

Circle graphs and monadic second-order logic

Bruno Courcelle^{?,??}

^a*LaBRI, Université Bordeaux 1, CNRS
351 Cours de la libération
33405 Talence Cedex, France.*

Abstract

This article is part of a project consisting in expressing, whenever possible, graph properties and graph transformations in *monadic second-order logic* or in its extensions using modulo p cardinality set predicates or auxiliary linear orders. A *circle graph* is the intersection graph of a set of chords of a circle. Such a set is called a *chord diagram*. It can also be described by a word with two occurrences of each letter, called a *double occurrence word*. If a circle graph is *prime* for the *split (or join) decomposition* defined by Cunningham, it has a unique representation by a chord diagram, and this diagram can be defined by monadic second-order formulas with the *even cardinality set predicate*. By using the (canonical) split decomposition of a circle graph, we define in monadic second-order logic with auxiliary linear orders all its chord diagrams. This construction uses the fact that the canonical split decomposition of a graph can be constructed in monadic second-order logic with help of an arbitrary linear order. We prove that the *order of first occurrences* of the letters in a double occurrence word w that represents a connected circle graph determines this word in a unique way. The word w can be defined by a monadic second-order formula from the word of first occurrences of letters. We also prove that a set of circle graphs has bounded clique-width if and only if all the associated chord diagrams have bounded tree-width.

Key words: Monadic second-order logic, split decomposition, circle graph, chord diagram, order-invariant monadic second-order property, monadic second-order transduction.

¹ Email: Bruno.Courcelle@labri.fr. This work has been supported by the GRAAL project of "Agence Nationale pour la Recherche".

1 Introduction

The present article is part of a global project consisting in trying to formalize as much as possible graph properties and graph transformations in *monadic second-order logic*, or in its extensions by cardinality predicates and linear orders as explained below. Such formalizations frequently require reformulations of properties and of transformations, and some additional constructions. Characterizations of graph classes by forbidden configurations are generally useful. Monadic second-order logic (MS logic in short) applied to graphs is interesting for several reasons. First because the graph properties expressed in this language have polynomial algorithms for graphs of bounded tree-width or bounded clique-width ([16, 15]), informally, that have a certain tree structure. The deep reasons behind this result and its applications are surveyed in [15, 30]. Second, because the logical expression of graph transformations yields results on graph structure, for example that a class of graphs has bounded tree-width. The main result of Section 5 is of this type. Third, because this logical expression is an essential component of the extension of the *theory of formal languages* to the description of sets of finite or countable graphs. This latter aspect is developed in the book chapter [10].

In this article we consider *circle graphs*. A circle graph is the *intersection graph* of a *set of chords of a circle*. Such a set is called a *chord diagram*. An equivalent characterization can be given in terms of words where *each letter has two occurrences*. If a letter represents a chord, a set of chords of a circle is a word corresponding to the sequence of extremities of chords read around the circle, and the chords represented by a and b intersect if and only if the word can be written $aubvawbx$ for some words u, v, w, x . Other characterizations are reviewed in the survey article by Kozyrev and Yushmanov [28] and in the book by Spinrad [36].

Circle graphs have been introduced by Even and Itai in [21] in connection with algorithms that sort permutations by using stacks. This aspect is detailed in the book by Golumbic [24]. Applications of circle graphs are diverse, and without trying to be exhaustive, we can cite container ship stowage [3] and reconstruction of long DNA strings from short subsequences [1]. In graph theory they are also intensively studied because of their links with the *double cover conjecture* [23] and the structure of Eulerian trails in 4-regular graphs [2]. Last but not least, they play a role relatively to *vertex-minor inclusion* and *rank-width* that seems similar to that of *planar graphs* with respect to *minor inclusion* and *tree-width* : it is conjectured that for every bipartite circle graph, every graph with large enough rank-width (or large enough clique-width) has a vertex-minor isomorphic to H . (For rank-width and vertex-minors, see Oum [33].) This conjecture is proved for line graphs by Oum [34].

The construction of a circle graph from a chord diagram is immediate. The opposite construction is more difficult. The best algorithms for recognizing circle graphs and constructing chord diagrams are by Gabor *et al.* [22] taking time $O(mn)$ where n is the number of vertices and m the number of edges, and by Spinrad [35] taking time $O(n^2)$.

Our concern is to express the construction of one, and even of all chord diagrams defining a given circle graph by monadic second-order formulas (MS formulas). The first step is to recognize whether a graph is a circle graph. The characterization of circle graphs by three forbidden vertex-minors (Bouchet [5]) can be expressed by a *C₂MS formula*, i.e., an MS formula using the *even cardinality set predicate* (Courcelle and Oum [17]). However, this expression is *not constructive* : it verifies the absence of obstructions, but this absence gives no clue on how to construct a chord diagram for the considered circle graph. Our results (Theorems 5 and 11) yield on the contrary constructive characterizations, by MS formulas of the form $\exists X_1, \dots, X_k. \varphi$ such that a k -tuple X_1, \dots, X_k satisfying φ can be used by another MS formula to build the desired chord diagram. This type of construction of logical structures is called a *monadic second-order transduction*, by reference to the theory of formal languages.

The formulas we construct use in most cases auxiliary linear orders. Such a use in the expression of graph properties is related with the still open problem of finding a logical characterization of *polynomial time graph properties*. First-order logic with least fixed point operators, called *fixed point logic* (FPL) characterizes polynomial time graph properties for linearly ordered graphs by a classical result by Immermann and Vardi (see the book by Libkin [29]). Concerning monadic second-order logic, there are many situations where an auxiliary linear order is useful or even, perhaps, necessary. This is the case of the *even cardinality set predicate* yielding the extension of monadic second-order logic denoted by C₂MS. Even cardinality is a typical example of an *order-invariant monadic second-order property*, that is, of a property of unordered structures that is expressible by an MS formula using an arbitrary linear order. Its truth value, i.e., the parity of the cardinality, does not depend on the chosen linear order. By extending the Immermann-Vardi Theorem from Boolean queries to polynomial time transformations of structures (Dawar [19], Makowsky and Pnueli [31], Ebbinghaus and Flum [20]), one gets that the transformation of an ordered circle graph into a chord diagram representing it can be expressed in FPL.

We are interested by expressing in monadic second-order logic the mapping from a circle graph to *all its chord diagrams*, with or without auxiliary orders, with or without the even cardinality set predicate. Before presenting our results, we present our main graph theoretical tools.

We will use the *split decomposition* (also called *join decomposition*) of undirected graphs defined by Cunningham [18], which decomposes in a unique way a connected graph into a tree of basic graphs called its *components* : cliques, stars and prime graphs, which are not decomposable. This decomposition is constructible by an order-invariant MS transduction (Courcelle [14]). It fits very well with circle graphs because the components of the decomposition of a circle graph are circle graphs. It is used as a preliminary step in the algorithm of [22] : deciding if a graph is a circle graph reduces to deciding if its prime components are circle graphs (stars and cliques are circle graphs), and chord diagrams for the components can be combined to form a chord diagram of the considered circle graph.

A prime circle graph has a unique chord diagram representation. We prove that this unique diagram can be constructed by a C_2MS formula (Theorem 5, Section 3).

For constructing a chord diagram of a non-prime graph, we need a linear order: this order is used to construct the split decomposition (Proposition 3, Section 2), to define chord diagrams for stars and cliques, and since even cardinality is an order-invariant MS property, MS formulas can be used instead of C_2MS formulas for constructing the chord diagrams of the prime components. For constructing all chord diagrams, we need to use all (or at least several) linear orders : this is necessary for constructing the chord diagrams of cliques and stars. This result is Theorem 11 of Section 4.

Formulating these results in terms of double occurrence words is also interesting. If two words define the same connected circle graph and have the same *subword of first occurrences of letters*, then they are equal (Theorem 10, a new result). The (full) double occurrence word w can be reconstructed from the given circle graph and the linear order on its vertices defined by its word of first occurrences by an MS formula. This proof uses the canonical split decomposition of the considered graph and its definability by monadic second-order formulas using the linear order arising from the word of first occurrences of w . (Theorem 11, Section 4.)

Our constructions of monadic second-order transductions yield the fact that a set of circle graphs has bounded clique-width if and only if the set of its chord diagrams (which are 3-regular Hamiltonian graphs) has bounded tree-width. (Circle graphs have unbounded clique-width, since permutation graphs that are of unbounded clique-width [25] are particular circle graphs [24]).

This article is organized as follows. Split decomposition and the basic constructions in monadic second-order logic concerning it are reviewed in Section 2. Section 3 deals with circle graphs having a unique chord representation. We prove our first monadic second-order definability result. In Section 4 we obtain

the MS definition of all chord diagrams of circle graphs. In Section 5 we relate the clique-width of a circle graph and the tree-widths of its chord diagrams. A first appendix reviews definitions, basic properties and technical lemmas on MS logic and graph transformations expressed in MS logic. A second appendix contains the proofs of two technical results: that uniquely representable connected circle graphs are prime for the split decomposition, and that if two words define the same connected circle graph and have the same subword of first occurrences of letters, then they are equal.

2 The split decomposition

In this section we review the *split decomposition* of undirected graphs defined in [18] also called sometimes the *join decomposition*. It is used as a preliminary step in the polynomial time recognition algorithms of circle graphs of [4] and [22]. It is presented in a more detailed way in [14], and also for directed graphs. In the present article, we will use it only for undirected graphs. All words, graphs and relational structures will be finite.

Splitting a graph ; split (or join) decomposition.

Graphs are undirected and *simple* (without loops and multiple edges) unless we specify otherwise. A *split* of a connected graph G is a bipartition $\{A, B\}$ of V_G such that $E_G = E_{G[A]} \cup E_{G[B]} \cup (A_1 \times B_1)$ for some nonempty $A_1 \subseteq A$, $B_1 \subseteq B$, and each of A and B has at least 2 elements. If $\{A, B\}$ is a split, then G can be expressed as the union of $G[A]$ and $G[B]$ linked by a complete bipartite graph. The inverse of splitting is the *join operation*, defined as follows. Let H and K be two disjoint graphs with distinguished vertices h in H and k in K . We define $H \boxtimes_{(h,k)} K$ as the graph with set of vertices $V_H \cup V_K - \{h, k\}$ and edges $x - y$ such that, either $x - y$ is an edge of H or of K , or we have an edge $x - h$ in H and an edge $k - y$ in K . The subscript (h, k) in $\boxtimes_{(h,k)}$ will be omitted whenever possible.

If $\{A, B\}$ is a split, then $G = H \boxtimes_{(h,k)} K$ where H is $G[A]$ augmented with a new vertex h and edges $x - h$ whenever there are in G edges between x and some u in B . The graph K is defined similarly from $G[B]$, with a new vertex k . These new vertices are called *markers*. We say that h and k are *neighbour markers* if they are created for a same split. The graphs H and

K are connected, have at least 3 vertices and strictly less vertices than G . A technical variant (used in [18]) consists in letting $h = k$. In this case the graphs H and K have in common the marker vertex h and nothing else and we will write $G = H \boxtimes_{(h,h)} K$. The advantage is that $H \cup K$ is a single connected graph. However, the marker must be identified in some way. When one iterates the decomposition process, it is easier to think of the components of the decomposition as disjoint graphs.

A connected graph without split is said to be *prime*. Connected graphs with at most 3 vertices are thus prime. We will only decompose graphs with at least 4 vertices.

A *decomposition* of a connected graph G is defined inductively as follows : $\{G\}$ is the only decomposition of size 1 ; if $\{G_1, \dots, G_n\}$ is a decomposition of size n , and $G_n = H \boxtimes_{(h,k)} K$, then $\{G_1, \dots, G_{n-1}, H, K\}$ is a decomposition of G of size $n + 1$. The graphs G_i are called the *components* of the decomposition. They are connected and have at least 3 vertices, unless G has at most 2 vertices. The graph G can be reconstructed without ambiguity provided the marker vertices and their matchings are specified. We say that two components are *neighbours* if they have neighbour marker vertices. From the inductive definition of decompositions, it is clear that the components of a decomposition form an unrooted tree for the neighbourhood relation.

It will be convenient to handle a decomposition $\mathcal{D} = \{G_1, \dots, G_n\}$ of a graph G as a single graph $Sdg(\mathcal{D})$ called a *split decomposition graph*. The components of \mathcal{D} being pairwise disjoint, we let $Sdg(\mathcal{D})$ be their union together with particular edges labelled by ε and called the ε -edges between any two neighbour marker vertices. The other edges are called the *solid* edges. Every vertex of G is a vertex of $Sdg(\mathcal{D})$. No two ε -edges share vertices. The graph G can be reconstructed in a unique way from $Sdg(\mathcal{D})$. Two decompositions \mathcal{D} and \mathcal{D}' of a graph G are *isomorphic* if there exists an isomorphism of $Sdg(\mathcal{D})$ onto $Sdg(\mathcal{D}')$ which is the identity on V_G . The objective is to construct for every connected graph a *canonical decomposition* by iterated splittings.

Figure 1 shows a graph G and Figure 2 shows the graph representing its canonical split decomposition. The dotted lines are the ε -edges.

To illustrate these definitions, we observe that a prime graph with at least 4 vertices is 2-connected, that there is no prime undirected graph with 4 vertices, that for each $n \geq 5$, the graph C_n is prime, and the graphs P_n, K_n, S_{n-1} , all with n vertices, are not. As usual, we denote by K_n the *n-clique*, i.e., the complete graph with n vertices, by S_n the *n-star* consisting of one vertex, the *center*, adjacent to n vertices (it is thus a tree), by P_n the undirected path with $n - 1$ edges and n vertices, by C_n the undirected cycle with n vertices. The graphs K_n, S_{n-1} for $n \geq 4$ are "highly decomposable", or *brittle*

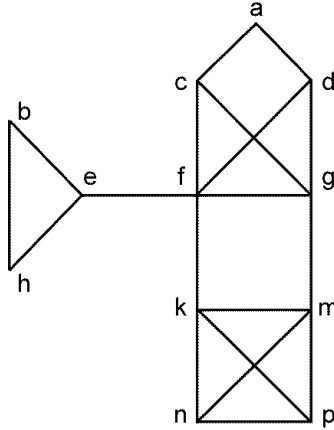


Figure 1. A graph G .

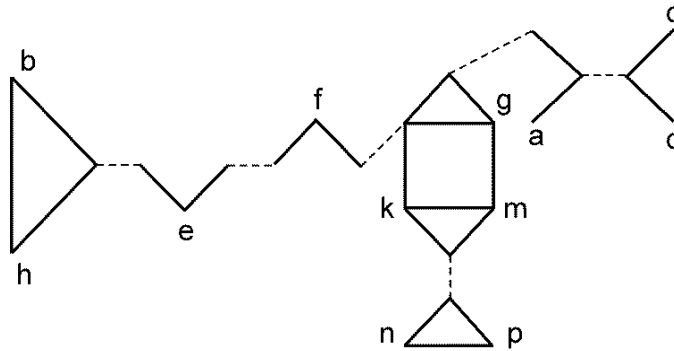


Figure 2. The split decomposition graph $Sdg(Split(G))$

in the terminology of [18]: every bipartition, each part of which has at least 2 vertices is a split. They are the only undirected graphs with this property. The 2-connected undirected graphs having 4 vertices are K_4 , C_4 , and K_4^- (i.e., K_4 minus one edge). None of them is prime.

Canonical decompositions

A decomposition of a connected graph G is *canonical* if and only if :

- (1) each component is either prime or is isomorphic to K_n or to S_{n-1} for n at least 3,
- (2) no two clique components are neighbour,
- (3) the two marker vertices of neighbour star components are both centers or both not centers.

Restrictions (2) and (3) can be justified as follows : if two clique components, isomorphic to K_n and K_m are neighbour they can be merged into a single one isomorphic to K_{n+m-2} . Similarly, if two star components, isomorphic to S_n

and S_m are neighbours, and the center of one is linked by an ε -edge to a non-center vertex of the other, they can be merged into a single star isomorphic to S_{n+m-1} . It is thus necessary to assume (2) and (3) in order to obtain a unique decomposition, because stars and cliques are brittle.

A split $\{A, B\}$ is *good* if it does not *overlap* any other split $\{C, D\}$ (where we say that $\{A, B\}$ and $\{C, D\}$ *overlap* if the intersections $A \cap C, A \cap D, B \cap C, B \cap D$ are all nonempty). Starting from a graph G and the decomposition $\{G\}$, one can refine it by iteratively splitting its components *with respect to good splits only*. Since a graph breaks into two strictly smaller graphs, one reaches a decomposition that cannot be refined by any split. Since one only applies good splits, one cannot generate neighbour components that are cliques or that are stars with a center marker neighbour to a non-center marker. It is thus canonical.

Proposition 1 [18, Theorem 3] : A connected undirected graph has a canonical decomposition, which can be obtained by iterated splittings relative to good splits. It is unique up to isomorphism.

In the sequel, we call this decomposition *the split decomposition*. By a *decomposition*, we will mean one which is not necessarily the canonical one. We have defined a single graph $Sdg(\mathcal{D})$ linking all components of a decomposition \mathcal{D} . We obtain in this way a binary relational structure on a fixed finite signature, actually an edge-labelled graph, from which the decomposed graph can be reconstructed by monadic second-order (MS in short) formulas, as we will see.

Evaluating split decomposition graphs.

For a split decomposition graph H , we let $Eval(H)$ can be the graph G defined as follows :

- (a) V_G is the set of vertices of H incident to no ε -edge,
- (b) the edges of G are the solid edges of H not adjacent to any ε -edge and the edges between x and y such that there is in H a path

$$x - u_1 - v_1 - u_2 - v_2 - \dots - u_k - v_k - y$$

where the edges $u_i - v_i$ are ε -edges and alternate with solid edges.

A *monadic second-order transduction* (an *MS transduction* in short) is a transformation of graphs, more generally of relational structures, expressible by MS formulas. Detailed definitions are given in the appendix.

Proposition 2 [14] : If \mathcal{D} is a decomposition of a connected graph G , then $Eval(Sdg(\mathcal{D})) = G$. The mapping $Eval$ is an MS transduction.

Proposition 3 [14] : There exists an order-invariant MS transduction that associates with a linearly ordered connected undirected graph the split decomposition graph representing its split decomposition.

Order invariant means that for any two linear orders, isomorphic relational structures are produced. See the appendix for more details.

3 Uniquely representable circle graphs

The logical expression of split decompositions of graphs stated in Proposition 3 is a basic tool for our study of circle graphs and the definition of their chord diagrams by MS formulas. In this section we review definitions and results from Bouchet [4, 5] and Gabor *et al.* [22], and we define by an C_2MS formula the (unique) representations of prime circle graphs by chord diagrams.

Circle graphs.

Let A be a countable set called the *set of letters*. We let W be the set of (finite) nonempty words over A having two occurrences or no occurrence of each letter. The elements of W are called *double occurrence words*. We let $V(w)$ be the set of letters occurring in w . The *alternance graph* $G(w)$ of w in W is the graph with set of vertices $V(w)$ and an undirected edge between a and b if and only if $w = u_1au_2bu_3au_4bu_5$ or $w = u_1bu_2au_3bu_4au_5$ for some u_1, \dots, u_5 in A^* . As in [2] we say in this case that letters a and b are *interlaced* in the word w .

The graphs $G(w)$ are also called *circle graphs* because they are the intersection graphs of finite sets of chords of circles defined as follows from w : if $w = a_1a_2\dots a_{2n}$, ($a_i \in A$), we let x_1, x_2, \dots, x_{2n} be consecutive points around a circle,

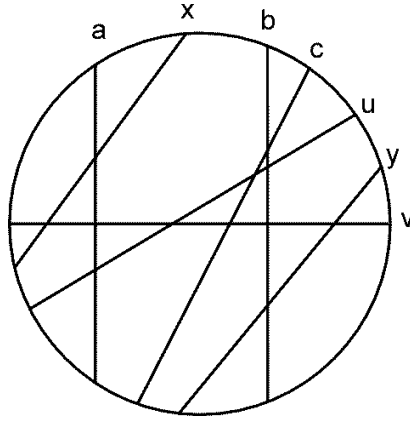


Figure 3. A chord diagram

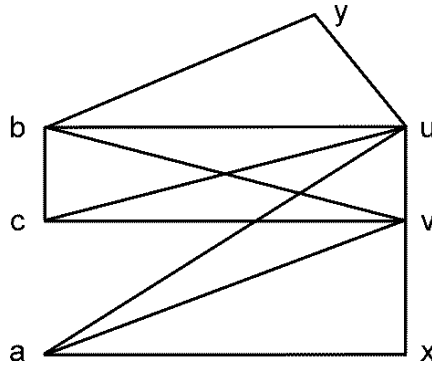


Figure 4. The associated circle graph

and we draw a chord named a between x_i and x_j if and only if $a_i = a_j = a$; the intersection graph of these chords is the graph with set of vertices $V(w)$ such that $a - b$ (which expresses in a short way : there is an undirected edge linking a and b) if and only if the chords a and b intersect. The graph is the same for any choice of chords as above because the exact positions do not affect crossings. The corresponding graph, equipped with a distinguished Hamiltonian cycle, is called the *chord diagram* of the double occurrence word. It is 3-regular. Figures 3 and 4 show the chord diagram and the circle graph associated with the word : $axbcuybycauxv$.

Circle graphs can also be geometrically represented as *overlap graphs* of intervals. See the survey by Kozyrev and Yushmanov [28] and the books by Golumbic [24] and Spinrad [36]. The representation of a circle graph as a set of chords is intuitively clear, but the one using a double occurrence word is more convenient for formal proofs. Both yield an appropriate relational structure (see Definition 3.2 below).

It is clear that $G(w) = G(w')$ if $w' = \tilde{w}$ (the *mirror image* of w) or if w and

w' are *conjugate*, denoted by $w \sim w'$, which means $w = uv$ and $w' = vu$ for some u, v in A^* . Let us say that w and w' are *equivalent*, denoted by $w \equiv w'$, if and only if either $w \sim w'$ or $\tilde{w} \sim w'$. This is an equivalence relation. Two equivalent words represent the same circle graph. A circle graph G is *uniquely representable* if $G = G(w) = G(w')$ implies $w \equiv w'$.

Every circle graph with at most 3 vertices is uniquely representable, as one can check in each case. The graphs C_4 , P_4 , the graph K_4^- are uniquely representable. The graphs K_4, S_3, I_4 are not. To take an example the star S_3 with center a is represented by the two inequivalent words $abcdadcb$ and $acbdadbc$.

Proposition 4 : A circle graph with at least 5 vertices is uniquely representable if and only if it is prime.

Proof : See [4, 22] for the "if" direction. The converse is claimed in [22] but a key assertion is declared as "clear" whereas it is not and deserves a proof. We give one in Appendix 2. \square

The split decomposition fits very well with circle graphs : a graph $H \boxtimes K$ is a circle graph if and only if H and K are circle graphs. Hence, every component of the canonical split decomposition of a circle graph is a circle graph. It follows in particular that a graph is a circle graph if and only if all its prime induced subgraphs are circle graphs.

The set of circle graphs has a characterization in terms of three forbidden *vertex-minors* ([5]; the terminology "vertex-minor" is from [33]). A graph H is a vertex-minor of a graph G if it is an induced subgraph of a graph G' obtained from G by a sequence of *local complementations* (see the definition in Appendix 2). The three forbidden vertex-minors are the cycles C_5, C_6, C_7 , each with one additional vertex and some edges. Vertex-minor inclusion is analogous to minor inclusion, however, its logical expression is more difficult. It is possible by means of MS formula written with the set predicate *Even* where $Even(X)$ expresses that a set X has even cardinality ([17]). This extension of MS logic is called *counting modulo 2 monadic second-order logic* and is denoted by C_2MS . A C_2MS -transduction is like an MS-transduction but written with C_2MS formulas. Our aim is to prove the following result which is a constructive version of the C_2MS definability of circle graphs :

Theorem 5 : There exists a C_2MS transduction that associates with every prime circle graph G a double occurrence word w such that $G(w) = G$.

In order to make this statement precise, we need to specify the relational structures which will represent double occurrence words.

Relational structures for double occurrence words and chord diagrams.

In order to handle finite words over an infinite alphabet we wish to use relational structures with finitely many relations, we cannot use the standard representations of words. With $w = a_1a_2\dots a_{2n}$ in W , we associate the relational structure $S(w) = \langle \{1, \dots, 2n\}, suc, slet \rangle$ where $suc(i, j)$ holds if and only if $j = i + 1$, with also $suc(2n, 1)$, and $slet(i, j)$ holds if and only if $i \neq j$ and $a_i = a_j$ ($slet$ means "same letter"), and the structure $\overline{S}(w) = \langle \{1, \dots, 2n\}, \overline{suc}, slet \rangle$ where $\overline{suc} = suc \cup suc^{-1}$.

It is clear that $w \equiv h(w')$ for some bijection h of the alphabet A extended into a monoid homomorphism $A^* \longrightarrow A^*$ if and only if $\overline{S}(w)$ is isomorphic to $\overline{S}(w')$, if and only if $S(w)$ is isomorphic to $S(w')$ or to its *reversal* $S(w')^{-1}$ obtained by replacing suc by suc^{-1} . Whether the letter at some position is a or b does not really matter. What matters is the bijection between the vertex set of $G(w)$ and the pairs of occurrences of each letter in w . For proving Theorem 5, we will construct a C_2 MS transduction associating with every prime circle graph G a structure $S(w)$ for some w in W such that $G(w) = G$.

The structures $S(w)$ and $\overline{S}(w)$ are graphs with $2n$ vertices and edges of two types. Any of them, depending on the context, will be called the chord diagram of w . The distinguished Hamiltonian cycle is represented by the relation suc or \overline{suc} . A connected circle graph is bipartite if and only if it has a planar chord diagram, if and only if all its chord diagrams are planar. In Section 5 we will compare in a similar way the clique-width of a circle graph and the tree-width of its chord diagrams.

The mapping that associates $G(w)$ with $S(w)$ is an MS transduction. Its easy definition is presented as an illustration of the notion of MS transduction in the appendix. The main results of this section and the next one consist in defining MS transductions that define $S(w)$ and $\overline{S}(w)$ from $G(w)$, hence that reconstruct some *forgotten information*. (The term "forgotten" is taken as in the notion of a *forgetful functor*).

Eulerian trails of 4-regular graphs

Before starting the proof of the theorem, we establish a technical lemma concerning the Eulerian trails of 4-regular simple graphs. Let H be a connected

4-regular simple graph. It has an *Eulerian trail*, defined as a cyclic sequence of vertices $E = (v_0, \dots, v_{k-1})$ such that $v_i - v_{i+1}$ for each $i = 0, \dots, k-1$, v_k is defined as equal to v_0 , and each edge of H is $v_i - v_{i+1}$ for exactly one i . This implies that each vertex occurs exactly twice in (v_0, \dots, v_{k-1}) . We consider (v_0, \dots, v_{k-1}) and $(v_i, \dots, v_{k-1}, v_0, \dots, v_{i-1})$ as the same cyclic sequences. We get a circle graph $G(E)$ with set of vertices V_H where $x - y$ if and only if $v_i = v_j = x$, $v_{i'} = v_{j'} = y$ and $i < i' < j < j'$ or vice versa by exchanging x and y .

We will build directed graphs with vertex set $V_H \times \{1, 2\}$. Consider a circuit C with vertex set $V_H \times \{1, 2\}$, formally defined as a cyclic sequence (x_0, \dots, x_{k-1}) where $x_i \rightarrow x_{i+1}$ for each $i = 1, \dots, k-1$ (which means : there exists a directed edge from x_i to x_{i+1}) and x_k is defined as equal to x_0 . We say it *represents the sequence* (v_0, \dots, v_{k-1}) if $x_i = (v_i, n_i)$ for each $i = 0, \dots, k-1$, (where $n_i = 1$ or 2). Several circuits may represent the same Eulerian trail, because the numbers 1 and 2 can be exchanged.

Lemma 6 : There exist two MS transductions that associate with every connected 4-regular simple graph H :

- (1) a set of circuits with vertex set $V_H \times \{1, 2\}$, that represent all Eulerian trails of H , and
- (2) the structures $\langle V_H, \text{edg}_H, \text{edg}_{G(E)} \rangle$ for all Eulerian trails E of H .

Proof : Let H be 4-regular. The graph $H \cup H^2$ has degree at most 16 ($= 4 + 3.4$), hence has a 17-vertex coloring $\gamma : V_H \rightarrow \{1, \dots, 17\}$, such that $\gamma(x) \neq \gamma(y)$ if x and y are at distance 1 or 2 in H . Let us fix such a coloring γ . It can be specified by a 17-tuple of sets of vertices Y_1, \dots, Y_{17} where $Y_i = \gamma^{-1}(i)$. An MS formula can check that such a tuple is indeed a 17-vertex coloring of $H \cup H^2$.

Consider now an Eulerian trail (v_0, \dots, v_{k-1}) of H . For $\alpha, \beta \in \{1, \dots, 17\}$, let $X_{\alpha, \beta}$ be the set of vertices v_i such that $0 \leq i \leq k-1$, $\alpha = \gamma(v_{i-1})$, $\beta = \gamma(v_{i+1})$, where $v_k = v_0$, $v_{-1} = v_{k-1}$.

The following properties hold :

- (a) If $\alpha = \beta$ then $X_{\alpha, \beta} = \emptyset$.
- (b) If $X_{\alpha, \beta} \cap X_{\alpha', \beta'} \neq \emptyset$, then, either $\alpha = \alpha'$ and $\beta = \beta'$ or $\{\alpha, \beta\} \cap \{\alpha', \beta'\} = \emptyset$.
- (c) Each vertex occurs in exactly two of the sets $X_{\alpha, \beta}$.

(d) For every edge $u - v$, if $u \in Y_\alpha$, $v \in Y_\beta$, then, for some γ and δ , either $u \in X_{\gamma,\beta}$ and $v \in X_{\alpha,\delta}$ or $u \in X_{\beta,\gamma}$ and $v \in X_{\delta,\alpha}$.

Due to fact (a), we will only use sets $X_{\alpha,\beta}$ for $\alpha \neq \beta$. These sets determine the trail: if we know that a vertex v follows on the trail a vertex u with color α and belongs to $X_{\alpha,\beta}$, then the vertex following v must have color β , hence is determined in a unique way, by the choice of the coloring. In total we need $289 (= 17^2 = 17 + 17^2 - 17)$ sets Y_1, \dots, Y_{17} and $X_{\alpha,\beta}$ for $\alpha \neq \beta$, to be used as parameters of a C_2 MS transduction (see Appendix 1 for definitions).

Claim 1 : The trail (v_0, \dots, v_{k-1}) can be reconstructed from the 289 sets Y_1, \dots, Y_{17} and $X_{\alpha,\beta}$, by means of MS formulas.

Proof : We let Y_1, \dots, Y_{17} and $X_{\alpha,\beta}, \dots$ be sets of vertices associated as explained above with a 17-vertex coloring γ and an Eulerian trail E of H . From Property (c), we can define $\delta(u, 1) = (\alpha, \beta)$, $\delta(u, 2) = (\alpha', \beta')$ if $u \in X_{\alpha,\beta}$, $u \in X_{\alpha',\beta'}$ and $(\alpha, \beta) < (\alpha', \beta')$ in the lexicographic order on pairs of integers. We define a binary relation on $V_H \times \{1, 2\}$ as follows :

$(u, i) \longrightarrow (w, j)$ if and only if :

- (i) $u - w$ in H ,
- (ii) $\delta(u, i) = (\alpha, \beta), \delta(w, j) = (\eta, \kappa)$ for some $\alpha, \beta, \eta, \kappa$ in $\{1, \dots, 17\}$,
- (iii) $\gamma(u) = \eta, \gamma(w) = \beta$.

We get thus a directed graph H^* with vertex set $V_H \times \{1, 2\}$. From properties (a)-(b) and the constraints on the coloring γ , it follows that every vertex in H^* has outdegree 1 and indegree 1. We prove that H^* is a circuit representing E .

Let $\alpha = \gamma(v_{-1}), \beta = \gamma(v_1)$. Hence $v_0 \in X_{\alpha,\beta}$. For some i_0 , $\delta(v_0, i_0) = (\alpha, \beta)$.

We let $x_0 = (v_0, i_0)$. We consider the unique directed path in H^* : $x_0 \longrightarrow x_1 \longrightarrow \dots \longrightarrow x_n, n < k$.

The element x_1 is the unique (w, j) such that $(v_0, i_0) \longrightarrow (w, j)$. This pair is equal to (v_1, i_1) and $\delta(v_1, i_1) = (\eta, \kappa), \gamma(v_0) = \eta, \gamma(v_2) = \kappa$.

Similarly, $x_2 = (v_2, i_2)$ for some $i_2 = 1$ or 2 . Using induction, we can see that, for all $m < k$, $x_m = (v_m, i_m)$ for some $i_m = 1$ or 2 . Hence the unique directed path in H^* starting from (v_0, i_0) is $(v_0, i_0) \longrightarrow (v_1, i_1) \longrightarrow \dots \longrightarrow (v_n, i_n)$ for some i_1, \dots, i_n , and $n = k - 1$. We also have $(v_n, i_n) \longrightarrow (v_0, i_0)$. Hence

H^* is a circuit representing the Eulerian trail (v_0, \dots, v_{k-1}) . The definition of the edge relation of H^* by (i)-(iii) is clearly expressible in MS logic in terms of the sets $Y_1, \dots, Y_{17}, \dots, X_{\alpha,\beta}, \dots$ \square

Assume now that $Y_1, \dots, Y_{17}, \dots, X_{\alpha,\beta}, \dots$ (for $\alpha, \beta \in \{1, \dots, 17\}, \alpha \neq \beta$) are arbitrary subsets of V_H , not necessarily arising from an Eulerian trail of H . One can construct an MS formula $\theta_1(Y_1, \dots, Y_{17}, \dots, X_{\alpha,\beta}, \dots)$ expressing that Y_1, \dots, Y_{17} define a 17-coloring of $H \cup H^2$, and properties (a)-(d) hold. An MS transduction μ using set parameters $Y_1, \dots, Y_{17}, \dots, X_{\alpha,\beta}, \dots$ (that satisfy θ_1) can build a directed graph H^* with vertex set $V_H \times \{1, 2\}$ and edge relation defined by conditions (i),(ii),(iii) of Claim 1. We denote it by $H^*(Y_1, \dots, Y_{17}, \dots, X_{\alpha,\beta}, \dots)$ if we need to specify the parameters. (Since properties (b) and (c) are assumed to hold, the mapping δ is well-defined). By backwards translation relative to μ applied to the MS formula expressing that H^* is a circuit going through all vertices in $V_H \times \{1, 2\}$, one obtains an MS formula $\theta_2(Y_1, \dots, Y_{17}, \dots, X_{\alpha,\beta}, \dots)$. It is clear that if H^* is a Hamiltonian circuit it represents an Eulerian trail of H , because by Property (d) each edge is traversed once and only once by the trail that is represented by H^* . By the first part of the proof, all Eulerian trails can be represented in this way. This gives the first assertion of Lemma 6.

Claim 2 : There exists an MS formula $\theta_3(x, y, Y_1, \dots, Y_{17}, \dots, X_{\alpha,\beta}, \dots)$ expressing in a 4-regular graph H that a tuple $(Y_1, \dots, Y_{17}, \dots, X_{\alpha,\beta}, \dots)$ of subsets of V_H defines an Eulerian trail E and that the binary relation :

$$\{(x, y) \mid x, y \in V_H, H \models \theta_3(x, y, Y_1, \dots, Y_{17}, \dots, X_{\alpha,\beta}, \dots)\}$$

is the adjacency relation $edg_{G(E)}$ of $G(E)$.

Proof : The relation $edg_{G(E)}$ is characterized by $(x, y) \in edg_{G(E)}$ if and only if :

For some $i, j \in \{1, 2\}$, we have in H^* (for $(Y_1, \dots, Y_{17}, \dots, X_{\alpha,\beta}, \dots)$ satisfying θ_2) a path of the form :

$$(x, i) \longrightarrow \dots \longrightarrow (y, j) \longrightarrow \dots \longrightarrow (x, 3 - i) \longrightarrow \dots \longrightarrow (y, 3 - j)$$

Since the edge relation of H^* is MS definable in $(H, Y_1, \dots, Y_{17}, \dots, X_{\alpha,\beta}, \dots)$, we obtain that $edg_{G(E)}$ is also MS definable in $(H, Y_1, \dots, Y_{17}, \dots, X_{\alpha,\beta}, \dots)$. This gives the desired formula θ_3 . \square

This proves the second assertion of Lemma 6. \square

We now prove the first main theorem of this subsection.

Proof of Theorem 5 : We only consider prime circle graphs with at least 5 vertices. The finitely many graphs with less vertices can be handled as particular cases.

Let w be a double occurrence word such that $G = G(w)$ is prime with at least 5 vertices, let $a, b \in V(w)$, $a \neq b$. We say that a and b are *neighbours* in w if $w \equiv abw'$ for some w' in A^* . (This notion of neighbourhood is not related with that of marker vertices used in Section 2.) This means that in the chord representation of w , chords a and b have two ends that are consecutive on the circle. If $G(w)$ is prime with at least 5 vertices, and by the unicity property of Proposition 4 ("if" direction), this notion depends only on the graph $G(w)$, and not on the word w representing it.

Claim 1 : If $G(w)$ is prime with at least 5 vertices and $w \equiv abw'$, then $w' = u_1au_2bu_3$ or $w' = u_1bu_2au_3$ for some nonempty words u_1, u_2, u_3 in A^* .

Proof : Since w is a double occurrence word, w' is either $u_1au_2bu_3$ or $u_1bu_2au_3$ for some u_1, u_2, u_3 in A^* .

First case : $w' = u_1au_2bu_3$. If u_1 or u_2 or u_3 is empty, then $\{\{a, b\}, V(w) - \{a, b\}\}$ is a split of $G(w)$.

Second case : $w' = u_1bu_2au_3$. If u_1 or u_3 is empty, then $G(w)$ is not connected, hence is not prime. If u_2 is empty, then $\{\{a, b\}, V(w) - \{a, b\}\}$ is a split.

These two cases are thus excluded by the hypothesis, which completes the proof. \square

It follows that each letter occurring in w has four different neighbours.

Let $S(y)$ be a chord diagram. Its *neighbourhood graph* is the graph $N(S(y))$ with vertex set $V(y)$ and an edge $a - b$ if and only if a and b are consecutive in the double occurrence word y .

If $G = G(w)$ is prime with at least 5 vertices, $N(S(w))$ depends only on G and can be denoted by $N(G)$. This graph is 4-regular. We will prove that its adjacency relation is definable by a C_2MS formula over the given graph G , and that w can be constructed from $N(G)$.

Example : Figure 5 shows with solid lines the graph $N(S(w))$ for the chord

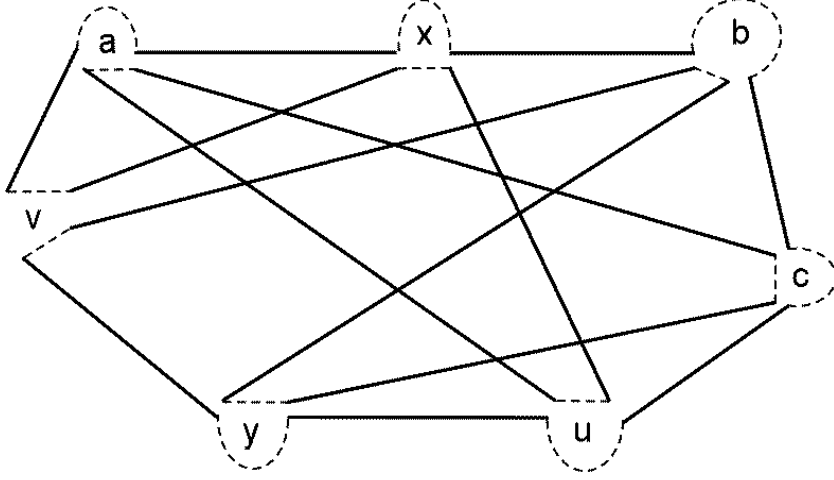


Figure 5. The neighbourhood graph $N(G)$

diagram $S(w)$ shown on Figure 3. The dotted lines around the vertices show the Eulerian trail which corresponds to the chord diagram of $G = G(w)$. \square

For $a, b \in V_G(\subset A)$, $a \neq b$, $u, v \in A - V_G$, we let $G(a, b; u, v)$ be the graph G augmented with the path $a - u - v - b$.

Claim 2 : $G(a, b; u, v)$ is a circle graph if and only if a, b are neighbours in G .

Proof : Let $G = G(w)$ where $w \equiv abw'$, then $G(a, b; u, v) = G(uavubvw')$, as one checks easily. Hence $G(a, b; u, v)$ is a circle graph.

Let us conversely assume that $G(a, b; u, v)$ is a circle graph $G(z)$. Let z_1 be obtained from z by deleting all occurrences of u and v . Hence $G(z_1) = G$, and $z_1 \equiv abw'$, since G is uniquely representable.

Let z_2 be obtained from z by deleting all occurrences of letters in $A - \{a, b, u, v\}$. Hence $G(z_2)$ is $a - u - v - b$ (i.e., P_4) or is the same graph with also an edge between a and b (i.e., C_4). Since the graphs P_4 and C_4 are uniquely representable, $z_2 \equiv uavubvz_3$, where z_3 is ba or ab respectively. We can thus transform the word z into an equivalent word z' in such a way that, by deleting from z' the letters in $A - \{a, b, u, v\}$ we get $uavubvz_3$. Furthermore, we can take such z' of the form $ux_1ax_2vx_3ux_4bx_5vx_6$ for some $x_1, \dots, x_5 \in (A - \{a, b, u, v\})^*$ and some $x_6 \in (A - \{u, v\})^*$.

Consider an occurrence of letter c in x_3 . Since c is not adjacent to u in $G(a, b; u, v)$ its other occurrence must be in x_1 , in x_2 or in x_3 . Since c is not adjacent to v , its other occurrence must be in x_3 , or in x_4 or in x_5 . Hence it must

be in x_3 . Hence x_3 is a double occurrence word. It defines one or more connected components, not containing a , contradicting the fact that $G(a, b; u, v)$ is connected. Hence x_3 must be empty.

Consider now an occurrence of letter c in x_1 or in x_2 . Since c is not adjacent to u and x_3 is empty, its other occurrence must be in x_1 or in x_2 . Hence the letters in x_1 and x_2 either form connected components not containing a , or if this is not the case, then a is a separating vertex in G . But G is connected and has no separating vertex since it is prime, hence x_1 and x_2 must be empty. By considering similarly v and b , one gets that x_4 and x_5 are empty. Hence the word obtained from z' by removing letters u and v is of the form abx'_6 . Hence a and b are neighbours, as was to be proved. \square

Claim 3 : That a and b are neighbours in G is expressible by a C_2 MS formula. Hence the mapping associating $N(G)$ with a prime circle graph G is a C_2 MS transduction.

Proof : The mapping from (G, a, b) to a graph isomorphic to $G(a, b; u, v)$ is an FO transduction, say η . A C_2 MS formula γ can test whether $G(a, b; u, v)$ is a circle graph by [17]. By backwards translation of γ through η (see Appendix 1), we get a C_2 MS formula $\gamma^\#(a, b)$ expressing that a and b are neighbours in G . The second assertion holds because the relation $edg_{N(G)}$ of the structure $N(G) = \langle V_G, edg_{N(G)} \rangle$ is defined by the C_2 MS formula $\gamma^\#$. \square

End of the proof of Theorem 5 : That a given graph G is prime is straightforward to write in MS logic. Hence, that G is a prime circle graph with at least 5 vertices is a C_2 MS property. Assuming it satisfied and with Claim 3, one can build from G and by a C_2 MS transduction the 4-regular graph $N(G)$. This graph is connected and has an Eulerian trail E such that $G(E) = G$. The Eulerian trails of $N(G)$ are defined by the 289-tuples $(Y_1, \dots, Y_{17}, \dots, X_{\alpha, \beta}, \dots)$ of subsets of V_G which satisfy formula θ_2 of Claim 1 of Lemma 6.

Since the binary relation $edg_{G(E)}$ on V_G can be defined from the tuple representing E (using formula θ_3 of Claim 2 of Lemma 6), one can find the tuples for which the corresponding trail E satisfies $edg_{G(E)} = edg_G$. The corresponding circuit graphs $N(G)^*(Y_1, \dots, Y_{17}, \dots, X_{\alpha, \beta}, \dots)$ (with vertex set $V_G \times \{1, 2\}$) represent double occurrence words w such that $G(w) = G$. Since G is uniquely representable, one obtains two structures $S(w)$ and $S(w)^{-1}$ up to isomorphism, one being the reversal of the other, and a unique structure $\overline{S}(w)$ up to isomorphism. \square

Corollary 7 : There exists an order invariant MS transduction that associates with a prime circle graph $G(w)$ the structure $\overline{S}(w)$ representing its unique chord diagram.

Proof : Because if the given graph G is linearly ordered, the set predicate $Even(X)$ can be expressed by an MS formula using the linear ordering (see [10]), and thus the C_2MS formulas and C_2MS transductions used in the previous results can be replaced by MS formulas and MS transductions. The transduction uses parameters, but all choices of parameters yield the same structure $\overline{S}(w)$ up to isomorphism (where $G = G(w)$). The linear order makes possible to specify the unique lexicographically minimal set of parameters satisfying the required condition, hence to eliminate parameters. From different linear orders, one gets different lexicographically minimal sets of parameters but the same output structure. Hence the MS transduction is order-invariant. \square

Comparability graphs.

A similar proof is done in [13] for comparability graphs. If a comparability graph is prime with respect to modular decomposition, it has a unique transitive orientation ("unique" is meant up to reversal, which does not modify the comparability graph; see Kelly [27] or [32]). Proposition 5.2 of [13] establishes that this orientation is MS definable. The proof uses the characterization of comparability graphs by forbidden induced subgraphs, so that to be a comparability graph is an MS definable property. An MS formula can check whether two edges $x - y$ and $w - z$ must be directed "in the same way" $x \longrightarrow y$ and $w \longrightarrow z$ (or $x \longleftarrow y$ and $w \longleftarrow z$) in any of the two transitive orientations. This formula applies the *MS definable test of comparability* to a graph consisting of G augmented with a path $x - u - v - w$ for new vertices u and v and a few other edges between u, v and the neighbours of x and w . Because G has a unique transitive orientation, the answers given for all edges $x - y$ assuming chosen the orientation $w \longrightarrow z$ are compatible, and one can thus, edge by edge determine it. There is thus a striking similarity with the proof of Theorem 5 which also rests on a membership test based on forbidden configurations and on the unicity of a representation that insures that all elementary tests do not arise from different incompatible representations.

A question

Can we use in Theorem 5 an MS transduction instead of a C_2MS transduction, that is, can we avoid using the $Even(X)$ predicate ? This would be true by

our proof if the set of circle graphs would be MS definable. But the logical characterization we use is based on their characterization by three forbidden vertex-minors, and we do not know how to express that a graph contains a given vertex-minor by an MS formula without using the even cardinality set predicate $Even(X)$ ([17]).

Conversely, if Theorem 5 holds for some MS transduction, then the set of prime circle graphs is MS definable. So is the set of circle graphs because, as we noticed at the beginning, a graph is a circle graph if and only if all its prime induced subgraphs are circle graphs.

We think unlikely that the set predicate $Even(X)$ can be avoided because the theory of circle graphs makes a crucial use of vector spaces over the 2 element field $GF(2)$, and $Even(X)$ is thus necessary for computing the values of sums over $GF(2)$.

4 A logical definition of all chord diagrams of a circle graph

If a circle graph splits as $H \boxtimes K$, then H and K are circle graphs. It follows that the components of the split decomposition of a circle graph are circle graphs. The prime ones have unique representations by Proposition 4. The representations of a star S_n with center a are the words $awa\tilde{w}$ where w ranges over the permutations of an alphabet with n letters not containing a (i.e., the words with one and only one occurrence of each letter). The representations of a clique K_n are the words ww where w ranges over the permutations of a finite alphabet with n letters. If we have a split decomposition of a circle graph G , and a representation of each component, then we can combine the representations of the components to build a representation of G . These constructions can be formalized in MS logic.

For prime graphs, the chord diagrams are obtained by Theorem 5. For a clique K_n with vertex set V ordered by $a_1 < a_2 < \dots < a_n$, an MS transduction taking as input $(V, <)$ can construct the chord diagram $S(a_1a_2\dots a_na_1a_2\dots a_n)$ representing $K_n = G(a_1a_2\dots a_na_1a_2\dots a_n)$. For a star S_n with center a and vertices ordered by $a < b_1 < b_2 < \dots < b_n$, an MS transduction can construct the chord diagram $S(ab_1b_2\dots b_nab_nb_{n-1}\dots b_1)$ representing $S_n = G(ab_1b_2\dots b_nab_nb_{n-1}\dots b_1)$. In both cases, all representations (up to equivalence) of K_n and S_n can be obtained by two fixed MS transductions taking as input all permutations of the set of vertices. Our aim is to prove that there exists an MS transduction that defines for every linearly ordered circle graph, a double occurrence word

representing it. We will actually prove a stronger result but we need first some definitions and lemmas on double occurrence words.

Consider two connected circle graphs H and K , represented by double occurrence words v and w , such that $V(v) \cap V(w) = \{a\}$. The graphs H and K have vertices labelled by the letters in $V(v)$ and $V(w)$, and a single vertex in common. We are in the case described at the end of Definition 2.1. We will say that the words $v = v_1av_2a$ and $w = w_1aw_2a$ are *composable*. By the connectivity assumptions on H and K , the words v_1, v_2, w_1, w_2 are not empty.

Lemma 8 : The connected graph $H \boxtimes_{(a,a)} K$ is the circle graph represented by the four words $v_1w_1v_2w_2$, $v_1w_2v_2w_1$, $v_1\widetilde{w}_1v_2\widetilde{w}_2$ and $v_1\widetilde{w}_2v_2\widetilde{w}_1$.

We let $v \boxtimes w$ denote this set of four words, up to equivalence. One may obtain four pairwise inequivalent words. This the case for example if $v = bcdabdca$ and $w = efgaegfa$. In particular cases, the set $v \boxtimes w$ may contain less than four words up to equivalence. The following proposition is a converse.

Proposition 9 : Let w be a double occurrence word such that $G(w)$ is connected. Let $\{A, B\}$ be a good split of $G(w)$ with corresponding decomposition $H \boxtimes_{(h,h)} K$. Then $w \sim w_H^1w_K^1w_H^2w_K^2$ where $G(w_H^1hw_H^2h) = H$ and $G(w_K^1hw_K^2h) = K$.

Proof : Let $A' \subseteq A$ and $B' \subseteq B$ be the sets of vertices of H (resp. K) linked to some vertex of K (resp. H). We say that letter a *crosses* letter b if in the chord representation of w , chords a and b intersect, i.e. if $a - b$ in $G(w)$.

Let w_H and w_K be the words obtained from w by removing the letters from B and from A respectively. Without loss of generality, we can assume that $w = w_H^1w_K^1w_H^2w_K^2\dots w_H^nw_K^n$ with all factors w_H^i, w_K^j nonempty, $w_H = w_H^1w_H^2\dots w_H^n$ and $w_K = w_K^1w_K^2\dots w_K^n$. Then our aim is to prove that $n = 2$, $w \sim w_H^1w_K^1w_H^2w_K^2$, $G(w_H^1hw_H^2h) = H$ and $G(w_K^1hw_K^2h) = K$.

If $n = 1$ then G is not connected. This is excluded by the hypothesis.

If $n = 2$, we must check that $G(w_H^1hw_H^2h) = H$ and $G(w_K^1hw_K^2h) = K$. Since every letter a of A' crosses every letter b of B' , each of w_K^1 and w_K^2 contains occurrences of all letters in B' . Select one say b , delete from w all letters from $B - \{b\}$, then one obtains the word $w_H^1bw_H^2b$ which defines H , with b instead of h . Using similarly some a in A' , one obtains that $w_K^1aw_K^2a$ defines K , with a instead of h . This gives the desired result.

It remains to prove that the case $n \geq 3$ yields a contradiction with the assumptions that $G(w)$ is connected and $\{A, B\}$ is a good split.

Without the hypothesis that $\{A, B\}$ is good we may have $n > 2$. Take for example $w = abadcfebfdce$, $A = \{a, b, c\}$, $B = \{d, e, f\}$. In this case, $n = 4$.

Claim 1 : If a belongs to $A - A'$ its two occurrences are in a same factor w_H^i .

Proof : Assume the contrary. Wlog, $a \in w_H^1 \cap w_H^i$ for $i > 1$. (This a short writing for " a has one occurrence in w_H^1 and the other in w_H^i "). No b in B crosses a . Hence B is the union of two sets B_1 and B_2 , such that all occurrences of elements of B_1 are in $w_K^1 w_K^2 \dots w_K^{i-1}$, and all occurrences of elements of B_2 are in $w_K^i \dots w_K^n$. Informally, B_1 and B_2 are separated by a in a chord representation and no chord of one set crosses a chord from the other.

If $B_i \cap B'$ is empty for some i , then $G[B_i]$ is a connected component of G and G is not connected, contradicting the assumption.

Hence every $a' \in A'$ crosses some b in B_1 and some b' in B_2 .

There exist j, j', k, k' such that $1 \leq j < j' < i \leq k < k' \leq n$, $b \in w_K^j \cap w_K^{j'}$ and $b' \in w_K^k \cap w_K^{k'}$.

We let A_0 be the set of letters from $A - A'$ having their two occurrences in $w_H^1 w_H^2 \dots w_H^j w_H^{j'+1} w_H^{j'+2} \dots w_H^k w_H^{k'+1} \dots w_H^n$. Note that a is in this set.

We let A_1 be the set of letters from $A - A'$ having their two occurrences in $w_H^{j+1} w_H^{j+2} \dots w_H^{j'}$ and A_2 be the set of those having their two occurrences in $w_H^{k+1} w_H^{j+2} \dots w_H^{k'}$.

Every a in A' has one occurrence in $w_H^{j+1} w_H^{j+2} \dots w_H^{j'}$ and one in $w_H^{k+1} w_H^{j+2} \dots w_H^{k'}$.

The sets A_0, A_1, A_2 form a partition of $A - A'$ because no c in $A - A'$ crosses b or b' . For the same reason, no c in A_0 crosses any c' in $A_1 \cup A_2$. Finally if c in A_0 crosses some a' in A' , then it has one occurrence in $w_H^1 w_H^2 \dots w_H^j w_H^{k'+1} \dots w_H^n$ and the other in $w_H^{j+1} w_H^{j'+2} \dots w_H^k$, hence it crosses every a'' in A' . It follows that either A_0 is singleton or $\{A_0, A - A_0\}$ is a split of $G[A]$

It follows that $\{A_0 \cup B_1, (A - A_0) \cup B_2\}$ is a split which overlaps $\{A, B\}$. (Since A_0 and B_1 are not empty $A_0 \cup B_1$ has at least two elements). This contradicts the initial assumption. \square

Claim 2 : If a belongs to A' its two occurrences are in two different factors, say $a \in w_H^i \cap w_H^{i+p}$ for $1 \leq i < i+p \leq n$ without loss of generality.

Proof : Because if, on the contrary, a in A' has its two occurrences in some w_H^i , then no b in B' can cross it. \square

Claim 3 : If $a, a' \in A'$, $a \in w_H^i \cap w_H^{i+p}$ and $a' \in w_H^i \cap w_H^{i+q}$ then $p = q$.

Proof : Assume on the contrary that $p < q$. Some $b \in B'$ belongs to $w_K^{i+p} \dots w_K^{i+q-1}$ otherwise, G is not connected. If b crosses a , it cannot cross a' and vice-versa. \square

Claim 4 : If $a \in w_H^i \cap w_H^j$ and $a' \in w_H^{i'} \cap w_H^{j'}$, and i, j, i', j' are pairwise distinct, then a and a' cross.

Proof : Otherwise, as in the proof of Claim 3, assuming without loss of generality $1 \leq i < i' < j' < j \leq n$, any $b \in B'$ with an occurrence in $w_K^{i'} \dots w_K^{j'-1}$ cannot cross a and a' . \square

By Claims 2-4, we obtain that n is even and in Claim 2, $p = n/2$ for all i .

We let \overline{A}_i be the nonempty set of letters occurring in w_H^i , and $A'_i = \overline{A}_i \cap A'$. We define \overline{B}_i and B'_i similarly. Every letter of A'_i crosses every letter of A'_j , for $j \notin \{i, i+p\}$, and crosses also every letter of B' .

It follows that $\{\overline{A}_1 \cup \overline{A}_{p+1} \cup \overline{B}_1 \cup \overline{B}_{p+1}, (A - (\overline{A}_1 \cup \overline{A}_{p+1})) \cup (B - (\overline{B}_1 \cup \overline{B}_{p+1}))\}$ is a split of G that overlaps $\{A, B\}$ (because if $n \geq 3$, we have $p \geq 2$, hence the second set of this pair is not empty and has at least 2 elements) contradicting the initial assumption. This completes the proof. \square

It follows that if this decomposition corresponds to a good split each word representing $H \boxtimes_{(h,h)} K$ can be obtained by the operations of Lemma 8 from all those representing H and K .

First occurrence words.

For every word w in A^* we denote by $F(w)$ the subword of w consisting of the first occurrence of each letter. For an example, $F(abbdacdcefef) = abdcef$.

Theorem 10 : If w, w' are double occurrence words such that $G(w) = G(w')$ is connected and $F(w) = F(w')$, then $w = w'$.

The proof is given in Appendix 2. Hence for every circle graph G defined by a double occurrence word w , this word is completely determined by G and $F(w)$. We will prove that w can be determined from G and $F(w)$ by an MS transduction. (Note that $F(w)$ is a particular linear order on V_G). Our next objective is to prove the following result :

Theorem 11 : 1) There exists an MS transduction that associates with every connected circle graph G and every linear order on V_G , a double occurrence word on V_G representing G .

2) There exists an MS transduction that associates with (G, \preceq) where G is a connected circle graph and \preceq a linear order on V_G , the unique double occurrence word w representing it such that $F(w) = (V_G, \prec)$, provided such a word does exist.

Relational structures that represent double occurrence words over alphabets of unbounded size are defined in Section 3. We will combine the structures representing two composable words, whence, ultimately, the structures associated with the components of the split decomposition of a circle graph.

Composition of relational structures representing double occurrence words

Let S and T be disjoint relational structures representing composable words v and w with common letter a . We build as follows a structure representing a word in $v \boxtimes w$. We let $s \in D_S$, $t \in D_T$ correspond to an occurrence in each word of letter a . We define $L(S, T, s, t)$ as the structure $U = \langle D_U, suc_U, slet_U, mark_U \rangle$ such that :

$$D_U = D_S \cup D_T,$$

$$suc_U = (suc_S \cup suc_T) \circ link_{s,t}$$

where \circ denotes the composition of binary relations (i.e., for $A, B \subseteq D^2$, $A \circ B = \{(x, y) \in D^2 \mid (x, z) \in A, (z, y) \in B \text{ for some } z \in D\}$),

$$link_{s,t} = \{(s, t), (\bar{t}, s), (\bar{s}, \bar{t}), (t, \bar{s})\} \cup \{(x, x) \mid x \in D_S \cup D_T - \{s, t, \bar{s}, \bar{t}\}\},$$

and \bar{s} and \bar{t} are the unique elements such that $slet_S(s, \bar{s})$, and $slet_T(t, \bar{t})$.

Furthermore, we let $mark_U(x)$ hold if and only if $x \in \{s, t, \bar{s}, \bar{t}\}$ (the notation *mark* recalls that s, t, \bar{s}, \bar{t} correspond to the marker vertices in the graph composition $G(v) \boxtimes_{(a,a)} G(w)$). Notice that $L(S, T, s, \bar{t}) = L(T, S, t, s)$. The structure $L(S, T, s, t)$ defines a double occurrence word :

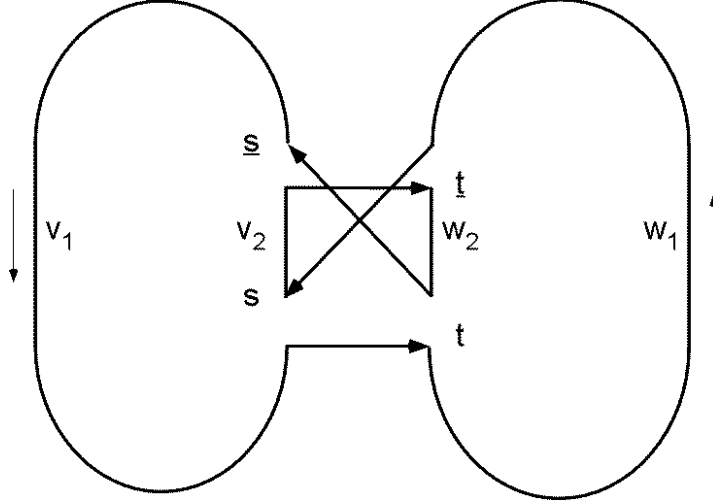


Figure 6. Linked structures ready for composition

$$u = av_1a'w_2av_2a'w_1,$$

associated with $v \equiv av_1av_2$, $w \equiv aw_1aw_2$ (note the use in u of a' in place of the letter a of w ; note also that the words v_1, v_2, w_1, w_2 are nonempty.) In order to obtain a double occurrence word in $v \boxtimes w$, it suffices to remove from u the letters a and a' . The elements of the domain of $L(S, T, s, t)$ corresponding to the occurrences of a and a' are those which satisfy the unary predicate $mark$.

Figure 6 shows the structure $L(S, T, s, t)$ representing the word $av_1a'w_2av_2a'w_1$ where S and T represent respectively the composable words av_1av_2 and aw_1aw_2 .

We let $Delete_{mark}$ be the transformation of structures such that $Z =$

$$\langle D_Z, suc_Z, slet_Z \rangle = Delete_{mark}(U) \text{ if :}$$

$$U = \langle D_U, suc_U, slet_U, mark_U \rangle$$

$$D_Z = D_U - mark_U,$$

$slet_Z$ is the restriction of $slet_U$ to D_Z ,

$suc_Z(x, y)$ holds if and only if y is the first iterated successor of x not in $mark_U$ (hence in D_Z).

The verification of the following lemma is straightforward.

Lemma 12 : If S and T represent the composable double occurrence words

v and w , if s and t correspond to the letter a common to v and w , then the structure $Delete_{mark}(L(S, T, s, t))$ represents a word in $v \boxtimes w$. The other words in $v \boxtimes w$ are obtained by the following structures :

$$Delete_{mark}(L(S, T, s, \bar{t})),$$

$$Delete_{mark}(L(S, T^{-1}, s, t)),$$

$$\text{and } Delete_{mark}(L(S, T^{-1}, s, \bar{t})).$$

Let us now assume that a circle graph G has a decomposition $\mathcal{D} = \{G_1, \dots, G_k\}$ and that for each component G_i , we have a structure S_i representing a double occurrence word w_i such that $G(w_i) = G_i$. Our objective is to build a structure $Link(\mathcal{D})$ from which one can obtain a double occurrence word for G . For $k = 2$, the structure $L(S_1, S_2, s, t)$ serves this purpose. We will actually generalize its construction by *linking* the structures S_1, \dots, S_k according to the neighbourhood relation of \mathcal{D} . Assuming the structures S_1, \dots, S_k pairwise disjoint, we let $S(\mathcal{D})$ be their union together with ε -edges : for each edge $e : G_i - G_j$, we choose s in S_i and t in S_j , such that the corresponding vertices in G_i and G_j are neighbour marker vertices in \mathcal{D} and we set an ε -edge $s - t$.

We make the tree $T(\mathcal{D})$ of components of \mathcal{D} into a rooted tree by choosing a root, say G_1 , and we orient its edges accordingly. Hence for $e : G_i \longrightarrow G_j$ in $T(\mathcal{D})$, the corresponding ε -edge is directed $s \longrightarrow t$.

We define $Link(\mathcal{D})$ as follows, like $L(S, T, s, t)$ is defined from $S \oplus T$:

$$(i) D_{Link(\mathcal{D})} = D_{S(\mathcal{D})},$$

(ii) $mark_{Link(\mathcal{D})}$ is the set of all s, \bar{s}, t, \bar{t} , for $s \in S_i, t \in S_j$ associated with an edge $e : G_i \longrightarrow G_j$ as described above,

(iii) $suc_{Link(\mathcal{D})} = suc_{S(\mathcal{D})} \circ link$, where $link$ is the relation :

$$\{(s, t), (\bar{t}, s), (\bar{s}, \bar{t}), (t, \bar{s}) \mid \text{for some edge } G_i \longrightarrow G_j, s \in S_i, t \in S_j$$

$$\text{and there is an } \varepsilon\text{-edge } s \longrightarrow t \}$$

$$\cup \{(x, x) \mid mark_{Link(\mathcal{D})}(x) \text{ does not hold}\}.$$

We also delete the ε -edges. They have been useful to specify the relation $link$, but are no longer. It is clear that the transformation of $S(\mathcal{D})$ into $Link(\mathcal{D})$ is an MS transduction. A root for the tree $T(\mathcal{D})$ can be chosen by means of a parameter, and from it, the directions of the edges of the tree can be defined by MS formulas.

Lemma 13 : If \mathcal{D} is a decomposition of a circle graph G with S_1, \dots, S_k as described above. Then $Delete_{mark}(Link(\mathcal{D}))$ represents a double occurrence word defining G .

Proof : The proof is by induction. The result is trivial if $k = 1$, i.e., if the decomposition has a single component.

Otherwise let us select a component, say G_k (without loss of generality) which is a leaf in the directed tree $T(\mathcal{D})$, having G_{k-1} (again without loss of generality) as father. Hence $\mathcal{D}' = \{G_1, \dots, G_{k-1}\}$ is the decomposition of a circle graph G' isomorphic to an induced subgraph of G . By using induction, we may assume that $Delete_{mark}(Link(\mathcal{D}'))$ is a double occurrence word S' for G' .

We have $Link(\mathcal{D}) = L(Link(\mathcal{D}'), S_k, s, t)$ where s, t correspond to the edge $G_{k-1} \rightarrow G_k$ in the tree $T(\mathcal{D})$.

$$\begin{aligned} \text{Hence } Delete_{mark}(Link(\mathcal{D})) &= Delete_{mark}(L(Link(\mathcal{D}'), S_k, s, t)) \\ &= Delete_{mark}(L(Delete_{mark}(Link(\mathcal{D}')), S_k, s, t)) \\ &= Delete_{mark}(L(S', S_k, s, t)) \end{aligned}$$

which is, using Lemma 8 a double occurrence word for G . \square

Proof of Theorem 11 :

1) We show that *some representation* can be constructed for a connected circle graph G given with a linear ordering \preceq of its vertices.

On a structure given with a linear order, the set predicate $Even(X)$ can be expressed by an MS formula (see [10]). Hence on these structures, every C_2 MS formula can be translated into an equivalent MS formula. In particular, an MS formula can check that the given graph is a circle graph.

By Proposition 3, one can construct from (G, \preceq) the graph $Sdg(Split(G))$ by an MS transduction which, by Lemma 16, can also build a linear order \preceq' on the vertices of $Sdg(Split(G))$. By Corollary 7 there exists a C_2 MS transduction that defines for each prime component of $Split(G)$ a double occurrence word representing it. However, since the components are linearly ordered by \preceq' , MS formulas are sufficient and this can be done by an MS transduction. For the other components, which are isomorphic to stars and to cliques, the linear order \preceq' makes possible to define an ordering as explained at the beginning of this section. (Let us recall from [9, 10] that in MS logic, one cannot specify a linear order on an arbitrary set. For example, one cannot define a linear order

on the leaves of stars of unbounded cardinality. It is thus important to have a linear order available).

Then one can obtain the structure $S(\mathcal{D})$ where the structures S_1, \dots, S_k corresponding to the k components of \mathcal{D} are linked by ε -edges.

The transformations $Link$ (applied to $S(\mathcal{D})$) and $Delete_{mark}$ are MS transductions. Hence, by using Proposition 18, one can combine these various MS transductions into a single one denoted by τ that associates with (G, \preceq) where G is a circle graph with vertices linearly ordered by \preceq , a structure $S(w)$ for a double occurrence word w such that $G(w) = G$.

2) We now modify this construction so as to obtain, if possible, a double occurrence word w such that $F(w)$ is the given linear order on V_G .

First observation :

As a consequence of Lemma 12, one can obtain from \mathcal{D} several structures $Link(\mathcal{D})$ giving different double occurrence words for the same circle graph.

For each edge $e : G_i \longrightarrow G_j$ of $T(\mathcal{D})$, there are four possibilities. The transduction τ defined in the first part of the proof can be equipped with parameters so as to output all possible results. We give some details. Let us assume a single structure $S(\mathcal{D})$ is fixed. Its transformation into $Link(\mathcal{D})$ can be parametrized by two sets X and Y :

(i) a subset X of $V_{S(\mathcal{D})}$ used as follows : for every ε -edge $S_i \longrightarrow S_j$, if X contains at least one element of S_j , then S_j is replaced by its reversal S_j^{-1} (the successor relation of S_j is reversed),

(ii) a subset Y of $V_{S(\mathcal{D})}$ used as follows : for every ε -edge $S_i \longrightarrow S_j$, if Y contains at least one element of S_j , then in the definition of $link$, t in S_j is replaced by \bar{t} and \bar{t} by t .

Recall that the ε -edges define a directed tree of components. Hence, for every ε -edge a single component S_j is used to indicate, via the sets X and Y , the transformations to be done to the links between the structures linked by this edge.

Second observation :

For each component of $Split(G)$ which is a clique or a star, the linear order given on V_G implies a unique representation that one can MS define.

Third observation. :

Assume a structure $S(w')$ has been constructed, from the graph G , its linear order \preceq and two sets X and Y . An MS formula can check that the successor function of \preceq coincides with the one of $S(w')$ for one of the two possible starting points, which are the two occurrences of the \preceq -smallest letter of w' (i.e., vertex of G). One can thus select, by an MS formula, the "good choices" of the sets X and Y . If no such sets do exist, then this means that the given linear ordering is not the first occurrence word of any representation of the given graph.

Hence to summarize, the construction is as follows : Given G and a linear order \preceq on V_G intended to represent $F(w)$ for some w to be constructed such that $G(w) = G$, one can first test by an MS formula whether G is a circle graph. (C₂MS is replaced by MS on ordered structures). If the answer is positive, one uses the linear order \preceq to build $Split(G)$ by an MS transduction. Then one uses \preceq again to fix the necessary orderings for the components that are cliques and stars. One uses \preceq also to define representations of the prime components by MS formulas as opposed to by C₂MS ones. Then, one "tries to find" the sets X and Y intended to "twist the links" and to "inverse" certain of these structures in order to find a double occurrence word w such that $F(w) = (V_G, \preceq)$. If they are found, then the MS transduction doing all this (poor fellow !) can produce the desired structure $S(w)$.

If there exists such a w , it is necessarily produced in this way because there is no choice for constructing the representations of the components, and all possible linkings are captured by the two sets X and Y . \square

Remarks 14 : (1) One might hope to be able to specify all double occurrence words representing G from a single linear order on V_G by varying some parameters. But a simple cardinality argument shows this is not possible : take for G the graph S_{n-1} with n vertices. It is represented by $(n-1)!/2 = O(2^{n \log(n)})$ pairwise inequivalent words. An MS transduction using k set parameters can only produce 2^{kn} different outputs.

(2) One can extend these results to nonconnected circle graphs by combining the results of the constructions performed on each connected component.

5 The clique-width of a circle graphs and the tree-width of its chord diagram.

The notions of *tree-width* and *clique-width* of graphs are well-known. The main facts are recalled in Appendix 1. Circle graphs have unbounded clique-width by the results of [25] since they include permutation graphs [24]. The structure $S(w)$ associated with a double occurrence word w is a graph with directed and undirected edges, called *the chord diagram* of w . It is a *chord diagram* of the circle graph $G(w)$ (and *the chord diagram* if G has a unique representation). Tree-width does not depend on edge directions, hence $S(w)$ has a tree-width $twd(S(w))$ (equal to that of the associated undirected graph $und(S(w))$). We will relate it with the clique-width $cwd(G(w))$ of $G(w)$.

Theorem 15 : A set of connected circle graphs has bounded clique-width if and only if the set of its chord diagrams has bounded tree-width. More precisely, there exist functions f and g such that for every double occurrence word w , $twd(S(w)) \leq f(cwd(G(w)))$ and $cwd(G(w)) \leq g(twd(S(w)))$.

Proof: In Appendix 1 we describe an MS transduction τ that transforms the structure $S(w)$ into $G(w)$ for every double occurrence word w . The graphs $S(w)$ have degree 3. The techniques of [12] show that there exists an MS transduction ω that inverses und , i.e., that associates with every undirected graph H of degree at most 3 the set of all graphs K with directed and undirected edges such that $und(K) = H$. It follows that for every double occurrence word w , $G(w) \in \tau(\omega(und(S(w))))$.

It follows from Propositions 19 and 20 that if the graph $und(S(w))$ has tree-width at most k , then it has clique-width at most $3 \cdot 2^{k-1}$ and $G(w)$ has clique-width bounded by $h_{\tau \circ \omega}(3 \cdot 2^{k-1})$. This gives the desired function g . (By Proposition 18, $\tau \circ \omega$ is an MS transduction).

Conversely, we wish to bound $twd(S(w))$ in terms of $cwd(G(w))$. By direct constructions, one can check that the chord diagrams of stars and cliques have tree-widths at most 3 and 4 respectively. By Corollary 7 there exists an order-invariant MS transduction that reconstructs $S(w)$ from $G(w)$ assumed to be prime. Hence, there exists by Proposition 19 a function f' such that $cwd(und(S(w))) \leq f'(cwd(G(w)))$ if $G(w)$ is prime. By the second assertion of Proposition 20, we obtain that $twd(S(w)) \leq f(cwd(G(w)))$ if $G(w)$ is prime, for some fixed function f .

If we could prove that for composable words v and w ,

$$twd(S(v \boxtimes w)) \leq \text{Max}\{twd(S(v)), twd(S(w))\} \quad (1)$$

the proof would be complete, because if a circle graph has clique-width at most k , then so have its prime factors in the split decomposition, and all its chord diagrams are built from those corresponding to stars, cliques and its prime factors. However, (1) does not hold. The chord diagrams of $K_{2,3}$ have tree-width 4 whereas this graph splits into S_2 and S_3 , with associated chord diagrams of tree-width 3. Hence, we must use an additional construction.

Let w be a double occurrence word of length $2n$; we modify its representing structure $S(w)$ and make it into a graph $\widehat{S}(w)$ called an *extended chord diagram* as follows :

- 1) We replace each directed edge $i \longrightarrow j$ by two edges $i - i^+$ and $j^- - j$ where i^+ is a new vertex also denoted by j^- , and we add the edges $i^- - i^+$ for all i .
- 2) The vertex set of $\widehat{S}(w)$ is the set $\widehat{V}(w) = \{i, i^+ \mid i = 1, \dots, 2n\}$, it has $4n$ elements.
- 3) For each undirected edge $i - j$ of $S(w)$ (corresponding to a pair of occurrences of a letter a in w), we add the edges $i^+ - j^+$, $i^- - j^-$, $i^+ - j^-$ and $i^- - j^+$. We fuse parallel edges, so that $\widehat{S}(w)$ is a simple graph, of degree at most 7.

We denote by $M(w, a)$ the set of vertices $\{i^+, j^+, i^-, j^-\}$ for i, j as in 3). By the assumption that $G(w)$ is connected, we never have $j^+ = i^-$, hence the set $M(w, a)$ has 4 elements. It induces a clique.

It is clear that $und(S(w))$ is a minor of $\widehat{S}(w)$, hence it is enough to bound $twd(\widehat{S}(w))$ in terms of $cwd(G(w))$ to get the result, because $twd(S(w)) \leq twd(\widehat{S}(w))$.

The transformation of $S(w)$ into $\widehat{S}(w)$ is an MS transduction. Hence, since the composition of two MS transductions is an MS transduction (Proposition 18), we have $twd(\widehat{S}(w)) \leq \widehat{f}(cwd(G(w)))$ if $G(w)$ is prime for some fixed function \widehat{f} . For the extended chord diagrams of S_n and K_n one can construct tree-decompositions of width at most 5. The proof will be complete with the following claim.

Claim : For composable words v and w

$$twd(\widehat{S}(v \boxtimes w)) \leq \text{Max}\{twd(\widehat{S}(v)), twd(\widehat{S}(w))\}.$$

Proof : Let $v = v_1av_2av_3$ and $w = aw_1aw_2$ be composable words with common letter a . Let $y = v_1w_1v_2w_2v_3$ be one of their compositions.

We make disjoint the graphs $\widehat{S}(v)$ and $\widehat{S}(w)$. We let i and j be the first and second occurrence of a in v , and k and l be the first and second occurrence of a in w . We denote by $\widehat{S}(v) - i - j$ the graph obtained from $\widehat{S}(v)$ by deleting the vertices i, j and their incident edges. We let $\widehat{S}(w) - k - l$ be defined similarly from by $\widehat{S}(w)$.

The extended chord diagram $\widehat{S}(v \boxtimes w)$ can be constructed as follows :

- 1) one takes the (disjoint) union of $\widehat{S}(v) - i - j$ and $\widehat{S}(w) - k - l$,
- 2) one "glues them" at $M(v, a)$ and $M(w, a)$ by fusing i^- and k^+ , i^+ and l^- , j^- and l^+ , and j^+ and k^- ,
- 3) one deletes some edges between these vertices.

For the three other compositions of v and w , one glues the graphs with fusions based on different matchings, for an example, one fuses i^- and l^- , i^+ and k^+ , j^+ and l^+ , and j^- and k^- , in order to get the graph $\widehat{S}(v_1 \widetilde{w}_1 v_2 \widetilde{w}_2 v_3)$.

Now since $M(v, a)$ induces a clique in $\widehat{S}(v)$ this graph has a tree-decomposition one box of which consists exactly of $M(v, a)$. The same holds for $\widehat{S}(v) - i - j$. Similarly, $\widehat{S}(w) - k - l$ has a tree-decomposition one box of which consists exactly of $M(w, a)$. By combining these tree-decompositions, one obtains one of $\widehat{S}(v \boxtimes w)$ of width :

$$\text{Max}\{twd(\widehat{S}(v) - i - j), twd(\widehat{S}(w) - k - l)\} = \text{Max}\{twd(\widehat{S}(v)), twd(\widehat{S}(w))\}.$$

This completes the proofs of the claim and of the theorem. \square

Theorem 15 remains valid if instead of $S(w)$ we consider the neighborhood graph $N(S(w))$, because $N(S(w))$ is obtained from $und(S(w))$ by edge contractions, so that $twd(N(S(w))) \leq twd(und(S(w)))$, and $twd(und(S(w))) \leq 2.twd(N(S(w))) + 1$, as one checks easily.

Question : Can one relate precisely the clique-width or the rank-width of a prime circle graph and the tree-width of its chord diagram?

6 Conclusion

As said in the introduction, the present article is part of a global project consisting in trying to formalize as much as possible graph properties and graph

constructions like hierchical decompositions in monadic second-order logic and some of its extensions which enjoy the same good algorithmic properties. We have studied circle graphs by using the split decomposition of [18, 14] and other logical tools from [17]. We have shown how to define by MS formulas all chord diagrams representing a given circle graph.

Two ideas have other application instances : the reconstruction by an MS transduction of some "forgotten information" and the use of a canonical decomposition for this reconstruction.

The "forgotten information" may be simply the directions of edges: an MS transduction using edge set quantifications can define for every undirected graph all its possible orientations [8]. In this simple case, no sophisticated decomposition is needed, just a depth-first spanning forest.

More involved are the following descriptions. All planar embeddings of a planar graph can be defined from its canonical decomposition in 3-connected components [11]; all graphs having the same cycle matroid as a given graph G can be obtained by Whitney's 2-isomorphism theorem from the decomposition of G in 3-connected components ([14], see also [37]) ; all transitive orientations of a comparability graph can be determined from its modular decomposition ([9, 13], see also [27,32]). All these characterizations use canonical decompositions of the considered graphs that can be constructed by MS transductions (in some cases with the help of an auxiliary linear order), and, a unicity property of the objects to be constructed for the components of these decompositions. We think that this approach can be applied to other types of geometric graph representations like those by intervals, by circular arcs or by intersecting rectangles.

Acknowledgement : I thank J. Makowsky for his numerous helpful comments and suggestions of references.

7 References

- [1] R. Arratia, B. Bollobás, D. Coppersmith, G. B. Sorkin, Euler circuits and DNA sequencing by hybridization. *Discrete Applied Mathematics*, **104** (2000) 63-96.

- [2] R. Arratia, B. Bollobás, G. B. Sorkin, A Two-Variable Interlace Polynomial. *Combinatorica*, **24** (2004) 567-584.

- [3] M. Avriel, M. Penn, N. Shpirer, Container ship stowage problem: complexity and connection to the coloring of circle graphs. *Discrete Applied Mathematics*, **103** (2000) 271-279.
- [4] A. Bouchet, Reducing prime graphs and recognizing circle graphs, *Combinatorica*, **7** (1987) 243-254.
- [5] A. Bouchet, Circle graph obstructions, *J. Combinatorial Theory B*, **60** (1994) 107-144.
- [6] D. Corneil, U. Rotics, On the relationship between clique-width and tree-width, *SIAM J. Computing*, **34** (2005) 825-847.
- [7] B. Courcelle, Monadic second-order graph transductions: A survey. *Theoretical Computer Science*, **126** (1994) 53–75.
- [8] B. Courcelle, The Monadic Second-Order Logic of Graphs VIII: Orientations. *Ann. Pure Appl. Logic* **72**(1995) 103-143.
- [9] B. Courcelle, The monadic second-order logic of graphs X : Linear orderings, *Theoret. Comput. Sci.* **160** (1996) 87-143.
- [10] B. Courcelle, The expression of graph properties and graph transformations in monadic second-order logic. In G. Rozenberg, ed., *Handbook of graph grammars and computing by graph transformations, Vol. 1: Foundations*, pages 313–400, World Scientific, 1997.
- [11] B. Courcelle, The monadic second-order logic of graphs XII : Planar graphs and planar maps, *Theoret. Comput. Sci.*, **237** (2000) 1-32.
- [12] B. Courcelle, The monadic second-order logic of graphs XIV: Uniformly sparse graphs and edge set quantifications. *Theor. Comput. Sci.*, **299** (2003) 1-36.
- [13] B. Courcelle, The monadic second-order logic of graphs XV : On a Conjecture by D. Seese, *Journal of Applied Logic*, **1** (2006) 79-114.

- [14] B. Courcelle, The monadic second-order logic of graphs XVI : Canonical graph decompositions, *Logical Methods in Computer Science*, **2** (2006) 1-46.
- [15] B. Courcelle, J. Makowsky, U. Rotics, On the fixed parameter complexity of graph enumeration problems definable in monadic second-order logic, *Discrete Applied Maths*, **108** (2001) 23-52.
- [16] B. Courcelle, S. Olariu, Upper bounds to the clique width of graphs. *Discrete Applied Maths*, **101** (2000) 77-114.
- [17] B. Courcelle, S. Oum, Vertex-minors, monadic second-order logic and a conjecture by Seese, *J. Combinatorial Theory B*, **97** (2007) 91-126.
- [18] W. Cunningham, Decomposition of directed graphs, *SIAM J. Algebraic Discrete Methods* **3** (1982) 214-228.
- [19] A. Dawar, Generalized quantifiers and logical reducibilities. *J. Logic and Computation*, **5** (1995) 213-226.
- [20] H.D. Ebbinghaus, J. Flum, *Finite Model Theory*, Springer, 1999.
- [21] S. Even, A. Itai, Queues, stacks and graphs, in *Theory of machines and computations*, A. Kohavi, A. Paz eds., Academic Press, New-York, 1971, pp. 71-86.
- [22] C. Gabor, W. Hsu, K. Supowit, Recognizing circle graphs in polynomial time, *J. of ACM* **36** (1989) 435-473.
- [23] F. Genest, Circle graphs and the cycle double cover conjecture, *Discrete Mathematics*, to appear.
- [24] M. Golumbic, *Algorithmic graph theory and perfect graphs*, Academic Press, 1980, (Second edition, Annals of Discrete Mathematics 57, 2004, Elsevier).
- [25] M. Golumbic, U. Rotics, On the clique-width of some perfect graph classes. *Int. J. Found. Comput. Sci.*, **11** (2000) 423-443.

- [26] F. Gurski, E. Wanke, The tree-width of clique-width bounded graphs without $K_{n,n}$, in *Proceedings WG 2000, Lec. Notes Comput. Sci.* **1928** (2000) 196-205.
- [27] D. Kelly, Comparability graphs, in *Graphs and order*, I. Rival ed., D. Reidel Pub. Co., 1985, pp. 3-40.
- [28] V. P. Kozyrev, S. V. Yushmanov, Representations of graphs and networks (coding, layouts and embeddings), *Journal of Mathematical Sciences*, (Springer) **61** (1992) 2152-2194.
- [29] L. Libkin, *Elements of finite model theory*, Springer, 2004.
- [30] J. Makowsky, Algorithmic uses of the Feferman–Vaught Theorem, *Annals of Pure and Applied Logic*, **126** (2004) 159–213
- [31] J. Makowsky, Y. B. Pnueli, Logics Capturing Relativized Complexity Classes Uniformly, *Logical and Computational Complexity. Selected Papers*, Lecture Notes in Computer Science **960** (1995) 463-479.
- [32] R. Möhring, F. Radermacher, Substitution decomposition for discrete structures and connections with combinatorial optimization, *Annals of Discrete Maths*, **19** (1984) 257-356.
- [33] S. Oum, Rank-width and vertex minors, *J. Combinatorial Theory B*, **95** (2005) 79-100.
- [34] S. Oum, Excluding a bipartite circle graph from line graphs, Submitted, 2006.
- [35] J. Spinrad, Recognition of Circle Graphs. *J. Algorithms* **16** (1994) 264-282.
- [36] J. Spinrad, *Efficient graph representations*, AMS, Field Institute Monographs, vol. 19, 2003.
- [37] N. White, *Theory of matroids*, Cambridge University Press, 1986.

8 Appendix 1 : Monadic second-order logic

We review Monadic Second-Order (MS) logic and transformations of structures expressed in this language, called *MS transductions*. The reader is referred to the book chapter [10], or to the preliminary sections the articles [7, 9, 11, 13, 14] for more detailed expositions. However all necessary definitions are given in the present section.

Relational structures and monadic second-order logic

Let $R = \{A, B, C, \dots\}$ be a finite set of *relation symbols* each of them given with a nonnegative integer $\rho(A)$ called its *arity*. We denote by $\mathcal{STR}(R)$ the set of *finite R-structures* $S = \langle D_S, (A_S)_{A \in R} \rangle$ where $A_S \subseteq D_S^{\rho(A)}$ if $A \in R$. If R consist of relation symbols of arity one or two, we say that the structures in $\mathcal{STR}(R)$ are *binary*.

A simple graph G can be defined as the $\{edg\}$ -structure $G = \langle V_G, edg_G \rangle$ where V_G is the vertex set and $edg_G \subseteq V_G \times V_G$ is a binary relation representing the edges. For undirected graphs, the relation edg_G is symmetric. If in addition we need vertex labels, we will represent them by unary relations. Binary structures can be seen as vertex- and edge- labelled graphs. If we have several binary relations say A, B, C , the corresponding graphs have edges of types A, B, C .

Monadic second-order logic (MS logic for short) is the extension of *First-order logic* (FO logic) by variables denoting subsets of the domains of the considered structures, and new atomic formulas of the form $x \in X$ expressing the membership of x in a set X . (Uppercase letters denote set variables, lowercase letters denote ordinary first-order variables). We denote by $MS(R, W)$ the set of *monadic second-order* formulas written with the set R of relation symbols and having their free variables in a set W consisting of *individual and set variables*. As a typical and useful example of MS formula, we give a formula with free variables x and y expressing that (x, y) belongs to the reflexive and transitive closure of a binary relation A :

$$\forall X(x \in X \wedge \forall u, v[(u \in X \wedge A(u, v)) \implies v \in X] \implies y \in X)$$

If the relation A is not given in the structure but defined by an MS formula $\alpha(u, v)$, then one replaces $A(u, v)$ by this formula with appropriate substitutions of variables. We denote by $TC[\alpha(u, v); x, y]$ the resulting formula.

Order-invariant MS properties

A property P of the structures S of a class $\mathcal{C} \subseteq \mathit{STR}(R)$ is *monadic second-order (MS)* if for some fixed formula φ in $MS(R, \emptyset)$, $P(S)$ holds if and only if $S \models \varphi$. We now extend this definition by allowing the use of auxiliary linear orders. Let \leq be a binary relation symbol not in R . A formula φ in $MS(R \cup \{\leq\}, \emptyset)$ is *order-invariant on a class $\mathcal{C} \subseteq \mathit{STR}(R)$* , if for every $S \in \mathcal{C}$, for every two linear orders \preceq and \preceq' on the domain D_S , we have $(S, \preceq) \models \varphi$ if and only if $(S, \preceq') \models \varphi$, where \preceq and \preceq' interpret \leq . We say that P is an *order-invariant MS property* of the structures of \mathcal{C} if $P(S)$ holds if and only if $(S, \preceq) \models \varphi$ for some linear order \preceq on D_S , where φ is an MS formula that is order-invariant on \mathcal{C} .

The property that a set has even cardinality is order-invariant on the class of all (finite) \emptyset -structures. Hence, every \mathbf{C}_2 MS property is an order invariant MS property. It is usually not decidable whether an MS formula is order-invariant on a class \mathcal{C} . However, we use formulas that are order-invariant by construction.

Monadic second-order transductions

We use MS formulas to define transformations of graphs and relational structures. As in language theory, a binary relation $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{B}$ where \mathcal{A} and \mathcal{B} are sets of words, graphs or relational structures is called a *transduction: $\mathcal{A} \rightarrow \mathcal{B}$* . An *MS transduction* is a transduction specified by MS formulas. It transforms a structure S , given with an n -tuple of subsets of its domain called the *parameters*, into a structure T , the domain of which is a subset of $D_S \times \{1, \dots, k\}$. Furthermore, each such transduction, has an associated *backwards translation*, a mapping that transforms effectively every MS formula φ relative to T , possibly with free variables, into one, say $\varphi^\#$, relative to S having free variables corresponding to those of φ (k times as many actually) together with those denoting the parameters. This new formula expresses in S the property of T defined by φ . We now give some details. More can be found in [7, 10].

We let R and Q be two finite sets of relation symbols. Let W be a finite set of set variables, called *parameters*. A (Q, R) -*definition scheme* is a tuple of formulas of the form :

$$\Delta = (\varphi, \psi_1, \dots, \psi_k, (\theta_w)_{w \in Q * k})$$

where $k > 0$, $Q * k := \{(q, \vec{j}) \mid q \in Q, \vec{j} \in [k]^{\rho(q)}\}$,

$\varphi \in MS(R, W)$, $\psi_i \in MS(R, W \cup \{x_1\})$ for $i = 1, \dots, k$,

and $\theta_w \in MS(R, W \cup \{x_1, \dots, x_{\rho(q)}\})$, for $w = (q, \vec{j}) \in Q * k$.

These formulas are intended to define a structure T in $\mathcal{STR}(Q)$ from a structure S in $\mathcal{STR}(R)$. Let $S \in \mathcal{STR}(R)$, let γ be a W -assignment in S . A Q -structure T with domain $D_T \subseteq D_S \times [k]$ is defined in (S, γ) by Δ if :

- (i) $(S, \gamma) \models \varphi$
- (ii) $D_T = \{(d, i) \mid d \in D_S, i \in [k], (S, \gamma, d) \models \psi_i\}$
- (iii) for each q in Q : $q_T = \{((d_1, i_1), \dots, (d_t, i_t)) \in D_T^t \mid (S, \gamma, d_1, \dots, d_t) \models \theta_{(q, \vec{j})}\}$, where $\vec{j} = (i_1, \dots, i_t)$ and $t = \rho(q)$.

Since T is associated in a unique way with S, γ and Δ whenever it is defined, i.e., whenever $(S, \gamma) \models \varphi$, we can use the functional notation $def_\Delta(S, \gamma)$ for T . The *transduction defined by Δ* is the binary relation :

$$def_\Delta := \{(S, T) \mid T = def_\Delta(S, \gamma) \text{ for some } W\text{-assignment } \gamma \text{ in } S\}.$$

Hence $def_\Delta \subseteq \mathcal{STR}(R) \times \mathcal{STR}(Q)$. A transduction $f \subseteq \mathcal{STR}(R) \times \mathcal{STR}(Q)$ is an *MS transduction* if it is equal, up to isomorphism, to def_Δ for some (Q, R) -definition scheme Δ . We will also write functionally : $def_\Delta(S) := \{def_\Delta(S, \gamma) \mid \gamma \text{ is a } W\text{-assignment in } S\}$.

An MS-transduction can be seen as a "nondeterministic" partial function associating with an R -structure one or more Q -structures. However, it is not really nondeterministic because the different outputs come from different choices of parameters. We will refer to the integer k by saying that Δ and def_Δ are *k-copying* ; if $k = 1$ we will say that they are *noncopying*. A noncopying definition scheme can be written more simply : $\Delta = (\varphi, \psi, (\theta_q)_{q \in Q})$. A definition scheme without parameters defines a *parameterless MS transduction*, which is actually a partial function : $\mathcal{STR}(R) \longrightarrow \mathcal{STR}(Q)$.

The MS transduction transforming $S(w)$ into $G(w)$.

A double occurrence word $w = a_1 a_2 \dots a_{2n}$ is represented by the relational structure $S(w) = \langle \{1, \dots, 2n\}, suc, slet \rangle$ where $suc(i, j)$ holds if and only if $j = i + 1$, with also $suc(2n, 1)$, and $slet(i, j)$ holds if and only if $i \neq j$ and $a_i = a_j$.

Let $S = \langle D, suc, slet \rangle$ be given, assumed to be isomorphic to $S(w)$ for some w . We must select from D a subset that will be the vertex set of the circle graph to be constructed.

- 1) We use for this a parameter X subject to the following condition :

$$\forall x, y [x \in X \wedge y \in X \implies \neg slet(x, y)] \wedge \forall x [x \in X \vee \exists y (y \in X \wedge slet(x, y))].$$

This formula will be ψ of the definition scheme : $\Delta = (\varphi, \psi, \theta_{edg})$ to be constructed. It expresses that X contains one and only one element of each pair defined by $slet$.

2) We need an auxiliary formula $\beta(x, y, z)$ expressing the following :

x, y, z are pairwise distinct and when one follows the Hamiltonian circuit of S , then, after x one sees y before z . We let α be the formula :

$$suc(u, v) \wedge u \neq z \wedge v \neq z.$$

Then $\beta(x, y, z)$ is the formula $x \neq z \wedge x \neq y \wedge y \neq z \wedge TC[\alpha(u, v); x, y]$.

3) Using $\beta(x, y, z)$ we can write $\iota(u, v, w, z)$ expressing that two pairs $\{u, v\}$ and $\{w, z\}$ are "interlaced" (cf. [2]) :

$$(\beta(u, w, v) \wedge \beta(v, z, u)) \vee (\beta(u, z, v) \wedge \beta(v, w, u)).$$

4) We can deduce a definition of the edge relation of G ; $\theta_{edg}(x, y)$ is the formula :

$$x \in X \wedge y \in X \wedge \exists u, v [slet(x, u) \wedge slet(y, v) \wedge \iota(x, u, y, v)].$$

In order to complete the definition of $\Delta = (\varphi, \psi, \theta_{edg})$, it remains to construct a formula φ expressing that the given structure S is isomorphic to $S(w)$ for some w . This is actually a routine construction, using the fact that transitive closures are expressible in MS logic. \square

Lemma 16 : Let $\tau : STR(R) \longrightarrow STR(Q)$ be an MS transduction. Let us add to R and to Q a binary relation symbol \leq intended to represent orders on the domains of structures. One can transform τ into an MS transduction $\tau' : STR(R \cup \{\leq\}) \longrightarrow STR(Q \cup \{\leq\})$ such that, for every S in $STR(R)$ and every linear order \preceq on its domain, $\tau'(S, \preceq) = (\tau(S), \preceq')$ where \preceq' is a linear order on the domain of $\tau(S)$.

Proof : Let τ be k -copying. It is easy to define formulas θ_w belonging to $MS(R \cup \{\leq\}, W \cup \{x_1, x_2\})$, for $w = (\leq, \vec{j}) \in \{\leq\} * k$ such that, in $\tau'(S, \preceq)$:

$$(d_1, i) \preceq' (d_2, j) \text{ if and only if either } d_1 \prec d_2 \text{ or } (d_1 = d_2 \text{ and } i \leq j).$$

It is clear that \preceq' is a linear order on the domain of $\tau(S)$ if \preceq is one on S . \square

A partial function $\tau: \mathcal{STR}(R) \longrightarrow \mathcal{STR}(Q)$ is an *order-invariant* MS transduction if there exists a $(Q, R \cup \{\leq\})$ -definition scheme Δ (possibly with parameters) such that :

- 1) the property $def_{\Delta}(S) \neq \emptyset$ is an order-invariant MS property of structures S in $\mathcal{STR}(R)$, equivalent to the fact that $\tau(S)$ is defined,
- 2) for any two linear orders \preceq and \preceq' on D_S , any two structures in $def_{\Delta}(S, \preceq)$ and in $def_{\Delta}(S, \preceq')$ are isomorphic to $\tau(S)$.

It follows that an output structure in $def_{\Delta}(S, \preceq, \gamma)$ depends, up to isomorphism, neither on the linear order \preceq nor on the assignment γ of values to parameters.

The transduction of Corollary 7 is order-invariant whereas that of Theorem 11 is not.

The fundamental property of MS transductions

The following proposition says that if $T = def_{\Delta}(S, \gamma)$, then the monadic second-order properties of T can be expressed as monadic second-order properties of (S, γ) . The usefulness of definable transductions is based on this proposition.

Let $\Delta = (\varphi, \psi_1, \dots, \psi_k, (\theta_w)_{w \in Q^{*k}})$ be a (Q, R) -definition scheme, written with a set of parameters W . Let V be a set of set variables disjoint from W . For every variable X in V , for every $i = 1, \dots, k$, we let X_i be a new variable. We let $V' = \{X_i \mid X \in V, i = 1, \dots, k\}$. Let S be a structure in $\mathcal{STR}(R)$ with domain D . For every mapping $\eta: V' \longrightarrow \mathcal{P}(D)$, we let $\eta^k: V \longrightarrow \mathcal{P}(D \times [k])$ be defined by $\eta^k(X) = \eta(X_1) \times \{1\} \cup \dots \cup \eta(X_k) \times \{k\}$. With this notation we can state :

Proposition 17: For every formula β in $MS(R, V)$ one can construct a formula $\beta^{\#}$ in $MS(Q, V' \cup W)$ such that, for every S in $\mathcal{STR}(R)$, for every assignment $\gamma: W \longrightarrow S$ for every assignment $\eta: V' \longrightarrow S$ we have:

$$(S, \eta \cup \gamma) \models \beta^{\#} \text{ if and only if :}$$

$$def_{\Delta}(S, \gamma) \text{ is defined, } \eta^k \text{ is a } V\text{-assignment in } def_{\Delta}(S, \gamma),$$

$$\text{and } (def_{\Delta}(S, \gamma), \eta^k) \models \beta.$$

If the definition scheme and the formula β are FO, then the formula $\beta^\#$ is also FO. Note that, even if $T = \text{def}_\Delta(S, \gamma)$ is well-defined, the mapping η^k is not necessarily a V -assignment in T , because $\eta^k(X)$ may not be a subset of the domain of T which is a possibly proper subset of $D_S \times \{1, \dots, k\}$. We call $\beta^\#$ the *backwards translation* of β relative to the transduction def_Δ .

The composition of two transductions is defined as their composition as binary relations. If they are both partial functions, then one obtains the composition of these functions.

Proposition 18 : 1) The composition of two MS transductions is an MS transduction.

2) The inverse image of an MS-definable class of structures under an MS transduction is MS-definable.

Tree-width, clique-width and MS transductions

The notions of *tree-width* and *clique-width* are well-known. Definitions and basic results can be found in [10, 15, 16, 17, 30]. We only review some facts used in Section 5.

Proposition 19 : For every order-invariant MS transduction τ from undirected graphs to directed or undirected graphs, there exists a function h_τ such that, for every simple undirected graph H , we have $\text{cwd}(\tau(H)) \leq h_\tau(\text{cwd}(H))$.

Proof : We will use the result that a set of simple undirected graphs has bounded clique-width if and only if it is the image of a set of binary trees under an MS transduction (by Theorem 5.6.8 of [10]).

The set $C(k)$ of graphs of clique-width at most k is the image of a set of binary trees under an MS transduction γ_k . A linear order on binary trees is MS definable ([9]), hence (with Proposition 18), γ_k can be modified into γ'_k producing from binary trees the graphs of $C(k)$, each with a linear order. (We do not claim that all linear orders on all graphs of $C(k)$ can be produced.) It follows that the graphs $\tau(H)$ for H in $C(k)$ are images of binary trees under $\tau \circ \gamma'_k$. Hence, they have a clique-width bounded in term of k . \square

Proposition 20 : For a simple undirected graph H we have $\text{cwd}(H) \leq 3 \cdot 2^{\text{twd}(H)-1}$ and $\text{twd}(H) \leq (3 \cdot \text{deg}(H) + 2) \cdot \text{cwd}(H) - 1$.

Proof : The first result is by Corneil and Rotics [6]. The second one follows from the result of Gurski and Wanke [26] saying that $twd(H) \leq (3p - 1) \cdot cwd(H) - 1$ if H has no subgraph isomorphic to $K_{p,p}$. \square

9 Appendix 2: Two technical proofs

We first prove the "only if" direction of Proposition 4 which is asserted but not proved in [22].

Proposition 4 ("Only if"): A connected uniquely representable circle graph with at least 5 vertices is prime.

We first explain why a proof is needed. Assume $G = H \boxtimes_{(h,h)} K = G(w_1 v_1 w_2 v_2)$, where $H = G(w_1 h w_2 h)$, $K = G(h v_1 h v_2)$. Clearly $G = G(w_1 \widetilde{v}_1 w_2 \widetilde{v}_2)$. But it may happen that $w_1 v_1 w_2 v_2 \equiv w_1 \widetilde{v}_1 w_2 \widetilde{v}_2$, and G may be nevertheless uniquely representable. This is the case of the graph $G = G(abcdbacd)$ defined by the two words $w_1 v_1 w_2 v_2 = abcdbacd$ and $w_1 \widetilde{v}_1 w_2 \widetilde{v}_2 = abdcbadc$. That this situation does not happen if G has at least 5 vertices deserves a proof.

The "proof" sketched in [22] is not correct for the following reasons. First it does not use the hypothesis that the considered graph has at least 5 vertices. However, this hypothesis is necessary as observed above. Second the argument goes as follows : if $G = H \boxtimes_{(h,h)} K = G(w_1 v_1 w_2 v_2)$, where $H = G(w_1 h w_2 h)$, $K = G(h v_1 h v_2)$ then $G = G(w_1 \widetilde{v}_1 w_2 \widetilde{v}_2)$ and it is left to the reader to check that $w_1 v_1 w_2 v_2$ and $w_1 \widetilde{v}_1 w_2 \widetilde{v}_2$ are not equivalent. However, even with the hypothesis that G has at least 5 vertices, this may be false. Consider for an example the graph with 6 vertices defined by the word :

$$abcdceae fdf = w_1 v_1 w_2 v_2 \text{ with } w_1 = bab, v_1 = cdc, w_2 = eae, v_2 = fdf.$$

The two words $w_1 v_1 w_2 v_2$, $w_1 \widetilde{v}_1 w_2 \widetilde{v}_2$ are equivalent, they are even equivalent to the two other words $w_1 v_2 w_2 v_1$, $w_1 \widetilde{v}_2 w_2 \widetilde{v}_1$ resulting from the composition of $w_1 h w_2 h$ and $h v_1 h v_2$ (cf. Lemma 8). The graph G is not UR (uniquely representable in short), because it is also defined by the word $beae bcdca fdf$ not equivalent to the initial one, but the argument sketched in [22] does not prove that.

Our proof will use the notion of *local complementation*, that we recall from [4, 17, 23, 33]. If $w = xayaz$ is a double occurrence word where a is a letter

and x, y, z are words, we denote by $w * a$ the word $xa\tilde{y}az$. The corresponding graph $G(w * a)$ is obtained from $G(w)$ by edge-complementing the subgraph of $G(w)$ induced by the vertices which are adjacent to a . It is denoted by $G(w) * a$. This is well-defined because if w' is equivalent to w , then $G(w' * a) = G(w * a)$. By Lemmas 2.1 and 2.2 of [4], $G(w * a)$ and $G(w)$ have the same splits and one is prime if and only if the other is. It is easy to prove that if $w' \equiv w$ then $w' * a \equiv w * a$. It follows that $G(w) * a$ is UR if and only if $G(w)$ is UR.

Proof : A star and a clique with at least 5 vertices are not UR. Just consider the inequivalent words $abcdewabcdew$ and $acbdewacbdew$ for a clique (where a, b, c, d, e are letters and w is any word) and similarly $abcdewa\tilde{w}edcb$ and $acbdewa\tilde{w}edbc$ for a star. (Since the local complementation of a clique is a star, the case of stars could be derived from that of cliques and the remarks that local complementation preserve the properties of being prime and UR.). Assume by way of contradiction that a circle graph G is connected, UR, has at least 5 vertices and is not prime. By the initial observation and the results of [18], it has a good split $\{A, B\}$ from which we get $G = H \boxtimes K$. We first consider two special cases.

First special case : $A = \{a, b\}$ and only a is linked to K .

Then G is represented by a word $w = babuav$. The hypothesis that G is UR implies that

$babuav \equiv bab\tilde{u}a\tilde{v}$. This gives two possibilities : $u = \tilde{u}$ and $v = \tilde{v}$ or $u = v$.

Subcase 1 : $u = \tilde{u}$ and $v = \tilde{v}$. Then $u = yc\tilde{y}$, $v = zc\tilde{z}$ for some letter c and some words y, z having no letter in common. If y and z are both nonempty, the word $babzyc\tilde{z}\tilde{y}ac$ represents G and is not equivalent to $babuav = babyc\tilde{y}azc\tilde{z}$. If $u = dxc\tilde{x}d$, $v = c$, then we use $babxc\tilde{x}adcd$ to obtain a contradiction.

Subcase 2 : $u = v$. This word has at least 3 letters, say c, d, \dots, f , hence $\{\{a, b, c\}, \{d, \dots, f\}\}$ is a split which overlaps $\{A, B\}$ hence $\{A, B\}$ is not good. Contradiction.

Second special case : $A = \{a, b\}$ and a, b are both linked to K (they cross B).

Then G is represented by a word $w = abuabv$ or $w = abubav$.

Subcase 1 : $w = abuabv$.

By using local complementation with respect to a we obtain the word $w * a = a\tilde{u}babv$. The graph $G(w * a)$ is also UR and satisfies the first special case. This is impossible.

Subcase 2 : $w = abubav$.

By using local complementation with respect to a vertex c in B adjacent to a and b , we obtain a graph $G(w * c)$ that is also UR and satisfies Subcase 1 just above. So this is impossible.

General case : We consider G with a good split $\{A, B\}$ from which we get $G = H \boxtimes K$.

Each of A and B has at least 3 letters, otherwise we can conclude using the two special cases. By Proposition 9, G is represented by a word $w_A^1 w_B^1 w_A^2 w_B^2$ with $w_A^1, w_A^2 \in A^*$ and $w_B^1, w_B^2 \in B^*$.

Subcase 1 : There is a unique letter a in A that crosses B . (We mean by "cross" that, in any chord representation, the chord a intersects some chord in B). Let b another letter in A . Let $G' = G[B \cup \{a, b\}]$. Without loss of generality, b is in w_A^1 . The graph G' is connected and has a representation by $babw_B^1 aw_B^2$.

Claim 1 : The graph G' is UR : Assume it is not. It has another representation by a word $babv_B^1 av_B^2$ which is inequivalent to the first.

Consider now the word $w_A^1 v_B^1 w_A^2 v_B^2$. It defines G . Hence it is equivalent to $w_A^1 w_B^1 w_A^2 w_B^2$. By deleting the letters from $A - \{a, b\}$ one should get equivalent words from these two equivalent words. But we get $babv_B^1 av_B^2$ and $babw_B^1 aw_B^2$ assumed not to be equivalent. Contradiction. Hence, G' is UR. \square

If $\{\{a, b\}, B\}$ is a good split of G' which is connected with at least 5 vertices but is UR, we get a contradiction with the second special case. If $\{\{a, b\}, B\}$ is not a good split, then $\{\{a\} \cup C, \{b\} \cup D\}$ is also a split for some bipartition $\{C, D\}$ of B . If there are edges between a and D in G there is no edge between C and D (otherwise there would be edges between b and a , a and D , C and D but no edge between b and C , hence $\{\{a\} \cup C, \{b\} \cup D\}$ would not be a split). Then $\{\{a\} \cup C, (A - \{a\}) \cup D\}$ is a split, so $\{A, B\}$ is not good. If there is no edge between a and D , there must be edges between C and D otherwise G is not connected, but $\{\{a\} \cup C, \{b\} \cup D\}$ is not a split because there is no edge between b and C . This is thus excluded which completes the proof for this subcase.

Subcase 2 : There are two letters a, b in A that cross B . Let $G' = G[B \cup \{a, b\}]$.

The graph G' has a representation by a word $abw_B^1abw_B^2$ or $abw_B^1baw_B^2$. The proof is the same in the two cases. We only consider the first one.

Claim 2: The graph G' is UR : Assume it is not. It has another representation by a word $abv_B^1abv_B^2$ which is inequivalent to the first.

Consider now the word $w_A^1v_B^1w_A^2v_B^2$. It defines G . Hence it is equivalent to $w_A^1w_B^1w_A^2w_B^2$. By deleting letters one should get from these two equivalent words two other equivalent words. But we get $abv_B^1abv_B^2$ and $abw_B^1abw_B^2$ assumed not to be equivalent. Hence, G' is UR. \square

If $\{\{a, b\}, B\}$ is a good split of G' which is connected with more than 5 vertices but is UR, we get a contradiction with the second special case. If $\{\{a, b\}, B\}$ is not a good split, then $\{\{a\} \cup C, \{b\} \cup D\}$ is also a split for some bipartition $\{C, D\}$ of B . Here we again distinguish subcases.

Subsubcase 1 : The vertices a and b are neighbours.

If there are edges between a and D , there are also between b and D , because there are edges between b and $B = C \cup D$ and $\{\{a, b\}, B\}$ is a split. Since G is connected there are edges between C and either a, b , or D . In all cases, using the fact that $\{\{a, b\}, B\}$ and $\{\{a\} \cup C, \{b\} \cup D\}$ are splits, we obtain that there are edges between C and a, b , and D . Moreover, there are subsets C_1 of C and D_1 of D connected uniformly by these edges. There is a subset A_1 of $A - \{a, b\}$ connected uniformly with C_1 and D_1 because $\{A, B\}$ is a split. It follows that $\{\{a\} \cup C, (A - \{a\}) \cup D\}$ is a split, so $\{A, B\}$ is not good. Contradiction.

If there is no edge between a and D , there are between a and C , whence between b and C (because there are edges between b and $B = C \cup D$ and there cannot be between b and D because $\{\{a, b\}, B\}$ is a split and there is no edge between a and D). Thus there are edges between C and D but there should be also between a and D because $\{\{a\} \cup C, \{b\} \cup D\}$ is a split. This gives a contradiction. Hence, this subcase cannot happen.

Subsubcase 2 : The vertices a and b are not neighbours.

We complement G locally at a vertex c of B linked to a and to b . Then a and b are no longer neighbours in $G * c$. Lemmas 2.1 of [4], saying that two graphs transformed by local complementation have the same splits and the fact that if X is a set of vertices containing c , then $G[X] * c = (G * c)[X]$, reduce this subcase to the previous one.

This complete the proof of the proposition. \square

We now prove the following proposition stated in Section 4. (See this section for notation).

Theorem 10 : If w, w' are double occurrence words such that $G(w) = G(w')$ is connected and $F(w) = F(w')$, then $w = w'$.

Proof : The proof is by induction on the length of w . The cases of words w of length 2 or 4 are trivial. Hence we consider w with at least 3 different letters, assuming the result for all shorter words.

Case 1 : $G(w)$ has a good split.

We let $G(w) = H \boxtimes_{(h,h)} K$ and w' be another word such that $G(w) = G(w')$ and $F(w) = F(w')$. It follows from Proposition 9 that $w \sim w_H^1 w_K^1 w_H^2 w_K^2$ where $G(w_H^1 h w_H^2 h) = H$ and $G(w_K^1 h w_K^2 h) = K$.

Without loss of generality, the first letter of w is in H . It is not h . We can write $w = u_1 v_1 u_2 v_2 u_3$ where $G(u_1 h u_2 h u_3) = H$, $G(h v_1 h v_2) = K$, for some words u_1, v_1, u_2, v_2, u_3 where only u_3 can be empty. Since $F(w) = F(w')$, the word w' begins as w , and thus, also using Proposition 9, $w' = u'_1 v'_1 u'_2 v'_2 u'_3$ where $G(u'_1 h u'_2 h u'_3) = H$, $G(h v'_1 h v'_2) = K$, and only u'_3 can be empty. We also have $F(u_1 v_1 u_2 v_2 u_3) = F(u'_1 v'_1 u'_2 v'_2 u'_3)$.

We claim that : $F(u_1 h u_2 h u_3) = F(u'_1 h u'_2 h u'_3)$ and $F(h v_1 h v_2) = F(h v'_1 h v'_2)$. Let a be a letter in $v_1 \cap v_2$. Let x be obtained from w by deleting all letters corresponding to vertices of K except a . Let x' be obtained similarly from w' . We have $F(x) = F(x')$. But $x = u_1 a u_2 a u_3$ and $x' = u'_1 a u'_2 a u'_3$, hence $F(u_1 h u_2 h u_3) = F(u'_1 h u'_2 h u'_3)$, hence using the induction hypothesis, we have $u_1 h u_2 h u_3 = u'_1 h u'_2 h u'_3$. By a similar argument we have $h v_1 h v_2 = h v'_1 h v'_2$. Hence $w = w'$.

Case 2 : $G(w)$ has no good split.

There are several subcases.

Subcase 1 : $G(w)$ is a clique, then $w = F(w)F(w)$.

Subcase 2 : $G(w)$ is a star (with center a), then w is equivalent to $aua\tilde{u}$ for some u , and the result follows easily.

Subcase 3 : $G(w)$ has 3 or 4 vertices. The connected graphs with 3 vertices are K_3 and S_2 . Thus they are treated in Subcases 1 and 2.

The connected graphs with 4 vertices are $K_4, S_3, C_4, K_4 - e, P_4$ and K_3 with a pending edge. The first two are treated by Subcases 1 and 2. The last four have good splits, hence they do not have to be considered here.

Theorems 1 and 10 of [18] establish that a connected graph either is a clique, or a star, or is prime, or has a good split. Hence, the only remaining case is here the following :

Subcase 4 : $G(w)$ is prime with at least 5 vertices.

Then $w = auav$ for some u and v , both non empty. By Proposition 4, the word w' if different of w is of the possible forms : $avau$, $a\tilde{u}a\tilde{v}$ or $a\tilde{v}a\tilde{u}$.

We will prove that $w = w'$.

Subsubcase 1 : $w' = avau$.

The hypothesis $F(w) = F(w')$ implies that $u = bu', v = bv'$. Hence $w = abu'abv', w = abv'abu'$. But $G(w)$ is not prime since $\{\{a, b\}, V(w) - \{a, b\}\}$ is a split. This is excluded.

Subsubcase 2 : $w' = a\tilde{u}a\tilde{v}$. Using the hypothesis $F(auav) = F(a\tilde{u}a\tilde{v})$ we examine several possibilities:

(i) Either u has length at least 2 and then $u = bu'b, F(abu'bav) = F(ab\tilde{u}'ba\tilde{v})$; but $G(w)$ is not prime since $\{\{a, b\}, V(w) - \{a, b\}\}$ is a split. This is excluded.

(ii) Or $u = b$ and then $v = v_1bv_2, F(abav_1bv_2) = F(abav_2b\tilde{v}_1)$, and as above, $G(w)$ is not prime.

(iii) u is empty, but $G(w)$ is not connected, this is excluded.

Subsubcase 3 : $w' = a\tilde{v}a\tilde{u}$.

We have $u = bu'$ and $v = v'b$. Hence $w = abu'av'b$ and $w' = ab\tilde{v}'a\tilde{u}'b$. By deleting b in w and w' , we obtain $x = au'av'$ and $x' = a\tilde{v}'a\tilde{u}'$ for which $G(x) = G(x')$ and $F(x) = F(x')$. We can apply the induction, thus $x = x'$ hence $w = w'$. The proof is complete. \square