Chapter 1

Introduction

The notion of clique-width of a graph was introduced in connection with graph grammars by Courcelle, Engelfriet and Rozenberg in [27] and has been extensively studied in recent years. Informally, the clique-width $cwd(G)$ of a graph G is the minimum number of labels needed to construct G, using the following operations: create a new vertex with a label, taking disjoint union of two graphs, connecting vertices with particular different labels, and renaming labels. An expression built from the above operations using k labels is called a k-expression constructing G. A related notion was introduced by Wanke [62].

Clique-width of graphs is interesting in an algorithmic sense. Many $\mathbb{NP}$-hard graph problems can be solved in linear time when restricted to graphs of clique-width at most k, for some fixed k, assuming that a k-expression defining the input graph is given. For example, all decision problems which are expressible in Monadic Second Order Logic using quantifiers on vertices and vertex sets ($MSO(\tau_1)$-Logic) are decidable in linear time on graphs of clique-width bounded by a constant([26]). Examples of such problems include k-Colorability for fixed k, [1] k-Partition into Cliques for fixed k, k-Domatic Number for fixed k, and Planarity. (For these and other graph problems considered in this thesis, see [40].) The $MSOL(\tau_1)$ has been extended to $LinEMSOL(\tau_{1,p})$ by counting mechanisms allowing the expressibility of optimization problems concerning maximum or minimum vertex sets ([28]). Examples of $\mathbb{NP}$-hard problems expressible in $LinEMSOL(\tau_{1,p})$ include Maximum Weight Stable Set,[2] Maximum Weight Clique, Vertex Cover, Domi-

---

[1] The 3-colorability of a graph $G = (V, E)$ can be expressed in $MSOL(\tau_1)$ as follows:
$\exists X, Y, Z \subseteq V$: $\forall v \in V(v \in X \vee v \in Y \vee v \in Z) \wedge \forall v \in V(v \in X \to \neg(v \in Y) \wedge \neg(v \in Z)) \wedge$
$\forall v \in V(v \in Y \to \neg(v \in X) \wedge \neg(v \in Z)) \wedge \forall v \in V(v \in Z \to \neg(v \in X) \wedge \neg(v \in Y)) \wedge$
$\forall u, v \in V(\{u, v\} \in E \to \neg(u \in X \wedge v \in X) \wedge \neg(u \in Y \wedge v \in Y) \wedge \neg(u \in Z \wedge v \in Z))$.

[2] The Maximum Weight Stable Set problem can be expressed in $LinEMSOL(\tau_{1,p})$ as follows: $\max_{S \subseteq V} \sum_{v \in S} f(v) : \forall u, w \in V(\{u, w\} \in E \to \neg(u \in S \wedge w \in S))$, where f is a given evaluation function associating integer weight values to the vertices of $G = (V, E)$.

nating Set, and Steiner Tree. All such optimization problems are solvable in linear time on graphs of clique-width bounded by a constant.

Another fact that makes the study of clique-width attractive is that the clique-width is a complexity measure on graphs somewhat similar to treewidth. Roughly speaking, the treewidth of a graph indicates how the structure of the graph is 'close' to a tree (connected graphs of treewidth one are exactly trees). The concept of treewidth was introduced by Robertson and Seymour [57], and is widely investigated. Similarly to clique-width, one of the important results concerning treewidth is that any graph problem expressible in $MSOL(\tau_2)$ and $LinEMSOL(\tau_{2,p})$, the logics similar to $MSOL(\tau_1)$ and $LinEMSOL(\tau_{1,p})$ where quantifications over subsets of edges are allowed, has a linear time algorithm when restricted to graphs of treewidth bounded by a constant ([26, 2, 29]). A relation between treewidth and clique-width has been shown by Courcelle and Olariu [30] (and improved by Corneil and Rotics [25]): If a graph has bounded treewidth, it has bounded clique-width. Thus, the graphs of bounded clique-width form a proper superclass of graphs of bounded treewidth (note that the complete graphs have clique-width 2 but arbitrarily large treewidth). With respect to the efficient solvability of problems expressible in Monadic Second Order Logic, graphs of bounded clique-width therefore generalize graphs of bounded treewidth in a 'right way': Clique-width is a more powerful concept than treewidth.

By the facts above, the following two problems are fundamental in the study of clique-width:

- Which graphs have clique-width bounded by a constant? (The characterization problem)

- Given a graph G and an integer k, decide whether or not G has clique-width at most k. (The recognition problem)

Both problems are still open and seem to be very difficult. The only known partial solution for the characterization problem is: Graphs of clique-width at most 2 are exactly the cographs (graphs without induced path P_4 on four vertices). A description of graphs of clique-width at most 3 still remains an open problem. The only known partial results concerning the recognition problem in case k is fixed are: Graphs of clique-width at most 2 can be recognized in linear time as recognizing cographs can be done in linear time [24], and graphs of clique-width at most 3 can be recognized in time $O(n^2m)$ [21] where n is the number of vertices and m is the number of edges of the input graph. The complexity status of recognizing graphs of clique-width at most 4 is still unknown. It is conjectured that if k is not fixed, the recognition problem may be $\mathbb{NP}$-complete (cf. [21, 25]). Note that the corresponding recognition problem for treewidth is resolved: "Given G and integer k, does G has treewidth at most k?" is $\mathbb{NP}$-complete [1]. If k is fixed, the problem is solvable in linear time [6].

Espelage, Gurski and Wanke showed in their recently published paper [34] this interesting result: Deciding clique-width for graphs of bounded treewidth is solvable in linear time.

Contributions of the thesis. The aim of this thesis is to identify new graph classes of bounded clique-width and new graph classes of unbounded clique-width. Our contributions can be considered as a step towards a solution of both the characterization problem and the recognition problem of clique-width. Moreover, as an important consequence of our study, several known polynomial time algorithms for optimization problems on certain graph classes can be improved to linear time by our results. The thesis is organized as follows.

Chapter 2 provides basic notions and facts used throughout the thesis.

Chapter 3 presents new and very restricted graph classes of unbounded clique-width. The main results are: K_4-free co-chordal graphs, P_8-free chordal bipartite graphs, and (P_6, diamond, K_4)-free weakly chordal graphs have unbounded clique-width.

As mentioned above, graphs of clique-width at most 2 are exactly the cographs, also called P_4-free graphs. Cographs were independently discovered and studied under different names in various fields (see e.g. [20, 53, 60, 45, 46, 59, 22]). They form a very popular graph class due to the fact that a lot of $\mathbb{NP}$-hard problems have linear time algorithms when restricted on this graph class [23].

Various reseachers introduced cograph generalizations in many directions: Tinhofer defined the *tree-cographs* where the recursion starts (instead of a vertex by cograph's definition) with any tree. Bacsó and Tuza [4] and Fouquet et al [38] gave different P_k-free graph characterizations for $k \geq 4$ (graphs without induced path on k vertices), Babel and Olariu [3] generalized cographs by bounding the number of P_4's in the considered graph in term of (q,t)-graphs: A graph is a (q,t)-*graph* if no set of at most q vertices induces more than t P_4's. This definition again generalized several cograph generalizations such as P_4-*sparse* introduced by Hoang [49] (= $(5,1)$-graphs) and P_4-*laden* by Giakoumakis [41] (a subclass of $(6,3)$-graphs).

In Chapters 4–7 of this thesis we investigate the structure and clique-width of *all* graph classes defined by forbidden one-vertex extensions of the P_4 in a systematic way: Let $\mathcal{F}$ denote the set of the 10 one-vertex extensions of the P_4 in Figure 1.1. For $\mathcal{F}' \subseteq \mathcal{F}$, there are 1024 classes of $\mathcal{F}'$-free graphs. We refer to these classes by the enumeration of their forbidden subgraphs with respect to Figure 1.1. For example, by definition, a graph G is P_4-*sparse* if every set of five vertices in G contains at most one P_4 ([48]). Obviously, a graph is P_4-sparse if and only if it is {2,3,4,5,6,8,9}-free.

In Chapter 7, we give a complete classification of these 1024 graph classes according

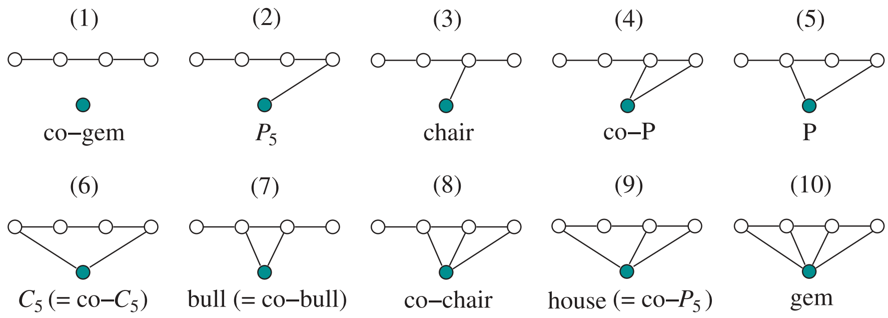

Figure 1.1: All one-vertex extensions of a P_4

to bounded and unbounded clique-width. The most difficult parts in doing this classification are the cases of {1,9}-free graphs and of {1,10}-free graphs.

Clique-width of {1,9}-free graphs is discussed in Chapter 5. The discussion relies on the fact that chordal co-gem-free graphs have bounded clique-width. The proof uses a certain tree structure of such a graph, and is quite involved and interesting in its own right.

Chapter 6 deals with clique-width of {1,10}-free graphs. Among the 1024 classes, the class of {1,10}-free graphs is perhaps the most natural generalization of cographs; it is the biggest class among all classes by excluding two one-vertex extensions of the P_4 which is self-complementary and has bounded clique-width. The discussion is very technical and uses intensively the structure of cographs induced by the neighborhood and non-neighborhood of a vertex in a {1,10}-free graph.

As an example of the use of clique-width in solving certain graph problems on graphs of bounded clique-width, we discuss the {3,4,10}-free graphs in Chapter 4. We show that such a graph has clique-width at most 9 and a 9-expression can be obtained in linear time. This improves known optimization results on {3,4,10}-free graphs in the literature.

Chapter 2

Preliminaries

2.1 Some basic graph notions

Notions and definitions not given here can be found in any standard textbook on graph theory or graph algorithms, e.g. [43].

Let $G = (V, E)$ be a finite undirected graph, and let $|V| = n$ and $|E| = m$. The *complement graph* $\overline{G} = (V, \overline{E})$ of G is defined by $\overline{E} = \{uv : u, v \in V, u \neq v$ and $uv \notin E\}$. Let $N_H(v) := \{u : u \in H, u \neq v, uv \in E\}$ denote the (open) *neighborhood* of v and $N_H[v] := N(v) \cup \{v\}$ the *closed neighborhood* of v in H for a subset $H \subseteq V$. When $H = V$, we omit the index. Sometimes, we write $x \sim y$ for $xy \in E$ and $x \not\sim y$ for $xy \notin E$.

G is called *connected* if every pair of vertices in G is connected by a path. Otherwise, G is *disconnected*. A *component* of G is a maximal connected subgraph of G. A *nontrivial component* of a graph contains at least two vertices.

A graph $G' = (V', E')$ is a *subgraph* of G if $V' \subseteq V$ and $E' \subseteq E$. A subgraph $G' = (V', E')$ is an *induced subgraph* of G if $E' = \{uv : uv \in E$ and $u, v \in V'\}$. We say that G' is induced by V' and write $G[V']$ for G'. Throughout this thesis, all subgraph containments are understood as induced subgraph containments. When there is no doubt, we sometimes do not differentiate a vertex set $U \subseteq V$ from the subgraph which U induces.

Let $\mathcal{F}$ denote a set of graphs. A graph G is *$\mathcal{F}$-free* if none of its induced subgraphs is isomorphic to a graph in $\mathcal{F}$.

A vertex set $U \subseteq V$ is a *clique* in G if the vertices in U are pairwise adjacent. A clique of k vertices will be denoted by K_k.

A vertex set $U \subseteq V$ is *stable* (or *independent*) in G if U is a clique in $\overline{G}$. A vertex set $U \subseteq V$ *dominates* G if every vertex outside U has a neighbor in U. A graph

$G = (V, E)$ is *bipartite* if its vertex set V can be partitioned into two disjoint stable sets A, B; if every vertex in part A is adjacent to every vertex in part B, G is called *complete bipartite*.

For $k \geq 1$, we write $P_k = v_1v_2 \ldots v_n$ for a chordless path with k vertices and $k-1$ edges, and for $k \geq 3$, we write $C_k = v_1v_2 \ldots v_nv_1$ for a chordless cycle with k vertices and k edges. A *hole* is an induced subgraph isomorphic to a C_k for $k \geq 5$. An *odd hole* is a hole having odd number of edges. Connected graphs without any cycle C_k, $k \geq 3$, are called *trees*. A graph is *chordal* if it is C_k-free for $k \geq 4$.

For a graph class $\mathcal{G}$ we write co-$\mathcal{G}$ for the class of complement graphs of graphs in $\mathcal{G}$. Thus, the class of *co-chordal graphs* consists of the complements of chordal graphs.

For an integer $n \geq 1$, we write nG for the graph consists of n disjoint copies of the graph G. We denote by $K_{n,m}$ the complete bipartite graph with a bipartition into a stable set of n vertices and a stable set of m vertices. The $K_{1,3}$ is also called *claw*. See Figure 1.1 for the definitions of a gem, chair, P, bull, house and their complements.

For disjoint vertex sets X and Y, the *join* (*co-join*) operation between X and Y creates edges (non-edges) between all vertex pairs $x \in X$ and $y \in Y$. Thus, X *has a join to* Y if for all $x \in X$, $y \in Y$, $xy \in E$, and X *has a co-join to* Y if for all $x \in X$, $y \in Y$, $xy \notin E$. We write x has a join (a co-join) to Y if $X = \{x\}$ and X has a join (a co-join) to Y.

For a vertex set H and a vertex $v \in V \setminus H$ we say v *distinguishes* H, if v has a neighbor and a nonneighbor in H. If a vertex not in H is adjacent to exactly k vertices in H then it is called *k-vertex with respect to H*. We also say H *has no k-vertex* if there is no k-vertex with respect to H. A vertex set $M \subseteq V$ is a *module* in G if for all vertices $x \in V \setminus M$, x has either a join or a co-join to M. Thus, M is a module if and only if no vertex outside M distinguishes M. The *trivial modules* of G are $\emptyset, V$ and the one-elementary vertex sets. A *homogeneous set* in G is a nontrivial module in G. A homogeneous set M is *maximal* if no other homogeneous set properly contains M. A graph containing no homogeneous set is called *prime*. Note that a graph is prime if and only if its complement is prime, and the smallest prime graph with more than two vertices is the P_4.

2.2 The clique-width of graphs

In this section we provide the main definition of the thesis, give examples illustrating the definition, and collect basic facts about clique-width.

Definition. 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 following four operations:

(i) create a single vertex v with an integer label ℓ (denoted by $\ell(v)$);

(ii) disjoint union of two graphs (i.e. co-join) (denoted by $\oplus$);

(iii) join between all vertices with label i and all vertices with label j for $i \neq j$ (denoted by $\eta_{i,j}$);

(iv) relabeling all vertices with label i by label j (denoted by $\rho_{i \to j}$).

Note that these four operations create labeled graphs. Unlabeled graphs are considered as graphs all of whose vertices have the same label.

It should be remarked that the notion of *NLC-width* introduced by Wanke [62] is very similar to clique-width: The *NLC-width* of a graph G, is the minimum number of labels needed to construct G using two graph operations called union ($\times_S$) and relabeling ($\circ_R$).

Definition. A *k-expression* of a graph G describes a sequence of operations (i)-(iv) generating G and using at most k pairwise different labels.

For example, the following 3-expression generates the $C_5 = v_1v_2v_3v_4v_5v_1$ with labels $1, 2, 3$:

$$\eta_{3,1}(\rho_{3\to 2}(\eta_{2,3}(\eta_{1,2}(1(v_1) \oplus 2(v_2)) \oplus \eta_{1,3}(3(v_3) \oplus 1(v_4)))) \oplus 3(v_5)).$$

As another example, the complete graph G with vertices $v_1, \ldots, v_n$ can be inductively expressed with two labels 1 and 2, such that finally all vertices get the same label 1, as follows: If $n = 1$, $\tau := 1(v_1)$ defines G. Let $n > 1$, and let τ' be a 2-expression defining $G - v_n$ such that finally all vertices of $G - v_n$ get the same label 1. Then $\tau := \rho_{2\to 1}(\eta_{1,2}(\tau' \oplus 2(v_n)))$ defines G. The expression τ can also be written in a more compact form as follows:

$\tau := 1(v_1)$
for $i := 2$ **to** n **do** $\tau := \rho_{2\to 1}(\eta_{1,2}(\tau \oplus 2(v_i)))$.

Other examples are discussed in the next section.

The following basic facts on clique-width will be used in our discussions.

Proposition 2.1 ([30]). *If H is an induced subgraph of a graph G, $cwd(H) \leq cwd(G)$.*

Proposition 2.2 ([30]). *For every graph G, $cwd(G) \leq 2 \cdot cwd(\overline{G})$.*

Proposition 2.3 ([28]). *For every graph G, the clique-width $cwd(G)$ of a graph G is the maximum of the clique-width of its prime induced subgraphs.*

Definition. A graph class $\mathcal{G}$ has *bounded clique-width* if there exists a constant k such that, for every member $G \in \mathcal{G}$, $cwd(G) \leq k$. If there is no such constant, $\mathcal{G}$ has *unbounded clique-width*.

It turns out that a graph class has bounded clique-width if and only if it has bounded *NLC-width*. More precisely, Johansson showed in [52] that, for every graph G, $NLC\text{-}width(G) \leq cwd(G)$ and $cwd(G) \leq 2 \cdot NLC\text{-}width(G)$.

Let $\mathcal{G}$ be a graph class. The following facts will be often used without further reference:

First, by the definition of clique-width, we have: If $\mathcal{G}$ has bounded clique-width, every subclass of $\mathcal{G}$ also has bounded clique-width. If $\mathcal{G}$ has unbounded clique-width, every superclass of $\mathcal{G}$ also has unbounded clique-width.

Second, by Proposition 2.2, $\mathcal{G}$ has (un)bounded clique-width if and only if co-$\mathcal{G}$ has (un)bounded clique-width.

Third, by Proposition 2.3, when considering clique-width of graphs in a class $\mathcal{G}$, we may restrict ourselves on the prime members in $\mathcal{G}$, theoretically. Algorithmically, a k-expression defining $G \in \mathcal{G}$ can be constructed from the k-expressions defining the prime members in $\mathcal{G}$ by using the *modular decomposition tree* of G. This fact is described in [28, Proposition 32]. Since the modular decomposition tree of a graph $G = (V, E)$ can be constructed in linear time ([31, 56, 32]), a k-expression defining $G \in \mathcal{G}$ can be constructed in linear time $O(|V|+|E|)$ given the k-expressions defining the prime members in $\mathcal{G}$.

Summarizing, the following algorithmic meaning of clique-width is known.

Theorem 2.1 ([28]). *Let $\mathcal{C}$ be a class of graphs of clique-width at most k such that there is an $O(f(|E|, |V|))$ algorithm, which for each graph G in $\mathcal{C}$, constructs a k-expression defining it. Then for every LinEMSOL$(\tau_{1,p})$ problem on $\mathcal{C}$, there is an algorithm solving this problem in time $O(f(|E|, |V|))$.*

2.3 Basic graph classes of bounded clique-width

This section collects graph classes of bounded clique-width which will be used in this thesis. Graph classes of unbounded clique-width will be considered in Chapter 3.

Cographs. *Cographs* (or *complement reducible graphs*) are recursively defined as follows ([22]):

- A single vertex graph is a cograph;
- If G_1 and G_2 are disjoint cographs then so is their union $G_1 \cup G_2$;
- If G is a cograph, then so is its complement $\overline{G}$.

There are several known characterizations of cographs (see e.g. [15]), of which the following two are of interest for our discussions:

(i) G is a cograph if and only if for every induced subgraph H of G, H or $\overline{H}$ is disconnected;

(ii) G is a cograph if and only if G is P_4-free.

It is well-known that cographs are exactly those graphs of clique-width at most 2. However, because we consider cographs extensively, we will show this fact here for the sake of completeness. Moreover, we point out that a 2-expression for a given cograph can be obtained in linear time.

Let G be a cograph. If G consists of exactly one vertex, say v, then $\tau := 1(v)$ defines G. Let G have more than one vertex. By (i), G or $\overline{G}$ is disconnected. If G is disconnected, then G is a co-join of two smaller cographs G_1 and G_2. If $\overline{G}$ is disconnected, then G is a join of two smaller cographs G_1 and G_2. In each case let τ_i be a 2-expression with labels 1 and 2 defining G_i, $i = 1, 2$. Then in the first case $\tau := \tau_1 \oplus \tau_2$ clearly defines G, and in the second case $\tau := \eta_{1,2}(\rho_{2\to 1}(\tau_1) \oplus \rho_{1\to 2}(\tau_2))$ clearly defines G. Moreover, since τ_i can be inductively constructed in time $O(n_i)$ (n_i is the number of vertices of G_i), τ is constructible in time $O(n_1)+O(n_2) = O(n)$ where n is the number of vertices of G.

Note that it can be shown by easy case analysis that an induced P_4 cannot be generated with 2 labels. Thus, by (ii) cographs are exactly those graphs with clique-width at most 2.

Paths and cycles. The clique-width of P_n is at most 3, and a 3-expression τ defining the $P_n = v_1 v_2 \dots v_n$, $n \geq 2$, with labels 1, 2, 3 can be obtained in time $O(n)$ as follows:

$\tau := \eta_{1,2}(1(v_1) \oplus 2(v_2))$
for $i := 3$ **to** n **do** $\tau := \rho_{3\to 2}(\rho_{2\to 1}(\eta_{3,2}(\tau \oplus 3(v_i))))$.

Therefore, since graphs of clique-width at most 2 are exactly the cographs, $cwd(P_n) = 3$ for $n \geq 4$. The clique-width of C_n is at most 4 and a 4-expression defining it can be constructed in time $O(n)$; see [55].

Complements of paths and cycles. The clique-width of $\overline{P_n}$, the complement graph of the path $P_n = v_1 v_2 \dots v_n$, $n \geq 2$, is at most 3, and a 3-expression τ defining