Regularity equals monadic second-order definability for quasi-trees

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

April 1, 2015

Abstract : *Quasi-trees* generalize trees in that the unique "path" between two nodes may be infinite and have any finite or countable order type, in particular that of rational numbers. They are used to define the rank-width of a countable graph in such a way that it is the least upper-bound of the rankwidths of its finite induced subgraphs. *Join-trees* are the corresponding directed trees and they are also useful to define the modular decomposition of a countable graph. 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 definable by (are the unique model 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, algebra, regular term, monadic second-order logic.

1 Introduction

We define and study countable *quasi-trees* that generalize trees in 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 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] and does not need quasi-trees.

Our objective is to obtain finitary descriptions (usable in algorithms) of certain quasi-trees. For this purpose we define in [Cou15] an algebra of quasi-trees with finitely many operations such that the finite and infinite terms over these operations define all quasi-trees. The *regular quasi-trees* are those defined by regular terms. We prove in [Cou15] that a quasi-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.

In this introductory article, we focus on *binary join-trees* that can be seen as directed subcubic quasi-trees. A join-tree is defined as a partial order $(N, \leq)$ such that every two elements have a least upper-bound (called their *join*) and each set $\{y \mid y \geq x\}$ (denoted by $[x, +\infty[$) is linearly ordered. The modular decomposition of a countable graph is based on a join-tree [CouDel]. In [Cou15], we define algebras of rooted trees, of ordered rooted trees, of join-trees, of ordered join-trees and of quasi-trees. Binary join-trees and subcubic quasi-trees form subalgebras of two of them. In all cases, an object is regular if and only if it is monadic second-order definable.

A linear order whose elements are labelled by letters from an alphabet is called an *arrangement*. Regular arrangements are studied in [Cou78] and [Hei], and their monadic second-order definability is proved in [Tho]. We use and generalize these results to our generalized trees.

2 Definitions, notation and basic facts.

All ordered sets, trees and logical structures are finite or countably infinite.

We denote by ω the first infinite ordinal and also the linear order $(\mathbb{N}, \leq)$.

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 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.

2.1 Finite and infinite terms

Let F be a finite set of operations, each given with a fixed arity. We call such a set a *signature*. We denote by $T(F)$ (resp. $T^{\infty}(F)$) the set of finite (resp.

finite and infinite) terms written with the symbols of F. A typical example (easily describable linearly) is, with f binary and a and b nullary, the term $t_\infty = f(a, f(b, f(a, f(b, f(.....)))))$ that is the unique solution in $T^\infty(F)$ of the equation $t = f(a, f(b, t))$. Positions in a term 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.

We have a structure of F-algebra on $T^\infty(F)$ 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 $h : T^\infty(F) \to \mathbb{M}$. Its restriction to finite terms is uniquely defined.

Regular terms

A term $t \in T^\infty(F)$ as *regular* if there is a mapping h from $Pos(t)$ into a finite set Q and a mapping $\tau : Q \to F \times Seq(Q)$ such that:

> if u is an occurrence of a symbol f of arity k, then $\tau(h(u)) = (f, (h(u_1), ..., h(u_k)))$ where $(u_1, ..., u_k)$ is the sequence of sons of u. ($Seq(Q)$ is the set of finite sequences of elements of Q).

Intuitively, $\tau : Q \to F \times Seq(Q)$ is the transition function of a top-down deterministic automaton with set of states Q, $h(\varepsilon)$ is the 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 of unknowns of such a system is in bijection with Q). The above term t_∞ is regular.

A term t can be represented by the relational structure $\lfloor t \rfloor := (Pos(t), \leq_t, (br_i)_{1\leq i\leq \rho(F)}, (lab_f)_{f\in F})$ 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. It is regular if and only if $\lfloor t \rfloor$ is MS-definable, i.e., is up to isomorphism, the unique model of an MS sentence (see Thomas, [Tho90]).

2.2 Arrangements

We review a notion introduced in [Cou78] and further studied in [Hei, Tho]. Let X be set (say of letters). A linear order $(V, \leq)$ equipped with a labelling mapping $lab : V \to X$ 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. Every linear order $(V, \leq)$ is identified with the simple arrangement $(V, \leq, lab)$ such that $lab(v) := v$ for each v.

An arrangement can be considered as a generalized word. 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. 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, $t_\bullet = \bullet(a, \bullet(b, \bullet(a, \bullet(b, \bullet(.....))))))$ denotes the infinite word $abab...$. Its value is defined from $Occ(t_\bullet, \{a,b\}) = 2^*1$ lexicographically ordered by $1 < 21 < 221 < ...$, $lab(2^i1) = a$ if i is even and $lab(2^i1) = b$ 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 $w = w \bullet (a \bullet w)$.

If $w = (V, \leq, lab) \in \mathcal{A}(X)$ and $h : X \to Y$, then, $h(w) := (V, \leq, h \circ lab) \in \mathcal{A}(Y)$.

An arrangement is *regular* if it is denoted by a regular term. (The term $t_\bullet$ is regular). The arrangement a^η is also 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 over X by the relational structure $\lfloor w \rfloor := (V, \leq, (lab_a)_{a \in X})$ where $lab_a(u)$ is true if and only if $lab(u) = a$.

2.3 Trees and rooted trees

A *tree* is a nonempty, finite or countable, undirected, simple, connected graph. The set of nodes of a tree T is N_T.

A *rooted tree* is a tree equipped with a distinguished node called its *root*. Its edges are directed towards the root. The *level* of a node x is the number of edges of the path from it to 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 from x to the root. Then $x \sqcup_T y$ defined as the least upper bound of $\{x,y\}$ (the *join* of x and 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 $(N_{T/x}, \leq_T \restriction N_{T/x})$ 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 greatest element max and for each $x \in N$, the set $[x, max]$ is finite and linearly ordered.

2.4 Join-trees

We have used join-trees in [CouDel] for defining modular decomposition of countable graphs.

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

1) N is a 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*. 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 $\widehat{X}$ and we say that $\widehat{X}$ is *the top* of X.

(2.2) *Definitions : Directions and degrees.*

Let $J = (N, \leq)$ be a join-tree and x 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*. 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.

(2.3) *Definition : Structured binary join-tree.*

Let $J = (N, \leq)$ be a BJ-tree. For each set $X \subseteq N$, we denote by $\downarrow X$ the union of the convex sets $]-\infty, x]$ for $x \in X$. 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 can have a greatest element that we do not specify). The conditions are:

1) exactly one line of $\mathcal{U}$ has no strict upper-bound, hence, no top; we call it the *axis*, denoted by A; we also require that if A has a smallest element, then its degree is 0 or 1,

2) each other line U has a top $\widehat{U}$,

3) for each x in N, the sequence $y_0 = x, y_1, y_2, ...$ such that $y_{i+1} = \widehat{U(y_i)}$ is finite. Its last element is $y_k \in A$ (hence 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 $i = 0, ..., k-1$ and $[y_k, +\infty[$ are convex subsets of pairwise distinct lines of $\mathcal{U}$. We have $[x, +\infty[= [y_0, y_1[\cup [y_1, y_2[\cup ... \cup [y_k, +\infty[$, $[y_i, y_{i+1}[= U_+(y_i)$ 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 binary join-tree*, an *SBJ-tree* for short. Every linear order is an SBJ-tree whose elements are all of depth 0.

(2.4) *Example :* Figure 1 shows a structuring of a BJ-tree, where $\mathcal{U} = \{U_0, ..., U_5\}$, $A = U_0$. 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.

(2.5) *Definition : SBJ-trees as relational structures.*

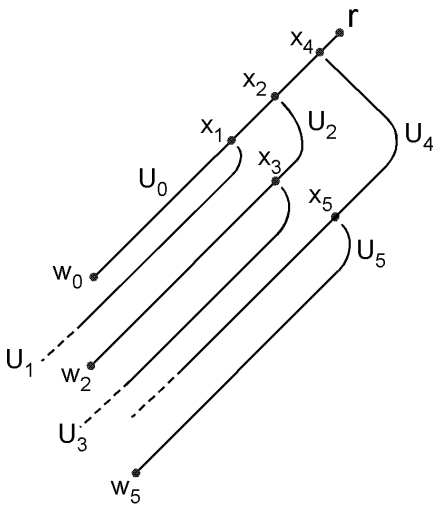

Figure 1: A structured join-tree.

Let $J = (N, \leq, \mathcal{U})$ be an SBJ-tree. Let $S(J)$ be 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).

If $X \subseteq N$ then $G(X) := (X, \rightarrow)$ is the directed graph such that $x \rightarrow y$ if and only if $x < y$ and $[x, y] \subseteq X$. We say that X is *laminar* if the connected components of $G(X)$ are lines, so that X is the union of pairwise disjoint lines of J, called the *components* of X.

(2.6) **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 structure $(N, \leq, N_0,\ N_1)$ is $S(J)$ for some SBJ-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 $U \in \mathcal{U} \wedge u = \widehat{U}$.

The proof is easy from the definitions. The construction of φ uses the fact that the finiteness of a linearly ordered set is MS-expressible.

(2.7) **Proposition** : Every join-tree has a structuring.

Proof sketch : Let $J = (N, \leq)$ be a join-tree. Let us choose an enumeration of N and a maximal line B_0 ; it contains each line $[x, +\infty[$ for $x \in B_0$. For each

$i > 0$, we choose a maximal line B_i containing the first node not in $B_{i-1} \cup ... \cup B_0$. We define $U_0 := B_0$ and, for $i > 0$, $U_i := B_i - (U_{i-1} \cup ... \cup U_0) = B_i - (B_{i-1} \cup ... \cup B_0)$. We define $\mathcal{U}$ as the set of lines U_i. $\square$

Each line is the linearly ordered set of leaves of an ordered binary rooted tree. By combining the trees of the lines of $\mathcal{U}$, we can build a binary tree that represents (is a precise MS-definable way) a BJ-tree. This type of construction has first been defined and used in [CouDel]. Proposition 3.6 below and the proofs in [Cou15] give it an algebraic meaning.

2.5 The rank-width of a countable graph

Rank-width and modular decomposition (cf. [CouDel]) motivate the study of quasi-trees. Rank-width is a width measure on finite graphs investigated first in [Oum] and [OumSey]. Here is its generalization to countable graphs. We let $\mathcal{G}$ be the class of finite or countable, loop-free, undirected graphs without parallel edges.

(2.8) *Definition* : *Rank-width.*

(a) Let $G \in \mathcal{G}$, let X and Y be pairwise disjoint sets of vertices. The associated *adjacency matrix* is $M : X \times Y \to \{0,1\}$ 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 $rk(M[\emptyset, W]) = rk(M[U, \emptyset]) = 0$.

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

(b) Let T be a binary join-tree with set of leaves V_G. We call it a *layout of* G. The *rank* of T is the least upper-bound of the ranks $rk(M[X \cap V_G, X^c \cap V_G])$ is $X \subseteq N_T$ is directed and downwards closed. The *rank-width* of G, denoted by $rwd(G)$, is the smallest rank of a layout. *Discrete rank-width*, denoted by $rwd^{dis}(G)$ is similar except that layouts are binary (countable) trees. Hence, $rwd(G) \leq rwd^{dis}(G)$. 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.

(2.9) **Theorem** [Cou14]: (1) If $G \subseteq_i H$, then $rwd(G) \leq rwd(H)$ and $rwd^{dis}(G) \leq rwd^{dis}(H)$,

(2) *Compactness* : $rwd(G) = \mathrm{Sup}\{rwd(H) \mid H \subseteq_i G \text{ and } H \text{ is finite}\}$,

(3) *Compactness with gap* : $rwd^{dis}(G) \leq 2.\mathrm{Sup}\{rwd(H) \mid H \subseteq_i G \text{ and } H \text{ is finite}\}$.

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].□

We now leave now rank-width and we consider binary join-trees.

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.

(3.1) Definition : *Operations on structured binary join-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:

$J \bullet J' := (N \cup N', \leq'', \mathcal{U}'')$ where :
$x \leq'' y :\Longleftrightarrow x \leq y \vee x \leq' y \vee (x \in N \wedge y \in A')$,
$\mathcal{U}'' := \{A \cup A'\} \cup (\mathcal{U} - \{A\}) \cup (\mathcal{U}' - \{A'\})$.

$J \bullet J'$ is an SBJ-tree with axis $A \cup A'$; its depth is the maximum of those of J and 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 $(N, \leq) + (N', \leq')$.

The empty SB-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 :
$x \leq' y :\Longleftrightarrow x \leq y \vee y = u$,
the axis is $\{u\}$.

Then $ext_u(J)$ 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 $ext_u(J)$.

Forgetting structuring:

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

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

(3.2) *The algebra* $\mathbb{SBJT}$

We let F be the signature $\{\bullet, ext, \Omega\}$. We obtain an algebra $\mathbb{SBJT}$ whose domain is the set of isomorphism classes of SBJ-trees. Concatenation is associative with neutral element Ω. We denote by $T^{\infty}(F)$ and $T(F)$ the sets of terms and finite terms over F.

(3.3) *Definitions : The value of a term.*

Let $t \in T^{\infty}(F)$.

(a) We compare positions of t as follows: $u \approx v$ if and only if every position w such that $u <_t w \leq_t u \sqcup v$ or $v <_t w \leq_t u \sqcup v$ is an occurrence of $\bullet$. This relation is an equivalence. We will also use the lexicographic order $\leq_{lex}$.

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

$N := Occ(t, ext)$, the set of occurences of ext (or ext_a if nodes are designated) in t,

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

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

(3.3.1) *Claim*: The mapping val is a value mapping into $\mathbb{SBJT}$.

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

(3.4) *Examples and remarks.*

(1) The term t that is the solution of the equation $t = t \bullet t$ denotes the empty structure Ω.

(2) Let t_1 be the solution in $T^{\infty}(F)$ of the equation $t = ext(ext(\Omega)) \bullet t$. We can write this term linearly by naming $a, b, c, d, e, f, ...$ the nodes created by the operations ext :

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

Its value is shown in Figure 2. The bold edges link nodes in the axis.

(3.5) *Definition : Flat terms*

(a) A term $t \in T^{\infty}(F)$ is *flat* if no occurrence of ext is below any other occurrence of ext. (Any two occurrences of ext are equivalent with respect to $\approx$). The value of a flat term is a simple arrangement over $Occ(t, ext)$.

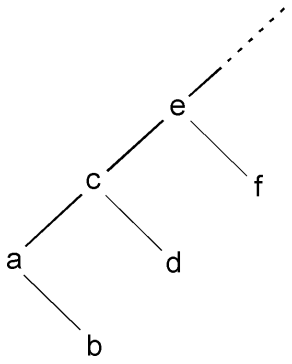

Figure 2: The SBJ-tree $val(t_1)$.

(b) Let $t \in T^\infty(F)$ and $u \in Pos(t)$. We denote by $\mathrm{Max}(t,u)$ the set of maximal occurrences of ext in t that are below or equal to u. We define $t\{u\}$ as the flat term obtained from t/u by replacing each of its subterms t/w by $ext_w(\Omega)$ for each $w \in \mathrm{Max}(t,u)$. In $t\{u\}$ the operations ext are indexed by positions from $Pos(t)$. It follows that $t\{\varepsilon\} = ext_\varepsilon(\Omega)$ if $t = ext(t')$. The value of $t\{u\}$ is a simple arrangement over $\mathrm{Max}(t,u)$.

For t_1 as in the previous example, $t_1\{\varepsilon\} = ext_a(\Omega) \bullet (ext_c(\Omega) \bullet (ext_e(\Omega) \bullet ...))$; it denotes the arrangement *ace...* (here we keep the original naming of positions). If $t_2 = ext(ext(\Omega)) \bullet (ext(ext(\Omega)) \bullet ext(ext(\Omega)))$ then $t_2\{\varepsilon\} = ext_1(\Omega) \bullet (ext_{21}(\Omega) \bullet ext_{22}(\Omega))$.

(3.5.1) *Claim* : Let $J = (N, \leq, \mathcal{U}) = val(t)$, cf. Definition 3.3. Then $val(t\{\varepsilon\}) = (A, \leq)$ (the axis) and, if $U \in \mathcal{U}$ and $\widehat{U} = p_t(u)$, we have $val(t\{u\}) = (U, \leq)$.

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

Proof sketch : Let $S = (N, \leq, \mathcal{U})$ be an SBJ-tree. For each U in $\mathcal{U}$, we define a flat term that denotes $(U, \leq)$. We combine these terms in order to get a term denoting S. $\square$

For the example of Figure 1, if t_i is a flat term denoting U_i, then we obtain the term $t_0[t_1, t_2[t_3], t_4[t_5]]$ where [...] denotes appropriate substitutions to occurrences of Ω.

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

An *SBJ-scheme* is a triple $S = (Q, w_{Ax}, (w_q)_{q \in Q})$ such that Q is a set, $w_{Ax} \in \mathcal{A}(Q)$ (is an arrangement over Q), and for each q, $w_q \in \mathcal{A}(Q)$. It is *regular* if Q is finite and the arrangements w_{Ax} and w_q are regular.

An SBJ-scheme S *describes* an SBJ-tree $J = (N_J, \leq, \mathcal{U})$ if there exists a mapping $r : N_J \to Q$ such that $r(A_J, \leq) = w_{Ax}$ and for every $x \in N_J, w_{r(x)} =$

$r(U, \leq)$ if $U \in \mathcal{U}$ and $\widehat{U} = x$, and $w_{r(x)} = \Omega$ if $x = \widehat{U}$, for no $U \in \mathcal{U}$. As $(U, \leq)$ is considered as the arrangement $(U, \leq, Id_U)$, its image under r is an arrangement over Q. We also say that S *describes* the BJ-tree $fgs(J)$.

(3.8) **Proposition**: Every SBJ-scheme S describes an SBJ-tree $J(S)$ that is unique up to isomorphism.

Proof sketch : Given $S = (Q, w_{Ax}, (w_q)_{q \in Q})$, we construct $J(S)$ by defining first its axis so as to be isomorphic to w_{Ax}, by a mapping r_{Ax}. Then we add the nodes at depth 1 by adding nonempty lines U isomorphic to $w_q \neq \Omega$ such that $\widehat{U} = x$ for each x in the axis such that $r_{Ax}(x) = q$. The isomorphism between such a line U and w_q is r_x. We proceed in a similar way with the nodes of depth 2,3.... We obtain an SBJ-tree. The mapping r is the union of the mappings r_{Ax} and r_x for all relevant x. Unicity holds because each step is forced, up to isomorphism. $\square$

(3.9) *Definition : Regular objects.*

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 $T^\infty(F)$.

(3.10) **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 sketch : (1)$\Longrightarrow$(2) Let $J = fgs(J')$ with J' denoted by a regular term t in $T^\infty(F)$. Let $h : Pos(t) \to Q$ and τ be as in Section 2.1. If x is an occurrence of ext with son u, the flat term $t\{u\}$ is regular. It defines the simple arrangement $(U, \leq)$ where $\widehat{U} = x$. The image $h(U, \leq)$ is a regular arrangement over Q.

Furthermore, if $h(x) = h(x') = q$, then the corresponding terms $t\{u\}$ and $t\{u'\}$ are isomorphic and $h(U, \leq)$ and $h(U', \leq)$ are also isomorphic (with $x' = p_t(u')$ and $\widehat{U'} = x'$). We can denote them by w_q. We let w_{Ax} be $h(val(t\{\varepsilon\}))$. These definitions give us a regular scheme describing J', hence, also J.

(2)$\Longrightarrow$(3) Let $J = (N, \leq)$ be a BJ-tree. (This property of $(N, \leq)$ is MS-expressible). Assume $J = fgs(J')$ where $J' = (N, \leq, \mathcal{U})$ is described by a regular SBJ-scheme R with $Q = \{1, ..., m\}$ and regular arrangements over Q : w_{Ax} and w_i for $i \in Q$. Let r be the corresponding mapping. For each $i \in Q$, let ψ_i be an MS sentence that characterize w_i up to isomorphism by the main result of [Tho86]. Similarly, ψ_{Ax} characterizes w_{Ax}. We claim that a relational structure $(N, \leq)$ is isomorphic to J if and only if :

there exist subsets $N_0, N_1, M_1, ..., M_m$ of N such that:

(i) $(N, \leq, N_0, N_1) = S(J'')$ for some SBJ-tree $J'' = (N, \leq, \mathcal{U})$,

(ii) $(M_1, ..., M_m)$ is a partition of N; we let r' maps each $w \in N$ to the unique i such that $w \in N_i$,

(iii) for every i and node u in M_i, the arrangement $r'(U)$ over Q such that $U \in \mathcal{U}$ and $u = \widehat{U}$ is isomorphic to w_i,

(iv) the arrangement $r'(A)$ over Q such that A is the axis of J'' is isomorphic to w_{Ax}.

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

By Proposition 2.6, Condition (i) is MS-expressible by $\varphi(N_0, N_1)$, and the property $U \in \mathcal{U} \wedge u = \widehat{U}$ is also MS-expressible in terms of N_0, N_1 by $\theta(u, U, U_0, N_1)$. Conditions (iii) and (iv) are MS-expressible by means of the sentences w_{Ax} and w_i suitably adapted to take $N_0, N_1, M_1, ..., M_m$ as arguments. Hence, J is the unique model of an MS sentence of the form:

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

(3)$\Longrightarrow$(1) By Definition 3.2, the mapping α that transforms the relational structure $\lfloor t \rfloor$ for t in $T^{\infty}(F)$ into the BJ-tree $J = (N, \leq) = fgs(val(t))$ is an MS-transduction: 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 (called the Backwards Translation Theorem, Theorem 7.10 in [CouEng]) that the set $L(\beta)$ of terms t in $T^{\infty}(F)$ such that $\alpha(\lfloor t \rfloor) \models \beta$ is MS-definable and thus, contains a regular term (a result by Rabin, see [Tho90]). This term denotes J, hence J is regular.$\square$

(3.11) **Corollary :** That two regular BJ-trees are isomorphic 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 (3.10) is effective: algorithms can convert a specification into another one. 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 $\alpha(\lfloor t \rfloor) \models \varphi$ (cf. the above proof of (3)$\Longrightarrow$(1)) if and only if $t \in L(\beta)$, which is decidable. $\square$

We have defined regular BJ-trees from regular terms, that have finitary descriptions. There are other infinite terms having finitary descriptions: the algebraic ones and more generally, those of Caucal's hierarchy [Cou83, Blu+]. Such terms yield effective notions of BJ-trees.

4 Conclusion

This algebraic approach and Theorem (3.10) extend to rooted trees, ordered rooted trees, join-trees and ordered join-trees of finite or countable degree. This theory is developped in [Cou15]. Quasi-trees can be characterized as the undirected join-trees. But they are defined independently, as follows.

(4.1) *Definition : Betweenness.*

If T is a tree, $x, y \in N_T$, its *betweenness relation* is the ternary relation B_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 :

$B_T(x, y, z) \Longleftrightarrow x, y, z$ are pairwise distinct and $x <_R y \leq_R x \sqcup_R z$ or $z <_R y \leq_R x \sqcup_T z$.

(4.2) **Proposition** : The betweenness relation $B = B_T$ of a tree T satisfies the following first-order properties for all u, x, y, z in N_T:

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

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

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

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 : $x \neq y \neq z \neq x \Rightarrow$
$B(x, y, z) \vee B(x, z, y) \vee B(y, x, z) \vee (\exists u. B(x, u, y) \wedge B(y, u, z) \wedge B(x, u, z))$.

(4.3) *Definition : Quasi-trees.*

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 of the conclusion of A7, there is at most one u satisfying $B(x, u, y) \wedge B(y, u, z) \wedge B(x, u, z)$. A *leaf* of (N, B) is a node z such that $B(x, z, y)$ holds for no x, y. Directions and degrees of nodes can be defined and we get the notion of subcubic quasi-tree.

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

$B_J(x, y, z) :\Longleftrightarrow x, y, z$ are pairwise distinct and $x < y \leq x \sqcup z$ or $z < y \leq x \sqcup z$,

and then, (N, B_J) is a quasi-tree denoted by $qt(J)$.

The algebra of quasi-trees is just the algebra of join-trees augmenting with the *forgetting operation qt* (similar to *fgs*). Subcubic quasi-trees are obtained in this way from BJ-trees. By selecting a suitable line in a subcubic quasi-tree and two nodes that fix its direction, we can make a quasi-tree into a BJ-tree. This construction is MS-definable. This indicates how Theorem (3.10) (in particular (2)$\Longrightarrow$(3)) extends to subcubic quasi-trees.

5 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.

[Cou78] B. Courcelle, Frontiers of Infinite Trees. ITA **12** (1978) (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, submitted.

[Cou15] B. Courcelle, Algebras of quasi-trees, in preparation, 2015.

[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.

[Die] R. Diestel, *Graph Theory*, 4th edition, Springer-Verlag, 2010.

[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