



Algebraic and logical descriptions of generalized trees

Bruno Courcelle

► To cite this version:

| Bruno Courcelle. Algebraic and logical descriptions of generalized trees. 2016. hal-01299077v1

HAL Id: hal-01299077

<https://hal.science/hal-01299077v1>

Preprint submitted on 7 Apr 2016 (v1), last revised 2 Oct 2016 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Algebraic and logical descriptions of generalized trees

Bruno COURCELLE
LaBRI, CNRS,
351 Cours de la Libération,
33405 Talence, France
courcell@labri.fr

April 6, 2016

*Abstract*¹ : *Quasi-trees* generalize trees in that the unique "path" between two nodes may be infinite and have any countable order type. They are used to define the rank-width of a countable graph in such a way that it is equal to the least upper-bound of the rank-widths of its finite induced subgraphs. *Join-trees* are the corresponding directed trees. They are useful to define the modular decomposition of a countable graph. We also consider *ordered join-trees*, that generalize rooted trees equipped with a linear order on the set of sons of each node. We define algebras with finitely many operations that generate (via infinite terms) these generalized trees. We prove that the associated regular objects (those defined by regular terms) are exactly the ones that are the unique models of monadic second-order sentences. These results use and generalize a similar result by W. Thomas for countable linear orders.

Keywords : Rank-width, quasi-tree, join-tree, ordered tree, algebra, regular term, monadic second-order logic.

1 Introduction

We define and study countable *generalized trees* such that the unique "path" between two nodes may be infinite and have any order type, in particular that of rational numbers. Our motivation comes from the notion of *rank-width*, a

¹This work has been supported by the French National Research Agency (ANR) within the IdEx Bordeaux program "Investments for the future", CPU, ANR-10-IDEX-03-02.

complexity measure of finite graphs investigated first in [Oum] and [OumSey]. Rank-width is based on graph decompositions formalized with finite subcubic trees. In order to extend rank-width to countable graphs in such a way that *the compactness property* holds, i.e., that the rank-width of a countable graph is the least upper-bound of those of its finite induced subgraphs, we base decompositions on *subcubic quasi-trees* [Cou14]. (For a comparison, the natural extension of tree-width to countable graphs has the compactness property [KriTho] without needing quasi-trees.) *Join-trees* can be seen as directed quasi-trees. A join-tree is a partial order (N, \leq) such that every two elements have a least upper-bound (called their *join*) and each set $\{y \mid y \geq x\}$ is linearly ordered. The modular decomposition of a countable graph is based on an *ordered join-tree* [CouDel].

Our objective is to obtain finitary descriptions (usable in algorithms) of join-trees, ordered join-trees and quasi-trees. For this purpose we will define algebras of such generalized trees that use finitely many operations and such that the finite and infinite terms over these operations define all countable relevant generalized trees. The *regular objects* are those defined by *regular terms*, i.e. that have finitely many different subterms, equivalently, that are the solutions of finite equation systems. We will prove that a generalized tree is regular if and only if it is *monadic second-order definable*, i.e., is the unique model (up to isomorphism) of a monadic second-order sentence.

A linear order whose elements are labelled by letters from an alphabet is called an *arrangement*. Regular arrangements were defined and studied in [Cou78] and [Hei], and their monadic second-order definability is proved in [Tho86]. We use the latter result for proving our extensions of it to generalized trees.

In Section 2 we review definitions. In Section 3 we first study binary join-trees and then, we extend the definitions and results concerning them to all join-trees. In Section 4, we study ordered join-trees, and in Section 5, we study quasi-trees. An appendix reviews definitions and facts about monadic second-order logic and the application of quasi-trees to the rank-width of countable graphs. An introductory article on these results is [Cou15].

2 Definitions, notation and basic facts.

All sets, trees and logical structures are finite or countably infinite. Isomorphism of ordered sets, trees and other logical structures is denoted by \simeq .

For partial orders $\leq, \preceq, \sqsubseteq, \dots$ we denote respectively by $<, \prec, \sqsubset, \dots$ the corresponding strict orders and $X < Y$ means that $x < y$ for every $x \in X$ and $y \in Y$.

Let (V, \leq) be a partial order. The least upper bound of x and y is denoted by $x \sqcup y$ if it exists and is called their *join*. A *line* is a subset Y of V that is linearly ordered and satisfies the following *convexity property*: if $x, z \in Y$,

$y \in V$ and $x \leq y \leq z$, then $y \in Y$. Particular notations for convex sets (not necessarily linearly ordered) are $[x, y]$ denoting $\{z \mid x \leq z \leq y\}$, $]x, y]$ denoting $\{z \mid x < z \leq y\}$, $] - \infty, x]$ denoting $\{y \mid y \leq x\}$ (even if V is finite), $]x, +\infty[$ denoting $\{y \mid x < y\}$ etc. If $X \subseteq V$, then $\downarrow(X)$ is the union of the sets $] - \infty, x]$ for x in X .

The first infinite ordinal and the linear order (\mathbb{N}, \leq) are denoted by ω .

The restriction of a relation R or a function f defined on a set V to a subset W is denoted by $R \upharpoonright W$ or $f \upharpoonright W$ respectively.

Monadic second-order logic (*MS logic* in short) logic is reviewed in the appendix. *MS-definable* and *MS-expressible* mean respectively definable and expressible in monadic second-order logic. A *sentence* is a formula without free variables.

2.1 Trees

A *tree* is a possibly empty, finite or countable, undirected graph that is connected and has no cycles. Hence, it has neither loops nor parallel edges. The set of nodes of a tree T is N_T .

A *rooted tree* is a nonempty tree equipped with a distinguished node called its *root*. The *level* of a node x is the number of edges of the path between it and the root and $Sons(x)$ denotes the set of its sons. We define on N_T the partial order \leq_T such that $x \leq_T y$ if and only if y is on the unique path between x and the root. The least upper bound of x and y , denoted by $x \sqcup_T y$ is their least common ancestor. We will specify a rooted tree T by (N_T, \leq_T) and we will omit the index T when T is clear. If x is a node of T , then T/x is the *subtree issued from x* , defined as $(N_{T/x}, \leq_T \upharpoonright N_{T/x})$ where $N_{T/x} :=] - \infty, x]$.

A partial order (N, \leq) is (N_T, \leq_T) for some rooted tree T if and only if it has a largest element *max* and for each $x \in N$, the set $[x, \text{max}]$ is finite and linearly ordered. These conditions imply that any two nodes have a join.

An *ordered tree* is a rooted tree such that each set $Sons(x)$ is linearly ordered by an order \sqsubseteq_x .

2.2 Finite and infinite terms

Let F be a finite set of operations f , each given with an arity $\rho(f)$. We call such a set a *signature*. The maximal arity of a symbol is denoted by $\rho(F)$. A *term over F* is finite or infinite. We denote by $T^\infty(F)$ the set of all terms over F and by $T(F)$ the set of finite ones. A typical example of an infinite term, easily describable linearly, is, with f binary and a and b nullary, the term $t_\infty := f(a, f(b, f(a, f(b, f(\dots\dots\dots))))))$ that is the unique solution in $T^\infty(F)$ of the equation $t = f(a, f(b, t))$.

Positions in terms are designated by Dewey words. The set $Pos(t)$ of positions of a term t is ordered by \leq_t , the reversal of the prefix order; it can be seen as a labelled ordered tree whose set of nodes is $Pos(t)$.

There is on $T^\infty(F)$ a canonical structure of F -algebra of which $T(F)$ is a subalgebra. If $\mathbb{M} = \langle M, (f_{\mathbb{M}})_{f \in F} \rangle$ is an F -algebra, a *value mapping* is a homomorphism $h : T^\infty(F) \rightarrow \mathbb{M}$. Its restriction to finite terms is uniquely defined.

In some cases, we will use algebras with two sorts. The corresponding modifications of the definitions are straightforward.

The partial order on terms.

Let F contain a special nullary symbol Ω intended to be the least term. One defines on $T(F)$ a partial order \leq as follows:

$$\begin{aligned} &\Omega \leq t \text{ for any } t \in T(F), \\ &f(t_1, \dots, t_k) \leq g(t'_1, \dots, t'_{k'}) \text{ if and only if } f = g, \text{ whence } k' = k, \text{ and} \\ &t_i \leq t'_i \text{ for each } i = 1, \dots, k. \end{aligned}$$

For terms in $T^\infty(F)$, the definition (subsuming the previous one) is :

$$t \leq t' \text{ if and only if } \text{Pos}(t) \subseteq \text{Pos}(t') \text{ and every occurrence in } t \text{ of a symbol in } F - \{\Omega\} \text{ is an occurrence in } t' \text{ of the same symbol (and an occurrence in } t \text{ of } \Omega \text{ is an occurrence in } t' \text{ of any symbol)}.$$

Every increasing sequence of terms has a least upper bound. More details on terms can be found in [Cou83].

Regular terms

A term $t \in T^\infty(F)$ is *regular* if there is a mapping h from $\text{Pos}(t)$ into a finite set Q and a mapping $\tau : Q \rightarrow F \times \text{Seq}(Q)$ ($\text{Seq}(Q)$ is the set of finite sequences of elements of Q) such that:

$$\begin{aligned} &\text{if } u \text{ is an occurrence of a symbol } f \text{ of arity } k, \text{ then } \tau(h(u)) = \\ &(f, (h(u_1), \dots, h(u_k))) \text{ where } (u_1, \dots, u_k) \text{ is the sequence of sons of } u. \end{aligned}$$

Intuitively, τ is the transition function of a top-down deterministic automaton with set of states Q ; $h(\varepsilon)$ is its initial (root) state and h defines its unique run. This is equivalent to requiring that t has finitely many different subterms, or is a component of a finite system of equations that has a unique solution in $T^\infty(F)$. (The set Q can be taken as the set of unknowns of such a system, see [Cou83].) The above term t_∞ is regular.

With a term t we associate the relational structure $\lfloor t \rfloor := (\text{Pos}(t), \leq_t, (br_i)_{1 \leq i \leq \rho(F)}, (lab_f)_{f \in F})$ where $br_i(u)$ is true if and only if u is the i -th son of his father and $lab_f(u)$ is true if and only if f occurs at position u . This term can be reconstructed in a unique way from any relational structure isomorphic to $\lfloor t \rfloor$.

A term t is regular if and only if $\lfloor t \rfloor$ is MS-definable, i.e., is, up to isomorphism, the unique model of a monadic second-order sentence (see Thomas, [Tho90]).

2.3 Arrangements and labelled sets

We review a notion introduced in [Cou78] and further studied in [Hei, Tho86]. Let X be a set. A linear order (V, \leq) equipped with a labelling mapping $lab : V \rightarrow X$ is called an *arrangement over X* . It is *simple* if lab is injective. We denote by $\mathcal{A}(X)$ the set of arrangements over X .

An arrangement over a finite set X can be considered as a generalized word. A linear order (V, \leq) is identified with the simple arrangement (V, \leq, Id_V) such that $Id_V(v) := v$ for each $v \in V$. In the sequel, Id will denote the identity function on any set.

An *isomorphism of arrangements* $i : (V, \leq, lab) \rightarrow (V', \leq', lab')$ is an order preserving bijection $i : V \rightarrow V'$ such that $lab' \circ i = lab$. Isomorphism is denoted by \simeq .

If $w = (V, \leq, lab) \in \mathcal{A}(X)$ and $r : X \rightarrow Y$, then, $\bar{r}(w) := (V, \leq, r \circ lab)$ is an arrangement over Y . If $r : V \rightarrow Y$, then $\bar{r}((V, \leq))$ is the arrangement (V, \leq, r) over Y since we identify (V, \leq) to the simple arrangement (V, \leq, Id) .

The concatenation of linear orders yield a concatenation of arrangements denoted by \bullet . We denote by Ω the empty arrangement and by a the one reduced to a single occurrence of $a \in X$. Clearly, $w \bullet \Omega = \Omega \bullet w = w$ for every $w \in \mathcal{A}(X)$. The infinite word $w = a^\omega$ is the arrangement over $\{a\}$ with underlying order ω ; it is described by the equation $w = a \bullet w$. Similarly, the arrangement $w = a^\eta$ over $\{a\}$ with underlying linear order (\mathbb{Q}, \leq) (that of rational numbers) is described by the equation $w = w \bullet (a \bullet w)$. We will generalize arrangements to tree structures.

Let X be a set of variables and $t \in T^\infty(\{\bullet, \Omega\} \cup X)$. Hence, $Pos(t) \subseteq \{1, 2\}^*$. The *value* of t is the arrangement $val(t) := (Occ(t, X), \leq_{lex}, lab)$ where $Occ(t, X)$ is the set of positions of elements of X and $lab(u)$ is the symbol occurring at position u . We say that t *denotes* w if w is isomorphic to $val(t)$.

For an example, $t_\bullet := \bullet(a, \bullet(b, \bullet(a, \bullet(b, \bullet(\dots\dots\dots))))))$ denotes the infinite word $abab\dots$. Its value is defined from $Occ(t_\bullet, \{a, b\}) = 2^*\mathbf{1}$ lexicographically ordered by $1 < 2\mathbf{1} < 22\mathbf{1} < \dots$, $lab(2^i\mathbf{1}) = a$ if i is even and $lab(2^i\mathbf{1}) = b$ if i is odd. The arrangements a^ω and a^η are denoted by the terms that are respectively the unique solutions in $T^\infty(\{\bullet, \Omega, a\})$ of the equations $w = a \bullet w$ and $w = w \bullet (a \bullet w)$.

An arrangement is *regular* if it is denoted by a regular term. The term t_\bullet is regular and the arrangements a^ω and a^η are regular.

An arrangement is regular if and only if it is a component of the *initial solution of a regular system of equations* over F [Cou78] or the value of a *regular expression* in the sense of [Hei]. We will use the result of [Tho86] that an arrangement over a finite alphabet is regular if and only if is MS-definable. For this result, we represent an arrangement $w = (V, \leq, lab)$ over X by the relational structure $[w] := (V, \leq, (lab_a)_{a \in X})$ where $lab_a(u)$ is true if and only if $lab(u) = a$.

An X -labelled set is a pair $m = (V, lab)$ where $lab : V \rightarrow X$, equivalently, a relational structure $(V, (lab_a)_{a \in X})$ where each element of V belongs to a unique set lab_a . We denote by $set(w)$ the X -labelled set obtained by forgetting the linear order of an arrangement w over X . Up to isomorphism, an X -labelled set m is defined by the cardinalities in $\mathbb{N} \cup \{\omega\}$ of the sets lab_a , hence is a finite or countable *multiset of elements of X* : a number in $\mathbb{N} \cup \{\omega\}$ associated with each $a \in X$ and represents its number of occurrences in m .

If X is finite, each X -labelled set is MS_{fin} -definable, that is, is the unique (finite or countably infinite) model up to isomorphism of a sentence of monadic second-order logic extended with a set predicate $Fin(U)$ expressing that the set U is finite. (See the appendix for details). It is also *regular*, hence, is $set(val(t))$ for some regular term in $T^\infty(\{\bullet, \Omega\} \cup X)$. The notion of regularity is thus trivial for X -labelled sets when X is finite.

3 Join-trees

Join-trees have been used in [CouDel] for defining the modular decomposition of countable graphs.

3.1 Join-trees, join-forests and their structurings

(3.1) *Definition : Join-tree.*

(a) A *join-tree* is a pair $J = (N, \leq)$ such that:

- 1) N is a possibly empty, finite or countable set called the set of *nodes*,
- 2) \leq is a partial order on N such that, for every node x , the set $[x, +\infty[$ (the set of nodes $y \geq x$) is linearly ordered,
- 3) every two nodes x and y have a join $x \sqcup y$.

A minimal node is a *leaf*. If N has a largest element, we call it the *root* of J . The set of strict upper-bounds of a nonempty set $X \subseteq N$ is a line L . If L has a smallest element, we denote it by \hat{X} and we say that \hat{X} is *the top* of X . Note that $\hat{X} \notin X$.

(b) A *join-forest* is a pair $J = (N, \leq)$ that satisfies conditions 1), 2) and the following weakening of 3):

- 3') if two nodes have an upper bound, they have a join.

The relation that two nodes have a join is an equivalence. Let N_s for $s \in S$ be its equivalence classes and $J_s := (N_s, \leq \upharpoonright N_s)$, more simply denoted by (N_s, \leq)

by leaving implicit the restriction to N_s . Then each J_s is a join-tree, and J is the union of these pairwise disjoint join-trees, called its *components*.

(c) A join-forest $J = (N, \leq)$ is *included in* a join-forest $J' = (N', \leq')$, denoted by $J \subseteq J'$ if $N \subseteq N'$, \leq is $\leq' \upharpoonright N$ and \sqcup is $\sqcup' \upharpoonright N$.

(3.2) *Definitions : Direction and degree.*

Let $J = (N, \leq)$ be a join-forest, and x be one of its nodes. Let \sim be the equivalence relation on $] -\infty, x[$ such that $z \sim y$ if and only if $z \sqcup y < x$. Each equivalence class C is called a *direction of J relative to x* . We have $\widehat{C} = x$. The set of directions relative to x is denoted by $Dir(x)$ and the *degree* of x is the number of its directions. The leaves are the nodes of degree 0.

A join-tree is *binary* if its nodes have degree at most 2. We call it a *BJ-tree* for short.

(3.3) *Definition : Structured join-trees and join-forests.*

(a) Let $J = (N, \leq)$ be a join-tree. A *structuring* of J is a set \mathcal{U} of nonempty lines forming a partition of N that satisfies some conditions, stated with the following notation : if $x \in N$, then $U(x)$ denotes the line of \mathcal{U} containing x , $U_-(x) := U(x) \cap] -\infty, x[$ and $U_+(x) := U(x) \cap [x, +\infty[$. (The set $[x, +\infty[$ has no top but it can have a greatest element). The conditions are:

- 1) exactly one line A of \mathcal{U} is *upwards closed* (i.e., $[x, +\infty[\subseteq A$ if $x \in A$), hence, has no strict upper-bound and no top; we call it the *axis*; each other line U has a top \widehat{U} ,
- 2) for each x in N , the sequence y_0, y_1, y_2, \dots such that $y_0 = x$, $y_{i+1} = \widehat{U(y_i)}$ is finite. Its last element is $y_k \in A$ (y_{k+1} is undefined). We call k the *depth* of x .

The nodes on the axis are those at depth 0. The lines $[y_i, y_{i+1}[$ for $i = 0, \dots, k-1$ and $[y_k, +\infty[$ are convex subsets of pairwise distinct lines of \mathcal{U} . We have $[x, +\infty[= [y_0, y_1[\cup [y_1, y_2[\cup \dots \cup [y_k, +\infty[$, where $[y_i, y_{i+1}[= U_+(y_i)$ for each $i < k$, $[y_k, +\infty[= U_+(y_k) \subseteq A$ and the depth of y_i is $k - i$.

We call such a triple (N, \leq, \mathcal{U}) a *structured join-tree*, an *SJ-tree* for short.

Every linear order is an SJ-tree whose elements are all of depth 0.

Remark : If $x < A$ for some x , then A has a smallest element, which is the node y_k of Condition 2) (because if $z \in A$ is smaller than y_k , then $x < z$, which contradicts the observation that $[y_{k-1}, y_k[\subseteq U(y_{k-1})$ and $U(y_{k-1}) \cap A = \emptyset$).

(b) Let $J = (N, \leq)$ be a join-forest whose components are J_s , $s \in S$. A *structuring* of J is a set \mathcal{U} of nonempty lines forming a partition of N such that, if \mathcal{U}_s is the set of lines of \mathcal{U} included in N_s (every line of \mathcal{U} is included in some N_s), then each triple $(N_s, \leq, \mathcal{U}_s)$ is a structuring of J_s .

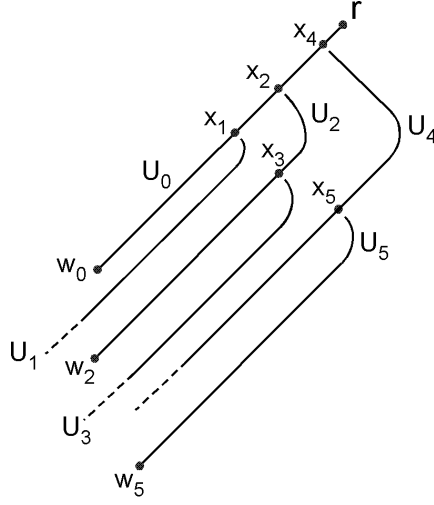


Figure 1: A structured binary join-tree.

(3.4) *Example* : Figure 1 shows a structuring of a binary join-tree $J = (N, \leq, \{U_0, \dots, U_5\})$ such that $A = U_0$. The directions relative to x_2 are $U_-(x_2) \cup U_1$ and $U_2 \cup U_3$. The maximal depth of a node is 2.

(3.5) **Proposition** : Every join-tree and every join-forest has a structuring.

Proof : Let $J = (N, \leq)$ be a join-tree. Let us choose an enumeration of N and a maximal line B_0 ; it is upwards closed. For each $i > 0$, we choose a maximal line B_i containing the first node not in $B_{i-1} \cup \dots \cup B_0$. We define $U_0 := B_0$ and, for $i > 0$, $U_i := B_i - (U_{i-1} \cup \dots \cup U_0) = B_i - (B_{i-1} \cup \dots \cup B_0)$. We define \mathcal{U} as the set of lines U_i . It is a structuring of J . The axis is U_0 .

If J is a join-forest, it has a structuring that is the union of structurings of its components. \square

Remark: Since each line B_i is maximal, if U_i has smallest element, this element is a node of degree 0 in J .

In view of our use of monadic second-order logic, we give a description of SJ-trees by relational structures.

(3.6) *Definition* : *SJ-trees as relational structures.*

(a) If $J = (N, \leq, \mathcal{U})$ is an SJ-tree, we define $S(J)$ as the relational structure (N, \leq, N_0, N_1) such that N_0 is the set of nodes at even depth and $N_1 := N - N_0$

is the set of those at odd depth. (N_0 and N_1 are sets but we consider them also as unary relations).

(b) Let $J = (N, \leq)$ be a join-tree and $X \subseteq N$. We say that X is *laminar* if, for all $x, y, z \in X$, if $x \leq z, y \leq z$ and $[x, z] \cup [y, z] \subseteq X$, then x and y are comparable (the intervals $[x, z]$ and $[y, z]$ are relative to J). This condition implies that the lines of J that are included in X and are maximal with this condition form a partition of X whose parts will be called its *components*.

(3.7) **Proposition** : For J and $S(J)$ as above, the following properties hold:

- 1) the sets N_0 and N_1 are laminar, \mathcal{U} is the set of their components and the axis A is a component of N_0 ,
- 2) there is an MS formula $\varphi(N_0, N_1)$ expressing that a relational structure (N, \leq, N_0, N_1) is $S(J)$ for some SJ-tree $J = (N, \leq, \mathcal{U})$,
- 3) there exist MS formulas $\theta_{Ax}(X, N_0, N_1)$ and $\theta(u, U, N_0, N_1)$ expressing, respectively, in a structure $(N, \leq, N_0, N_1) = S(N, \leq, \mathcal{U})$, that X is the axis and that $U \in \mathcal{U} \wedge u = \widehat{U}$.

Proof : 1) is clear from the definitions.

2) That a partial order (N, \leq) is a join-tree is first-order expressible. The formula $\varphi(N_0, N_1)$ will include this condition. Let $J = (N, \leq)$ be a join-tree; let N be the union of two disjoint laminar sets N_0 and N_1 and \mathcal{U} be the set of their components. Then, $J = (N, \leq, \mathcal{U})$ is an SJ-tree and $S(J) = (N, \leq, N_0, N_1)$ if and only if:

- i) every component of N_1 has a top in N_0 and every component of N_0 except one has a top in N_1 ,
- ii) for each U in \mathcal{U} , the sequence U_0, U_1, \dots of lines of \mathcal{U} such that $U_0 = U$, $\widehat{U_0} \in U_1, \dots, \widehat{U_i} \in U_{i+1}$ terminates at some U_k that has no top, hence is included in N_0 .

These conditions are necessary. They rephrase Definition (3.3) hence are sufficient. The integer k in Condition ii) is the common depth of all nodes in U .

That a set X is laminar is MS-expressible, and one can build an MS-formula $\psi(U, X)$ expressing that U is a component of X assumed to be laminar. This formula can be used to express that N is the union of two disjoint laminar sets N_0 and N_1 that satisfy Conditions i) and ii). For expressing Condition ii), we define for each U in \mathcal{U} a set of nodes W : it is the least set such that $\widehat{U} \in W$, and, for each $w \in W$, the top of $U(w)$ belongs to W if it is defined (where $U(w)$ is the unique set in \mathcal{U} that contains w). The set W is linearly ordered (it consists of $\widehat{U_0} < \dots < \widehat{U_i} \dots$) and Condition ii) says that it must be finite. To write the formula, we use the fact that the finiteness of a linearly ordered set is MS-expressible (see the Appendix).

3) The construction of φ actually uses the MS formulas $\theta_{Ax}(X, N_0, N_1)$ and $\theta(u, U, N_0, N_1)$. \square

3.2 Description schemes of structured binary join-trees

In order to introduce technicalities step by step, we first consider binary join-trees. They are actually sufficient for defining the rank-width of a countable graph. See Section 5 and the appendix.

(3.8) *Definition : Structured binary join-trees.*

Let $J = (N, \leq)$ be a binary join-tree. A *structuring* of J is a set \mathcal{U} of lines satisfying the conditions of Definition (3.3) and, furthermore:

- i) if the axis A has a smallest element, then its degree is 0 or 1,
- ii) each $x \in N$ is the top of at most one set $U \in \mathcal{U}$, denoted by U^x , and $U^x := \emptyset$ if x is the top of no $U \in \mathcal{U}$.

We call (N, \leq, \mathcal{U}) a *structured binary join-tree*, an *SBJ-tree* in short.

(3.9) **Proposition:** 1) Every BJ-tree J has a structuring.

2) The class of structures $S(J)$ for SBJ-trees J is monadic second-order definable.

Proof: 1) We use the construction of Proposition (3.5) for $J = (N, \leq)$. The remark following it implies that, if the axis $A = U_0$ has a smallest element, this element has degree 0. It implies also that, if $\widehat{U_i} = x$, then x cannot have degree 0 in the BJ-tree J_{i-1} induced by $U_{i-1} \cup \dots \cup U_0$ because each line B_i is chosen maximal, and it cannot have degree 2 or more in J_{i-1} because J is binary. Hence it has degree 1 in J_{i-1} . It follows that x is the top of no line U_j for $j < i$. Hence ii) holds and the construction yields an SBJ-tree (N, \leq, \mathcal{U}) .

2) The formula φ of Proposition (3.7) can easily be modified so as to express that (N, \leq, N_0, N_1) is $S(J)$ for some SBJ-tree J . \square

(3.10) *Definition : Description schemes for SBJ-trees.*

(a) A *description scheme for an SBJ-tree*, in short an *SBJ-scheme*, is a triple $\Delta = (Q, w_{Ax}, (w_q)_{q \in Q})$ such that Q is a set, $w_{Ax} \in \mathcal{A}(Q)$ (is an arrangement over Q) and $w_q \in \mathcal{A}(Q)$ for each q .

It is *regular* if Q is finite and the arrangements w_{Ax} and w_q are regular.

(b) We recall that a linear order (V, \leq) is identified with the arrangement (V, \leq, Id) . If $W \subseteq V$ and $r : V \rightarrow Q$, then $\bar{r}((W, \leq))$ is the arrangement $(W, \leq \upharpoonright W, r \upharpoonright W) \in \mathcal{A}(Q)$ that we will denote more simply by (W, \leq, r) leaving implicit the restrictions of \leq and r to W .

An SBJ-scheme Δ *describes* an SBJ-tree $J = (N, \leq, \mathcal{U})$ with axis A if there exists a mapping $r : N \rightarrow Q$ such that $\bar{r}((A, \leq)) \simeq w_{Ax}$ and $\bar{r}((U^x, \leq)) \simeq w_{r(x)}$ for every $x \in N$. We will also say that Δ *describes* the BJ-tree $fgs(J) := (N, \leq)$.

The mapping r need not be surjective, this means that some elements of Q and the corresponding arrangements may be useless, and thus can be removed from Δ .

(c) Each SBJ-tree $J = (N, \leq, \mathcal{U})$ has a *standard* description scheme $\Delta(J) := (N, (A, \leq), ((U^x, \leq))_{x \in N})$. The identity mapping $r : N \rightarrow N$ shows that $\Delta(J)$ describes J .

One can obtain as follows from $\Delta(J)$ a SBJ-scheme $\Delta := (Q, w_{Ax}, (w_q)_{q \in Q})$ that describes J :

- (i) one chooses a partition Q of N satisfying conditions (ii) and (iii) below, where for each $x \in N$, $r(x)$ denotes the part q of Q containing x ,
- (ii) for all nodes x in a part q of Q , the arrangements (U^x, \leq, r) are isomorphic to w_q ,
- (iii) $w_{Ax} \simeq (A, \leq, r)$.

(3.11) **Proposition:** Every SBJ-scheme Δ describes a unique SBJ-tree where unicity is up to isomorphism.

We denote this SBJ-tree by $Unf(\Delta)$, called the *unfolding* of Δ : see the remark following the proof.

Proof: Let $\Delta = (Q, w_{Ax}, (w_q)_{q \in Q})$ be an SBJ-scheme, defined with arrangements $w_{Ax} = (V_{Ax}, \preceq, lab_{Ax})$ and $w_q = (V_q, \preceq, lab_q)$ such that, without loss of generality, the sets V_{Ax} and V_q are pairwise disjoint and the same symbol \preceq denotes their orders.

We construct $(N, \leq, \mathcal{U}) = Unf(\Delta)$ as follows.

a) N is the set of finite nonempty sequences (v_0, v_1, \dots, v_k) such that:

$$v_0 \in V_{Ax}, v_i \in V_{q_i} \text{ for } 1 \leq i \leq k, \text{ where} \\ q_1 = lab_{Ax}(v_0), q_2 = lab_{q_1}(v_1), \dots, q_k = lab_{q_{k-1}}(v_{k-1}).$$

b) $(v_0, v_1, \dots, v_k) \leq (v'_0, v'_1, \dots, v'_j)$ if and only if $k \geq j$, $(v_0, v_1, \dots, v_{j-1}) = (v'_0, v'_1, \dots, v'_{j-1})$ and $v_j \preceq v'_j$.

c) The axis A is the set of one-element sequences (v) for $v \in V_{Ax}$; for $x = (v_0, v_1, \dots, v_k)$ we define $U(x)$ as the set of sequences $(v_0, v_1, \dots, v_{k-1}, v)$ for $v \in V_{q_k}$ so that we have $\widehat{U(x)} = (v_0, v_1, \dots, v_{k-1})$.

Note that $(v_0, \dots, v_k) < (v_0, \dots, v_j)$ if $j < k$ and that $(v_0, \dots, v_{k-1}, v_k) \leq (v_0, \dots, v_{k-1}, v)$ if and only if $v_k \preceq v$.

Claim : Δ describes J .

We define $r : N \rightarrow Q$ as follows:

if $x \in A$, then $x = (v)$ for some $v \in V_{Ax}$ and $r(x) := \text{lab}_{Ax}(v)$;
if $x \in N$ has depth $k \geq 1$, then $x = (v_0, v_1, \dots, v_k)$ for some v_0, v_1, \dots, v_k
as in a) and $r(x) := \text{lab}_{q_k}(v_k)$.

It follows that $\bar{r}((A, \leq)) \simeq w_{Ax}$ and that, for $x = (v_0, v_1, \dots, v_k)$ (of depth k), $\bar{r}((U^x, \leq)) \simeq w_{q_k} = w_{r(x)}$, which proves the claim. \square

We now prove unicity. Assume that Δ describes $J = (N, \leq, \mathcal{U})$ with axis A and also $J' = (N', \leq', \mathcal{U}')$ with axis A' , by means of mappings $r : N \rightarrow Q$ and $r' : N' \rightarrow Q$. We will construct an isomorphism $h : J \rightarrow J'$ as the common extension of bijections $h_k : N_k \rightarrow N'_k$ (where N_k (N'_k) is the set of nodes of J (of J') of depth at most k) such that :

they map \leq to \leq' ,
they map the lines of \mathcal{U} to those of \mathcal{U}' of same depth,
 $r' \circ h_k = r \upharpoonright N_k$.

Case $k = 0$. We have $\bar{r}((A, \leq)) = (A, \leq, r) \simeq w_{Ax} \simeq \bar{r}'((A', \leq)) = (A', \leq', r')$ which gives the order preserving bijection $h_0 : N_0 = A \rightarrow N'_0 = A'$ such that $r' \circ h_0 = r \upharpoonright N_0$.

Case $k > 0$. We assume inductively that h_{k-1} has been constructed.

Let $U \in \mathcal{U}$ be such that $x = \widehat{U}$ has depth $k - 1$; hence, $U \cap N_{k-1} = \emptyset$. Then $(U, \leq, r) \simeq w_{r(x)}$. Let $x' = h_{k-1}(x)$; we have $r'(x') = r(x)$. Hence there is $U' \in \mathcal{U}'$ such that $x' = \widehat{U'}$, $U' \cap N'_{k-1} = \emptyset$ and $(U', \leq', r') \simeq w_{r'(x')} = w_{r(x)}$. Hence, there is an order preserving bijection $h_U : U \rightarrow U'$ such that $r' \circ h_U = r \upharpoonright U$.

We define h_k as the extension of the injective mappings h_{k-1} and h_U for all $U \in \mathcal{U}$ such that the depth of \widehat{U} is $k - 1$. These mappings have pairwise disjoint domains whose union is N_k .

The extension to N of all these mappings h_k is the desired isomorphism h .

\square

Remark: We call *unfolding* the transformation of Δ into $Unf(\Delta)$ because it generalizes the unfolding of a directed graph G into a finite or countable rooted tree. The unfolding is done from a particular vertex s of G , and the nodes of the tree are the sequences of the form (x_0, \dots, x_k) such that $s = x_0$ and there is a directed edge in G from x_i to x_{i+1} , for each $i < k$. If Δ is such that the arrangements w_{Ax} and w_q are reduced to a single element, the corresponding directed graph has all its vertices of outdegree one and the tree resulting from the unfolding consists of one infinite path: the SBJ-tree $Unf(\Delta)$ is the order type ω^- of negative integers and the sets in \mathcal{U} are singletons.

(3.12) **Proposition:** A BJ-tree is monadic second-order definable if it is described by a regular BJ-scheme.

Proof: That $J = (N, \leq)$ is a BJ-tree is first-order expressible. Assume that $J = fgs(J')$ where $J' = (N, \leq, \mathcal{U}) \simeq Unf(\Delta)$ for some regular SBJ-scheme

$\Delta = (Q, w_{Ax}, (w_q)_{q \in Q})$ such that $Q = \{1, \dots, m\}$. Let r be the corresponding mapping: $N \rightarrow Q$ (cf. Definition (3.10.b)). For each $q \in Q$, let ψ_q be an MS sentence that characterizes w_q , up to isomorphism, by the main result of [Tho86]. Similarly, ψ_{Ax} characterizes w_{Ax} . We claim that a relational structure (X, \leq) is isomorphic to J if and only if there exist subsets $N_0, N_1, M_1, \dots, M_m$ of X such that:

- (i) (X, \leq) is a BJ-tree and $(X, \leq, N_0, N_1) = S(J'')$ for some SBJ-tree $J'' = (X, \leq, \mathcal{U}')$,
- (ii) (M_1, \dots, M_m) is a partition of X ; we let r' maps each $x \in X$ to the unique $q \in Q$ such that $x \in M_q$,
- (iii) for every q and node x in M_q , the arrangement $\overline{r'}((U^x, \leq))$ is isomorphic to w_q (where $U^x \in \mathcal{U}'$),
- (iv) the arrangement $\overline{r'}((A', \leq))$ over Q where A' is the axis of J'' is isomorphic to w_{Ax} .

Conditions (ii)-(iv) express that Δ describes J'' , hence that J'' is isomorphic to J' , and so, that $(X, \leq) \simeq fgs(J') = J$.

By Proposition (3.9), Condition (i) is expressed by an MS-formula $\varphi(N_0, N_1)$, and the property $U \in \mathcal{U} \wedge x = \widehat{U}$ is expressed in terms of N_0, N_1 by an MS-formula $\theta(x, U, N_0, N_1)$. Conditions (iii) and (iv) are expressed by means of the MS-sentences ψ_{Ax} and ψ_q suitably adapted to take $N_0, N_1, M_1, \dots, M_m$ as arguments. Hence, J is (up to isomorphism) the unique model of an MS sentence of the form:

$$\exists N_0, N_1. [\varphi(N_0, N_1) \wedge \exists M_1, \dots, M_m. \varphi'(N_0, N_1, M_1, \dots, M_m)]$$

where φ' expresses conditions (ii)-(iv). \square

Theorem (3.21) will establish a converse.

3.3 The algebra of binary join-trees

We define three operations on *structured binary join-trees* (SBJ-trees). The (finite and infinite) terms over these operations define all SBJ-trees.

(3.13) *Definition : Operations on SBJ-trees.*

Concatenation along axes.

Let $J = (N, \leq, \mathcal{U})$ and $J' = (N', \leq', \mathcal{U}')$ be disjoint SBJ-trees, with respective axes A and A' . We define:

$$\begin{aligned} J \bullet J' &:= (N \cup N', \leq'', \mathcal{U}'') \text{ where :} \\ x \leq'' y &: \iff x \leq y \vee x \leq' y \vee (x \in N \wedge y \in A'), \\ \mathcal{U}'' &:= \{A \cup A'\} \cup (\mathcal{U} - \{A\}) \cup (\mathcal{U}' - \{A'\}). \end{aligned}$$

$J \bullet J'$ is an SBJ-tree with axis $A \cup A'$; the depth of a node in $J \bullet J'$ is the same as in J or J' .

This operation generalizes the concatenation of linear orders: if (N, \leq) and (N', \leq') are disjoint linear orders, then the SBJ-tree $(N, \leq, \{N\}) \bullet (N', \leq', \{N'\})$ corresponds to the concatenation of (N, \leq) and (N', \leq') usually denoted by $(N, \leq) + (N', \leq')$.

If $K = (M, \leq, \mathcal{V})$ is an SBJ-tree with axis B , and $B = A \cup A'$ such that $A < A'$, then $K = J \bullet J'$ where $N = \downarrow(A)$, $N' = M - N$, \mathcal{U} is the set of lines of \mathcal{V} included in $N - A$, \mathcal{U}' is the set of lines of \mathcal{V} included in $N' - A'$ and the orders of J and J' are the restrictions of \leq to N and N' .

The empty SBJ-tree:

The nullary symbol Ω denotes the empty SBJ-tree.

Extension:

Let $J = (N, \leq, \mathcal{U})$ be an SBJ-tree, and $u \notin N$. Then:

$ext_u(J) := (N \cup \{u\}, \leq', \{\{u\}\} \cup \mathcal{U})$ where :

$x \leq' y \iff x \leq y \vee y = u$,

the axis is $\{u\}$.

Clearly, $ext_u(J)$ is an SBJ-tree. The depth of $v \in N$ is its depth in J plus 1. When handling SBJ-trees up to isomorphism, we use the notation $ext(J)$ instead of $ext_u(J)$. Note that the axis of J is turned into an ordinary line of $\{\{u\}\} \cup \mathcal{U}$ with top equal to u .

Forgetting structuring:

If J is an SBJ-tree as above, $fgs(J) := (N, \leq)$ is the underlying BJ-tree (binary join-tree).

Anticipating the sequel, we observe that a linear order $a_1 < \dots < a_n$, identified with the SBJ-tree $(\{a_1, \dots, a_n\}, \leq, \{\{a_1, \dots, a_n\}\})$ is defined by the term $t := ext_{a_1}(\Omega) \bullet ext_{a_2}(\Omega) \bullet \dots \bullet ext_{a_n}(\Omega)$. The binary (actually unary) join-tree $(\{a_1, \dots, a_n\}, \leq)$ is defined by the term $fgs(t)$ and also, in a different way, by the term $fgs(ext_{a_n}(ext_{a_{n-1}}(\dots(ext_{a_1}(\Omega))\dots)))$.

(3.14) *Definition: The algebra \mathbf{SBJT}*

We let F be the signature $\{\bullet, ext, \Omega\}$. We obtain an algebra \mathbf{SBJT} whose domain is the set of isomorphism classes of SBJ-trees. Concatenation is associative with neutral element Ω .

(3.15) *Definition : The value of a term.*

(a) In order to define the value of a term t in $T^\infty(F)$, we compare its positions as follows:

$u \approx v$ if and only if every position w such that $u <_t w \leq_t u \sqcup_t v$ or $v <_t w \leq_t u \sqcup_t v$ is an occurrence of \bullet .

This relation is an equivalence. We will also use the lexicographic order \leq_{lex} (positions are Dewey words). If w is an occurrence of a binary symbol, then $s_1(w)$ is its first (left) son and $s_2(w)$ its second (right) one.

(b) We define the *value* $val(t) := (N, \leq, \mathcal{U})$ of t in $T^\infty(F)$ as follows:

$N := Occ(t, ext)$, the set of occurrences in t of ext ,

$u \leq v : \iff u \leq_t w \leq_{lex} v$ for some $w \in N$ such that $w \approx v$,

\mathcal{U} is the set of equivalence classes of \approx .

Equivalently, we have :

$u \leq v : \iff u \leq_t v$ or $u \leq_t s_1(u \sqcup_t v)$, $v \leq_t s_2(u \sqcup_t v)$ and $v \approx u \sqcup_t v$
(the position $u \sqcup_t v$ is an occurrence of \bullet),

and so:

$u \perp v : \iff u \leq_t s_1(u \sqcup_t v)$, $v \leq_t s_2(u \sqcup_t v)$ and there is an occurrence of ext between v and $u \sqcup_t v$ or vice-versa by exchanging u and v .

(c) We now consider terms t written with the operations ext_a (such that a is the node created by applying this operation). For each a , the operation ext_a must have at most one occurrence in t . Assuming this condition satisfied, then $val(t) := (N, \leq, \mathcal{U})$ where :

N is the set of nodes a such that ext_a has an occurrence in t that we will denote by u_a ,

$a \approx b : \iff u_a \approx u_b$, with \approx as in (a),

$a \leq b : \iff u_a \leq u_b$, with \leq as in (b),

\mathcal{U} is the set of equivalence classes of \approx .

(3.15.1) *Claim:* The mapping val in (b) is a value mapping $T^\infty(F) \rightarrow \text{SBJT}$.

We say that t *denotes* an SBJ-tree J if J is isomorphic to $val(t)$, and, in this case, we also say that $fgs(t)$ denotes the BJ-tree $fgs(J)$.

(3.16) *Examples and remarks.*

(1) The term t that is the unique solution in $T^\infty(F)$ of the equation $t = t \bullet t$ denotes the empty SBJ-tree Ω .

(2) Let t_1 be the solution in $T^\infty(F)$ of the equation $t = ext(ext(\Omega)) \bullet t$ that we write by naming a, b, c, d, e, f, \dots the nodes created by the operations ext (i.e., the occurrences of ext):

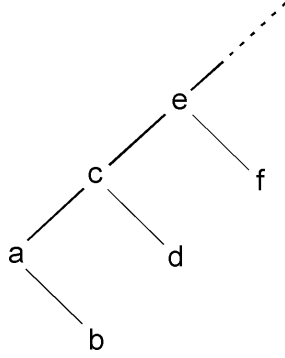


Figure 2: The SBJ-tree $val(t_1)$.

$$t_1 = ext_a(ext_b(\Omega)) \bullet (ext_c(ext_d(\Omega)) \bullet (ext_e(ext_f(\Omega)) \bullet \dots)).$$

Its value is shown in Figure 2. The bold edges link nodes in the axis. The nodes b and f are incomparable because the corresponding occurrences of ext , that are 111 and 2211, have least common ancestor ε and 221 is an occurrence of ext between 2211 and ε .

(3.17) *Definition* : The description scheme associated with a term.

(a) Let $t \in T^\infty(F)$ and $u \in Pos(t)$. We denote by $\text{Max}(t, ext, u)$ the set of maximal occurrences of ext in t that are below u or equal to it. Positions are denoted by Dewey words, hence, these sets are linearly ordered by \leq_{lex} . We denote by $W(t, u)$ the simple arrangement $(\text{Max}(t, ext, u), \leq_{lex})$.

Let $J = (N, \leq, \mathcal{U})$ be the value of t (cf. Definition (3.15)) and x be an occurrence of ext with son u . Then, by this definition, $(U^x, \leq) = (\text{Max}(t, ext, u), \leq_{lex})$.

For t_1 as in the previous example, $W(t_1, \varepsilon) = ace\dots$, $W(t_1, 1) = a$, $W(t_1, 11) = b$ and $W(t_1, 111) = \Omega$.

(b) We define $\Delta(t)$ as the SBJ-scheme $(\text{Occ}(t, ext), W(t, \varepsilon), (W(t, s(x)))_{x \in \text{Occ}(t, ext)})$ where $s(x)$ is the unique son of an occurrence x of ext .

(3.18) **Lemma** : If $t \in T^\infty(F)$, then $val(t)$ is described by $\Delta(t)$.

Proof: Let $val(t) = (N, \leq, \mathcal{U})$. The conditions of Definition (3.10.b) hold with the identity on $\text{Occ}(t, ext)$ as mapping r because $(U^x, \leq) = (\text{Max}(t, ext, s(x)), \leq_{lex})$ as observed in Definition (3.17.a). \square

(3.19) **Proposition** : Every SBJ-tree is the value of a term.

Proof : Let $J = (N, \leq, \mathcal{U})$ be an SBJ-tree. For each k , we let J_k be the SBJ-tree $(N_k, \leq, \mathcal{U}_k)$ where N_k is the set of nodes of depth at most k and \mathcal{U}_k is

the set lines $U \in \mathcal{U}$ of depth at most k . By induction on k , we define for each k a term t_k that defines J_k such that $t_k \leq t_{k'}$ if $k < k'$ and the least upper bound of the terms t_k is the desired term t whose value is J .

We define terms using the symbols ext_a where a names the node created by the corresponding occurrence of the extension operation.

If $k = 0$, then $J_0 = (A, \leq, \{A\})$. There exists a term $t \in T^\infty(\{\bullet\}, Ext_A)$ whose value is J_0 , where Ext_A is the set of terms $ext_a(\Omega)$ for $a \in A$ (we use Ext_A as a set of nullary symbols).

Let $k \geq 1$, where t_{k-1} defines J_{k-1} . Then J_k is obtained from J_{k-1} by adding below some nodes x at depth $k-1$ the line U^x (if $U^x = \emptyset$, there is nothing to add below x). Let $t_x \in T^\infty(\{\bullet\}, Ext_{U^x})$ whose value is (U^x, \leq) . We obtain t_k by replacing in t_{k-1} each subterm $ext_x(\Omega)$ by $ext_x(t_x)$.

It is clear that $t_{k-1} \leq t_k$ and that the least upperbound of the terms t_k defines J . \square

For the example of Figure 1 (Example (3.4)), if a term t_i denotes the SBJ-tree $(U_i, \leq, \{U_i\})$ for each i , then the term $t_0[t_1, t_2[t_3], t_4[t_5]]$ has value J , where the brackets $[\cdot]$ denote appropriate substitutions to occurrences of Ω . More precisely, we substitute t_5 to the occurrence of Ω that is the son of the occurrence of ext_{x_5} in t_4 , then we substitute the so obtained term $t_4[t_5]$ to the occurrence of Ω that is the son of the occurrence of ext_{x_4} in t_0 etc.

3.4 Regular binary join-trees

(3.20) *Definition : Regular BJ- and SBJ-trees.*

A BJ-tree (resp. an SBJ-tree) T is *regular* if it is denoted by $fgs(t)$ (resp. by t) where t is a regular term in $T^\infty(F)$.

(3.21) **Theorem:** The following properties of a BJ-tree J are equivalent:

- (1) J is regular,
- (2) J is described by a regular scheme,
- (3) J is MS-definable.

Proof : (1) \implies (2) Let $J = fgs(J')$ with J' denoted by a regular term t in $T^\infty(F)$. Let $h : Pos(t) \rightarrow Q$ and τ be as in the definition of a regular term in Section 2.2. Without loss of generality, we can assume that $h(Pos(t)) = Q$. If this is not the case, we replace Q by $Q' := Q - h(Pos(t))$ and τ by its restriction to Q' .

Claim: (1) For each $u \in Pos(t)$, the arrangement $\bar{h}(W(t, u)) = (\text{Max}(t, ext, u), \leq_{lex}, h)$ over Q is regular.

(2) If u' is another position in t and $h(u') = h(u)$, then $t/u' = t/u$ and furthermore² $\bar{h}(W(t, u')) \simeq \bar{h}(W(t, u))$.

Leaving its routine proof, we define $\Delta := (Q, w_{Ax}, (w_q)_{q \in Q})$ as follows:

- (i) $w_{Ax} := \bar{h}(W(t, \varepsilon))$,
- (ii) if $q \in Q$ then $w_q := \bar{h}(W(t, s(u)))$ where $s(u)$ is the unique son of an occurrence u of ext such that $h(u) = q$; if v is another occurrence of ext such that $h(v) = q$, then $h(s(v)) = h(s(u))$ and so by the claim, $\bar{h}(W(t, s(v))) \simeq \bar{h}(W(t, s(u)))$. Hence, w_q is well-defined up to isomorphism.

Informally, Δ is obtained from $\Delta(t)$ by replacing the labelling mapping Id of the arrangements $W(t, u)$ by h , so that these arrangements are turned into arrangements $\bar{h}(W(t, u))$ over Q . Clearly, Δ is a regular scheme. As mapping r showing that it describes J' (cf. Definition (3.10)), hence also J , we take the restriction of h to $\text{Occ}(t, ext)$ that is the set of nodes of $J' = \text{val}(t)$.

(2) \implies (3) is proved in Proposition (3.12).

(3) \implies (1) By Definition (3.2), the mapping α that transforms the relational structure $[t]$ for t in $T^\infty(F)$ into the BJ-tree $J = (N, \leq) = \text{fgs}(\text{val}(t))$ is an MS-transduction³ because an MS formula can identify the nodes of J among the positions of t and another one can define \leq .

Let $J = (N, \leq)$ be an MS-definable BJ-tree. It is, up to isomorphism, the unique model of an MS sentence β . It follows by a standard argument (recalled as Theorem (A.1) in the appendix) that the set of terms t in $T^\infty(F)$ such that $\alpha([t]) \models \beta$ is MS-definable and thus, contains a regular term (a result by Rabin, see [Tho90]). This term denotes J , hence J is regular. \square

(3.22) **Corollary :** The isomorphism problem for regular BJ-trees is decidable.

Proof : A regular BJ-tree can be given, either by a regular term, a regular scheme or an MS sentence. The proof of Theorem (3.21) is effective: algorithms can convert any of these specifications into another one. Hence, two regular BJ-trees can be given, one by an MS sentence β , the other by a regular term t . They are isomorphic if and only if $\alpha([t]) \models \beta$ (cf. the proof of (3) \implies (1) of Theorem (3.21)) if and only if $t \in L(\beta)$, which is decidable (see [Tho90]). \square

3.5 Logical and algebraic descriptions of join-trees

We now extend to join-trees the definitions and results of the previous sections. Structured join-trees are defined in Section 3.1 (Definition (3.3)). We extend to them the definitions and results of Sections 3.2-3.4.

²Unless $u = u'$, the sets $\text{Max}(t, ext, u)$ and $\text{Max}(t, ext, u')$ are not equal, so that the arrangements $\bar{h}(W(t, u))$ and $\bar{h}(W(t, u'))$ are isomorphic but not equal.

³MS logic, MS-transductions and related definitions and results are reviewed in the appendix.

The major novelty is that the argument of the extension operation *ext* will be an SJ-forest, equivalently a set of SJ-trees, instead of a single SBJ-tree. We will need an algebra with two sorts, the sort of SJ-trees and that of SJ-forests and monadic second-order formulas written with a finiteness predicate (cf. Section 2.3 and the appendix).

(3.23) *Definition : Description schemes for SJ-trees.*

(a) A *description scheme for an SJ-tree*, in short an *SJ-scheme*, is a 5-tuple $\Delta = (Q, D, w_{Ax}, (m_q)_{q \in Q}, (w_d)_{d \in D})$ such that Q, D are sets, $w_{Ax} \in \mathcal{A}(Q)$, $w_d \in \mathcal{A}(Q)$ for each $d \in D$ and $m_q = (M_q, lab_q)$ is a D -labelled set (cf. Section 2.3) for each $q \in Q$. Without loss of generality, we will assume that the domains V_{Ax} and V_d of the arrangements w_{Ax}, w_d and the sets M_q are pairwise disjoint, because these arrangements and labelled sets will be used up to isomorphism. Informally, M_q encodes the different lines U such that $\widehat{U} = x$ where x is labelled by q , and each of these lines is defined, up to isomorphism, by the arrangement w_d where d is its label in D , defined by lab_q .

We say that Δ is *regular* if $Q \cup D$ is finite and the arrangements w_{Ax} and w_d are regular. The finiteness of D implies that each D -labelled set m_q is regular.

(b) Let $J = (N, \leq, \mathcal{U})$ be an SJ-tree with axis A ; for each $x \in N$, we denote by \mathcal{U}^x the set of lines $U \in \mathcal{U}$ such that $\widehat{U} = x$. An SBJ-scheme Δ as in a) *describes* J if there exist mappings $r : N \rightarrow Q$ and $\tilde{r} : \mathcal{U} - \{A\} \rightarrow D$ such that:

- (b.1) the arrangement (A, \leq, r) over Q is isomorphic to w_{Ax} ,
- (b.2) for each $x \in N$, the D -labelled set $(\mathcal{U}^x, \tilde{r})$ is isomorphic to $m_{r(x)}$,
- (b.3) for each $U \in \mathcal{U} - \{A\}$, the arrangement (U, \leq, r) over Q is isomorphic to $w_{\tilde{r}(U)}$.

We will also say that Δ *describes* the join-tree $fgs(J) := (N, \leq)$.

(c) Each SBJ-tree $J = (N, \leq, \mathcal{U})$ has a *standard* description scheme $\Delta(J) := (N, \mathcal{U} - \{A\}, (A, \leq), ((\mathcal{U}^x, Id))_{x \in N}, ((U, \leq))_{U \in \mathcal{U} - \{A\}})$. The identity mappings $: N \rightarrow N$ and $: \mathcal{U} - \{A\} \rightarrow \mathcal{U} - \{A\}$ show that $\Delta(J)$ describes J .

(3.24) **Proposition:** Every SJ-scheme Δ describes a unique SJ-tree $J = Unf(\Delta)$ where unicity is up to isomorphism.

Proof: Let $\Delta = (Q, D, w_{Ax}, (m_q)_{q \in Q}, (w_d)_{d \in D})$ be an SJ-scheme, defined with arrangements $w_{Ax} = (V_{Ax}, \preceq, lab_{Ax})$ and $w_d = (V_d, \preceq, lab_d)$, and labelled sets $m_q = (M_q, lab_q)$ such that the sets V_{Ax} , V_d and M_q are pairwise disjoint and the same symbol \preceq denotes the orders of the arrangements w_{Ax} and w_d .

We construct $J = Unf(\Delta) = (N, \leq, \mathcal{U})$ as follows.

a) N is the set of finite nonempty sequences $(v_0, s_1, v_1, s_2, \dots, s_k, v_k)$ such that:

$v_0 \in V_{Ax}, v_i \in V_{d_i}$ and $s_i \in M_{q_{i-1}}$ for $1 \leq i \leq k$, where

$q_0 = \text{lab}_{Ax}(v_0)$, $d_1 = \text{lab}_{q_0}(s_1)$, $q_1 = \text{lab}_{d_1}(v_1)$, $d_2 = \text{lab}_{q_1}(s_2)$, ..., $q_i = \text{lab}_{d_i}(v_i)$, $d_{i+1} = \text{lab}_{q_i}(s_{i+1})$ for $1 \leq i \leq k-1$.

b) $(v_0, s_1, v_1, \dots, s_k, v_k) \leq (v'_0, s'_1, v'_1, \dots, s'_j, v'_j)$ if and only if :

$k \geq j$, $(v_0, s_1, v_1, \dots, s_j) = (v'_0, s'_1, v'_1, \dots, s'_j)$ and $v_j \preceq v'_j$ ($v_j, v'_j \in V_{d_j}$).

c) the axis A is the set of one-element sequences (v) for $v \in V_{Ax}$ and, for $x = (v_0, s_1, v_1, \dots, s_k, v_k)$, $U(x)$ is the set of sequences (hence of nodes) $(v_0, s_1, v_1, s_2, \dots, s_k, v)$ for $v \in V_{d_k}$, so that $\widehat{U(x)} = (v_0, s_1, v_1, \dots, s_{k-1}, v_{k-1})$.

Note that $(v_0, s_1, v_1, \dots, v_k) < (v_0, s_1, v_1, \dots, v_j)$ if $j < k$ and that $(v_0, s_1, v_1, \dots, s_k, v_k) \leq (v_0, s_1, v_1, \dots, s_k, v)$ if and only if $v_k \preceq v$.

In order to prove that Δ describes J , we define $r : N \rightarrow Q$ and $\tilde{r} : \mathcal{U} - \{A\} \rightarrow D$ as follows:

if $x \in A$, then $x = (v)$ for some $v \in V_{Ax}$ and $r(x) := \text{lab}_{Ax}(v)$;

if $x \in N$ has depth $k \geq 1$, then $x = (v_0, s_1, v_1, \dots, s_k, v_k)$ for some $v_0, s_1, \dots, s_k, v_k$ and $r(x) := \text{lab}_{d_k}(v_k)$;

if $U \in \mathcal{U} - \{A\}$, then $U = U(x)$ for some $x = (v_0, s_1, v_1, \dots, s_k, v_k)$, $k \geq 1$, and $\tilde{r}(U) := d_k$.

We check the three conditions of Definition (3.23.b). We have $(A, \leq, r) \simeq w_{Ax}$, hence (b.1) holds. For checking (b.2), we consider $x = (v_0, s_1, v_1, \dots, s_k, v_k) \in N$, $k \geq 1$. The sets U in \mathcal{U}^x are those of the form $\{(v_0, s_1, v_1, \dots, s_k, v_k, s, v) \mid v \in V_{d_{k+1}}\}$ for all $s \in M_{q_k}$ where $q_k = \text{lab}_{d_k}(v_k) = r(x)$, hence (b.2) holds. For checking (b.3), we let $U = U(x)$ for some $x = (v_0, s_1, v_1, \dots, s_k, v_k)$, $k \geq 1$; it is the set of sequences $(v_0, s_1, v_1, s_2, \dots, s_k, v)$ for $v \in V_{d_k}$ ordered by \preceq on the last components. Hence, $(U, \leq, \text{lab}_{d_k})$ is isomorphic to w_{d_k} , which proves the property since $\tilde{r}(U) := d_k$.

Unicity is proved as in Proposition (3.11). \square

The following proposition extends Proposition (3.12).

(3.25) **Proposition:** A join-tree is MS_{fin} -definable if it is described by a regular SJ-scheme.

Proof: Let (N, \leq) be a join-tree J (this property is first-order expressible). Assume that $J = \text{fgs}(J')$ where $J' = (N, \leq, \mathcal{U}) \simeq \text{Unf}(\Delta)$ for some regular SJ-scheme $\Delta = (Q, D, w_{Ax}, (m_q)_{q \in Q}, (w_d)_{d \in D})$ such that $Q = \{1, \dots, m\}$ and $D = \{1, \dots, p\}$. Let r, \tilde{r} be the corresponding mappings (cf. Definition 3.23.b). For

each $d \in D$, let ψ_d be an MS sentence that characterizes w_d up to isomorphism, by the main result of [Tho86]. Similarly, ψ_{Ax} characterizes w_{Ax} .

A D -labelled set m is described up to isomorphism by a p -tuple (m^1, \dots, m^p) where m^i is the number (possibly ω) of elements having label i .

By Proposition (3.7), there is a bipartition (N_0, N_1) of N that describes the structuring \mathcal{U} ; from this bipartition, we can MS-define the axis A , the lines forming \mathcal{U} and the node \widehat{U} for each $U \in \mathcal{U} - \{A\}$.

There is a partition (Y_1, \dots, Y_m) of N that describes r by $Y_i := r^{-1}(i)$. There is a partition (Z_1, \dots, Z_p) where Z_j is the union of the lines $U \in \mathcal{U} - \{A\}$ such that $\widehat{r}(U) = j$.

Consider a relational structure $(X, \leq, N_0, N_1, Y_1, \dots, Y_m, Z_1, \dots, Z_p)$. By MS formulas, one can express the following properties:

- (i) (X, \leq, N_0, N_1) is $S(J'')$ for some SJ-tree $J'' = (X, \leq, \mathcal{U}')$; its axis is denoted by A' ,
- (ii) (Y_1, \dots, Y_m) is a partition of X ; we let $r(x) := i$ if and only if $x \in Y_i$,
- (iii) (Z_1, \dots, Z_p) is a partition of X such that each Z_j is a union of sets $U \in \mathcal{U}' - \{A'\}$ such that $(U, \leq, r) \simeq w_j$,
- (iv) $(A', \leq, r) \simeq w_{Ax}$,
- (v) for each $i \in Q$ and $x \in Y_i$, the number of lines $U \in \mathcal{U}'^x$ that are contained in Z_j is m_i^j .

These formulas are constructed as follows: $\varphi(N_0, N_1)$ for (i) is from Proposition (3.7). The formula for (ii) is standard. All other formulas are intended to work when (i) and (ii) hold. For (iii), we use a suitable adaptation of ψ_i and the fact from Proposition (3.7) that, if (i) holds, we can MS-define from (N_0, N_1) the axis A' , the lines forming \mathcal{U}' and the node \widehat{U} for each $U \in \mathcal{U}'$. The mapping r is given by (Y_1, \dots, Y_m) . For (iv), we do as for (iii) with ψ_{Ax} .

For (v), we do as follows. We write an MS formula $\gamma(x, N_0, N_1, Z, W)$ expressing that W consists of one node of each set $U \in \mathcal{U}' - \{A'\}$ that is contained in Z and is such that $\widehat{U} = x$. For any x and Z , all sets W satisfying $\gamma(x, N_0, N_1, Z, W)$ have same cardinality. Then, Property (v) holds if and only if, for all $i = 1, \dots, m$, $x \in Y_i$ and $j = 1, \dots, p$, if $\gamma(x, N_0, N_1, Z_j, W)$ holds, then W has cardinality m_i^j . If some number m_i^j is ω , we need the finiteness predicate $Fin(W)$ to express this condition⁴.

Let $\beta(N_0, N_1, Y_1, \dots, Y_m, Z_1, \dots, Z_p)$ express conditions (ii)-(v) in (X, \leq) . If a join-tree (X, \leq) satisfies $\varphi(N_0, N_1) \wedge \beta(N_0, N_1, Y_1, \dots, Y_m, Z_1, \dots, Z_p)$, it has a structuring \mathcal{U}' described by N_0, N_1 : we let $J'' := (X, \leq, \mathcal{U}')$. The sets $Y_1, \dots, Y_m, Z_1, \dots, Z_p$ yield a scheme Δ that describes J'' (by Conditions (iii)-(v)), hence J'' is isomorphic to J' by the unicity property of Proposition (3.24), and so, we have $(X, \leq) \simeq fgs(J') = J$.

Hence, J is (up to isomorphism) the unique model of the MS_{fin} sentence :

$$\exists N_0, N_1 (\varphi(N_0, N_1) \wedge \exists Y_1, \dots, Y_m, Z_1, \dots, Z_p. \beta(N_0, N_1, Y_1, \dots, Y_m, Z_1, \dots, Z_p)).$$

⁴If the nodes of J have degree at most $a \in \mathbb{N}$, then $m_i^j \leq a$ for all i, j and the finiteness predicate is not needed, hence, J is MS-definable.

□

Theorem (3.30) will establish a converse.

(3.26) *Definition : Operations on SJ-trees and SJ-forests.*

We recall from Definition (3.1) that a join-forest is the union of disjoint join-trees. A structured join-forest (an SJ-forest, cf. Definition (3.4)) is the union of disjoint SJ-trees. It has no axis (each of its components has an axis, but we do not single out any of them). We will use objects of three types : join-trees, SJ-trees and SJ-forests, but a 2-sorted algebra will suffice (similarly as above in SBJT, we have not introduced a separate sort for BJ-trees). The two sorts are \mathbf{t} for SJ-trees and \mathbf{f} for SJ-forests.

Concatenation of SJ-trees along axes.

The concatenation $J \bullet J'$ disjoint SJ-trees J and J' is defined exactly as in Definition (3.13) for SBJ-trees.

The empty SJ-tree is denoted by the nullary symbol $\Omega_{\mathbf{t}}$.

Extension of an SJ-forest into an SJ-tree.

Let $J = (N, \leq, \mathcal{U})$ be an SJ-forest and $u \notin N$. Then $ext_u(J)$ is an SJ-tree defined as in Definition (3.13). When handling SJ-trees up to isomorphism, we will use the notation $ext(J)$ instead of $ext_u(J)$.

The empty SJ-forest is denoted by the nullary symbol $\Omega_{\mathbf{f}}$.

Making an SJ-tree into an SJ-forest.

This is done by the unary operation mkf that is actually the identity on the triples that define SJ-trees and SJ-forests.

The union of two disjoint SJ-forests is denoted by \uplus .

The types of these operations are thus:

$$\begin{aligned} \bullet &: \mathbf{t} \times \mathbf{t} \rightarrow \mathbf{t}, \\ \Omega_{\mathbf{t}} &: \mathbf{t}, \\ \uplus &: \mathbf{f} \times \mathbf{f} \rightarrow \mathbf{f}, \\ \Omega_{\mathbf{f}} &: \mathbf{f}, \\ ext &: \mathbf{f} \rightarrow \mathbf{t}, \\ mkf &: \mathbf{t} \rightarrow \mathbf{f}. \end{aligned}$$

In addition, we have, as in Definition (3.13):

Forgetting structuring: If J is an SJ-tree, $fgs(J)$ is the underlying join-tree.

(3.27) *Definition: The algebra \mathbb{SJT}*

We let F' be the 2-sorted signature $\{\bullet, \uplus, ext, mkf, \Omega_t, \Omega_f\}$ where the types of these six operations are as above. We obtain an F' -algebra \mathbb{SJT} whose domains are the sets of isomorphism classes of SJ-trees and of SJ-forests. Concatenation is associative with neutral element Ω_t and disjoint union is associative and commutative with neutral element Ω_f .

(3.28) *Definition : The value of a term.*

The definition is actually identical to that for SBJ-trees (Definition (3.15)). We recall it for the reader's convenience. The equivalence relation \approx is as in this definition. The *value* $val(t) = (N, \leq, \mathcal{U})$ of $t \in T^\infty(F')$ is defined as follows:

$$\begin{aligned} N &:= \text{Occ}(t, ext), \text{ the set of occurrences in } t \text{ of } ext, \\ u \leq v &: \iff u \leq_t w \leq_{lex} v \text{ for some } w \in N \text{ such that } w \approx v, \\ \mathcal{U} &\text{ is the set of equivalence classes of } \approx. \end{aligned}$$

If t has sort \mathbf{t} (resp. \mathbf{f}) then $val(t)$ is an SJ-tree (resp. an SJ-forest). It is clear that we have a value mapping : $T^\infty(F') \rightarrow \mathbb{SJT}$.

For terms written with the operations ext_a , then $val(t) := (N, \leq, \mathcal{U})$ where :

$$\begin{aligned} N &\text{ is the set of nodes } a \text{ such that } ext_a \text{ has an occurrence in } t, \text{ actually} \\ &\text{ a unique one, that we will denote by } u_a, \\ a \leq b &: \iff u_a \leq u_b, \\ a \approx b &: \iff u_a \approx u_b, \text{ and} \\ \mathcal{U} &\text{ is the set of equivalence classes of } \approx. \end{aligned}$$

(3.29) *Definition : Regular join-trees.*

A join-tree (resp. an SJ-tree) T is *regular* if it is denoted by $fgs(t)$ (resp. by t) where t is a regular term in $T^\infty(F')$ of sort \mathbf{t} .

(3.30) **Theorem:** The following properties of a join-tree J are equivalent:

- (1) J is regular,
- (2) J is described by a regular scheme,
- (3) J is MS_{fin} -definable.

Proof: (1) \implies (2). Similar to that of Theorem (3.21).

(2) \implies (3) By Proposition (3.25).

(3) \implies (1) As in the proof of Theorem (3.21), the mapping α that transforms the relational structure $[t]$ for t in $T^\infty(F')_{\mathbf{t}}$ into the join-tree $J = (N, \leq) = fgs(val(t))$ is an MS-transduction. Let $J = (N, \leq)$ be an MS_{fin} -definable join-tree. It is, up to isomorphism, the unique model of an MS_{fin} sentence β . The set L of terms t in $T^\infty(F')_{\mathbf{t}}$ such that $\alpha([t]) \models \beta$ is thus MS_{fin} -definable. However, since the relational structures $[t]$ have MS-definable linear orderings, L is also MS-definable, hence, it contains a regular term. This term denotes J , hence J is regular. \square

The same proof as for Corollary (3.22) yields:

(3.31) **Corollary :** The isomorphism problem for regular join-trees is decidable.

The rooted trees of unbounded degree, without order on the sets of sons of their nodes are the join-trees defined by the terms in $T^\infty(F' - \{\bullet\})_{\mathbf{t}}$. Theorem (3.30) and Corollary (3.31) hold for them.

4 Ordered join-trees

(4.1) *Definition : Ordered join-trees and join-hedges.*

Let $J = (N, \leq)$ be a join-forest. For each node x , $Dir(x)$ is the set of its directions (cf. Definition (3.2)). We denote by $x \perp y$ the fact that nodes x and y are incomparable with respect to \leq (so that $x < x \sqcup y$ and $y < x \sqcup y$ if $x \sqcup y$ is defined).

(a) We say that a join-tree $J = (N, \leq)$ is *ordered* (is an *OJ-tree*) if each set $Dir(x)$ is equipped with a linear order \sqsubseteq_x . (In this way, we generalize the notion of an ordered tree.) From these orders, we define a single linear order \sqsubseteq on N as follows:

$$x \sqsubseteq y \text{ if and only if } x \leq y \text{ or, } x \perp y \text{ and } \delta \sqsubseteq_{x \sqcup y} \delta' \text{ where } \delta, \delta' \in Dir(x \sqcup y), x \in \delta \text{ and } y \in \delta'.$$

(b) The linear order \sqsubseteq satisfies the following properties, for all x, y, x', y' :

- (i) $x \leq y$ implies $x \sqsubseteq y$,
- (ii) if $x \leq y$, $x' \leq y'$ and $y \perp y'$, then $x \sqsubseteq x'$ if and only if $y \sqsubseteq y'$.

(4.1.1) *Claim* : If $J = (N, \leq)$ is a join-tree and \sqsubseteq is a linear order on N satisfying conditions (i) and (ii), then J is ordered by the family of orders $(\sqsubseteq_x)_{x \in N}$ such that, for all δ, δ' in $Dir(x)$, we have $\delta \sqsubseteq_x \delta'$ if and only if $\delta = \delta'$ or $y \sqsubset y'$ for some $y \in \delta$ and $y' \in \delta'$ if and only if $\delta = \delta'$ or $y \sqsubset y'$ for all $y \in \delta$ and $y' \in \delta'$.

Proof sketch: Consider different directions $\delta, \delta' \in Dir(x)$ such that $y \sqsubset y'$ for some $y \in \delta$ and $y' \in \delta'$. We have also $y_1 \sqsubset y'_1$ for any $y_1 \in \delta$ and $y'_1 \in \delta'$ because $(y \sqcup y_1) < x, (y' \sqcup y'_1) < x$ and $(y \sqcup y_1) \perp (y' \sqcup y'_1)$, hence, Condition (ii) implies that $y \sqcup y_1 \sqsubset y' \sqcup y'_1$ and $y_1 \sqsubset y'_1$.

Hence, each relation \sqsubseteq_x is a linear order on $Dir(x)$. It is clear that \sqsubseteq is derived from the relations \sqsubseteq_x by (a). \square

It follows that an ordered join-tree can be equivalently defined as a triple (N, \leq, \sqsubseteq) such that (N, \leq) is a join-tree and \sqsubseteq is a linear order that satisfies Conditions (i) and (ii). These conditions are first-order expressible.

(c) We define a *join-hedge* as a triple $H = (N, \leq, \sqsubseteq)$ such that (N, \leq) is a join-forest and \sqsubseteq is a linear order that satisfies Conditions (i) and (ii). Let $J_s, s \in S$ be the join-trees composing H . Each of them is ordered by \sqsubseteq according to Claim (4.1.1), and the index set S is linearly ordered by \sqsubseteq_S such that $s \sqsubset_S s'$ if and only if $s \neq s'$ and $x \sqsubset y$ for all x in J_s and y in $J_{s'}$. Hence H is also a simple arrangement of pairwise disjoint join-trees.

(4.2) *Definition: Structured ordered join-trees and structured join-hedges.*

(a) Informally, a *structured ordered join-tree* (an *SOJ-tree*) is an OJ-tree (N, \leq, \sqsubseteq) equipped with a structuring \mathcal{U} (of (N, \leq)). For each node x , the set $Dir(x)$ of its directions consists of the following sets:

the sets $\downarrow(U)$ for each line $U \in \mathcal{U}^x$ (we recall that $\downarrow(U) := \{y \mid y \leq z \in U \text{ for some } z\}$),

the set $\downarrow(U_-(x))$ (cf. Section 2) if $U_-(x)$ is not empty; in this case we call it the *central direction* of x .

Hence, x has no central direction if it is the smallest element of $U(x)$. We get a linear order on \mathcal{U}^x that we also denote by \sqsubseteq_x . If $U, U' \in \mathcal{U}^x$, then $U \sqsubset_x U'$ if and only if $y \sqsubset y'$ for all $y \in U$ and $y' \in U'$.

We now define formally an SOJ-tree as a tuple $(N, \leq, \sqsubseteq, A, \mathcal{U}^-, \mathcal{U}^+)$ such that (N, \leq, \sqsubseteq) is an OJ-tree and $\mathcal{U} := \{A\} \cup \mathcal{U}^- \cup \mathcal{U}^+$ is a structuring of (N, \leq) such that, for each node x :

if $U \in \mathcal{U}^x \cap \mathcal{U}^-$ and $U' \in \mathcal{U}^x \cap \mathcal{U}^+$, then $U \sqsubset_x U'$ and furthermore,
if x has a central direction δ , then $U \sqsubset_x \delta \sqsubset_x U'$.

We define then $Dir^-(x)$ as the set of directions $\downarrow(U)$ for $U \in \mathcal{U}^x \cap \mathcal{U}^-$ and similarly $Dir^+(x)$ for $U \in \mathcal{U}^x \cap \mathcal{U}^+$. The reason for distinguishing \mathcal{U}^- and \mathcal{U}^+ will appear in the definition of the operations on SOJ-trees.

Let x be such that $[x, +\infty[\cap U \neq \emptyset$ for some U in \mathcal{U} . Then Condition 2) of Definition (3.3.a) shows that there is a node y_i in U . We say that x is *to the left* (resp. *to the right*) of U if $x \in \delta \in \text{Dir}^-(y_i)$ (resp. $x \in \delta \in \text{Dir}^+(y_i)$) for some δ .

(b) A *structured join-hedge* (an *SJ-hedge*) is a 4-tuple $J = (N, \leq, \sqsubseteq, \mathcal{U})$ such that (N, \leq, \sqsubseteq) is a join-hedge and \mathcal{U} is a structuring of the join-forest (N, \leq) . No line in \mathcal{U} is defined as an axis and we do not distinguish two sets \mathcal{U}^- and \mathcal{U}^+ .

As in Propositions (3.5) and (3.9), we have :

(4.3) **Proposition** : Every ordered join-tree and every join-hedge has a structuring.

Proof: Let (N, \leq, \sqsubseteq) be an OJ-tree and \mathcal{U} be any structuring of the join-tree (N, \leq) . Let A be its axis. We need only partition each set $\mathcal{U}^x - \{A\}$ into two sets $\mathcal{U}^x \cap \mathcal{U}^-$ and $\mathcal{U}^x \cap \mathcal{U}^+$.

If x has a central direction δ , we let $\mathcal{U}^x \cap \mathcal{U}^-$ consist of the lines U in \mathcal{U}^x such that $\downarrow(U) \sqsubseteq_x \delta$, and $\mathcal{U}^x \cap \mathcal{U}^+$ consist of those such that $\delta \sqsubseteq_x \downarrow(U)$. Otherwise, we partition \mathcal{U}^x in such a way that $\downarrow(U) \sqsubseteq_x \downarrow(U')$ for every $U \in \mathcal{U}^x \cap \mathcal{U}^-$ and $U' \in \mathcal{U}^x \cap \mathcal{U}^+$ (this partition is not unique).

For a join-hedge (N, \leq, \sqsubseteq) , we take a structuring \mathcal{U} of the join-forest (N, \leq) .

□

We now establish the MS-definability of these structurings. If $J = (N, \leq, \sqsubseteq, A, \mathcal{U}^-, \mathcal{U}^+)$ is an SOJ-tree, we define $S(J)$ as the structure $(N, \leq, \sqsubseteq, A, N_0^-, N_0^+, N_1^-, N_1^+)$ such that A is the axis, N_0^- (resp. N_0^+) is the union of the lines $U \in \mathcal{U}^-$ (resp. $U \in \mathcal{U}^+$) of even depth and N_1^- (resp. N_1^+) is the union of the lines $U \in \mathcal{U}^-$ (resp. $U \in \mathcal{U}^+$) of odd depth.

(4.4) **Proposition** : Let $J = (N, \leq, \sqsubseteq)$ be an OJ-tree.

1) There is an MS formula $\varphi(A, N_0^-, N_0^+, N_1^-, N_1^+)$ expressing that a structure $(N, \leq, \sqsubseteq, A, N_0^-, N_0^+, N_1^-, N_1^+)$ is $S(J)$ for some SOJ-tree $J = (N, \leq, \sqsubseteq, A, \mathcal{U}^-, \mathcal{U}^+)$.

2) There exists an MS formula $\theta^-(u, U, N_0^-, N_0^+, N_1^-, N_1^+)$ expressing in a structure $(N, \leq, \sqsubseteq, A, N_0^-, N_0^+, N_1^-, N_1^+) = S(N, \leq, \sqsubseteq, A, \mathcal{U}^-, \mathcal{U}^+)$ that $U \in \mathcal{U}^- \wedge u = \widehat{U}$; similarly, there exists an MS formula $\theta^+(u, U, N_0^-, N_0^+, N_1^-, N_1^+)$ expressing that $U \in \mathcal{U}^+ \wedge u = \widehat{U}$.

Proof: Easy modification of the proof of Proposition (3.7).

(4.5) *Definition* : *Description schemes for SOJ-trees.*

(a) A *description scheme for an SOJ-tree*, in short an *SOJ-scheme*, is a 6-tuple $\Delta = (Q, D, w_{Ax}, (w_q^-)_{q \in Q}, (w_q^+)_{q \in Q}, (w_d)_{d \in D})$ such that Q, D are sets,

$w_{Ax} \in \mathcal{A}(Q)$, $(w_d)_{d \in D}$ is a family of arrangements over Q and $(w_q^-)_{q \in Q}$ and $(w_q^+)_{q \in Q}$ are families of arrangements over D . Without loss of generality, we will assume that the domains of these arrangements are pairwise disjoint, and the same symbol \preceq denotes their orders. Informally, $(w_q^-)_{q \in Q}$ and $(w_q^+)_{q \in Q}$ encodes the sets of lines, ordered by \sqsubseteq_x of the two sets $Dir^-(x)$ and $Dir^+(x)$ where x is labelled by q .

We say that Δ is *regular* if $Q \cup D$ is finite and the arrangements w_{Ax} , w_d , w_q^- and w_q^+ are regular.

(b) Let $J = (N, \leq, \sqsubseteq, A, \mathcal{U}^-, \mathcal{U}^+)$ be an SOJ-tree. An SOJ-scheme Δ as in (a) *describes* J if there exist mappings $r : N \rightarrow Q$ and $\tilde{r} : \mathcal{U}^- \cup \mathcal{U}^+ \rightarrow D$ such that:

$$(b.1) \quad (A, \leq, r) \simeq w_{Ax},$$

$$(b.2) \quad \text{for each } x \in N, \text{ the arrangement } (\mathcal{U}^x \cap \mathcal{U}^-, \sqsubseteq_x, \tilde{r}) \text{ over } D \text{ is isomorphic to } w_{r(x)}^-,$$

$$(b.3) \quad \text{for each } x \in N, \text{ the arrangement } (\mathcal{U}^x \cap \mathcal{U}^+, \sqsubseteq_x, \tilde{r}) \text{ over } D \text{ is isomorphic to } w_{r(x)}^+,$$

$$(b.4) \quad \text{for each } U \in \mathcal{U}^- \cup \mathcal{U}^+, \text{ the arrangement } (U, \leq, r) \text{ over } Q \text{ is isomorphic to } w_{\tilde{r}(U)}.$$

We will also say that Δ *describes* the join-tree $fgs(J) := (N, \leq)$.

(4.6) **Proposition :** Every SOJ-scheme describes an SOJ-tree that is unique up to isomorphism.

Proof: Let $\Delta = (Q, D, w_{Ax}, (w_q^-)_{q \in Q}, (w_q^+)_{q \in Q}, (w_d)_{d \in D})$ be an SOJ-scheme, defined with arrangements $w_{Ax} = (V_{Ax}, \preceq, lab_{Ax})$, $w_d = (V_d, \preceq, lab_d)$, $w_q^- = (W_q^-, \preceq, lab_q^-)$ and $w_q^+ = (W_q^+, \preceq, lab_q^+)$ such that the sets V_{Ax}, V_d, W_q^- and W_q^+ are pairwise disjoint. Furthermore, we extend \prec by letting $s \prec s'$ for all $s \in W_q^-$, $s' \in W_q^+$ and $q \in Q$. We construct $J = Unf(\Delta) = (N, \leq, \sqsubseteq, A, \mathcal{U}^-, \mathcal{U}^+)$ as follows. Clauses a) to d) are essentially as in Proposition (3.24).

a) N is the set of finite nonempty sequences $(v_0, s_1, v_1, s_2, \dots, s_k, v_k)$ such that:

$$\begin{aligned} v_0 &\in V_{Ax}, v_i \in V_{d_i} \text{ and } s_i \in W_{q_{i-1}}^- \cup W_{q_{i-1}}^+ \text{ for } 1 \leq i \leq k, \text{ where} \\ q_0 &= lab_{Ax}(v_0), d_1 = lab_{q_0}(s_1), q_1 = lab_{d_1}(v_1), d_2 = lab_{q_1}(s_2), \dots, \\ q_i &= lab_{d_i}(v_i), d_{i+1} = lab_{q_i}(s_{i+1}) \text{ for } 1 \leq i \leq k-1. \end{aligned}$$

$$\begin{aligned} \text{b) } (v_0, s_1, v_1, \dots, s_k, v_k) &\leq (v'_0, s'_1, v'_1, \dots, s'_j, v'_j) \text{ if and only if :} \\ k &\geq j, (v_0, s_1, v_1, \dots, s_j) = (v'_0, s'_1, v'_1, \dots, s'_j) \text{ and } v_j \preceq v'_j \text{ } (v_j, v'_j \in V_{d_j}). \end{aligned}$$

c) The axis A is the set of one-element sequences (v) for $v \in V_{Ax}$.

d) If $x = (v_0, s_1, v_1, \dots, s_k, v_k)$, the line $U(x)$ is the set of sequences $(v_0, s_1, v_1, s_2, \dots, s_k, v)$ for $v \in V_{d_k}$; it belongs to \mathcal{U}^- if $s_k \in W_{q_{k-1}}^-$ and to \mathcal{U}^+ if $s_k \in W_{q_{k-1}}^+$; in both cases, $\widehat{U(x)} = (v_0, s_1, v_1, \dots, s_{k-1}, v_{k-1})$.

e) $x = (v_0, s_1, v_1, \dots, s_k, v_k) \sqsubseteq y = (v'_0, s'_1, v'_1, \dots, s'_j, v'_j)$ if and only if ,

either $x \leq y$ or, for some $\ell < \{j, k\}$, we have

e.1) $(v_0, s_1, v_1, \dots, v_\ell) = (v'_0, s'_1, v'_1, \dots, v'_\ell)$ and $s_{\ell+1} \prec s'_{\ell+1}$,

e.2) or $(v_0, s_1, v_1, \dots, s_\ell) = (v'_0, s'_1, v'_1, \dots, s'_\ell)$, $s_{\ell+1} \in W_{q_\ell}^-$ and $v'_\ell \prec v_\ell$,

e.3) or $(v_0, s_1, v_1, \dots, s_\ell) = (v'_0, s'_1, v'_1, \dots, s'_\ell)$, $s'_{\ell+1} \in W_{q_\ell}^+$ and $v_\ell \prec v'_\ell$.

In Case e.1), x and y are in different directions of $z := (v_0, s_1, v_1, \dots, v_\ell)$ that are not its central direction; in Case e.2), x is to the left of the central direction δ of z and $y \leq u$ where $u := (v_0, s_1, v_1, \dots, v'_\ell)$ is here below z on δ ; in Case e.3), y is to the right of the central direction δ' of u and $x \leq z$ where z is below u on δ' .

In order to prove that Δ describes J , we define $r : N \rightarrow Q$ and $\tilde{r} : \mathcal{U}^- \cup \mathcal{U}^+ \rightarrow D$ as follows:

if $x \in A$, then $x = (v)$ for some $v \in V_{Ax}$ and $r(x) := lab_{Ax}(v)$;

if $x \in N$ has depth $k \geq 1$, then $x = (v_0, s_1, v_1, \dots, s_k, v_k)$ for some $v_0, s_1, \dots, s_k, v_k$ and $r(x) := lab_{d_k}(v_k)$;

if $U \in \mathcal{U}^- \cup \mathcal{U}^+$, then $U = U(x)$ for some $x = (v_0, s_1, v_1, \dots, s_k, v_k)$, and $\tilde{r}(U) := d_k$.

In the last case, as $d_k = lab_{q_{k-1}}(s_k)$, it depends only on s_k and v_{k-1} (via q_{k-1}). It follows that $\tilde{r}(U)$ is the same if we consider U as $U(y)$ with $y = (v_0, s_1, v_1, \dots, s_k, v)$ hence, is well-defined.

We check the four conditions of Definition (4.5.b). We have $(A, \leq, r) \simeq w_{Ax}$, hence (b.1) holds. For (b.2) and (b.3), we consider $x = (v_0, s_1, v_1, \dots, s_k, v_k) \in N$. The sets U in \mathcal{U}^x are those of the form $\{(v_0, s_1, v_1, \dots, s_k, v_k, s, v) \mid v \in V_{d_{k+1}}\}$ for all $s \in W_{q_k}^- \cup W_{q_k}^+$ where $q_k = lab_{d_k}(v_k) = r(x)$, hence (b.2) and (b.3) hold.

For checking (b.4), we let $U = U(x)$ for some $x = (v_0, s_1, v_1, \dots, s_k, v_k)$, $k > 0$; then U is the set of sequences $(v_0, s_1, v_1, s_2, \dots, s_k, v)$ such that $v \in V_{d_k}$ ordered by \preceq on the last components. Hence, (U, \leq, lab_{d_k}) is isomorphic to w_{d_k} , which proves the property since $\tilde{r}(U) := d_k$.

Unicity is proved as in Proposition (3.11). \square

(4.7) **Proposition:** An SOJ-tree is MS-definable if it is described by a regular SOJ-scheme.

Proof: Similar to the proofs of Propositions (3.12) and (3.25). \square

Note that, we need not the finiteness predicate as in Proposition (3.25) because we deal with arrangements that are linearly ordered structures, and not with labelled sets.

Next we define an algebra SOJT with two sorts: \mathbf{t} for SOJ-trees and \mathbf{h} for SJ-hedges.

(4.8) *Definition : Operations on SOJ-trees and SJ-hedges.*

Concatenation of SOJ-trees along axes.

Let $J_1 = (N_1, \leq_1, \sqsubseteq_1, A_1, \mathcal{U}_1^-, \mathcal{U}_1^+)$ and $J_2 = (N_2, \leq_2, \sqsubseteq_2, A_2, \mathcal{U}_2^-, \mathcal{U}_2^+)$ be disjoint SOJ-trees. We define their concatenation as follows:

$$\begin{aligned} J_1 \bullet J_2 &:= (N_1 \cup N_2, \leq, \sqsubseteq, A_1 \cup A_2, \mathcal{U}_1^- \cup \mathcal{U}_2^-, \mathcal{U}_1^+ \cup \mathcal{U}_2^+) \text{ where :} \\ x \leq y &:\iff x \leq_1 y \vee x \leq_2 y \vee (x \in N_1 \wedge y \in A_2), \\ x \sqsubseteq y &:\iff x \leq y \vee x \sqsubseteq_1 y \vee x \sqsubseteq_2 y, \\ &\vee (x \perp y \wedge x \in N_1 \wedge y \in N_2 \wedge y \in U \in \mathcal{U}_2^+ \cap \mathcal{U}_2^{\perp y}) \\ &\vee (x \perp y \wedge x \in N_2 \wedge y \in N_1 \wedge x \in U \in \mathcal{U}_2^- \cap \mathcal{U}_2^{\perp y}), \text{ for some } U. \end{aligned}$$

The relations $x \perp y$ and $x \sqcup y$ are relative to \leq . It is clear that $J_1 \bullet J_2$ is an SOJ-tree.

The empty SOJ-tree is denoted by the nullary symbol $\Omega_{\mathbf{t}}$.

Extension of two SJ-hedges into a single SOJ-tree:

Let $H_1 = (N_1, \leq_1, \sqsubseteq_1, \mathcal{U}_1)$ and $H_2 = (N_2, \leq_2, \sqsubseteq_2, \mathcal{U}_2)$ be disjoint SJ-hedges and $u \notin N_1 \cup N_2$. Then:

$$\begin{aligned} ext_u(H_1, H_2) &:= (N_1 \cup N_2 \cup \{u\}, \leq, \sqsubseteq, \{u\}, \mathcal{U}_1, \mathcal{U}_2) \text{ where :} \\ x \leq y &:\iff x \leq_1 y \vee x \leq_2 y \vee y = u, \\ x \sqsubseteq y &:\iff x \leq y \vee x \sqsubseteq_1 y \vee x \sqsubseteq_2 y \vee (x \in N_1 \wedge y \in N_2). \end{aligned}$$

Clearly, $ext_u(J)$ is an SOJ-tree. When handling SOJ-trees and SJ-hedges up to isomorphism, we replace the notation $ext_u(H_1, H_2)$ by $ext(H_1, H_2)$.

The empty SJ-hedge is denoted by the nullary symbol $\Omega_{\mathbf{h}}$.

Making an SOJ-tree into an SJ-hedge.

This is done by the unary operation mkh such that, if $J = (N, \leq, \sqsubseteq, A, \mathcal{U}^-, \mathcal{U}^+)$ is an SOJ-tree, then

$$mkh(J) := (N, \leq, \sqsubseteq, \{A\} \cup \mathcal{U}^- \cup \mathcal{U}^+).$$

The concatenation of two disjoint SJ-hedges.

Let $H_1 = (N_1, \leq_1, \sqsubseteq_1, \mathcal{U}_1)$ and $H_2 = (N_2, \leq_2, \sqsubseteq_2, \mathcal{U}_2)$ be disjoint SJ-hedges. Their "horizontal" concatenation is:

$$H_1 \otimes H_2 := (N_1 \cup N_2, \leq_1 \cup \leq_2, \sqsubseteq, \mathcal{U}_1 \cup \mathcal{U}_2) \text{ where :}$$

$$x \sqsubseteq y :\iff x \sqsubseteq_1 y \vee x \sqsubseteq_2 y \vee (x \in N_1 \wedge y \in N_2).$$

We let F'' be the 2-sorted signature $\{\bullet, \otimes, ext, mkh, \Omega_t, \Omega_h\}$ whose operation types are :

$$\begin{aligned} \bullet &: t \times t \rightarrow t, \\ \Omega_t &: t, \\ \otimes &: h \times h \rightarrow h, \\ \Omega_h &: h, \\ ext &: h \times h \rightarrow t, \\ mkh &: t \rightarrow h. \end{aligned}$$

In addition, we have, as in Definitions (3.13) and (3.26) :

Forgetting the structuring:

If J is an SOJ-tree $(N, \leq, \sqsubseteq, A, \mathcal{U}^-, \mathcal{U}^+)$, then $fgs(J) := (N, \leq, \preceq)$ is the underlying OJ-tree.

(4.9) *Definition : The value of a term*

If u is an occurrence of a binary symbol in a term t , we denote by $s_1(u)$ its first son and by $s_2(u)$ the second one (cf. Definition (3.15)).

The value $val(t) = (N, \leq, \sqsubseteq, A, \mathcal{U}^-, \mathcal{U}^+)$ of a term $t \in T^\infty(F'')_t$ is defined in a similar way as for $t \in T^\infty(F')_t$ (Definitions (3.15) and (3.28)):

$$\begin{aligned} N &:= \text{Occ}(t, ext), \\ x \leq y &:\iff x \leq_t w \leq_{lex} y \text{ for some } w \in N \text{ such that } w \approx y, \\ A &:= \text{Max}(t, ext, \varepsilon), \end{aligned}$$

with \approx the equivalence relation on N defined as in Definition (3.15.a):

$$\begin{aligned} \mathcal{U}^- &\text{ is the set of equivalence classes of } \approx \text{ of nodes in } \text{Max}(t, ext, s_1(u)) \\ &\text{ for some occurrence } u \text{ of } ext, \\ \mathcal{U}^+ &\text{ is the set of equivalence classes of } \approx \text{ of nodes in } \text{Max}(t, ext, s_2(u)) \\ &\text{ for some occurrence } u \text{ of } ext. \end{aligned}$$

Hence, $U(x) \in \mathcal{U}^-$ if $x \leq_t s_1(\widehat{U(x)})$ and $U(x) \in \mathcal{U}^+$ if $x \leq_t s_2(\widehat{U(x)})$.

Next we define \sqsubseteq .

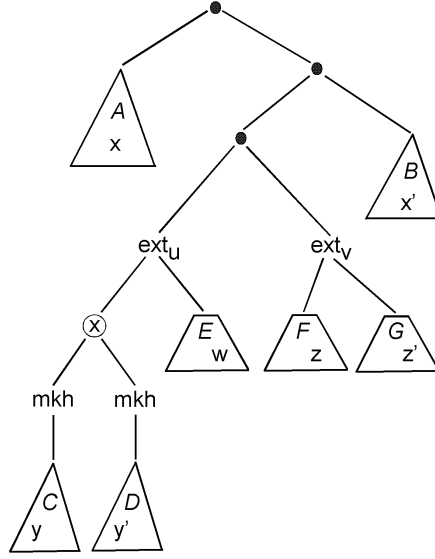


Figure 3: Term T of Example (4.10).

$x \sqsubseteq y :\iff x \leq y$ or $x \perp y$ (\perp is relative to \leq , not to \leq_t) and we have one of the following cases:

- (i) $x \sqcup_t y$ is an occurrence of \otimes or ext , $x \leq_t s_1(x \sqcup_t y)$ and $y \leq_t s_2(x \sqcup_t y)$,
- (ii) $x \sqcup_t y$ is an occurrence of \bullet , $x \leq_t s_1(x \sqcup_t y)$ and $y \leq_t s_2(z)$ where z is the unique maximal occurrence of ext such that $y <_t z \leq_t s_2(x \sqcup_t y)$,
- (iii) $x \sqcup_t y$ is an occurrence of \bullet , $y \leq_t s_1(x \sqcup_t y)$ and $x \leq_t s_1(z)$ where z is the unique maximal occurrence of ext such that $x <_t z \leq_t s_2(x \sqcup_t y)$.

If $t \in T^\infty(F'')_{\mathbf{h}}$ its value $val(t)$ is $(N, \leq, \sqsubseteq, \mathcal{U})$ with (N, \leq, \sqsubseteq) defined as above and \mathcal{U} as in Definition (3.28). Hence if $(N_s, \leq_s, \sqsubseteq_s, A_s, \mathcal{U}_s^-, \mathcal{U}_s^+)$ are the SOJ-trees composing $val(t)$, then, \mathcal{U} consists of the axes A_s and the lines in the sets $\mathcal{U}_s^-, \mathcal{U}_s^+$.

- (4.9.1) *Claim* : (1) The mapping val is a value mapping $T^\infty(F'') \rightarrow \text{SOJT}$.
(2) The transformation α of $[t]$ into (N, \leq, \sqsubseteq) is an MS-transduction.

Proof : (1) is clear from the definitions.

(2) holds because the conditions of Definition (4.9) are expressible in $[t]$ by MS formulas. \square

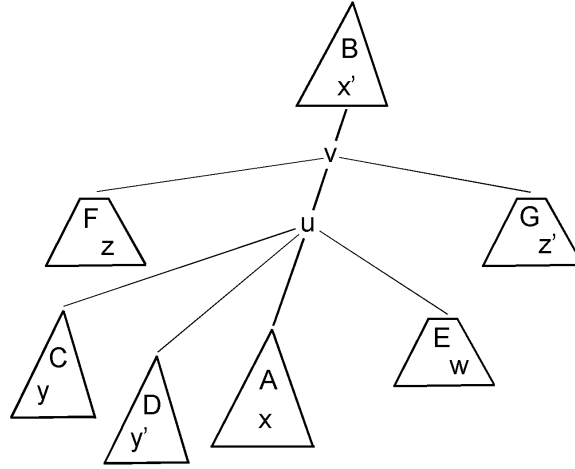


Figure 4: The OJ-tree $val(T)$ of Example (4.10).

(4.10) *Example:* We now illustrate this definition. Figure 3 shows a term T where A, B, C and D are subterms of type \mathbf{t} and E, F and G are subterms of type \mathbf{h} . They contain occurrences of ext that define nodes x, x', y, y', w, z and z' of $val(T)$.

The OJ-tree $val(T)$ is shown on Figure 4, where we designate by A, B, \dots, G the trees and hedges defined by the terms A, B, \dots, G .

We have the following comparisons for $<$:

- $\{z, z', u\} < v$, because $\{z, z'\} <_T v$, $u <_{lex} v$ and $u \approx v$,
- $\{y, y', w\} < u$, because $\{y, y', w\} <_T u$,
- $x \leq \{u, v\}$ because $x \leq_T a <_{lex} \{u, v\}$ and $a \approx u \approx v$ where a is the root position of A ,
- $v < x'$ if and only if x' is on the axis of B , because in this case, $v \approx x'$ and otherwise v and x' are incomparable with respect to \leq ;
- in all cases we have $v <_{lex} x'$.

For \sqsubset we have:

- $z \sqsubset y \sqsubset y' \sqsubset x \sqsubset w \sqsubset u \sqsubset z' \sqsubset v$
- and $x' \sqsubset z$ if x' is to the left of the axis of B ; otherwise $v \sqsubset x'$.

All inequalities for $<$ yield the corresponding inequalities for \sqsubset . We now compare z, y, y', x, w, z' that pairwise incomparable for $<$.

By Case (i) of Definition (4.9), we get $\{y, y'\} \sqsubset w, y \sqsubset y'$ and $z \sqsubset z'$.

By Case (ii), we get $x \sqsubset w, \{x, w\} \prec z'$ and $\{y, y'\} \prec w$.

By Case(iii) we get $\{z, y, y'\} \prec x$ and $z \prec \{y, y'\}$.

Finally, if x' is to the left of the axis of B , then Case (iii) gives $x' \sqsubset z$, and if it is to the right, then Case (ii) gives $z \sqsubset x'$. \square

(4.11) **Theorem:** The following properties of an OJ-tree J are equivalent:
(1) J is regular,
(2) J is described by a regular SOJ-scheme,
(3) J is MS-definable.

Proof: (1) \implies (2): Similar to that of Theorem (3.21).

(2) \implies (3): By Proposition (4.7).

(3) \implies (1) As observed in Claim (4.9.2), the mapping α that transforms the relational structure $[t]$ for t in $T^\infty(F'')_t$ into the OJ-tree $(N, \leq, \preceq) = fgs(val(t))$ is an MS-transduction. Let $J = (N, \leq, \preceq)$ be an MS-definable OJ-tree. It is, up to isomorphism, the unique model of an MS-sentence β . The set of terms t in $T^\infty(F'')_t$ such that $\alpha([t]) \models \beta$ is thus MS-definable, hence, it contains a regular term. This term denotes J , hence J is regular. \square

As in Corollaries (3.22) and (3.31) we deduce that the isomorphism problem for regular OJ-trees is decidable.

5 Quasi-trees

Quasi-trees can be viewed intuitively as "undirected join-trees". As in [Cou14], we define them in terms of a ternary betweenness relation.

(5.1) *Definition : Betweenness.*

(a) Let $L = (X, \leq)$ be a linear order. Its *betweenness relation* is the ternary relation on X such that $B_L(x, y, z)$ holds if and only if $x < y < z$ or $z < y < x$.

(b) If T is a tree, its *betweenness relation* is the ternary relation on N_T , such that $B_T(x, y, z)$ holds if and only if x, y, z are pairwise distinct and y is on the unique path between x and z . If R is a rooted tree and $T = Und(R)$ is the tree obtained from T by forgetting its root and edge directions, then :

$$B_T(x, y, z) \iff x, y, z \text{ are pairwise distinct and } x <_R y \leq_R x \sqcup_R z \\ \text{or } z <_R y \leq_R x \sqcup_R z.$$

(5.2) **Proposition** [Cou14] : (a) The betweenness relation B of a linear order (X, \leq) satisfies the following properties for all $x, y, z, u \in X$.

$$A1 : B(x, y, z) \Rightarrow x \neq y \neq z \neq x.$$

$$A2 : B(x, y, z) \Rightarrow B(z, y, x).$$

$$A3 : B(x, y, z) \Rightarrow \neg B(x, z, y).$$

$$\begin{aligned}
A4 : & B(x, y, z) \wedge B(y, z, u) \Rightarrow B(x, y, u) \wedge B(x, z, u). \\
A5 : & B(x, y, z) \wedge B(x, u, y) \Rightarrow B(x, u, z) \wedge B(u, y, z). \\
A6 : & B(x, y, z) \wedge B(x, u, z) \Rightarrow \\
& y = u \vee [B(x, u, y) \wedge B(u, y, z)] \vee [B(x, y, u) \wedge B(y, u, z)]. \\
A7' : & x \neq y \neq z \neq x \Rightarrow B(x, y, z) \vee B(x, z, y) \vee B(y, x, z).
\end{aligned}$$

(b) The betweenness relation B of a tree T satisfies the properties A1-A6 for all x, y, z, u in N_T together with the following weakening of A7':

$$\begin{aligned}
A7 : & x \neq y \neq z \neq x \Rightarrow B(x, y, z) \vee B(x, z, y) \vee B(y, x, z) \vee \\
& \exists w. (B(x, w, y) \wedge B(y, w, z) \wedge B(x, w, z)).
\end{aligned}$$

(5.3) **Proposition** : Let B be a ternary relation on a set X that satisfies properties A1-A7' for all x, y, z, u in X . Let a and b be distinct elements of X . There is a unique linear order $L = (X, \leq)$ such that $a < b$ and $B_L = B$. It is first-order definable in the logical structure (X, B, a, b) .

Proof: Let X, B be as in the statement. Let Y consist of n elements of X , $n > 2$. By Lemma 13 of [Cou14], it can be enumerated as x_1, \dots, x_n in such a way that $B(x_i, x_j, x_k)$ holds for all $1 \leq i < j < k \leq n$. The only other possible enumeration satisfying this property is x_n, \dots, x_1 . Hence, the only two linear orders on Y having $B \upharpoonright Y$ as betweenness relation are $x_1 < \dots < x_n$ and its reversal. It follows that if a and b are distinct elements of X (that may be infinite), there is a unique linear order $L = (X, \leq)$ such that $a < b$ and $B_L = B$. We denote it by $\leq_{a,b}$.

We now define $\leq_{a,b}$ by a first-order formula.

Case 1 : There are no u, v such that $B(u, b, v)$ holds. Then, we have

$$x \leq_{a,b} y : \Longleftrightarrow x = y \vee y = b \vee B(x, y, b).$$

Otherwise and similarly, if there are no u, v such that $B(u, a, v)$ holds, we have

$$x \leq_{a,b} y : \Longleftrightarrow x = y \vee x = a \vee B(a, x, y).$$

Case 2 : Otherwise, we define the following binary relation :

$$\begin{aligned}
Z(x, y) : & \Longleftrightarrow x \neq y \wedge \\
& [(B(x, a, b) \wedge \neg B(y, x, a)) \vee (x = a \wedge \neg B(y, a, b))] \vee \\
& (B(a, x, b) \wedge \neg B(y, x, b)) \vee (x = b \wedge B(a, b, y)) \vee \\
& (B(a, b, x) \wedge B(b, x, y)).
\end{aligned}$$

It is easy to see that $x <_{a,b} y$ implies $Z(x, y)$. (That $Z(a, b)$ holds follows from the clause $x = a \wedge \neg B(y, a, b)$ with $y = b$). For the converse, assume that $Z(x, y)$ holds and $x <_{a,b} y$ does not. Then, we have $y <_{a,b} x$. By looking at the different positions of x, y, a and b , we get a contradiction. Hence $x \leq_{a,b} y$ if and only if $x = y \vee Z(x, y)$, which is expressed by a first-order formula $\xi(a, b, x, y)$. \square

(5.4) *Definition* [Cou14] : *Quasi-trees*.

(a) A *quasi-tree* is a structure $S = (N, B)$ such that B is a ternary relation on N that satisfies conditions A1-A7. In a quasi-tree, the four cases of the conclusion of A7 are exclusive and in the fourth case, there is a unique w satisfying $B(x, w, y) \wedge B(y, w, z) \wedge B(x, w, z)$ (this is proved in Definition 11 of [Cou14]), that is denoted by $M_S(x, y, z)$.

A *leaf* of S is a node z such that $B(x, z, y)$ holds for no x, y . A *line* of S is set of nodes L such that $z \in L$ whenever $x, y \in L$ and $B(x, z, y)$ holds; an *end* of L is an element y such that there are no x, z in L such that $B(x, y, z)$ holds. We say that S is *discrete* if each set $]x, y[_B := \{z \mid B(x, z, y)\}$ is finite.

We say that a quasi-tree $S = (N, B)$ is a *subquasitree* of a quasi-tree $S' = (N', B')$, which we denote by $S \subseteq S'$, if $N \subseteq N'$ and $B = B' \upharpoonright N$. This condition implies that $M_S = M_{S'} \upharpoonright N$.

(b) From a join-tree $J = (N, \leq)$, we define a ternary relation B_J on N by:

$$B_J(x, y, z) : \Longleftrightarrow x \neq y \neq z \neq x \text{ and } (x < y \leq x \sqcup z) \vee (z < y \leq x \sqcup z).$$

(5.5) **Proposition** : (1) The structure $qt(J) := (N, B_J)$ associated with a join-tree $J = (N, \leq)$ is a quasi-tree. Every line of J is a line of $qt(J)$. If J is a subjoin-tree of J' , then $qt(J)$ is a subquasitree of $qt(J')$.

(2) Every quasi-tree S is $qt(J)$ for some join-tree J .

(3) A quasi-tree is discrete if and only if it is $qt(J)$ for some tree J .

(4) Let $S = (N, B)$ be a quasi-tree, L a line of S and a, b distinct elements of L . Assume furthermore that, either b is the largest element of L with respect to its linear order⁵ $\leq_{a,b}$ or L has no largest element and there is no w in $N - L$ such that $B(u, v, w)$ holds for all $u, v \in L$ such that $u \leq_{a,b} v$. Then there is a unique join-tree $J = (N, \leq)$ such that $S = qt(J)$, $a < b$ and L is an upwards closed line of J . Its order \leq is first-order definable in (N, B, L, a, b) .

Proof: (1) Let $J = (N, \leq)$ be a join-tree. If it is finite, then it is (N_T, \leq_T) for a finite tree T , and thus $qt(J)$ is a quasi-tree by Proposition (5.2.b).

Otherwise consider distinct elements x, y, z, u of N . By Corollary 14 of [Cou14], there is a set $N' \subseteq N$ of cardinality at most 7 that contains x, y, z, u and is closed under \sqcup . Then $J' = (N', \leq \upharpoonright N')$ is a finite join-tree, $J' \subseteq J$ and $qt(J') = (N', B_J \upharpoonright N')$ is a quasi-tree by the initial observation, so that A1-A7

⁵Proposition (5.3) is applicable to $(L, B \upharpoonright L)$ that satisfies Conditions A1-A7'.

hold for B_J and the elements x, y, z, u . As x, y, z, u are arbitrary, A1-A7 hold for B_J and all $x, y, z, u \in N$. Hence, (N, B_J) is a quasi-tree.

That every line of J is a line of $qt(J)$ follows from the definitions. (The converse does not hold. However, see Assertion (4).) The assertion about subjoin-trees is also easy to prove.

(2) Let $S = (N, B)$ be a quasi-tree and b be any element of N . We define (cf. Proposition (5.3)) :

$$x \leq_b y : \Longleftrightarrow x = y \vee y = b \vee B(x, y, b).$$

Then (N, \leq_b) is a join-tree J with root b and $S = qt(J)$ by Lemma 13 of [Cou14].

(3) is proved in Proposition 16 of [Cou14].

(4) Let us first motivate the technical condition on L . Let $J = (N, \leq)$ be a join-tree and $S = qt(J)$. If L is an upwards closed line in J , then there are two cases: either L has a maximal element that is the root of J or not. In the latter case, for every $x \in N$, there is $w \in L$ such that $x < w$: to prove this, take w such that $x \sqcup u < w$ where u is any element of L ; in terms of the ternary relation B_J of S , we observe that, if $x \in N - L$, we cannot have $B_J(u, v, x)$ for all $u, v \in L$ such that $u < v$, because, otherwise, letting $w, w' \in L$ with $x < w < w'$, we would have $B_J(w, w', x)$ and $B_J(x, w, w')$ contradicting $A2 \wedge A3$. The technical conditions on a line L of S in the statement are thus necessary for making it into an upwards closed line of a joint-tree J as desired. We now start the proof.

Let $S = (N, B)$ be a quasi-tree with line L and distinct elements a, b of L . Let $\leq_{a,b}$ be the linear order on L from Proposition (5.3).

Case 1 : b is the largest element of $(L, \leq_{a,b})$. Then (N, \leq_b) (cf. (2) for \leq_b) is a quasi-tree satisfying the required conditions. If $J = (N, \leq)$ is another one, then $[x, +\infty[_J = [x, b]_J$ for all x and thus, $x \leq y$ if and only if $x = y \vee y = b \vee B(x, y, b)$. Hence, $\leq = \leq_b$.

Case 2 : L has no largest element and there is no w in $N - L$ such that $B(u, v, w)$ holds for all $u, v \in L$ such that $u <_{a,b} v$. It follows that for every $u \in L$, there is $v \in L$ such that $u <_{a,b} v$.

We now define, for $x, y \in N$:

$$R(x, y) : \Longleftrightarrow x \neq y \wedge \exists u, v \in L [u <_{a,b} v \wedge B(x, y, u) \wedge B(y, u, v)],$$

and we will prove that this relation is the desired strict order on N .

For all $u, v \in L$ such that $u <_{a,b} v$, we define $N_{u,v} := \{x \in N \mid x = u \vee B(x, u, v)\}$. Then $S_{u,v} := (N_{u,v}, B \upharpoonright N_{u,v})$ is a subquasi-tree of S and u is the largest element of $L \cap N_{u,v}$ with respect to $\leq_{a,b}$. We define $\leq_{u,v}$ on $N_{u,v}$ by:

$$x \leq_{u,v} y : \Longleftrightarrow x = y \vee y = u \vee B(x, y, u).$$

We obtain by Case 1 a join-tree $J_{u,v} := (N_{u,v}, \leq_{u,v})$ with root u such that $S_{u,v} = qt(J_{u,v})$. It is clear that, if $v <_{a,b} u' <_{a,b} v'$, then $N_{u,v} \subseteq N_{u',v'}$ and $\leq_{u,v}$ and $\leq_{u',v'}$ coincide on $N_{u,v}$; the associated join operations coincide also on $N_{u,v}$. Hence $J_{u,v}$ is a sub-join-tree of $J_{u',v'}$.

By the assumption on L , every $x \in N$ belongs to some set $N_{u,v}$: assume the opposite, then $x \notin L$; assume now that $x \notin N_{u,v}$, hence by A7, we have $B(x, u, v)$ or $B(u, x, v)$ or $B(u, v, x)$ or, for some w , $B(u, w, v) \wedge B(x, w, u) \wedge B(x, w, v)$.

The first case implies $x \in N_{u,v}$, the second one $x \in L$, and the last one $w \in L$, $u <_{a,b} w <_{a,b} v$ and $x \in N_{w,v}$. All these cases contradict our assumptions on x . It follows that $B(u, v, x)$ holds for all u, v such that $u <_{a,b} v$. But this contradicts the assumption on L .

Hence, N is the union of the sets $N_{u,v}$ and S is the union of the quasi-trees $S_{u,v}$. We define $J = (N, \leq)$ as the union of the join-trees $J_{u,v}$. Its order is defined by $x = y \vee R(x, y)$. It is a join-tree and $qt(J) = S$ because $qt(J_{u,v}) = S_{u,v}$ for all u, v . It is the unique one satisfying the desired conditions because these conditions determine its restrictions to the sets $N_{u,v}$ in a unique way.

The property $x = y \vee R(x, y)$ is defined by a first-order formula $\theta(L, a, b, x, y)$.

□

(5.6) **Proposition:** A quasi-tree is MS_{fin} -definable if it is described by a regular SJ-scheme.

Proof: Here is the idea. From a given quasi-tree S , we specify by MS-formulas all SJ-trees J such that $S = qt(fgs(J))$ and we select one, if there exists any, that is described by the considered regular SJ-scheme. The construction of the necessary formulas will use Proposition (5.5) and the proof of Proposition (3.25).

Let $S = (N, B)$ be a quasi-tree (this property is first-order expressible). Assume that $S = qt(fgs(J))$ where $J = (N, \leq, \mathcal{U}) \simeq Unf(\Delta)$ for some regular SJ-scheme $\Delta = (Q, D, w_{Ax}, (m_q)_{q \in Q}, (w_d)_{d \in D})$ such that $Q = \{1, \dots, m\}$ and $D = \{1, \dots, p\}$ as in the proof of Proposition (3.25). Let r, \tilde{r} be the corresponding mappings. For each $d \in D$, let ψ_d be an MS sentence that characterizes w_d up to isomorphism, and similarly ψ_{Ax} for w_{Ax} .

The axis A of J is upwards closed. If it has a maximal element b (w.r.t. \leq), then $\leq = \leq_b$ by Proposition (5.5.2). Otherwise, $\leq = \leq_{a,b}$ where a, b are any two elements of A such that $a < b$. In this case, there is no w in $N - A$ such that $B_J(u, v, w)$ holds for all $u, v \in A$ such that $u \leq_{a,b} v$. In both cases, by Proposition (5.5) the partial order \leq on N is defined by a first-order formula $\nu(A, a, b, x, y)$ written in such a way that b is the maximal element of A if $a = b$. We will denote \leq by $\leq_{A,a,b}$ to stress that it is defined from A, a, b .

By Proposition (3.7), there is a bipartition (N_0, N_1) of N that describes the structuring \mathcal{U} , and such that $A \subseteq N_0$. From this bipartition, we can MS-define the lines forming \mathcal{U} and the node \hat{U} for each $U \in \mathcal{U} - \{A\}$.

There is a partition (Y_1, \dots, Y_m) of N that describes r by $Y_i := r^{-1}(i)$. There is a partition (Z_1, \dots, Z_p) of $N - A$ such that Z_j is the union of the lines $U \in \mathcal{U} - \{A\}$ such that $\tilde{r}(U) = j$.

Consider relational structures $R = (X, B, A, N_0, N_1, Y_1, \dots, Y_m, Z_1, \dots, Z_p, a, b)$ with domain X such that B is a ternary relation, the other relations are unary (are subsets of X) and $a, b \in X$. By MS formulas, one can express in any such R the following properties:

- (i) (X, B) is a quasi-tree S' , A is a nonempty line, $a, b \in A$ and $a = b$ if and only if a is an end of A ,
- (ii) if \leq is the partial order on N defined by $\nu(A, a, b, x, y)$, then (X, \leq, N_0, N_1) is $S(J')$ for some SJ-tree $J' = (X, \leq, \mathcal{U}')$ with axis A .
- (iii) (Y_1, \dots, Y_m) is a partition of X ; we let $r(x) := i$ if and only if $x \in Y_i$.
- (iv) (Z_1, \dots, Z_p) is a partition of $X - A$ such that each Z_j is a union of sets $U \in \mathcal{U}' - \{A\}$ such that $(U, \leq, r) \simeq w_j$.
- (v) $(A, \leq, r) \simeq w_{Ax}$,
- (vi) for each $i \in Q$ and $x \in Y_i$, the number of lines $U \in \mathcal{U}'^x$ that are contained in Z_j is m_i^j . (A D -labelled set m is described by a p -tuple (m^1, \dots, m^p) where m^j is the number of elements having label j .)

These formulas are constructed as in the proof of Proposition (3.25). The main difference is that \leq is not in the given structure S but is defined in it by $\nu(A, a, b, x, y)$; this formula uses auxiliary arguments A, a, b . As in Proposition (3.25), we need the finiteness predicate if some number m_i^j is ω .

Let $\varphi'(A, N_0, N_1, a, b)$ express Conditions (i) and (ii) in (X, B) . Let $\beta(A, N_0, N_1, Y_1, \dots, Y_m, Z_1, \dots, Z_p, a, b)$ express conditions (iii)-(vi). If a quasi-tree (X, B) satisfies :

$$\varphi'(A, N_0, N_1, a, b) \wedge \beta(A, N_0, N_1, Y_1, \dots, Y_m, Z_1, \dots, Z_p, a, b),$$

the corresponding join-tree $(X, \leq_{A,a,b})$ has a structuring \mathcal{U}' with axis A described by N_0, N_1 . We let $J' := (X, \leq, \mathcal{U}')$. The sets $Y_1, \dots, Y_m, Z_1, \dots, Z_p$ yield a scheme Δ that describes J' (by Conditions (iii)-(vi)), hence J' is isomorphic to J by the unicity property of Proposition (3.24), and so, we have $(X, \leq) = fgs(J') \simeq fgs(J)$ and $qt(fgs(J')) \simeq qt(fgs(J)) = S$.

Hence, S is (up to isomorphism) the unique model (X, B) of the MS sentence :

$$\begin{aligned} &\exists A, N_0, N_1, a, b. [\varphi'(A, N_0, N_1, a, b) \wedge \\ &\quad \exists Y_1, \dots, Y_m, Z_1, \dots, Z_p. \beta(A, N_0, N_1, Y_1, \dots, Y_m, Z_1, \dots, Z_p, a, b)]. \end{aligned}$$

□

The next theorem establishes a converse. As algebra for quasi-trees, we take the algebra \mathbb{SJT} of join-trees together with the (external) *forgetting operation* qt (similar to fgs). We say that a quasi-tree S is *described* by an SJ-scheme if this scheme describes a join-tree J such that $qt(J) = S$. It is *regular* if it is $qt(J)$ for some regular join-tree J .

(5.7) **Theorem:** The following properties of a quasi-tree S are equivalent:

- (1) S is regular,
- (2) S is described by a regular SJ-scheme,
- (3) S is MS_{fin} -definable.

The isomorphism of regular quasi-trees is decidable.

Proof: (1) \implies (2): Similar to that of Theorem (3.21).

(2) \implies (3): By Proposition (5.6).

(3) \implies (1): The mapping α that transforms the relational structure $[t]$ for t in $T^\infty(F')_{\mathbf{t}}$ into the quasi-tree $S = qt(fgs(val(t)))$ is an MS-transduction by Claim (4.9.2). The proof continues as in Theorem (3.21).

The decidability of the isomorphism problem is as in Corollary (3.22). \square

We make these results more precise for subcubic quasi-trees, that are useful for defining the rank-width of countable graphs, see the appendix.

(5.8) *Definition : Directions* ([Cou14])

Let $S = (N, B)$ be a quasi-tree and x a node of S .

(a) We say that $y, z \in N - \{x\}$ are the *same direction relative to x* (or of x) if, either $y = z$ or $B(y, z, x)$ or $B(z, y, x)$ or $B(y, u, x) \wedge B(z, u, x)$ for some node u . Equivalently, $y \sqcup_x z <_x x$ ($<_x$ is as in Proposition (5.5)). Hence, if $B(y, x, z)$ holds, y and z are in different directions relative to x . This relation is an equivalence, denoted by $y \sim_x z$, and its classes are the *directions of x* .

(b) The *degree of x* is the number of classes of \sim_x . A node has degree 1 if and only if it is a leaf. We say that S is *subcubic* if its nodes have degree at most 3. If $S = Q(T)$ for a tree T , then a direction of x is associated with each neighbour y of x and is the set of nodes of the connected component of $T - \{x\}$ that contains y .

(c) If $S = qt(J)$ for a join tree $J = (N, \leq)$, then, the directions of x in S are those of x in J together with $]x, +\infty[$ if this set is not empty. It follows that S is subcubic if J is a BJ-tree.

(5.9) **Lemma :** Every subcubic quasi-tree is $qt(fgs(J))$ for some SBJ-tree J .

Proof: We choose a maximal line A of the given subcubic quasi-tree S and distinct element a, b of A . By proposition (5.5.4), the partial order $\leq_{A,a,b}$ gives a binary join-tree K . By using the method of Proposition (3.5) with $U_0 := A$, we obtain a structuring J of K , making it into an SBJ-tree as defined in Definition (3.8). \square

(5.10) **Theorem:** The following properties of a subcubic quasi-tree S are equivalent:

- (1) S is regular,
- (2) S is described by a regular SBJ-scheme,
- (3) S is MS-definable.

Proof: By Lemma (5.9) and Proposition (3.19), every subcubic quasi-tree S is $qt(fgs(val(t)))$ for some term $t \in T^\infty(F)$.

Property (1) means that $S = qt(fgs(val(t)))$ for some regular term in $T^\infty(F)_t$. Let (1') mean that $S = qt(fgs(val(t)))$ for some regular term in $T^\infty(F)$. Then (1') \implies (2) by the similar implication in Theorem (3.21).

(2) \implies (3) by the similar implication in Theorem (3.21) and the observation that, in a quasi-tree S , the SBJ-trees J such that $S = qt(fgs(J))$ can be specified by MS formulas in terms of a 5-tuple (A, N_0, N_1, a, b) satisfying the formula $\varphi'(A, N_0, N_1, a, b)$ of the proof of Proposition (5.6).

(3) \implies (1') by the observation that the mapping α that transforms the relational structure $[t]$ for t in $T^\infty(F)$ into the subcubic quasi-tree $qt(fgs(val(t)))$ is an MS-transduction. The proof goes then as in Theorem (3.21).

The implication (1') \implies (1) is trivial and (1) implies that S is MS_{fin} definable by Theorem (5.6). But a term $t \in T^\infty(F)$ that defines S is MS-definable, and the relational structure representing a term has an MS-definable linear order. It follows that S has an MS-definable linear order, hence that S is MS-definable. \square

6 Conclusion

We have defined quasi-trees and join-trees of different kinds from regular terms. These terms have finitary descriptions. There are other infinite terms that have finitary descriptions: the algebraic ones [Cou83] and more generally, those of Caucal's hierarchy [Blu+]. Such terms also yield effective (algorithmically usable) notions of join-trees and quasi-trees. It is unclear whether the corresponding isomorphism problems are decidable.

7 References

- [Blu+] A. Blumensath, T. Colcombet and C.Löding, Logical theories and compatible operations, in *Logic and Automata: History and perspectives*, J. Flum et al. eds, Amsterdam University Press, 2008, pp. 73-106.
- [BluCou] A. Blumensath and B. Courcelle, Monadic second-order definable graph orderings. *Logical Methods in Computer Science* **10** (2014), issue 1.
- [Cou78] B. Courcelle, Frontiers of Infinite Trees. *ITA (Informatique Théorique et Applications)* **12** (1978) 317-339 (former name of the journal: *RAIRO Informatique théorique*).
- [Cou83] B. Courcelle, Fundamental properties of infinite trees. *Theor. Comput. Sci.* **25** (1983) 95-169.
- [Cou04] B. Courcelle, Clique-width of countable graphs: a compactness property. *Discrete Mathematics* **276** (2004) 127-148.
- [Cou14] B. Courcelle, Several notions of rank-width for countable graphs, 2014, to appear in *J. Comb. Theory, Ser. B*.

- [Cou15] B. Courcelle, Regularity equals monadic second-order definability for quasi-trees, in *Fields of Logic and Computation II, Lec. Notes Comput. Sci.* **9300** (2015) 129-141.
- [CouDel] 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.
- [CouEng] B. Courcelle and J. Engelfriet, *Graph structure and monadic second-order logic, a language theoretic approach*, Cambridge University Press, 2012.
- [Hei] S. Heilbrunner, An algorithm for the solution of fixed-point equations for infinite words. *ITA* **14** (1980) 131-141.
- [KriTho] I. Kriz and R. Thomas, Clique-sums, tree-decompositions and compactness. *Discrete Mathematics* **81** (1990) 177-185.
- [Oum] S. Oum, Rank-width and vertex-minors, *J. Comb. Theory, Ser. B* **95** (2005) 79-100.
- [OumSey] S. Oum and P. Seymour, Approximating clique-width and branch-width. *J. Comb. Theory, Ser. B* **96** (2006) 514-528.
- [Tho86] W. Thomas: On Frontiers of Regular Trees. *ITA* **20** (1986) 371-381.
- [Tho90] W. Thomas : Automata on Infinite Objects. in *Handbook of Theoretical Computer Science*, Volume B, Elsevier 1990, pp. 133-192

8 Appendix

A.1: Monadic second-order logic and related notions.

Monadic second-order logic extends first-order logic by the use of *set variables*, i.e., of variables X denoting subsets of the domain of the considered logical structure, and the atomic formulas $x \in X$, to express membership of x in X . We call *first-order* a formula where set variables are not quantified. For example, a first-order formula can express $X \subseteq Y$. A *sentence* is a formula without free variables.

Logical structures, graphs, trees, partial orders are finite or countably infinite.

Let \mathcal{R} be a finite set of relation symbols, each symbol R being given with an arity $\rho(R)$. We call it a *relational signature*. For every set of variables \mathcal{W} , we denote by $MS(\mathcal{R}, \mathcal{W})$ the set of MS formulas written with \mathcal{R} and free variables in \mathcal{W} .

An \mathcal{R} -*structure* is a tuple $S = (D_S, (R_S)_{R \in \mathcal{R}})$ where D_S is a finite or countable set, its *domain* and R_S is a relation on D_S of arity $\rho(R)$. A property P of \mathcal{R} -structures is *monadic second-order definable* (*MS-definable*) if it is equivalent to the validity, in every \mathcal{R} -structure S , of a monadic second-order sentence φ , which we denote by $S \models \varphi$.

For example, a graph G is identified with the $\{edg\}$ -structure (V_G, edg_G) where V_G is its vertex set and $edg_G(x, y)$ means that there is an edge from x to y , or between x and y if G is undirected. To take an example, 3-colorability is expressed by the MS-sentence :

$$\exists X, Y [X \cap Y = \emptyset \wedge \neg \exists u, v (edg(u, v) \wedge [(u \in X \wedge v \in X) \vee (u \in Y \wedge v \in Y) \wedge (u \notin X \cup Y \wedge v \notin X \cup Y)])].$$

Many properties of partial orders can also be expressed by MS sentences. We take examples that are useful in our proofs.

(a) The formula $Lin(X)$ defined as $\forall x, y. [(x \in X \wedge y \in X) \implies (x \leq y \vee y \leq x)]$ expresses that a subset X of N , partially ordered by \leq is linearly ordered.

(b) The formula $Lin(X) \wedge \exists a, b. [Min(X, a) \wedge Max(X, b) \wedge \theta(X, a, b)]$ expresses that X is linearly ordered and finite, where $Min(X, a)$ and $Max(X, b)$ are first-order formulas expressing respectively that X has a least element a and a largest one b , and $\theta(X, a, b)$ is an MS formula expressing that :

- (i) each element x of X except b has a successor c in X (i.e., c is the least element of $\{y \in X \mid y > x\}$), and
- (ii) $(a, b) \in Suc^*$, where Suc is the above defined successor relation (depending on X).

Assertion (ii) is expressed by the MS formula:

$$\forall U [U \subseteq X \wedge a \in U \wedge \forall x, y ((x \in U \wedge (x, y) \in Suc) \implies y \in U) \implies b \in U].$$

The reader will easily build the first-order formulas expressing Property (i), that $U \subseteq X$ and $(x, y) \in Suc$.

Without a linear order, the finiteness of a set X is not MS-expressible. It is thus useful to enrich MS logic with a *finiteness predicate* $Fin(X)$ expressing that a set X is finite. We denote by MS_{fin} the corresponding extension of MS logic.

If S is a relational structure $(N, \leq_t, (br_i)_{1 \leq i \leq \rho(F)}, (lab_f)_{f \in F})$ isomorphic to $[t]$ that represents a term $t \in T^\infty(F)$, then a linear order \sqsubseteq on N is MS-definable as follows:

$$x \sqsubseteq y : \iff x \leq_t y \vee (x \sqsubset_t y \text{ and } x \text{ is below the } i\text{-th son of } x \sqsubset_t y \\ \text{and } y \text{ is below the } j\text{-th son of } x \sqsubset_t y \text{ with } i < j).$$

The definability of linear orders by MS-formulas is studied in [CouBlu].

Monadic second-order transductions (*MS transductions*) are transformations of logical structures specified by MS or MS_{fin} formulas. We use them in the proofs of Theorems (3.21), (3.30), (4.11), (5.6) and (5.10). For these proofs, we only need very simple MS transductions, called *noncopying and parameterless* in [CouEng], and that we simply call *MS transductions*.

Let \mathcal{R} and \mathcal{R}' be two relational signatures. A *definition scheme* of type $\mathcal{R} \rightarrow \mathcal{R}'$ is a tuple of formulas of the form $\mathcal{D} = \langle \chi, \delta, (\theta_R)_{R \in \mathcal{R}'} \rangle$ such that $\chi \in MS(\mathcal{R})$, $\delta \in MS(\mathcal{R}, \{x\})$ and $\theta_R \in MS(\mathcal{R}, \{x_1, \dots, x_{\rho(R)}\})$ for each R in \mathcal{R}' . We define $\widehat{\mathcal{D}}(S) := S' = (D_{S'}, (R_{S'})_{R \in \mathcal{R}'})$ as follows:

S' is defined if and only if $S \models \chi$,
 $D_{S'}$ is the set of elements d of D_S such that $S \models \delta(d)$,
 $R_{S'}$ is the set of tuples $(d_1, \dots, d_{\rho(R)})$ of elements of D_S
such that $S \models \theta_R(d_1, \dots, d_{\rho(R)})$.

The mapping that associates the join-tree (N, \leq) with $[t]$ for $t \in T^\infty(F)$ (cf. Definition (3.15)) is an MS-transduction defined by $\mathcal{D} = \langle \chi, \delta, \theta_\leq \rangle$ such that χ expresses that the considered input structure S is isomorphic to $[t]$ for some $t \in T^\infty(F)$, $\delta(x)$ is $lab_{ext}(x)$ (expressing that x is in N) and $\theta_\leq(x, y)$ expresses that $x \leq y$, cf. Definition (3.15.b).

Our main tool is the following (well-known) result:

(A. 1) **Theorem** : Let \mathcal{D} be a definition scheme as above and $\varphi \in MS_{fin}(\mathcal{R}', \mathcal{X})$. There exists a formula $\varphi^{\mathcal{D}} \in MS_{fin}(\mathcal{R}, \mathcal{X})$ such that, for every \mathcal{R} -structure S , for every \mathcal{X} -assignment ν in D_S , we have $(S, \nu) \models \varphi^{\mathcal{D}}$ if and only if:

- (i) $S \models \chi$ (so that $\widehat{\mathcal{D}}(S) = S'$ is well-defined),
- (ii) ν is an \mathcal{X} -assignment in $D_{S'}$ (that is $\nu(x) \in D_{S'}$ and $\nu(X) \subseteq D_{S'}$ for $x, X \in \mathcal{X}$) and
- (iii) $(S', \nu) \models \varphi$.

Proof : The proof is given in [CouEng] (*Backwards Translation Theorem*, Theorem 7.10) for finite structures, for which the finiteness predicate $Fin(X)$ is of no use. However, it works for infinite structures as well and the predicate $Fin(X)$, that translates back to itself (under the assumption that $\nu(X) \subseteq D_{S'}$).

The formula $\varphi^{\mathcal{D}}$ is the conjunction of χ , a formula expressing (ii) and a formula φ' obtained from φ by replacing each atomic formula $R(x_1, \dots, x_r)$ by $\theta_R(x_1, \dots, x_{\rho(R)})$, i.e., by its definition given by \mathcal{D} . \square

It follows that, if the monadic theory of a class of structures \mathcal{S} is decidable and $S' = \widehat{\mathcal{D}}(S)$ for some definition scheme \mathcal{D} , then the monadic theory of \mathcal{S}' is decidable, because $S' \models \varphi$ holds for all S' in \mathcal{S}' if and only if $S \models \varphi^{\mathcal{D}}$ holds for all S in \mathcal{S} .

A.2 : Rank-width of countable graphs.

Rank-width and modular decomposition (cf. [CouDel]) motivate the study of quasi-trees and join-trees respectively. Here, we present the rank-width of countable graph. Rank-width is a width measure on finite graphs investigated first in [Oum] and [OumSey]. Here is its generalization to countable graphs.

We consider finite or countable, loop-free, undirected graphs without parallel edges. Let G be a graph, and X, Y be disjoint sets of vertices. The associated

adjacency matrix is $M : X \times Y \rightarrow \{0, 1\}$ with $M[x, y] = 1$ if and only if x and y are adjacent. If $U \subseteq X$ and $W \subseteq Y$, we denote by $M[U, W]$ the matrix that is the restriction of M to $U \times W$. Ranks are over $GF(2)$. The *rank* of M , defined as the maximum cardinality of an independent set of rows (equivalently, of columns) is denoted by $rk(M)$; it belongs to $\mathbb{N} \cup \{\omega\}$. It is convenient to take $rk(M[\emptyset, W]) = rk(M[U, \emptyset]) = 0$.

Fact: If $X \cup Y$ is infinite, then $rk(M) = \sup\{rk(M[U, W]) \mid U \subseteq X, W \subseteq Y, U \text{ and } W \text{ are finite}\}$.

The following def from [Cou14]. A *layout* of G is a subcubic quasi-tree T whose set of leaves is V_G . Its *rank* is the least upper-bound of the ranks $rk(M[X \cap V_G, X^c \cap V_G])$ such that X and $X^c = N_T - X$ are two convex subsets of N_T . (X is *convex* if $x, z \in X \wedge B(x, y, z) \implies y \in X$.) The *rank-width* of G , denoted by $rw(G)$, is the smallest rank of a layout. Its *discrete rank-width*, denoted by $rw^{dis}(G)$, is similar except that layouts are subcubic trees. Hence, $rw(G) \leq rw^{dis}(G)$. For finite graphs, we get the rank-width of [Oum]. The notation $G \subseteq_i H$ means that G is an induced subgraph of H .

(A.2) **Theorem** [Cou14]: For every graph G :

- (1) if $H \subseteq_i G$, then $rw(H) \leq rw(G)$ and $rw^{dis}(H) \leq rw^{dis}(G)$,
- (2) *Compactness* : $rw(G) = \sup\{rw(H) \mid H \subseteq_i G \text{ and } H \text{ is finite}\}$,
- (3) *Compactness with gap* : $rw^{dis}(G) \leq 2 \cdot \sup\{rw(H) \mid H \subseteq_i G \text{ and } H \text{ is finite}\}$.

The *gap function* in (3) is $n \mapsto 2n$, showing a weak form of compactness. A related gap concerns the clique-width of countable graphs [Cou04].

Proof sketch: (1) is clear from the definitions.

(2) is proved by Koenig's Lemma.

(3) is based on the representation of a countable linear order as the set of leaves of an ordered binary tree; this construction is adapted from [CouDel]. \square