Partitioning Chordal Graphs into Independent Sets and Cliques

Pavol Hell

School of Computing Science, Simon Fraser University
Burnaby, B.C., Canada, V5A1S6
E-mail: `pavol@cs.sfu.ca`

Sulamita Klein [1]   Loana Tito Nogueira

COPPE-Sistemas, Universidade Federal do Rio de Janeiro
Caixa Postal 68511, 21945-970, Rio de Janeiro, RJ, Brasil
E-mail: {`sula, loana`}`@cos.ufrj.br`

Fábio Protti

Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro
Caixa Postal 2324, 20001-970, Rio de Janeiro, RJ, Brasil
E-mail: `fabiop@nce.ufrj.br`

Abstract

We consider the following generalization of split graphs: A graph is said to be a (k, ℓ)-graph if its vertex set can be partitioned into k independent sets and ℓ cliques. (Split graphs are obtained by setting $k = l = 1$). Much of the appeal of split graphs is due to the fact that they are chordal, a property not shared by (k, ℓ)-graphs in general. (For instance, being a $(k, 0)$-graph is equivalent to being k-colourable.) However, if we keep the assumption of chordality, nice algorithms and characterization theorems are possible. Indeed, our main result is a forbidden subgraph characterization of chordal (k, ℓ)-graphs. We also give an $O(n(m + n))$ recognition algorithm for chordal (k, ℓ)-graphs. When $k = 1$, our algorithm runs in time $O(m + n)$.

In particular, we obtain a new simple and efficient greedy algorithm for the recognition of split graphs, from which it is easy to derive the well known forbidden subgraph characterization of split graphs. The algorithm and the characterization extend, in a natural way, to the 'list' (or 'pre-colouring extension') version of the split partition problem - given a graph with some vertices pre-assigned to the independent set, or to the clique, is there a split partition extending this pre-assignment ?

Another way to think of our main result is the following min-max property of chordal graphs: the maximum number of independent (i.e., disjoint and nonadjacent) K_r's equals the minumum number of cliques that meet all K_r's.

Key words:
Chordal graphs, Split Graphs, Min-Max Theorems, Greedy Algorithms,
Pre-colouring Extension, List Partitions

---

1 Introduction

A graph G is a (k, ℓ)-graph [2] if its vertices can be partitioned into k independent sets and ℓ cliques. (A clique is a complete subgraph, not necessarily maximal.) Thus (k, ℓ)-graphs are a natural generalization of split graphs [10], which are precisely $(1, 1)$-graphs. Since split graphs are chordal [10], many basic optimization problems can be solved efficiently for them; they can also be efficiently recognized [10]. When k or ℓ is greater than one, there are (k, ℓ)-graphs which are not perfect (and hence not chordal). Still, in [3], $O((n+m)^2)$ recognition algorithms for $(2, 1)-$, $(1, 2)-$, and $(2, 2)$-graphs, are given. Polynomial algorithms for the recognition of these three classes of graphs also follow from more general algorithms for 'sparse-dense partition problems' of Feder *et al.* [4]. On the other hand, when $k \geq 3$ or $\ell \geq 3$, recognizing (k, ℓ)-graphs is easily seen to be an NP-complete problem [2]. (For instance, the class of $(k, 0)$-graphs is precisely the class of k-colourable graphs.)

We focus on the case of *chordal* (k, ℓ)-graphs, and give a forbidden subgraph characterization, and a polynomial time recognition algorithm. Specifically, we prove that a chordal graph is a (k, ℓ)-graph if and only if it does not have $\ell + 1$ independent copies of K_{k+1}. (A set of subgraphs is *independent* if they are pairwise disjoint and nonadjacent.)

A special case of this result, for chordal $(2, 1)$-graphs, was first reported in [8]. An extended abstract of the present paper has also appeared in [6].

An alternate view of our result states that the maximum number of independent K_r's in a chordal graph equals the minimum number of cliques that meet all K_r's. In other words, if we denote by $f(G, r)$ the maximum number of independent copies of K_r in G, and by $g(G, r)$ the minimum number of cliques of G which meet all K_r of G, then we show that for chordal graphs

[1] Partially supported by CNPq. Complete affiliation: Instituto de Matemática and COPPE-Sistemas.

$f(G,r) = g(G,r)$. (Note that when $r = 1$, $f(G,r)$ is the independence number of G, and $g(G,r)$ the clique covering number of G.) Our $O(n(m+n))$ algorithm identifies $f(G,r)$ independent K_r's and the same number of cliques that meet all K_r's.

Our recognition algorithm actually finds a minimum value of ℓ such that G is a (k,ℓ)-graph. The algorithm is more efficient when $k = 1$, i.e., when we seek a partition into *one* independent set and a set of cliques. When both k and ℓ are one, we specialize the algorithm to yield a new simple and efficient recognizition algorithm for split graphs. (Note that in this case we need no restriction to chordal graphs.) The value of the algorithm is underscored by the fact that it easily adapts to solve the list version of the split partition problem - finding an extension of a given pre-assignment of some vertices to the independent set, or clique. As a byproduct of the algorithm we also obtain a forbidden subgraph characterization of when such an extension is possible.

Let G be a graph. If $S, S' \subseteq V(G)$, we denote by $N_S(S')$ the *neighbourhood* of S' in S, i.e., the set of vertices of S which are either in S' or adjacent to a vertex of S'. Moreover, if $N_S(S') \neq \emptyset$, then we say that S and S' are *adjacent.*

We shall write $N_S(v)$ instead of $N_S(\{v\})$; note that this *neighbourhood of* v *in* S contains v if $v \in S$.

2 The Theorems

In this section we present our characterization of chordal (k,ℓ)-graphs in terms of forbidden subgraphs. The following lemmas will be useful:

Lemma 1 *Let C and C' be two cliques in a chordal graph G. Then some vertex of C' is adjacent to all the vertices of $N_C(C')$.*

Proof. We shall prove that, in fact, the neighborhoods of the vertices of C' in C are linearly ordered by inclusion. Suppose that two distinct vertices $v_1, v_2 \in C'$ have incomparable neighborhoods in C, i.e., that neither of the sets $N_C(v_1), N_C(v_2)$ contains the other. Then there exist distinct vertices $u_1, u_2 \in C$ such that u_1 is adjacent to v_1 but not to v_2, and u_2 is adjacent to v_2 but not to v_1. This is impossible, since u_1, u_2, v_2, v_1 would induce a chordless four-cycle. The lemma follows by considering the vertex $v \in C'$ with maximal $N_C(v)$. $\square$

Lemma 2 *Let C and K be two disjoint cliques of a chordal graph G. Then there exists a clique C' with the following property: C' intersects K, and it also intersects all the cliques adjacent to K which are intersected by C.*

Proof. Let $L = N_C(K)$. By Lemma 1 some vertex of K is adjacent to every vertex of L, and hence can be added to L to obtain a clique that intersects K, as well as all the cliques of G intersected by L. Consider now a clique K' of G which intersects C but is disjoint from L. It follows from the definition of L that such a clique does not intersect K. We need to consider such a K', if it contains a vertex a adjacent to K. Let A denote the set of all such vertices a, i.e., vertices which are adjacent to K and belong to some clique intersecting C but not L. We claim that each $a \in A$ is adjacent to all vertices of L. Indeed, if $b \in L$ is not adjacent to a, then there exist vertices $c \in C \setminus L$, and $s, t \in K$ (possibly $s = t$), such that b, c, a, s, t, b is a chordless cycle. Similarly, we claim that any two $a, a' \in A$ are adjacent in G. Otherwise, there exist vertices $c, c' \in C \setminus L$ (possibly $c = c'$), and $s, s' \in K$ (possibly $s = s'$), such that a, c, c', a', s', s, a is a chordless cycle in G. Thus, the set $L \cup A$ induces a clique, and Lemma 1 guarantees that there is a vertex $u \in K$ adjacent to all vertices of $L \cup A$. Now the clique induced by $L \cup A \cup \{u\}$ intersects K, and, by the definition of A, it also intersects all the cliques intersected by C which are adjacent to K. $\square$

Note that the lemma also holds when C and K are not disjoint (with $C' = C$).

Lemma 3 *Let $C_1, C_2, \ldots, C_p$ be a collection of pairwise adjacent cliques in a chordal graph G. Then there exists a clique C in G which intersects each $C_i, i = 1, 2, \ldots, p$.*

Proof. The result easily follows when $p \leq 2$. Assume now $p > 2$. By induction, there exists a clique C that intersects C_i for every $i \in \{1, \ldots, p-1\}$. If C intersects C_p, nothing remains to prove. Otherwise, apply Lemma 2 to C and C_p. $\square$

A simple necessary condition for a graph G to be a (k, ℓ)-graph is that it does not contain $\ell+1$ independent K_{k+1}'s. Indeed, consider any partition of such a G into k independent sets and ℓ cliques. Any K_{k+1} in G would have to contain a vertex from one of the cliques in the partition, and hence from amongst any $(\ell+1)$ such K_{k+1}'s some two must intersect the same clique, and thus have an edge joining them. (This means they were not independent.) It turns out that for chordal graphs the above condition is also sufficient. (Note that the condition simply says that $(\ell+1)K_{k+1}$ is not an induced subgraph of G.) We shall derive this fact from the following result:

Theorem 4 *Let G be a chordal graph, and let $r \geq 1$ be an integer. Then $f(G, r) = g(G, r)$.*

It is clear that $f(G, r) \leq g(G, r)$, for any G and any $r \geq 1$. In order to prove the equality for chordal graphs, we proceed as follows:

Let G be a graph. Let us define $\mathcal{K}^r(G)$ as the graph with a vertex corresponding to each K_r in G, and two vertices adjacent in $\mathcal{K}^r(G)$ if and only if the corresponding K_r's are not independent in G.

Lemma 5 *For any graph G, $f(G, r)$ is the independence number of $\mathcal{K}^r(G)$. For a chordal graph G, $g(G, r)$ is the clique covering number of $\mathcal{K}^r(G)$.*

Proof. The first statement is obvious. The second statement follows from the observation that we can modify any clique cover $\mathcal{C}$ of $\mathcal{K}^r(G)$, to construct a collection of (the same number of) cliques which meet all K_r's of G, by applying Lemma 3 to each clique in $\mathcal{C}$. $\square$

Lemma 6 *If G is chordal then $\mathcal{K}^r(G)$ is also chordal.*

Proof. Assume that $W_1, W_2, \ldots, W_q, W_1$ $(q \geq 4)$ is a chordless cycle in $\mathcal{K}^r(G)$. This means that W_i and W_j are consecutive in the cycle if and only if the corresponding K_r's in G are adjacent. Consider a sequence of vertices of G $S = (u_1, w_1, u_2, w_2, \ldots, u_q, w_q)$ such that $u_i, w_i \in W_i$, u_i is adjacent to the K_r corresponding to W_{i-1}, and w_i is adjacent to the K_r corresponding to W_{i+1} , for every $i \in \{1, \ldots, q\}$ (indices are taken circularly in the range $1 \ldots q$). It is clear that if i and j are non-consecutive indices, then the subsets $\{u_i, w_i\}$ and $\{u_j, w_j\}$ are non-adjacent. Occasionaly, it might occur that $u_i = w_i$ or $w_i = u_{i+1}$ for some i, but these equalities cannot hold simultaneously. This means that every vertex occurring in S appears at most twice, and two ocurrences of a same vertex necessarily use consecutive positions in S. These observations show that we can construct a cycle C_0 in G from S by removing repeated ocurrences of vertices. This construction ensures that at least one vertex from $\{u_i, w_i\}$ is taken, for every $i \in \{1, \ldots, q\}$. Thus, C_0 contains at least four vertices. Moreover, C_0 is clearly a chordless cycle, a contradiction. $\square$

Theorem 4 follows naturally from Lemmas 5 and 6.

Proof of Theorem 4. By Lemma 6 $\mathcal{K}^r(G)$ is chordal, and therefore perfect. Thus the independence number of $\mathcal{K}^r(G)$ is equal to its clique covering number. Lemma 5 completes the proof. $\square$

The characterization of chordal (k, ℓ)-graphs by forbidden subgraphs follows as a consequence of the previous theorem.

Theorem 7 *A chordal graph is a (k, ℓ)-graph if and only if it does not contain $(\ell + 1)K_{k+1}$ as an induced subgraph.*

Proof. We have shown that a chordal (k, ℓ)-graph cannot contain $\ell + 1$ independent copies of K_{k+1}, i.e., cannot contain $(\ell + 1)K_{k+1}$ as an induced subgraph. On the other hand, Theorem 4 implies that if a chordal graph G

does not contain $\ell+1$ independent copies of K_{k+1}, then $g(G, k+1) \leq \ell$. This means that G contains ℓ cliques whose removal leaves a subgraph G' without K_{k+1}. Since G is perfect, G' is k-colourable, whence G admits a partition into k independent sets and ℓ cliques. $\square$

3 The Algorithms

Since k and ℓ are fixed, there are only polynomially many subgraphs of G with $(\ell+1)(k+1)$ vertices, and so Theorem 7 gives a polynomial time recognition algorithm for chordal (k, ℓ)-graphs. There are, however, more efficient algorithms. The $O(n(m+n))$ algorithm we present below also provides us with a second proof of Theorem 4.

We first review the standard greedy colouring algorithm for chordal graphs. (Note that testing for the existence of a k-colouring is equivalent to recognizing $(k, 0)$-graphs.) Suppose the vertices of G are given in a perfect elimination ordering $1, 2, \ldots, n$, [10]. The *Reverse Greedy Algorithm* proceeds in the order $n, n-1, \ldots, 2, 1$, assigning to each vertex the least available colour. In other words, to colour G by the colours $s_1, s_2, \ldots$, we colour the vertex n by s_1, and having coloured $n, n-1, \ldots, i+1$, we colour i by s_d where d is the smallest subscript such that no neighbour of i amongst $i+1, i+2, \ldots, n$ has been coloured s_d. Note that at this point i lies in a K_d, since it has a neighbour of each of the colours $s_1, s_2, \ldots, s_{d-1}$, which are mutually adjacent. (Any two neighbours of i amongst $i+1, i+2, \ldots, n$ are adjacent, since $1, 2, \ldots, n$ is a perfect elimination ordering.) It follows that the Reverse Greedy Algorithm delivers, in time $O(m+n)$, both a minimum colouring and a maximum clique.

We are now ready to describe our algorithm. Let $k > 0$ be an integer. The algorithm finds the minimum value of ℓ (possibly $\ell = 0$) for which G is a (k, ℓ)-graph. We will be colouring the vertices of the input chordal graph G by the colours $s_1, s_2, \ldots, s_k$ and $c_1, c_2, \ldots, c_\ell$. Throughout the execution of the algorithm, the vertices coloured by each s_d will form an independent set, and the vertices coloured by each c_a will form a clique. We shall denote by S_i the set consisting of i, together with all vertices amongst $1, 2, \ldots, i-1$ coloured $s_1, s_2, \ldots, s_k$.

The following fact is easily obtained from these definitions, using the properties of a perfect elimination ordering:

Lemma 8 *If vertex i is adjacent to the first vertex j coloured by c_a, and $j < i$, then i is adjacent to all vertices $x < i$ coloured by c_a.*

The lemma will allow us to easily test whether or not a vertex i can be added to the clique formed by vertices coloured by c_a.

Algorithm for the Recognition of Chordal (k, ℓ)-Graphs

Assume G is a chordal graph with a perfect elimination ordering $1, 2, \ldots, n$.

- Colour the vertex 1 by s_1.
- Having coloured the vertices $1, 2, \ldots, i-1$ without using the colour c_1:
  · remove the colours from $1, 2, \ldots, i-1$ and colour $1, 2, \ldots, i$ by colours $s_1, s_2, \ldots, s_k$ (using the Reverse Greedy Algorithm), if possible, or else
  · keep the colouring of $1, 2, \ldots, i-1$, and colour i by c_1.
- Having coloured the vertices $1, 2, \ldots, i-1$ and having used the colours $c_1, c_2, \ldots, c_a$:
  · colour i by c_b, where $b \leq a$ is the least subscript such that i is adjacent to the first vertex coloured c_b, if such subscripts exists, or else
  · remove the colours from the vertices of $S_i \setminus i$ and colour S_i by colours $s_1, s_2, \ldots s_k$ (using the Reverse Greedy Algorithm), if possible, or else
  · keep the colouring of $1, 2, \ldots, i-1$, and colour i by c_{a+1}.

We set ℓ to be the largest value of a such that there is a vertex coloured c_a, or $\ell = 0$ if all vertices are coloured with $s_1, s_2, \ldots, s_k$.

Since the work of the algorithm is dominated by the at most n applications of the Reverse Greedy Algorithm, the time bound $O(n(m+n))$ follows. The correctness will follow from the next proposition:

Proposition 9 *If the algorithm uses colour c_p, then G contains an induced pK_{k+1}.*

Proof: Let v_a be the first vertex (in the perfect elimination ordering) using the colour c_a. The subgraph of G induced by $S_{v_a - 1}$ is k-coloured, but our algorithm found it impossible to add v_a so that S_{v_a} is still k-colourable. Thus there exists a subgraph X_a isomorphic to K_{k+1}, containing v_a and some k vertices of $S_{v_a - 1}$. It only remains to show that the subgraphs $X_1, X_2, \ldots X_p$ are independent. Suppose a vertex x from a subgraph X_a is adjacent or equal to a vertex x' from a subgraph $X_{a'}$. Assume $a < a'$.

If $x' \leq x$, then from the fact that x' is adjacent or equal to $v_{a'}$, we conclude that x is adjacent or equal to $v_{a'}$. Now v_a and $v_{a'}$ must be adjacent, since $x \leq v_a$ and $x \leq v_{a'}$. This means that $v_{a'}$ is adjacent to the first vertex coloured by c_a, contradicting the fact that our algorithm could not colour $v_{a'}$ by c_a.

If $x' > x$, then x' is seen to be adjacent or equal to v_a by a similar argument. If $x' \leq v_a$ then $v_{a'}$ and v_a must be adjacent, and $v_{a'}$ should have been coloured by c_a, as in the previous case. On the other hand, if $x' > v_a$, then x' is adjacent to the first vertex coloured c_a. Thus our algorithm should have coloured x'

by c_a, contradicting the fact that it was coloured by some s_d (in case x' is a member of $S_{v_{d'}-1}$) or by $c_{d'}$ (in case $x' = v_{d'}$). $\square$

Corollary 10 *The following statements are equivalent:*

1. *The algorithm partitions G into k independent sets and ℓ cliques*
2. *the graph G is a (k, ℓ)-graph*
3. *the graph G does not contain an induced $(\ell + 1)K_{k+1}$.*

Proof: The implications 1 implies 2, and 2 implies 3 are obvious, and Proposition 9 proves that 3 implies 1. $\square$

Note that the equivalence of 1 and 2 proves the correctness of the algorithm, while the equivalence of 1 and 3 provides us with a second proof of theorem 7.

We close this section by noting that the algorithm finds, for any k and chordal graph G, the minimum value ℓ such that G is a (k, ℓ)-graph.

4 The Case of One Independent Set, Emphasizing Split Graphs

When $k = 1$, we can somewhat simplify the algorithm, since we do not need the Reverse Greedy Algorithm to test whether a vertex can be added to an independent set, maintaining independence.

Algorithm for the Recognition of Chordal $(1, \ell)$-Graphs

Assume G is a chordal graph with a perfect elimination ordering $1, 2, \ldots, n$.

- Colour the vertex 1 by s_1,
- and continue colouring vertices $i = 2, 3, \ldots$ by s_1 as long as possible (i has no edges to $1, 2, \ldots, i - 1$),
- and then colour the first j that cannot be so coloured by c_1.
- Having coloured the vertices $1, 2, \ldots, i - 1$ using colours $s_1, c_1, c_2, \ldots, c_a$,
  - · colour i by c_b, where $b \leq a$ is the first subscript such that i is adjacent to the first vertex coloured c_b, if such subscripts exists, or
  - · colour i by s_1 if it is nonadjacent to all vertices coloured s_1, or else
  - · colour i by c_{a+1}.

It is clear that this algorithm can be implemented to run in time $O(m + n)$.

The situation is simplest when $k = \ell = 1$, and in this case we don't need to explicitely assume chordality. (Split graphs are automatically chordal.) Since split graphs are of some interest [10,5], we restate the algorithm once more, as it applies to the recognition of split graphs:

Algorithm for the Recognition of Split Graphs

Assume G is any graph.

- Find a perfect elimination ordering $1, 2, \ldots, n$ of G [10].
- Colour 1 by s, and continue colouring $i = 2, 3, \ldots$ by s as long as possible (i is not adjacent to a previously coloured vertex), then introduce colour c for the next vertex j.
- If all of $1, 2, \ldots i-1$ have been coloured, and both colours s and c have been used, then colour i by c if it is adjacent to the first vertex j coloured by c; otherwise, colour i by s if it is nonadjacent to all vertices previously coloured by s.

If the algorithm fails because there is no perfect elimination ordering, then the algorithm given in [10] exhibits an induced C_4, C_5, or $C_k, k \geq 6$. If it fails to colour all vertices, then according to Proposition 9, G contains an induced $2K_2$. Here is a short version of the proof, that will be used in a generalization below:

If a vertex i is reached which cannot be coloured by s or by c, then i is nonadjacent to the first vertex j coloured by c, and is adjacent to some vertex k that was previously coloured by s. We claim that j and k cannot be adjacent. If $k < j$, this follows from the properties of a perfect elimination ordering. If $k > j$, then if k were adjacent to j the algorithm would have coloured it by c.

Vertex j was coloured by c because it was adjacent to a vertex ℓ previously coloured by s. Since $\ell < j < i$, and i, j are nonadjcent, ℓ must be nonadjacent to i. Addiditonally, the two vertices k, ℓ are nonadjacent, as they are both coloured by s. Thus i, j, k, ℓ form an induced $2K_2$ in G.

Since each $C_k, k \geq 6$ also contains an induced $2K_2$, we obtain the following well known characterization of split graphs [10]:

Corollary 11 *A graph G is a split graph if and only if it does not contain an induced $2K_2, C_4$ or C_5.*

5 Pre-colouring Extension

In [4] we introduced the notion of a list partition. In the context of split graphs, it specializes to the following concept of a 'pre-colouring extension'.

(Pre-colouring extensions for ordinary colourings have been much studied in the literature, cf. [1,7].)

A *pre-coloured graph* G is a graph with some vertices coloured by either s or c, so that every two vertices coloured c are adjacent in G, but no two vertices coloured s are adjacent in G. A *split extension* of a pre-coloured graph G is a partition of $V(G)$ into an independent set containing all vertices coloured by s, and a clique containing all vertices coloured by c.

Consider the pre-coloured graph A consisting of three vertices a, a', a'' and two edges aa', aa'', with vertex a pre-coloured s. It is clear that A does not admit a split extension, as the nonadjacent vertices a' and a'' would both have to be in the clique. Similarly, the pre-coloured graph B with three vertices b, b', b'' and one edge $b'b''$ in which b is pre-coloured c, does not admit a split extension.

Theorem 12 *A pre-coloured graph G admits a split extension if and only if it does not contain an induced $2K_2, C_4, C_5, A$, or B.*

The proof follows again from the following modification of the above algorithm:

Algorithm for Split Extension of Pre-coloured Graphs

- Proceed as above, obtaining a perfect elimination ordering $1, 2, \ldots, n$ of G, then colouring $i = 1, 2, 3, \ldots$ by s as long as possible. Let j be the first vertex where this is impossible. This may be because
  - (as above) j is adjacent to a vertex previously coloured by s, but also because
  - j is adjacent to a vertex (occuring later in the ordering) which was pre-coloured by s, or also because
  - j itself has been pre-coloured by c.

  In all these cases, colour j by c.
- In the general step, if all of $1, 2, \ldots i-1$ have been coloured, and both colours s and c have been used, then
  - colour i by c if i is adjacent to j and is not pre-coloured by s, or if i is pre-coloured by c, otherwise
  - colour i by s if i is nonadjacent to all vertices coloured or pre-coloured by s and is not pre-coloured by c, or if i is pre-coloured by s.

The modified algorithm is analyzed in the same way as the earlier algorithm. If it fails to find a perfect elimination ordering then G contains an induced C_4, C_5, or $2K_2$. Otherwise, it only fails when a vertex i is reached which cannot be coloured by s or by c. (Thus i is not pre-coloured.)

If i cannot be coloured by c because it is nonadjacent to the first vertex j coloured by c, and cannot be coloured by s because it is adjacent to some vertex k previously coloured by s, then, as above, j and k must be nonadjacent.

If j was coloured by c because it was adjacent to a vertex ℓ previously coloured by s, we conclude as above that i, j, k, ℓ form an induced $2K_2$ in G. If j was coloured by c because it was adjacent to a vertex ℓ pre-coloured by s, then ℓ and k are still nonadjacent. If ℓ is also nonadjacent to i, we have a $2K_2$ as before. Otherwise we have an induced copy of A, with $a = \ell, a' = j, a'' = i$. Finally, if j was pre-coloured by c, then we have an induced copy of B, with $b = j, b' = i, b'' = k$.

If i cannot be coloured by c because it is nonadjacent to the first vertex j coloured by c, but cannot be coloured by s because it is adjacent to some vertex k pre-coloured by s, then k, i, j form a copy of A if j is adjacent to k. Otherwise (j and k are nonadjacent), we argue about ℓ as in the previous case.

On the other hand, if i cannot be coloured by s because it is adjacent to a vertex k previously coloured by s, but cannot be coloured by c because it is nonadjacent to a vertex j pre-coloured by c, then we may assume that $j > i$, otherwise the proof still applies. In this case j, k must not be adjacent, and so j, i, k form a copy of B.

Finally, if i is adjacent to a vertex k pre-coloured by s and nonadjacent to a vertex j pre-coloured by c, then we have an induced A if j, k are adjacent, or an induced B if j, k are nonadjacent. $\square$

ADDED IN PROOF

We have improved the algorithms for the recognition of chordal (k, ℓ)-graphs and for computing $f(G, r)$ for chordal graphs to time $O(m+n)$; these improved algorithms, together with an extension of the min-max relation to weighted chordal graphs will appear in our paper "Packing r-cliques in chordal graphs".

References

[1] M.O. Albertson and E.H. Moore. Extending graph colorings. *J. Combinatorial Theory* B 77 (1999) 83–95.

[2] A. Brandstädt. Partitions of graphs into one or two independent sets and cliques. *Discrete Mathematics* 152 (1996) 47–54.

[3] A. Brandstädt. The complexity of some problems related to graph 3-colorability. *Discrete Applied Mathematics* 89 (1998) 59–73.

[4] T. Feder, P. Hell, S. Klein, and R. Motwani. Complexity of graph partition problems. In F. W. Thatcher and R. E. Miller, eds., *Proceedings of the 31st Annual ACM Symposium on Theory of Computing - STOC'99*, pages 464–472. Plenum Press, New York, 1999.

[5] S. Foldes and P. Hammer, Split Graphs. *Proc. 8th Southeastern Conf. on Combinatorics, Graph Theory and Computing* (F. Hoffman *et al.*, eds.). Louisiana State Univ., Baton Rouge, Louisiana, 311-315.

[6] P. Hell, S. Klein, L. T. Nogueira, and F. Protti. On generalized split graphs. GRACO'2001, *Eletronic Notes in Discrete Mathematics*, Vol. 7 (2001), Elsevier.

[7] J. Kratochvíl and A. Sebö. Coloring precolored perfect graphs. *J. Graph Theory* 25 (1997) 207–215.

[8] L. T. Nogueira. Grafos Split e Grafos Split Generalizados. Master Thesis, COPPE-Sistemas, Universidade Federal do Rio de Janeiro, Brazil, 1999. (In Portuguese)

[9] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science* 1 (1976) 237–267.

[10] M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs.* Academic Press, New York, 1980.