RAIRO-Theor. Inf. Appl. 60 (2026) 13
https://doi.org/10.1051/ita/2026009

RAIRO - Theoretical Informatics and Applications
www.rairo-ita.org

ON DESCRIBING TREES AND QUASI-TREES FROM THEIR LEAVES

Bruno Courcelle*

Abstract. Generalized trees, we call them *O-trees*, are defined as *hierarchical* partial orders, *i.e.*, such that the elements larger than any one are linearly ordered. This partial order generalizes the ancestor relation of a rooted tree. As for rooted trees, the root of an O-tree is its unique maximal element if there is such; however, an O-tree may have no root. *Quasi-trees* are, roughly speaking, undirected O-trees. For O-trees and quasi-trees, we define relational structures on their leaves that characterize them up to isomorphism. These structures have axiomatizations by universal first-order sentences. Our main purpose is to reconstruct O-trees and quasi-trees from their leaves and relevant relations on leaves, by logically defined transformations, called *transductions. Monadic second-order (MSO) transductions* are defined by monadic second-order (MSO) formulas. We use actually *CMSO transductions* where the letter "C", meaning *counting,* refers to the use of set predicates that count cardinalities of finite sets modulo fixed integers. O-trees and quasi-trees make it possible to define respectively, the *modular decomposition* and the *rank-width* of a countable graph. Their constructions from their leaves by transductions of different types apply to rank-decompositions, to modular decomposition and to other canonical graph decompositions.

Mathematics Subject Classification. 05C05, 03B70

Received April 5, 2025. Accepted February 16, 2026.

1. Introduction

In graph theory, a *tree* is a connected graph without cycles (hence without loops and parallel edges). A *rooted tree* has a distinguished node called its *root*, from which one gets a partial order, the *ancestor relation.* The root is thus the unique maximal element. A more general notion of tree has been defined, in particular by Fraïssé [1] (Section 2.11). The notion of a rooted tree is generalized into that of a *hierarchical* partial order, *i.e.*, such that the set of elements larger than any one is linearly ordered. Such a partial order generalizes the ancestor relation of a rooted tree. We call *O-trees* these generalized trees defined as partial orders[1], by keeping the standard terminology for usual trees. If an O-tree has a (necessarily unique) maximal element, this element is its *root*[2]. A dense linear order, as that of rational numbers is an O-tree without root and leaves.

An O-tree is a *join-tree* if any two nodes y, z have a least common "ancestor" $y \sqcup z$, *i.e.*, a least upper-bound with respect to the associated partial order, called shortly their *join.*

Keywords and phrases: Generalized tree, quasi-tree, monadic second-order logic, monadic second-order transduction.

Labri, Bordeaux University, 351 Cours de la Libération, 33400 Talence, France.

* Corresponding author: bruno.courcelle@labri.fr

[1] They are called simply *trees* by Fraïssé and *order-trees* in [12] as opposed to *successor-trees*, defined by a father relation. They are not *ordered trees*, where the sons of a node are linearly ordered.

[2] Fraïssé reverses the order, so that the root is the minimal element if it exists. In fundamental computer, the root of a tree is placed at the top, we follow this convention.

Apart from its intrinsic interest in the theory of relations, the notion of join-tree permits to define the modular decomposition of a countable graph [2] by extending the existing notion for finite graphs. Its formalization in Monadic Second-Order (MSO) logic involves the description of a join-tree from its leaves forming a set L, related by the ternary relation R such that $R(x, y, z)$ holds if and only if $x \leq y \sqcup z$. We call such (L, R) a *leaf structure*. This logical description has been studied in [3–5] for finite trees, and in [2] for infinite join-trees. The article [4] shows applications to several canonical graph decompositions, in particular, the *split* and the *bi-join* decompositions.

In [6] we also defined *quasi-trees* as, roughly speaking, undirected join-trees. The motivation was to define the *rank-width* of a countable graph in such a way that it be the least upper bound of those of its finite induced subgraphs, as for tree-width [7]. The definition of a quasi-tree is based on a ternary relation B on nodes called *betweenness*: $B(x, y, z)$ means that y is on the unique path between x and z in the case of a tree. It has been defined and studied in other cases [8–10]. We call *leaves* the nodes y that never occur in the middle of any triple (x, y, z) in B. Betweenness is empty on leaves and is thus useless for describing quasi-trees from their leaves. Hence, we introduce a 4-ary relation on leaves that we call the *separation relation*: $S(x, y, z, u)$ means that x and y on the one hand, and z and u on the other, are separated by at least two internal nodes. We extend this notion to the induced substructures of quasi-trees that we call *partial quasi-trees*. We represent them topologically as embeddings in the trees of half-lines that we have already defined in [11].

In the present article, we obtain first-order (FO) axiomatizations of the leaf structures of join-trees and of the separation structures of quasi-trees. We also examine the constructions of O-trees and quasi-trees by logically specified transformations of relational structures called *monadic second-order (MSO) transductions*. We extend to large classes of join-trees a construction by Bojanczyk [3] for finite trees[3] based on *counting monadic second-order (CMSO) logic*, the extension of MSO logic that uses set predicates $C_n(X)$ expressing that a set X is finite and that its cardinality is a multiple of n.

The constructions of [5] for finite trees and of [2] for join-trees use auxiliary linear orders and are *order-invariant* which means that the resulting structures do not depend, up to isomorphism, on the chosen order. That a set has even cardinality is an order-invariant MSO property. Order-invariant MSO formulas are more expressive than CMSO ones [13].

All countable *leafy* join-trees (such that every internal node is the join of two leaves) can be constructed from their leaf structures (L, R) by a fixed order-invariant MSO transduction. The constructions of [3, 4] use instead the less powerful CMSO transductions. We extend their constructions to certain leafy join-trees. Furthermore, by using substitutions to leaves in O-trees, we build *CMSO-constructible* classes of infinite join-trees, that is, classes of join-trees that can be recontructed from their leaf structures by CMSO-transductions.

Section 1 reviews O-trees and axiomatizes the leaf structures of join-trees. Section 2 investigates the constructions of leafy join-trees from their leaves by CMSO-transductions. Section 3 examines the related questions for quasi-trees.

2. Order-trees

All trees and related logical structures will be finite or countable. We will denote by $\simeq$ the isomorphism of relational structures, and *u.t.i.* will mean *up to isomorphism* when comparing two relational structures.

2.1. O-trees and join-trees

Definitions and notation are from [1, 2, 11, 14, 15].

Definition 2.1. *O-trees.*

(a) A partial order $\leq$ on a set N is *hierarchical* if it satisfies the *linearity condition, i.e.*, if for all x, y, z in N, $x \leq y \wedge x \leq z$ implies $y \leq z \vee z \leq y$. If so, $T = (N, \leq)$ is an *O-forest* and N is its set of *nodes*. If any two

[3] A different construction using a CMSO-transduction is given in [4].

nodes have an upper bound, we call it an *O-tree*. The comparability graph of an O-tree is connected and so is its Gaifman graph[4].

A minimal element is called a *leaf* and L denotes the set of leaves. A node that is not a leaf is *internal*. A maximal element of N is called a *root*. An O-tree has at most one root.

A finite O-tree $(N, \leq)$ is a rooted tree, N is its set of nodes and $\leq$ is its *ancestor* relation. We will use for finite and infinite rooted trees the standard notions of *father* and *son*. The *covering relation* of a partial order $\leq$ is $<_c$ such that $x <_c y$ if and only if $x < y$ and there is no z such that $x < z < y$. If in an O-tree $(N, \leq)$ we have $x <_c y$, we say that x is a *son* of y and y is the (unique) *father* of x.

(b) If $T = (N, \leq)$ is an O-forest, we denote by T_x the O-tree whose set of nodes is $N_x := \{y \in N \mid y \leq x\}$ ordered by the restriction of $\leq$ to N_x. Its root is x. We call it a *sub-O-tree* of T. More generally, if $M \subseteq N$ is *downwards closed*, that is if $y \leq x$ and $x \in M$ imply $y \in M$, then $(M, \leq_M)$ is an O-forest (or an O-tree) and is also called a *sub-O-forest* (or a *sub-O-tree*) of T.

(c) An O-tree $T = (N, \leq)$ is a *join-tree* [2] if every two nodes x and y have a least upper-bound, called their *join* and denoted by $x \sqcup y$. If x, y and z are nodes of a join-tree, the joins $x \sqcup y$, $x \sqcup z$ and $y \sqcup z$ are all equal, or two of them are equal and larger than the third one. Then, $(x \sqcup y) \sqcup (z \sqcup u)$ is $x \sqcup y$ or $z \sqcup u$ or we have $x \sqcup z = y \sqcup z = x \sqcup u = y \sqcup u$.

(d) An O-forest is *leafy* if every internal node is the join of two leaves. No node can have a unique son. Two incomparable nodes may have no join.

We studied already O-trees in [11, 15] by focusing on their monadic second-order definability and its equivalence with regularity, a notion derived from the classical theory of infinite rooted trees [16].

Example 2.2. (1) The ordered set of rational numbers $(\mathbb{Q},\leq)$ is an O-tree without root and leaves.

(2) Let $U := (\mathbb{N} \times \{0, 1\} \cup \{a, b\}, \geq_U)$. It is an O-tree whose order $\leq_U$ is defined by $(n, i) \leq_U (m, 0)$ if $m \leq n$ and i is 0 or 1, $a, b \leq_U (n, 0)$, for all m, n. Its leaves are the nodes $(n, 1)$ together with a, b. The root is $(0, 0)$. It is leafy. The nodes a and b have no join because $\{y \mid y \geq_U a, y \geq_U b\}$ is isomorphic to $\mathbb{N}$ with the reverse order.

We now construct somewhat weird examples of O-trees.

(3) We let A and B be the disjoint countable dense sets of real numbers such that B is the set of nonnegative rational numbers and $A \subset \mathbb{R} - \mathbb{Q}$ is $(\pi + \mathbb{Q}) \cap]0, +\infty[$. Hence 0 is the least element of $A \cup B$.

We define on the set of words[5] $(A \cup B)^*$ the following partial order : $w \leq w'$ if and only if:

w' is a prefix of w (that is, $w = w'u$ for some word u),
or $w = uxv$, $w' = uy$ where $u, v \in (A \cup B)^*$, $x, y \in A \cup B$ and $x < y$.

The elements larger than $w = ux$ are its prefixes and those of the form uy for $y > x$ (where $x, y \in A \cup B$). Hence $(A \cup B)^*$ ordered in this way is an O-tree with root ε (the empty word), and without leaves as we have wv below w for any word v. It is a join-tree: two incomparable words w,w' are of the form $w = uxv$ and $w' = uyv'$ for some $x \neq y$ in $A \cup B$ and nonempty words v, v'. Then $w \sqcup w' = u \max(x, y)$.

We now define its restriction $T := (N, \leq)$ where $N := A^* \cup A^*B$. By the above observations, it is a join-tree with root ε. Its leaves are the words $u0$ for all u in A^*. The join of two incomparable nodes is in A^*. We go below an internal node u in A^* by appending a word to it, or by replacing its last symbol by a smaller one.

Figure 1 illustrates this definition with $a, b, c, d \in A$ and $p, q, r, s \in B$ such that $0 < r < q < d < s < c < b < p < a$. This figure shows some nodes of T. To clarify the figure, the nodes u in A^* are marked (u) and those in A^*B are marked $[u]$. The dotted lines indicate countably many nodes between two shown nodes ($A \cup B$ is dense). The rightmost branch on Figure 1 has only nodes in A^*. The shown leaves are $[0], [a0]$ and $[ac0]$. In T, we have:

$$(b) \sqcup (ad) = (b) \sqcup [as] = [q] \sqcup [a0] = (a) \text{ and } [aq] \sqcup [acs] = (ac).$$

[4]The *Gaifman graph* of a relational structure (D, R) has vertex set D and edges between any two elements occuring in some tuple of R.

[5]Words are finite sequences of elements from a possibly infinite set.

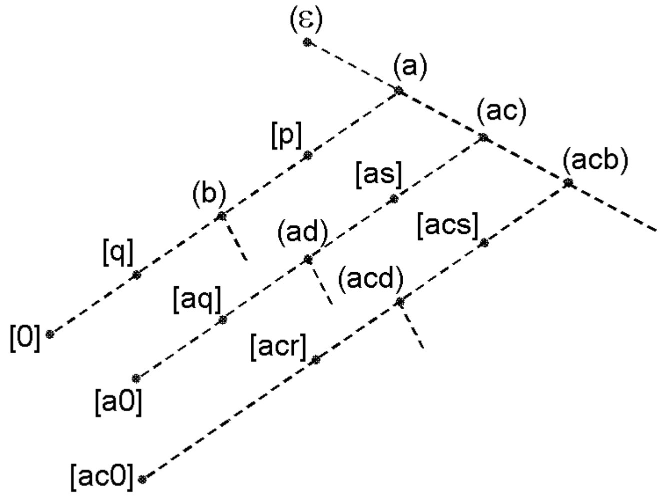

FIGURE 1. The join-tree T of Example 2.2 (3).

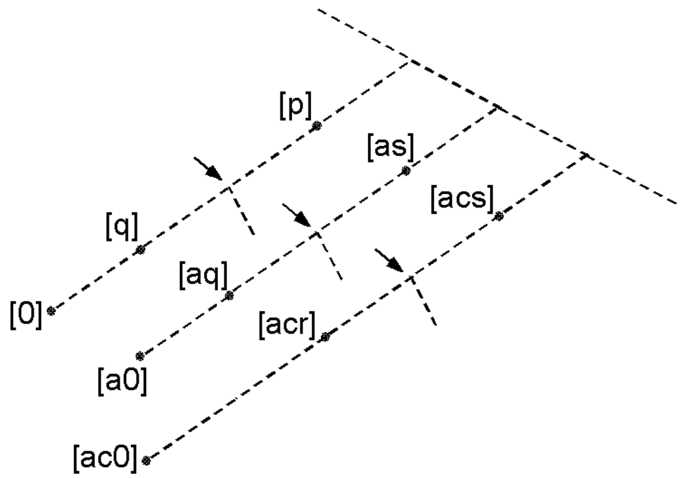

FIGURE 2. The O-tree S of Example 2.2 (4).

Any node ua, where $a \in A$, is the join $u0 \sqcup ua0$ but T is not leafy: the leaves below ub, where $b \in B$ are $u0$ and the $ub'v0$'s for $b' < b$. Any two have upper-bounds of the form uc for $c \in A$, $c < b$, but no least one as A is dense without a smallest element. The root is not the join of any two nodes.

(4) From T as above, we delete all nodes in A^*, *i.e.*, those marked (u) on Figure 1. The resulting O-tree S has the following properties:

> no two incomparable nodes have a join (it would have to be in A^*), for every internal node x, the sub-O-tree T_x has infinitely many leaves and x has a single direction, in T as well as in T_x.

We get the O-tree S of Figure 2. It has no root. The three arrows in Figure 2 point to the Dedekind cuts replacing the nodes $(b), (ad)$ and (acd) deleted from T.

The notion of direction in an O-tree is defined in [2, 14], Definition 3.2, in both articles. An internal node w has a *single direction* if for all nodes $w', w'' < w$, we have $w', w'' \leq s$ for some $s < w$. In the O-tree $(\mathbb{Q}, \leq)$, each node has a single direction. In S, the nodes strictly below ub, where $b \in B$, are uc for $c \in B$, $c < b$ and uav for $a \in A$, $a < b$ and v in A^*B. Any two of these forms are below ud for some $d \in B, d < b$ since B is dense.

We let $\mathcal{T}$ be the class of leafy join-trees. This class excludes an ascending chain like $(\mathbb{Q}, \leq)$ and join-trees with nodes having a unique son or a unique direction (*cf.* S in Example 2.2). Our objective is to describe a leafy join-tree $T = (N, \leq)$ from its leaves. As the order $\leq$ is trivial on leaves, we use instead a ternary relation.

Definition 2.3. [2, 5, 17]. *Relations between leaves.*

For T in $\mathcal{T}$, we let R be the ternary relation on its set of leaves L, defined by $Rxyz :\Longleftrightarrow x \leq y \sqcup z$. We write $Rxyz$ for $R(x, y, z)$.

We have $x \sqcup y \leq z \sqcup u \Longleftrightarrow Rxzu \wedge Ryzu$. Clearly, $Rxyz$ implies $Rxzy$, and $Rxxz$ holds for all x and z. If T has only one node x, that is thus the root and also a leaf, then $Rxxx$ holds. If T has one root and two leaves x and y, then $Rxxy$ and $Ryxy$ hold. Other properties will be given below.

We call (L, R) the *leaf structure* of T, denoted more precisely by (L_T, R_T).

Proposition 2.4. *If T and S are two trees in $\mathcal{T}$ such that $(L_T, R_T) = (L_S, R_S)$, then $T \simeq S$.*

We will say that T is *characterized* by its leaf structure (L_T, R_T).

Proof. This is proved for finite trees in Proposition 5.1 of [5] but the proof works for leafy join-trees. We recall it.

Let $T \in \mathcal{T}$ and $(L, R) := (L_T, R_T)$. Let j be the surjective mapping: $L \times L \longrightarrow N_T$ such that $j(x, y) := x \sqcup_T y$. We have $j(x, y) \leq_T j(z, u)$ if and only if $Rxzu \wedge Ryzu$. Hence, N_T is in bijection with $L \times L / \equiv$ where $(x, y) \equiv (z, u) :\Longleftrightarrow Rxzu \wedge Ryzu \wedge Rzxy \wedge Ruxy$. The quotient order[6] $[(x, y)]_\equiv \leq [(z, u)]_\equiv$ is equivalent to $j(x, y) \leq_T j(z, u)$ and is definable from R.

If $T = (N_T, \leq) \in \mathcal{T}$, one can reconstruct T from N_T and R_T up to isomorphism. If T and S are in $\mathcal{T}$ and $(L_T, R_T) = (L_S, R_S)$, then $T \simeq S$. $\square$

The following proposition and Theorem 2.10 yield an axiomatization of the leaf structures of the join-trees in $\mathcal{T}$ by a single *universal FO* ($\forall FO$ in short) sentence, that is, written with a sequence of universal quantifications applied to a quantifier-free formula. Furthermore, these formulas will be written without function symbols.

Proposition 2.5. *Let $T = (N_T, \leq)$ be a join-tree. The following properties hold for all leaves x, y, z, u and v where $R := R_T$:*

A1: $Rxxy$,
A2: $Rxyz \Longrightarrow Rxzy$,
A3: $Rxyy \Longrightarrow x = y$,
A4: $Rxyz \wedge Ryuv \wedge Rzuv \Longrightarrow Rxuv$,
A5: $Rxyz \wedge Rxuv \Longrightarrow (Ryuv \wedge Rzuv) \vee (Ruyz \wedge Rvyz)$.

Furthermore, we have:

$$x \sqcup_T z = y \sqcup_T z = x \sqcup_T y \Longleftrightarrow R_T xyz \wedge R_T yxz \wedge R_T zxy.$$

Proof. Easy verifications. See Figure 3 illustrating A4. $\square$

Proposition 2.6. *For every ternary structure (L, R), the following properties are consequences of the universally quantified formulas A1–A5:*

[6] We denote by $[(x, y)]_\equiv$ the equivalence class of (x, y), and by $[(x, y)]$ if $\equiv$ is understood from the context.

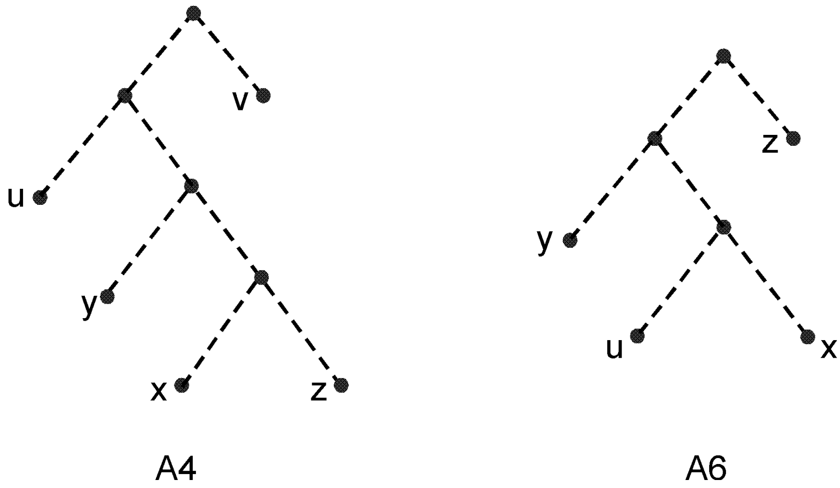

FIGURE 3. See Propositions 2.5 and 2.6.

A6: $Rxyz \wedge Ruxy \Longrightarrow Ruyz$,
A7: $Rxyz \vee Rzxy$
A8: $\neg Rxyz \Longrightarrow Ryxz \wedge Rzxy$,
A9: $Rxyz \Longrightarrow Ryxz \vee Rzxy$.

Proof. (1) Property A6 (illustrated in Figure 3) is a consequence of A1 and A4 because:

$Ruxy \wedge Rxyz \Longrightarrow Ruxy \wedge Rxyz \wedge Ryyz$ and
$Ruxy \wedge Rxyz \wedge Ryyz \Longrightarrow Ruyz$ by A4.

(2) Property A7 is a consequence of A1, A2 and A5: by A1 and A2, we have $Ryxy \wedge Ryyz$. Then A5 yields $(Rxyz \wedge Ryyz) \vee (Ryxy \wedge Rzxy)$. By A1 and A2, this reduces to $Rxyz \vee Rzxy$.

(3) We also have $Rxzy \vee Ryxz$, hence $Rxyz \vee Ryxz$by A2. We obtain[7]: $Rxyz \vee (Ryxz \wedge Rzxy)$ that can be rewritten into A8.

(4) From A1–A5, we prove A9: $Rxyz \Longrightarrow Ryxz \vee Rzxy$ by contradiction; assume this is false for some x, y and z. We have $Rxyz \wedge \neg Ryxz \wedge \neg Rzxy$. By A8, we get $Rxyz \wedge (Rxyz \wedge Rzyx) \wedge \neg Rzxy$ which gives a contradiction by A2.

□

Remark 2.7. (a) Property A5 is not a consequence of A1–A4. If one interprets $Rxyz$ as $x \in [y, z] \cup [z, y]$ on rational numbers, then A1–A4 (whence A6) hold but not A5, as $[y, z]$ and $[u, v]$ may be disjoint. Properties A7–A9 proved with A5 do not hold either.

(b) The following facts have been proved by the prover Z3 [18].

A1–A3, A6 do not imply A4 (*cf.* (1) above).
A1–A3, A7 imply neither A4 nor A5 (*cf.* (2) above).
A1–A4, A9 do not imply A5 (*cf.* (4) above).
However, A1–A4, A7 imply A5.

(c) Let us define

$R_0xyz :\Longleftrightarrow Rxyz \wedge Ryxz \wedge Rzxy$, and
$R_1xyz :\Longleftrightarrow Rxyz \wedge Ryxz \wedge \neg Rzxy$.

[7] Compare it to the last assertion of Proposition 2.5.

Then, by A1,A2 and A8 we have

A10: $R_0xyz \oplus R_1xyz \oplus R_1yxz \oplus R_1zxy$

where denotes the exclusive "or".

Property R_0xyz means that $x \sqcup z = y \sqcup z = x \sqcup y$. Property R_1xyz holds in the right part of Figure 3.

Construction 2.8. *From A1–A5 to a join-tree.*

Let (L, R) satisfy properties A1 to A5, expressed by $\forall FO$ (first-order) sentences. We construct T in $\mathcal{T}$ such that $(L, R) \simeq (L_T, R_T)$. We use the construction of Proposition 2.4 by assuming only A1–A5, whereas in that construction, we assumed that (L, R) is (L_T, R_T) for some tree T in $\mathcal{T}$.

We define as follows a structure $(L \times L, \sqsubseteq)$ by writing xy for $(x, y) \in L \times L$ and defining $xy \sqsubseteq zu :\Longleftrightarrow Rxzu \wedge Ryzu$. Lemma 2.9 will show that $\sqsubseteq$ is a quasi-order. Then, we let $T := (N, \leq)$ be the corresponding quotient structure. That is, we let

$xy \equiv zu :\Longleftrightarrow xy \sqsubseteq zu \wedge zu \sqsubseteq xy :\Longleftrightarrow Rxzu \wedge Ryzu \wedge Rzxy \wedge Ruxy$,

we define $N := L \times L/\equiv$ and $\leq$ as the quotient partial order $\sqsubseteq / \equiv$.

Lemma 2.9. *The relation $\sqsubseteq$ is a quasi-order such that, if $xy \sqsubseteq zu$ and $xy \sqsubseteq ws$, then $zu \sqsubseteq ws$ or $ws \sqsubseteq zu$.*

Proof. That $xy \sqsubseteq xy$ (and $xy \sqsubseteq yx$) follows from A1 and A2.

Property A4 shows that $Rxzu$ and $zu \sqsubseteq ws$ imply $Rxws$. Similarly, $Ryzu$ and $zu \sqsubseteq ws$ imply $Ryws$. It follows that $xy \sqsubseteq zu \sqsubseteq ws$ implies $xy \sqsubseteq ws$, hence that $\sqsubseteq$ is transitive and is a quasi-order.

We now prove what will yield the linearity condition for its associated partial order: if $xy \sqsubseteq zu$ and $xy \sqsubseteq ws$ then, $Rxzu$ and $Rxws$ hold, and we get $zu \sqsubseteq ws$ or $ws \sqsubseteq zu$ by A5.

□

Theorem 2.10. *The leaf structures of join-trees are the ternary structures that satisfy A1–A5.*

Proof. We use Construction 2.8 to prove the converse of Proposition 2.5. From (L, R) satisfying A1–A5, we get $T := (N, \leq)$. By Lemma 2.9, $\leq$ is a partial order and it is hierarchical. Hence, T is an O-forest.

It follows from[8] A3 that its leaves are the elements $[xx]$ for $x \in L$. Note that $xx \equiv zu$ if and only if $x = z = u$.

The join of two leaves $[xx]$ and $[yy]$ is $[xy]$, hence T is an O-tree. Furthermore, $([xx], [yy], [zz]) \in R_T$ if and only if $Rxyz$ holds because $Rxyz$ holds if and only if $xx \sqsubseteq yz$.

Claim 2.11. *Any two elements $[xy]$ and $[zu]$ have a join.*

Proof. If $xy \sqsubseteq zu$ then $[xy] \sqcup [zu] = [zu]$. If $zu \sqsubseteq xy$ then $[xy] \sqcup [zu] = [xy]$.

If $Rzxy$ holds, we have $zz \sqsubseteq zu$ and $zz \sqsubseteq xy$, hence $xy \sqsubseteq zu$ or $zu \sqsubseteq xy$ by A5. Similarly if $Ruxy$, $Rxzu$, or $Ryzu$ holds, then $[xy] \sqcup [zu]$ is defined and is either $[xy]$ or $[zu]$.

If none of these facts holds, from $\neg Rzxy$, $\neg Ruxy$, $\neg Rxzu$ and $\neg Ryzu$, we get by A8 (*cf.* Proposition 2.6) and A2:

$$(Rxzy \wedge Ryzx) \wedge (Rxuy \wedge Ryux) \wedge (Ruxz \wedge Rzxu) \wedge (Rzyu \wedge Ruyz),$$

which gives $xu \sqsubseteq yz, yu \sqsubseteq xz, xz \sqsubseteq yu$ and $yz \sqsubseteq xu$. Hence, $xz \equiv yu$ and $yz \equiv xu$.

We also have $xz \equiv xu$ since $Rxxu \wedge Rzxu$ holds and similarly, $yz \equiv yu$. Finally, $xz \equiv xu \equiv yz \equiv yu$.

If $xy \sqsubseteq ws$ and $zu \sqsubseteq ws$, we get $xz \sqsubseteq ws$, hence $[xy] \sqcup [zu] = [xz]$. □

It follows that T is a leafy join-tree. As said above, its leaves are the singleton equivalence classes $[xx]$ for $x \in L$. □

This construction applies if L is $\{x\}$ or $\{x, y\}$. In both cases, R is not empty by A1. We obtain respectively a single node $[xx]$ or a *star* with root $[xy]$ and leaves $[xx]$ and $[yy]$.

We denote by $\tau(L, R)$ the isomorphism class of T constructed as above from (L, R).

[8] We use A3 only here.

Remark 2.12. (1) Remark 2.7(b) shows that leaf structures are also axiomatized by A1–A4 and A7.

(2) Some properties of $\tau(L, R)$ are FO-expressible in (L, R). For examples, the properties that for $x, y, z, u \in L$, the node $x \sqcup y$ is the root or that $x \sqcup y$ is a son of $z \sqcup u$, *i.e.,* that $x \sqcup y <_c z \sqcup u$, *cf.* Definition 2.1(b).

(3) The validity of a $\forall$FO sentence is preserved under taking induced substructures. Hence, if $T = (N, \leq) \in \mathcal{T}$ and $X \subseteq L_T$, then the (induced) substructure[9] $(X, R_T[X])$ of (L_T, R_T) satisfies properties A1–A5. It defines the join-tree $(Y, \leq')$ such that $Y = \{x \sqcup y \mid x, y \in X\} \subseteq N$ (this set contains X) and $\leq'$ is the restriction of $\leq$ to Y. $\square$

The following corollary is a *Backwards Translation Theorem* relative to the transduction τ. It means that the validity of a sentence φ in $\tau(S)$ is equivalent the validity of a sentence ψ in S, constructed from φ and the formulas defining τ. The case of MSO transductions and sentences is detailed in [19], Theorem 7.10.

Corollary 2.13. *Let φ be a first-order sentence over structures $(N, \leq)$. One can construct a first-order sentence ψ such that for every ternary structure (L, R), the join-tree $\tau(L, R)$ is defined and $\tau(L, R) \models \varphi$ if and only if $(L, R) \models \psi$.*

Proof sketch: It follows from Construction 2.8 that the mapping $(L, R) \longmapsto \tau(L, R)$ is the composition of two transformations as follows:

$$(L, R) \longmapsto (L \times L, \sqsubseteq) \longmapsto (L \times L/\equiv, \sqsubseteq/\equiv) \simeq \tau(L, R).$$

Since $\equiv$ is first-order definable in $(L \times L, \sqsubseteq)$, one can build a sentence θ such that, if $\tau(L, R)$ is defined (*i.e.*, satisfies A1–A5), then $\tau(L, R) \models \varphi$ if and only if $(L \times L, \sqsubseteq) \models \theta$. From θ, one can construct ψ_1 such that $(L \times L, \sqsubseteq) \models \theta$ if and only if $(L, R) \models \psi_1$. Then, one defines ψ as $\psi_1 \wedge \psi_2$ where ψ_2 is the conjunction of the universally quantified axioms A1–A5. $\square$

Backwards translation as in Theorem 7.10 of [19] works well for MSO (Monadic Second-Order) sentences φ and ψ and MSO-transductions. However, the intermediate step $(L, R) \longmapsto (L \times L, \sqsubseteq)$ is *not* an MSO transduction, but a *vectorial FO-transduction of dimension 2* because of $L \times L$, the domain of the output structure. We refer to [20] for a review of different types of FO-transductions, either vectorial or not.

In Section 3, we will consider cases where τ can be defined alternatively by a transduction such that a CMSO[10] sentence ψ can be constructed from a CMSO sentence φ to satisfy the corollary. In such an alternative construction, the set of nodes N is built as a subset of $L \times \{0, ..., k\}$ for some fixed k, and not of $L \times L$.

2.2. Leafy O-forests

We want to extend the results we already have for join-trees to leafy O-forests.

For an O-forest $T = (N, \leq)$ and leaves x, y, we let $N_{\geq}(x, y) := \{u \in N \mid u \geq x \text{ and } u \geq y\}$. Each set $N_{\geq}(x, y)$ is linearly ordered; it is empty if x and y have no upper bound. If T is an O-tree, none of these sets is empty.

We let R be the ternary relation on the set L of leaves defined as follows:

$$Rxyz :\Longleftrightarrow x \leq N_{\geq}(y, z),$$

that is, $x \leq u$ for every node u in $N_{\geq}(y, z)$. If $y \sqcup z$ is defined, then $N_{\geq}(y, z) = N_{\geq}(y \sqcup z)$. If T a join-tree, then $Rxyz \Longleftrightarrow x \leq y \sqcup z$, and this new definition is equivalent to that of Definition 2.3.

We recall a construction from [11], Section 1.3: from an O-forest T, one can construct a join-tree $J(T)$ by adding to it the "missing joins". Hence, T embeds into $J(T)$ and $J(T) = T$ if T is a join-tree. In Example 2.2 (4), by adding to S the "missing joins" we get back T.

Proposition 2.14. *(Proposition 1.4 of [11]): let $T = (N, \leq)$ be an O-forest. It can be embedded into a join-tree $J(T)$ whose nodes are the sets $N_{\geq}(y, z)$ for $y, z \in N$. Its order is $\supseteq$, the reverse set inclusion, and $x \in N$ is*

[9]If $S = (D, R, ...)$ is a relational structure and $X \subseteq D$, then $S[X] := (X, R[X], ...)$ is the *substructure* of S *induced on* X, where $R[X]$ is the restriction of R to X, *i.e.*, the set of tuples in R whose components are in X. This corresponds to the notion of an induced subgraph. We will only consider substructures of this form.

[10]See Introduction and Section 2.1 for definitions.

mapped to $N_{\geq}(x) \in N_{J(T)}$. *Its leaves are the sets* $N_{\geq}(x)$ *for* $x \in L_T$. *If* $T \in \mathcal{T}$, *then* $N_{J(T)}$ *is the set of sets* $N_{\geq}(y,z)$ *for* $y, z \in L_T$. *If* T *is not an O-tree, there are* y, z *such that* $N_{\geq}(y,z)$ *is empty, and then,* $\emptyset$ *is the root of* $J(T)$.

Proposition 2.15. *If* T *is an O-forest, then* $R_T = R_{J(T)}$ *and this relation satisfies A1–A5.*

Proof. Let $x, y, z \in L_T$. We have $N_{\geq}(y) \sqcup_{J(T)} N_{\geq}(z) = N_{\geq}(y,z)$. Hence $x \leq N_{\geq}(y,z)$, *i.e.* $R_T xyz$ holds if and only if $N_{\geq}(x) \supseteq N_{\geq}(y,z) = N_{\geq}(y) \sqcup_{J(T)} N_{\geq}(z)$, equivalently, if and only if $(N_{\geq}(x), N_{\geq}(y), N_{\geq}(z)) \in R_{J(T)}$. □

The O-forest T is isomorphic to a substructure of $J(T)$. The formulas A1–A5 are universal. Remark 2.12(3) recalls that their validity is preserved under taking substructures. As they hold in $J(T)$, they hold in T. It follows that, from (L_T, R_T) where T is a leafy O-forest, one can define $J(T)$ but not T: clearly, two different O-forests can have the same associated ternary relation R.

Note that Property A1 implies that the Gaifman graph of (L, R) is connected. Hence this property cannot hold for an O-forest consisting of several disjoint O-trees.

A leafy O-forest T is not characterized by (L_T, R_T). We define U_T as the binary relation such that, if $x, y \in L_T$, then $U_T xy$ holds if and only if $x \sqcup_T y$ is defined.

Proposition 2.16. *A leafy O-forest* T *is characterized by (can be reconstructed from) its associated extended leaf structure* (L_T, R_T, U_T).

Proof. Given (L_T, R_T), one defines $J(T)$ from which one eliminates the nodes $N_{\geq}(x,y)$ such that $U_T xy$ does not hold.

□

3. Constructing join-trees by monadic second-order transductions

Monadic second-order transductions are transformations of relational structures formalized by MSO formulas.

3.1. Review of definitions and results

A *monadic second-order (MSO) transduction* transforms a relational structure $S = (D_S, R_S, ...)$ into $T = (D_T, R'_T, ...)$ by means of MSO formulas collected in a *definition scheme* Δ. (More details can be found in [2, 5, 14, 19]).

To simplify notation, we let S have a unique relation R, and T have a unique binary relation R'. The extension to more relations and of different arities is straightforward.

A definition scheme Δ uses set variables $X_1, ..., X_p$ intended to denote subsets of D_S called the *parameters*, and a *copying constant* $k \geq 1$. Then $\Delta = (\chi, \delta_1, ..., \delta_k, (\theta_{i,j})_{i,j \in [k]})$. It is a tuple of MSO formulas[11] to be evaluated in a structure $S = (D_S, R_S)$, such that χ has free variables among $X_1, ..., X_p$, each δ_i has free variables among $X_1, ..., X_p, x$ and each $\theta_{i,j}$ has free variables among $X_1, ..., X_p, x, y$.

For subsets $X_1, ..., X_p$ of D_S, a structure $T := \widehat{\Delta}(S, X_1, ..., X_p)$ is defined if and only if $S \models \chi(X_1, ..., X_p)$. We have[12]:

$D_T := D_1 \times \{1\} \cup ... \cup D_k \times \{k\}$ where for each i,
$D_i := \{x \in D_S \mid S \models \delta_i(X_1, ..., X_p, x)\}$ and
$R'_T := \{((x,i),(y,j)) \mid i, j \in [k], (x,y) \in D_i \times D_j$ and $S \models \theta_{i,j}(X_1, ..., X_p, x, y)\}$.

The structure T depends on S and p (set) parameters. Hence, S has some image under the multivalued transduction defined by Δ if and only if $S \models \overline{\chi}$, where $\overline{\chi}$ is the sentence $\exists X_1, ..., X_p.\chi$. However, in our constructions,

[11] For each positive integer k, we denote $\{1, ..., k\}$ by $[k]$ and $\{0, 1, ..., k\}$ by $[0, k]$.
[12] It may be convenient to have $D_T := D_0 \times \{0\} \cup ... \cup D_k \times \{k\}$. The copying constant is then $k + 1$.

the structures obtained from different tuples of parameters will be isomorphic. We will write $T = \widehat{\Delta}(S)$ where T is defined *u.t.i.* as usual.

The expressive power of MSO formulas and MSO transductions can be augmented by using *modulo counting set predicates*: if $n \geq 2$, $C_n(X)$ expresses that the cardinality of a set X is finite and is a multiple of n. We refer to this extension by *CMSO*. The finiteness of a set X is thus expressible in terms of a single set predicate C_n.

Another extension consists in assuming that the input structure S is equipped with a linear order $\leq$ isomorphic to that of $\mathbb{N}$ if the structure is infinite. (We only consider finite and countably infinite structures). We also assume that the associated transductions are *order-invariant*, which means that the output structures do not depend, *u.t.i.*, on the chosen order. (This property is not decidable from an arbitrary definition scheme. It is provable in particular constructions.) We denote this extension by $\leq$-MSO (read *order-invariant MSO*). The property $C_n(X)$ is $\leq$-MSO expressible [5]. Order-invariant MSO sentences are strictly more powerful than the CMSO ones on finite structures [13].

We obtain the notions of CMSO- and *order-invariant* MSO transductions. The latter are more powerful than CMSO transductions, themselves stronger than the basic ones.

The composition of two MSO (resp. CMSO, $\leq$-MSO) transductions is an MSO (resp. a CMSO, a $\leq$-MSO) transduction.

The *Backwards Translation Theorem* says that, if $T = \widehat{\Delta}(S, X_1, ..., X_p)$, then an MSO (resp. a CMSO, a $\leq$-MSO) sentence φ is valid in T if and only if some "backwards translated" MSO (resp. CMSO or $\leq$-MSO) formula $\psi(X_1, ..., X_p)$ is valid in S, where $\widehat{\Delta}$ is, respectively, an MSO, a CMSO or a $\leq$-MSO-transduction. See Chapter 7 of [19]. The corresponding transformation of sentences depends only on Δ.

The *linearly expanding FO-transductions* are the MSO ones whose defining formulas use neither set quantifications, nor the set predicates C_n, nor any auxiliary order. They can use set variables as parameters. The just cited *Backwards Translation Theorem* holds for FO-sentences and such transductions.

The transformation $(L, R) \longmapsto (L \times L, \sqsubseteq)$ used in Corollary 2.13 is not an FO-transduction in this sense. It is a *vectorial FO-transduction of dimension 2,* because the domain of the output structure is a Cartesian product[13].

Going back to join-trees, the following theorem collects several results. More will be given in Section 3.4 below.

Theorem 3.1. *The following table indicates that, for certain subclasses of $\mathcal{T}$, the mapping $(L, R) \longmapsto \tau(L, R) \in \mathcal{T}$ is an MSO-transduction, or a CMSO-transduction or an $\leq$-MSO-transduction.*

Subclass of $\mathcal{T}$	*Type of transduction τ*
finite, degree $\leq k$	*MSO*
finite	*CMSO*
height $\leq k$	*FO*
each subtree is finite	*CMSO*
$\mathcal{T}$	*$\leq$-MSO*

Proof. For finite trees of bounded degree, the reference is Theorem 5.6 of [5]. Theorem 5.3 of [5] shows that for general finite trees, an $\leq$-MSO transduction can be used. This result is improved in [3, 4] with CMSO formulas in place of order-invariant ones. We reproduce below in Theorem 3.4 the proof of [3] and we extend it to certain infinite join-trees (Theorem 3.12). The (easy) case of bounded height is Proposition 3.3. Proposition 6.1 of [2] establishes the case of infinite join-trees. □

The following refers to Proposition 2.16.

Corollary 3.2. *The mapping $(L_T, R_T, U_T) \mapsto T$ producing an O-forest T is an $\leq$-MSO transduction. It is an MSO- or a CMSO-transduction in the special cases of Theorem 3.1.*

[13] See [20] for a comparative study of different types of transductions defined by FO formulas.

3.2. Bounded height

The *height* of a rooted tree is the least upper-bound of the distances of the leaves to the root. For each integer $k > 0$, we denote by $\mathcal{H}_k$ the class of rooted trees in $\mathcal{T}$ of height at most k, and by $\mathcal{S}_k$ the class of their associated leaf structures. Nodes may have infinite degrees.

Proposition 3.3. *The mapping $(L, R) \longmapsto \tau(L, R)$ is a FO-transduction from $\mathcal{S}_k$ to $\mathcal{H}_k$.*

Proof. Let (L, R) be the leaf structure of some T in $\mathcal{T}$. We recall from Theorem 2.10 that this property is FO-definable. The following properties of leaves x, y, z, u are FO-expressible in (L, R):

(1) $x \sqcup y$ is a son of $z \sqcup u$ in T, *cf.* Remark 2.12(2),
(2) $\mathrm{depth}(x \sqcup y) = i$ where $\mathrm{depth}(x \sqcup y)$ is the distance of $x \sqcup y$ to the root; the root is at depth 0.

□

The formulas expressing (2) use the one expressing (1).

The transduction encodes a node m at distance i from the root by a pair (x, i) where x is a leaf below m. Hence m is uniquely defined from (x, i). Several leaves x can be used to encode m, but set parameters X_i can select unique leaves for each internal node. We continue the proof.

We fix k. The leaves are at positive depth at most k. That $(L, R) \in \mathcal{S}_k$ is FO-expressible by (2) above.

For $0 \leq i < k$, we let D_i be the set of internal nodes at depth i, and X_i be a set of leaves such that, for each $m \in D_i$, there is one and only one leaf z in X_i such that $z < m$. We let such z be $Rep(m)$: the leaf *representing* m.

An FO-formula $\chi_i(X_i)$ can check that X_i is as requested. For each $i = 0, ..., k-1$, another one, $\rho_i(X_0, ..., X_{k-1}, x, y, z)$ can check that $R_T zxy$ holds, $z \in X_i$ and $\mathrm{depth}(x \sqcup y) = i$, so that $z = Rep(x \sqcup y)$. The transduction we are constructing uses parameters $X_0, ..., X_{k-1}$ to define D_T as $X_0 \times \{0\} \cup X_1 \times \{1\} \cup ... \cup X_{k-1} \times \{k-1\} \cup L \times \{k\}$. Here, $L \times \{k\}$ is the set of leaves of T, and $(u, 0)$ is its root, where $X_0 = \{u\}$ for some (arbitrarily chosen) leaf u.

Its order relation is defined from the following clauses:

$(z, k) < (z', i)$ where $i < k$ if and only if $Rzxy$ holds for some $x, y \in L$ such that $\mathrm{depth}(x \sqcup y) = i$, $z' \in X_i$ (so that $z' = Rep(x \sqcup y)$ and $z < x \sqcup y$),
$(z, j) < (z', i)$ where $i < j$ if and only if $Rzxy \wedge Rz'vw$ holds for some $x, y, v, w \in L$ such that $\mathrm{depth}(x \sqcup y) = i$, $\mathrm{depth}(v \sqcup w) = j$, $z \in X_j$ and $z' \in X_i$ (so that $z = Rep(x \sqcup y)$,$z' = Rep(v \sqcup w)$ and $v \sqcup w < x \sqcup y$).

An FO formula $\chi(X_0, ..., X_{k-1})$ can express that the sets $X_0, ..., X_{k-1}$ satisfy the above conditions. We omit further details.

All formulas used here are first-order but they use the set variables $X_0, ..., X_{k-1}$ as parameters.

3.3. Using CMSO transductions

We extend a construction by Bojanczyck[14] [3] that builds a CMSO-transduction producing a finite leafy rooted tree from its leaf structure (L, R). It yields an alternative to Construction 2.8 giving Theorem 2.10. We will prove that this very construction applies to certain infinite join-trees. In Section 3.4, we will identify classes of join-trees constructible from their leaves by CMSO transductions.

We recall that in a rooted tree T in $\mathcal{T}$, each internal node has at least two sons. The set of leaves below an internal node u is denoted by $Lf_T(u)$.

Theorem 3.4. *[3]. There exists a CMSO transduction that associates with a finite leaf structure (L, R) a finite leafy rooted tree T, such that $R_T = R$. Furthermore, this transduction defines $L_T := L \times \{1\}$ and $N_T - L_T \subseteq L \times \{2\}$.*

[14] A similar more complicated one is in [4].

Proof idea: As for the proof of Proposition 3.3, we will use the notion of a *representing leaf*. An internal node u of a finite leafy rooted tree is $x \sqcup_T y$ for two distinct leaves x and y and can be identified with the set $Lf_T(u) = Lf_T(x \sqcup_T y) = \{z \in L_T \mid R_T zxy\}$. The construction will select for each such $u = x \sqcup_T y$ a representing leaf w, denoted by $Rep(u)$ that belongs to $Lf_T(u)$ and that is CMSO-definable from x and y. Then u in $N_T - L_T$ will be defined as $(w, 2) \in L_T \times \{2\}$. Each leaf x will be defined as $(x, 1)$. □

The proof is based on the selection, for each internal node u of a "first" and a "second" son. Then u is represented by the unique leaf x such that the sequence $x = a_p, a_{p-1}, ..., a_1, u$ is an ascending path in T, where x is the second son of a_{p-1}, a_{i+1} is the second son of a_i for $i := 1, ..., p-2$ and a_1 is the first son of u. This method is used in [2, 5] where an auxiliary linear order on leaves helps to define the first and second sons of each internal node. Here, we will uses numbers in $\{0, 1, 2\}$ to label leaves and counting modulo 3 cardinality set predicates.

We present the necessary technical definitions and lemmas.

Definition 3.5. *Weights.*

Let (L, R) be the leaf structure of an O-tree T. We call $\sigma : L \longrightarrow \{0, 1, 2\}$ a *weight* mapping. We define the *weight* of a finite set $Y \subseteq L$ as follows: $\sigma(Y)$ is the sum modulo 3 of the weights $\sigma(x)$ of all x in Y. We extend σ to internal nodes u by letting $\sigma(u) := \sigma(Lf_T(u))$.

Assume now that T is finite. We say that σ is *good* if every internal node u has a unique son having a maximal weight relative to the order $0 < 1 < 2$. This son will be called the *σ-son* of u (the first one in our informal presentation).

Lemma 3.6. *(Claim B5 of [3]). For every finite leafy rooted tree T, for every i in $\{0, 1, 2\}$ and every choice of a "preferred" son for each internal node, there is a weight $\sigma : L_T \longrightarrow \{0, 1, 2\}$ such that $\sigma(rt_T) = i$ and the σ-sons of the internal nodes are the prefered ones.*

Proof. By induction on the size of T. There is nothing to do if r_T is a leaf. Otherwise, let $b_1, ..., b_p, p \geq 2$ be the sons of r_T and b_1 be its preferred son. Then:

if $\sigma(rt_T) = 0$, we let $\sigma(b_1) := 2$, $\sigma(b_2) := 1$ and $\sigma(b_i) := 0$ for $i \geq 3$,
if $\sigma(rt_T) = 1$, we let $\sigma(b_1) := 1$ and $\sigma(b_i) := 0$ for $i \geq 2$,
if $\sigma(rt_T) = 2$, we let $\sigma(b_1) := 2$ and $\sigma(b_i) := 0$ for $i \geq 2$.

By the induction hypothesis, σ can be defined appropriately on the subtrees T_{b_i} for $i \geq 1$. □

Proof of Theorem 19. Let (L, R) be the leaf structure a finite leafy rooted tree $T = (N, \leq)$ (not reduced to a single node). Hence, $|L| \geq 2$ and $|N| \geq 3$.

For $i = 1, 2$, let $C_{i,3}(U)$ mean that the set U is finite and $|U| = i$ mod 3. It is easy to express $C_{1,3}$ and $C_{2,3}$ in terms[15] of C_3.

Let X_1 and X_2 be set variables intended to define the leaves x such that, respectively, $\sigma(x) = 1$ and $\sigma(x) = 2$. The leaves outside of these two sets will have by σ the value 0.

If $U \subseteq L$, and σ is given by disjoint subsets X_1 and X_2 of L, we have:

$$\sigma(U) = i \text{ if and only if } |U \cap X_1| + 2.\, |U \cap X_2| = i \text{ mod } 3,$$

hence, this property is CMSO-expressible by means of the set predicates $C_{1,3}$ and $C_{2,3}$.

It follows that for every internal node u of T defined as $x \sqcup_T y$, CMSO formulas $\alpha_i(X_1, X_2, x, y)$ can check whether $\sigma(u) = i$ where σ is defined by the disjoint sets X_1, X_2 because the set $Lf_T(x \sqcup y)$ is FO-definable in (L, R) from x and y. Hence, there is a CMSO formula $\beta(X_1, X_2, x, y, z, u)$ expressing in (L, R) that σ defined

[15] In a possibly infinite structure, the finiteness of a set U is expressed by $C_3(U) \vee C_{1,3}(U) \vee C_{2,3}(U)$. A similar expression is possible for each C_n.

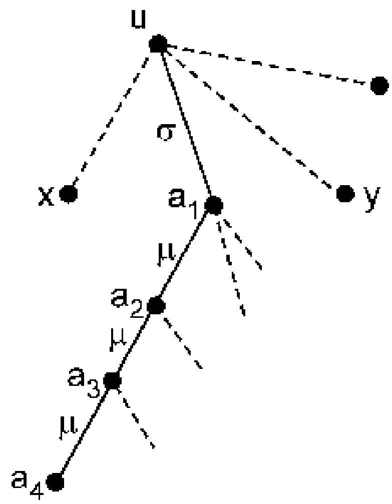

FIGURE 4. See the proof of Theorem 3.4.

in T by X_1, X_2 is good and that $x \sqcup_T y$ is the σ-son of $z \sqcup_T u$. We allow $x = y = x \sqcup_T y$. A CMSO-formula $\overline{\beta}(X_1, X_2)$ can express that this mapping σ is good.

We need a second good weight mapping μ, defined similarly by means of sets of leaves X_3 and X_4. Furthermore, we want that each μ-son of an internal node differs from the σ-son. A CMSO formula $\gamma(X_1, X_2, X_3, X_4)$ can express these conditions. Clearly, $\beta(X_3, X_4, x, y, z, u)$ expresses in (L, R) that μ defined by X_3, X_4 is good and that $x \sqcup_T y$ is the μ-son of $z \sqcup_T u$.

Let $T = \tau(L, R) = (N, \leq)$. We are ready to define a leaf $Rep(x \sqcup_T y)$ below any internal node $x \sqcup_T y$, intended to represent it, and, as a consequence, the subtree $T_{x \sqcup y}$ whose set of leaves is $Lf_T(x \sqcup_T y)$.

Given x and $y \neq x$, we let $x \sqcup_T y, a_1, ..., a_p$ be the unique sequence of nodes of T such that a_1 is the σ-son of $x \sqcup_T y$, a_i is the μ-son of a_{i-1} for $i \geq 2$, until we reach a leaf a_p that we define as $Rep(x \sqcup_T y)$. The leaf a_p is uniquely defined from $x \sqcup_T y$. We have $Rep(x \sqcup_T y) = a_1$ if a_1 is a leaf.

Figure 4 illustrates this definition with $p = 4$. Dotted lines represent finite paths.

Claim 3.7. *If $Rep(x \sqcup_T y) = Rep(x' \sqcup_T y')$, then $x \sqcup_T y = x' \sqcup_T y'$.*

Proof. Given a_p, the sequence $a_{p-1}, ..., a_1, x \sqcup_T y$ is uniquely defined as each element is the father of the previous one. Its stops as soon as one finds a_1 that is a σ-son, and its father is $x \sqcup_T y$. □

Claim 3.8. *(cf. Theorem 5.3 in [5]): a CMSO formula $\rho(X_1, X_2, X_3, X_4, x, y, z)$ can express that $z = Rep(x \sqcup_T y)$.*

Proof. We have $z = Rep(x \sqcup_T y)$ if and only if there exists a set $\{v_1, ..., v_{p-1}\} \subseteq L$ such that $z \sqcup_T v_1$ is the σ-son of $x \sqcup_T y$, and, for each $i = 2, ..., p-1$ the node $z \sqcup_T v_i$ is the μ-son of $z \sqcup_T v_{i-1}$, and z is the μ-son of $z \sqcup_T u_{p-1}$. Given a set U intended to be such $\{v_1, ..., v_{p-1}\}$, this condition can be expressed by a CMSO formula, because CMSO logic can express transitive closure. On Figure 4, we have $z = a_4$.

Hence, we obtain the tree $\tau(L, R)$ with set of leaves $L \times \{1\}$ and set of internal nodes $Z \times \{2\}$ where Z is the set of leaves that represent some internal node. □

Another CMSO-transduction proving Theorem 3.4 is defined in [4]: it uses the set predicate C_2 instead of C_3, and the copying constant is 5. That article develops applications to several canonical graph decompositions shaped as trees: the modular decomposition, the split decomposition and the bi-join decomposition among others.

The proof of Theorem 5.3 of [5] uses a auxiliary linear order on leaves in order to distinguish two sons of each internal node, used as shown in Figure 4. The present proof uses instead sets $X_1, ..., X_4$ from which, however, one cannot define a linear order of L.

3.4. CMSO-constructible classes

We will construct subclasses of $\mathcal{T}$ for which some analog of Theorem 3.4 holds with CMSO-transductions having possibly more parameters and larger copying constants. Notation is as in Section 3.1.

Definition 3.9. *CMSO-constructible classes of leafy join-trees.*
(1) A subclass $\mathcal{C}$ of $\mathcal{T}$ is *CMSO-constructible* if

(a) the corresponding class of leaf structures is CMSO-definable by some sentence $\overline{\chi}$ and
(b) there is a CMSO-transduction $\widehat{\Delta}$ that transforms each ternary structure (L, R) satisfying $\overline{\chi}$ into the corresponding join-tree $\tau(L, R)$, *u.t.i.*

(2) A join-tree $T = (N, \leq)$ is *downwards-finite* if its sub-join-trees T_x (*cf.* Definition 2.1(b)) are finite, for all x in N.
(3) We let $\mathcal{D}$ be the subclass of $\mathcal{T}$ consisting of the finite rooted trees and of the downwards-finite, leafy join-trees without root (they are necessarily infinite)[16].

Remark 3.10. (1) The wording "CMSO-constructible" hides several aspects of the definition, but it is chosen for shortness sake.
(2) Each internal node of a join-tree in $\mathcal{D}$ has at least two sons.
(3) An example of a join-tree in $\mathcal{D}$ is[17] $(\mathbb{N} \times \{0, 1\} - \{(0, 0)\}, \leq)$ whose partial order $\leq$ is defined by: $(i, j) \leq (k, l)$ if and only if $(i, j) = (k, l)$ or $i \leq k$ and $l = 1$. Its leaves are the nodes $(0, 1)$ and $(i, 0)$ for all $i \geq 1$.
(4) The class $\mathcal{H}_k$ is CMSO-constructible (it is even FO-constructible) by Proposition 3.3.

Lemma 3.11. *An infinite leafy join-tree T is in $\mathcal{D}$ if and only if:*
(1) for each leaf x and internal node $y < x$, the interval $[y, x]_T$ is finite, and T consists of an infinite ascending branch $x = a_0, a_1, a_2, ..., a_n, ...$ such that for each $i > 0$, the node a_i has finitely many sons $a_{i-1}, b_1, ..., b_p$, where $p \geq 1$, and the subtrees T_{a_i} are finite, if and only if
(2) the same holds for some leaf x.

Proof. Clear from the definitions. We have $p > 0$ because T is leafy. Each subtree T_{b_j} is finite. □

Theorem 3.12. *The class $\mathcal{D}$ is CMSO-constructible by means of the transduction used for Theorem 3.4.*

Proof. The leaf structures (L, R) of the join-trees in $\mathcal{D}$ are those that satisfy A1–A5 of Proposition 2.5 and the following CMSO expressible property (that holds trivially if L is finite):

for all $x, y \in L$, the set $\{z \in L \mid Rzxy\}$ is finite[18].

For the case where L is infinite, we need only prove the existence of two weight functions σ and μ as in Theorem 3.4. Let T be as in Lemma 3.11. We define $\sigma(a_i) := 2$ for each a_i and $\sigma(b_j) := 0$ for all its sons b_j. The values of σ on the nodes of the subtrees T_{b_j} exist by Lemma 3.6 as these trees are finite. Hence, the σ-son of a_i is a_{i-1}.

We define $\mu(a_i) := 0$ for each a_i such that i is even and 1 if i is odd. For each i, we choose one of the sons of a_i, say b_1, to be the μ-son and we define $\mu(b_1) := 1$ if i is odd and $\mu(b_1) := 2$ if i is even. All other sons of a_i will have by μ the weight 0. On each subtree T_{b_j} we choose μ-sons to be different from the already chosen

[16] Actually, a downwards-finite, infinite join-tree cannot have a root.
[17] An infinite ascending comb without root. It is an infinite *caterpilar*.
[18] The finiteness of a set U is expressed by $C_3(U) \vee C_{1,3}(U) \vee C_{2,3}(U)$. A similar expression can be based on C_n for any $n \geq 2$.

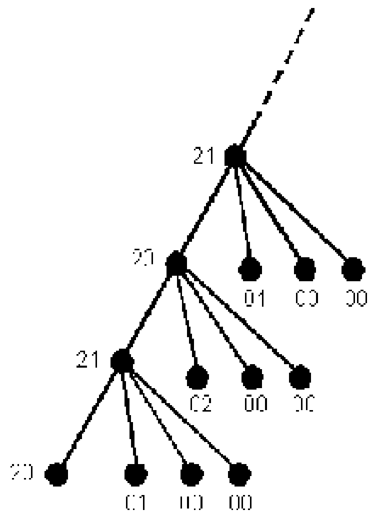

FIGURE 5. The two good weight functions in the proof of Theorem 3.12.

σ-sons. This is possible because each internal node has at least two sons. Weights for μ can be computed on these subtrees by Lemma 3.6 again. Then the proof of Theorem 3.4 works. □

Figure 5 illustrates this construction. The pair ij at a node x indicates that $\sigma(x) = i$ and $\mu(x) = j$.

The CMSO-transduction defined in [4] extends to the same class $\mathcal{D}$ as that of [3]. Our next aim is to construct larger CMSO-constructible classes of join-trees by combining several known ones. We use the transductions of [3] and, equivalently, of [4], as building blocks of more complex transductions.

Theorem 3.13. *If $\mathcal{C}$ and $\mathcal{C}'$ are CMSO-constructible classes, then, so is their union.*

Proof. Let Δ be a definition scheme for $\mathcal{C}$ whose parameters are $X_1, ..., X_p$ and copying constant is k, and Δ' be similar for $\mathcal{C}'$ with parameters $Y_1, ..., Y_q$ and copying constant k'. One can construct a definition scheme Θ whose parameters are $Z, X_1, ..., X_p, Y_1, ..., Y_q$ and copying constant is $\max(k, k')$. The extra parameter Z decides whether the leaf structure under consideration should define a join-tree in $\mathcal{C}$ (if Z is empty) or in $\mathcal{C}'$ (if Z is not empty). Depending on the case, Θ uses Δ with its parameters $X_1, ..., X_p$ or Δ' with its parameters $Y_1, ..., Y_q$. We omit the routine details. □

Every rooted O-tree T is of the form $*(T_1, ..., T_m, ...)$ consisting of a (possibly infinite) disjoint union of O-trees $T_1, ..., T_m, ...$, together with a new root above all of them (denoted by $*$). Then T is a leafy join-tree if and only $T_1, ..., T_m, ...$ are so and their (unordered) list has at least two O-trees. We exclude the trivial case where T is reduced to a root. However, some of the subtrees T_i may be reduced to a root that is thus a leaf of T. If all T_i are single nodes, then T is called a *star*.

If $\mathcal{C} \subseteq \mathcal{T}$, we let $\mathcal{C}^+ \subseteq \mathcal{T}$ be the class of join-trees $T = *(T_1, ..., T_m, ...)$ such that each T_i is in $\mathcal{C}$ or is reduced to a root. Hence, such T is leafy and belongs to $\mathcal{T}$.

Theorem 3.14. *If a class $\mathcal{C}$ is CMSO-constructible, then, so is $\mathcal{C}^+$.*

The proof will use *relativizations* of CMSO sentences to CMSO-definable sets (actually the sets of leaves of the T_i's). This is a standard notion in logic[19] that we review.

[19] See *e.g.*, [19], Section 5.2.1.

Let $\varphi(X_1,...,X_n,x_1,...,x_p)$ be a CMSO formula relative to relational structures $S=(D,R,...)$ and let Y be a set variable. One can construct a CMSO formula $\varphi[Y](X_1,...,X_n,x_1,...,x_p)$ with free variables $Y,X_1,...,X_n,x_1,...,x_p$ such that, for every structure S as above, for all subsets $Y,X_1,...,X_n$ of D and all $x_1,...,x_p\in D$ we have:

$S\models\varphi[Y](X_1,...,X_n,x_1,...,x_p)$ if and only if
$x_1,...,x_p\in Y$ and $S[Y]\models\varphi(X_1\cap Y,...,X_n\cap Y,x_1,...,x_p)$
where $S[Y]$ is the substructure of S induced on Y (*cf.* Section 2.2).

The construction consists replacing an atomic formula $C_m(X_i)$ by $C_m(X_i\cap Y)$, a quantification $\exists x....$ by $\exists x.x\in Y\wedge...$ and a set quantification $\exists X....$ by $\exists X.X\subseteq Y\wedge...$. We omit the details.

Proof. By using Theorem 2.10 and Remark 2.12(2), we can write an FO-sentence α expressing that a ternary structure (L,R) is the leaf structure of a rooted join-tree T in $\mathcal{T}$. Assuming this, $T=*(T_1,....,T_m,...)$ where $T_1,....,T_m,...$ belong to $\mathcal{T}$, have pairwise disjoint sets of nodes $N_1,....,N_m,...$ and sets of leaves $L_1,....,L_m,...$, so that $L_T=L_1\cup....\cup L_m\cup...$ They may have no root. Each relation R_{T_i} is the restriction of R_T to L_i. As T is leafy, its root is the join of two leaves that belong to distinct subtrees T_i.

Next, we express in (L,R) that the T_i's are in $\mathcal{C}$. A key point is to identify by CMSO formulas their sets of leaves L_i. For x and $y\in L$, we define $x\sim y:\Longleftrightarrow\exists z.\neg Rzxy$.

Claim 3.15. *The relation $\sim$ is an equivalence relation and its classes are the sets L_i. Furthermore, $y\sim z\wedge Rxyz\Longrightarrow x\sim y$.*

Proof. We need only prove transitivity of $\sim$. Assume $x\sim y\wedge y\sim z$. We have $\neg Ruxy\wedge\neg Rvzy$ for some u,v. We claim that $\neg Ruxz\vee\neg Rvxz$ holds, proving $x\sim z$. We prove this from A1–A5 and their consequences A6 and A7. A7 yields $Rxyz\vee Rzxy$. Assume the first. If $Rvxz$ holds, then together with[20] $Rxzy$, we get $Rvzy$ by A6, contradicting the hypothesis. Assume now $Rzxy$. If $Ruxz$ holds, then $Ruxy$ holds by A6, again a contradiction. Hence we have $\neg Ruxz\vee\neg Rvxz$ and so $x\sim z$.

The condition $\exists z.\neg Rzxy$ guarantees that $x\sqcup_T y$ is not the root of T, hence that x and y must belong to a same set L_i.

Assume now $y\sim z\wedge Rxyz$. Hence, $\neg Ruyz$ holds for some u. If we have $Ruxy$, then with $Rxyz$ and by A6, we have $Ruyz$, contradicting the choice of u. Hence, we have $x\sim y$ and R is stable under $\sim$. ☐

For every equivalence class M of $\sim$, the ternary structure $(M,R[M])$ is the leaf structure of some leafy join-tree that we will denote by T_M. These join-trees T_M are pairwise disjoint, otherwise, we would have two classes $M\neq M'$ and $x,y\in M$, $z,u\in M'$ such that $x\sqcup y=z\sqcup u$ in T, hence $Rzxy$ so that $z\in M$ by Claim 3.15, hence a contradiction. They are the join-trees $T_1,...,T_m,...$

We let $\Delta=(\chi,\delta_1,...,\delta_k,(\theta_{i,j})_{i,j\in[k]})$ be a definition scheme for $\mathcal{C}$, with associated CMSO sentence $\overline{\chi}$, expressing that $\widehat{\Delta}(L,R)$ is defined.

Claim 3.16. *A CMSO sentence μ can express in (L,R) that each T_M is in $\mathcal{C}$.*

Proof. An FO formula $\eta(Y)$ can express in (L,R) that a set $Y\subseteq L$ is an equivalence class of $\sim$. Then, a CMSO sentence can express that for each equivalence class M, we have $(M,R[M])\models\overline{\chi}$. ☐

Hence, together with α, we have a CMSO sentence expressing that (L,R) is the leaf structure of some leafy joint-tree T in $\mathcal{C}^+$.

In order to construct T, we construct $\Theta=(\chi',\delta'_0,...,\delta'_k,(\theta'_{i,j})_{i,j\in[0,k]}))$ defining T such that $N_T=\{u\}\times\{0\}\cup U_1\times\{1\}\cup...\cup U_k\times\{k\}$ where u is an arbitrary leaf and $(u,0)$ will be the root of T. We will use the parameters $X_1,...,X_p$ from Δ and an additional one X_0, intended to hold u.

To be convenient, a p-tuple $(X_1,...,X_p)$ must verify:

[20] In such proofs, we leave implicit the use of A2.

for each equivalence class M, we have $(M, R[M]) \models \chi(X_{M,1}, ..., X_{M,p})$ where $X_{M,i} := X_i \cap M$ for each i.

From such $(X_1, ..., X_p)$, we define, for each M and $i \in [k]$:

$U_{M,i} := \{x \in M \mid (M, R[M]) \models \delta_i(X_{M,1}, ..., X_{M,p}, x)\}$.

It follows that $N_{T_M} = U_{M,1} \times \{1\} \cup ... \cup U_{M,k} \times \{k\}$. To obtain N_T as desired, we define U_i as the union of the sets $U_{M,i}$ for all M.

Finally, we define $\leq_T$ as follows, for $(x,i) \leq (y,j)$ in N_T: $(x,i) \leq (y,j)$ if and only if:

either $(y,j) = (u,0)$,
or x and y are in some M such that $(M, R[M]) \models \theta_{i,j}(X_{M,1}, ..., X_{M,p}, x, y)$.

From this description, one can (one could) write the formulas $\chi', \delta'_0, ..., \delta'_k, \theta'_{i,j}$ to form Θ. □

This proof is actually a useful preparation for a similar more complicated construction.

Definition 3.17. *Substitutions to leaves.*

(1) Let $T = (N_T, \leq_T)$ be an O-tree. For each $x \in L_T$, let $U_x = (N_x, \leq_x)$ be an O-tree (it may be reduced to a single node). Since our constructions yield objects *u.t.i.*, we can take them pairwise disjoint and disjoint from T. We let $S = T[U_x/x; x \in L_T]$ be the partial order $(N_S, \leq_S)$ such that:
N_S is the union of $N_T - L_T$ and the sets N_x,
and $u \leq_S v$ if and only if

either $u, v \in N_T - L_T$ and $u \leq_T v$,
or $u, v \in N_x$, $x \in L_T$, and $u \leq_x v$,
or $u \in N_x$, $x \in L_T$, $v \in N_T - L_T$ and $x \leq_T v$.

(2) A small extension of this definition will be useful: we can assume that each $x \in L_T$ is also a leaf of U_x. There is nothing to change in the definitions (in the second case above, we may have $u = x$). In this case $L_T \subseteq N_S$ as each $x \in L_T$ belongs to N_x. In particular, if $U_x = \{x\}$, where $x \in L_T$, there is nothing to substitute to x.

If T is $*(x, y, z)$, then $T[U_x/x, U_y/y, U_z/z] = *(U_x, U_y, U_z)$. Figure 6 illustrates Definition 3.17(2). It shows two leaves x and y (among others) of an O-tree T, an O-tree U_x having x as a leaf and, to the right, the O-tree $T[U_x/x, U_y/y]$ where $U_y = y$. (We omit the other leaves of T).

With these hypotheses and notation:

Lemma 3.18. *(1) $S = (N_S, \leq_S)$ is an O-tree.*
(2) If T and the U_x's are leafy join-trees, then S is a leafy join-tree.

Proof. (1) is straightforward.
(2) From the definitions, we observe that the joins in S are as follows:

if $u, v \in N - L_T$ then $u \sqcup_S v = u \sqcup_T v$,
if $u, v \in N_x$, $x \in L_T$, then $u \sqcup_S v = u \sqcup_x v$,
if $u \in N_x$, $x \in L_T$, $v \in N_T - L_T$ then $u \sqcup_S v = x \sqcup_T v$,
if $u \in N_x$, $v \in N_y, x, y \in L_T$ and $y \neq x$ then $u \sqcup_S v = x \sqcup_T y$.

Hence, S is a join-tree. It is leafy by the above observations. □

Definition 3.19. *Contracting subtrees.*

(1) Let $S = (N_S, \leq_S)$ be a leafy join-tree. A node x is *finite* if the sub-join-tree S_x is finite. Since S is leafy, a subtree S_x is finite if it has finitely many leaves.

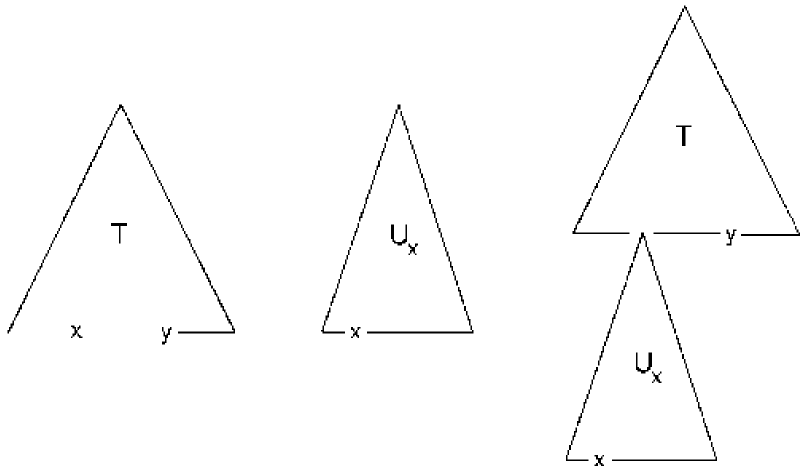

FIGURE 6. Illustration for Definition 3.17(2).

If z is a leaf, then $Fin(z) \subseteq N_{\geq}(z)$ denotes the set of finite nodes above z or equal to it. Either $Fin(z)$ has a maximal element y, or there is an infinite ascending branch $z = a_0, a_1, a_2, ..., a_n, ...$ of finite nodes where each a_i, for $i > 0$, has a finite set of sons $\{a_{i-1}, b_1, ..., b_p\}$ with $p > 0$, as in Lemma 3.11.

In the first case, we let $U_z := S_y$. It is a finite leafy tree, possibly reduced to z; its root is y. In the second, we define U_z is the union of the subtrees S_{a_i}. In both cases, $U_z \in \mathcal{D}$.

(2) We may have $U_z = U_{z'}$ for distinct leaves z, z'. We let Z contain one and only one leaf z for each U_z.

(3) In $S = (N_S, \leq_S)$, we "contract" as follows the trees U_z:

for all $z \in Z$, we delete from S the nodes of U_z, except z.

We obtain a subset N_T of N_S and $T := (N_T, \leq_T)$ where $\leq_T$ is the restriction of $\leq_S$ to N_T.

With these definitions and notation:

Lemma 3.20. *If $S = (N_S, \leq_S)$ is a leafy join-tree then T is also a leafy join-tree. We have $L_T = Z$. The join mapping $\sqcup_T$ is the restriction of $\sqcup_S$ to N_T. Furthermore $S = T[U_z/z; z \in Z]$.*

Proof. These facts are clear. We use here the extension (2) of the basic definition of substitution. We have an equality $S = T[U_z/z; z \in Z]$, not only an isomorphism. □

We denote T by $Contr(S)$. If $S \in \mathcal{D}$, then $Contr(S)$ is a single node. If $Contr(S)$ is a star (*cf.* Theorem 27), then $S \in \mathcal{D}^+$. We denote by $\mathcal{C}[\mathcal{D}]$ the class of join-trees S in $\mathcal{T}$ such that $Contr(S) \in \mathcal{C}$.

Theorem 3.21. *If a subclass $\mathcal{C}$ of $\mathcal{T}$ is CMSO-constructible, then so is the class $\mathcal{C}[\mathcal{D}]$.*

Proof. We will use the definition scheme $\Delta = (\chi, \delta_1, \delta_2, (\theta_{i,j})_{i,j \in [2]})$ with parameters $X_1 X_2, X_3, X_4$ for the class $\mathcal{D}$ defined in Theorem 3.12. We will combine it with a definition scheme for $\mathcal{C}$ of the form $\Delta^{\mathcal{C}} = (\chi^{\mathcal{C}}, \delta_1^{\mathcal{C}}, ..., \delta_k^{\mathcal{C}}, (\theta_{i,j}^{\mathcal{C}})_{i,j \in [k]})$ with parameters $Y_1, ..., Y_p$ and copying constant k, and get a definition scheme $\Theta = (\chi', \delta'_0, ..., \delta'_k, (\theta'_{i,j})_{i,j \in [0,k]})$ for $\mathcal{C}[\mathcal{D}]$ whose parameters are $X_1 X_2, X_3, X_4, Y_1, ..., Y_p, Z$ and copying constant is $k + 1$. □

We consider a join-tree S given by its leaf structure (L, R). We use notions from Definition 3.19(2). Our first objective is to identify the sub-join-trees U_z of S to be contracted.

Claim 3.22. *CMSO formulas can express in (L, R), for all x, y and z in$L$, that $x \sqcup_S y$ is a finite node and that $x$is a leaf of U_z.*

Proof. Clear because $x \sqcup_S y$ is finite if and only if the set $\{z \in L \mid Rzxy\}$ is finite. Furthermore x is a leaf of U_z if and only if $x \sqcup_S z$ is finite. □

Hence, we obtain the leaf structure $(L_{U_z}, R[L_{U_z}])$ of U_z. The leaf structure of T is then $(Z, R[Z])$, where Z is specified in Definition 3.19(2).

We first express that $S \in \mathcal{C}[\mathcal{D}]$ and we also specify the appropriate parameters.

A formula $\chi'(X_1X_2, X_3, X_4, Y_1, ..., Y_p, Z)$ can express in (L, R) the following conditions:

(i) Properties A1–A5 hold, so that (L, R) is the leaf structure of a join-tree S in $\mathcal{T}$,
(ii) the parameter Z is as required in Definition 3.19(2),
(iii) for each $z \in Z$, $(L, R) \models \chi[L_{U_z}](X_1X_2, X_3, X_4)$,
(iv) $Y_1, ..., Y_p \subseteq Z$ and
(v) $(L, R) \models \chi^{\mathcal{C}}[Z](Y_1, ..., Y_p)$, which implies that $Contr(S) \in \mathcal{C}$.

Next, we describe N_S in terms of these parameters. Without loss of generality, we assume that $\Delta^{\mathcal{C}}$ is built so that it produces from any appropriate leaf structure (L', R') a join-tree whose set of nodes is

$L' \times \{1\} \cup D_2 \times \{2\} \cup ... \cup D_k \times \{k\}$,

its set of leaves being $L' \times \{1\}$.

The set of nodes of the join-tree $\widehat{\Theta}(L, R)$ (isomorphic to S) is defined as
$L \times \{1\} \cup I \times \{0\} \cup D_2 \times \{2\} \cup ... \cup D_k \times \{k\}$ where:
$L \times \{1\}$ is the set of leaves of S, hence the union of the pairwise disjoint sets of leaves of the U_z's,
$I \times \{0\}$ is the union of the sets of internal nodes of the pairwise disjoint U_z's,
$N_{Contr(S)} = Z \times \{1\} \cup D_2 \times \{2\} \cup ... \cup D_k \times \{k\}$,
$Z \times \{1\}$ is the set of leaves of $Contr(S)$.

The sets $D_2 \times \{2\} \cup ... \cup D_k \times \{k\}$ are defined by $\Delta^{\mathcal{C}}$ applied to $(Z, R[Z])$, that is the leaf structure of $Contr(S)$.

The set of nodes of each U_z is $L_{U_z} \times \{1\} \cup I_{U_z} \times \{2\}$ where I_{U_z} is defined from parameters $X_1 \cap L_{U_z}, ..., X_4 \cap L_{U_z} \subseteq L_{U_z}$ by Δ applied to $(L_{U_z}, R[L_{U_z}])$. From this description, we can (we could) write the formulas $\delta'_0, ..., \delta'_k$.

It remains to construct the formulas $\theta'_{i,j}$ intended to formalize the order relation $\leq$ of S. We describe it as follows in terms of the associated strict partial order:

$x < y$ if and only if,
either $x, y \in N_{U_z}$ for some $z \in Z$ and $x <_{U_z} y$,
or $x \in N_{U_z}$ for some $z \in Z$, y is internal in $Contr(S)$ and $z <_{Contr(S)} y$,
or $x, y \in N_{Contr(S)}$ and $x <_{Contr(S)} y$.

We omit an explicit writing of the formulas.

Remark 3.23. If $\mathcal{C}$ and $\mathcal{E}$ are subclasses of $\mathcal{T}$, one can defined $\mathcal{C}[\mathcal{E}]$ as the class of trees $T[U_z/z; z \in L_T]$ such that T is in $\mathcal{C}$ and each U_z is in $\mathcal{E}$. Assuming that $\mathcal{C}$ and $\mathcal{E}$ are CMSO-constructible, we would like to prove that $\mathcal{C}[\mathcal{E}]$ is so by a proof similar to that of Theorem 3.21. Given the leaf structure (L, R) of some S potentialy in $\mathcal{C}[\mathcal{E}]$, we would need a CMSO identification of the subtrees U_z of S. We had it for the proofs of Theorems 3.14 and 3.21.

Our next construction is based on identifying subtrees of O-trees.

Definition 3.24. *Subtrees of O-trees*

(1) We recall from Definition 2.1 that in an O-tree $T = (N, \leq)$, the covering relation $x <_c y$ means that x is a son of y. We let $\approx$ be the equivalence relation generated by $<_c$. We have $x \approx y$ if and only if there is z such that $x <^*_c z$ and $y <^*_c z$ where * denotes the reflexive and transitive closure of $<_c$. If A is a subset of N such that any two elements are $\approx$-equivalent, then $T[A] := (A, \leq_A)$ where $\leq_A$ is the restriction of $\leq$ to A is a *subtree* of T (a "real tree", not a sub-O-tree). It may have no root and/or no leaf.

(2) An equivalence class C is called a *connected component*, and so is called the subtree $T[C]$ of T that it induces. A connected O-tree, having a single component, is thus a tree. A connected component may be reduced to a single node.

(3) We let $T/_\approx$ be the quotient structure $(N/_\approx, \leq_\approx)$ where $N/_\approx$ is the set of connected components and $C \leq_\approx D$ if and only if $x <_c y$ for some $x \in C$ and $y \in D$. It is an "O-tree of trees".

Proposition 3.25. *If T is an O-tree, then $T/_\approx$ and its connected components are O-trees. If it is a join-tree, so are $T/_\approx$ and its connected components.*

The proof is easy. The following examples show that $T/_\approx$ and its connected components may have no leaves, even if T is a leafy join-tree.

Example 3.26. Let $A = (N_A, \leq_A)$ be a leafy connected join-tree without root. For example $N_A = \mathbb{N} \times \{0,1\} - \{(0,1)\}$ and $(i,0) \leq_A (j,0)$ and $(i,1) <_A (j,0)$ where $i \leq j$. Let a be a fixed internal node.

1) We define a leafy join-tree T with set of nodes $N_A \times \mathbb{N}$ ordered as follows, for all i,j in $\mathbb{N}$ and $x, y \in N_A$:

$(x,i) \leq (y,i)$ if and only if $x \leq_A y$
$(x,i) < (y,j)$ where $i \neq j$ holds if and only if $j < i$ and $a \leq_A y$.

Each $N_A \times \{i\}$ is a connected component. The quotient tree$T/_\approx \simeq (\mathbb{N}, \geq)$ has root 0 and no leaf. Each node has a son, and $T/_\approx$ has a unique connected component.

2) We now define the leafy join-tree $S := (N_S, \leq_S)$ where $N_S := \mathbb{N} \cup (N_A \times \mathbb{N})$ and the following partial order, for all i,j in $\mathbb{N}$ and $x, y \in N_A$:

$i \leq_S j$ if and only if $j \leq i$,
$(x,i) <_S j$ if and only if $j \leq i$,
$(x,i) <_S (y,i)$ if and only if $x \leq_A y$.

Its connected components are $\mathbb{N}$ and the sets $N_A \times \{i\}$. The subtree $S[\mathbb{N}] = (N_S, \leq_S)$ has no leaf.

Theorem 3.27. *Let T be a leafy join-tree such that*
(1) its connected components are all in a CMSO-constructible class $\mathcal{C}$, and
(2) their leaves are leaves of T. Then T is CMSO-constructible.

Proof. Similar to the previous proofs. $\square$

By Theorem 3.13, the class $\mathcal{C}$ may be the union of finitely many CMSO-constructible classes.

Open problem: Exhibit a join-tree in $\mathcal{T}$ that provably cannot be constructed from its leaf structure by any CMSO-transduction.

Definition 3.28. *Alternative definition of leaf structures.*

Leafy finite rooted trees can also be described as families of nonoverlapping sets[21]. Let $\mathcal{W} := (L, \mathcal{M})$ where L is a finite set and $\mathcal{M}$ is a set of subsets of L satisfying the following conditions:

$L \in \mathcal{M}$ and $\{x\} \in \mathcal{M}$ for each $x \in L$,
if $X, Y \in \mathcal{M}$ and $X \cap Y$ is not empty, then $X \subseteq Y$ or $Y \subseteq X$ (the sets in $\mathcal{M}$ are not *overlapping*).

Then $(\mathcal{M}, \subseteq)$ is a finite leafy rooted tree $T(\mathcal{W})$ with root L and set of leaves the singletons $\{x\}$ for $x \in L$. Conversely, every finite leafy rooted tree $T = (N_T, \leq)$ yields a pair $\mathcal{W} = (L_T, \mathcal{M})$ where $\mathcal{M} := \{N_{\leq x} \cap L_T \mid x \in N_T\}$ and $T \simeq T(\mathcal{W})$.

We make $\mathcal{W}$ into the logical structure $(L, M(.))$ where $M(.)$ is the set predicate such that $M(X)$ holds if and only if $X \in \mathcal{M}$. In some cases, this set predicate can be defined by an MSO formula over a relational structure, for example (L, edg) if $\mathcal{M}$ is the set of *strong modules* of a graph G defined by this structure where $L = V_G$, see [2, 4]; in this case, $T(L, \mathcal{M})$ is the *modular decomposition* of G

For MSO logic, the descriptions of $T = T(\mathcal{W})$ by (L_T, R_T) and by $(L_T, M(.))$ are equivalent: R_T can be defined as the set of triples $(x, y, z) \in L_T \times L_T \times L_T$ such that $x \in X$ where X is the inclusion-minimal set

[21] This description is used in [3, 4].

such that $M(X)$ holds and that contains y and z. Conversely, a subset X of L_T is in $\mathcal{M}$ if and only if it is $\{x \in L_T \mid R_T(x,y,z) \text{ holds}\}$ for some $y,z \in L_T$.

It follows that every CMSO formula $\varphi(X_1,...,X_p)$ over the leaf structure (L_T, R_T) of a join-tree can be rewritten into an equivalent one $\psi(X_1,...,X_p)$ over the corresponding "set" structure $(L_T, M(.))$, and vice-versa. This facts extend to infinite join-trees (we omit the details). The constructions of CMSO transductions based on structures (L,R) and $(L,M(.))$ are thus equivalent, as they are convertible into one another.

4. Quasi-trees and their substructures

Quasi-trees can be considered as undirected join-trees. If they are used as layouts (see below), the rank-width of a countable graph is the least upper-bound of the rank-widths of its finite induced subgraphs. They have been further studied in [11, 14, 15].

The ancestor relation of a join-tree is no longer well-defined in the corresponding undirected structure. We replace it by a ternary relation called betweenness.

Definition 4.1. [6, 11, 15]: *Betweenness in trees*

The notation $\neq xyz...$ means that $x, y, z...$ are pairwise distinct.

(a) A tree T is an undirected connected graph without loops, parallel edges and cycles. It has no root, that is, no distinguished node. It may be (countably) infinite. It is defined as $T = (N, edg)$ where N is the set of nodes and *edg* is the binary edge relation. Any two nodes are linked by a path (by connectedness) that is actually unique.

The *degree* of a node is the number of incident edges. A *leaf* is a node of degree 1. A tree is *subcubic* if each node has degree at most 3.

(b) The *betweenness* of a tree T is the ternary relation B such that $Bxyz$ holds if and only if $\neq xyz$ and y is on the path between x and z.

(c) The betweenness of a rooted tree is that of the associated unrooted undirected tree.

For shortening formulas relative to ternary structures (N,B), we will denote $Bxyz \vee Byxz \vee Bxzy$ by $Axyz$ to mean that x, y and z are *aligned* on a same "path". Furthemore $B^+x_1...x_n$, for $n \geq 4$, will stand for the conjunction

$$Bx_1x_2x_3 \wedge Bx_2x_3x_4 \wedge ... \wedge Bx_{n-2}x_{n-1}x_n$$

and $[x,y]$ will denote the set $\{x,y\} \cup \{u \in N \mid Bxuy\}$, called an *interval* or a "path".

Betweenness in partial orders has been studied in [10] and in previous articles cited there.

Proposition 4.2. *'(Proposition 9 of [6]): For every tree T, the following properties hold for all nodes x, y, z and u:*

B1: $Bxyz \Longrightarrow \neq xyz$,
B2: $Bxyz \Longrightarrow Bzyx$,
B3: $Bxyz \Longrightarrow \neg Bxzy$,
B4: $Bxyz \wedge Byzu \Longrightarrow Bxyu \wedge Bxzu$,
B5: $Bxyz \wedge Bxuy \Longrightarrow Bxuz \wedge Buyz$,
B6: $Bxyz \wedge Bxuz \Longrightarrow y = u \vee B^+xyuz \vee B^+xuyz$,
B7: $\neq xyz \Longrightarrow Axyz \vee \exists w.(Bxwy \wedge Bywz \wedge Bxwz)$.

Remark 4.3. (1) Except B7, these properties are universal. Property B7 asserts the existence of a node w if $\neq xyz \wedge \neg Axyz$. The unicity of such w is a consequence of the universally quantified B1–B6 (see Lemma 11 of [6]) in arbitrary ternary structures. We will denote w by $M(x,y,z)$. Using the notation $M(x,y,z)$ will mean that we have $\neq xyz \wedge \neg Axyz$.

(2) The relation B is empty if and only if T has at most two nodes, and then, B1–B7 hold trivially.

Definition 4.4. [6]: *Quasi-trees and related notions.*

A *quasi-tree* is a pair $Q = (N, B)$ where N is the set of *nodes* and B is a ternary relation that satisfies Properties B1–B7 (for all nodes). We say that x is *internal* if $Byxz$ holds for some nodes y and z; otherwise, x is a *leaf*[22]. We denote by L_Q the set of leaves.

A quasi-trees $Q = (N, B)$ is *leafy* if every internal node is $M(x, y, z)$ for some leaves x, y, z. We cannot have $|N| = 3$. We accept $|N| \leq 2$ and $B = \emptyset$ as there are no internal nodes and thus, each node is a leaf. We let $\mathcal{Q}$ be the class of leafy quasi-trees. Leafy join-trees are defined in a similar way.

Remark 4.5. With B4 that we will always assume in our proofs concerning ternary structures (N, B), $B^+x_1...x_n$ implies $Bx_ix_jx_k$ for all $1 \leq i < j < k \leq n$.

(2) For leaves x, y, z, we never have $Bxyz$, hence properties B1–B6 are vacuously true and B7 reduces to:
B7': $\neq xyz \Longrightarrow \exists w.(Bxwy \wedge Bywz \wedge Bxwz)$.

(3) In quasi-trees, we consider an infinite interval $[x, y]$ between distinct nodes x and y as a "path". It is lineraly ordered as follows:
$u < v :\Longleftrightarrow (u = x \wedge v = y) \vee (\neq xuv \wedge Bxuv) \vee (\neq uvy \wedge Buvy)$.
Hence, $x \leq u \leq y$ for all $u \in [x, y]$.

(4) An example of a quasi-tree is $([0, 1], B)$ where $[0, 1]$ is the closed interval of rational numbers between 0 and 1, and $Bxyz$ holds if and only if $x < y < z \vee z < y < x$. There is an infinite "path" between its two leaves 0 and 1. This quasi-tree is not leafy. If we delete 0 and 1, we obtain a quasi-tree without leaves.

(5) Lemma 2.7 of [11] shows that B1–B7 imply the following:
B8: $\neq xyzu \wedge Bxyz \wedge \neg Ayzu \Longrightarrow Bxyu$.
This is a universal property, hence it is invariant under taking substructures. The induced substructures of quasi-trees, we call them *partial quasi-trees*, are the models of B1–B6 and B8 by Theorem 3.1 of [11]. We will study them below in a logical perspective.

(6) If a ternary structure (N, B) of cardinality at least 3 satisfies Property B7, then its Gaifman graph is connected. This connectedness property is not implied by B1–B6 that are universal.

The next proposition shows that a quasi-tree can be seen as the undirected version of a join-tree.

Proposition 4.6. *(1). If $Q = (N, B) \in \mathcal{Q}$ and $r \in N$, then $T(Q, r) := (N, \leq_r)$ is a join-tree with root r, whose defining ordering is:*

$$x \leq_r y :\Longleftrightarrow x = y \vee y = r \vee Bxyr.$$

Its join operation satisfies $x \sqcup_r r := r$ and $x \sqcup_r y := M(x, y, r)$ if $\neq rxy$.
(2) Conversely, every rooted join-tree in $\mathcal{T}$ is of this form.

Proof. Lemma 14 of [6] proves the first assertion. Conversely, if $T = (N, \leq)$ is a rooted join-tree with root r, we define B as follows:

$$Bxyz :\Longleftrightarrow (x < y < z) \vee (z < y < x) \vee (x \neq z \wedge x \sqcup z < y).$$

It is easy to see that $\leq$ is $\leq_r$, defined from B as above. The hypothesis that T is a join-tree implies that $|N| \geq 3$. ☐

In the following proofs, we will use B2 without explicit mention.

Proposition 4.7. *Let x, y, z be three leaves of a quasi-tree (N, B), $a := M(x, y, z)$ and u be any node.*
(i) We have $Buax \vee (Buay \wedge Buaz)$.
(ii) The internal node a is $M(u, x, y)$, $M(u, y, z)$ or $M(u, x, z)$.
(iii) If $Buax$ holds, then a is $M(u, x, y)$ or $M(u, x, z)$.

Proof. (i). Assume that $Buax$ does not hold. If $Baux$ holds, we have B^+xuay and B^+xuaz. Hence $Buay$ and $Buaz$. Otherwise, $b := M(x, a, u)$ is defined (because x is a leaf, so that $Baxu$ cannot hold). We have $Bxba$ and

[22] This definition applies to every ternary structure (N, B).

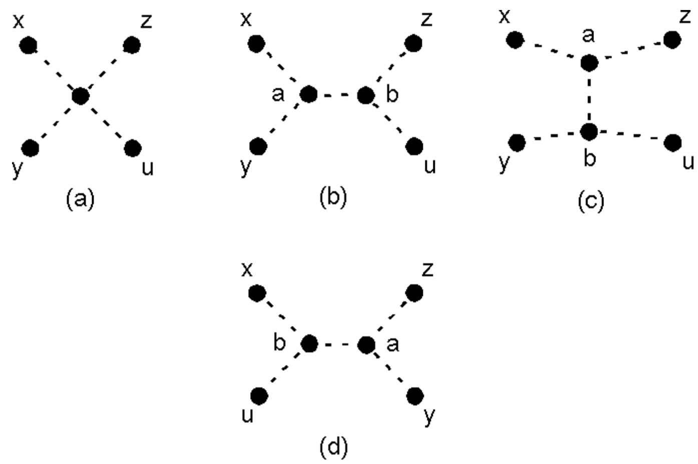

FIGURE 7. See Proposition 4.8 and Remark 4.9(1).

$Bxay$, hence we have B^+xbay by B5, whence $Bbay$. As $Buba$ holds, B4 gives B^+ubay. Hence $Buay$ holds. The proof is similar for z.

(ii) We have two cases. If $Buax$ does not hold, then $Buay \wedge Buaz$ holds and yields $a = M(u, y, z)$. If $Buax$ hold and so does $Buay$, we have $a = M(u, x, y)$. If $Buay$ does not hold, we have $a = M(u, x, z)$ by the very first case.

(iii) Let $Buax$ hold. If $Buay$ holds, we have $a = M(u, x, y)$. Otherwise we have $Buax \wedge Buaz$ by (i). Hence, $a = M(u, x, z)$. □

We recall that $[x, y]$ denotes the *interval* $\{x, y\} \cup \{u \in N \mid Bxuy\}$.

Proposition 4.8. *Let x, y, z, u be pairwise distinct leaves of a quasi-tree $Q = (N, B)$. Let $a := M(x, y, z)$ and $b := M(x, u, z)$.*

We have the following cases.

(1) $|[x, y] \cap [z, u]| = 1$. Then $a = b, [x, y] \cap [z, u] = [x, z] \cap [y, u] = [x, u] \cap [y, z] = \{a\}$, $M(x, y, u) = M(y, z, u) = a$, and $Bvaw$ holds for any two distinct leaves v, w in $\{x, y, z, u\}$.

(2) $[x, y] \cap [z, u] = \emptyset$. Then $a \neq b$, $[x, z] \cap [y, u] = [x, u] \cap [y, z] = [a, b]$, B^+xabz,B^+xabu, B^+yabz and B^+yabu hold, $M(x, y, u) = a$ and $M(y, z, u) = b$.

(3) $|[x, y] \cap [z, u]| \geq 2$. Then either $[x, z] \cap [y, u] = \emptyset$ or $[x, u] \cap [y, z] = \emptyset$; the consequences follow from Case (2) by appropriate substitutions of variables.

We will express Case (1) by $Exyzu$ where E is a 4-ary relation, and similarly Case (2) by $Sxyzu$.

Proof. (1) Let $T(Q, z) = (N, \leq_z)$ be the join-tree with root z constructed by Proposition 4.6. Then $x <_z a, b <_z z$.

If $|[x, y] \cap [z, u]| = 1$ then $a = b$, the consequences are clear. See Figure 7(a).

(2) We consider again $T(Q, z)$. If$[x, y] \cap [z, u] = \emptyset$, then $a <_z b$. The consequences are clear. See Figure 7(b).

(3) If $|[x, y] \cap [z, u]| \geq 2$, we consider $T(Q, y)$. If $a <_y x \sqcup_y u$, then $[x, z] \cap [y, u] = \emptyset$. If $a >_y x \sqcup_y u$, then $[x, u] \cap [y, z] = \emptyset$. See Figures 7(c) and 7(d) respectively. □

Remark 4.9. (1) Figure 7 shows the four possible relative configurations of four leaves x, y, z and u. The dotted lines represent "paths".

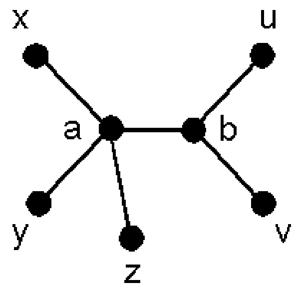

FIGURE 8. See Remark 4.9(3).

(2) $M(x,y,z) = M(x,y,u) = M(x,z,u)$implies $M(x,y,z) = M(y,z,u)$ and $[xy] \cap [zu] = \{M(x,y,z)\}$. This follows from the definitions, we are in Case (1). However, $M(x,y,z) = M(x,y,u)$ does not imply $M(x,y,z) = M(x,z,u)$. See Figure 7(b).

(3) It is not true that $u \neq v \wedge Exyzu \wedge Exyzv \Longrightarrow Exyuv$. Take for an example $a := M(x,y,z), Babu$ and $Babv$. We have then $b = M(a,u,v) \neq a$. See Figure 8.

(4) The relations E_Q and S_Q can be determined from the restriction of B_Q to $L_Q \times I_Q \times L_Q$, because from this restriction, one can determine the intersections $[x,y] \cap [z,u]$, $[x,z] \cap [y,u]$ and $[x,u] \cap [y,z]$.

4.1. Leafy quasi-trees

A quasi-tree may have no leaf. Hence, one cannot hope to reconstruct an arbitrary quasi-tree from its leaves. This reconstruction is only possible for leafy quasi-trees.

By Definition 4.4, a leafy quasi-tree Q is the union of the intervals $[x,y]_Q$ for all leaves x,y. If X is a set of leaves of a quasi-tree $Q = (N,B)$ and $Y := X \cup \{M(x,y,z) \mid x,y,z \in X\}$, then $(Y, B[Y])$ is a leafy quasi-tree: for proving this, just consider the rooted join-tree $T(Q,u)$ for some u in X, *cf.* Proposition 4.6. Then the remark in Definition 2.1(c) shows that Y, ordered by $\leq_u$, is a join-tree, which yields the result. We omit details.

Our aim is now to describe a leafy quasi-tree Q from an appropriate relation on its set of leaves L, similarly to what we did for join-trees by means of the ternary relation R in Theorem 2.10. The 4-ary relations E and S are defined after Proposition 4.8. We call S the *separation relation* of Q, and (L,S) its *separation structure.* The next proposition is an immediate corollary of Proposition 4.8. The symbol $\oplus$ denotes the exclusive or.

Proposition 4.10. *The following properties hold for all leaves x,y,z,u of a quasi-tree.*

S1: $Sxyzu \vee Exyzu \Longrightarrow \neq xyzu$.
S2: $Sxyzu \Longleftrightarrow Szuxy \Longleftrightarrow Syxzu$.
S3: $Exyzu \Longleftrightarrow Eyxzu \Longleftrightarrow Eyzux$.
S4: $\neq xyzu \Longrightarrow Exyzu \oplus Sxyzu \oplus Sxzyu \oplus Sxuyz$.

The following properties are consequences of S1 and S4.

S'4: $Exyzu \Longleftrightarrow \neq xyzu \wedge \neg Sxyzu \wedge \neg Sxzyu \wedge \neg Sxuyz$
S"4: $Sxyzu \Longrightarrow \neq xyzu \wedge \neg Sxzyu \wedge \neg Sxuyz$.

Property S3 shows that relation E is stable under every permution of its four arguments. Property S4 is illustrated in Figure 7. By S'4 the relation E is definable from S and we will use it to shorten formulas. Hence, all our results will be formulated in terms of 4-ary structures (L,S). By Properties S2 and S4, the relation S is stable by less permutations of its arguments than E.

Property S4 is formally similar to property A10 of Remark 2.7.

Example 4.11. Figure 8 shows a finite quasi-tree for which we have:

$a = M(x,y,z) = M(x,y,u) = M(x,z,u)$.
$b = M(x,u,v) = M(y,u,v) \neq a$.
$Sxyuv \wedge Sxzuv \wedge Exyzu \wedge \neg Eyzuv$. □

The following propoposition is similar to Proposition 2.4.

We will denote by EA where A is a set of cardinality at least 4, the property that $Exyzu$ holds for any four distinct elements (called nodes in a quasi-tree) x,y,z,u in A. We will denote by SAB where A,B are disjoint sets of cardinality at least 2 the property that $Sxyzu$ holds for any distinct x,y in A and z,u in B.

Lemma 4.12. *Let Q be a leafy quasi-tree with set of internal nodes I_Q. Let x,y be distinct leaves and a,b,c be distinct internal nodes.*

(i) $Bxay$ holds if and only if $a = M(x,u,y)$ for some leaf u.

(ii) $Bxab$ holds if and only if $a = M(x,u,v)$ and $b = M(x,v,w)$ for some leaves u,v and w such that $Sxuvw$ holds.

(iii) $Babc$ holds if and only if $Buab, Bubc$ and $Buac$ hold for some leaf u.

Proof. (i) Assume $Bxay$. By Proposition 4.7, we have $a = M(x,u,v)$ for some leaves u,v. If $Buay$ holds, then $a = M(x,u,y)$. If not, we have $Byax$ and $Byav$ hence $a = M(x,v,y)$. Conversely, if $a = M(x,u,y)$ then $Bxay$ holds.

(ii) Assume $Bxab$. We have $b = M(x,v,w)$ for some leaves v,w by Proposition 4.6. Hence we have B^+xabv, and so $Bxav$. Then $a = M(x,u,v)$ for some leaf u by (i). Clearly, $Sxuvw$ holds.

Conversely, let $a = M(x,u,v)$ and $b = M(x,v,w)$ for some leaves u,v and w such that $Sxuvw$ holds. We have $Bxav$ and $Bxbv$. We cannot have $a = b$ since $Exuvw$ does not hold. Hence, we have B^+xabv or B^+xbav. Because of $Sxuvw$ the latter cannot hold. Hence we have B^+xabv and so $Bxab$.

(iii) Assume $Babc$. Then $a = M(u,v,c)$ for some leaves u,v by Proposition 4.6. Hence, we have $Buac$ and also B^+uabc and so we have $Buab, Bubc$ and $Buac$. Conversely, if $Buab, Bubc$ and $Buac$ hold, then B^+uabc holds by a now routine argument, and so does $Babc$. □

Proposition 4.13. *A leafy quasi-tree $Q = (N,B)$ in $\mathcal{Q}$ is characterized by the structure (L_Q, S_Q).*

Proof. If Q has exactly three leaves, then S_Q is empty, but since Q is leafy, it has a unique internal node and is characterized by (L_Q, S_Q). Leaving this case, we will reconstruct Q *u.t.i.* from $(L,S) = (L_Q, S_Q)$ by using Propositions 4.7 and Lemma 4.12.

We define $j : L \times L \times L \to N$ by $j(x,y,z) := M(x,y,z)$ provided x,y and z are pairwise distinct. We denote by m,p,q such triples.

We denote by xyz a triple of leaves (x,y,z).

We have $j(m) = j(p)$ where $m = xyz$ and $p = uvw$ if and only if

$$(x,y,z) \text{ is a permutation of } (u,v,w) \text{ or } E\{x,y,z,u,v,w\} \text{ holds, } \textit{cf.} \text{ Proposition 4.8.} \tag{1}$$

We write then $m \equiv p$. This equivalence relation is definable from E, hence from S *via* S3. For distinct leaves x and y, and pairwise nonequivalent triples m,p,q as above, we have, by Lemma 4.12:

$$B(x,j(m),y) \Longleftrightarrow m \equiv xzy \text{ for some leaf } z, \tag{2}$$
$$B(x,j(m),j(p)) \Longleftrightarrow m \equiv xuv,\ p \equiv xvw \text{ and } Sxuvw \text{ holds for some leaves } u,v \text{ and } w, \tag{3}$$
$$B(j(m),j(p),j(q)) \Longleftrightarrow B(u,j(m),j(p)),\ B(u,j(p),j(q)) \text{ and } B(u,j(m),\ j(q)) \text{ hold for some leaf } u. \tag{4}$$

If $R = (N,B') \in \mathcal{Q}$ is such that $(L_R, S_R) = (L_Q, S_Q)$, if j is as above and j' is defined similarly for R, then we have identity between the sets of leaves of R and Q and furthermore, $j(m) = j(p)$ if and only if $j'(m) = j'(p)$ because $S_R = S_Q$ and the equalities $j(m) = j(p)$ and $j'(m) = j'(p)$ are defined in the same ways from S_Q and S_R. We obtain a bijection between the internal nodes of R and Q.

By (2)–(4) above, the betweenness relations of R and Q are defined in the same ways from S_Q and S_R. Hence, $R = (N_R, B_R)$ and $Q = (N_Q, B_Q)$ are isomorphic. □

In a separation structure (L, S), the relation S is empty if $|L| \leq 3$ or if $Exyzu$ holds for any four leaves x, y, z, u. In the latter case, the quasi-tree has a single internal node. Proposition 4.13 is valid. We admit quasi-trees $(N, \emptyset)$ where $|N| \leq 2$.

Corollary 4.14. *Let Q be a leafy quasi-tree with set of internal nodes I_Q. Its relations S_Q, E_Q and B_Q are FO definable in the the relational structure $(N_Q, B_Q \cap (L_Q \times I_Q \times L_Q))$.*

Proof. Note that M_Q is FO definable in $(N_Q, B_Q \cap (L_Q \times I_Q \times L_Q))$. The statements follow from the definitions Remark 4.9(4) and Proposition 4.13. □

Our next aim is to find an FO axiomatization of the separation relations that characterize leafy quasi-trees from their leaves.

Proposition 4.15. *The class of separation structures of leafy quasi-trees is hereditary.*

Proof. Let $X \subseteq L_Q$ where $Q = (N, B)$ is a quasi-tree. Let $Y := X \cup \{M_Q(x, y, z) \mid x, y, z \in X\}$. We get a leafy quasi-tree (*cf.* the beginning of Section 3.1) $P = (Y, B[Y])$, possibly reduced to one or two nodes: in this case $B[Y]$ is empty.

We claim that $(X, S_Q[X]) = (L_P, S_P)$.

We use Proposition 4.8. Note that $[x, y]_P \cap [z, u]_P \subseteq [x, y]_Q \cap [z, u]_Q$. If S_Qxyzu holds where $x, y, z, u \in X$, then $[x, y]_Q \cap [z, u]_Q$ is empty and so is $[x, y]_P \cap [z, u]_P$. Hence, we have S_Pxyzu. Conversely, if we have S_Pxyzu, then $[x, y]_P \cap [z, u]_P$ contains the two distinct nodes $M_P(x, y, z)$ and $M_P(x, z, u)$, so does $[x, y]_Q \cap [z, u]_Q$ hence S_Qxyzu holds. □

Construction 4.16. *A leafy quasi-tree constructed from its leaves.*

Let (L, S) be a 4-ary structure satisfying Properties S1–S4 (where the relation E is FO-defined from S by Property S'4) and Property EQ to be defined below. We construct as follows a ternary structure (N, B):

> $N := P/\equiv$ where $P \subseteq L \times L \times L$ and $\equiv$ is an equivalence relation on P,
> $B := B'/\equiv$ where B' is a ternary relation on P that is stable under $\equiv$, which means that if $m \equiv m', p \equiv p'$ and $q \equiv q'$ then $B'(m, p, q)$ implies $B'(m', p', q')$.
> Furthermore, $P, \equiv$ and B' will be FO definable in (L, S).

For this purpose, we let P be the set of triples in $L \times L \times L$ (denoted by xyz) whose components are all equal or pairwise distinct.

We define $xyz \equiv uvw$ if and only if either (x, y, z) is a permutation of (u, v, w) or $E\{x, y, z, u, v, w\}$ holds (the relation E extended to finite sets of fixed size is FO-definable from S by S'4). This property is expressible by a quantifier-free formula $\eta(x, y, z, u, v, w)$.

We let EQ be the ∀FO sentence expressing the transitivity of $\equiv$, hence that this relation is actually an equivalence relation.

We only consider structures (L, S) that satisfy S1–S4 and EQ.

We define $B'(m, p, q)$ where $m = x_1x_2x_3, p = y_1y_2y_3$ and $q = z_1z_2z_3$as follows:

> $m, p, q \in P$, they are pairwise not equivalent, $\neq y_1y_2y_3$ and
> $B'(m, p, q) :\Longleftrightarrow B'_0(m, p, q) \vee B'_0(q, p, m) \vee B'_1(m, p, q)$ where
> (a) $B'_0(m, p, q)$ holds if and only if one has:
> (i) either $x_1 = x_2 = x_3, z_1 = z_2 = z_3$ and $p \equiv x_1uz_1$ for some u in $\{y_1, y_2, y_3\}$,
> (ii) or $x_1 = x_2 = x_3, \neq z_1z_2z_3$, $p \equiv x_1uv$, $q \equiv x_1vw$ and Sx_1uvw holds for some u in $\{y_1, y_2, y_3\}$ and distinct v, w in $\{z_1, z_2, z_3\}$.
> (b) $B'_1(m, p, q)$ holds if and only if, for some $u \in L$, we have:
> $B'_0(uuu, m, p) \wedge B'_0(uuu, p, q) \wedge B'_0(uuu, m, q)$.

This definition is based on Equivalences (2)–(4) of the proof of Proposition 4.13. From its definition, Property B' is stable under $\equiv$. An FO formula $\beta(x_1, x_2, x_3, y_1, y_2, y_3,\ z_1, z_2, z_3)$ can be written to express that $B'(x_1x_2x_3, y_1y_2y_3, z_1z_2z_3)$ holds in a 4-ary structure (L, S). □

Proposition 4.17. *For every FO sentence φ over a ternary structure (N, B), one can construct an FO sentence $\widehat{\varphi}$ such that, if (N, B) is defined by Construction 4.16 from (L, S), then $(L, S) \models \widehat{\varphi}$ if and only if $(N, B) \models \varphi$.*

Proof. The transformation of (L, S) into (N, B) is a vectorial FO-transduction of dimension 3 (*cf.* Corollary 2.13) followed by a quotient. We avoid rescourse to general statements to be developped in [20]. The result follows by an induction on the structure of subformulas $\psi(x, y, ...)$ of φ with free variables x, y, *etc.* transformed into $\widehat{\psi}(x_1, x_2, x_3,\ y_1, y_2, y_3, ...)$. Three main steps are as follows:

$\exists x.\psi(x, y, ...)$ becomes $\exists x_1, x_2, x_3.\pi(x_1, x_2, x_3) \wedge \widehat{\psi}(x_1, x_2, x_3,\ y_1, y_2, y_3, ...)$ where π defines P,
$x = y$ becomes $\eta(x_1, x_2, x_3, y_1, y_2, y_3)$ (we recall that η defines $\equiv$),
$B(x, y, z)$ becomes $\beta(x_1, x_2, x_3, y_1, y_2, y_3, z_1, z_2, z_3)$ where β defines B'.

□

Theorem 4.18. *The class of separation structures of leafy quasi-trees is first-order definable by a sentence Φ.*

Proof. A 4-ary structure (L, S) is the separation structure of a leafy quasi-tree if and only if it satisfies Properties S1–S4, the FO sentence EQ and an FO sentence that we will define.

We let P consists of the triples (x, y, z) in $L \times L \times L$ (written xyz) such that either $x = y = z$ or $\neq xyz$. We identify xxx to the leaf x. Construction 4.16 defines a transformation of (L, S) into (N, B).

By Definition 4.4, there is a FO sentence φ expressing that a ternary structure (N, B) is a leafy quasi-tree. Proposition 4.13, yields a FO sentence $\widehat{\varphi}$ expressing in (L, S) that $(N, B) \models \varphi$, *i.e.*, that (N, B) is a leafy quasi-tree.

It remains to express that (L, S) is the separation structure of (N, B). This is possible by a FO-sentence θ because B is defined in (L, S) by β. Hence Φ is the conjunction of S1–S4, EQ and $\widehat{\varphi} \wedge \theta$. □

The sentence Φ is cumbersome to write and tells nothing about the relations S and E in quasi-trees for which we only know presently Properties S1–S4 by Proposition 4.10.

The theorem by Los and Tarski (see [26] or Hodges [21], Theorem 6.5.4) says that if a class of finite and infinite relational structures is hereditary and axiomatizable by a single FO sentence, then it is also by a single $\forall$FO sentence. This is the case of separation structures by Proposition 4.17.

Open problem: *Can one find a readable FO axiomatization of the separation structures of quasi-trees*, similar to those of Theorem 2.10 for the leaf structures of join-trees, of Definition 4.4 for quasi-trees and of Remark 4.5(5) for partial quasi-trees.

Here is an example of an axiom that one can add to S1–S4 (it is illustrated in Figure 8):

S5: $z \neq v \wedge Exyzu \wedge Sxyuv \Longrightarrow Sxzuv$.

Proof sketch. of its validity: let $a := M(x, y, u)$ and $b := M(x, u, v)$; from the hypotheses, we have B^+xabu, $Bzau$, hence B^+zabu. With B^+xabv, we get $Sxzuv$. □

Property S5 is not implied by S1–S4. We may have S1–S4 together with

$$Exyzu \wedge Sxyuv \wedge z \neq v \wedge \neg Sxzuv \wedge Exzuv :$$

observe that S1–S4 only concern 4-tuples $xyzu$ and never pairs of 4-tuples like $xyzu$ and $xyuv$.

Next, we look for MSO transductions that transform a structure (L, S) satisfying Φ into an associated leafy quasi-tree.

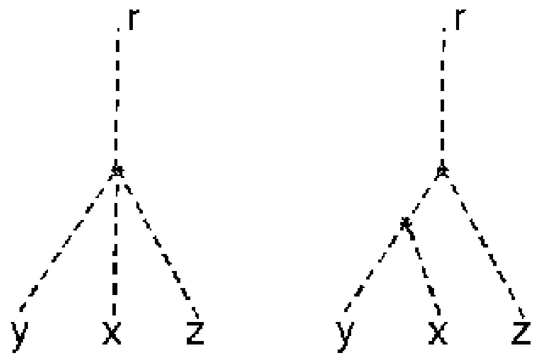

FIGURE 9. See Theorem 4.19.

Theorem 4.19. There exists a $\leq$-MSO transduction that transforms a 4-ary structure (L, S) satisfying Φ into a leafy quasi-tree Q such that $(L_Q, S_Q) \simeq (L, S)$. A CMSO-transduction can do that if L, whence Q, is finite.

Proof. We first consider a leafy quasi-tree $Q = (N, B)$ and one of its leaves r. Let $T(Q, r)$ be the rooted join-tree $(N, \leq_r)$, *cf.* Proposition 4.6. Note that r is not $x \sqcup_r y$ for any leaves x, y of $T(Q, r)$. Let $T'(Q, r) := (N - \{r\}, \leq'_r)$ where $\leq'_r$ is the restriction of $\leq_r$ to $N - \{r\}$. Then, $T'(Q, r)$ is a leafy join-tree. It may have no root. Its set of leaves is $L_Q - \{r\}$.

Claim 4.20. *The ternary relation $R' := R_{T'(Q,r)}$ is FO definable from S_Q:*

for all x, y and z in $L_Q - \{r\}$, $R'xyz$ holds if and only if
either $x = y = z$ or $x = y \neq z$ or $x = z \neq y$ or $E_Q xyzr$ or $S_Q xyzr$ or $S_Q xzyz$. □

Figure 9 illustrates the cases $E_Q xyzr$ and $S_Q xyzr$ of this claim.

We now start from (L, S) satisfying Φ, hence that it is the separation structure of a unique (*u.t.i.*) quasi-tree. An MSO-transduction can transform the separation structure (L, S) of a leafy join-tree into a leaf structure $(L - \{r\}, R')$ of a leafy join-tree $T'(Q, r) = (N - \{r\}, \leq'_r)$ by means of the claim. Then, an $\leq$-MSO transduction (with copying constant 2) can transform $(L - \{r\}, R')$ into $(N - \{r\}, \leq'_r)$ by the result of [2] recalled in Theorem 3.1. An MSO-transduction (with copying constant 2) can transform $(N - \{r\}, \leq'_r)$ into the join-tree $(N, \leq_r)$, by adding r to the domain and because:

$x \leq_r y$ if and only if $x \leq'_r y \vee y = r \vee x = y = r$.

Finally, an FO transduction (*cf.* Section 2.1) can transform $(N, \leq_r)$ into the quasi-tree $Q = (N, B)$ because B is FO definable from $\leq_r$. The composition of these four transductions yields an $\leq$ -MSO transduction as claimed. If L is finite, so is N. The construction of Theorem 3.4 makes it possible to replace auxiliary linear orders by counting modulo 3 set predicates. □

The choice of r is irrelevant because for any other leaf r', the structure $(L - \{r\}, R')$ and the corresponding one $(L - \{r'\}, R'')$ are exchangeable by MSO transductions.

4.2. Rank-width and separation structures

The rank-width of finite graphs is an important notion in the theory of *Fixed Parameter Tractability*, see [19, 22]. Quasi-trees are useful for defining the rank-width of a countable graph. We review definitions and background results from [6, 22, 23].

Graphs are undirected and *simple*, that is, without loops and parallel edges.

Definition 4.21. *Rank-width of finite graphs*

(a) A *layout* of a finite graph $G = (V_G, edg_G)$ is a finite subcubic tree $T = (N_T, edg_T)$ such that V_G is the set of its leaves. A tree is *cubic* (resp. *subcubic*) if its nodes are of degree 1 or 3 (resp. 1,2 or 3).

(b) A *cut* of T is a bipartition of N_T into sets C_x and C_y associated with an edge $x - y$, where C_x is the set of nodes z such that $z = x$ or $Bzxy$ holds. Hence, C_x induces a subtree $(C_x, edg_T[C_x])$ containing x, and of course, similarly for C_y.

(c) If X and Y are subsets of V, then $M_G[X, Y]$ is the rectangular adjacency matrix of G over the field $GF(2)$ such that $M_G[x, y] = 1$ if and only if there is an edge $x - y$ in G. It is symmetric if $X = Y$.

(d) The *rank-width of G relative to T*, denoted by $rwd(G, T)$, is the maximal rank (over $GF(2)$) of the rectangular matrices $M_G[V \cap C_x, V \cap C_y]$ for all edges $x - y$ of T. The *rank-width of G*, denoted by $rwd(G)$, is the minimal rank-width $rwd(G, T)$ for all layouts T of G. □

For each k, one can decide in cubic time if a graph G has rank-width at most k by [22], Theorems 9.5 and 12.8. Furthermore, a CMSO sentence can express that a graph $G = (V_G, edg_G)$ has rank-width at most k by ([23], Corollary 2.6 and Section 6.4). However, such a sentence does not yield a CMSO transduction that could construct a layout witnessing that G has rank-width $\leq k$ because it is based of checking the absence of "forbidden" configurations called *vertex-minors* (see the survey [22]) from a finite set.

For defining the rank-width of a countable graph, quasi-trees offer the appropriate notion of layout because then, the rank-width of a countable graph is the least-upper bound of those of its finite induced subgraphs [6]. We first consider finite layouts T described by their separation structures. In particular, they permit to express the property $rwd(G, T) = k$ by a CMSO effectively constructible sentence.

Lemma 4.22. *Let T be a finite leafy quasi-tree (based on a tree).*

(1) For pairwise distinct leaves u, v, w, s, we have:

$S_T uvws$ if and only if $u, v \in C_x$ and $w, s \in C_y$ for some edge $x - y$ of T.

(2) The tree T is cubic (equivalently subcubic) if and only if $E_T = \emptyset$.

Proof. (1). If $u, v \in C_x$ and $w, s \in C_y$, then $[u, v] \subseteq C_x$ and $[w, s] \subseteq C_y$, and C_x and C_y are disjoint, hence $S_T uvws$ holds. Conversely, if $[u, v] \cap [w, s] = \emptyset$ then Proposition 4.8 (2) shows that $M(u, v, w) \neq M(u, w, s)$; any edge $x - y$ on the path from $M(u, v, w)$ to $M(u, w, s)$ gives $u, v \in C_x$ and $w, s \in C_y$ (or vice-versa).

(2) A leafy subcubic quasi-tree is cubic, as it cannot have nodes of degree 2[23]. If $E_T uvws$ holds[24], then $M(u, v, w)$ has degree at least 4. Conversely, if x has adjacent nodes u', v', w', s', then there are leaves u, v, w, s, respectively in $C_{u'}, C_{v'}, C_{w'}, C_{s'}$ (defined from $x - u'$, $x - v'$ *etc.*), and disjoint paths between x and these leaves. Hence, $E_T uvws$ holds. □

We recall from Lemma 4.12 that for disjoint sets of leaves A and B, the notation $S_T AB$ means that $S_T uvws$ holds for any distinct $u, v \in A$ and distinct $w, s \in B$.

Lemma 4.23. *If T is a layout of a finite graph G, then $rwd(G, T)$ is the maximal rank of the matrix $M_G[A, B]$ for all sets $A, B \subseteq L_T = V_G$ such that $S_T AB$ holds.*

Proof. The previous lemma shows that, if $A \subseteq C_x$ and $B \subseteq C_y$ for some edge $x - y$, then $S_T AB$ holds. Conversely, assume we have $S_T AB$ for some subsets of V_G. If follows from the proof of Lemma 4.25 below that $A \subseteq C_x$ and $B \subseteq C_y$ for some edge $x - y$. We omit details. □

We now extend these definitions and results to infinite graphs. For finite graphs, the following definitions coincide with the previous ones. A leafy quasi-tree Q is *cubic* if its relation E_Q is empty. This definition is equivalent to Definition 18 of [6].

Definition 4.24. [6]: *The rank-width of countable graphs*

(a) A *layout* of a graph $G = (V_G, edg_G)$ is a cubic and leafy quasi-tree $Q = (N_Q, B_Q)$ such that V_G is its set of leaves.

[23] See [6] for detailed definitions.
[24] See Property 4.8(1) and Figure 7(a).

(b) A set $C \subseteq N_Q$ is *convex* if $[u,v] \subseteq C$ whenever $u, v \in C$. A *cut* of Q is a bipartition of N_Q into convex sets C and D having at least two nodes. Hence, $S_Q CD$ holds because $[u,v] \cap [w,s] = \emptyset$ if $u, v \in C$ and $w, s \in D$.

(c) The *rank-width of* G *relative to* Q, denoted by $rwd(G,Q)$ is the maximal rank of a rectangular matrix $M_G[V \cap C, V \cap D]$ for a cut $\{C, D\}$ of Q. If C or D is singleton, the corresponding rank is 1 or 0. We can neglect such cuts as the rank of a graph with at least one edge is at least 1.

(d) The *rank-width of* G, denoted by $rwd(G)$ is the minimal rank-width $rwd(G,Q)$ for a layout Q of G. It may be infinite. We might accept as layout a quasi-tree that is subcubic or not leafy. The resulting notion of rank-width is the same by Proposition 22 of [6].

Lemma 4.25. *Let A, B be disjoint sets of leaves of a leafy quasi-tree $Q = (N, B)$. If $S_Q AB$ holds, then $A \subseteq C$ and $B \subseteq D$ for some cut $\{C, D\}$.*

Proof. Let u be a leaf of a leafy quasi-tree $Q = (N, B)$ and $T := (N, \leq)$ be the join-tree $T(Q, u)$ with root u defined in Proposition 4.6. Let x, y, z be other leaves of Q: we have $x \sqcup y < x \sqcup z$ if and only if $S_Q xyzu$ holds. This fact follows from Proposition 4.8 since $x \sqcup_T y = M_Q(x, y, u)$ and $x \sqcup z = M_Q(x, z, u)$.

Let A, B be disjoint sets of leaves of Q as above, such that $S_Q AB$ holds. Let $u \in B$ and $T = (N, \leq)$ be the join-tree $T(Q, u)$. We define C and D.

Let $x \in A$. For each $y \in A$, we have $x \leq x \sqcup y < u$. Let $C := \{w \in N \mid w \leq x \sqcup y \text{ for some } y \in A\}$. It is convex because for any two w, w' in C, we have $w \sqcup w' \leq x \sqcup y$ for some $y \in A$, hence all nodes in $[w, w']_Q$ are below $w \sqcup w'$, hence below $x \sqcup y$, whence in C.

We prove by contradiction that $B \cap C$ is empty. If $z \in B \cap C$, we have $z \leq x \sqcup y$ for some $x, y \in A$. But we have $S_Q xyzu$ which implies that $a := M(x, y, u) = x \sqcup y < b := M(x, z, u) = x \sqcup z$. We cannot have $z \leq x \sqcup y$.

Consider now $D := N - C$. We have $B \subseteq D$. Let w in D that is not a leaf. There is in B a leaf $z < w$. As C is downwards closed, the interval $[z, u]_Q$ is contained in D and contains $[w, u]_Q$. For another w' in D we have similarly $z' < w'$ for some leaf z' in B. It follows that the interval $[w, w']_Q$ is contained in $[w, t]_Q \cup [t, w']_Q$ where $t := w \sqcup w'$, hence D is convex. □

Theorem 4.26. *[6], Theorem 23: the rank-width of a graph is the least upper-bound of those of its finite induced subgraphs.*

The following proposition makes no use of forbidden vertex-minors.

Proposition 4.27. *For each k, a CMSO sentence can express, in a relational structure (V, edg, S), that (V, S) is the separation structure of a layout Q of the graph $G := (V, edg)$ and that $rwd(G, Q) \leq k$.*

Proof. First we observe that an FO sentence φ can express that (V, S) is the separation structure of a layout Q of G, where $V = L_Q$. This follows from Theorem 4.18 and the observation that Q is subcubic if and only if E_Q is empty.

The rank of a matrix $M_G[X, Y]$ where X is finite is the cardinality of a maximal set[25] $Z \subseteq X$ such that the row vectors $M_G[z, Y]$ for $z \in Z$ are independent. These vectors are independent if and only if their sum is not $\overrightarrow{0}$, that is, if and only if the sum in $GF(2)$ of the values $M_G[z, y]$'s is not 0 for some $y \in Y$. A CMSO formula $\beta(Z, Y)$ can express this in (V, edg) by means of the cardinality predicate C_2. Hence, a CMSO formula $\alpha(X, Y, Z)$ can express that $Z \subseteq X$ is such a maximal set, and so, that the rank of $M_G[X, Y]$ is $|Z|$.

It follows that, a relational structure (V, edg, S) as above, defines a layout $Q = (V, S)$ of $G := (V, edg)$ such that $rwd(G, Q) \leq k$ if and only if it satisfies φ and the following property:

for all finite subsets X, Y, Z of V, if $S_Q XY$ and $\alpha(X, Y, Z)$ hold, then $|Z| \leq k$.

A CMSO sentence ρ_k can express these conditions because the finiteness of a set is CMSO expressible[26], $\alpha(X, Y, Z)$ is CMSO, φ and $S_Q XY$ are FO, and so is the property $|Z| \leq k$ for each fixed integer k.

This upper-bound may not exist: then $rwd(G, Q) = \omega$.

[25] Maximal for inclusion.

[26] A set X is finite if and only if either X or X minus any element satisfies C_2. The set predicate C_2 is only valid on finite sets.

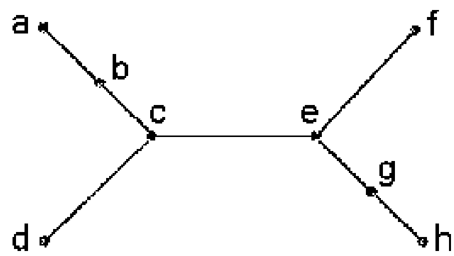

FIGURE 10. For Example 4.29.

The sentences ρ_k can be replaced by a unique CMSO sentence $\rho(Z)$ such that $rwd(G, Q)$ is the maximal cardinality of a finite set Z such that $(V, edg, S) \models \rho(Z)$. Hence, an alternative formulation is

$rwd(G, Q)$ is the maximal cardinality of a finite subset Z of V such that $(V, edg, S) \models \rho(Z)$

where $\rho(Z)$ is the CMSO formula expressing that there exist finite subsets X, Y of V such that $\varphi \wedge \alpha(X, Y, Z)$ and $S_Q XY$ hold. We have here a unique CMSO expression that does not depend on k, but uses instead a "maximum cardinality" set predicate, *cf.* [24] □

An open problem consists in finding classes of graphs for which a CMSO or a $\leq$-MSO transduction can produce a separation structure yielding rank-width at most a given finite number k. The article [25] shows that from graphs of bounded linear clique-width (equivalently of bounded linear rank-width), a CMSO transduction can build a clique-width term of width $f(k)$. This term can be converted into a layout witnessing bounded rank-width by an MSO-transduction.

4.3. Partial quasi-trees

When considering a class of relational structures, it is natural to investigate their induced substructures (see Rem. 2.12(3)).

Definition 4.28. *Partial quasi-trees as Induced Betweenness Relations in Quasi-Trees.*
Let $Q = (N, B)$ be a quasi-tree and $X \subseteq N$. Then $Q[X] := (X, B[X])$ where $B[X]$ is the restriction of B to X, is a *partial quasi-tree*. It is not necessarily a quasi-tree because some nodes $M(x, y, z)$ where $x, y, z \in X$ may not be in X. Its leaves are defined as for quasi-trees in Definition 4.4. An internal node of Q may become a leaf in $Q[X]$ but not vice-versa. In [11] we denoted by **IBQT** (for *Induced Betweenness in Quasi-Trees*) the class of partial quasi-trees. It is *hereditary* by definition, *i.e.*, it is closed under taking induced substructures. Theorem 3.1 of [11] states that it is axiomatized by an $\forall$FO sentence consisting of B1–B6 and the following

B8: $\neq xyzu \wedge Bxyz \wedge \neg Ayzu \Longrightarrow Bxyu$.

A pair $(N, \emptyset)$ is a "degenerated" partial quasi-tree obtained by deleting all internal nodes of a leafy quasi-tree. It satisfies B1–B6 and B8, and each node in N is a leaf.

We write $Q' \longrightarrow Q$ if Q' is a quasi-tree and Q is a partial quasi-tree obtained from Q by deleting some internal nodes. It follows that $L_Q = L_{Q'}$. It is not possible to reconstruct Q' from Q in a unique way, even if every node of Q belongs to some triple of B. Here is an example.

Example 4.29. Let Q' be the quasi-tree (N', B') shown (as a finite tree) in Figure 10. If we delete the nodes c and e, we get a partial quasi-tree $Q = (N, B)$ whose nodes are a, b, d, f, g, h and its betweenness relation consists of $Babd, Babf, Bfgh, Bdgh$ and the triples derived from B^+abgh. We have $Q' \longrightarrow Q$.

However, we also have $Q'' \longrightarrow Q$ where Q'' is associated with the tree obtained from that of Figure 10 by fusing c and e (*i.e.*, by contracting the edge $c - e$). However, we have $S_{Q'}adfh$ and $E_{Q''}adfh$, whence $\neg S_{Q''}adfh$.

The separation relations $S_{Q'}$ and $S_{Q''}$ (on L_Q) distinguish Q'' from Q'. They keep the "memory" of the possible quasi-trees from which Q is derived.

Theorem 4.30. *Let (N, B, S) be a relational structure such that B is ternary and S is 4-ary. There is at most one leafy quasi-tree $Q' = (N', B')$ such that $Q' \longrightarrow Q = (N, B)$, $L_Q = L_{Q'}$ and $S_{Q'} = S$. The existence of such a pair is FO expressible. If it does exist, one can construct Q' by an $\leq$-MSO transduction, and by a CMSO one if L is finite.*

Hence, $Q = (N, B)$ is a partial quasi-tree derived from a unique leafy quasi-tree Q' specified by a 4-ary relation S on L_Q defined as in Definition 4.4.

Lemma 4.31. *Let $Q = (N, B)$ be a leafy quasi-tree such that S_Q is not empty. Each leaf belongs to some 4-tuple of S_Q.*

Proof. Assume we have S_Qxyuv. Let z be a leaf not among x, y, u, v. By Property S4, we have $E_Qxyzu \vee S_Qxyzu \vee S_Qxzyu \vee S_Qxuyz$. The last three cases give the result, and $Exyzu \wedge S_Qxyuv$ yields S_Qszuv by S5 of Theorem 4.18. □

Proof of Theorem 4.30. Let (N, B, S) and (Q, Q') be as in the statement.

Case 1. $S \neq \emptyset$. By Lemma 4.31, $L_Q = L_{Q'}$ is determined from $S = S_{Q'}$. It follows from Proposition 4.13 that Q' is determined in a unique way. So is Q since $N - L_Q$ is the set of nodes y such that $Bxyz$ holds for some leaves x, z.

Case 2. $S = \emptyset, B \neq \emptyset$. Then, Q' is a star with center r and $Q' = Q$. Then B_Q contains all triples (x, r, y) for distinct x, y in L.

Case 3. $S = \emptyset, B = \emptyset$. Here, Q' is a star and Q is Q' minus the center.

The other properties follow from these observations and from Theorems 4.18 and 4.19 for Case 1. □

It follows that if (L, S) and (L, S') are the leaf structures of two leafy quasi-trees Q and Q' and if $S \subseteq S'$, then $S \subseteq S'$ and $Q \simeq Q'$. We may have $S = \emptyset$.

4.4. Topological embeddings

In order to illustrate topologically the notion of separation in a partial quasi-tree, we recall from [11] a characterization of quasi-trees and their induced substructures. We review the relevant definitions from that article.

Definition 4.32. *Topological embeddings of quasi-trees.*

(1) A set $\mathcal{L}$ of half-lines (we mean half *straight* lines) in the plane is a *tree of half-lines* if it satisfies the following conditions:

its union $\mathcal{L}^{\#}$ is connected,
any point x belonging to two half-lines is the end of at least one of them,
and no circle embeds homeomorphically into $\mathcal{L}^{\#}$.

See Figure 11(a).

(2) Let N be a countable subset of $\mathcal{L}^{\#}$ for such a tree $\mathcal{L}$. For distinct x, y, z in N, we let $B_{\mathcal{L}}xyz$ mean that y is on the unique path in the topological space $\mathcal{L}^{\#}$, denoted by $[x, z]_{\mathcal{L}}$, where a *path* is a union of finitely many line segments globally homeomorphic to the real interval [0,1].

(3) The structure $(N, B_{\mathcal{L}})$ is a quasi-tree by Theorem 4.4 of [11]. □

Examples are shown in Figure 11. Part (a) shows the embedding of a quasi-tree Q whose nodes are $a, b, c, d, e, f, g, u, v, w$ and x. Its leaves are a, c, d, e and g. It is the quasi-tree of a finite tree with root u. Part (b) shows an induced substructure of it where nodes u, v, w and x have been deleted (*cf.* Definition 4.32).

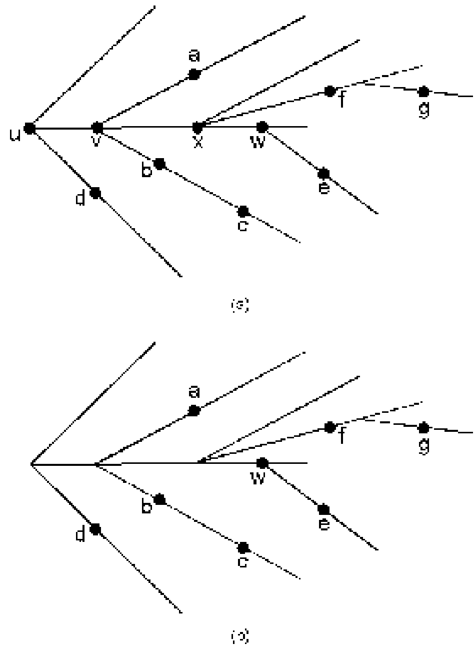

FIGURE 11. See Definition 4.32.

Theorem 4.4 of [11] shows also that every partial quasi-tree is isomorphic to $(N, B_{\mathcal{L}})$ for a tree $\mathcal{L}$ of half-lines.

Lemma 4.33. *In a quasi-tree $Q = (N, B_{\mathcal{L}})$ every internal node $M_Q(x, y, z)$ is a point m of $\mathcal{L}^{\#}$ that belongs two or three half-lines and is the origin of at least one of them.*

Proof. If x, y, z would be on a same line, then $M_Q(x, y, z)$ would be undefined. Hence, m must belong to two or three half-lines, and be the origin of at least one of them. It may be the origin of two or three of them. □

An example in Figure 11(a) is $v = M_Q(a, b, d)$, belonging to three half-lines.

Let $Q = (N, B_{\mathcal{L}})$ be a leafy quasi-tree embedded in a tree $\mathcal{L}$ of half lines. Let c be a point in $\mathcal{L}^{\#}$ not in N. The topological space $\mathcal{L}^{\#} - \{c\}$ consists of two connected components. Any two nodes x, y are in a same component, written $x \sim_c y$ if and only if $c \notin [x, y]_{\mathcal{L}}$. The two equivalence classes are denoted by X_c and Y_c; they form a cut of Q. With this notation:

Proposition 4.34. *If X_c and Y_c contain at least two leaves, they define as follows a 4-ary relation on pairwise distinct leaves:*

$S_c xyzu :\Longleftrightarrow$ *x and y are in an equivalence class and z and u are in the other.*

The separation relation S of Q is the union of the relations S_c.

Proof. If $S_c xyzu$ holds, then the path in $\mathcal{L}^{\#}$ between x and z must go through c. We have $c \in [x, z]_{\mathcal{L}} \cap [y, z]_{\mathcal{L}} \cap [x, u]_{\mathcal{L}} \cap [y, u]_{\mathcal{L}}$. Since x and y are in the same connected component of $\mathcal{L}^{\#} - \{c\}$, the path $[x, y]_{\mathcal{L}}$ does not go through c. Hence $M(x, y, z)$ belongs to this path and $x \sim_c M(x, y, z) \sim_c y$. Similarly $M(x, z, u)$ is in the other connected component hence $M(x, y, z) \neq M(x, z, u)$ and we have $Sxyzu$. Note that $c \in [M(x, y, z), M(x, z, u)]_{\mathcal{L}} - N$.

Conversely, assume $Sxyzu$. There is a point c not in N on the path $[M(x, y, z), M(x, z, u)]_{\mathcal{L}}$ because this path consists of finitely many segments forming a connectred subset of $\mathcal{L}^{\#}$. It follows that $x, M(x, y, z)$ and y

are in one of the connected components of $\mathcal{L}^{\#} - \{c\}$ and $z, M(x,y,z)$ and u are in the other. Hence we have $S_c xyzu$. □

Remark 4.35. From an embedding of a quasi-tree Q in a tree of half-lines $\mathcal{L}$, we get an embedding of any partial quasi-tree included in Q. If Q is leafy and we delete all internal nodes in the embedding, the induced betweenness relation $B_Q[L]$ is empty and gives no information about the resulting embedding, as L is a set without any structure. However, the separation relation S_Q describes, up to homeomorphism, the part of a topological space $\mathcal{L}^{\#}$ that represents the relative positions of the nodes $M(x,y,z)$ in the original embedding.

5. Conclusion

We have axiomatized relations that make possible to characterize leafy join-trees and quasi-trees in terms of their leaves. For certain cases, we could construct CMSO-transductions that map a leaf or a separation structure to the corresponding O-tree or quasi-tree.

The main open question is to obtain a readable ∀FO axiomatization of the separation structures of leafy quasi-trees.

Acknowledgments

I thank M. Bojanczyk for the nice construction of Theorem 3.4 that suggested me several extensions to infinite join-trees and quasi-trees, and M. Kanté for his comments. I thank M. Raskin for his help with the FO prover Z3.

Funding

This research received no external funding.

Conflicts of interest

The authors have nothing to disclose.

Data availability statement

This article has no associated data generated.

References

[1] R. Fraïssé, Theory of relations. *Studies in Logic*, Vol. 145. North-Holland (2000).
[2] B. Courcelle and C. Delhommé, The modular decomposition of countable graphs. Definition and construction in monadic second-order logic. *Theor. Comput. Sci.* **394** (2008) 1–38.
[3] M. Bojanczyk, *The category of MSO transductions* (2023) arXiv:230518039.
[4] R. Campbell *et al.*, CMSO-transducing tree-like graph Decompositions (2025) arXiv:2412.04970.
[5] B. Courcelle, The monadic second-order logic of graphs X: linear orderings. *Theor. Comput. Sci.* **160** (1996) 87–143.
[6] B. Courcelle, Several notions of rank-width for countable graphs. *J. Comb. Theory B* **123** (2017) 186–214.
[7] I. Kriz and R. Thomas, Clique-sums, tree-decompositions and compactness. *Discret. Math.* **81** (1990) 177–185.
[8] M. Changat, P. Narasimha-Shenoi and G. Seethakuttyamma, Betweenness in graphs: a short survey on shortest and induced path betweenness. *AKCE Int. J. Graphs Combinat.* **16** (2019) 96–109.
[9] V. Chvatal, Antimatroids, betweenness, convexity, in *Research Trends in Combinatorial Optimization*, Springer (2008) 57–64.
[10] B. Courcelle, Betweenness of partial orders. *RAIRO Theor. Informatics Appl.* **54** (2020) 7.
[11] B. Courcelle, Axiomatization of betweenness in order-theoretic trees. *Log. Methods Comput. Sci.* **17** (2021).
[12] A. Blumensath and B. Courcelle, Monadic second-order definable graph orderings. *Log. Methods Comput. Sci.* **10** (2014).
[13] T. Ganzow and S. Rubin, Order-invariant MSO is stronger than counting MSO in the finite, *STACS 2008, 25th Annual Symposium on Theoretical Aspects of Computer Science, Bordeaux, France. Proceedings LIPIcs 1, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Germany* (2008) 313–324.

[14] B. Courcelle, Algebraic and logical descriptions of generalized trees. *Log. Methods Comput. Sci.* **13** (2017).
[15] B. Courcelle, Induced betweenness in order-theoretic trees. *Discret. Math. Theor. Comput. Sci.* **23** (2021).
[16] B. Courcelle, Fundamental properties of infinite Trees. *Theor. Comput. Sci.* **25** (1983) 95–169.
[17] B. Courcelle, The monadic second-order logic of graphs XVI: canonical graph decompositions. *Log. Methods Comput. Sci.* **2** (2006).
[18] The Z3 prover. https://www.microsoft.com/en-us/research/project/z3-3/.
[19] B. Courcelle and J. Engelfriet, Graph Structure and Monadic Second-order Logic, A Language Theoretic Approach. Cambridge University Press (2012).
[20] B. Courcelle, On first-order transductions of relational structures (2026) in preparation.
[21] W. Hodges, Model Theory. Cambridge University Press (1993).
[22] D. Kim and S. Oum, Vertex-minors of graphs: a survey. *Discret. Appl. Math.* **351** (2024) 54–73.
[23] B. Courcelle and S. Oum, Vertex-minors, monadic second-order logic, and a conjecture by Seese. *J. Comb. Theory B* **97** (2007) 91–126.
[24] B. Courcelle and I. Durand, Computations by fly-automata beyond monadic second-order logic. *Theor. Comput. Sci.* **619** (2016) 32–67.
[25] M. Bojanczyk, M. Grohe and M. Pilipczuk, Definable decompositions for graphs of bounded linear cliquewidth. *Log. Methods Comput. Sci.* **17** (2021) Paper No. 5.
[26] Y. Chen and J. Flum, Forbidden induced subgraphs and the Loś-Tarski theorem. *J. Symb. Log.* **89** (2024) 516–548.