On the Clique–Width of Perfect Graph Classes
Extended Abstract

Martin Charles Golumbic and Udi Rotics*

Department of Mathematics and Computer Science
Bar-Ilan University
Ramat-Gan, Israel
{golumbic,rotics}@macs.biu.ac.il

Abstract. Graphs of clique–width at most k were introduced by Courcelle, Engelfriet and Rozenberg (1993) as graphs which can be defined by k-expressions based on graph operations which use k vertex labels.
In this paper we study the clique–width of perfect graph classes.
On one hand, we show that every distance–hereditary graph, has clique–width at most 3, and a 3–expression defining it can be obtained in linear time. On the other hand, we show that the classes of unit interval and permutation graphs are not of bounded clique–width. More precisely, we show that for every $n \in \mathcal{N}$ there is a unit interval graph I_n and a permutation graph H_n having n^2 vertices, each of whose clique–width is exactly $n+1$. These results allow us to see the border within the hierarchy of perfect graphs between classes whose clique–width is bounded and classes whose clique–width is unbounded.
Finally we show that every $n \times n$ square grid, $n \in \mathcal{N}$, $n \geq 3$, has clique–width exactly $n + 1$.

1 Introduction

The notion of clique–width of graphs was first introduced by Courcelle, Engelfriet and Rozenberg in [CER93], as graphs which can be defined by k-expressions based on graph operations which use k vertex labels. The clique–width of a graph G, denoted by $cwd(G)$, is defined as the minimum number of labels needed to construct G, using the 3 graph operation: disjoint union ($\oplus$), connecting vertices with specified labels (η) and renaming labels (ρ). More details, are given in section 2.

A detailed study of clique–width is [CO98]. Clique–width has analogous properties to tree–width: If the clique–width of a class of graphs $\mathcal{C}$ is bounded by k (and the k–expression can be computed from its corresponding graph in time $T(|V|+|E|)$, then every decision, optimization, enumeration or evaluation problem on $\mathcal{C}$ which can be defined by a Monadic Second Order formula ψ can be solved in time $c_k \cdot O(|V| + |E|) + T(|V| + |E|)$ where c_k is a constant which

* Supported in part by postdoctoral fellowships at Bar-Ilan University and the University of Toronto.

Widmayer et al. (Eds.): WG'99, LNCS 1665, pp. 135–147, 1999.

depends only on ψ and k, where $|V|$ and $|E|$ denote the number of vertices and edges of the input graph, respectively. For details, cf. [CMRa,CMRb].

In this paper we study the clique–width of perfect graph classes. We first show that:

Theorem 1. *For every distance–hereditary graph G, $cwd(G) \leq 3$, and a 3–expression defining it can be constructed in time $O(|V| + |E|)$.*

Let $d_G(x, y)$ denote the length of the shortest path connecting vertices x and y in the graph G. A graph G is called *distance hereditary* if for every connected induced subgraph H of G, $d_G(x, y) = d_H(x, y)$ holds for every pair of vertices from H. These graphs were introduced by E. Howorka [How77] and have been studied intensively in recent years, cf. [DM88,HM90,Dra94,DNB97,BD98]. Linear time $O(|E| + |V|)$ algorithms were presented for all the following problems on distance hereditary graphs: dominating set [BD98], Steiner tree [BD98], maximum weighted clique [HM90], maximum weighted stable set [HM90], diameter [Dra94], and diametral pair [DNB97].

Since all these problems are in the class of Monadic Second Order Logic optimization problems presented in [CMRa], it follows from Theorem 1 above and from Theorem 4 of [CMRa] that all these problems, and many others, have linear time solutions on the class of distance–hereditary graphs. For example:

Corollary 1. *All the following problems have linear time $O(|V| + |E|)$ solution on the class of distance–hereditary graphs: minimum dominating set, minimum connected dominating set, minimum Steiner tree, maximum weighted clique, maximum weighted stable set, diameter, domatic number for fixed k, vertex cover, and k–colorability for fixed k.*

Other problems which are known to have linear time solutions on the class of distance hereditary graphs are: central vertex [Dra94], radius [Dra94], minimum r–dominating clique [Dra94],and the connected r–domination problem [BD98]. These problems cannot be added to the list of problems mentioned in Corollary 1 above, since they are not included in the class of Monadic Second Order Logic optimization problems presented in [CMRa].

Clearly Theorem 1 above also holds for any subclass of the class of distance hereditary graphs. For example:

Corollary 2. *Let $\mathcal{C}$ be any of the following graph classes, (defined in [PW99]): block graphs, block duplicate graphs, restricted block duplicate graphs, restricted unimodular chordal graphs, (6,2)–chordal bipartite graphs and Ptolematic graphs. For every graph $G \in \mathcal{C}$, $cwd(G) \leq 3$, and a 3–expression defining it can be constructed in time $O(|V| + |E|)$.*

We say that a class of graphs $\mathcal{C}$ is *not of bounded clique–width* if there is no fixed integer k, such that for every graph $G \in \mathcal{C}$, $cwd(G) \leq k$. We continue by showing that:

Theorem 2. *The class of unit interval graphs is not of bounded clique–width.*

Theorem 3. *The class of permutation graphs is not of bounded clique–width.*

Since many graph classes contain the classes of unit interval or permutation graphs, it follows that many perfect graph classes are not of bounded clique–width. For example:

Corollary 3. *All the following graph classes (defined in [Gol80]) and their complements are not of bounded clique–width: interval graphs, circle graphs, circular arc graphs, unit circular arc graphs, proper circular arc graphs, directed path graphs, undirected path graphs, comparability graphs, chordal graphs, and strongly chordal graphs.*

The reason the complements of all graph classes mentioned in Corollary 3 are not of bounded clique–width, is that for every graph G, $cwd(\overline{G}) \leq 2 * cwd(G)$, (cf. [CO98]).

Finally, we show that:

Theorem 4. *For every $n \times n$ square grid G, $n \in \mathcal{N}$, $n \geq 3$, $cwd(G) = n + 1$.*

Corollary 4. *For every $n \times m$ rectangular grid G, $n, m \in \mathcal{N}$, $n, m \geq 3$, $min\{n, m\} + 1 \leq cwd(G) \leq min\{n, m\} + 2$.*

Theorem 4 above improves the result of Makowsky and Rotics (cf. [MR99]), who showed that for every $n \times n$ square grid G, $cwd(G) \geq n/3$. The clique–width of the 2×2 grid is easily seen to equal 2.

In this extended abstract we just sketch the proofs of the theorems mentioned above. The detailed proofs will be presented in the full paper.

2 Background

In this section we define the notions of graph operations and clique–width, as presented in [CO98].

Definition 1 ((k–graph)). *A* k-graph *is a labeled graph with (vertex) labels in $\{1, 2, \ldots, k\}$. A k-graph G, is represented as a structure $\langle V, E, V_1, \ldots, V_k \rangle$, where V and E are the sets of vertices and edges respectively, and $V_1, \ldots, V_k$ form a partition of V, such that V_i is the set of vertices labeled i in G. Note that some V_i's may be empty. A non-labeled graph $G = \langle V, E \rangle$, will be considered as a 1-graph with all vertices labeled by 1.*

Definition 2 (($G \oplus H$)). *For k-graphs G, H such that $G = \langle V, E, V_1, \ldots, V_k \rangle$ and $H = \langle V', E', V'_1, \ldots, V'_k \rangle$ and $V \cap V' = \emptyset$ (if this is not the case then replace H with a disjoint copy of H), we denote by $G \oplus H$, the disjoint union of G and H such that:*

$$G \oplus H = \langle V \cup V', E \cup E', V_1 \cup V'_1, \ldots, V_k \cup V'_k \rangle$$

Note that $G \oplus G \neq G$.

Definition 3 (($\eta_{i,j}(G)$)). *For a k-graph G as above we denote by $\eta_{i,j}(G)$, where $i \neq j$, the k-graph obtained by connecting all the vertices labeled i to all the vertices labeled j in G. Formally:*

$$\eta_{i,j}(G) = \langle V, E', V_1, \ldots, V_k \rangle \ , \ where$$

$$E' = E \cup \{(u, v) : u \in V_i, \ v \in V_j\}$$

Definition 4 (($\rho_{i \to j}(G)$)). *For a k-graph G as above we denote by $\rho_{i \to j}(G)$ the k–graph obtained by the renaming of i into j in G such that:*

$$\rho_{i \to j}(G) = \langle V, E, V_1', \ldots, V_k' \rangle, \ where$$

$V_i' = \emptyset$, $V_j' = V_j \cup V_i$, and $V_p' = V_p$ for $p \neq i, j$.

These graph operations have been introduced in [CER93] for characterizing graph grammars. For every vertex v of a graph G and $i \in \{1, \ldots, k\}$, we denote by $i(v)$ the k-graph consisting of one vertex v labeled by i.

Example 1. A clique with four vertices u, v, w, x can be expressed as:

$$\rho_{2 \to 1}(\eta_{1,2}(2(u) \oplus \rho_{2 \to 1}(\eta_{1,2}(2(v) \oplus \rho_{2 \to 1}(\eta_{1,2}(1(w) \oplus 2(x)))))))$$

Definition 5 ((k–expression)). *With every graph G one can associate an algebraic expression which defines G built using the 3 types of operations mentioned above. We call such an expression a k–expression defining G, if all the labels in the expression are in $\{1, \ldots, k\}$. Trivially, for every graph G, there is an n–expression which defines G, where n is the number of vertices of G.*

Definition 6 ((The clique–width of a graph G, $cwd(G)$)). *Let $\mathcal{C}(k)$ be the class of graphs which can be defined by k–expressions. The clique–width of a graph G, denoted $cwd(G)$, is defined by: $cwd(G) = Min\{k : G \in \mathcal{C}(k)\}$.*

$\mathcal{C}(1)$ is the class of edge-less graphs, cographs are exactly the graphs of clique–width at most 2, and trees have clique–width at most 3 (cf. [CO98]).

In the following sections when considering a k–expression t which defines a graph G, it will often be useful to consider the tree structure, denoted as $tree(t)$, corresponding to the k–expression t. For that we shall need the following definitions.

Definition 7 (($tree(t)$)). *Let t be any k–expression, and let G be the graph defined by t. We denote by $tree(t)$ the parse tree constructed from t in the usual way. The leaves of this tree are the vertices of G, and the internal nodes correspond to the operations of t, and can be either binary corresponding to $\oplus$ or unary corresponding to η or ρ.*

Definition 8 (($tree(a, t)$)). *Let t be any k–expression, a be any node in t, we denote by $tree(a, t)$ the subtree of $tree(t)$ rooted at a.*

Definition 9 ((t_1 is a sub–expression of t_2)). *Let t_1 be a k–expression and let t_2 be an l–expression, $k \leq l$. We say that t_1 is a sub–expression of t_2 if there exists a node a such that $tree(t_1)$ is the sub–tree of $tree(t_2)$ rooted at a. In other words $tree(t_1)$ is equal to $tree(a, t_2)$.*

Definition 10 ((The label of a vertex v at an internal node a)). *Let t be any k–expression, and let G be the graph defined by t. Let a be any internal node of $tree(t)$ and let v be any vertex of G occurring in $tree(a, t)$, i.e. v is a leaf of $tree(a, t)$. The labels of v may change by the ρ operations in t. However, whenever an operation is applied on a sub–expression t_1 of t which contains v, the label of v (like the labels of all the other vertices occurring in t_1) is well defined. The* label of v at a *is defined as the label that v has immediately before the operation a is applied on the subtree of $tree(t)$ rooted at a.*

3 Distance Hereditary Graphs

Let $d_G(x, y)$ denote the length of the shortest path connecting vertices x and y in the graph G. Recall that a graph G is called *distance hereditary* if for every connected induced subgraph H of G, $d_G(x, y) = d_H(x, y)$ holds for every pair of vertices from H. For every vertex x, we denote by $N(x)$ the set of all neighbors of x (not including x). A *leaf* is a vertex having exactly one neighbor. We say that x and y are *twins* if they have the same neighborhood outside x and y, i.e. $N(x) - \{y\} = N(y) - \{x\}$. The vertices x and y are called *true twins* (resp. *false twins*) if x and y are twins and x is adjacent (resp. not adjacent) to y.

Definition 11 ((Pruning sequence, cf. [HM90])). *Let G be a graph with n vertices denoted by $v_1, \ldots, v_n$, and let $S = \{s_2, \ldots, s_n\}$ be a sequence of pairs of the form $\langle (v_i, v_j), type \rangle$, where $j < i$ and type is either $leaf$, $true$ or $false$. We say that S is a pruning sequence for G, if for $2 \leq i \leq n$, if $s_i = \langle (v_i, v_j), leaf \rangle$ (resp. if $s_i = \langle (v_i, v_j), false \rangle$, or $s_i = \langle (v_i, v_j), true \rangle$) then the subgraph of G induced by $\{v_1, \ldots, v_i\}$ is obtained from the subgraph induced by $\{v_1, \ldots, v_{i-1}\}$ by adding the vertex v_i and making it a leaf (resp. a false twin, or a true twin) of the vertex v_j.*

Theorem 5 (Hammer and Maffray [HM90]). *For every connected graph G, G is distance hereditary if and only if there exists a pruning sequence for G. Moreover, there is a linear time algorithm which constructs a pruning sequence for a given graph G, if it exists, or claims that there is no pruning sequence for G.*

Definition 12 ((Pruning–tree)). *Let G be a graph having n vertices denoted by $v_1, \ldots, v_n$, and let $S = \{s_2, \ldots, s_n\}$ be a pruning sequence for G. The* pruning–tree *corresponding to the pruning sequence S of G, is the labeled ordered tree T constructed as follows:*

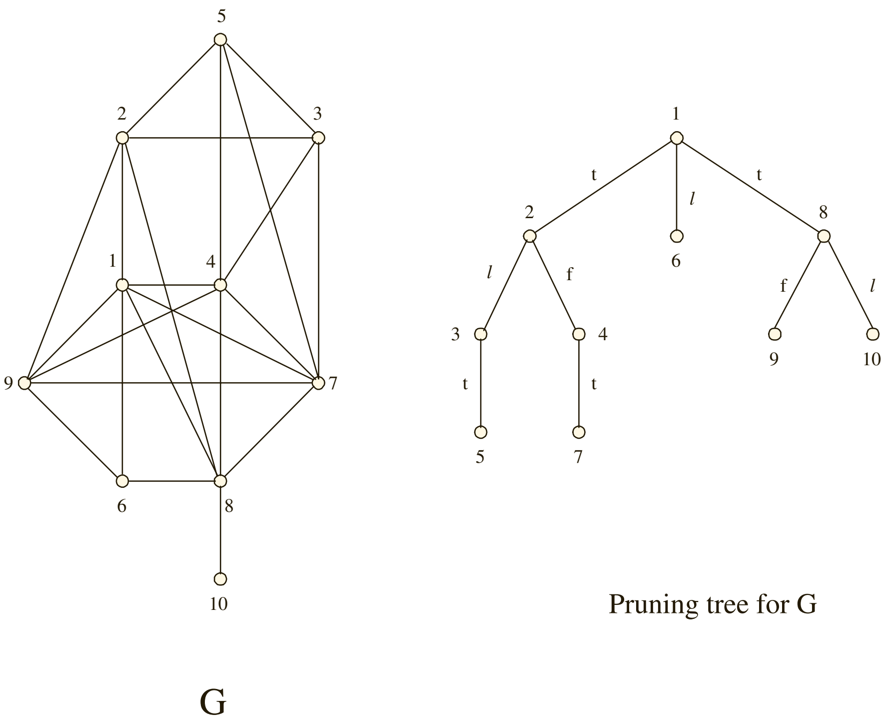

Fig. 1. A connected distance hereditary graph G and a pruning tree for G.

(i) *Set T_1 as the tree consisting of a single root vertex v_1, and set $i := 1$.*
(ii) *Set $i := i + 1$. If $i > n$ then set $T := T_n$ and stop.*
(iii) *Let $s_i = \langle (v_i, v_j), leaf \rangle$ (resp. $s_i = \langle (v_i, v_j), false \rangle$, or $s_i = \langle (v_i, v_j), true \rangle$), then set T_i as the tree obtained from T_{i-1} by adding the new vertex v_i and making it a rightmost son of the vertex v_j, and labeling the edge connecting v_i to v_j by $leaf$ (resp. by $false$ or $true$).*
(iv) *Go back to step (ii) above.*

Example 2. Figure 1 illustrates a connected distance hereditary graph G. The vertices of G are denoted by $\{1, 2, \ldots, 10\}$. Figure 1 also illustrates the pruning–tree corresponding to the pruning sequence S of G defined by

$$S = \{ \langle (2,1), true \rangle, \langle (3,2), leaf \rangle, \langle (4,2), false \rangle, \langle (5,3), true \rangle, \langle (6,1), leaf \rangle, \\ \langle (7,4), true \rangle, \langle (8,1), true \rangle, \langle (9,8), false \rangle, \langle (10,8), leaf \rangle \}.$$

In the figure we denoted true (resp. false or leaf) shortly by t (resp. f or l).

Definition 13 ((T_a)). *Let T be any rooted tree, and let a be any node occurring in T. We denote by T_a the sub–tree of T rooted at a.*

Definition 14 ((True/false twin son, leaf son, twin descendant)). *Let G be a graph having a pruning sequence S, T be the pruning–tree corresponding to S and let v and u be any two vertices of G. We say that v is a* true twin son *(resp.* false twin son, leaf son*) of u, if v is a son of u in T and the edge connecting v to u in T is labeled with true (resp. $false$, $leaf$). We say that v is a* twin descendant *of u, if v is either the same vertex as u, or v is a descendant of u in T such that all the edges of the path connecting v to u in T are labeled with true or $false$.*

Lemma 1. *Let G be a graph having a pruning sequence S with corresponding pruning–tree T, and let a be any internal node in T whose sons in T are denoted by $a_1, \ldots, a_l$ ordered from left to right. For all $1 \leq i < j \leq l$, and for every two vertices v and u occurring in T_{a_i} and T_{a_j} respectively, v is adjacent to u in G if and only if a_i is either a leaf son or a true twin son of a, a_j is either a true or false twin son of a, and v and u are twin descendants of a_i and a_j respectively.*

Let $A \subseteq V$ be a subset of the vertices of $G = \langle V, E \rangle$. We denote by $G[A]$ the subgraph of G induced by A. Furthermore, if $T_{a_1}, \ldots, T_{a_k}$ are disjoint sub–trees of a pruning–tree, then $G[T_{a_1} \cup \ldots \cup T_{a_k}]$ is the subgraph of G induced by the vertices of $T_{a_1} \cup \ldots \cup T_{a_k}$.

The following lemma follows immediately from Lemma 1 above.

Lemma 2. *Let G be a graph having a pruning sequence S, with corresponding pruning–tree T, and let a be any internal node in T whose sons ordered from left to right are $a_1, \ldots, a_l$. For $1 \leq i \leq l$, we have the following:*

(i) If a_i is a false twin son of a, then $G[\{a\} \cup T_{a_i} \cup T_{a_{i+1}} \cup \ldots \cup T_{a_l}]$ is equal to the disjoint union of $G[\{a\} \cup T_{a_{i+1}} \cup \ldots \cup T_{a_l}]$ and $G[T_{a_i}]$.
(ii) If a_i is either a leaf or a true twin son of a, then $G[\{a\} \cup T_{a_i} \cup T_{a_{i+1}} \cup \ldots \cup T_{a_l}]$ can be constructed by taking the disjoint union of $G[\{a\} \cup T_{a_{i+1}} \cup \ldots \cup T_{a_l}]$ and $G[T_{a_i}]$, and connecting all the twin descendants of a_i to a and to all the twin descendants of $a_{i+1}, \ldots, a_l$.

Theorem 1 *For every distance hereditary graph G, $cwd(G) \leq 3$, and a 3–expression defining it can be constructed in time $O(|V| + |E|)$.*
Proof:
Let G be a distance hereditary graph. We assume that G is connected, since if G is not connected we can construct a 3–expression for G by applying the disjoint union operation (i.e. the $\oplus$ operation) on the 3–expressions obtained for the connected components of G. By Theorem 5 above there is a pruning sequence S for G, which can be obtained in linear time. Let T be the pruning–tree corresponding to the pruning sequence S.

Claim. For each internal node a of the pruning tree T, there is a 3–expression t_a which defines the labeled graph G', such that $G' = G[T_a]$, all the twin descendants of a are labeled with 2 in G', and all the other vertices of G' are labeled with 1.

proof of claim: We shall prove the claim by induction on the height of sub–trees of T. The claim trivially holds for all the sub–trees of T of height 1. Suppose the claim holds for all the sub–trees of T of height $n-1$. Let a be any internal node of T such that T_a is of height n and let $a_1, \ldots, a_l$ be the sons of a ordered from left to right. By the induction hypothesis there are 3-expressions $t_{a_1}, \ldots, t_{a_l}$ which defines the disjoint labeled graphs $G[T_{a_1}], \ldots, G[T_{a_l}]$, respectively, such that all the vertices which are twin descendants of $a_1, \ldots, a_l$ are labeled with 2 and all other vertices in these graphs are labeled with 1. We construct the expression t_a as follows:

Procedure A

(i) Set $e_{l+1} := 2(a)$ and set $i := l + 1$.
(ii) Set $i := i - 1$. If $i = 0$ then set $t_a := e_1$ and stop.
(iii) If a_i is either a leaf son or a true twin son of a then set

$$e_i := \rho_{3\to 2}(\eta_{2,3}(t_{a_i} \oplus \rho_{2\to 3}(e_{i+1})))$$

(iv) If a_i is a false twin son of a then set $e_i := t_{a_i} \oplus e_{i+1}$.
(v) Go back to step (ii) above

From Lemma 2 above it follows that for $1 \leq i \leq l$, the 3–expression e_i constructed by the above procedure, defines the graph $G[\{a\} \cup T_{a_i} \cup \ldots \cup T_{a_l}]$. Hence, the 3–expression t_a constructed by the above procedure (which is equal to e_1), defines the graph $G[T_a]$. Since in the graph defined by t_a, all the twin descendants of a are labeled with 2 and all the other vertices are labeled with 1, this completes the proof of Claim 3.

Let x be the root of the pruning–tree T. By the above claim there is a 3–expression t_x which defines the graph G. Moreover, using Procedure A above, it is easy to see that the 3–expression t_x which defines G can be constructed in linear time, and by that the proof of Theorem 1 is completed. □

4 Unit Interval Graphs and Permutation Graphs Are Not of Bounded Clique–Width

In this section we show that the classes of unit interval graphs and permutation graphs are not of bounded clique–width. Below (cf. definition 15) we define the graph I_n which is a unit interval graph with n^2 vertices (cf. Fact 1). Informally, the vertices of the graph I_n can be thought as being arranged in an $n \times n$ square array, such that all the vertices occurring in the same column form a clique, vertices in non-consecutive columns are not connected, and a vertex $v_{i,j}$ occurring in row i and column j is adjacent to all the vertices occurring in column $j + 1$ and in rows $1, \ldots, i - 1$. Figure 2 illustrates the graph I_4, and Figure 3 shows its representation as intersecting intervals.

Definition 15 ((The graph I_n)). *We denote by I_n the graph $\langle V, E\rangle$, where the set of vertices V is defined by:*

$$V = \{v_{i,j} : 1 \leq i \leq n,\ 1 \leq j \leq n\}$$

and the set of edges E is defined by: $E = E' \cup E''$, where

$$E' = \{\ (v_{i_1,j}, v_{i_2,j}) : 1 \leq i_1 \leq n, \\ 1 \leq i_2 \leq n, 1 \leq j \leq n, i_1 \neq i_2\}$$

$$E'' = \{\ (v_{i_1,j}, v_{i_2,j+1}) : 1 \leq j \leq n-1, \\ 2 \leq i_1 \leq n, 1 \leq i_2 \leq i_1 - 1\}$$

Fact 1 *For every $n \in \mathcal{N}$, the graph I_n is a unit interval graph.*

Fact 1 above can be verified by constructing for every $n \in \mathcal{N}$, a unit interval graph presentation for the graph I_n , similar to the one illustrated in Figure 3 for the graph I_4. For example, let $\varepsilon = 1/2n$ and define the (closed) interval corresponding to $v_{i,j}$ to be $J_{i,j} = [j + i\varepsilon, j + 1 + (i-1)\varepsilon]$, $(1 \leq i, j \leq n)$.

Lemma 3. *For every $n \in \mathcal{N}$, $n \geq 2$, $cwd(I_n) = n + 1$.*

Theorem 2 follows immediately from Lemma 3.

A graph $G = \langle V, E\rangle$ is a *permutation graph* if and only if there are two linear ordering of its vertices R_1 and R_2, such that for every two vertices v and u in G, v is adjacent to u if and only if v occurs before u in the linear order R_1 and v occurs after u in the linear order R_2, cf. [Gol80]. Below (cf. definition 16) we define the graph H_n which is a permutation graph (cf. Fact 2). Informally, the vertices of the graph H_n can be put in an $n \times n$ square array, such that all the vertices occurring in the same column form a clique, vertices in non-consecutive columns are not connected, a vertex v occurring in row i and an *odd* column j is adjacent to all the vertices occurring in column $j+1$ and in rows $1, \ldots, i-1$, and a vertex v occurring in row i and *even* column j is adjacent to all the vertices occurring in column $j + 1$ in rows $i + 1, i + 2, \ldots, n$.

Definition 16 ((The graph H_n)). *We denote by H_n the graph $\langle V, E\rangle$, where the set of vertices V is defined by:*

$$V = \{v_{i,j} : 1 \leq i \leq n,\ 1 \leq j \leq n\}$$

and the set of edges E is defined by: $E = E' \cup E'' \cup E'''$, where

$$E' = \{\ (v_{i_1,j}, v_{i_2,j}) : 1 \leq i_1 \leq n, \\ 1 \leq i_2 \leq n, 1 \leq j \leq n, i_1 \neq i_2\}$$

$$E'' = \{\ (v_{i_1,j}, v_{i_2,j+1}) : 1 \leq j \leq n-1, j \textit{ odd}, \\ 2 \leq i_1 \leq n, 1 \leq i_2 \leq i_1 - 1\}$$

$$E''' = \{\ (v_{i_1,j}, v_{i_2,j+1}) : 2 \leq j \leq n-1, j \textit{ even}, \\ 1 \leq i_1 \leq n-1, i_1 + 1 \leq i_2 \leq n\}$$

Fact 2 *For every $n \in \mathcal{N}$, the graph H_n is a permutation graph.*

Lemma 4. *For every $n \in \mathcal{N}$, $n \geq 2$, $cwd(H_n) = n + 1$.*

Theorem 3 follows immediately from Lemma 4.

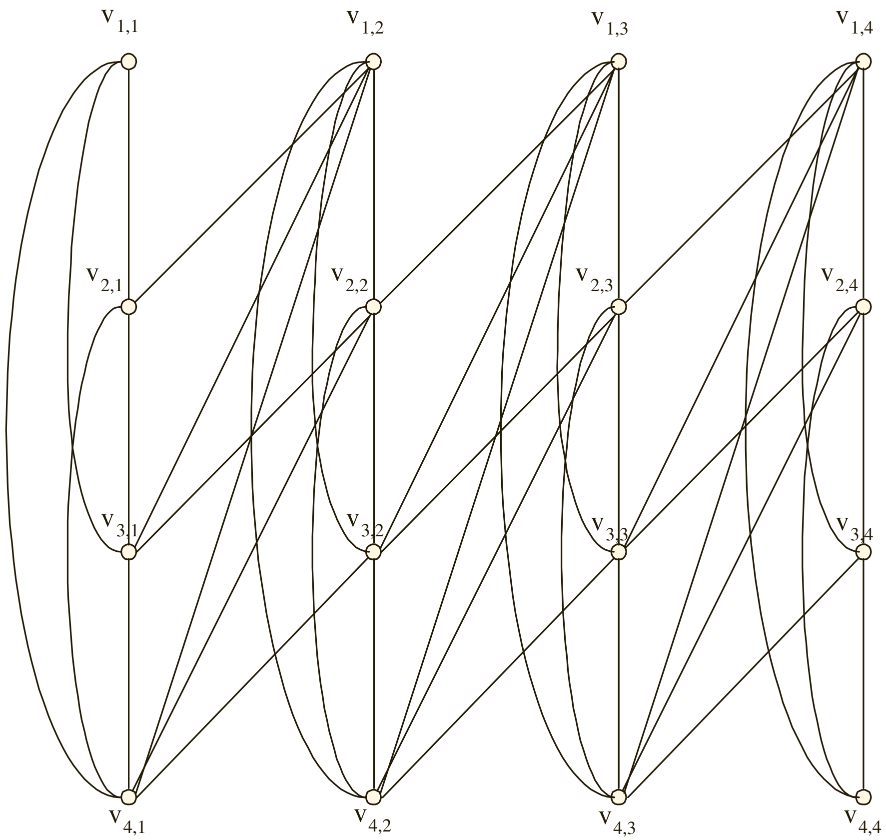

Fig. 2. The graph I_4

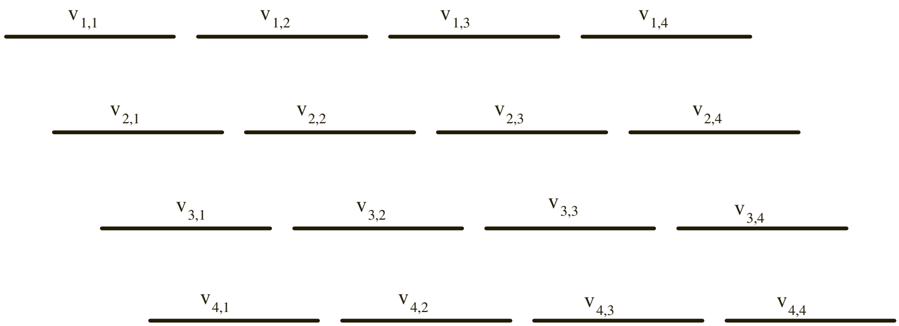

Fig. 3. The unit interval representation of the graph I_4

5 Square Grids

In this section we show that every $n \times n$ square grid, $n \geq 3$, has clique–width exactly $n+1$. Throughout this section we denote by $v_{i,j}$ the vertex of the grid occurring in row i and column j.

Lemma 5. *For every $n \times n$ square grid G, $cwd(G) \leq n+1$.*

Proof:
[Sketch] Let G be an $n \times n$ square grid. We shall prove the lemma by constructing an $n+1$–expression f which defines G. For that we first construct an $n+1$–expression c which defines the subgraph G_L of the grid G induced by the vertices occurring in the lower triangle of G, such that all the vertices of the diagonal of G_L are labeled with labels from 1 to n, and all the other vertices of G_L are labeled with $n+1$. Similarly we construct an $n+1$–expression d which defines the subgraph of the grid G induced by the vertices occurring in the subgraph G_R of G induced by the upper triangle of G. Finally, we construct the $n+1$–expression f, by adding all the vertices of the main diagonal of the grid G, and connecting them to the vertices in the diagonals of the graphs G_L and G_R. □

We now show that $n+1$ is also the lower bound for $cwd(G)$. Recall (cf. definition 8 above) that for a k–expression t and for every internal node a of $tree(t)$, we denote by $tree(a,t)$ the sub–tree of $tree(t)$ rooted at a.

Lemma 6. *Let G be an $n \times n$ square grid, $n \in \mathcal{N}$, $n \geq 3$, let t be a k–expression which defines G, let a be the highest $\oplus$ node in $tree(t)$, let b and c be the highest $\oplus$ nodes in the sub–trees rooted at the left and right sons of a, respectively. If neither graph defined by $tree(b,t)$ and $tree(c,t)$ contains a full row of the grid G, then $k \geq n+1$. Similarly, if neither graph defined by $tree(b,t)$ and $tree(c,t)$ contains a full column of the grid G, then $k \geq n+1$.*

Lemma 7. *Let G be an $n \times n$ square grid, let t be a k–expression which defines G, let d be an internal $\oplus$ node in $tree(t)$. If the graph defined by $tree(d,t)$ contains a full row of the grid and does not contain a full column of the grid, then $k \geq n+1$. Similarly, if the graph defined by $tree(d,t)$ contains a full column of the grid and does not contain a full row of the grid, then $k \geq n+1$.*

Lemma 8. *Let G be an $n \times n$ square grid, let t be a k–expression which defines G, let d be an internal $\oplus$ node in $tree(t)$ and let e and f be the highest $\oplus$ nodes in the sub–trees rooted at the left and right sons of d, respectively. If the graph defined by $tree(d,t)$ contains a full row of the grid and a full column of the grid, and neither graph defined by $tree(e,t)$ or $tree(f,t)$ contains a full row or a full column of the grid, then $k \geq n+1$.*

Theorem 4 *For every $n \times n$ square grid G, $n \in \mathcal{N}$, $n \geq 3$, $cwd(G) = n+1$.*
Proof:
Let G be an $n \times n$ square grid $n \in \mathcal{N}$, $n \geq 3$. By Lemma 5 above $cwd(G) \leq n+1$.

We shall show that $cwd(G) > n$, which implies that $cwd(G) = n+1$. Suppose that there is a k–expression t which defines G, and $k \leq n$. Let a be the highest $\oplus$ node in $tree(t)$, let b and c be the highest $\oplus$ nodes in the sub–trees rooted at the left and right sons of a, respectively. If neither graph defined by $tree(b,t)$ and $tree(c,t)$ contains a full column of the grid, or neither contains a full row of the grid, then by Lemma 6 above $k \geq n+1$, a contradiction. Hence, we assume without loss of generality, that the graph defined by $tree(c,t)$ contains a full row of the grid. Suppose there exist a node d in $tree(c,t)$, such that either $tree(d,t)$ contains a full row of the grid and does not contain a full column of the grid, or $tree(d,t)$ contains a full column of the grid and does not contain a full row of the grid. In either case, by Lemma 7 above $k \geq n+1$, a contradiction. Hence, there exist a node d in $tree(c,t)$ such that $tree(d,t)$ contains a full row and a full column of the grid, and both $tree(e,t)$ and $tree(f,t)$ do not contain a full column or row of the grid, where e and f are the highest $\oplus$ nodes in the sub–trees rooted at the left and right sons of d, respectively. In this case, by Lemma 8 above $k \geq n+1$, a contradiction.

Since we have considered all possible cases, we conclude that the assumption that $k \leq n$, was not correct, which implies that $k \geq n+1$. □

Corollary 4 *For every $n \times m$ rectangular grid G, $n,m \in \mathcal{N}$, $n,m \geq 3$, $min\{n,m\}+1 \leq cwd(G) \leq min\{n,m\}+2$.*

Acknowledgments

We are indebted to Bruno Courcelle, Johann Makowsky, Derek Corneil and Michel Habib for their helpful comments.

References

BD98. A. Brandstädt and F.F. Dragan. A linear-time algorithm for connected r-domination and steiner tree on distance-hereditary graphs. *Networks*, 31:177–182, 1998. 136

CER93. B. Courcelle, J. Engelfriet, and G. Rozenberg. Handle-rewriting hypergraph grammars. *J. Comput. System Sci.*, 46:218–270, 1993. 135, 138

CMRa. B. Courcelle, J.A. Makowsky, and U. Rotics. Linear time solvable optimization problems on certain structured graph families, extended abstract. Graph Theoretic Concepts in Computer Science, 24th International Workshop, WG'98, volume 1517 of Lecture Notes in Computer Science, pages 1-16. Springer Verlang, 1998. 136

CMRb. B. Courcelle, J.A. Makowsky, and U. Rotics. On the fixed parameter complexity of graph enumeration problems definable in monadic second order logic. To appear in Disc. Appl. Math. 136

CO98. B. Courcelle and S. Olariu. Upper bounds to the clique-width of graphs. to appear in Disc. Appl. Math. (http://dept-info.labri.u-bordeaux.fr/∼courcell/ActSci.html), 1998. 135, 137, 138

DM88. A. D'Atri and M. Moscarini. Distance–hereditary graphs Steiner trees and connected domination. *SIAM J. Comput.*, 17:521–538, 1988. 136

DNB97. F.F. Dragan, F. Nicolai, and A. Brandstädt. LexBFS-orderings and powers of graphs. Graph Theoretic Concepts in Computer Science, 22th International Workshop, WG'96, volume 1197 of Lecture Notes in Computer Science, pages 166-180, 1997. 136

Dra94. F.F. Dragan. Dominating cliques in distance-hereditary graphs. Algorithm theory—SWAT'94, volume 824 of Lecture Notes in Computer Science, pages 370-381, 1994. 136

Gol80. M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, New York, 1980. 137, 143

HM90. P. L. Hammer and F. Maffray. Completely separable graphs. *Disc. Appl. Math.*, 27:85–99, 1990. 136, 139

How77. E. Howorka. A characterization of distance-hereditary graphs. *Q. J. Math. Oxford Ser. (2)*, 28:417–420, 1977. 136

MR99. J.A. Makowsky and U. Rotics. On the classes of graphs with few P_4's. To appear in the International Journal of Foundations of Computer Science (IJFCS), 1999. 137

PW99. U. N. Peled and J. Wu. Restricted unimodular chordal graphs. To appear in Journal of Graph Theory, 1999. 136