same number of edges, we can define subgraph using minor as shown below:

Suppose that x is a subgraph of y. Then we can think of y as being built from x in two steps. In the first step, we add to x a number of isolated points to give x' such that $|x'|_{gr} = |y|_{gr}$. In the second step, we only add extra edges to x' to get y.
We can formalize the two step construction as follows:

$$x \leq_s y := \exists x' \; vertdesc(x', x) \;\wedge\; edgedesc(y, x'); \text{ where}$$
$$edgedesc(x, y) := y \leq_m x \;\wedge\; |x|_{gr} = |y|_{gr}$$
$$vertdesc(x, y) := y \leq_m x \;\wedge\; samesize(x, y)$$

Such extensions also have serious implications for the kind of graph theoretic statements that can be made. This is because, though the structure of graph orders is rich, they have limited access to the "inner structure" of a graph. For instance, it is not clear how minimum degree of a graph can be defined using graph orders. We already know that we can do arithmetic over the order of a graph. By adding the predicate $sameSize(x, y)$, we can do arithmetic over the size. Consequently, concepts of minimum and average degree can be expressed and theorems about them written in the extended language. We can capture the size of a graph using $sameSize$:

$$||x||_{gr} = n := \mathcal{N}(n) \ \wedge \ \exists y \, \mathcal{C}(y) \ |y|_{gr} = n \ \wedge \ sameSize(x, y)$$

Minimum degree of a graph

$$\begin{aligned} minDeg(x, y) :=& \mathcal{N}(x) \ \wedge \ \exists z \ deleteNeighbours(y, z) \\ & \wedge \ \forall z' \ deleteNeighbours(y, z') \supset ||z'||_{gr} \leq_s ||z||_{gr} \\ & \wedge \ x = ||z||_{gr}; \text{ where} \\ deleteNeighbours(y, z) :=& z \leq_s y \ \wedge \ hasIso(z) \ \wedge \ |z|_{gr} = |y|_{gr} \\ hasIso(x) :=& \exists y \ uc_s(x, y) \ \wedge \ |y|_{gr} <_s |x|_{gr} \end{aligned}$$

Average degree of a graph (integer ceiling)

$$\begin{aligned} \lceil AvgDeg(x, y) \rceil :=& \mathcal{N}(x) \ \wedge \ (||y||_{gr} \leq_s x \times |y|_{gr}) \\ & \wedge \ \forall z \ (\mathcal{N}(z) \wedge z <_s x) \supset (z \times |y|_{gr} <_s ||y||_{gr}) \end{aligned}$$

We can also define $\lfloor AvgDeg(x, y) \rfloor$ i.e. the floor instead of the ceiling defined above. Modifying the definition above by dividing by two gives us floor and ceiling versions of number of edges per vertex i.e. $\lfloor \epsilon(x, y) \rfloor$ and $\lceil \epsilon(x, y) \rceil$ respectively.

Theorem 9 (Diestel, Proposition 1.2.2 [3]). *Every graph g with at least one edge has a subgraph g' with $\delta(g') > \lfloor \epsilon(g') \rfloor \geq \lfloor \epsilon(g) \rfloor$ (where δ denotes minimum degree):*

$$\forall x \ \neg \mathcal{N}(x) \supset \exists y \ y \leq_s x \ \wedge \ \lfloor \epsilon(y) \rfloor <_s minDeg(y) \ \wedge \ \lfloor \epsilon(x) \rfloor \leq_s \lfloor \epsilon(y) \rfloor$$

7.3 Differences in Definability

From the work of Wires [19] it is known that all the graph families we defined in Lemmas 1 and 3 and many more are definable in $(\mathcal{G}, \leq_i, P_3)$. Thus they are definable in all three orders. But as we saw, while maximum degree was definable easily in subgraph, it takes more work in minor. Similarly, though cardinality is trivial in minor and subgraph, it seems to take much more work to define in induced subgraph. On the other hand, here is a predicate definable easily in induced minor which we do not know how to define in the other two :
$\alpha(n, x)$ iff $n \in N$ and $|n|$ is the independence number of x.

$$\alpha(n, x) := \mathcal{N}(n) \ \wedge \ n \leq_i x \ \wedge \ \forall y \ (\mathcal{N}(y) \ \wedge \ y \leq_i x) \supset y \leq_i n$$

Perhaps the most interesting direction of work lies in pinning down these differences, especially as pertains to the definability of predicates which are important from the point of view of graph theory and to determine what part of the"inner structure" of graphs can be determined by their relationships with other graphs. For this, we need to develop tools to prove indefinability, which is a challenging task.

Appendix

Proof of Lemma 2

Lemma 2: Let x_n be an tree with at most one degree 3 node and no node of degree 4 or more. Then for any other graph x_0, $x_n \leq_m x_0$ iff $x_n \leq_s x_0$.

We prove the result by induction on the number of contraction operations in transforming x_0 to x_n.

Base Case: There are no contraction operations, there is nothing to be done.

For the induction step there are two cases we consider:

Case 1 : x_n has no degree 3 node. Let x_n be a path $u_0, u_1, ..., u_m$. Let $o_1, .., o_n$ be the sequence of minor operations in normal form with x_i being obtained from x_{i-1} via operation o_i. o_n must be a contraction operation (else all operations are deletions and we are done). Therefore x_{n-1} is either a path of length $m+1$ or a graph such that $V(x_{n-1}) = V(x_n) \cup \{u'\}$ and there exists an i with $E(x_{n-1}) = E(x_n) \cup \{u'u_i\}$ or $E(x_{n-1}) = E(x_n) \cup \{u'u_i, u'u_{i+1}\}$. In all cases, we can delete an endpoint of x_{n-1} or u' respectively in order to obtain x_n. Thus there is a sequence $o_1, .., o_{n-1}, o'_n$ of operations (o'_n is a deletion) to obtain x_n from x_0. Since this sequence has a smaller number of contractions, by induction hypothesis, x_n is a subgraph of x_0.

Case 2 : x_n has exactly one degree three node. Let x_n consist of a degree 3 node u with paths p_1, p_2, p_3. As before, consider the sequence of minor operations. In one case x_{n-1} is a graph with a degree 3 node attached to three paths exactly one of which has length one more than previously. We can delete the end point of the appropriate path to get x_n from x_{n-1}. Another possibility is that x_{n-1} is a graph with a vertex $u' \notin V(x_n)$ such that u' is attached to either one or two adjacent points of one of the paths p_1, p_2, p_3. As before, we can delete u' to get x_n from x_{n-1}. Then by induction hypothesis x_n is a subgraph of x_{n-1}.

Proof of Maximum Degree Definability in Theorem 4

Theorem 4: Connectivity, maximum degree and maximum path length are definable in minor.

In order to apply observation 2, we construct the following family:

$\mathcal{S}\cup\mathcal{N}(x)$ iff x is formed by addition of some arbitrary number of isolated vertices to a star.

$$\begin{aligned}\mathcal{S}\cup\mathcal{N}(x) :=& \mathcal{F}(x) \wedge \exists y\ hasStarComp(y,x) \wedge onlystar(x,y)\\ &\text{where}\\ hasStarComp(y,x) :=& \mathcal{S}(y) \wedge y \leq_m x \wedge \forall z\ conn(z) \wedge z \leq_m x \supset z \leq_m y\\ onlyStar(x,y) :=& \forall x'\ \mathcal{F}(x') \wedge |x'|_{gr} = |x|_{gr} \wedge uc_m(x',x) \supset\\ & \forall y'\ (conn(y') \wedge y' \leq_m x') \supset uc_m(|y'|_{gr}, |y|_{gr})\end{aligned}$$

onlyStarComp asserts that y is a star minor of x and in addition, every connected minor of x is also a minor of y. To fulfill this condition, x has to contain y as a connected component.

onlyStar asserts that any forest x' which is formed by adding an edge to x (by observation 2) has the property that all its connected minors have order one more than the order of y.

Clearly, any graph formed by adding isolated vertices to a star has these properties.

$\mathcal{S}\cup\mathcal{N}subGraph$ states that there is a subgraph y of x which is a $\mathcal{S}\cup\mathcal{N}$ of same order as x. Note that for $S_n \cup N_m$ and $S_{n'} \cup N_{m'}$ with $n+m = n'+m'$, $S_n \cup N_m \leq_m S_{n'} \cup N_{m'}$ iff $n \leq n'$. Thus maximal y satisfying the formula $\mathcal{S}\cup\mathcal{N}subGraph$ contains the largest star occuring as a subgraph of x. We extract the star from this object to obtain the maximum degree of x.

Proof of Lemma 4

Lemma 4: The following predicates are definable in first order arithmetic:

1. $nchoose2(n,x)$ iff $x = \binom{n}{2}$ where $\binom{n}{2} = n \times (n-1)/2$.
   $nchoose2(n,x) := 2 \times x + n = n^2$.

2. $div(n,x,y)$ iff n is the quotient when x is divided by y; denoted $n = x/y$.
   $div(n,x,y) := \exists z\ x = y \times n + z \wedge z < y$

3. $rem(n,x,y)$ iff n is the remainder when x is divided by y; denoted $n = rem(x,y)$. $rem(n,x,y) := \exists z\ x = y \times z + n \wedge n < y$
   We note that the exponentiation relation $x^y = z$ is known to be definable in arithmetic (see [13]).
4. $pow2(i,x)$ iff $x = 2^i$.
   $pow2(i,x) := \exists y\ y = 2 \wedge y^i = x$

5. $bit(i,x)$ iff the i^{th} bit of the binary representation of x is a 1.
   $bit(i,x) := rem(x,2^i) = rem(x,2^{i-1})$

6. $length(n,x)$ iff the length of the binary representation of x is n. We will denote this unique n by $|x|$.
   $length(n,x) := bit(n,x) \wedge \forall n'\ n < n' \supset \neg bit(n',x)$

Details of Subsection 6.1

$graphOrder(x, n)$ iff $n \in \mathcal{N}$ and the order of x is $|n|$.

$$graphOrder(x, n) := isGraph(x) \wedge 2 \times |x| = 2 + n \times (n - 1)$$

We will denote by $|g_x|$ the order of the graph represented by x.
$edgeExists(x, i, j)$ iff x denotes a graph and $v_i v_j \in E(g_x)$.

$$\begin{aligned} edgeExists(x, i, j) :=& \exists n\ graphOrder(x, n) \wedge \\ & ((1 \leq i < j \leq n \wedge bit((2ni - 2n - i^2 - i + 2j)/2, x)) \\ & \vee (1 \leq j < i \leq n \wedge bit((2nj - 2n - j^2 - j + 2i)/2, x)) \end{aligned}$$

By doing some counting, we can see that the tuple $\{v_i, v_j\}, i < j$ occurs at bit position $(2ni - 2n - i^2 - i + 2j)/2 = n(n-1)/2 - (n-i)(n-i+1)/2 + j - i$.

Defining Permutations and Isomorphism. Any permutation of vertices of a vertex labelled graph induces a permutation on the edges of a graph. To identify all numbers which represent the same graph under our encoding, we will need to represent permutations on $[n]$ and their actions.

$perm(x, n)$ iff x represents a permutation on $[n]$.

$$\begin{aligned} perm(x, n) :=& |x| = 1 + n \times \lfloor log(n) \rfloor \wedge \forall i\ 1 \leq i \leq n\ \exists! j\ 1 \leq j \leq n \\ & i = (rem(x, 2^{j\ |n|}) - rem(x, 2^{(j-1)\ |n|}))/2^{(j-1)|n|} \end{aligned}$$

We represent a permutation by a bit string which is of length $n \times \lfloor log(n) \rfloor + 1$, note that $\lfloor log(n) \rfloor$ is the same as $|n|$ i.e. the length of the string n. The most significant digit is to be ignored, after which every block of $\lfloor log(n) \rfloor$ bits represents a number from 1 to n. In addition, every such block must be unique (in order to guarantee that it is a permutation). The permutation sends $i \in [n]$ to the number represented by the i^{th} block from the left. The formula checks that every $i \in [n]$ is obtained from a unique block $j \in [n]$.

$applyperm(x, i, j, n)$ iff x is a permutation on $[n]$ and sends i to j for $i, j \in [n]$.

$$\begin{aligned} applyperm(x, i, j, n) :=& perm(x, n) \wedge \\ & (rem(x, 2^{(n-i+1)|n|}) - rem(x, 2^{(n-i)|n|}))/2^{(n-i)|n|} = j \end{aligned}$$

We can now define the isomorphism of graphs:

$$\begin{aligned} & sameGraph(x, y) := \exists n\ |x| = |y| = 1 + \binom{n}{2} \wedge \exists z\ perm(z, n) \wedge \\ & \forall i\ \forall j\ 1 \leq i < j \leq n \supset (edgeExists(x, i, j) \iff (\exists i' \exists j'\ applyPerm(z, i, i', n) \\ & \wedge applyPerm(z, j, j', n) \wedge edgeExists(y, i', j')\ )) \end{aligned}$$

The formula states that for x and y to represent the same graph, there must exist a permutation z such that for any tuple $\{v_i, v_j\}$ of vertices of x, $v_i v_j \in E(g_x)$ iff $v_{z(i)} v_{z(j)} \in E(g_y)$.

References

1. Bienstock, D., Langston, M.A.: Algorithmic implications of the graph minor theorem. Handb. Oper. Res. Manag. Sci. **7**, 481–502 (1995)
2. Bodlaender, H.L.: Treewidth: algorithmic techniques and results. In: Privara, I., Ruzicka, P. (eds.) MFCS '97. LNCS, vol. 1295, pp. 19–36. Springer, Berlin (2003)
3. Diestel, R.: Graph Theory. Springer, Berlin (2005)
4. Enderton, H.: A Mathematical Introduction to Logic. Academic press, Waltham (2001)
5. Ershov, Y.L., Lavrov, I.A., Taimanov, A.D., Taitslin, M.A.: Elementary theories. Russ. Math. Surv. **20**(4), 35–105 (1965)
6. Hatami, H., Norine, S.: Undecidability of linear inequalities in graph homomorphism densities. J. Am. Math. Soc. **24**(2), 547–565 (2011)
7. Immerman, N.: Descriptive Complexity. Springer Science & Business Media, Berlin (2012)
8. Ježek, J., McKenzie, R.: Definability in substructure orderings, iv: finite lattices. Algebra Univers. **61**(3–4), 301–312 (2009)
9. Ježek, J., McKenzie, R.: Definability in substructure orderings, i: finite semilattices. Algebra Univers. **61**(1), 59–75 (2009)
10. Ježek, J., McKenzie, R.: Definability in substructure orderings, iii: finite distributive lattices. Algebra Univers. **61**(3–4), 283–300 (2009)
11. Ježek, J., McKenzie, R.: Definability in substructure orderings, ii: finite ordered sets. Order **27**(2), 115–145 (2010)
12. Kach, A.M., Montalban, A.: Undecidability of the theories of classes of structures. J. Symbolic Logic **79**(04), 1001–1019 (2014)
13. Kaye, R.: Models of Peano Arithmetic. Oxford University Press, USA (1991)
14. Kudinov, O.V., Selivanov, V.L.: Definability in the infix order on words. In: Diekert, V., Nowotka, D. (eds.) DLT 2009. LNCS, vol. 5583, pp. 454–465. Springer, Heidelberg (2009)
15. Kudinov, O.V., Selivanov, V.L., Yartseva, L.V.: Definability in the subword order. In: Ferreira, F., Löwe, B., Mayordomo, E., Mendes Gomes, L. (eds.) CiE 2010. LNCS, vol. 6158, pp. 246–255. Springer, Heidelberg (2010)
16. Kunos, Á.: Definability in the embeddability ordering of finite directed graphs. Order **32**(1), 117–133 (2015)
17. Kuske, D.: Theories of orders on the set of words. RAIRO-Theor. Inform. Appl. **40**(01), 53–74 (2006)
18. Robertson, N., Seymour, P.D.: Graph minors. xx. wagner's conjecture. J. Comb. Theor. B **92**(2), 325–357 (2004)
19. Wires, A.: Definability in the substructure ordering of simple graphs (2012). http://www.math.uwaterloo.ca/awires/SimpleGraphs.pdf