Testing first-order properties for subclasses of sparse graphs∗

Zdeněk Dvořák†   Daniel Král††   Robin Thomas‡

Abstract

We present a linear-time algorithm for deciding first-order logic (FOL) properties in classes of graphs with bounded expansion, a notion recently introduced by Nešetřil and Ossona de Mendez. Many natural classes of graphs have bounded expansion: graphs of bounded tree-width, all proper minor-closed classes of graphs, graphs of bounded degree, graphs with no subgraph isomorphic to a subdivision of a fixed graph, and graphs that can be drawn in a fixed surface in such a way that each edge crosses at most a constant number of other edges. We deduce that there is an almost linear-time algorithm for deciding FOL properties in classes of graphs with locally bounded expansion.

More generally, we design a dynamic data structure for graphs belonging to a fixed class of graphs of bounded expansion. After a linear-time initialization the data structure allows us to test an FOL property in constant time, and the data structure can be updated in constant time after addition/deletion of an edge, provided the list of possible edges to be added is known in advance and their simultaneous addition results in a graph in the class. All our results also hold for relational structures and are based on the seminal result of Nešetřil and Ossona de Mendez on the existence of low tree-depth colorings.

1 Introduction

A celebrated theorem of Courcelle [1] states that for every integer k ≥ 1 and every property Π definable in monadic second-order logic (MSOL) there is a linear-time algorithm to decide whether a graph of tree-width at most k satisfies Π. While the theorem itself is probably not useful in practice because of the large constants involved, it does provide an easily verifiable condition that a certain problem is (in theory) efficiently solvable. Courcelle’s result led to the

∗A preliminary version of this paper appeared in FOCS 2010.
†Institute for Theoretical Computer Science (ITI), Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic. E-mail: {rakdver,kral}@kam.mff.cuni.cz. Institute for Theoretical Computer Science (ITI) is supported by Ministry of Education of Czech Republic as projects 1M0545.
‡School of Mathematics, Georgia Institute of Technology, Atlanta, GA. Partially supported by NSF under No. DMS-0739366.
development of a whole new area of algorithmics, known as algorithmic meta-
theorems; see the survey [19] or the forthcoming one [18]. For specific problems
there is very often a more efficient implementation, for instance following the
axiomatic approach of [29].

While the class of graphs of tree-width at most \(k\) is fairly large, it does
not include some important graph classes, such as planar graphs or graphs of
bounded degree. Courcelle’s theorem cannot be extended to these classes unless
P=NP, because testing 3-colorability is NP-hard for planar graphs of maximum
degree at most four [17] and yet 3-colorability is expressible in monadic second
order logic.

Thus in an attempt at enlarging the class of input graphs, we have to restrict
the set of properties to be tested. One of the first result in this direction was
a linear-time algorithm of Eppstein [11, 12] for testing the existence of a fixed
subgraph in planar graphs. He then extended his algorithm to minor-closed
classes of graphs with locally bounded tree-width [13]. Since testing containment
of a fixed subgraph can be expressed in first order logic (FOL) by a \(\Sigma_1\)-sentence,
this can be regarded as a precursor to FOL formula testing. Prior to our work,
the following were the most general results:

- a linear-time algorithm of Seese [30] to test FOL properties of graphs of
  bounded degree,
- a linear-time algorithm of Frick and Grohe [15] for deciding FOL properties
  of planar graphs,
- an almost linear-time algorithm of Frick and Grohe [15] for deciding FOL
  properties for classes of graphs with locally bounded tree-width,
- a fixed parameter algorithm of Dawar, Grohe and Kreutzer [2] for deciding
  FOL properties for classes of graphs locally excluding a minor, and
- a linear-time algorithm of Nešetřil and Ossona de Mendez [22] for deciding
  \(\Sigma_1\)-properties for classes of graphs with bounded expansion.

Our main theorem gives a common generalization of these five results. In order
to state it we need a couple of definitions. For an integer \(r \geq 0\), a graph \(G\) is
an \(r\)-shallow minor of a graph \(H\) if \(G\) can be obtained from a subgraph of \(H\)
by contracting vertex-disjoint subgraphs of radii at most \(r\) (and removing the
resulting loops and parallel edges). A class \(\mathcal{G}\) of graphs has bounded expansion
if there exists a function \(f : \mathbb{N} \to \mathbb{R}^+\) such that for every integer \(r \geq 0\) every
\(r\)-shallow minor of a member of \(\mathcal{G}\) has edge-density at most \(f(r)\). (The edge-
density of a graph \(G\) is \(|E(G)|/|V(G)|\).) A preliminary version of our main
theorem can be stated as follows.

**Theorem 1.** Let \(\mathcal{G}\) be a class of graphs with bounded expansion, and let \(\Pi\) be
a first-order property of graphs. Then there exists a linear-time algorithm that
correctly decides whether a graph from \(\mathcal{G}\) satisfies \(\Pi\).
In fact, we prove a more general theorem: we prove there exists a linear-time algorithm for $L$-structures “guarded” by a member of $\mathcal{G}$, and we design several data structures that allow the $L$-structure to be modified and support FOL property testing in constant time.

Using known techniques we derive the following corollary from Theorem 1. A class $\mathcal{G}$ of graphs has locally bounded expansion if there exists a function $g : \mathbb{N} \times \mathbb{N} \to \mathbb{R}^+$ such that for every two integers $d, r \geq 0$, for every graph $G \in \mathcal{G}$ and for every $v \in V(G)$, every $r$-shallow minor of the $d$-neighborhood of $v$ in $G$ has edge-density at most $g(d, r)$. We say that there exists an almost linear-time algorithm to solve a problem $\Pi$ if for every $\varepsilon > 0$ there exists an algorithm to solve $\Pi$ with running time $O(n^{1+\varepsilon})$, where $n$ is the size of the input instance.

**Corollary 2.** Let $\mathcal{G}$ be a class of graphs with locally bounded expansion, and let $\Pi$ be a first-order property of graphs. Then there exists an almost linear-time algorithm that correctly decides whether a graph from $\mathcal{G}$ satisfies $\Pi$.

We announced our results in the survey paper [7]. Dawar and Kreutzer [3] posted an independent proof of Theorem 1 and a proof of Corollary 2 for the more general classes of nowhere-dense graphs (introduced below). However, the proofs in [3] are incorrect. A correct proof of Theorem 1 different from ours, appears in [18].

Thus it remains an interesting open problem whether Corollary 2 can be generalized to the more general classes of nowhere-dense graphs. This is of substantial interest from the point of view of fixed parameter tractability, because nowhere density of classes of graphs gives a natural limitation (subject to a widely believed complexity-theory assumption). Indeed, we prove the following in Theorem 5 below. Let $L$ be a language consisting of one binary relation symbol and let $\mathcal{G}$ be a class of graphs closed under taking subgraphs that is not nowhere dense. We prove that if testing whether an input graph from $\mathcal{G}$ satisfies a given $\Sigma_1$-sentence $\varphi$ is fixed parameter tractable when parametrized by the size of $\varphi$, then FPT=W[1].

In the rest of this section we introduce terminology and state all our results.

### 1.1 Logic theory definitions

Our logic terminology is standard, except for the following. All function symbols have arity one, and hence all functions are functions of one variable. If $L$ is a language, then an $L$-term is simple if it is a variable or it is of the form $f(x)$ where $f$ is a function symbol and $x$ is a variable. An $L$-formula is simple if all terms appearing in it are simple. The rest of our logic terminology is standard, and so readers familiar with it may skip the rest of this subsection.

A language $L$ consists of a disjoint union of a finite set $L^r$ of relation symbols and a finite set $L^f$ of function symbols. Each relation symbol $R \in L^r$ is associated with an integer $a(R) \geq 0$, called the arity of $R$. In this paper all function symbols have arity one.

If $L$ is a language, then an $L$-structure $A$ is a triple $(V, (R^A)_{R \in L^r}, (f^A)_{f \in L^f})$ consisting of a finite set $V$ and for each $m$-ary relation symbol $R \in L^r$ a set
$R^A \subseteq V^m$, the interpretation of $R$ in $A$, and for each function symbol $f \in L^f$ a function $f^A : V \to V$ of one variable, the interpretation of $f$ in $A$. We define $V(A) := V$. For example, graphs may be regarded as $L$-structures, where $L$ is the language consisting of a single binary relation. We define the size $|A|$ of $A$ to be $|V(A)| + \sum_{R \in L^r} |R^A| + |L^f| |V(A)|$. A language $L'$ extends a language $L$ if every function symbol of $L$ is a function symbol of $L'$ and the same holds for relation symbols. If a language $L'$ extends a language $L$, $A$ is an $L$-structure and $A'$ is an $L'$-structure such that $V(A) = V(A')$ and $A$ and $A'$ have the same interpretations of symbols of $L$, then we say that $A'$ is an expansion of $A$.

Assume that we have an infinite set of variables. An $L$-term is defined as follows:

1. each variable is an $L$-term, and
2. if $f \in L^f$ and $t$ is an $L$-term, then $f(t)$ is an $L$-term.

Each $L$-term is obtained by a finite number of applications of these two rules. We say that an $L$-term is simple if it is a variable or is of the form $f(x)$ where $f \in L^f$ and $x$ is a variable. A term $t$ appears in a term $t'$ if either $t = t'$ or $t = f(t'')$ for some $t \in L^f$ and $t$ appears in $t''$.

An atomic $L$-formula $\varphi$ is either the symbol $\top$ (which represents a tautology); or its negation $\bot$; or $R(t_1, \ldots, t_m)$, where $R$ is an $m$-ary relation symbol of $L$ and $t_1, \ldots, t_m$ are $L$-terms; or $t_1 = t_2$, where $t_1$ and $t_2$ are $L$-terms. A term $t$ appears in $\varphi$ if it appears in one of the terms $t_1, \ldots, t_m$. An $L$-formula is defined recursively as follows: every atomic $L$-formula is an $L$-formula, and if $\varphi_1$ and $\varphi_2$ are $L$-formulas and $x$ is a variable, then $\neg \varphi_1$, $\varphi_1 \lor \varphi_2$, $\varphi_1 \land \varphi_2$, $\exists x \varphi_1$ and $\forall x \varphi_1$ are $L$-formulas. Every $L$-formula is obtained by a finite application of these rules. We write $t_1 \neq t_2$ as a shortcut for $\neg(t_1 = t_2)$.

A term $t$ appears in an $L$-formula $\varphi_1 \lor \varphi_2$ if it appears in $\varphi_1$ or $\varphi_2$, and we define appearance for the other cases analogously. An $L$-formula is simple if all terms appearing in it are simple. A variable $x$ appears freely in an $L$-formula $\varphi$ if either $\varphi$ is atomic and $x$ appears in $\varphi$; or $\varphi = \varphi_1 \lor \varphi_2$ or $\varphi = \varphi_1 \land \varphi_2$ and $x$ appears freely in at least one of the formulas $\varphi_1$ and $\varphi_2$; or $\varphi = \exists y \varphi'$ or $\varphi = \forall y \varphi'$, $x$ is distinct from $y$ and $x$ appears freely in $\varphi'$. Occurrences of $x$ in the formula $\varphi$ not inside the scope of a quantifier bounding $x$, i.e., those that witness that $x$ appears freely in $\varphi$, are called free and the variables that appear freely in $\varphi$ are also referred to as free variables. If $\varphi$ is a formula, then the notation $\varphi(x_1, \ldots, x_n)$ indicates that all variables that appear freely in $\varphi$ are among $x_1, \ldots, x_n$. An $L$-sentence is an $L$-formula such that no variable appears freely in it. If $\varphi(x_1, \ldots, x_n)$ is an $L$-formula and $A$ is an $L$-structure, then for $v_1, \ldots, v_n \in V(A)$, we define $A \models \varphi(v_1, \ldots, v_n)$ in the usual way. We denote the length of a formula $\varphi$ by $|\varphi|$.

1.2 Classes of sparse graphs

All graphs considered in this paper are simple and finite. The notion of a class of graphs of bounded expansion was introduced by Nešetřil and Ossona de Mendez.
in [20] and in the series of journal papers [21, 22, 23]. Examples of classes of graphs with bounded expansion include proper minor-closed classes of graphs, classes of graphs with bounded maximum degree, classes of graphs excluding a subdivision of a fixed graph, classes of graphs that can be embedded on a fixed surface with bounded number of crossings per each edge and many others, see [25]. Many structural and algorithmic properties generalize from proper minor-closed classes of graphs to classes of graphs with bounded expansion, see [7, 26].

A class $G$ of graphs is nowhere-dense if
\[
\lim_{r \to \infty} \limsup_{k \to \infty} \frac{\log |E(G)|}{\log |V(G)|} \leq 1,
\]
where the supremum is taken over all graphs $G$ on at least $k$ vertices such that $G$ is an $r$-shallow minor of a member of $G$. It can be shown that every class of graphs with (locally) bounded expansion is nowhere-dense [24], but the converse is false. One can also define a “locally nowhere-dense” class of graphs, but it turns out that such classes are nowhere-dense [24].

If $L$ is a language, then the Gaifman graph of an $L$-structure $A$ is the undirected graph $G_A$ with vertex set $V(G_A) = V(A)$ and an edge between two vertices $a, b \in V(A)$ ($a \neq b$) if and only if there exist $R \in L^r$ and a tuple $(a_1, \ldots, a_r) \in R^A$ such that $a, b \in \{a_1, \ldots, a_r\}$ or there exists a function $f \in L_f$ such that $b = f(a)$ or $a = f(b)$. We say that the relational structure $A$ is guarded by a graph $G$ with vertex set $V(G) = V(A)$ if $G_A \subseteq G$. Observe that if $G$ belongs to a class of graphs with bounded expansion, then every subgraph of $G$ has a vertex of bounded degree, and hence the number of complete subgraphs of $G$ is linear in $|V(G)|$ by a result of [31]. It follows that the size $|A|$ of any $L$-structure $A$ guarded by a graph belonging to a fixed class of graphs with bounded expansion is $O(|V(A)|)$.

1.3 Our results

We first state versions of Theorem 1 and Corollary 2 for $L$-structures.

**Theorem 3.** Let $G$ be a class of graphs with bounded expansion, $L$ a language and $\varphi$ an $L$-sentence. There exists a linear-time algorithm that decides whether an $L$-structure guarded by a graph $G \in G$ satisfies $\varphi$.

**Corollary 4.** Let $G$ be a class of graphs with locally bounded expansion, $L$ a language and $\varphi$ an $L$-sentence. There exists an almost linear-time algorithm that decides whether an $L$-structure guarded by a graph $G \in G$ satisfies $\varphi$.

Our approach differs from the methods used to prove the results from [2, 15, 30] mentioned above and is based on a seminal result of Nešetřil and Ossona de Mendez [21] on the existence of low tree-depth colorings for graphs with bounded expansion, stated below as Theorem 7.

We also consider dynamic setting and design the following data structures, where the first can be viewed as a dynamic version of Theorem 3. In the descriptions below we use $n$ to denote the number of vertices of the graph $G$. 
for every class $G$ of graphs with bounded expansion, a language $L$ and an $L$-sentence $\varphi$, a data structure that is initialized with a graph $G \in G$ and an $L$-structure $A$ guarded by $G$ in time $O(n)$ and supports:

- adding a tuple to a relation of $A$ in time $O(1)$ provided $A$ stays guarded by $G$,
- removing a tuple from a relation of $A$ in time $O(1)$,
- answering whether $A \models \varphi$ in time $O(1)$,

for every class $G$ of graphs with bounded expansion, an integer $d_0$ and a language $L$, a data structure that is initialized with a graph $G \in G$ and an $L$-structure $A$ guarded by $G$ in time $O(n)$ and supports:

- adding a tuple to a relation of $A$ in time $O(1)$ provided $A$ stays guarded by $G$,
- removing a tuple from a relation of $A$ in time $O(1)$,
- answering whether $A \models \varphi$ for any $\Sigma_1$-sentence $\varphi$ with at most $d_0$ variables in time $O(|\varphi|)$ and outputting one of the satisfying assignments, and

if $G$ is only a class of nowhere-dense graphs, then for every $\varepsilon > 0$ the data structure is initialized in time $O(n^{1+\varepsilon})$ and supports:

- adding a tuple to a relation of $A$ in time $O(n^\varepsilon)$ provided $A$ stays guarded by $G$,
- removing a tuple from a relation of $A$ in time $O(n^\varepsilon)$, and
- answering whether $A \models \varphi$ for any $\Sigma_1$-sentence $\varphi$ with at most $d_0$ variables in time $O(|\varphi|)$ and if so, outputting one of the satisfying assignments.

The second of these data structures is needed in our linear-time algorithm for 3-coloring triangle-free graphs on surfaces [10], also see [8].

1.4 A hardness result

Theorem 1 and Corollary 2 fall within the realm of fixed parameter tractability (FPT). We say that a decision problem $\Pi$ parametrized by a parameter $t$ is fixed parameter tractable if there exists an algorithm for $\Pi$ with running time $O(f(t)n^c)$, where $n$ is the size of the input, $f$ is an arbitrary function and $c$ is a constant independent of $t$. Analogously to the polynomial hierarchy starting with the classes P and NP, there exists a hierarchy of classes $\text{FPT} \subseteq \text{W}[1] \subseteq \text{W}[2] \subseteq \cdots$ of parametrized problems, where $\text{FPT}$ is the class of problems that are fixed parameter tractable. See e.g. [5, 14, 27] for more details.

Theorem 5. Let $G$ be a class of graphs closed under taking subgraphs. If $G$ is not nowhere-dense and the problem of deciding $\Sigma_1$-properties in $G$ is fixed parameter tractable, then $\text{FPT}=\text{W}[1]$. 

6
Proof. An \(r\)-subdivision of a graph \(G\) is the graph obtained from \(G\) by subdividing every edge exactly \(r\) times. Since \(G\) is not nowhere-dense, there exists an integer \(r\) such that \(G\) contains an \(r\)-subdivision of every graph \([24]\). By hypothesis there exists an FPT algorithm \(A\) to decide the existence of an \(r\)-subdivision of the complete graph \(K_m\) in an input graph from \(G\), where the latter problem is parametrized by \(m\). This implies that testing the existence of a complete subgraph of order \(m\) is fixed parameter tractable for general graphs \(G\), because it is equivalent to testing whether the \(r\)-subdivision of \(G\) has a subgraph isomorphic to the \(r\)-subdivision of \(K_m\), and the latter can be tested using the algorithm \(A\). But testing the existence of a \(K_m\) subgraph is a well-known W[1]-complete problem \([4]\), and hence FPT=W[1], as desired. \(\square\)

Dawar and Kreutzer \([3]\) proved the related result that if \(G\) fails to be nowhere dense in an “effective” way and deciding FOL properties in \(G\) is fixed parameter tractable, then FPT=AW[\(*\)].

2 Classes of graphs with bounded expansion

In this section, we survey results on classes of graphs with bounded expansion and classes of nowhere-dense graphs that we need in the paper. We say that the expansion of a graph \(G\) is bounded by a function \(g: \mathbb{N} \to \mathbb{R}^+\) if for every integer \(r \geq 0\) every \(r\)-shallow minor of \(G\) has edge-density at most \(g(r)\). We say that a class of graphs \(G\) has expansion bounded by \(g\) if every member of \(G\) has expansion bounded by \(g\). Thus \(G\) is a class of graphs of bounded expansion if and only if its expansion is bounded by some function \(\mathbb{N} \to \mathbb{R}^+\).

A rooted forest \(F\) is a directed graph such that every weak component is an out-branching (a rooted tree with every edge directed away from the root). The depth of \(F\) is the number of vertices of a longest directed path in \(F\). The closure of \(F\) is the undirected graph with vertex-set \(V(F)\) and edge-set all pairs of vertices joined by a directed path in \(F\). The tree-depth of an (undirected) graph \(G\) is the smallest integer \(s\) such that \(G\) is a subgraph of the closure of a rooted forest of depth \(s\). For an integer \(d \geq 1\) a vertex coloring of a graph \(G\) is a low tree-depth coloring of order \(d\) if for every \(s = 1, 2, \ldots, d\) the union of any \(s\) color classes induces a subgraph of \(G\) of tree-depth at most \(s\). Thus, in particular, every low tree-depth coloring of order \(d\) of \(G\) is a proper coloring of \(G\).

The existence of a low tree-depth coloring with bounded number of colors is one of the major structural results for graphs with bounded expansion. Before we state this result formally, we have to introduce more definitions.

Let \(D\) be a directed graph, and and let \(G'\) be the graph obtained from the underlying undirected graph of \(D\) by adding all edges \(xy\) such that:

- there exists a vertex \(z\) such that \(D\) contains an edge oriented from \(x\) to \(z\) and an edge oriented from \(z\) to \(y\) (transitivity), or
- there exists a vertex \(z\) such that \(D\) contains an edge oriented from \(x\) to \(z\) and an edge oriented from \(y\) to \(z\) (fraternity).
We call $G'$ the augmentation of $D$. The following is a mild strengthening of a result of Nešetřil and Ossona de Mendez \[21\], but the proof is adequate.

**Theorem 6.** There exist functions $f_1$ and $f_2$ with the following property. Let $D$ be a directed graph with maximum in-degree at most $\Delta$ such that its underlying undirected graph has expansion bounded by $g$, and let $G'$ be the augmentation of $D$. Then the expansion of $G'$ is bounded by the function $g'(r) = f_1(g(f_2(r)), \Delta)$.

Let $G$ be a graph from a class $\mathcal{G}$ with bounded expansion. We denote by $\nabla_0(G)$ the maximum edge density among the subgraphs of $G$. Thus every subgraph of $G$ has a vertex of degree at most $2\nabla_0(G)$, and hence $G$ has an orientation with maximum in-degree at most $2\nabla_0(G)$. Consider the following sequence of graphs: Let $G_0 := G$ and assume that we have constructed $G_0, G_1, \ldots, G_{k-1}$. Let $D$ be a directed graph of maximum in-degree at most $2\nabla_0(G_{k-1}) + 2$ such that $G_{k-1}$ is its underlying undirected graph, and let $G_k$ be the augmentation of $D$. We say that $G_k$ is a $k$-th augmentation of $G$. By Theorem 6 the class $\mathcal{G}_k$ consisting of the graphs $G_k$ obtained as described above starting from graphs in $\mathcal{G}$ has bounded expansion. It follows that $2\nabla_0(G_k) \leq d$ for some constant $d$ depending only on the class $\mathcal{G}$ and $k$, and thus $G_k$ is $d$-degenerate. In particular, the chromatic number of a $k$-th augmentation of a graph $G \in \mathcal{G}$ is bounded by a constant that depends only on $\mathcal{G}$.

If $G$ is an orientation of a graph, then any proper coloring of its $(3d^2 + 1)$-th augmentation $G'$ of $G$ is a low tree-depth coloring of order $d$ \[21\], also see \[7\] for further details. Moreover, the subgraph $H$ of $G'$ induced by the vertices of any $s \leq d$ color classes contains the underlying undirected graph of a rooted forest $F$ of depth at most $s$ such that $H$ is a subgraph of the closure of $F$. We refer to this property of the augmentation as depth-certifying and call $F$ a depth-certifying forest. The next theorem from \[21\] follows.

**Theorem 7.** Let $\mathcal{G}$ be a class of graphs with bounded expansion and $d \geq 1$ an integer. There exist integers $k$ and $K$ such that any proper coloring of a $k$-th augmentation of a graph $G \in \mathcal{G}$ is a low tree-depth coloring of order $d$ and the $k$-th augmentation is depth-certifying. In particular, $G$ has a low tree-depth coloring of order $d$ with at most $K$ colors. Moreover, such a coloring of $G \in \mathcal{G}$ and the corresponding depth-certifying forests can be found in linear time.

Similarly, for classes of nowhere dense graphs, we obtain \[24\].

**Theorem 8.** Let $\mathcal{G}$ be a class of nowhere-dense graphs and $d \geq 1$ an integer. There exists $k$ such that any proper coloring of a $k$-th augmentation of a graph $G \in \mathcal{G}$ is a low tree-depth coloring of order $d$ and the $k$-th augmentation is depth-certifying. In particular, for every $\epsilon > 0$ the graph $G$ has a low tree-depth coloring of order $d$ with at most $O(n^\epsilon)$ colors. Moreover, such a coloring of $G \in \mathcal{G}$ and the corresponding depth-certifying forests can be found in almost linear time.
3 Deciding FOL properties in linear time

In this section, we prove Theorem\ref{thm:linear-time}. We start with a lemma which allows us to remove quantifiers from an FOL formula (Lemma\ref{lem:quantifier-elimination}). However, we need more definitions. Let $L$ be a language and let $X$ be a set of $L$-terms. An $X$-template $T$ is a rooted forest with vertex set $V(T)$ equipped with a mapping $\alpha_T : X \to V(T)$ such that $\alpha_T^{-1}(w) \neq \emptyset$ for every vertex $w$ of $T$ with no descendants. If $\varphi$ is a quantifier-free $L$-formula, then a $\varphi$-template is an $X$-template where $X$ is the set of all terms appearing in $\varphi$. The depth of $T$ is the maximum depth of a tree of $T$. Two $X$-templates $T$ and $T'$ are isomorphic if there exists a bijection $f : V(T) \to V(T')$ such that

- $f$ is an isomorphism of $T$ and $T'$ as rooted forests; in particular, $w$ is a root of $T$ if and only if $f(w)$ is a root of $T'$, and
- $f(\alpha_T(t)) = \alpha_{T'}(t)$ for every $L$-term $t \in X$.

The number of non-isomorphic $X$-templates of a given depth is finite, as stated in the next proposition. The proof is straightforward and is left to the reader.

**Proposition 9.** For every finite set of terms $X$ and every integer $d$, there exist only finitely many non-isomorphic $X$-templates of depth at most $d$.

Let $L$ be a language and let $X$ be a set of $L$-terms with variables $\{x_1, \ldots, x_n\}$. An embedding of an $X$-template $T$ in a rooted forest $F$ is a mapping $\nu : V(T) \to V(F)$ such that $\nu(r)$ is a root of $F$ for every root $r$ of a tree of $T$ and $\nu$ is an isomorphism of $T$ and the subforest of $F$ induced by $\nu(V(T))$. Let $S$ be an $L$-structure guarded by the closure of $F$, and $v_1, \ldots, v_n \in V(S)$. We say that the embedding $\nu$ is $(v_1, \ldots, v_n)$-admissible for $S$ if for every term $t \in X$, we have $\nu(\alpha_T(t)) = t(v_1, \ldots, v_n)$, where $t(v_1, \ldots, v_n)$ denotes the element of $V(S)$ obtained by substituting $v_i$ for $x_i$ in the term $t$ and evaluating the functions forming the term $t$ (in particular, if $x_i \in X$, then $\nu(x_i) = v_i$). We will need the following lemma.

**Lemma 10.** Let $d \geq 1$ be an integer, let $F$ be a rooted forest of depth at most $d$, let $L$ be a language, let $\varphi(x_1, \ldots, x_n)$ be an $L$-formula, let $S$ be an $L$-structure guarded by the closure of $F$, and let $v_1, \ldots, v_n \in V(S)$. Then there exists a $\varphi$-template $T$ and an embedding of $T$ into $F$ that is $(v_1, \ldots, v_n)$-admissible for $S$.

**Proof.** Let $X$ be the set of all $L$-terms that appear in $\varphi$, and let $Y$ be the set of all evaluations $t(v_1, \ldots, v_n)$ of all terms $t(x_1, \ldots, x_n)$ from $X$. Let $T$ be the smallest subforest of $F$ that includes all vertices from $Y$ and the root of every component of $F$ that includes an element of $Y$. For a term $t(x_1, \ldots, x_n)$ in $X$ let $\alpha_T(t) := t(v_1, \ldots, v_n)$, and let $\nu$ be the identity mapping. Then $T$ and $\nu$ are as desired. \qed
If $F$ is a rooted forest, then the function $p : V(F) \rightarrow V(F)$ is the $F$-predecessor function if $p(v)$ is the parent of $v$ unless $v$ is a root of $F$; if $v$ is a root of $F$, $p(v)$ is set to be equal to $v$.

We now show that it can be tested by a quantifier-free formula whether an embedding is admissible.

**Lemma 11** (Testing admissibility). Let $d \geq 0$ be an integer, $L$ a language including a function symbol $p$ and $X$ a finite set of terms with variables $x_1, \ldots, x_n$. If $T$ is an $X$-template of depth at most $d$, then there exists a quantifier-free formula $\xi_T(x_1, \ldots, x_n)$ such that for every rooted forest $F$ and every $L$-structure $S$ guarded by the closure of $F$ such that $p^S$ is the $F$-predecessor function in $S$, and for every $n$-tuple $v_1, \ldots, v_n \in V(S)$, the $L$-structure $S$ satisfies $\xi_T(v_1, \ldots, v_n)$ if and only if there exists an embedding of $T$ in the forest $F$ that is $(v_1, \ldots, v_n)$-admissible for $S$.

**Proof.** Let $q : V(T) \rightarrow V(T)$ be the $T$-predecessor function. Set $\xi_T(x_1, \ldots, x_n)$ to be the conjunction of all formulas

- $p^k(t) = p^{k'}(t')$ if $q^k(\alpha_T(t)) = q^{k'}(\alpha_T(t'))$, and
- $p^k(t) \neq p^{k'}(t')$ if $q^k(\alpha_T(t)) \neq q^{k'}(\alpha_T(t'))$,

for all pairs of not necessarily distinct terms $t, t' \in X$ and all pairs of integers $k$ and $k'$, $0 \leq k, k' \leq d + 1$ (note that including the formulas with $t = t'$ allows for testing the depth of $t$ in $F$). Here $p^k$ denotes the function $p$ iterated $k$ times.

It is straightforward to show that for $v_1, \ldots, v_n \in V(S)$, an $(v_1, \ldots, v_n)$-admissible embedding for $S$ of $T$ in $F$ exists if and only if $S \models \xi_T(v_1, \ldots, v_n)$. \qed

The following lemma will be extremely useful in the proof of Lemma 13. Recall that an $L$-term is simple if it is a variable or a function image of a variable, and an $L$-formula is simple if all terms appearing in it are simple.

**Lemma 12.** Let $d \geq 0$ be an integer, $L$ a language, $\varphi(x_0, \ldots, x_n)$ a simple quantifier-free $L$-formula that is a conjunction of atomic formulas and their negations, and $T$ a $\varphi$-template. There exist a language $\overline{L}$ that extends $L$ and a quantifier-free $\overline{L}$-formula $\overline{\varphi}_T(x_1, \ldots, x_n)$ such that the following holds:

- $\overline{L}$ is obtained from $L$ by adding a function symbol $p$ and finitely many relation symbols $U_1, \ldots, U_k$ of arity at most one,
- $\overline{\varphi}_T$ is quantifier-free and the variables $x_1, \ldots, x_n$ are the only variables that appear in $\overline{\varphi}_T$, but $\overline{\varphi}_T$ need not be simple, and
- for every rooted forest $F$ of depth at most $d$ and every $L$-structure $S$ guarded by the closure of $F$, there exists an $\overline{L}$-structure $\overline{S}$ such that $\overline{S}$ is an expansion of $S$ and for every $v_1, \ldots, v_n \in V(S)$,

\[
S \models \varphi(v_0, v_1, \ldots, v_n) \text{ for some } v_0 \in V(S) \text{ such that there exists a } (v_0, \ldots, v_n)\text{-admissible embedding of } T \text{ in } F \text{ for } S \text{ if and only if } \overline{S} \models \overline{\varphi}_T(v_1, \ldots, v_n),
\]
where \( p^\boxdot \) is the \( F \)-predecessor function and the relations \( U_1^\boxdot, \ldots, U_k^\boxdot \) can be computed (by listing the singletons they contain) in linear time given \( F \) and \( S \).

**Proof.** Fix a \( \varphi \)-template \( T \) of depth at most \( d \) and let \( q \) be the \( T \)-predecessor function. Let \( X \) be the set of all terms appearing in \( \varphi \). Let \( \xi_T \) be the formula from Lemma 11 applied to the language obtained from \( L \) by adding the function symbol \( p \). Finally, let \( K = \max(\Delta, c) + 1 \) where \( \Delta \) is the maximum degree of \( T \) and \( c \) is the number of components of \( T \).

Let \( t = f(x_i) \) be a term appearing in \( \varphi \), for some function symbol \( f \in L^f \) and a variable \( x_i \) with \( 0 \leq i \leq n \). If \( \alpha_T(t) \) is neither an ancestor nor a descendant of \( \alpha_T(x_i) \), then for every rooted forest \( F \) of depth at most \( d \), every \( L \)-structure \( S \) guarded by the closure of \( F \) and every choice of \( v_1, \ldots, v_n \in V(S) \), there is no \( (v_0, \ldots, v_n) \)-admissible embedding for \( S \) of \( T \) into \( F \), because \( v_i \) and \( f^S(v_i) \) are adjacent in the Gaifmann graph of \( S \); in particular, one is a descendant of the other in \( F \). Hence, we can set \( \overline{\varphi}_T \) to \( \bot \). So, we can assume the following:

The images under \( \alpha_T \) of all function images of each variable \( x_i \) are ancestors or descendants of \( \alpha_T(x_i) \).

If \( \alpha_T(x_0) \) is an ancestor of a vertex \( \alpha_T(t) \), say \( q^k(\alpha_T(t)) = \alpha_T(x_0) \) for \( k \geq 0 \), where \( t \) is a term such that \( x_0 \) does not appear in \( t \), then \( \overline{\varphi}_T \) will be the formula obtained from \( \varphi \land \xi_T \) by replacing each \( x_0 \) with the term \( p^k(t) \). Clearly, \( \overline{S} \models \overline{\varphi}_T(v_1, \ldots, v_n) \) if and only if there is a choice of \( v_0 \in V(F) \) such that \( S \models \varphi(v_0, \ldots, v_n) \) and there is a \( (v_0, \ldots, v_n) \)-admissible embedding of \( T \) in \( F \) for \( S \). So, we can assume the following:

Every term \( t \) such that \( \alpha_T(t) \) is contained in the subtree of \( T \) rooted at \( \alpha_T(x_0) \) is \( x_0 \) or a function image of \( x_0 \).

We now define an auxiliary formula \( \varphi' \) to be the formula obtained from \( \varphi \) by replacing all subformulas of the form:

- \( t = t' \), where \( t \) and \( t' \) are terms such that \( \alpha_T(t) \neq \alpha_T(t') \),
- \( t \neq t' \), where \( t \) and \( t' \) are terms such that \( \alpha_T(t) = \alpha_T(t') \), and
- \( R(t_1, \ldots, t_m) \) such that \( \alpha_T(t_1), \ldots, \alpha_T(t_m) \) are not vertices of a clique in the closure of \( T \),

by \( \bot \). Such a subformula is not satisfied for any choice of \( v_0 \) for which there exists a \( (v_0, \ldots, v_n) \)-admissible embedding of \( T \) in \( F \) for \( S \). It follows that for every \( v_0 \) such that there is a \( (v_0, \ldots, v_n) \)-admissible embedding of \( T \) in \( F \) for \( S \), \( S \models \varphi'(v_0, \ldots, v_n) \) if and only if \( S \models \varphi(v_0, \ldots, v_n) \).

Suppose first that the tree of \( T \) containing the vertex \( \alpha_T(x_0) \) also contains an \( \alpha_T \)-image of another variable. Let \( v \) be the nearest ancestor of \( \alpha_T(x_0) \) in \( T \) such that there exists a term \( t_v \in X \) such that \( x_0 \) does not appear in \( t_v \) and \( v \) is an ancestor of \( \alpha_T(t_v) \). Note that \( v \neq \alpha_T(x_0) \) by (2). Let \( d_v \) be the depth of
\(v\) in \(T\), \(d_{x_0}\) the depth of \(\alpha_T(x_0)\) and \(m\) the number of children of \(v\) in \(T\). Let \(t_1, \ldots, t_{m-1}\) be terms such that \(\alpha_T(t_i), 1 \leq i \leq m - 1\), are vertices of different subtrees rooted at a child of \(v\) and not containing \(\alpha_T(x_0)\). Observe that the variable \(x_0\) does not appear in \(t_1, \ldots, t_{m-1}\) by \([1]\).

Let \(X_0\) be the subset of \(X\) consisting of terms in which \(x_0\) appears, and let \(T_0\) be the \(X_0\)-template obtained from \(T\) by taking the minimal rooted subtree containing \(\alpha_T(X_0)\) and the root of the tree containing \(\alpha_T(x_0)\), and restricting the function \(\alpha_T\) to \(X_0\). Further, let \(X'_0\) be the subset of \(X\) consisting of \(X_0\) and the terms \(t\) such that \(\alpha_T(t)\) lies on the path between the root and \(\alpha_T(x_0)\), and let \(X''_0\) be the subset of \(X_0\) consisting of the terms mapped by \(\alpha_T\) to a descendant of \(v\).

We define a unary relation \(U_1(w)\) to be the set of elements \(w\) of \(F\) at depth \(d_v + 1\) such that the subtree of \(w\) in \(F\) contains an element \(v_0\) at depth \(d_{x_0}\) (in \(F\)) with the following properties:

- there is a \((v_0)\)-admissible embedding of the template \(T_0\) in \(F\) for \(S\), and
- all clauses appearing in the conjunction \(\varphi'\) with terms from \(X'_0\) only and with at least one term from \(X''_0\) are true with \(x_0 = v_0\) and the terms \(t \in X'_0 \setminus X_0\), say \(\alpha_T(t) = q^k(\alpha_T(x_0)))\), replaced with \((\overline{p^S})^k(v_0)\).

The relation \(U_1(w)\) can be computed as follows: for every element \(v_0 \in V(S)\) at depth \(d_{x_0}\) of \(F\), evaluate all terms in \(X_0\) and test whether the tree \(T_0\) and the rooted subtree of \(F\) containing the values of the terms are isomorphic as rooted trees (this can be done in time linear in the size of \(T_0\) which is constant). If they are isomorphic, evaluate the clauses in the conjunction \(\varphi'\) with terms from \(X'_0\) only and with at least one term from \(X''_0\) with the terms in \(X'_0 \setminus X_0\) replaced with \((\overline{p^S})^k(v_0)\). If all of them are true, add the predecessor \(w\) of \(v_0\) at depth \(d_v + 1\) in \(F\) to \(U_1\). Since the time spent by checking every vertex \(v_0\) at depth \(d_{x_0}\) of \(F\) is constant, the time needed to compute \(U_1\) is linear.

We further define a unary relation \(U_i(w), 2 \leq i \leq m + 1\), to be the set of elements \(w\) of \(F\) at depth \(d_v\) such that \(U_i(w')\) is true for at least \(i - 1\) children \(w'\) of \(w\). Clearly, the relations \(U_i(w), 2 \leq i \leq m + 1\), can be computed in linear time when the relation \(U_1\) has been determined.

Let \(\varphi'\) be the formula obtained from \(\varphi'\) by removing all clauses with terms from \(X'_0\) only that contain at least one term from \(X''_0\). Observe that if \(t\) is a term in \(\varphi'\) such that \(x_0\) appears in \(t\), i.e., \(t \in X'_0 \setminus X''_0\), then \(\alpha_T(t)\) lies on the path between \(v\) and the root. Let \(\varphi''\) be the formula obtained from \(\varphi''\) by replacing every term \(t\), in which \(x_0\) appears, with \(p^k(t_v)\), where \(k\) is the integer such that \(\alpha_T(t) = q^k(\alpha_T(t_v))\). Let \(T'\) be the \((X \setminus X_0)\)-template obtained from \(T\) by taking the minimal rooted subtree containing \(\alpha_T(X \setminus X_0)\) and restricting the function \(\alpha_T\) to \(X \setminus X_0\). The formula \(\overline{\varphi_T}\) will then be the conjunction of the following formulas:

- the formula \(\varphi''(x_1, \ldots, x_n)\),
- the formula \(\xi_{T'}\) from Lemma \([1]\) for the \((X \setminus X_0)\)-template \(T'\), and
• the formulas
  \[
  \neg \left( \bigwedge_{i \in Y} U_1(p^{k_i-1}(t_i)) \bigvee U_{|Y|+2}(p^k(t_v)) \right)
  \]
  for all subsets \( Y \) of the set \{1, \ldots, m - 1\} where \( k \) is the integer such that \( q^k(\alpha_T(t_v)) = v \) and \( k_i, i = 1, \ldots, m - 1 \), are the integers such that \( q^{k_i}(\alpha_T(t_i)) = v \).

  If \( \mathfrak{S} \models \varphi_T(v_1, \ldots, v_n) \), then there is a \((v_1, \ldots, v_n)\)-admissible embedding of \( T' \) in \( F \) and the vertex \( v = p^k(t_v) \) has a son \( w \) such that \( U_1(w) \) is true and the subtree of \( F \) rooted in \( w \) does not contain the value of any term appearing in \( X \setminus X_0 \) (this is guaranteed by the last type of formulas in the definition of \( \varphi_T \)). In particular, the subtree rooted in \( w \) contains a vertex \( v_0 \) such that the \((v_1, \ldots, v_n)\)-admissible embedding of \( T' \) can be extended to a \((v_0, \ldots, v_n)\)-admissible embedding of \( T \) in \( F \) for \( S \) and all clauses in the conjunction \( \varphi' \) containing terms from \( X_0' \) are satisfied with \( x_0 = v_0 \). Since \( \mathfrak{S} \models \varphi'(v_1, \ldots, v_n) \), it follows that \( S \models \varphi(v_0, \ldots, v_n) \). The argument that the existence of \( v_0 \) such that \( S \models \varphi(v_0, \ldots, v_n) \) and the existence of a \((v_0, \ldots, v_n)\)-admissible embedding of \( T \) in \( F \) for \( S \) implies that \( \mathfrak{S} \models \varphi_T(v_1, \ldots, v_n) \) follows the same lines.

  The case that the tree of \( T \) that contains the vertex \( \alpha_T(x_0) \) does not contain \( \alpha_T\)-image of another variable is handled similarly. In this case, the predicate \( U_1 \) is defined for the roots of the trees of \( F \), and the predicates \( U_2, \ldots, U_{m+1} \) (where \( m \) is the number of components of \( T \)) are nullary predicates such that \( U_i \) is true if \( U_1 \) is satisfied for at least \( i - 1 \) roots of the trees in \( F \).

• We now prove the lemma that forms the core of our first algorithm.

  \textbf{Lemma 13 (Quantifier elimination lemma).} \( \text{Let } d \geq 0 \text{ be an integer, } L \text{ a language and } \varphi(x_1, \ldots, x_n) \text{ a simple } L\text{-formula of the form } \exists x_0 \varphi'(x_0, \ldots, x_n) \text{ such that } \varphi'(x_0, \ldots, x_n) \text{ is a quantifier-free } L\text{-formula with free variables } x_0, \ldots, x_n. \text{ There exist a language } \bar{L} \text{ and an } \bar{L}\text{-formula } \overline{\varphi} \text{ such that the following holds:} \)

  \begin{itemize}
  \item \( \bar{L} \) is obtained from \( L \) by adding a function symbol \( p \) and finitely many relation symbols \( U_1, \ldots, U_k \) of arity at most one,
  \item \( \overline{\varphi} \) is quantifier-free and the variables \( x_1, \ldots, x_n \) are the only variables that appear in \( \overline{\varphi} \), but \( \overline{\varphi} \) need not be simple, and
  \item for every rooted forest \( F \) of depth at most \( d \) and every \( L\text{-structure } S \) guarded by the closure of \( F \), there exists an \( \bar{L}\text{-structure } \mathfrak{S} \) with \( V(S) = V(\mathfrak{S}) \) such that \( \mathfrak{S} \) is an expansion of \( S \) and for every \( v_1, \ldots, v_n \in V(S) \),
  \[
  S \models \varphi(v_1, \ldots, v_n) \text{ if and only if } \mathfrak{S} \models \overline{\varphi}(v_1, \ldots, v_n)
  \]
  \end{itemize}

  where \( p^\mathfrak{S} \) is the \( F\)-predecessor function and the relations \( U_1^\mathfrak{S}, \ldots, U_k^\mathfrak{S} \) can be computed (by listing the singletons they contain) in linear time given \( F \) and \( S \).
Proof. Let $d$, $L$ and $\varphi'$ be fixed. We assume without loss of generality that the formula $\varphi'$ is in the disjunctive normal form and all the variables $x_0, \ldots, x_n$ appear in $\varphi'$. Let $F$ be a rooted forest of depth at most $d$, and let $S$ be an $L$-structure.

The proof proceeds by induction on the length of $\varphi'$. If $\varphi'$ is a disjunction of two or more conjunctions, i.e., $\varphi' = \varphi_1 \lor \varphi_2$, we apply induction to the formulas $\exists x_0 \varphi_1$ and $\exists x_0 \varphi_2$. We obtain languages $L_1$ and $L_2$, and for $i = 1, 2$ an $L_i$-formula $\mathcal{P}_i$ and an $L_i$-structure $\mathcal{S}_i$. We assume that the new unary relation symbols of $L_1$ and $L_2$ are distinct and set $\mathcal{T}_0 = L_1' \cup L_2'$, $\mathcal{T}_1 = L_1' \cup \{p\}$ and $\mathcal{P}_0 = \mathcal{P}_1 \lor \mathcal{P}_2$. We define the $L$-structure $\mathcal{S}$ by $V(\mathcal{S}) = V(S)$ and by taking the interpretations of symbols from $\mathcal{S}_1$ and $\mathcal{S}_2$.

Thus in the remainder of the proof we may assume that $\varphi'$ is a conjunction. Let $v_1, v_2, \ldots, v_n \in V(S)$. By Lemma 10 we have $S \models \varphi(v_1, \ldots, v_n)$ if and only if there exist $v_0 \in V(S)$ and a $\varphi'$-template $T$ of depth at most $d$ such that $S \models \varphi'(v_0, \ldots, v_n)$ and there exists an embedding of $T$ into $F$ that is $(v_0, \ldots, v_n)$-admissible for $S$. By Proposition 9 the number of $\varphi'$-templates of depth at most $d$ is bounded by a function of $\varphi$ and $d$. By Lemma 12 for every $\varphi'$-template $T$ of depth at most $d$, there exist a language $L_T$, a quantifier-free $L_T$-formula $\varphi_T$ and an $L_T$-structure $S_T$ that is an expansion of $S$ such that for every $v_1, v_2, \ldots, v_n \in V(S)$, $S_T \models \varphi_T(v_1, \ldots, v_n)$ if and only if there exists $v_0$ such that there is a $(v_0, v_1, \ldots, v_n)$-admissible embedding of $T$ in $F$ for $S$ and $S \models \varphi'(v_0, v_1, \ldots, v_n)$. We may assume that for distinct $\varphi'$-templates $T$ and $T'$, if a function or relation symbol belongs both $L_T$ and $L_T'$, then it belongs to $L_T$. Let $\mathcal{T}$ be the language consisting of all function and relation symbols of all $L_T$, let the formula $\mathcal{P}$ be obtained as the disjunction of the $L$-formulas $\mathcal{P}_T$, where the disjunction runs over all choices of $\varphi'$-templates $T$, and let the $L$-structure $\mathcal{S}$ be obtained by taking the union of the interpretations of all $S_T$. Then $\mathcal{T}$, $\mathcal{P}$ and $\mathcal{S}$ are as desired. \qed

In order to apply Lemma 13 the formula needs to be simple but the lemma produces a formula that need not be simple. The following lemma copes with this issue:

**Lemma 14.** Let $L$ be a language and $\varphi(x_1, \ldots, x_n)$ an $L$-formula with $q$ quantifiers. There exists a language $L'$ that extends $L$, a simple $L'$-formula $\varphi'(x_1, \ldots, x_n)$ with $q$ quantifiers and an integer $k$ with the following properties. For every $L$-structure $A$ guarded by a graph $G$, there exists an $L'$-structure $A'$ guarded by a $k$-th augmentation $G'$ of $G$ such that $A \models \varphi(v_1, \ldots, v_n)$ if and only if $A' \models \varphi'(v_1, \ldots, v_n)$ for any $v_1, \ldots, v_n \in V(A) = V(A')$. Moreover, an $L'$-structure $A'$ and a $k$-th augmentation $G'$ satisfying the above specifications can be computed in time $O(|V(G')| + |E(G')|)$.

**Proof.** We may assume that $\varphi$ is not simple, for otherwise there is nothing to prove. Let $f$ and $g$ be function symbols of $L$ such that the $L$-term $g(f(t))$ appears in $\varphi$ for some $L$-term $t$. Let $L_1$ be the extension of $L$ obtained by adding a new function symbol $h$, and for an $L$-structure $A$ we define an $L_1$-structure $A_1$ as the expansion of $A$, where the interpretation of $h$ is defined by
$h^{A_1}(v) = g^A(f^A(v))$ for all $v \in V(A)$. Let $\varphi_1$ be obtained from $\varphi$ by replacing the appearance of $g(f(t))$ by $h(t)$. Then clearly $A \models \varphi(v_1, \ldots, v_n)$ if and only if $A_1 \models \varphi'(v_1, \ldots, v_n)$ for all $v_1, \ldots, v_n \in V(A) = V(A_1)$. Let $D'$ be an orientation of $G$ of maximum in-degree $2\Delta_0(G)$, and let $D$ be obtained from $D'$ by adding all directed edges with head $v$ and tail $f(v)$ and all directed edges with head $v$ and tail $g(v)$. Finally, let $G_1$ be the augmentation of $D$. Then $G_1$ is a first augmentation of $G$ and $A_1$ is guarded by $G_1$. By repeating this construction at most $k$ times, where $k$ depends only on $\varphi$, we arrive at a desired formula $\varphi'$ and $k$-th augmentation $G'$ of $G$. It is clear that both can be computed in time $O(|V(G')| + |E(G')|)$. 

We are now ready to prove Theorem 15. Let $\mathcal{G}$ be a class of graphs with bounded expansion, $L$ a language and $\varphi(x_1, \ldots, x_n)$ an $L$-formula. There exists a language $L'$, an integer $m$ and a linear-time algorithm that given an $L$-structure $A$ guarded by $G \in \mathcal{G}$ computes an $m$-th augmentation $G'$ of $G$, an $L'$-structure $A'$ guarded by $G'$, and a quantifier-free $L'$-formula $\varphi'(x_1, \ldots, x_n)$ such that for all $v_1, \ldots, v_n \in V(A) = V(A')$

$$A \models \varphi(v_1, \ldots, v_n) \text{ if and only if } A' \models \varphi'(v_1, \ldots, v_n).$$

In particular, if $n = 0$, the algorithm decides whether $A \models \varphi$.

**Proof of Theorems 3 and 7.** It suffices to show how to compute $G', A', \varphi'$ satisfying the specifications of the theorem, except that $\varphi'$ has one fewer quantifier than $\varphi$. A proof of the theorem is then obtained by iterating this algorithm.

If $\varphi$ is quantifier-free, then there is nothing to prove. Hence, assume that $\varphi$ contains at least one quantifier. By Lemma 4 we may assume that $\varphi$ is simple.

Since $\forall x \psi$ is equivalent to $\neg \exists x \neg \psi$, we can assume that $\varphi$ contains a subformula of the form $\exists x_0 \psi$, where $\psi$ is a formula with variables $x_0, x_1, \ldots, x_N$ and with no quantifiers. Let $N_0$ be the number $N + 1$ increased by the number of distinct function images of $x_0, x_1, \ldots, x_N$ appearing in $\psi$. Let $K = 3N_0^2 + 1$ and consider a coloring of a $K$-th augmentation $G'$ of $G$: this coloring is a depth-certifying low-tree-depth coloring of $G$ of order $N_0$ (see the discussion before Theorem 14). Let $K_0$ be the number of colors used by this coloring and $C_i$, $i = 1, \ldots, K_0$, a unary relation containing vertices with the $i$-th color. Note that $K_0$ is bounded by a constant depending on $\mathcal{G}$ and $K$ only. Clearly, computing the coloring and determining the coloring can be performed in linear time.

For each function symbol $f \in L$ and $i = 1, \ldots, K_0$, let $C_i^f$ be the predicate defined so that $C_i^f(v)$ is true for $v \in V(A)$ if and only if the color of $f(v)$ is $i$. The colors of the variables and their function images appearing in $\psi$ can be described by an $N_0$-tuple $\alpha$ of numbers between 1 and $K_0$. Let $\Lambda$ be the set of all such $N_0$-tuples. For $\alpha \in \Lambda$, let $\varphi_\alpha$ be the conjunction of the terms of the form $C_\alpha(x_i)$ and $C_i^f(x_i)$ that verifies that the assignment of the colors of the values of $x_0, \ldots, x_N$ and their function images are consistent with $\alpha$. Clearly, $\exists x_0 \psi$ is equivalent to the disjunction of the $L$-formulas $\exists x_0(\psi \land \varphi_\alpha)$, where the disjunction ranges through all $\alpha \in \Lambda$. 


For each $\alpha \in \Lambda$ let $L_\alpha$ be the language with function symbols $f_\alpha$ for all $f \in L'$ and relation symbols $R_\alpha$ for all $R \in L'$. For an $L$-structure $A$ and $\alpha \in \Lambda$ we define an $L_\alpha$-structure $A_\alpha$ by $f_\alpha^A(v) = f^A(v)$ if both $v$ and $f^A(v)$ have a color in $\alpha$ and by $f_\alpha^A(v) = v$ otherwise; and for each relation symbol $R \in L'$ we let $R_\alpha^A$ be defined by restricting $R^A$ to the vertices with color $\alpha$. Let $\varphi'_\alpha$ be the $L_\alpha$-formula obtained from $\exists x_0 (\psi \land \varphi_\alpha)$ by replacing each function symbol $f$ by $f_\alpha$ and each relation symbol $R$ by $R_\alpha$. For $\alpha \in \Lambda$, the formula $\exists x_0 (\psi \land \varphi_\alpha)$ is true for $A$ if and only if $\varphi'_\alpha$ is true for $A_\alpha$. Note that $A_\alpha$ is guarded by the graph $H_\alpha$ obtained from $G$ by removing the edges incident with the colors not in $\alpha$. Since $3N_0^2 + 1 \leq K$, $G'$ contains an out-branching $F_\alpha$ of depth at most $N_0$ whose closure contains $H_\alpha$.

For each $\alpha \in \Lambda$ we apply Lemma [13] to $d$ replaced by $N_0$, the language $L_\alpha$, formula $\varphi''_\alpha$, $L_\alpha$-structure $A_\alpha$ and rooted forest $F_\alpha$, obtaining a language $\overline{L}_\alpha$, a formula $\psi_\alpha$ and a structure $A'_\alpha$ such that

$$A_\alpha \models \varphi''_\alpha(v_1, \ldots, v_N) \text{ if and only if } A'_\alpha \models \psi_\alpha(v_1, \ldots, v_N)$$

for all $v_1, \ldots, v_N \in V(A_\alpha) = V(A)$.

Let $L'$ be the language obtained by taking the union of $L$ and all $L_\alpha$ over $\alpha \in \Lambda$, and let $\varphi'$ be the $L'$-formula obtained from $\varphi$ by replacing the subformula $\exists x_0 \psi$ by the disjunction of the $L_\alpha$-formulas $\psi_\alpha$ for all choices of $\alpha \in \Lambda$. Let $A'$ be the $L'$-structure obtained by taking the interpretations of symbols from $A$ and $A_\alpha$ for all $\alpha \in \Lambda$. Then $A'$ is guarded by the graph $G'$, and contains one less quantifier than $\varphi$, as required. Since each application of Lemmas [13] and [14] can be performed in linear time and the number of their applications is bounded by a function depending on $\varphi$ and $G$ only, the reduction of $\varphi$ to $\varphi'$ and computing $G'$ and $A'$ can be done in linear time. $\square$

4 Deciding FOL properties in graphs with locally bounded expansion

The following theorem is implicit in [15] (also see [18]).

**Theorem 16.** Let $G$ be a class of graphs and for an integer $d \geq 0$ let $G_d$ be the class of graphs consisting of all induced subgraphs of $d$-neighborhoods of graphs in $G$. Let $G_d$ be nowhere dense for all integers $d \geq 0$. Furthermore, let $L$ be a language and $L'$ the language obtained from $L$ by adding a new binary relation symbol. Suppose that for every $d$ and every $L'$-formula $\varphi'(x)$, there exists a linear-time algorithm that lists all elements $v$ of an input $L'$-structure guarded by a graph from $G_d$ that satisfy $\varphi'(v)$. Then, for every $L$-sentence $\varphi$ there exists an almost linear-time algorithm that decides whether an input $L$-structure guarded by a graph from $G$ satisfies $\varphi$.

**Proof.** The proof of [15] Theorem 1.2] can serve as a proof of this theorem with the following modifications. The proof in [15] relies on Lemma 4.4, Corollary 6.3, Corollary 8.2 and Lemma 8.3 from the same paper, and those assume that $G$ has
“bounded local tree-width”. However, the use of Lemma 4.4 can be replaced by the hypothesis of our theorem, the conclusion of Lemma 8.3 can be deduced from the hypothesis of our theorem, and the conclusions of the remaining statements follow from the fact that every \( L \)-structure \( A \) guarded by a member of \( \mathcal{G}_d \) satisfies \( |A| = O(|V(A)|^{1+\epsilon}) \) for every \( \epsilon > 0 \). This, in turn, follows from the fact that for every \( \epsilon > 0 \), every subgraph of every member of \( \mathcal{G}_d \) has a vertex of degree at most \( O(|V(H)|^{\epsilon}) \).

**Proof of Corollary 4.** Let \( \mathcal{G}, L \) and \( \varphi \) be as in Corollary 4. In particular, the class \( \mathcal{G}_d \) has bounded expansion for every \( d \). By Theorem 15, for every integer \( d \) and every \( L' \)-formula \( \varphi'(x) \), there exists a language \( L'' \) and a quantifier-free \( L'' \)-formula \( \varphi''(x) \) such that every \( L' \)-structure \( A \) guarded by a graph from \( \mathcal{G}_d \) can be transformed in linear time to an \( L'' \)-structure \( A' \) with \( V(A) = V(A') \) such that \( A \models \varphi'(v) \) if and only if \( A' \models \varphi''(v) \) for every \( v \in V(A) \). In particular, it is possible to list in linear time all \( v \in V(A) \) such that \( A \models \varphi''(v) \) since evaluating the latter formula requires constant time. So, the assumptions of Theorem 16 are satisfied.

**5 Dynamic data structures for \( \Sigma_1 \)-queries**

In this section, we provide a data structure for answering \( \Sigma_1 \)-queries. The update time is constant but the price we have to pay is that the graph that guards the relational structure must be fixed before the computation starts. Before we start our exposition, we need to introduce more definitions.

Let \( L \) be a language with no function symbols. For an integer \( k \), a \( k \)-labelled \( L \)-structure is an \( L \)-structure \( S \) with a partial injective mapping \( \alpha : [1,k] \rightarrow V(S) \), i.e., \( \alpha \) need not be defined for all integers between 1 and \( k \). In our further consideration, we will also allow \( k \) to be equal to zero.

The *trunk* of a \( k \)-labelled \( L \)-structure \( S \) is the \( L \)-structure obtained from \( S \) by removing all relations with elements only from \( \alpha([1,k]) \). A \( k \)-labelled \( L \)-structure \( S \) is *hollow* if \( S \) is equal to its trunk. Two \( k \)-labelled \( L \)-structures \( S_1 \) and \( S_2 \) are \( k \)-isomorphic if their trunks are isomorphic through an isomorphism commuting with mappings \( \alpha_1 \) and \( \alpha_2 \). In particular, every \( k \)-labelled \( L \)-structure is \( k \)-isomorphic to its trunk.

Suppose now that an \( L \)-structure \( S \) is guarded by the closure of a rooted tree \( T \). The *depth* of a root of \( T \) is zero, and the depth of every other vertex of \( T \) is the depth of its immediate predecessor plus one. For a vertex \( v \) of \( T \) at depth \( d \), let \( P_T(v) \) denote the vertex-set of the path from the root of \( T \) to \( v \) and \( T'(v) \) the elements the subtree of \( v \) (including \( v \) itself). Then, \( S(v) \) denotes the set of all \( d \)-labelled \( L \)-structures \( S' \) such that \( S' \) is an induced substructure of \( S \) with elements only in \( P_T(v) \cup T(v) \) and \( \alpha(i) = w \) for every vertex \( w \in P_T(v) \cap V(S') \) at depth \( i \). If a vertex of \( P_T(v) \) at depth \( i - 1 \) is not contained in \( S' \), then \( \alpha(i) \) is not defined.

We are now ready to prove a lemma that contains the core of our data structure.
Lemma 17. Let $L$ be a language with no function symbols, $d_0$ a fixed integer and $F$ a rooted forest of depth at most $d_0$. There exists a data structure representing an $L$-structure $S$ guarded by the closure of $F$ such that

- the data structure is initialized in linear time,
- the data structure representing an $L$-structure $S$ can be changed to the one representing an $L$-structure $S'$ by adding or removing a tuple from one of the relations in constant time provided that both $S$ and $S'$ are guarded by the closure of $F$, and
- the data structure decides in time bounded by $O(|\varphi|)$ whether a given $\Sigma_1$-$L$-sentence $\varphi$ with at most $d_0$ variables is satisfied by $S$, and if so, it outputs one of the satisfying assignments.

Proof. For every vertex $v$ of $F$ at depth $d$, we will store the following two lists:

- the list of all relations from $S$ that contain $v$ and all their elements are in $P_F(v)$, and
- the list of all (non-$d$-isomorphic) $d$-labelled hollow $L$-structures with at most $d_0$ elements that are $d$-isomorphic to a $d$-labelled $L$-structure contained in $S(v)$.

Since there are only finitely many non-$d$-isomorphic $d$-labelled $L$-structures with at most $d_0$ elements for every $d \leq d_0$, the length of each list of the second type is bounded by a constant depending only on $d_0$ and $L$. If $v$ is a non-leaf vertex of $F$, there will be a third list associated with $v$:

- the list of all (non-isomorphic) $(d+1)$-labelled hollow $L$-structures $S'$ with at most $d_0$ elements that appear in the second list of at least one child of $v$; for each such $S'$, there will be stored the list of all children of $v$ whose second list contains $S'$.

In addition, there will be a global list of all (non-isomorphic) $L$-structures with at most $d_0$ elements that appear as induced $L$-substructures in $S$.

Let us describe how all these lists are initialized. The initialization of the first type of lists is trivial: just put each relation to the list of its element that is farthest from the root. This can clearly be done in constant time per relation.

Initialization of other types of lists is more difficult. Fix a tree $T$ of $F$. We proceed from the leaves towards the root of $T$. Let $v$ be a vertex of $T$ at depth $d$. If $v$ is a leaf of $T$ at depth $d$, then the second list of $v$ contains only those hollow $d$-labelled $L$-structures $S'$ with $V(S') \subseteq P_F(v)$ such that if $v \in V(S')$, then $S'$ contains precisely all relations of $S$ induced by $V(S')$ and containing $v$, and if $v \notin V(S')$, then $S'$ contains no relations at all.

Suppose now that $v$ is not a leaf of $T$. The third list associated with $v$ can be initialized by merging the second type of lists of children of $v$. We describe how it can be decided whether a $d$-labelled hollow $L$-structure $S'$ should be contained
in the list of \(v\) of the second type. Assume that \(S(v)\) contains a \(d\)-labelled hollow \(L\)-structure \(S''\) that is \(d\)-isomorphic to \(S'\).

Then, \(V(S'')\) can be decomposed into disjoint subsets \(V_0, V_1, \ldots, V_m\) such that \(V_0 = V(S'') \cap P_T(v)\), each of the sets \(V_i\), \(i = 1, \ldots, m\), is fully contained in a subtree of a child \(v_i\) of \(v\), and different subsets \(V_1, \ldots, V_m\) are contained in different subtrees. Observe that every relation of \(S''\) must be contained in \(V_0 \cup V_i\) for some \(i = 1, \ldots, m\). Moreover, the only relations of \(S''\) contained in \(V_0\) are those that contain \(v\).

Hence, the existence of \(S''\) can be tested by considering all partitions of \(V(S')\) into disjoint subsets \(V_0, V_1, \ldots, V_m\) such that \(\alpha([1, d]) \subseteq V_0\), \(|V_0 \setminus \alpha([1, d])| \leq 1\) and every relation of \(S''\) is contained in \(V_0 \cup V_i\) for some \(i = 1, \ldots, m\), and then testing the existence of children \(v_1, \ldots, v_m\) such that the second list of \(v_i\) contains a \((d+1)\)-labelled hollow \(L\)-structure \((d+1)\)-isomorphic to the \((d+1)\)-labelled hollow \(L\)-structure of \(S'\) induced by \(V_0 \cup V_i\); if \(|V_0 \setminus \alpha([1, d])| = 1\), then \(\alpha(d+1)\) is defined to be equal to the unique element of \(V_0 \setminus \alpha([1, d])\) and we also test whether the relations of \(S'\) containing \(\alpha(d+1)\) are precisely those relations of \(S\) restricted to \(P_T(v)\) that contain \(v\), i.e., those in the first list of \(v\).

We now describe how to test the existence of children \(v_1, \ldots, v_m\). Let \(W\) be the set of children of \(v\) such that: if \(v\) has at most \(m\) children with their second list containing a \((d+1)\)-labelled hollow \(L\)-structure \((d+1)\)-isomorphic to the substructure of \(S'\) induced by \(V_0 \cup V_i\), then \(W\) contains all such children of \(v\). If \(v\) has more than \(m\) such children, then \(W\) contains arbitrary \(m\) of these children. Clearly, \(|W| \leq m^2 \leq d^2\). In order to test the existence of such children \(v_1, \ldots, v_m\) of \(v\), we form an auxiliary bipartite subgraph \(B\): one part of \(B\) is formed by numbers \(1, \ldots, m\) and the other part by children of \(v\) contained in \(W\). A child \(w \in W\) is joined to a number \(i\) if the second list of \(w\) contains a \((d+1)\)-labelled hollow \(L\)-structure \((d+1)\)-isomorphic to the substructure of \(S'\) induced by \(V_0 \cup V_i\).

If \(B\) has a matching of size \(m\), then this matching determines the choice of children \(v_1, \ldots, v_m\). On the other hand, if such children exist, \(B\) contains a matching of size \(m\): indeed, if \(v_i \notin W\), then \(v\) has at least \(m\) children whose second lists contain \((d+1)\)-labelled hollow \(L\)-structure \((d+1)\)-isomorphic to the substructure of \(S'\) induced by \(V_0 \cup V_i\), and we can change \(v_i\) to one of these \(m\) children that is different from \(v_i\) for \(i' \neq i\). Hence, we can assume that \(v_i \in W\) for every \(i = 1, \ldots, m\) which implies that \(B\) has a matching of size \(m\).

Since the order of \(B\) is at most \(m^2 + m\) and the number of disjoint non-empty partitions of \(V(S')\) to \(V_0, \ldots, V_m\) is bounded, testing the existence of a \(d\)-labelled hollow \(L\)-structure \(S''\) can be performed in constant time for \(v\).

It remains to construct the global list containing \(L\)-structures \(S_0\) with at most \(d_0\) elements that appear in \(S\) as induced substructures. We proceed similarly as when determining the lists of inner elements of the forest \(F\). For every \(L\)-structure \(S'\) with at most \(d_0\) elements, we compute the list of trees of \(F\) that contain \(S'\), i.e., \(S'\) is contained in the second list of the root of \(F\). Now, \(S_0\) is an induced substructure of \(S'\) if and only if there exist element-disjoint \(L\)-structures \(S_1, \ldots, S_m\) such that \(S_0 = S_1 \cup \cdots \cup S_m\) and \(S_1, \ldots, S_m\) appear in \(m\) mutually distinct trees of \(F\). For each such partition of \(S_0\) into \(S_1, \ldots, S_m\), we can test
whether \( S_1, \ldots, S_m \) appear in the list of roots of \( m \) distinct trees of \( F \) using the auxiliary bipartite graph described earlier. Since all structures involved contain at most \( d_0 \) elements, this phase requires time linear in the number of trees of \( F \).

We have shown that the data structure can be initialized in linear time. Let us now focus on updating the structure and answering queries. Consider a tuple \((v_1, \ldots, v_k)\) that is added to a relation \( R \) or removed from a relation \( R \). Let \( r \) be the root of a tree in \( F \) that contains all the elements \( v_1, \ldots, v_k \) and assume that \( v_1, \ldots, v_k \) appear in this order on a path from \( r \). By the definition, the only lists affected by the change are those associated with vertices on the path \( P(v_k) \). Recomputing each of these lists requires constant time (we proceed in the same way as in the initialization phase except we do not have to run through the children of the vertices on the path to determine which of them contain particular \( k \)-labelled hollow \( L \)-substructure \( S' \) in their lists). Since the number of vertices on the path \( P(v_k) \) is at most \( d_0 \), updating the data structure requires constant time only.

It remains to describe how queries are answered. Let \( \varphi \) be a \( \Sigma_1 \)-sentence with \( d \leq d_0 \) variables. We generate all possible \( L \)-structures \( S_0 \) with \( |V(S_0)| = d \) and check whether they satisfy the formula \( \varphi \). Let \( SS_0 \) be the set of those satisfying \( \varphi \). The set \( SS_0 \) can be generated in time \( O(|\varphi|) \) since \( L \) and \( d_0 \) are fixed.

Observe that \( S \) satisfies \( \varphi \) if and only if it has an induced substructure isomorphic to a structure in \( SS_0 \). This can be tested in constant time by inspecting the global list. Providing the satisfying assignment can be done in constant time if during the computation for each substructure a certificate why it was included in the list is stored (which requires constant time overhead only).

We are now ready to describe the data structures. We start with the one for graphs with bounded expansion.

**Theorem 18.** Let \( L \) be a language with no function symbols, \( d_0 \) a fixed integer and \( \mathcal{G} \) a class of graphs with bounded expansion. There exists a data structure representing a \( L \)-structure \( S \) such that

- given a graph \( G \in \mathcal{G} \), the data structure is initialized in linear time with \( S \) being initially empty,
- the data structure representing an \( L \)-structure \( S \) can be changed to the one representing an \( L \)-structure \( S' \) by adding or removing a tuple from one of the relations in constant time provided that both \( S \) and \( S' \) are guarded by \( G \), and
- the data structure decides in time bounded by \( O(|\varphi|) \) whether a given \( \Sigma_1 \)-\( L \)-sentence \( \varphi \) with at most \( d_0 \) variables is satisfied by \( S \), and if so, it outputs one of the satisfying assignments.

**Proof.** By Theorem 7 there exists a constant \( K \) (depending on \( \mathcal{G} \) only) such that for every \( G \in \mathcal{G} \) we can find in linear time a low-tree-depth coloring of \( G \) of order \( d_0 \) using at most \( K \) colors together with the depth-certifying forests. For every \( d_0 \) color classes of \( G \), we apply Lemma 17. Since \( K \) is a constant, the
number of data structures we maintain is bounded by a constant which depends on $\mathcal{G}$ only. If the given sentence $\varphi$ is satisfied, it is also satisfied in one of the unions of color classes. This is tested using the data structure from Lemma 17. If $\varphi$ is not satisfied in any of the auxiliary data structures, it is not satisfied in $S$ either and we report that.

The following is a variation of the above theorem for nowhere dense graphs.

**Theorem 19.** Let $L$ be a language with no function symbols, $k_0$ a fixed integer, $\varepsilon$ a positive real number and $\mathcal{G}$ a class of nowhere-dense graphs. There exists a data structure representing an $L$-structure $S$ such that

- given a graph $G \in \mathcal{G}$, the data structure is initialized in time $O(n^{1+\varepsilon})$ with $S$ being initially empty,
- the data structure representing an $L$-structure $S$ can be changed to the one representing an $L$-structure $S'$ by adding or removing a tuple from one of the relations in time $O(n^{\varepsilon})$ provided that both $S$ and $S'$ are guarded by $G$, and
- the data structure decides in time bounded by $O(|\varphi|)$ whether a given $\Sigma_1$-$L$-sentence $\varphi$ with at most $k_0$ variables is satisfied by $S$, and if so, it outputs one of the satisfying assignments.

**Proof.** Let $\varepsilon > 0$. The proof is based on the same data structure as Theorem 18 but uses an extra ingredient. Since $\mathcal{G}$ is nowhere dense, $K$ is no longer a constant; instead, we may select $K = O(n^{\varepsilon/k_0})$. In order to be able to answer queries in time $O(n^{\varepsilon})$ we compute all induced substructures $S'$ of $S$ satisfying $|V(S')| \leq k_0$. This can be done in time $O(n^{\varepsilon})$. To answer a query whether $S \models \varphi$ we test whether $S' \models \varphi$ for some induced substructure $S'$ of $S$ with $|V(S')| \leq k_0$. Since $\varphi$ is a $\Sigma_1$-sentence, this is a correct test for $S \models \varphi$. If a change to $S$ is desired, then we make the corresponding change to the data structure from Theorem 18 and recompute the set of induced substructures in time $O(n^{\varepsilon})$. 

6 Dynamic data structure for FOL-properties

In this section, we present our dynamic data structure for testing FOL properties. The main result of this section reads as follows:

**Theorem 20.** Let $\mathcal{G}$ be a class of graphs with bounded expansion, $L$ a language and $\varphi$ an $L$-sentence. There exists a data structure that is initialized with an $n$-vertex graph $G \in \mathcal{G}$ and an $L$-structure $A$ guarded by $G$ in time $O(n)$ and supports the following operations:

- adding a tuple to a relation of $A$ in constant time provided $A$ stays guarded by $G$,
- removing a tuple from a relation of $A$ in constant time, and
• it answers in constant time whether $A \models \varphi$.

Note that in Theorem 20 we do not allow to change function values of functions from $L$ to simplify our exposition; this does not present a loss of generality as one can model functions as binary relations.

Theorem 20 follows from a dynamized version of Theorem 15 (we state the theorem in the variant with no free variables for simplicity).

**Theorem 21.** Let $G$ be a class of graphs with bounded expansion, $L$ a language and $\varphi$ an $L$-sentence. There exists a language $L'$ and a quantifier-free $L'$-sentence $\varphi'$ and a data structure representing an $L'$-structure $A'$ that can be initialized with an $n$-vertex graph $G \in G$ and an $L$-structure $A$ guarded by $G$ in time $O(n)$, $V(A) = V(A')$, and that satisfies:

- $A \models \varphi$ if and only if $A' \models \varphi'$, in particular, testing whether $A \models \varphi$ can be performed in constant time,
- adding a tuple to a relation of $A$ can be done in constant time provided $A$ stays guarded by $G$, and
- removing a tuple from a relation of $A$ can be done in constant time.

In order to prove Theorem 21 we first establish a dynamized version of Lemma 12.

**Lemma 22.** Let $d \geq 0$ be an integer, $L$ a language, $\varphi(x_0, \ldots, x_n)$ a simple quantifier-free $L$-formula that is a conjunction of atomic formulas and their negations, and $T$ a $\varphi$-template. There exists an integer $K$ and an $\overline{L}$-formula $\overline{\varphi}_T$ such that the following holds:

- $\overline{L}$ is the language with $\overline{L}^r = L' \cup \{U_1, \ldots, U_k\}$ and $\overline{L}^f = L^f \cup \{p\}$ where $U_1, \ldots, U_k$ are new nullary or unary relations, $k \leq K$,
- $\overline{\varphi}_T$ is quantifier-free and the variables $x_1, \ldots, x_n$ are the only variables that appear freely in $\overline{\varphi}_T$, but $\overline{\varphi}_T$ need not be simple, and
- for every rooted forest $F$ with depth at most $d$ and every $L$-structure $S$ guarded by the closure of $F$, there exists an $\overline{L}$-structure $\overline{S}$ with $V(S) = V(\overline{S})$ such that for every $v_0, \ldots, v_n \in V(S)$
  $$\overline{S} \models \overline{\varphi}_T(v_1, \ldots, v_n)$$
  where $p^\overline{S}$ is the $F$-predecessor function and the relations $U_1^\overline{S}, \ldots, U_k^\overline{S}$ can be computed (by listing the singletons they contain) in linear time given $F$ and $S$. The interpretation of other symbols of $L$ is preserved in $\overline{S}$, and
adding or removing a tuple to a relation of $S$ results in adding and removing a constant number of singletons from unary relations among $U^S_1, \ldots, U^S_k$, and the changes to all relations $U^S_1, \ldots, U^S_k$ can be computed in constant time, provided $S$ stays guarded by the closure of $F$.

Proof. We need to describe how the relations $U^S_1, \ldots, U^S_k$ can be updated in constant time after adding or removing a tuple to a relation of $S$. Let us consider in more detail the case analyzed in the proof of Lemma 12 and leave to the reader the case mentioned at the end of the proof of Lemma 12. Recall (see the proof of Lemma 12 for notation) that $U_1^1(w)$ is a unary relation containing elements $w$ of $F$ at depth $d_v + 1$ such that the subtree of $w$ in $F$ contains an element $v_0$ at depth $d_{x_0}$ (in $F$) with the following properties:

- there is a ($v_0$)-admissible embedding of the template $T_0$ in $F$ for $S$, and
- all clauses appearing in the conjunction $\varphi'$ with terms from $X'_0$ and with at least one term from $X''_0$ are true with $x_0 = v_0$ and the terms $t \in X'_0 \setminus X_0$, say $\alpha_T(t) = q^k(\alpha_T(x_0))$, replaced with $p^S,k(v_0)$.

Since none of the functions of $S$ changes, the first condition cannot change when adding or removing a tuple to a relation of $S$. The second one can change only when a tuple containing a term from $X''_0$ with $x_0 = v_0$ is added or removed from a relation. Since all the values of the terms in $X''_0$ with $x_0 = v_0$ appear only in a subtree of $w$, only a single element can be added to or removed from $U_1$. Based on the tuple we add or remove, we can identify which $w$ can be added or removed to $U_1$ and using the data structure introduced in the proof of Lemma 17 we can test in constant time the existence of $v_0$ satisfying the second condition (note that the values of all terms from $X_0$ with $x_0 = v_0$ are in the subtree of $w$ and the values of the terms in $X'_0 \setminus X_0$ are on the path from $w$ to the root).

Once the relation $U_1$ is updated, the relations $U_2, \ldots, U_k$ can be updated in constant time as well: keep a counter at every vertex at depth $d_v$ determining the number of children in $U_1$.

We are now ready to prove Theorem 21.

Proof of Theorem 21. Note that when the $L$-sentence $\varphi$ is fixed in Theorem 15 the language $L'$ and the $L'$-sentence $\varphi'$ are also fixed. Hence, the only object that changes when the relations of $A$ change are relations in $A'$; the functions in $A'$ stay the same and thus the $m$-th augmentation of the graph $G$ from Theorem 15 that guards $A'$ is independent of $A$ as long as $A$ stays guarded by $G$.

We now have to inspect the proofs of Lemma 13 and Theorem 15 in more detail. Since the graph $G$ and all its augmentations appearing in the inductive proof of Theorem 15 stay the same, the coloring used in Lemma 13 also does not change when $A$ gets altered. In particular, at each step of the inductive proof of Theorem 15 every rooted forest $F$ to which Lemma 13 is applied stays the
same. In the proof of Lemma 13, we replace the use of Lemma 12 with the use of Lemma 22 and observe that every change in $S$ results in a constant number of changes in $S$ and these changes can be identified in constant time. Hence, in the inductive proof of Theorem 15, a single change in $A$ results in constantly many changes to the structure obtained in the first inductive step, which result in constantly many changes to the structure obtained the second inductive step (each change in the structure obtained in the first inductive step yields only constantly many changes), etc. Since the time to update the final $L'$-structure $A'$ is constant for each of constantly many choices that propagates through the induction from a single change of $A$, the overall update time is constant.

References

[1] B. Courcelle: The monadic second-order logic of graph I. Recognizable sets of finite graphs, Inform. and Comput. 85 (1990), 12–75.

[2] A. Dawar, M. Grohe, S. Kreutzer: Locally excluding a minor, in: Proc. LICS’07, IEEE Computer Society Press, 270–279.

[3] A. Dawar, S. Kreutzer: Parameterized Complexity of First-Order Logic, Electronic Colloquium on Computational Complexity, TR09-131 (2009).

[4] R. G. Downey, M. R. Fellows: Fixed-parameter tractability and completeness II: On completeness of W[1], Theoret. Comput. Sci. 141 (1995), 109–131.

[5] R. G. Downey, M. R. Fellows: Parameterized complexity, Springer, 1999.

[6] Z. Dvořák, K. Kawarabayashi, R. Thomas: Three-coloring triangle-free planar graphs in linear time, in: Proc. SODA’09, ACM&SIAM, 2009, 1176–1182.

[7] Z. Dvořák, D. Král’: Algorithms for classes of graphs with bounded expansion, in: Proc. WG’09, LNCS vol. 5911, Springer, 2009, 17–32.

[8] Z. Dvořák, D. Král’, R. Thomas: Coloring triangle-free graphs on surfaces, in: Proc. SODA’09, ACM&SIAM, 2009, 120–129.

[9] Z. Dvořák, D. Král’, R. Thomas: Deciding first-order properties for sparse graphs, in: Proc. 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS) (2010), 133–142.

[10] Z. Dvořák, D. Král’, R. Thomas: Three-coloring triangle-free graphs on surfaces VI. A linear-time algorithm, in preparation.

[11] D. Eppstein: Subgraph isomorphism in planar graphs and related problems, in: Proc. SODA’95, ACM&SIAM, 632–640.
[12] D. Eppstein: Subgraph isomorphism in planar graphs and related problems, J. Graph Algorithms Appl. 3 (1999), 1–27.

[13] D. Eppstein: Diameter and treewidth in minor-closed graph families, Algorithmica 27 (2000), 275–291.

[14] J. Flum, M. Grohe: Parameterized complexity theory, Birkhäuser, 2006.

[15] M. Frick, M. Grohe: Deciding first-order properties of locally tree-decomposable structures, J. ACM 48 (2001), 1184–1206.

[16] H. Gaifman: On local and non-local properties, in: Proc. Herbrands Symp. Logic Colloq., North-Holland, 1982.

[17] M. Garey, D. Johnson, L. Stockmeyer: Some simplified NP-complete graph problems, Theoret. Comput. Sci. 1 (1976) 237–267.

[18] M. Grohe and S. Kreutzer, Methods for Algorithmic Meta Theorems, To appear in Martin Grohe, Johann Makowsky (Eds), Model Theoretic Methods in Finite Combinatorics, AMS Contemporary Mathematics Series.

[19] S. Kreutzer: Algorithmic meta-theorems, to appear in a workshop volume for a workshop held in Durham 2006 as part of the Newton institute special programme on Logic and Algorithms. An extended abstract appeared in: Proc. IWPEC’08, LNCS vol. 5018, Springer, 2008, 10–12.

[20] J. Nešetřil, P. Ossona de Mendez: Linear time low tree-width partitions and algorithmic consequences, in: Proc. STOC’06, 391–400.

[21] J. Nešetřil, P. Ossona de Mendez: Grad and classes with bounded expansion I. Decompositions., Eur. J. Comb. 29 (2008), 760–776.

[22] J. Nešetřil, P. Ossona de Mendez: Grad and classes with bounded expansion II. Algorithmic aspects., Eur. J. Comb. 29 (2008), 777–791.

[23] J. Nešetřil, P. Ossona de Mendez: Grad and classes with bounded expansion III. Restricted graph homomorphism dualities., Eur. J. Comb. 29 (2008), 1012–1024.

[24] J. Nešetřil, P. Ossona de Mendez: On nowhere dense graphs, Eur. J. Comb. 32 (2011), 600–617.

[25] J. Nešetřil, P. Ossona de Mendez and D. Wood: Characterisations and Examples of Graph Classes with Bounded Expansion, preprint (arXiv:0902.3265v2).

[26] J. Nešetřil, P. Ossona de Mendez: Structural properties of sparse graphs, in: M. Grötschel, G. O. H. Katona (eds.): Building Bridges Between Mathematics and Computer Science, Bolyai Society Mathematical Studies vol. 19, Springer, 2008.
[27] R. Niedermeier: Invitation to fixed-parameter algorithms, Oxford University Press, 2006.

[28] D. Peleg: Distance-dependent distributed directories, Info. Computa. 103 (1993), 270–298.

[29] N. Roberson, P. D. Seymour: Graph minors. XIII: the disjoint paths problem, J. Combin. Theory Ser. B 63 (1995), 65–110.

[30] D. Seese: Linear time computable problems and first-order descriptions, Mathematical Structures in Computer Science, 5 (1996), 505–526.

[31] D. Wood: On the maximum number of cliques in a graph, Graphs Combin. 23 (2007), 337–352.