Expressive Completeness of Two-Variable First-Order Logic with Counting for First-Order Logic Queries on Rooted Unranked Trees

Peer-reviewed author version

HELLINGS, Jelle; GYSSENS, Marc; VAN DEN BUSSCHE, Jan & Van Gucht, Dirk (2023) Expressive Completeness of Two-Variable First-Order Logic with Counting for First-Order Logic Queries on Rooted Unranked Trees. In: 2023 38TH ANNUAL ACM/IEEE SYMPOSIUM ON LOGIC IN COMPUTER SCIENCE, LICS, IEEE, p. 1 -13.

DOI: 10.1109/LICS56636.2023.10175828
Handle: http://hdl.handle.net/1942/41525

Expressive Completeness of Two-Variable First-Order Logic with Counting for First-Order Logic Queries on Rooted Unranked Trees

Jelle Hellings
Department of Computing and Software
McMaster University
Hamilton, Ontario, Canada
jhellings@mcmaster.ca

Marc Gyssens
Data Science Institute
Hasselt University
Diepenbeek, Belgium
marc.gyssens@uhasselt.be

Jan Van den Bussche
Data Science Institute
Hasselt University
Diepenbeek, Belgium
jan.vandenbussche@uhasselt.be

Dirk Van Gucht
Luddy School of Informatics, Computing, and Engineering
Indiana University
Bloomington, IN, USA
vgucht@cs.indiana.edu

***Abstract*—We consider the class of finite, rooted, unranked, unordered, node-labeled trees. Such trees are represented as structures with only the parent-child relation, in addition to any number of unary predicates for node labels. We prove that every unary first-order query over the considered class of trees is already expressible in two-variable first-order logic with counting. Somewhat to our surprise, we have not seen this result being conjectured in the extensive literature on logics for trees. Our proof is based on a global variant of local equivalence notions on nodes of trees. This variant applies to entire trees, and involves counting ancestors of locally equivalent nodes.**

***Index Terms*—finite-variable logics, counting logics, bounded equivalence of tree nodes, bounded equivalence of trees, Ehrenfeucht-Fraïssé game, expressiveness**

I. Introduction

Trees are ubiquitous in computer science and data science and they have many applications, e.g., in databases, data structures and algorithms, program analysis, software verification, formal language theory, and linguistics.

For these purposes, various classes of trees have been studied: *node-labeled* trees, *(un)ranked* trees (nodes have a fixed (arbitrary) number of children), *(un)ordered* (siblings are (not) arranged in a linear order), *rooted* trees (there is exactly one node without a parent), and combinations thereof.

There are numerous formalisms to describe, analyze, and manipulate trees:

- *declarative, logic-based formalisms:* e.g., monadic second order logic (MSO), first-order logic (FO), temporal logics (LTL, CTL), and sub-logics of these, and finite-variable logics with or without counting quantifiers;
- *algebraic languages with navigational and path expressions:* e.g., relation algebras and XPath;
- *hybrid declarative-algebraic formalisms:* e.g., Conditional XPath; and
- *procedural formalisms:* e.g., tree automata.

Good surveys about these formalisms and their meta-theory, can be found in, e.g., [1]–[3] and [4], Chapter 7.

Essentially, two different tree representations are used in these formalisms:

- In the declarative and algebraic formalisms, a tree is specified as a (finite) structure over a vocabulary that minimally contains a binary parent-child predicate, and, in the case of node-labeled trees, unary label predicates. Depending on the subject and the requirements of study, the vocabulary may additionally contain some of the binary predicates right-sibling, left-sibling, their reflexive-transitive closures right-siblings and left-siblings, descendant, and ancestor. (One may furthermore also consider ternary predicates, such as least common ancestor [5].)
- In the procedural, automata-based, formalisms, a tree is specified as a prefix-closed set of natural number strings which "names" the nodes in a top-down, left-to-right, order. This way of encoding permits a tree-automaton to "read" the tree and make appropriate state transitions. (For more details, see, e.g., [6]).

A common theme across the study of tree formalisms is to determine the classes of boolean and unary queries that they can define. Here,

- a boolean query maps each tree to a boolean value, and, hence, can be viewed as a tool to specify the class of trees for which the query is `true`;
- a unary query maps each tree into a subset of its nodes, and, hence, can be viewed as a tool to differentiate the nodes in the output from other nodes of the tree.

It is possible that different tree formalisms express the same boolean and unary queries. In this paper, we focus on the class of unranked, unordered, rooted trees with only the parent-child relation, in addition to any number of unary node label predicates. Our main result is that every unary query expressible in first-order logic, FO, over such trees is already expressible in two-variable first-order logic with counting quantifiers, $\mathrm{FO}^2 + \mathrm{C}$. The same result for boolean queries follows readily. Of course, the same result for binary or higher-arity queries does not hold, e.g., the query asking for all sibling pairs is not expressible in $\mathrm{FO}^2 + \mathrm{C}$. However, our result does readily imply that, still over the class of trees under consideration, every k-ary FO query, $k \geq 1$, is also expressible in $\mathrm{FO}^{k+1} + \mathrm{C}$.

In Section II, below, we outline our strategy to prove the main result, and explain how the technical part of the paper is organized accordingly. Here, we wish to emphasize that the proof also entails an equivalence-preserving translation algorithm from FO to $\mathrm{FO}^2 + \mathrm{C}$. (The translation in the other direction is, of course, straightforward.)

An important source of inspiration for us was the work of Straubing [7] on strings with successor in the context of formal languages. From his work, it follows that our result was already known for boolean queries on strings with successor. However, our results are more generally proven for unary queries, and, while the case of boolean queries follows readily from that of unary queries, this is not so the other way around. Finally, we observe that our results can easily be specialized to the case of ranked, ordered trees by using special node predicates child-i to indicate that a node is to be interpreted as the ith child of its parent.

II. Proof Strategy

Here, we give an outline of our proof strategy for the main result and explain how the technical part of the paper is organized accordingly.

Our proof strategy consists of two steps.

In the first step, given $r \geq 0$, we define an equivalence relation on structures $(\mathfrak{T}, n)$, with n a node of tree $\mathfrak{T}$, such that equivalent structures cannot be distinguished by unary first-order logic sentences of quantifier rank r. This equivalence relation on trees, $\approx^{\mathrm{r}}$, is proposed in Section VII, and is based on a global equivalence notion on trees, $\approx_{b,d,k}$, defined in Section VI. It involves a local equivalence on tree nodes, $\approx_{b,d}$, in the style of similar relations in, e.g., [8]–[13], and a counting condition on ancestors at a given distance of locally equivalent tree nodes. The local equivalence relation on tree nodes is presented and studied in Section V. A key tool in this study is the concept of a bounded neighborhood of a set of nodes of a tree. The actual result of this first step is proved in Section VII by establishing a winning strategy for the Duplicator in the Ehrenfeucht-Fraïssé game of r rounds on equivalent structures.

In the second step, we show that, for each $r \geq 0$, there exists $s \geq 0$ and $c \geq 0$ such that every equivalence class $[(\mathfrak{T}, n)]$ for $\approx^{\mathrm{r}}$ can be described by a unary formula $\psi_{\mathfrak{T},n}$ with quantifier rank at most s in two-variable first-order logic with counting, in which counting is bounded by c. As a consequence of this result and the result of Section VII, a unary first-order formula φ of quantifier rank r can be written as the infinite disjuction $\bigvee_{n \in [\![\varphi]\!]_{\mathfrak{T}}} \psi_{\mathfrak{T},n}$. Without loss of generality, we may assume that the label predicates in the formulae $\psi_{\mathfrak{T},n}$ also occur in φ. Hence, given the bounds s and c on the quantifier rank and the counting, there can only be finitely many such formulae up to equivalence, yielding the desired $\mathrm{FO}^2 + \mathrm{C}$ formula. This second step in our proof strategy is the subject of Section VIII.

The main challenge in this work is to make the bounded equivalence notion (1) strong enough to ensure that the Duplicator has a winning strategy for the Ehrenfeucht-Fraïssé game in the first step of our proof strategy, yet also (2) weak enough so that it can be translated into $\mathrm{FO}^2 + \mathrm{C}$.

III. Related Work

Most work on logics for trees consider richer vocabularies than merely the parent-child relation, typically including the sibling and ancestor-descendant predicates. Indeed, these logics correspond better to automata theory, as well as to languages used in practice. Below, we restrict our discussion of results in the literature to those which our results are most closely related, i.e., those who use a similar tree model:

- Benedikt and Segoufin considered regular tree languages which are first-order definable as boolean queries[1] over node-labeled, unranked, unordered, rooted trees in a vocabulary consisting of unary label predicates and the binary parent-child predicate [11]. (A regular tree language is a language recognized by a tree automaton.) They point out that there is no obvious way to lift the notion of locally threshold testability from string to tree languages, and actually propose first-order definability (with only the parent-child relation) as a reasonable substitute. This serves as an additional motivation for this setting and for our result. Benedikt and Segoufin establish that a regular tree language is first-order definable in their setting if and only if it is aperiodic and, for some fixed value of k, closed under k-guarded horizontal and vertical swaps. So, in a sense, Benedikt and Segoufin look "from above" (regular tree languages) to find languages equivalent to FO, while we look "from below" (two-variable logic) for that purpose. In view of Benedikt and Segoufin's results, our main result establishes that tree-aperiodic, k-guarded swap-invariant regular unranked-tree languages are precisely those that can also be defined in $\mathrm{FO}^2 + \mathrm{C}$. (For further discussion of this work, we refer to the survey paper by Bojańczyk on regular tree languages satisfying invariance properties and their relationship with the expressive power of various tree logics [2].)
- Charatonik and Witkowski consider $\mathrm{FO}^2 + \mathrm{C}$ over (finite) unranked trees and forests and establish that its satisfiability problem is decidable in NEXPTIME [14].

[1] See also our comment in the final paragraph of the Introduction on unary versus boolean queries.

- Marx examined the expressive power of FO over node-labeled, unranked trees in a vocabulary containing, besides the parent-child predicate, the right-sibling, left-sibling, ancestor, and descendant predicates [15]. He established that this language has the same expressive power as Conditional XPath, which is the three-variable logic $\mathrm{FO}^3 + \mathrm{C}$ augmented with certain path expressions. Thus, Marx established a link between definability in FO and definability in an extension of FO^3. Note that Conditional XPath does not need counting quantifiers, since they can be simulated using the sibling order predicates.
- With ten Cate and de Rijke, Marx established additional results related to finite-variable FO logics and XPath [16], [17]. In their work, sibling and ancestor predicates are assumed, however. Relative to our work, the result that Core XPath is as expressive as $\mathrm{FO}^2 + \mathrm{C}$ in a vocabulary containing the parent-child, right-sibling, left-sibling, and descendant predicates is the most notable [17].
- Two-variable first-order logic with counting can be viewed as graded modal logic, duly extended with two-way and global modalities. Thus, on the surface, our result may appear related to van Benthem-like results to the effect that first-order logic formulas, invariant (over graphs) under suitable notions of bisimulation, are expressible in suitable variants of graded modal logic [18]. Our result is qualitatively different, however, since it focuses on expressiveness over trees, and does not assume any kind of invariance over graphs.

To conclude, we want to mention the work of Hellings et al. [19], [20] on Tarski's Relation Algebra on unordered trees, in which it is shown that some special sublanguages of FO^3 can already be expressed in FO^2, and the work of Gyssens et al. [12] and Fletcher et al. [13] on the navigational expressiveness (i.e., expressive power at instance rather than query level) of Core XPath with counting quantifiers on node-labeled unordered trees. These works in particular inspired us to make the conjecture which we prove in the present paper. Somewhat to our surprise, we have not seen this result being conjectured in the extensive literature on logics for trees.

IV. Preliminaries

A. Graphs and trees

In this paper, we consider a vocabulary consisting of an infinite number of unary predicate symbols $\ell_1(x), \ell_2(x), \ldots$ and one binary predicate symbol $E(x, y)$. The unary predicate symbols represent node labels, and the binary predicate the edge relation. We denote by $\mathcal{P}$ the set of all unary predicate symbols of the vocabulary. Any finite structure over this vocabulary is a (node-labeled) graph. We denote a graph as $\mathfrak{G} = (\mathcal{N}, \lambda, \mathcal{E})$, where $\mathcal{N}$ is the set of nodes of $\mathfrak{G}$, $\lambda : \mathcal{P} \to 2^{\mathcal{N}}$ the labeling function associating to each unary (label) predicate the set of nodes of $\mathfrak{G}$ satisfying it, and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ the edge relation of $\mathfrak{G}$. If, for all $\lambda \in \mathcal{P}$, $\lambda(\ell) = \emptyset$, then $\mathfrak{G}$ is called unlabeled.

Let $\mathfrak{G} = (\mathcal{N}, \lambda, \mathcal{E})$ and $N \subseteq \mathcal{N}$. The subgraph of $\mathfrak{G}$ generated by N is the graph $\mathfrak{G}_N = (N, \lambda|_N, \mathcal{E} \cap N \times N)$, where $\lambda|_N$ is the restriction of λ to N.

We are concerned with (node-labeled) unranked, unordered, rooted trees, which we shall refer to as simply "trees". A tree is therefore a graph with an anti-reflexive and anti-symmetric edge relation which is acyclic and in which exactly one node, the root, has indegree 0 and all other nodes have indegree 1. We denote a tree as $\mathfrak{T} = (\mathcal{N}, \lambda, \mathcal{E})$, where $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ is the parent-child relation of $\mathfrak{T}$.

In this paper, whenever we consider two distinct trees, $\mathfrak{T}_1 = (\mathcal{N}_1, \lambda_1, \mathcal{E}_1)$ and $\mathfrak{T}_2 = (\mathcal{N}_2, \lambda_2, \mathcal{E}_2)$, we shall assume, without loss of generality, that $\mathcal{N}_1 \cap \mathcal{N}_2 = \emptyset$ to avoid ambiguity.

Let $\mathfrak{T} = (\mathcal{N}, \lambda, \mathcal{E})$ and $m, n \in \mathcal{N}$. The distance $\Delta_{\mathfrak{T}}(m, n)$ is the sum of the lenghts of the unique paths from p to m and from p to n, where p is the closest common ancestor of m and n. If $\Delta_{\mathfrak{T}}(m, n) = 1$, we say that m and n are adjacent. Notice that this distance notion satisfies the triangle inequality: for $m, n, q \in \mathcal{N}$, $\Delta_{\mathfrak{T}}(m, n) \leq \Delta_{\mathfrak{T}}(m, q) + \Delta_{\mathfrak{T}}(q, n)$. We generalize this notion of distance, as follows. For $N \subseteq \mathcal{N}$ and $n \in \mathcal{N}$, $\Delta_{\mathfrak{T}}(N, n) = \min_{m \in N} \Delta_{\mathfrak{T}}(m, n)$. Also, for $M, N \subseteq \mathcal{N}$, $\Delta_{\mathfrak{T}}(M, N) = \min_{m \in M, n \in N} \Delta_{\mathfrak{T}}(m, n)$.

Let $\mathfrak{T} = (\mathcal{N}, \lambda, \mathcal{E})$ and $N \subseteq \mathcal{N}$. We denote by $N^{\uparrow *}$ the superset of N containing all ancestors of a node of N in $\mathfrak{T}$. For $b > 0$, we say that N is b-restricted if no more than b nodes of $N^{\uparrow *}$ are siblings in $\mathfrak{T}$. Notice that $|N| \leq b$ implies b-restrictedness, but not the other way around. For $d > 0$, we denote by $N^{\uparrow d}$ the set $\{n \in N^{\uparrow *} \mid \Delta_{\mathfrak{T}}(N, n) \leq d\}$.

Finally, let $\mathfrak{T} = (\mathcal{N}, \lambda, \mathcal{E})$ and $N \subseteq \mathcal{N}$. The set of nodes N is connected in $\mathfrak{T}$ if the subgraph of $\mathfrak{T}$ generated by N is again a tree. Let $N_1, N_2 \subseteq \mathcal{N}$. We say that N_1 is disconnected from N_2 if $N_1 \cap N_2 = \emptyset$ and no node of N_1 is adjacent to a node of N_2.

B. First-order logic and counting

First-order logic with counting is first-order logic extended with so-called counting quantifiers of the form $\exists^{\geq k}$, $k \geq 0$, with the obvious meaning. Given the vocabulary described in Subsection IV-A, formulae in first-order logic with counting are defined by the grammar

$$\varphi := \mathtt{true} \mid x = y \mid \ell(x) \mid E(x, y) \mid \varphi \wedge \varphi \mid \neg\varphi \mid \exists^{\geq k} x\, \varphi,$$

in which x and y are variables, $\ell \in \mathcal{P}$, and $k \geq 0$.[2] Throughout this work, we use the customary abbreviations as well as the following shorthand: $\exists^{=k} x\, \varphi := \left(\exists^{\geq k} x\, \varphi\right) \wedge \neg \left(\exists^{\geq k+1} x\, \varphi\right)$. For a formula φ in first-order logic with counting, $\mathsf{qr}(\varphi)$ denotes the quantifier rank of φ.

A formula $\varphi(x_1, \ldots, x_m)$ in first-order logic with counting with free variables $x_1, \ldots, x_m$ is interpreted as an m-ary query on graphs (hence, in particular, also on trees), where, given a graph $\mathfrak{G} = (\mathcal{N}, \lambda, \mathcal{E})$, the evaluation of φ on $\mathfrak{G}$ is defined as $[\![\varphi]\!]_{\mathfrak{G}} = \{(n_1, \ldots, n_m) \mid \mathfrak{G} \models \varphi(n_1, \ldots, n_m)\}$, which is an m-ary relation on $\mathcal{N}$. In particular, a boolean query is a nullary query where the only two possible evaluation results, $\{()\}$

[2] The traditional existential quantifier $\exists$ has been omitted from the grammar, because it is equivalent to $\exists^{\geq 1}$, but will be used as an abbreviation.

and $\emptyset$, are interpreted as `true` and `false`, respectively. For unary queries, the evaluation result—strictly speaking a unary relation on $\mathcal{N}$—may alternatively be interpreted as a set of nodes.

We write FO + C to denote the set of all first-order with counting queries, and FO to denote the set of all first-order queries. Obviously, FO + C is no more expressive than FO. We write FO^k + C to denote the set of all FO + C queries which use at most k distinct variables. Finally, we write FO^k to denote the set of all FO queries which use at most k distinct variables.

In what follows, we shall not distinguish between a formula and the corresponding query.

Finally, we say that two queries φ and ψ are equivalent on trees if, for all trees $\mathfrak{T}$, $[\![\varphi]\!]_{\mathfrak{T}} = [\![\psi]\!]_{\mathfrak{T}}$.

V. Bounded Equivalence of Tree Nodes

In this section, we develop a toolkit that allows us to look "locally" at trees, by only taking into account nodes that are within a given distance of a given node. In addition, when comparing the numbers of children of nodes, we only count up to a given bound.

At the basis of this toolkit is an equivalence notion on tree nodes, variations of which have been considered in, e.g., [8]–[13]. This bounded equivalence notion is built up in two steps. First, we only take into account descendants of a given tree node. This leads to the notion of downward bounded equivalence (Subsection V-A). Subsequently, we use this notion as a stepping stone to define general bounded equivalence (Subsection V-B).

We then define bounded neighborhoods of a set of tree nodes (Subsection V-C). In a sense that we shall make precise, bounded neighborhoods "witness" bounded equivalence.

Throughout this section, we assume $b \geq 2$ and $d \geq 0$ for the counting bound, respectively the distance within which we take nodes into account.

A. Downward Bounded Equivalence

We define downward (b, d)-bounded equivalence of two tree nodes recursively. These nodes may belong to the same tree or to different trees. For stating the definition, the following notions turns out to be very handy:

Notation 1. Let $\mathfrak{T} = (\mathcal{N}, \lambda, \mathcal{E})$ and $n \in \mathcal{N}$. We denote by $[n]_{\downarrow b,d}$ the set of all siblings of n which are downward (b, d)-equivalent to n.

Definition 2. Let $\mathfrak{T}_1 = (\mathcal{N}_1, \lambda_1, \mathcal{E}_1)$, $\mathfrak{T}_2 = (\mathcal{N}_2, \lambda_2, \mathcal{E}_2)$, and $n_1, n_2 \in \mathcal{N}_1 \cup \mathcal{N}_2$. We define downward (b, d)-bounded equivalence of n_1 and n_2, denoted $n_1 \approx_{\downarrow b,d} n_2$, recursively on d.

1) $n_1 \approx_{\downarrow b,0} n_2$ if, for all $\ell \in \mathcal{P}$, $\ell(n_1) \Leftrightarrow \ell(n_2)$;
2) if $d > 0$, then $n_1 \approx_{\downarrow b,d} n_2$ if, for all $\ell \in \mathcal{P}$, $\ell(n_1) \Leftrightarrow \ell(n_2)$ and
   a) for each child m_1 of n_1, there is a child m_2 of n_2 such that $m_1 \approx_{\downarrow b,d-1} m_2$;
   b) for each child m_2 of n_2, there is a child m_1 of n_1 such that $m_1 \approx_{\downarrow b,d-1} m_2$;
   c) for each child m_1 of n_1 and m_2 of n_2 for which $m_1 \approx_{\downarrow b,d-1} m_2$, $\min(|[m_1]_{\downarrow b,d-1}|, b) = \min(|[m_2]_{\downarrow b,d-1}|, b)$.

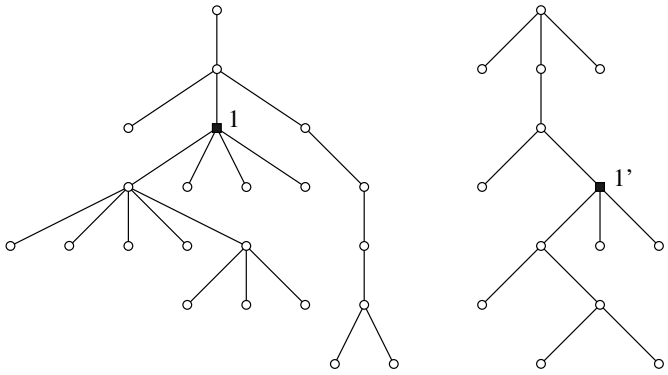

Fig. 1. The filled square nodes in the trees, marked 1 and $1'$, are downward $(2, 2)$-bounded equivalent. Moreover, they are also $(2, 2)$-bounded equivalent.

We emphasize that downward (b, d)-bounded equivalence can be considered for two nodes of two different trees or two nodes of the same tree (which we actually do to state Condition 2c of Definition 2). This relation has the following properties:

Proposition 3. *Let $\mathfrak{T}_1 = (\mathcal{N}_1, \lambda_1, \mathcal{E}_1)$ and $\mathfrak{T}_2 = (\mathcal{N}_2, \lambda_2, \mathcal{E}_2)$.*

1) *The relation $\approx_{\downarrow b,d}$ is an equivalence relation on $\mathcal{N}_1 \cup \mathcal{N}_2$.*
2) *If $d > d' \geq 0$, then $n_1 \approx_{\downarrow b,d} n_2$ implies $n_1 \approx_{\downarrow b,d'} n_2$; hence, the equivalence relation $\approx_{\downarrow b,d}$ on $\mathcal{N}_1 \cup \mathcal{N}_2$ is a refinement of the equivalence relation $\approx_{\downarrow b,d'}$ on $\mathcal{N}_1 \cup \mathcal{N}_2$.*

We now illustrate the notion of downward bounded equivalence by an example.

Example 4. Consider Fig. 1 showing two, for the sake of simplicity, unlabeled trees. In unlabeled trees, all nodes are mutually downward $(2, 0)$-bounded equivalent. Hence, there are only three equivalence classes for downward $(2, 1)$-bounded equivalence: the leaf nodes, the nodes with exactly one child, and the nodes with at least 2 children. Let us call these nodes of type 0, type 1, and type 2, respectively. Now, consider the two black square nodes, marked 1 and $1'$. They have each at least 2 children of type 0, no children of type 1, and 1 child of type 2. Hence, these nodes are downward $(2, 2)$-bounded equivalent.

B. Bounded Equivalence

To decide downward (b, d)-bounded equivalence, we only looked at *descendants* at distance at most d from the nodes concerned. We now use this notion as a stepping stone to define general (b, d)-bounded equivalence, for which we look at *all* nodes at distance at most d from the nodes concerned.

Definition 5. Let $\mathfrak{T}_1 = (\mathcal{N}_1, \lambda_1, \mathcal{E}_1)$, $\mathfrak{T}_2 = (\mathcal{N}_2, \lambda_2, \mathcal{E}_2)$, $n_1 \in \mathcal{N}_1$, and $n_2 \in \mathcal{N}_2$. We define (b, d)-bounded equivalence of n_1 and n_2, denoted $n_1 \approx_{b,d} n_2$, recursively on d.

1) $n_1 \approx_{b,0} n_2$ if $n_1 \approx_{\downarrow b,0} n_2$ (i.e., for all $\ell \in \mathcal{P}$, $\ell(n_1) \Leftrightarrow \ell(n_2)$);
2) if $d > 0$, then $n_1 \approx_{b,d} n_2$ if $n_1 \approx_{\downarrow b,d} n_2$ and either
   a) n_1 is the root of $\mathfrak{T}_1$ and n_2 is the root of $\mathfrak{T}_2$, or
   b) n_1 is not the root of $\mathfrak{T}_1$ and n_2 is not the root of $\mathfrak{T}_2$ and $m_1 \approx_{b,d-1} m_2$, where m_1 is the parent of n_1 in $\mathfrak{T}_1$ and m_2 is the parent of n_2 in $\mathfrak{T}_2$.

Of course, $\mathfrak{T}_1$ and $\mathfrak{T}_2$ may be the same tree or different trees in Definition 5. Hence, given $\mathfrak{T}_1 = (\mathcal{N}_1, \lambda_1, \mathcal{E}_1)$ and $\mathfrak{T}_2 = (\mathcal{N}_2, \lambda_2, \mathcal{E}_2)$, it makes sense to consider (b,d)-bounded equivalence as a relation on $\mathcal{N}_1 \cup \mathcal{N}_2$. We can then state the following analogon to Proposition 3:

Proposition 6. *Let $\mathfrak{T}_1 = (\mathcal{N}_1, \lambda_1, \mathcal{E}_1)$ and $\mathfrak{T}_2 = (\mathcal{N}_2, \lambda_2, \mathcal{E}_2)$.*
1) *The relation $\approx_{b,d}$ is an equivalence relation on $\mathcal{N}_1 \cup \mathcal{N}_2$.*
2) *If $d > d' \geq 0$, then $n_1 \approx_{b,d} n_2$ implies $n_1 \approx_{b,d'} n_2$; hence, the equivalence relation $\approx_{b,d}$ on $\mathcal{N}_1 \cup \mathcal{N}_2$ is a refinement of the equivalence relation $\approx_{b,d'}$ on $\mathcal{N}_1 \cup \mathcal{N}_2$.*

The correctness of Proposition 6 follows from Proposition 3 and a straightforward inductive argument.

Example 7. Consider again Fig. 1 used in Example 4. We already established that both black square nodes, marked 1 and $1'$, are downward $(2,2)$-bounded equivalent. They are not the root of their respective tree. Using the terminology in Example 4, the parents of nodes 1 and $1'$ are both of type 2 and, hence, are downward $(2,1)$-bounded equivalent. These nodes are again not the root of their respective tree. Hence, their parents (i.e., the grandparents of nodes 1 and $1'$) exist, and, since the trees in Fig. 1 are unlabeled, they are $(2,0)$-bounded equivalent. Hence, the parents of nodes 1 and $1'$ are $(2,1)$-bounded equivalent, and nodes 1 and $1'$ themselves are $(2,2)$-bounded equivalent.

C. Bounded Neighborhoods

In this subsection, we introduce bounded neighborhoods of sets of nodes of a tree as a tool to "witness" bounded equivalence in a sense that will be made precise. Of particular interest are bounded neighborhoods of single nodes.

Definition 8. Let $\mathfrak{T} = (\mathcal{N}, \lambda, \mathcal{E})$ and $N \subseteq P \subseteq \mathcal{N}$ such that P is b-restricted. A set $B \subseteq \mathcal{N}$ is a (b,d)-neighborhood (nbhd) of N in $\mathfrak{T}$ with respect to (w.r.t.) P if it satisfies the following conditions:
1) all nodes of B are at distance at most d of N;
2) all nodes of $N^{\uparrow d}$ are in B;
3) all nodes of $B \setminus N^{\uparrow d}$ have a parent in $\mathfrak{T}$ which is also in B;
4) if $p \in B$ and $\Delta_{\mathfrak{T}}(N,p) < d$, then, for each child m of p in $\mathfrak{T}$, B contains all nodes of $[m]_{\downarrow b,d-d'} \cap P^{\uparrow *}$, where $d' = \Delta_{\mathfrak{T}}(N,p) + 1$;
5) if $p \in B$ and $\Delta_{\mathfrak{T}}(N,p) < d$, then, for each child m of p in $\mathfrak{T}$, B contains exactly $\min(|[m]_{\downarrow b,d-d'}|, b)$ nodes of $[m]_{\downarrow b,d-d'}$, where $d' = \Delta_{\mathfrak{T}}(N,p) + 1$.

Condition 5 suggests that, in general, not all nodes of $[m]_{\downarrow b,d-d'}$ can be part of B. To have some control over which nodes are in B, the reference set P has been introduced in Definition 8. Indeed, Condition 4 states that all nodes of $[m]_{\downarrow b,d-d'} \cap P^{\uparrow *}$ must be in B.

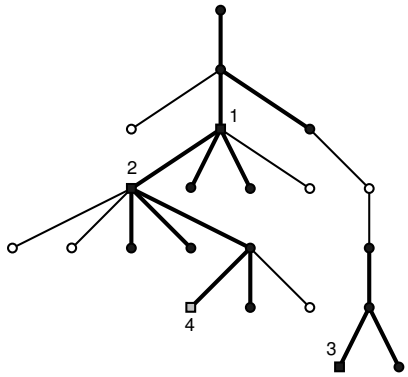

Fig. 2. The set of all filled nodes in Fig. 2 is a $(2,2)$-nbhd of the set $N = \{1,2,3\}$ of the three black square nodes in the tree shown w.r.t. $P = \{1,2,3,4\}$, i.e., N augmented with the grey square node. For emphasis, the subgraph generated by the neighborhood is visualized by thicker edges.

Of particular interest in this paper are (b,d)-nbhds of singleton sets. If, in Definition 8, there is $n \in \mathcal{N}$ such that $N = \{n\}$, then we shall talk about a (b,d)-nbhd of n in $\mathfrak{T}$ w.r.t. P rather than a (b,d)-nbhd of $\{n\}$ in $\mathfrak{T}$ w.r.t. P.

Definition 8 is declarative in nature, which is helpful for proving properties of (b,d)-bounded nbhds, but not for constructing them. Therefore, we present an algorithm below which is guaranteed to generate a (b,d)-bounded nbhd.

Algorithm 9. *Input:* $\mathfrak{T} = (\mathcal{N}, \lambda, \mathcal{E})$; $N \subseteq P \subseteq \mathcal{N}$ with P b-restricted.
Output: a (b,d)-nbhd B of N in $\mathfrak{T}$ w.r.t. P.
Method:
1) Initialize B as $N^{\uparrow d}$, and mark all these nodes as visited;
2) for $i := 0$ to $d-1$, do, for each $p \in B$ with $\Delta_{\mathfrak{T}}(N,p) = i$, and for each unvisited child $m \in \mathcal{N}$ of p,
   a) add to B all unvisited nodes of $[m]_{\downarrow b,d-i-1} \cap P^{\uparrow *}$;
   b) add to B other unmarked nodes of $[m]_{\downarrow b,d-i-1}$ until there are exactly $\min(|[m]_{\downarrow b,d-i-1}|, b)$ nodes of $[m]_{\downarrow b,d-i-1}$ in B;
   c) mark all nodes of $[m]_{\downarrow b,d-i-1}$ as visited.

Proposition 10. *Algorithm 9 is correct.*

Example 11. In Fig. 2, a $(2,2)$-nbhd is exhibited of the set $N = \{1,2,3\}$ of the black square nodes in the tree shown w.r.t. $P = \{1,2,3,4\}$, i.e., N augmented with the grey square node. This bounded neighborhood can be constructed using Algorithm 9.

Here are some useful properties of bounded neighborhoods:

Proposition 12. *Let $\mathfrak{T} = (\mathcal{N}, \lambda, \mathcal{E})$ and $N \subseteq P \subseteq \mathcal{N}$ with P b-restricted. Let B a (b,d)-nbhd of N in $\mathfrak{T}$ w.r.t. P.*
1) *If $m \in B$, then all ancestors of m up to the closest ancestor in $N^{\uparrow d}$ are also in B.*
2) *All nodes of $P^{\uparrow *}$ at distance at most d of N are in B.*

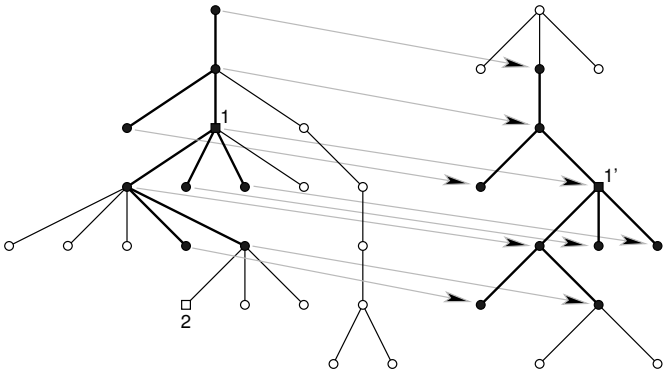

Fig. 3. In the left tree, the set of all filled nodes is a $(2,2)$-nbhd of the black square node marked 1 w.r.t. $P_1 = \{1,2\}$, i.e., the set consisting of the square nodes. In the right tree, the set of all filled nodes is a $(2,2)$-nbhd of the square node marked $1'$ w.r.t. $P_2 = \{1'\}$. For emphasis, the subgraphs generated by the neighborhoods are visualized by thicker edges. The grey arrows represent a bijection between both bounded neighborhoods satisfying Proposition 13.

3) *If $p \in B$ and $\Delta_{\mathfrak{T}}(N,p) \leq d' < d$, then, for each child m of p in $\mathfrak{T}$, B contains at least $\min(|[m]_{\downarrow b,d-d'-1}|, b)$ nodes of $[m]_{\downarrow b,d-d'-1}$.*
4) *If $m \in \mathcal{N}$ and $\Delta_{\mathfrak{T}}(N,m) \leq d$, then, for all $n \in N$ with $\Delta_{\mathfrak{T}}(n,m) \leq d$, there exists $m' \in B$ such that n and m have the same closest common ancestor as n and m', $\Delta_{\mathfrak{T}}(n,m) = \Delta_{\mathfrak{T}}(n,m')$, and $m \approx_{b,d-\Delta_{\mathfrak{T}}(n,m)} m'$.*

Observe that, in general, a node at distance at most d of N is not necessarily in B. However, if this node happens to be in P, it must be in B, by Statement 2. We will exploit this in the proof of Theorem 32.

We now explain how bounded neighborhoods of single nodes "witness" bounded equivalence:

Proposition 13. *Let $\mathfrak{T}_1 = (\mathcal{N}_1, \lambda_1, \mathcal{E}_1)$, $\mathfrak{T}_2 = (\mathcal{N}_2, \lambda_2, \mathcal{E}_2)$, $n_1 \in \mathcal{N}_1$, $n_2 \in \mathcal{N}_2$, $P_1 \subseteq \mathcal{N}_1$, $P_2 \subseteq \mathcal{N}_2$, $n_1 \in P_1$, $n_2 \in P_2$ with P_1 and P_2 b-restricted. Let B_1 be a (b,d)-nbhd of n_1 in $\mathfrak{T}_1$ w.r.t. P_1 and B_2 a (b,d)-nbhd of n_2 in $\mathfrak{T}_2$ w.r.t. P_2. If $n_1 \approx_{b,d} n_2$, then there exists a bijection f between B_1 and B_2 which maps n_1 to n_2, which is a partial isomorphism between $\mathfrak{T}_1$ and $\mathfrak{T}_2$, and which satisfies, for each node m_1 of B_1, $m_1 \approx_{b,d-\Delta_{\mathfrak{T}_1}(n_1,m_1)} f(m_1)$.*

Example 14. Consider again the trees shown in Fig. 1. In Example 7, we argued that the black square nodes marked 1 and $1'$ are $(2,2)$-bounded equivalent. In Fig. 3, we have added $(2,2)$-nbhds to each of these nodes. Now, there must be a bijection between both neighborhoods satisfying Proposition 13. The grey arrows in Fig. 3 exhibit such a bijection.

Unfortunately, there is no obvious way to generalize Proposition 13 to bounded neighborhoods of arbitrary sets of nodes. What we can do, however, is make the connection between a bounded neighborhood of an arbitrary set of nodes and the bounded neighborhoods of the nodes of which this set is composed more precise.

Proposition 15. *Let $\mathfrak{T} = (\mathcal{N}, \lambda, \mathcal{E})$ and $N \subseteq P \subseteq \mathcal{N}$ with P b-restricted. Let B be a (b,d)-nbhd of N in $\mathfrak{T}$ w.r.t. P. Then, for each $n \in N$, there exists a (b,d)-nbhd B_n of n in $\mathfrak{T}$ w.r.t. P such that $B = \bigcup_{n \in N} B_n$.*

So, a bounded neighborhood of a set of nodes is always the union of bounded neighborhoods of the nodes it contains, which, to some extent, can be seen as a "soundness" condition for Definition 8.

Example 16. Consider again Example 11, exhibiting a tree with a $(2,2)$-nbhd of a set $N = \{1,2,3\}$ of three nodes w.r.t. a set $P = \{1,2,3,4\}$. The graphical illustration of that example is copied in the leftmost panel of Fig. 4. The three other panels show $(2,2)$-nbhds of the black square nodes 1, 2, and 3 individually, all w.r.t. P, satisfying Property 15. The $(2,2)$-nbhd of N w.r.t. P, to the left, is indeed the union of the $(2,2)$-nbhds of 1, 2, respectively, 3, w.r.t. P, to the right.

At this point, one could have wondered why we did not simply *define* a bounded neighborhood of a set of nodes as the union of bounded neighborhoods of its individual nodes. The problem with this alternative approach is that, whenever Algorithm 9 leaves some choice in Step 2b as to which nodes are added to the neighborhoods, we cannot enforce any "coordination" between these choices, and, hence, their union may contain more nodes than strictly necessary. As a consequence, Proposition 20, a key result in many proofs, would no longer hold.

Example 17. Continuing with Example 16, consider the $(2,2)$-nbhds of the nodes 1, 2, and 3 individually, all w.r.t. $P = \{1,2,3,4\}$, shown in the three panels to the right in Fig. 5. Their union is *not* equal to the $(2,2)$-nbhd of $N = \{1,2,3\}$ w.r.t. P with which we started in Example 16, and which is shown in the leftmost panel of Fig. 4. It is not even a superset of that. Moreover, it contains more nodes than the $(2,2)$-nbhd in the leftmost panel of Fig. 4, which is indicative of the lack of "coordination" between the choices made to construct the bounded neighborhoods of nodes 1, 2, and 3 individually in the right three panels of Fig. 5.

The idea that the union of arbitrary bounded neighborhoods of individual neighborhoods is in general "redundant" is captured by the following result:

Proposition 18. *Let $\mathfrak{T} = (\mathcal{N}, \lambda, \mathcal{E})$, $N \subseteq P \subseteq \mathcal{N}$ with P b-restricted. Let, for $n \in N$, B_n be a (b,d)-nbhd of n in $\mathfrak{T}$ w.r.t. P. Then, there exists a (b,d)-nbhd B of N in $\mathfrak{T}$ w.r.t. P for which $B \subseteq \bigcup_{n \in N} B_n$.*

Example 19. Continuing with Example 17, the leftmost panel of Fig. 5 shows a $(2,2)$-nbhd of the set $N = \{1,2,3\}$ w.r.t. $P = \{1,2,3,4\}$ which is contained in the union of the $(2,2)$-nbhds of of the red nodes 1, 2, and 3 individually, all also w.r.t. P, shown in the other panels. Notice that the $(2,2)$-nbhd of N w.r.t. P in Fig. 5 has the same number of nodes as the $(2,2)$-nbhd of N w.r.t. P in Fig. 4.

It is perhaps useful to point out that, if the neighborhoods of the individual nodes would not all have been taken with respect to reference sets containing N, it is possible that their

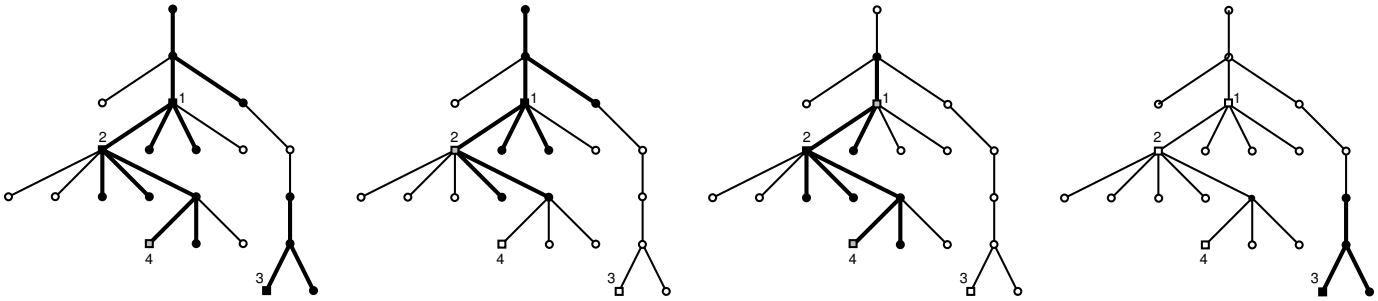

Fig. 4. In the left-most tree, we see the $(2,2)$-nbhd of the set $N = \{1,2,3\}$ of the black square nodes w.r.t. $P = \{1,2,3,4\}$, i.e., N augmented with the grey square node, shown in Fig. 2. In the three copies of this tree to the right, we see $(2,2)$-nbhds of the red nodes 1, 2, and 3 individually, all w.r.t. P, satisfying Property 15. The same graphical conventions were used as in Figs. 2 and 3.

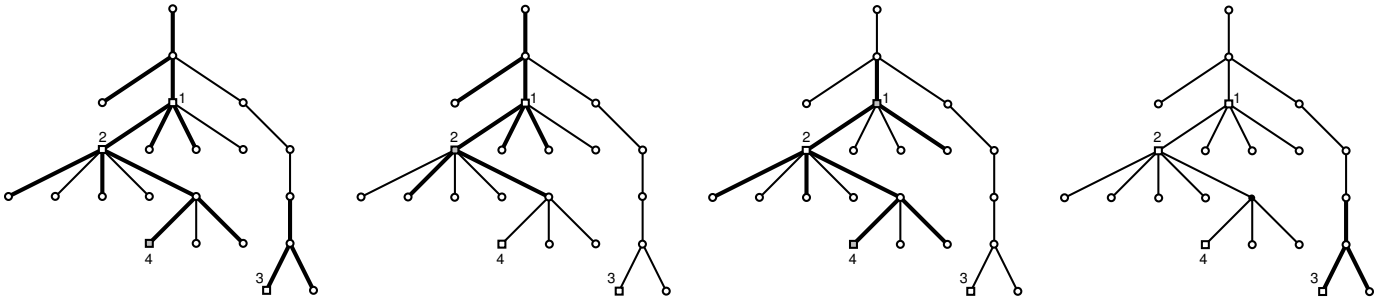

Fig. 5. This figure should be compared with Fig. 4. The three panels to the right show $(2,2)$-nbhs of nodes 1, 2, and 3 individually, all w.r.t. $P = \{1,2,3,4\}$. Observe that their union is *not* equal to the $(2,2)$-nbhd of the set $N = \{1,2,3\}$ w.r.t. P exhibited in the leftmost panel of Fig. 4. It is not even a superset of that. According to Proposition 18, however, there exists a $(2,2)$-nbhd of N w.r.t. P which is *contained* in the union of these three bounded neighborhoods. Such a bounded neighborhood is shown in the leftmost panel.

union does *not* contain a neighborhood of N, regardless of the reference set.

The following result states that all bounded neighborhoods of the same set of nodes are actually isomorphic:

Proposition 20. *Let $\mathfrak{T} = (\mathcal{N}, \lambda, \mathcal{E})$, $N \subseteq P_1 \subseteq \mathcal{N}$, $N \subseteq P_2 \subseteq \mathcal{N}$ with P_1 and P_2 b-restricted. Let B_1 and B_2 be (b,d)-nbhds of N in $\mathfrak{T}$ w.r.t. P_1, respectively P_2. Then, there exists a bijection f between B_1 and B_2 which fixes all nodes of N, which is a partial automorphism of $\mathfrak{T}$, and which satisfies, for each node m_1 of B_1, $m_1 \approx_{b,d-\Delta_{\mathfrak{T}}(N,m_1)} f(m_1)$.*

Like Proposition 13, Proposition 20 holds irrespective of the reference sets P_1 and P_2. We will exploit this in the proof of Theorem 32 to introduce additional values in the reference set.

Example 21. The left panels of Figs. 4 and 5 show $(2,2)$-nbhds of the same set of nodes in the same tree, and they are indeed isomorphic.

From Propositions 15, 18, and 20, we can immediately derive the following corollary, which provides a justification for the notion of bounded neighborhood in Definition 8:

Corollary 22. *Let $\mathfrak{T} = (\mathcal{N}, \lambda, \mathcal{E})$ and $N \subseteq P$ with P b-restricted. For $n \in N$, let B_n be a (b,d)-nbhd of n w.r.t. P. Then, $\bigcup_{n\in N} B_n$ is a (b,d)-nbhd of N w.r.t. P if and only if $|\bigcup_{n\in N} B_n|$ is minimal. Moreover, all (b,d)-nbhds of N w.r.t. P are isomorphic.*

Propositions 15 and/or 18 can sometimes be used to bootstrap results from the level of bounded neighborhoods of single nodes to the level of bounded neighborhoods of sets of nodes. We present an example of this to obtain a result which we need later on.

Proposition 23. *Let $\mathfrak{T} = (\mathcal{N}, \lambda, \mathcal{E})$ and $N \subseteq P \subseteq \mathcal{N}$ with P b-restricted. Let B a (b,d)-nbhd of N in $\mathfrak{T}$ w.r.t. P, $m \in B \cap P$, and $d' \leq d - \Delta_{\mathfrak{T}}(N,m)$. Then, B contains a (b,d')-nbhd of m in $\mathfrak{T}$ w.r.t. P.*

In combination with Proposition 18, the following is now an immediate corollary to Proposition 23:

Corollary 24. *Let $\mathfrak{T} = (\mathcal{N}, \lambda, \mathcal{E})$ and $N \subseteq P \subseteq \mathcal{N}$ with P b-connected. Let B be a (b,d)-nbhd of N w.r.t. P. Let $N' \subseteq P$ such that, for all $n' \in N'$, $d' \leq d - \Delta_{\mathfrak{T}}(N,n')$. Then, B contains a (b,d')-nbhd of N' in $\mathfrak{T}$ w.r.t. P.*

To complete our study of bounded neighborhoods, we need two results relating to connectedness.

Proposition 25. *Let $\mathfrak{T} = (\mathcal{N}, \lambda, \mathcal{E})$ and $N_1, N_2 \subseteq P \subseteq \mathcal{N}$ with P b-restricted. Let B_{N_1} and B_{N_2} be (b,d)-nbhds of N_1 and N_2, in $\mathfrak{T}$ w.r.t. P. Then B_{N_1} and B_{N_2} are disconnected if only if $\Delta_{\mathfrak{T}}(N_1,N_2) > 2d+1$.*

Proposition 26. *Let $\mathfrak{T} = (\mathcal{N}, \lambda, \mathcal{E})$ and $N_1, N_2 \subseteq P \subseteq \mathcal{N}$ with P b-restricted. Let B_{N_1} and B_{N_2} be (b,d)-nbhds of N_1 and N_2, in $\mathfrak{T}$ w.r.t. P. If B_{N_1} and B_{N_2} are disconnected, then $B_{N_1} \cup B_{N_2}$ is a (b,d)-nbhd of $N_1 \cup N_2$ in $\mathfrak{T}$ w.r.t. P.*

Example 27. Consider again Example 11 and Fig. 2. The subgraph generated by the exhibited $(2,2)$-nbhd of the set $N = \{1,2,3\}$ w.r.t. $P = \{1,2,3,4\}$ is disconnected. One can easily verify that the set of nodes of the connected component "to the left" in the tree is a $(2,2)$-nbhd of $N' = \{1,2\}$ w.r.t. P. Similarly, the set of nodes of the connected component "to the bottom right" in the tree is a $(2,2)$-nbhd of node 3 w.r.t. P. In accordance with Proposition 25, $\Delta_{\mathfrak{T}}(N',3) = 6 > 2.2+1$. Also, the union of both neighborhoods is a $(2,2)$-nbhd of $N = N' \cup \{3\}$ w.r.t. P, in accordance with Proposition 26.

We need one more result, which follows from Propositions 15, 18, 20, 25, and 26.

Corollary 28. *Let $\mathfrak{T} = (\mathcal{N}, \lambda, \mathcal{E})$ and $N \subseteq P \subseteq \mathcal{N}$ with P b-restricted. Let B_1 be a (b,d)-nbhd of N in $\mathfrak{T}$ w.r.t. P. Let $p \in B_1$ such that $P \cup \{p\}$ is also b-restricted. Then, there exists a (b,d)-nbhd B_2 of N in $\mathfrak{T}$ w.r.t. $P \cup \{p\}$ containing p and a bijection f between B_1 and B_2 which fixes all nodes of N, which is a partial automorphism of $\mathfrak{T}$, and which satisfies, for each node m_1 of B_1, $m_1 \approx_{b,d-\Delta_{\mathfrak{T}}(N,m_1)} f(m_1)$.*

Like Proposition 13, we will use Corollary 28 in the proof of Theorem 32 to introduce additional values in the reference set.

VI. Bounded Equivalence of Trees

We are now going to define a "global" equivalence on trees, based on the "local" bounded equivalence notion elaborated upon in Section V and a counting condition which does not directly involve bounded equivalent nodes, but their ancestors at a given distance.

Definition 29. Let $\mathfrak{T}_1 = (\mathcal{N}_1, \lambda_1, \mathcal{E}_1)$ and $\mathfrak{T}_2 = (\mathcal{N}_2, \lambda_2, \mathcal{E}_2)$. Let $b \geq 2$ and $d, k \geq 0$. Then $\mathfrak{T}_1$ and $\mathfrak{T}_2$ are (b,d,k)-bounded equivalent, denoted $\mathfrak{T}_1 \approx_{b,d,k} \mathfrak{T}_2$, if

1) for each node $m_1 \in \mathcal{N}_1$, there is a node $m_2 \in \mathcal{N}_2$ such that $m_1 \approx_{b,d} m_2$, and vice versa;[3]
2) for each node $m \in \mathcal{N}_1 \cup \mathcal{N}_2$ and for each $d' = 0, \ldots, d$ the number of ancestors at distance $2d'+1$ of nodes of $\mathfrak{T}_1$ that are (b,d')-bounded equivalent to m and the number of ancestors at distance $2d'+1$ of nodes of $\mathfrak{T}_2$ that are (b,d')-bounded equivalent to m are either equal or both at least k.

Bounded-equivalent trees must resemble each other close to the root, in the following sense:

Proposition 30. *Let $\mathfrak{T}_1 = (\mathcal{N}_1, \lambda_1, \mathcal{E}_1)$ and $\mathfrak{T}_2 = (\mathcal{N}_2, \lambda_2, \mathcal{E}_2)$. Let $b \geq 2$ and $d, k \geq 0$, and let $\mathfrak{T}_1 \approx_{b,d,k} \mathfrak{T}_2$. Let r_1 be the root of $\mathfrak{T}_1$ and r_2 the root of $\mathfrak{T}_2$. Let $n_1 \in \mathcal{N}_1$ be such that $\Delta_{\mathfrak{T}_1}(r_1, n_1) \leq d$. Then, there exists $n_2 \in \mathcal{N}_2$ such that $\Delta_{\mathfrak{T}_1}(r_1, n_1) = \Delta_{\mathfrak{T}_2}(r_2, n_2)$ and $n_1 \approx_{b,d-\Delta_{\mathfrak{T}_1}(r_1,n_1)} n_2$.*

Proof: From Condition 1 of Definition 29, it follows that $r_1 \approx_{b,d} r_2$. Now, let B_1 be a (b,d)-nbhd of r_1 in $\mathfrak{T}_1$ with respect to $\{r_1, n_1\}$ and B_2 a (b,d)-nbhd of r_2 in $\mathfrak{T}_2$ with respect to $\{r_2\}$. By Statement 2 of Proposition 12, $n_1 \in B_1$. By Proposition 13, there is a bijection $f : B_1 \to B_2$ mapping r_1 to r_2 which is a partial isomorphism between $\mathfrak{T}_1$ and $\mathfrak{T}_2$ satisfying, for all $p_1 \in B_1$, $p_1 \approx_{b,d-\Delta_{\mathfrak{T}_1}(r_1,p_1)} f(p_1)$. Hence, $n_2 := f(n_1)$ satisfies all the requirements. ∎

[3] As a consequence (Proposition 6), for all $d' = 0, \ldots, d$, we also have that, for each node $m_1 \in \mathcal{N}_1$, there is a node $m_2 \in \mathcal{N}_2$ such that $m_1 \approx_{b,d'} m_2$, and vice versa.

VII. First-Order Indistinguishability of Bounded-Equivalent Trees

In this section, we define, for $r \geq 0$, $\approx^{\mathrm{r}}$-equivalence on structures $(\mathfrak{T}, n)$, with n an node of tree $\mathfrak{T}$. We show that $\approx^{\mathrm{r}}$-equivalent structures cannot be distinguished by unary FO queries of quantifier rank r. To establish this, we show that the Duplicator has a winning strategy on the Ehrenfeucht-Fraïssé game (e.g., [4], Chapter 3) of r rounds on $\approx^{\mathrm{r}}$-equivalent structures.

Definition 31. Let $\mathfrak{T}_1 = (\mathcal{N}_1, \lambda_1, \mathcal{E}_1)$, $\mathfrak{T}_2 = (\mathcal{N}_2, \lambda_2, \mathcal{E}_2)$, $n_1 \in \mathcal{N}_1$, and $n_2 \in \mathcal{N}_2$. Let $d = 7^r - 1$, $b = r+2$, and $k = 4d+4$. Then, $(\mathfrak{T}_1, n_1) \approx^{\mathrm{r}} (\mathfrak{T}_2, n_2)$ if $\mathfrak{T}_1 \approx_{b,d,k} \mathfrak{T}_2$ and $n_1 \approx_{b,d} n_2$.

Theorem 32. *Let $\mathfrak{T}_1 = (\mathcal{N}_1, \lambda_1, \mathcal{E}_1)$, $\mathfrak{T}_2 = (\mathcal{N}_2, \lambda_2, \mathcal{E}_2)$, $n_1 \in \mathcal{N}_1$, and $n_2 \in \mathcal{N}_2$. If $(\mathfrak{T}_1, n_1) \approx^{\mathrm{r}} (\mathfrak{T}_2, n_2)$, then the Duplicator has a winning strategy on the Ehrenfeucht-Fraïssé game of r rounds on $(\mathfrak{T}_1, n_1)$ and $(\mathfrak{T}_2, n_2)$.*

Proof: For convenience, we play the game on $\mathfrak{T}_1$ and $\mathfrak{T}_2$ to which we add a 0th round in which n_1 and n_2 are chosen. For $i = 0, \ldots, r$, we denote the nodes chosen in the ith round in $\mathfrak{T}_1$ and $\mathfrak{T}_2$ by n_{1i} and n_{2i}, respectively. We also put $N_{1i} = \{n_{10}, \ldots, n_{1i}\}$ and $N_{2i} = \{n_{20}, \ldots, n_{2i}\}$. For $i = 0, \ldots, r$, let $d_i = 7^{r-i} - 1$. We denote by r_1 the root of $\mathfrak{T}_1$ and by r_2 the root of $\mathfrak{T}_2$. We prove, by induction on i, that the Duplicator can answer the moves of the Spoiler in such a way that, for $i = 0, \ldots, r$, Condition (i) below is satisfied after the ith round:

> There exists a (b,d_i)-nbhd B_{1i} of N_{1i} in $\mathfrak{T}_1$ w.r.t. N_{1i}, a (b,d_i)-nbhd B_{2i} of N_{2i} in $\mathfrak{T}_2$ w.r.t. N_{2i}, and a bijection $f_i : B_{1i} \to B_{2i}$ mapping n_{10} to n_{20}, …, n_{1i} to n_{2i} which is a partial isomorphism between $\mathfrak{T}_1$ and $\mathfrak{T}_2$ satisfying, for each node p_1 of B_{1i}, $p_1 \approx_{b,d_i-\Delta_{\mathfrak{T}_1}(N_{1i},p_1)} f_i(p_1)$.

Since $B_{1r} = N_{1r}$ and $B_{2r} = N_{2r}$, Condition (r) expresses precisely that the Duplicator wins.

Induction basis: Let B_{10} be a (b,d)-nbhd of $n_{10} = n_1$ in $\mathfrak{T}_1$ w.r.t. $\{n_1\}$ and B_{20} a (b,d)-nbhd of $n_{20} = n_2$ in $\mathfrak{T}_2$ w.r.t. $\{n_2\}$. Since $n_1 \approx_{b,d} n_2$, we know, by Proposition 13, that there exists a bijection $f_0 : B_{10} \to B_{20}$ mapping n_1 to n_2 which is a partial isomorphism between $\mathfrak{T}_1$ and $\mathfrak{T}_2$ satisfying, for each node p_1 of B_{10}, $p_1 \approx_{b,d-\Delta_{\mathfrak{T}_1}(n_1,p_1)} f_0(p_1)$. Since $N_{10} = \{n_{10}\} = \{n_1\}$ and $d_0 = d$, Condition (0) is satisfied.

Induction step: Assume that, for some i, $0 < i \leq r$, Condition $(i-1)$ is satisfied after the $(i-1)$st round. Without loss of generality, the Spoiler chooses n_{1i} in $\mathfrak{T}_1$. We now explain how the Duplicator must respond to guarantee that Condition (i) is satisfied after the ith round.

Before going into detail, we first give a general outline of the induction step. There are two possibilities: the Spoiler chooses n_{1i} "close" to $N_{1(i-1)}$, or the Spoiler chooses n_{1i} "far" from $N_{1(i-1)}$. In the former case, we can modify $B_{1(i-1)}$ to ensure that it contains n_{1i}. The partial isomorphism f_{i-1} is then used to find an appropriate answer for the Duplicator which allows us to guarantee that Condition (i) is satisfied after the ith round. In the latter case, the Duplicator must choose n_{2i} "far" from $N_{2(i-1)}$. It is for this purpose that the perhaps somewhat awkward Counting Condition 2 of Definition 29 of bounded-equivalent trees was devised. Before explaining how we use it, we want to point out that it is not helpful to count the number of nodes that are (b, d_i)-bounded equivalent to n_{1i} in both $\mathfrak{T}_1$ and $\mathfrak{T}_2$ up to a certain bound and then requiring that these numbers are equal. Such a condition would only help us to establish that the Duplicator can choose n_{2i} outside some bounded neighborhood of $N_{2(i-1)}$, but *not* "far" from $N_{2(i-1)}$. (Indeed, at each distance from $N_{2(i-1)}$, nodes of $\mathfrak{T}_2$ may have been eliminated to create the bounded neighborhood.) It does help, however, to count *ancestors* at "sufficient distance" of nodes that are (b, d_i)-equivalent to n_{1i}, and compare these counts up to a certain bound (Condition 2 of Definition 29). Indeed, if a node is "close" to, say, n_{2j}, $0 \leq j < i$, then an ancestor at "sufficient distance" of that node is necessarily an ancestor of n_{2j}, and may belong to some bounded neighborhood of n_{2j}, even if the node itself does not. So, if the Duplicator can chose n_{2i} in such a way that it has a ancestor at "sufficient distance" in $\mathfrak{T}_2$ which, for all $j = 0, \ldots, i-1$, is not an ancestor of n_{2j} "too close" to n_{2j}, we can actually prove that n_{2i} is "far" from $N_{2(i-1)}$. There is one caveat, however. This reasoning will only work if the node n_{1i} has an ancestor at "sufficient distance" in $\mathfrak{T}_1$, which need not be the case if it is "too close" to the root of $\mathfrak{T}_1$. In that case, we use Proposition 30 to ensure that the Duplicator can chose a (b, d_i)-bounded equivalent node n_{2i} at the same distance of the root of $\mathfrak{T}_2$. Using the triangle inequality, it is then possible to prove that n_{2i} is "far" from $N_{2(i-1)}$. Finally, if n_{1i} is "far" from $N_{1(i-1)}$ and n_{2i} is "far" from $N_{2(i-1)}$, we know that, for $j = 1, 2$, any (b, d_i) neighborhood of n_{ji} in $\mathfrak{T}_j$ w.r.t. N_{ji} will be disconnected from any (b, d_i) neighborhood of $N_{j(i-1)}$ in $\mathfrak{T}_j$ w.r.t. N_{ji}. Proposition 26 then allows us to treat n_{1i} and n_{2i} separately from the previously selected nodes, more or less as in the Induction Basis, to guarantee that Condition (i) is satisfied after the ith round.

We now give the details. As explained above, we must distinguish between two cases. The threshold between "close" and "far" turns out to be $6d_i + 6$:

1) *The spoiler chooses n_{1i} such that $\Delta_{\mathfrak{T}_1}(N_{1(i-1)}, n_{1i}) \leq 6d_i + 6$ ("close").* In particular, $\Delta_{\mathfrak{T}_1}(N_{1(i-1)}, n_{1i}) \leq d_{i-1}$. Hence, if $B_{1(i-1)}$ is also a (b, d_{i-1})-nbhd of $N_{1(i-1)}$ w.r.t. to N_{1i} (instead of just $N_{1(i-1)}$), then $n_{1i} \in B_{1(i-1)}$. Proposition 20 allows us to modify $B_{1(i-1)}$ in such a way that this is indeed the case. The Duplicator now chooses $n_{2i} = f(n_{1i})$.[4] Corollary 28 also allows us to change $B_{2(i-1)}$ to a (b, d_{i-1})-nbhd of $N_{2(i-1)}$ w.r.t. to N_{2i} (instead of just $N_{2(i-1)}$). Now, $d_{i-1} \geq d_i$ and $d_{i-1} - (6d_i + 6) = d_i$. By Corollary 24, there exists a (b, d_i)-nbhd $B_{1i} \subseteq B_{1(i-1)}$ of N_{1i} in $\mathfrak{T}$ w.r.t. N_{1i}. Let $B_{2i} = f_{i-1}(B_{1i})$. Obviously, $f_i = f_{i-1}|_{B_{1i}}$ is a bijection from B_{1i} to B_{2i}, and it is a partial isomorphism between $\mathfrak{T}_1$ and $\mathfrak{T}_2$, because f_{i-1} is. Now, let p_1 be a node of B_{1i}. Then, $\Delta_{\mathfrak{T}_1}(N_{1i}, p_1) \leq d_i$. If $\Delta_{\mathfrak{T}_1}(N_{1i}, p_1) = \Delta_{\mathfrak{T}_1}(N_{1(i-1)}, p_1)$, then,

$$d_{i-1} - \Delta_{\mathfrak{T}_1}(N_{1(i-1)}, p_1) \geq d_i - \Delta_{\mathfrak{T}_1}(N_{1i}, p_1);$$

otherwise, $\Delta_{\mathfrak{T}_1}(N_{1i}, p_1) = \Delta_{\mathfrak{T}_1}(n_{1i}, p_1)$. By the triangle inequality, it also follows in this case that

$$\begin{aligned} d_{i-1} &- \Delta_{\mathfrak{T}_1}(N_{1(i-1)}, p_1) \\ &\geq d_{i-1} - \Delta_{\mathfrak{T}_1}(N_{1(i-1)}, n_{1i}) - \Delta_{\mathfrak{T}_1}(n_{1i}, p_1) \\ &\geq d_i - \Delta_{\mathfrak{T}_1}(N_{1i}, p_1). \end{aligned}$$

Condition $(i-1)$ and Proposition 6 then yield that $p_1 \approx_{b, d_i - \Delta_{\mathfrak{T}_1}(N_{1i}, p_1)} f_i(p_1)$. Finally, from this property and the fact that B_{1i} is a (b, d_i)-nbhd of N_{1i} in $\mathfrak{T}_1$ w.r.t. N_{1i}, it follows that B_{2i} is a (b, d_i)-nbhd of N_{2i} in $\mathfrak{T}_2$ w.r.t. N_{2i}. We may thus conclude that, in this case, Condition (i) is satisfied after the ith round.

2) *The spoiler chooses n_{1i} such that $\Delta_{\mathfrak{T}_1}(N_{1(i-1)}, n_{1i}) \geq 6d_i + 7$ ("far").* In particular, $\Delta_{\mathfrak{T}_1}(N_{1(i-1)}, n_{1i}) > 2d_i + 1$. Hence, by Proposition 25, any (b, d_i)-nbhd of $N_{1(i-1)}$ in $\mathfrak{T}_1$ w.r.t. N_{1i} is disconnected from any (b, d_i)-nbhd of n_{1i} in $\mathfrak{T}_1$ w.r.t. N_{1i}. In this case, the Duplicator aims to choose $n_{2i} \in \mathcal{N}_2$ such that

$$n_{1i} \approx_{b, d_i} n_{2i} \text{ and } \Delta_{\mathfrak{T}_1}(N_{2(i-1)}, n_{2i}) > 2d_i + 1. \quad (1)$$

If this is possible, then any (b, d_i)-nbhd of $N_{2(i-1)}$ in $\mathfrak{T}_2$ w.r.t. N_{2i} is also disconnected from any (b, d_i)-nbhd of n_{2i} in $\mathfrak{T}_2$ w.r.t. N_{2i}. As explained in our proof strategy, we have to distinguish between the case that n_{1i} has an ancestor at "sufficient distance" in $\mathfrak{T}_1$ and the case that n_{1i} has no such ancestor to show that Aim (1) above can be achieved. Here, "sufficient distance" is $2d_i + 1$. We start with the latter case:

   a) $\Delta_{\mathfrak{T}_1}(r_1, n_{1i}) \leq 2d_i$. Since both $2d_i \leq d_{i-1}$ and $d_{i-1} - 2d_i \geq d_i$, it follows from Propositions 30 and 6 that the Duplicatior can choose $n_{2i} \in \mathcal{N}_2$ such that $\Delta_{\mathfrak{T}_1}(r_1, n_{1i}) = \Delta_{\mathfrak{T}_2}(r_2, n_{2i})$ and $n_{1i} \approx_{b, d_i} n_{2i}$. Now, let $0 \leq j < i$. By the triangle inequality,

$$\begin{aligned} \Delta_{\mathfrak{T}_1}(r_1, n_{1j}) &\geq \Delta_{\mathfrak{T}_1}(n_{1j}, n_{1i}) - \Delta_{\mathfrak{T}_1}(r_1, n_{1i}) \\ &\geq (6d_i + 7) - 2d_i = 4d_i + 7. \end{aligned}$$

   Also, $d_{i-1} \geq 4d_i + 7$. By Condition $(i-1)$, $n_{1j} \approx_{b, d_{i-1}} n_{2j}$. Hence, $\min(\Delta_{\mathfrak{T}_1}(r_1, n_{1j}), d_{i-1}) =$

[4] In particular, if the Spoiler chooses $n_{1i} = n_{1j}$, $0 \leq j < i$, then the Duplicator chooses $n_{2i} = n_{2j}$.

$\min(\Delta_{\mathfrak{T}_2}(r_2, n_{2j}), d_{i-1})$. We then have, using the triangle inequalty, that

$$\begin{aligned}\Delta_{\mathfrak{T}_2}(n_{2j}, n_{2i}) &\geq \Delta_{\mathfrak{T}_2}(r_2, n_{2j}) - \Delta_{\mathfrak{T}_2}(r_2, n_{2i}) \\ &\geq (4d_i + 7) - 2d_i > 2d_i + 1,\end{aligned}$$

and, hence, $\Delta_{\mathfrak{T}_2}(N_{2(i-1)}, n_{2i}) > 2d_i + 1$, as was required to realize Aim (1).

b) $\Delta_{\mathfrak{T}_1}(r_1, n_{1i}) \geq 2d_i + 1$. Hence, n_{1i} has an ancestor, say p_1, at distance $2d_i + 1$ of n_{1i}. We define the following sets, for $j = 1, 2$:

$$\begin{aligned}Q_j = \{q_j \in \mathcal{N}_j \mid\ & q_j \text{ is an ancestor at distance at} \\ & \text{most } 4d_i + 2 \text{ of a node in } N_{j(i-1)}, \text{ and there} \\ & \text{exists } m_j \in \mathcal{N}_j \text{ such that } q_j \text{ is an ancestor at} \\ & \text{distance } 2d_i + 1 \text{ of } m_j \text{ and } n_{1i} \approx_{b,d_i} m_j\}.\end{aligned}$$

Now suppose, for the sake of argument, that the Duplicator can choose n_{2i} in $\mathfrak{T}_2$ such that $n_{1i} \approx_{b,d_i} n_{2i}$, and $\Delta_{\mathfrak{T}_2}(r_2, n_{2i}) \geq 2d_i + 1$. Hence, n_{2i} has an ancestor, say p_2, at distance $2d_i + 1$ of n_{2i}. Suppose furthermore that the Duplicator can ensure that

$$p_2 \notin Q_2. \tag{2}$$

If, for some j, $0 \leq j < i$, $\Delta_{\mathfrak{T}_2}(n_{2j}, n_{2i}) \leq 2d_i + 1$, then, p_2 must also be an ancestor of n_{2j} and, by the triangle inequality, $\Delta_{\mathfrak{T}_2}(p_2, n_{2j}) \leq 4d_i + 2$, contradicting $p_2 \notin Q_2$. Hence, $\Delta_{\mathfrak{T}_2}(N_{2(i-1)}, n_{2i}) > 2d_i + 1$, realizing Aim (1). We are now going to establish properties of Q_1 and Q_2 from which we can derive that such a choice for n_{2i} is possible.

Property 1: $p_1 \notin Q_1$. Indeed, by the triangle inequality, for all $j = 0, \ldots, i-1$,

$$\begin{aligned}\Delta_{\mathfrak{T}_1}(p_1, n_{1j}) &\geq \Delta_{\mathfrak{T}_1}(n_{1j}, n_{1i}) - \Delta_{\mathfrak{T}_1}(p_1, n_{1i}) \\ &\geq (6d_i + 7) - (2d_i + 1) > 4d_i + 2.\end{aligned}$$

Property 2: $Q_1 \subseteq B_{1(i-1)}$ and $Q_2 \subseteq B_{2(i-1)}$. Let $q_1 \in Q_1$. Hence, there exists j, $0 \leq j < i$, such that q_1 is an ancestor of n_{1j} at distance at most $4d_i + 2$. Since $4d_i + 2 \leq d_{i-1}$, $B_{1(i-1)}$ must contain q_1, by Condition 2 of Proposition 12. Analogously, $Q_2 \subseteq B_{2(i-1)}$.

Property 3: $f_{i-1}(Q_1) = Q_2$. By symmetry, it suffices to prove that $q_1 \in Q_1$ implies $f_{i-1}(q_1) \in Q_2$. Thus, let $q_1 \in Q_1$. Hence, there exists j, $0 \leq j < i$, such that q_1 is an ancestor of n_{1j} at distance at most $4d_i + 2$. Moreover q_1 is also an ancestor of a node, say m_1, such that $\Delta_{\mathfrak{T}_1}(q_1, m_1) = 2d_i + 1$ and $n_{1i} \approx_{b,d_i} m_1$. By Proposition 15, there exists a (b, d_{i-1})-nbhd $B'_{1j} \subseteq B_{1j}$ of n_{1j} in $\mathfrak{T}_1$ w.r.t. N_{1i}. By the triangle inequalty,

$$\begin{aligned}\Delta_{\mathfrak{T}_1}(n_{1j}, m_1) &\leq \Delta_{\mathfrak{T}_1}(q_1, n_{1j}) + \Delta_{\mathfrak{T}_1}(q_1, m_1) \\ &\leq (4d_i + 2) + (2d_i + 1) \leq 6d_i + 6.\end{aligned}$$

In Case 1, we argued that $d_{i-1} - (6d_i + 6) = d_i \geq 0$. By Statement 4 of Proposition 12, B'_{1j} (and, hence, B_{1j}) contains a node m'_1 such that n_{1j} and m'_1 have the same closest common ancestor as n_{1j} and m_1, $\Delta_{\mathfrak{T}_1}(n_{1j}, m_1) = \Delta_{\mathfrak{T}_1}(n_{1j}, m'_1)$, and $m_1 \approx_{b,d_i} m'_1$. Hence, also $n_{1i} \approx_{b,d_i} m'_1$. Since q_1 is a common ancestor of n_{1j} and m_1, it follows that q_1 is an ancestor of m'_1 and that $\Delta_{\mathfrak{T}_1}(q_1, m'_1) = 2d_i + 1$. Now, consider $p_2 = f_{i-1}(p_1)$ and $m'_2 = f_{i-1}(m'_1)$. Since

$$\begin{aligned}d_{i-1} &- \Delta_{\mathfrak{T}_1}(N_{1(i-1)}, m'_1) \\ &\geq d_{i-1} - \Delta_{\mathfrak{T}_1}(n_{1j}, m'_1) \geq d_i,\end{aligned}$$

Condition $(i-1)$ and Proposition 6 yield $m'_1 \approx_{b,d_i} m'_2$, and, hence, also $n_{1i} \approx_{b,d_i} m'_2$. Since f_{i-1} is a partial tree isomorphism, q_2 is an ancestor of n_{2j} at distance at most $4d_i + 2$, and also an ancestor of m'_2 at distance $2d_i + 1$. Hence, $q_2 \in Q_2$.

Property 4: $|Q_1| = |Q_2| \leq 4d + 3$. By Properties 2 and 3, $|Q_1| = |Q_2|$. Now, $|N_{1(i-1)}| \leq i$ and each node in $N_{1(i-1)}$ has at most $4d_i + 3$ ancestors at distance at most $4d_i + 2$. Hence, $|Q_1| \leq i(4d_i + 3) \leq 4d + 3$. To obtain this bound, we used that, for all $i > 0$, $i.7^{r-i} \leq 7^r$ and $i \geq 1$.

From comparing Property 4 with Condition 2 of Definition 29 of bounded-equivalent trees, and from Property 1, it follows that there exists $n_{2i} \in \mathcal{N}_2$ such that $n_{1i} \approx_{b,d_i} n_{2i}$, n_{2i} has an ancestor at distance $2d_i + 1$, say p_2, and $p_2 \notin Q_2$. Thus, the Duplicator managed to realize Aim (2), and, hence, also Aim (1).

From the above analysis, we conclude that, in Case 2, the Duplicator can always realize Aim (1).

By reasoning as in Case 1, we can establish the existence of a (b, d_i)-nbhd B_{1i}^{old} of $N_{1(i-1)}$ in $\mathfrak{T}_1$ w.r.t. N_{1i}, a (b, d_i)-nbhd B_{2i}^{old} of $N_{2(i-1)}$ in $\mathfrak{T}_2$ w.r.t. N_{2i}, and a bijection $f_i^{\text{old}} : B_{1i}^{\text{old}} \to B_{2i}^{\text{old}}$ mapping n_{10} to n_{20}, $\ldots$, $n_{1(i-1)}$ to $n_{2(i-1)}$ which is a partial tree automorphism satisfying, for each node p_1 of B_{1i}^{old}, $p_1 \approx_{b,d_i - \Delta_{\mathfrak{T}_1}(N_{1(i-1)}, p_1)} f_i^{\text{old}}(p_1)$. By reasoning as for the Induction Basis, we can establish the existence of a (b, d_i)-nbhd B_{1i}^{new} of n_{1i} in $\mathfrak{T}_1$ with respect to N_{1i}, a (b, d_i)-nbhd B_{2i}^{new} of n_{2i} in $\mathfrak{T}_2$ with respect to N_{2i}, and a bijection $f_i^{\text{new}} : B_{1i}^{\text{new}} \to B_{2i}^{\text{new}}$ mapping n_{1i} to n_{2i} which is a partial tree automorphism satisfying, for each node p_1 of B_{1i}^{new}, $p_1 \approx_{b,d_i - \Delta_{\mathfrak{T}_1}(n_{1i}, p_1)} f_i^{\text{new}}(p_1)$. Since $\Delta_{\mathfrak{T}_1}(N_{1(i-1)}, n_{1i}) > 2d_i + 1$ and $\Delta_{\mathfrak{T}_2}(N_{2(i-1)}, n_{2i}) > 2d_i + 1$, B_{1i}^{new} is disconnected from B_{1i}^{old} and B_{2i}^{new} is disconnected from B_{2i}^{old}. By Proposition 26, $B_{1i} := B_{1i}^{\text{old}} \cup B_{1i}^{\text{new}}$ is a (b, d_i)-nbhd of N_{1i} w.r.t. N_{1i} and $B_{2i} := B_{2i}^{\text{old}} \cup B_{2i}^{\text{new}}$ is a (b, d_i)-nbhd of N_{2i} w.r.t. N_{2i}. Now, for $p_1 \in B_{1i}^{\text{old}}$, $\Delta_{\mathfrak{T}_1}(N_{1i}, p_1) = \Delta_{\mathfrak{T}_1}(N_{1(i-1)}, p_1)$; for $p_1 \in B_{1i}^{\text{new}}$, $\Delta_{\mathfrak{T}_1}(N_{1i}, p_1) = \Delta_{\mathfrak{T}_1}(n_{1i}, p_1)$. Let $f_i := f_i^{\text{old}} \cup f_i^{\text{new}}$. Then f_i is a bijecton between B_{1i}

and B_{2i} mapping n_{10} to n_{20}, …, n_{1i} to n_{2i} which is a partial tree automorphism satisfying, for each node p_1 of B_{1i}, $p_1 \approx_{b,d_i-\Delta_{\mathfrak{T}_1}(N_{1i},p_1)} f_i(p_1)$. Hence, also in this case, Condition (i) is satisfied afther the ith round.

In summary, we have shown, assuming that Condition $(i-1)$ is satisfied after the $(i-1)$st round of the game, that the Duplicator can always answer the move of the Spoiler in the ith round in such a way that Condition (i) is always satisfied after the ith round, completing the inductive argument.

We have already observed that Condition (r) at the end of the game described above implies the Duplicator won. Since we made no assumptions on the moves of the Spoiler, we may conclude that the Duplicator has a winning strategy. ∎

Corollary 33. *Let $\mathfrak{T}_1 = (\mathcal{N}_1, \lambda_1, \mathcal{E}_1)$, $\mathfrak{T}_2 = (\mathcal{N}_2, \lambda_2, \mathcal{E}_2)$, $n_1 \in \mathcal{N}_1$, and $n_2 \in \mathcal{N}_2$. Let φ be a unary first-order query with $\mathsf{qr}(\varphi) = r$. If $(\mathfrak{T}_1, n_1) \approx^{\mathrm{r}} (\mathfrak{T}_2, n_2)$, then $n_1 \in [\![\varphi]\!]_{\mathfrak{T}_1}$ if and only if $n_2 \in [\![\varphi]\!]_{\mathfrak{T}_2}$.*

VIII. Main Result

As explained in Section II, we must show that, given $r \geq 0$, the equivalence classes for $\approx^{\mathrm{r}}$ can be described by $\mathrm{FO}^2 + \mathrm{C}$ formulae of which the quantifier rank and counting are bounded by functions of r. This is done in Theorem 37, based on the results in Lemmas 34 and 35.[5]

Lemma 34. 1) *Let $d \geq 0$. There exists a unary FO^2 query $\mathsf{ran}_d(x)$ of quantifier rank $d+1$ such that, for $\mathfrak{T} = (\mathcal{N}, \lambda, \mathcal{E})$ and for $n \in \mathcal{N}$, $n \in [\![\mathsf{ran}_d(x)]\!]_{\mathfrak{T}}$ if and only if n is at distance d from the root of $\mathfrak{T}$.*

2) *Let φ be a unary $\mathrm{FO}^2 + \mathrm{C}$ query. There exists a unary $\mathrm{FO}^2 + \mathrm{C}$ query $\mathsf{dsc}_{d,\varphi}(x)$ of quantifier rank $\mathsf{qr}(\varphi) + d$ such that, for $\mathfrak{T} = (\mathcal{N}, \lambda, \mathcal{E})$ and for $n \in \mathcal{N}$, $n \in [\![\mathsf{dsc}_{d,\varphi}(x)]\!]_{\mathfrak{T}}$ if and only if n has a descendant m at distance d from n with m in $[\![\varphi]\!]_{\mathfrak{T}}$.*

Proof:

1) Clearly, $\mathsf{ran}_0(x) := \neg\exists y\, E(y,x)$; if $d > 0$, then $\mathsf{ran}_d(x) := \exists y\,(E(y,x) \wedge \mathsf{ran}_{d-1}(y))$.
2) Clearly, $\mathsf{dsc}_{0,\varphi}(x) := \varphi(x)$; if $d > 0$, then $\mathsf{dsc}_{d,\varphi}(x) := \exists y\,(E(x,y) \wedge \mathsf{dsc}_{d-1,\varphi}(y))$. ∎

Let $\mathcal{R} \subseteq \mathcal{P}$ be a *finite* set of label predicates. We say that $\mathfrak{T} = (\mathcal{N}, \lambda, \mathcal{E})$ is an $\mathcal{R}$-tree if, for all $\ell \in \mathcal{P} \setminus \mathcal{R}$, $\lambda(\ell) = \emptyset$.

Lemma 35. 1) *Let $\mathfrak{T} = (\mathcal{N}, \lambda, \mathcal{E})$ be an $\mathcal{R}$-tree, and let $n \in \mathcal{N}$. There exists a unary quantifier-free FO^2 query $\mathsf{lab}_{\mathfrak{T},n}(x)$ such that, for each $\mathcal{R}$-tree $\mathfrak{T}' = (\mathcal{N}', \lambda', \mathcal{E}')$ and for each $n' \in \mathcal{N}'$, $n' \in [\![\mathsf{lab}_{\mathfrak{T},n}(x)]\!]_{\mathfrak{T}'}$ if and only if, for all $\ell \in \mathcal{P}$, $n \in \lambda(\ell)$ if and only if $n' \in \lambda'(\ell)$.*

2) *Let $\mathfrak{T} = (\mathcal{N}, \lambda, \mathcal{E})$ be an $\mathcal{R}$-tree, let $n \in \mathcal{N}$, and let $b \geq 2$ and $d \geq 0$. There exists a unary $\mathrm{FO}^2 + \mathrm{C}$ query $\mathsf{dbe}_{\mathfrak{T},n,b,d}(x)$ of quantifer rank d and counting bounded by b such that, for each $\mathcal{R}$-tree $\mathfrak{T}' = (\mathcal{N}', \lambda', \mathcal{E}')$ and for each $n' \in \mathcal{N}'$, $n' \in [\![\mathsf{dbe}_{\mathfrak{T},n,b,d}(x)]\!]_{\mathfrak{T}'}$ if and only if $n \approx_{\downarrow b,d} n'$.*

3) *Let $\mathfrak{T} = (\mathcal{N}, \lambda, \mathcal{E})$ be an $\mathcal{R}$-tree with root r, let $n \in \mathcal{N}$, and let $b \geq 2$ and $d \geq 0$. There exists a unary $\mathrm{FO}^2 + \mathrm{C}$ query $\mathsf{beq}_{\mathfrak{T},n,b,d}(x)$ of quantifer rank d and counting bounded by b such that, for each $\mathcal{R}$-tree $\mathfrak{T}' = (\mathcal{N}', \lambda', \mathcal{E}')$ and for each node $n' \in \mathcal{N}'$, $n' \in [\![\mathsf{beq}_{\mathfrak{T},n,b,d}(x)]\!]_{\mathfrak{T}'}$ if and only if $n \approx_{b,d} n'$.*

4) *Let $\mathfrak{T} = (\mathcal{N}, \lambda, \mathcal{E})$ be an $\mathcal{R}$-tree, and let $b \geq 2$ and $d, k \geq 0$. There exists a boolean $\mathrm{FO}^2 + \mathrm{C}$ query $\mathsf{bte}_{\mathfrak{T},b,d,k}$ of quantifer rank $3d+2$ and counting bounded by $\max(b,k)$ such that, for each $\mathcal{R}$-tree $\mathfrak{T}' = (\mathcal{N}', \lambda', \mathcal{E}')$, $[\![\mathsf{bte}_{\mathfrak{T},b,d,k}]\!]_{\mathfrak{T}'} = \mathtt{true}$ if and only if $\mathfrak{T} \approx_{b,d,k} \mathfrak{T}'$.*

Proof:

1) Let $T = \{\ell \in \mathcal{R} \mid n \in [\![\ell]\!]_{\mathfrak{T}}\}$ and $F = \mathcal{R} \setminus T$. Then, $\mathsf{lab}_{\mathfrak{T},n}(x) := \big(\bigwedge_{\ell \in T} \ell(x)\big) \wedge \big(\bigwedge_{\ell \in F} \neg\ell(x)\big)$.
2) We construct $\mathsf{dbe}_{\mathfrak{T},n,b,d}(x)$ recursively on d. We put $\mathsf{dbe}_{\mathfrak{T},n,b,0}(x) := \mathsf{lab}_{\mathfrak{T},n}(x)$. Let $d > 0$. For $m \in \mathcal{N}$, we introduce the following shorthand:

$$\exists^{c_1(m)} := \begin{cases} \exists^{=|[m]_{\downarrow b,d-1}|} & \text{if } |[m]_{\downarrow b,d-1}| < b; \\ \exists^{\geq b} & \text{otherwise.} \end{cases}$$

Then,

$$\mathsf{dbe}_{\mathfrak{T},n,b,d}(x) := \mathsf{lab}_{\mathfrak{T},n}(x) \wedge \mathsf{dri}_{\mathfrak{T},n,b,d}(x) \wedge \mathsf{dle}_{\mathfrak{T},n,b,d}(x),$$

where, in the right-hand side, $\mathsf{dri}_{\mathfrak{T},n,b,d}(x)$ is short for

$$\bigwedge_{(n,m)\in\mathcal{E}} \Big(\exists^{c_1(m)} y\,(E(x,y) \wedge \mathsf{dbe}_{\mathfrak{T},m,b,d-1}(y))\Big)$$

and $\mathsf{dle}_{\mathfrak{T},n,b,d}(x)$ is short for

$$\forall y \left(\bigvee_{(n,m)\in\mathcal{E}} (E(x,y) \Rightarrow \mathsf{dbe}_{\mathfrak{T},m,b,d-1}(y))\right).$$

Now, $\mathsf{dri}_{\mathfrak{T},n,b,d}(x)$ expresses that, for each child m of n, there exists a child m' of the node of $\mathfrak{T}'$ represented by x such that $\min(|[m]_{\downarrow b,d-1}|, b) = \min(|[m']_{\downarrow b,d-1}|, b)$. The expression $\mathsf{dle}_{\mathfrak{T},n,b,d}(x)$ expresses that, for each child m' of the node of $\mathfrak{T}'$ represented by x, there exists a child m of n such that $m \approx_{\downarrow b,d-1} m'$. Hence, $\mathsf{dri}_{\mathfrak{T},n,b,d}(x) \wedge \mathsf{dle}_{\mathfrak{T},n,b,d}(x)$ implies that, for each child m' of the node of $\mathfrak{T}'$ represented by x, there exists a child m of n such that $\min(|[m]_{\downarrow b,d-1}|, b) = \min(|[m']_{\downarrow b,d-1}|, b)$.
3) We construct $\mathsf{beq}_{\mathfrak{T},n,b,d}(x)$ recursively on d. We put $\mathsf{beq}_{\mathfrak{T},n,b,0}(x) := \mathsf{dbe}_{\mathfrak{T},n,b,0}(x)$. Let $d > 0$. If $n = r$, then $\mathsf{beq}_{\mathfrak{T},n,b,d}(x) := \mathsf{dbe}_{\mathfrak{T},n,b,d}(x) \wedge \neg(\exists y\, E(y,x))$. If $n \neq r$ and m is the parent of n, then $\mathsf{beq}_{\mathfrak{T},n,b,d}(x) := \mathsf{dbe}_{\mathfrak{T},n,b,d}(x) \wedge \exists y\,\big(E(y,x) \wedge \mathsf{beq}_{\mathfrak{T},m,b,d-1}(y)\big)$.
4) We put $\mathsf{bte}_{\mathfrak{T},b,d,k} := \mathsf{te1}_{\mathfrak{T},b,d} \wedge \mathsf{te2}_{\mathfrak{T},b,d,k}$, where $\mathsf{te1}_{\mathfrak{T},b,d}$ expresses Condition 1 of Definition 29, and,

[5] In the proofs of Lemmas 34 and 35 and Theorem 37, the variables used in the $\mathrm{FO}^2 + \mathrm{C}$ formulae we construct, usually recursively, are x and y. If, in the process, we define a unary $\mathrm{FO}^2 + \mathrm{C}$ formula $\psi(x)$, then, whenever we write $\psi(y)$ in a recursion step, we mean $\psi(x)$ in which the variables x and y have been swapped.

in conjunction with $\mathsf{te1}_{\mathfrak{T},b,d}$, $\mathsf{te2}_{\mathfrak{T},b,d,k}$ expresses Condition 2. First, $\mathsf{te1}_{\mathfrak{T},b,d} :=$

$$\left(\bigwedge_{n\in\mathcal{N}} \left(\exists x\ \mathsf{beq}_{\mathfrak{T},n,b,d}(x)\right)\right) \wedge \forall x \left(\bigvee_{n\in\mathcal{N}} \mathsf{beq}_{\mathfrak{T},n,b,d}(x)\right),$$

which expresses that, for each node n in $\mathfrak{T}$, there exists a node n' in $\mathfrak{T}'$ such that $n \approx_{b,d} n'$, and vice versa. Second,

$$\mathsf{te3}_{\mathfrak{T},b,d,k} = \bigwedge_{d'=0}^{d} \bigwedge_{n\in\mathcal{N}} \mathsf{t3s}_{\mathfrak{T},n,b,d',k},$$

in which $\mathsf{t3s}_{\mathfrak{T},n,b,d',k}$ expresses that the minimum of k and the number of ancestors at distance $2d'+1$ of nodes of $\mathfrak{T}'$ that are (b,d')-bounded equivalent to n equals the minimum of k and the number of ancestors at distance $2d'+1$ of nodes of $\mathfrak{T}$ that are (b,d')-bounded equivalent to n. So, for $n \in \mathcal{N}$, let $K_{d'}(n)$ be the set of ancestors at distance $2d'+1$ of nodes in $\mathfrak{T}$ that are (b,d')-bounded equivalent to n. We introduce the following shorthand:

$$\exists^{c_2(n)} := \begin{cases} \exists^{=|K_{d'}(n)|} & \text{if } |K_{d'}(n)| < k; \\ \exists^{\geq k} & \text{otherwise.} \end{cases}$$

Then, $\mathsf{t3s}_{\mathfrak{T},n,b,d',k} := \exists^{c_2(n)} x\ \mathsf{dsc}_{2d'+1,\varphi}(x)$, where $\varphi(y) := \mathsf{beq}_{\mathfrak{T},n,b,d'}(y)$. ∎

Let φ be an FO query, and let $\mathcal{P}_\varphi \subseteq \mathcal{P}$ be the set of all label predicates that occur in φ. Obviously, $\mathcal{P}_\varphi$ is finite. Let $\mathfrak{T} = (\mathcal{N}, \lambda, \mathcal{E})$. The φ-restriction of $\mathfrak{T}$ is the tree $\mathfrak{T}_\varphi = (\mathcal{N}, \lambda_\varphi, \mathcal{E})$ where, for all $\ell \in \mathcal{P}$,

$$\lambda_\varphi(\ell) = \begin{cases} \lambda(\ell) & \text{if } \ell \in \mathcal{P}_\varphi; \\ \emptyset & \text{otherwise.} \end{cases}$$

The following is now immediate:

Lemma 36. *Let φ be an* FO *query, and let $\mathfrak{T} = (\mathcal{N}, \lambda, \mathcal{E})$. Let $\mathfrak{T}_\varphi$ be the φ-restriction of $\mathfrak{T}$. Then, $[\![\varphi]\!]_{\mathfrak{T}} = [\![\varphi]\!]_{\mathfrak{T}_\varphi}$.*

So, without loss of generality, we may assume that only a finite number of label predicates are at play.

We are now finally ready to prove our main result:

Theorem 37. *Let φ be a unary* FO *query. There exists an* $\mathrm{FO}^2 + \mathrm{C}$ *query ψ that is equivalent to φ on trees.*

Proof: By Lemma 36, we may restrict our attention to $\mathcal{P}_\varphi$-trees, with $\mathcal{P}_\varphi \subseteq \mathcal{P}$ the label predicates actually occurring in φ.

Let $r = \mathsf{qr}(e)$, $d = 7^r - 1$, $b = r + 2$, and $k = 4d + 4$. For a $\mathcal{P}_\varphi$-tree $\mathfrak{T} = (\mathcal{N}, \lambda, \mathcal{E})$ and $n \in \mathcal{N}$, we define the predicate

$$\mathsf{squery}_{\mathfrak{T},n}(x) := \mathsf{bte}_{\mathfrak{T},b,d,k} \wedge \mathsf{beq}_{\mathfrak{T},n,b,d}(x),$$

with $\mathsf{beq}_{\mathfrak{T},n,b,d}(x)$ and $\mathsf{bte}_{\mathfrak{T},b,d,k}$ the queries introduced in Lemma 35 to expresses (b,d)-bounded equivalence of tree nodes, respectively (b,d,k)-bounded equivalence of trees. By Lemma 35, $\mathsf{squery}_{\mathfrak{T},n}(x)$ is a unary $\mathrm{FO}^2 + \mathrm{C}$ query of quantifier rank $3d + 2$ and counting bounded by $\max(b,k)$. Since we only consider the finitely many label predicates occurring in r, there are only finitely many such formulae up to equivalence. For a $\mathcal{P}_\varphi$-tree $\mathfrak{T}' = (\mathcal{N}', \lambda', \mathcal{E}')$ and $n' \in \mathcal{N}'$, $n' \in [\![\mathsf{squery}_{\mathfrak{T},n}(x)]\!]_{\mathfrak{T}'}$ if and only if $(\mathfrak{T}, n) \approx^{\mathrm{r}} (\mathfrak{T}', n')$. Hence, by Corollary 33, if $n' \in [\![\mathsf{squery}_{\mathfrak{T},n}(x)]\!]_{\mathfrak{T}'}$, then $n' \in [\![\varphi]\!]_{\mathfrak{T}'}$ if and only if $n \in [\![\varphi]\!]_{\mathfrak{T}}$.

Now, let $\mathcal{T}$ be the set of all $\mathcal{P}_\varphi$-trees (over a suitable universe of nodes). We define[6]

$$\mathsf{query}(x) := \bigvee_{\mathfrak{T}\in\mathcal{T}} \bigvee_{n\in[\![\varphi]\!]_{\mathfrak{T}}} \mathsf{squery}_{\mathfrak{T},n}(x).$$

While the above disjunction is infinite, it contains only finitely many disjuncts up to equivalence. Hence, after elimination of duplicates, $\mathsf{query}(x)$ is a unary $\mathrm{FO}^2 + \mathrm{C}$ query of quantifier rank $3d + 2$ and counting bounded by $\max(b,k)$. From the properties of the disjuncts, it follows that, for a $\mathcal{P}_\varphi$-tree $\mathfrak{T}' = (\mathcal{N}', \lambda', \mathcal{E}')$, $[\![\mathsf{query}(x)]\!]_{\mathfrak{T}'} \subseteq [\![\varphi]\!]_{\mathfrak{T}'}$. Conversely, assume that, for a $\mathcal{P}_\varphi$-tree $\mathfrak{T}' = (\mathcal{N}', \lambda', \mathcal{E}')$ and $n' \in \mathcal{N}'$, $n' \in [\![\varphi']\!]_{\mathfrak{T}'}$. By construction, $n' \in [\![\mathsf{squery}_{\mathfrak{T}',n'}(x)]\!]_{\mathfrak{T}'} \subseteq [\![\mathsf{query}(x)]\!]_{\mathfrak{T}'}$. Thus, $[\![\mathsf{query}(x)]\!]_{\mathfrak{T}'} = [\![\varphi]\!]_{\mathfrak{T}'}$. ∎

Corollary 38. *Let φ be a boolean* FO *query. There exists an* $\mathrm{FO}^2 + \mathrm{C}$ *query which is equivalent to φ on trees.*

Proof: Let $\varphi'(x) := (x = x) \wedge \varphi$. By Theorem 37, there is a unary $\mathrm{FO}^2 + \mathrm{C}$ query $\psi(x)$ which is equivalent to $\varphi'(x)$ on trees. Then, clearly, $\exists x\, \psi(x)$ is a boolean $\mathrm{FO}^2 + \mathrm{C}$ query which is equivalent to φ on trees. ∎

Finally, our main result also has ramifications for FO queries of arbitrary arity:

Corollary 39. *Let $k \geq 1$ and φ be an* FO *query of arity k. There exists a query in* $\mathrm{FO}^{k+1} + \mathrm{C}$ *which is equivalent to φ on trees.*

Proof: We prove this by induction on k. The induction basis, $k = 1$, is provided by Theorem 37. For the induction step, assume as induction hypothesis that, for some $k > 1$, every FO query of arity $k-1$ is equivalent to an FO^k+C query. Let $\varphi(x_1, \ldots, x_k)$ be an FO query of arity k. Let $\ell \in \mathcal{P}$ be a label predicate not occurring in φ, and consider the $(k-1)$-ary $\mathrm{FO} + \mathrm{C}$ formula $\varphi'(x_1, \ldots, x_{k-1}) :=$

$$\exists^{=1} x_k\, \ell(x_k) \wedge \forall x_k\, (\ell(x_k) \Rightarrow \varphi(x_1, \ldots, x_k)).$$

By the induction hypothesis, there is a $(k-1)$-ary $\mathrm{FO}^k + \mathrm{C}$ query ψ' equivalent to φ' on trees. Let $\psi(x_1, \ldots, x_k)$ be the k-ary $\mathrm{FO}^{k+1} + \mathrm{C}$ query obtained from $\psi'(x_1, \ldots, x_{k-1})$ by substituting each subformula $\ell(x_i)$, $1 \leq i \leq k-1$, by $x_i = x_k$. (We assume that x_k does *not* occur in $\psi'(x_1, \ldots, x_{k-1})$.) Clearly, ψ is equivalent to φ. ∎

Obviously, our main motivation to include label predicates is that, without them, trees would not be very helpful as data structures. Corollary 39 shows that their presence may also be important at a more fundamental level.

[6] An empty disjunction is interpreted as `false`.

IX. Conclusion and Future Work

Here, we proved that unary and boolean FO queries on finite, rooted, unranked, unordered, node-labeled trees can be expressed in $\mathrm{FO}^2 + \mathrm{C}$. As a corollary, k-ary FO queries on trees, $k \geq 2$, can be expressed in $\mathrm{FO}^{k+1} + \mathrm{C}$. To achieve this result, we developed a toolkit to exploit locally bounded equivalences. This toolkit may also be useful to study equivalences between other query languages (cf. the work of Gyssens et al. [12] and Fletcher et al. [13]). Of course, there are several aspects related to our work which were not covered here. For example, what is the complexity of our translation from FO to $\mathrm{FO}^2 + \mathrm{C}$? Can such a translation help in solving satisfiability questions? These are obviously topics for future research. Other questions raised by this work are the following:

- It is well known that FO can be algebratized to the relational algebra. This algebratization had an enormous impact on development of database management systems: queries are first translated to relational algebra expressions, then optimized using rewrite rules, and finally evaluated using specialized data structures and algorithms. It is also known that, for unary graph queries, FO^2 is equivalent with the semi-join algebra, and that, for binary graph queries, FO^3 is equivalent with the calculus of relations (Tarski's Relation Algebra). Of course, these algebratizations also apply to tree models. It would therefore be of interest to find an algebratization for $\mathrm{FO}^2 + \mathrm{C}$. Actually, we suggest that, on unranked, unordered trees, this could be the Core Path + counting projections language introduced by Fletcher et al. [13]. As for optimization, we suggest translations that introduce semi-joins rather than relational compositions, as studied by Hellings et al. [21]. Indeed, semi-joins and counting are very efficiently supported in database systems.
- A natural direction for future research is to investigate (or even characterize) on which other classes of structures (beyond the trees we considered here) FO collapses to $\mathrm{FO}^2 + \mathrm{C}$.
- Another direction for future research is a finer analysis of the relationship between various finite-variable logics. For example, it follows from our paper that unary FO^3 queries on trees can be translated to unary $\mathrm{FO}^2 + \mathrm{C}$ queries. It would be of interest to determine to which extent counting quantifiers could be limited in the translation. For example, would quantifiers which can only count up to, say 3, suffice? Similar questions could be asked for unary FO^k queries, $k > 3$. Of course, one must also consider the relationship between the sizes of the original query and the corresponding $\mathrm{FO}^2 + \mathrm{C}$ query.
- To bootstrap our main result on the translation of unary FO tree queries to $\mathrm{FO}^2 + \mathrm{C}$ to the translation of k-ary FO queries to $\mathrm{FO}^{k+1} + \mathrm{C}$, we took advantage of the presence of label predicates. When we set up the context in which we studied the research question solved in this paper, we included label predicates from a purely applicational point of view: without them, trees would be rather useless as a means to store data. It was therefore surprising for us to see that they also play a more fundamental role. It would therefore be interesting to see if the bootstrapping can also be realized without resorting to label predicates.

Acknowledgments

We thank the reviewers for their constructive comments which helped us to improve the paper.

References

[1] P. Barceló and L. Libkin, "Temporal logics over unranked trees," in *20th IEEE Symposium on Logic in Computer Science (LICS 2005), 26-29 June 2005, Chicago, IL, USA, Proceedings*. IEEE Computer Society, 2005, pp. 31–40.

[2] M. Bojańczyk, "Effective characterizations of tree logics," in *Proceedings of the 27th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2008, June 9-11, 2008, Vancouver, BC, Canada*, M. Lenzerini and D. Lembo, Eds. ACM, 2008, pp. 53–66.

[3] L. Libkin, "Logics for unranked trees: An overview," *Log. Methods Comput. Sci.*, vol. 2, no. 3, 2006.

[4] ——, *Elements of Finite Model Theory*, ser. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2004.

[5] N. Immerman and D. Kozen, "Definability with bounded number of bound variables," *Inf. Comput.*, vol. 83, no. 2, pp. 121–139, 1989.

[6] M. Benedikt, L. Libkin, and F. Neven, "Logical definability and query languages over ranked and unranked trees," *ACM Trans. Comput. Log.*, vol. 8, no. 2, p. 11, 2007.

[7] H. Straubing, *Finite Automata, Formal Logic, and Circuit Complexity*, ser. Progress in Theoretical Computer Science. Birkhauser, 1994.

[8] W. Hanf, "Model-theoretic methods in the study of elementary logic," in *The Theory of Models*, ser. Studies in Logic and the Foundations of Mathematics, J. Addison, L. Henkin, and A. Tarski, Eds. North-Holland, 2014, pp. 132–145.

[9] H. Gaifman, "On local and non-local properties," in *Proceedings of the Herbrand Symposium*, ser. Studies in Logic and the Foundations of Mathematics, J. Stern, Ed. Elsevier, 1982, vol. 107, pp. 105–135.

[10] R. Fagin, L. Stockmeyer, and M. Vardi, "On monadic NP vs monadic co-NP," *Inf. Comput.*, vol. 120, no. 1, pp. 78–92, 1995.

[11] M. Benedikt and L. Segoufin, "Regular tree languages definable in FO and in FO_{mod}," *ACM Trans. Comput. Log.*, vol. 11, no. 1, pp. 4:1–4:32, 2009.

[12] M. Gyssens, J. Paredaens, D. Van Gucht, and G. H. L. Fletcher, "Structural characterizations of the semantics of XPath as navigation tool on a document," in *Proceedings of the 25th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 26-28, 2006, Chicago, Illinois, USA*, S. Vansummeren, Ed. ACM, 2006, pp. 318–327.

[13] G. H. L. Fletcher, M. Gyssens, J. Paredaens, D. Van Gucht, and Y. Wu, "Structural characterizations of the navigational expressiveness of relation algebras on a tree," *J. Comput. Syst. Sci.*, vol. 82, no. 2, pp. 229–259, 2016.

[14] W. Charatonik and P. Witkowski, "Two-variable logic with counting and trees," *ACM Trans. Comput. Log.*, vol. 17, no. 4, p. 31, 2016.

[15] M. Marx, "Conditional XPath," *ACM Trans. Database Syst.*, vol. 30, no. 4, pp. 929–959, 2005.

[16] B. ten Cate and M. Marx, "Navigational XPath: calculus and algebra," *SIGMOD Rec.*, vol. 36, no. 2, pp. 19–26, 2007.

[17] M. Marx and M. de Rijke, "Semantic characterizations of navigational XPath," *SIGMOD Rec.*, vol. 34, no. 2, pp. 41–46, 2005.

[18] M. Otto, "Graded modal logic and counting bisimulation," arXiv:1910.00039, 2019.

[19] J. Hellings, "On Tarski's Relation Algebra: querying trees and chains and the semi-join algebra," Ph.D. dissertation, Hasselt University and transnational University of Limburg, 2018.

[20] J. Hellings, Y. Wu, M. Gyssens, and D. Van Gucht, "The power of Tarski's Relation Algebra on trees," *J. Log. Algebr. Methods Program.*, vol. 126, 2022.

[21] J. Hellings, C. L. Pilachowski, D. Van Gucht, M. Gyssens, and Y. Wu, "From Relation Algebra to Semi-join Algebra: An approach to graph query optimization," *Comput. J.*, vol. 64, no. 5, pp. 789–811, 2021.