Greedy algorithms, H-colourings and a complexity-theoretic dichotomy

Antonio Puricella and Iain A. Stewart,
Department of Mathematics and Computer Science,
University of Leicester, Leicester LE1 7RH, U.K.

Abstract

Let H be a fixed undirected graph. An H-colouring of an undirected graph G is a homomorphism from G to H. If the vertices of G are partially ordered then there is a generic non-deterministic greedy algorithm which computes all lexicographically first maximal H-colourable subgraphs of G. We show that the complexity of deciding whether a given vertex of G is in a lexicographically first maximal H-colourable subgraph of G is **NP**-complete, if H is bipartite, and $\mathbf{\Sigma}_2^p$-complete, if H is non-bipartite. This result complements Hell and Nešetřil's seminal dichotomy result that the standard H-colouring problem is in **P**, if H is bipartite, and **NP**-complete, if H is non-bipartite. Our proofs use the basic techniques established by Hell and Nešetřil, combinatorially adapted to our scenario.

1 Introduction

In what is now a seminal result, Hell and Nešetřil [6] established a dichotomy for the H-colouring problem when H is an undirected graph: the H-colouring problem is in **P**, if H is bipartite, and is **NP**-complete otherwise. Such a (dichotomy) result can also be thought of as a generic result in that it provides a complete, exact classification of the computational complexities of an infinite class of problems (in this case, the class of H-colouring problems). Other such generic results exist. For example, Miyano [8] proved a very general result relating to hereditary properties of graphs: he showed that the problem of deciding whether a given vertex of a given undirected graph G, whose vertices are linearly ordered, lies in the lexicographically first maximal subgraph of G satisfying some fixed polynomial-time testable, non-trivial, hereditary property π is **P**-complete. (Notice that the existence of an H-colouring of an undirected graph G, *i.e.*, a homomorphism from G to H, is a particular hereditary property of G.)

A number of other dichotomy results (involving unequivocal complexity-theoretic classifications) and generic results (applicable to an infinite class of problems) have since been obtained. Examples of other dichotomy results include: Feder and Hell's result [4] that the list homomorphism problem for reflexive graphs is solvable in polynomial-time if the target graph is an interval graph, and **NP**-complete otherwise; Feder, Hell and Huang's [5] result that the list homomorphism problem for irreflexive graphs is solvable in polynomial-time if the complement of the target graph is a circular arc graph of clique covering number two, and **NP**-complete otherwise; Díaz, Serna and Thilikos's result [2] that the complexity of the list (H, C, K)-colouring problem mirrors that of the list homomorphism problem; and Dyer and Greenhill's result [3] that the problem of counting the H-colourings of a graph is solvable in polynomial-time if every connected component of H is a complete reflexive graph with all loops present or a complete bipartite irreflexive graph (with no loops present), and $\sharp\mathbf{P}$-complete otherwise. Examples of other generic results include: Miyano's result [9] that the problem of deciding whether a given vertex of a given undirected graph G, whose vertices are linearly ordered, lies in the lexicographically first maximal connected subgraph of G satisfying some fixed polynomial-time testable, hereditary property π that is determined by the blocks and non-trivial on connected graphs is $\mathbf{\Delta}_2^p$-complete; and Puricella and Stewart's result [11] that the problem of deciding whether a given vertex of a given undirected graph G, whose vertices are partially ordered, lies in a lexicographically first maximal subgraph of G satisfying some fixed polynomial-time testable, non-trivial, hereditary property π is **NP**-complete.

Dichotomy and generic results such as those highlighted above are particularly attractive as they give a concise and simplified view of a parameterized world of natural problems. In this paper, we consider the problem of deciding whether a given vertex of a given undirected graph G, whose vertices are partially ordered, lies in a lexicographically first maximal H-colourable subgraph of G (where the undirected graph H is fixed). In particular, we prove that this problem is **NP**-complete, if H is bipartite, and $\mathbf{\Sigma}_2^p$-complete, if H is non-bipartite; thus establishing yet another complexity-theoretic dichotomy result. Our proofs use the techniques established by Hell and Nešetřil in [6] although they are combinatorially adapted according to our circumstances. However, part of Hell and Nešetřil's constructions can be applied verbatim and this substantially shortens our exposition.

2 Basic definitions

For standard graph-theoretic definitions the reader is referred to [1], and for standard complexity-theoretic definitions to [10].

Let $G = (V, E)$ be an undirected graph and suppose that the vertices of V are linearly ordered. Given a subset $S = \{s_0, s_1, s_2, \ldots, s_k\}$ of V, where the induced ordering is $s_0 < s_1 < \ldots < s_k$, we can define a *lexicographic order* on the set of all subsets of S as follows (we call it lexicographic because we consider $s_0, s_1, \ldots, s_k$ to be our alphabet):

- for subsets $U = \{u_1, u_2, \ldots, u_p\}$ and $W = \{w_1, w_2, \ldots, w_k\}$ of S, where $u_1 < u_2 < \ldots < u_p$ and $w_1 < w_2 < \ldots < w_k$, we say that U is *lexicographically smaller than* W if:
  - there is a number t, where $1 \leq t \leq p$, such that $u_t < w_t$ and $u_i = w_i$, for all i such that $1 \leq i < t$; or
  - $k > p$ and $u_i = w_i$, for all i such that $1 \leq i \leq p$.

Let π be some property of graphs (our graphs are all undirected). If we take $S = V$ then we can talk about the *lexicographically first maximal subgraph* of G that satisfies π (as Miyano does in [8]).

Now let $G = (V, E)$ be an undirected graph, let P be a partial order on V and let $s \in V$. We assume that the partial order P is given in the form of an acyclic digraph detailing the immediate predecessors, *i.e.*, the *parents*, and the immediate successors, *i.e.*, the *children*, of each vertex. We think of a partial order P as encoding a collection of linear orders of the form $s = s_0 < s_1 < s_2 < \ldots < s_k$, where s_{j+1} is a child of s_j, for $0 \leq j < k$, and s_k has no children. Note that a partial order can encode an exponential number of linear orders.

Let π be some property of graphs. Now we can talk of the *lexicographically first maximal subgraphs* of G satisfying π; where we get one such subgraph for every linear order encoded within P. A property π on graphs is *hereditary* if whenever we have a graph with the property π then the deletion of any vertex and its incident edges does not produce a graph violating π, *i.e.*, π is preserved by vertex-induced subgraphs. It is straightforward to see that the sets of vertices that induce these lexicographically first maximal subgraphs of G satisfying some hereditary property π can be obtained using the following non-deterministic algorithm GREEDY(π) (if P is a linear order then this algorithm computes the lexicographically first maximal subgraph of G satisfying π). The algorithm GREEDY(π) takes as input 3 arguments: an undirected graph $G = (V, E)$, a directed acyclic graph $P = (V, D)$ and a specified vertex $s \in V$; and is as follows:

```
input(G,P,s)
  S := ∅
  current-vertex := s
  if π(S∪{current-vertex},G) then                       (*)
    S := S∪{current-vertex}
    fi
  while current-vertex has at least one child in P do
    current-vertex := a child of current-vertex in P
    if π(S∪{current-vertex},G) then                     (**)
      S := S∪{current-vertex}
      fi
    od
output(S)
```

where $\pi(S \cup \{current\text{-}vertex\}, G)$ is a predicate evaluating to 'true' if, and only if, the subgraph of G induced by the vertices of $S \cup \{current\text{-}vertex\}$ satisfies π. We say that a vertex v is the *current-vertex* if we have 'frozen' an execution of the algorithm GREEDY(π) immediately prior to executing either line ($*$) or line ($**$) and the value of the variable *current-vertex* at this point is v.

A property π is called *non-trivial* on a class of graphs if there are infinitely many graphs from this class satisfying π but π is not satisfied by all graphs of the class.

Let $\mathcal{C}$ be a class of graphs and let π be some property of graphs. The problem GREEDY(partial order, $\mathcal{C}$, π) has: as its instances tuples (G, P, s, x), where G is a graph from $\mathcal{C}$, P is a partial order of the vertices of G and s and x are vertices of G; and as its yes-instances those instances for which there exists an execution of the algorithm GREEDY(π) on input (G, P, s) resulting in the output of a set of vertices containing the vertex x. The problem GREEDY(linear order, $\mathcal{C}$, π) is defined similarly except that P is a linear order. As mentioned earlier, when π is polynomial-time testable, non-trivial and hereditary, Miyano [8] proved that GREEDY(linear order, undirected graphs, π) is **P**-complete, and Puricella and Stewart [11] proved that GREEDY(partial order, undirected graphs, π) is **NP**-complete.

Let G and H be graphs. A *homomorphism from G to H* is a map f from the vertices of G to the vertices of H such that if (u, v) is an edge of G then $(f(u), f(v))$ is an edge of H. The *H-colouring problem* is the problem whose instances are graphs G and whose yes-instances are those graphs G for which there is a homomorphism from G to H.

If U is a subset of vertices of the graph G then $\langle U \rangle_G$ is the subgraph of G induced by the set of vertices U. A graph is 3-*colourable* if the vertices can be coloured with a unique colour from red, white and blue so that two adjacent vertices are coloured differently; and the 3-*colouring problem* has as an instance a graph G and as a yes-instance a graph G that is 3-colourable.

3 A complete problem

Our proof of our main result in the next section follows the strategy adopted by Hell and Nešetřil. Essentially, we assume that H is a non-bipartite graph for which the problem GREEDY(partial order, undirected graphs, H-colouring) is not $\mathbf{\Sigma}_2^p$-complete and apply a sequence of constructions to yield that a known $\mathbf{\Sigma}_2^p$-complete problem is not complete, thereby obtaining a contradiction. Our 'known' problem $\mathbf{\Sigma}_2^p$-complete is GREEDY(partial order, undirected graphs, 3-colourable).

Theorem 1 *The problem GREEDY(partial order, undirected graphs,* 3*-colourable) is $\mathbf{\Sigma}_2^p$-complete.*

Proof Throughout this proof, the problem GREEDY(partial order, undirected graphs, 3-colourable) shall be denoted $\mathcal{G}$. We shall prove completeness by reducing from the problem NOT CERTAIN 3-COLOURING OF BOOLEAN EDGE-LABELLED GRAPHS, henceforth to be abbreviated as problem $\mathcal{N}$. An instance of $\mathcal{N}$ of size n consists of an undirected graph H on n vertices, some of whose edges are labelled with the disjunction of two (possibly identical) literals over the set of Boolean variables $\{X_{i,j} : i,j = 1,2,\ldots,n\}$ (the same literal may appear in more than one disjunction). A truth assignment t on the Boolean variables of $\{X_{i,j} : i,j = 1,2,\ldots,n\}$ makes some of the labels on the edges of H true and some false. Form the graph $t(H)$ by retaining the edges labelled true, as well as any unlabelled edges, and dispensing with the edges labelled false. A yes-instance is an instance H for which there exists a truth assignment t resulting in a graph $t(H)$ that cannot be 3-coloured. This problem was proven to be $\mathbf{\Sigma}_2^p$-complete in [12].

Given an instance H of the problem $\mathcal{N}$, we shall construct an instance (G,P,s,x) of the problem $\mathcal{G}$ where G is an undirected graph, P is a partial order on these same vertices and s and x are two distinguished vertices. Moreover, H will be a yes-instance of $\mathcal{N}$ if, and only if, (G,P,s,x) is a yes-instance of $\mathcal{G}$; and the construction will be such that it can be completed using logspace.

Let $H = (U,F)$ and suppose that $U = \{1,2,\ldots,n\}$. We build the undirected graph G from H as follows.

(a) For each vertex $i \in U$, 'attach' a copy of K_4 by identifying vertex i with one of the vertices of the clique. Denote the other three vertices by a_i, b_i^1 and b_i^2. We refer to the original vertices of U as *H-vertices*, the vertices of $\{a_i : i = 1,2,\ldots,n\}$ as *a-vertices* and the vertices of $\{b_i^1, b_i^2 : i = 1,2,\ldots,n\}$ as *b-vertices*.

(b) Retain any unlabelled edge (i,j) of F (between H-vertices i and j).

(c) For any labelled edge (i,j) of F (between H-vertices i and j), where $i < j$ and where the label is $L_{i,j}^1 \vee L_{i,j}^2$, replace the edge with a copy of the graph G_1 shown in Fig. 1. We use, for example, $L_{i,j}^1$ to refer to the first literal labelling edge (i,j) and also a vertex within a graph G_1: this causes no confusion. The vertices of $\{L_{i,j}^1, L_{i,j}^2, \bar{L}_{i,j}^1, \bar{L}_{i,j}^2 : (i,j) \in F, \text{ where } i < j\}$ are called *L-vertices*. Every L-vertex of any G_1 has an associated literal, *e.g.*, if the literal $L_{4,6}^1 = \neg X_{3,2}$ then the associated literal of vertex $L_{4,6}^1$ is $\neg X_{3,2}$ and the associated literal of vertex $\bar{L}_{4,6}^1$ is $X_{3,2}$. So, an L-vertex of some G_1 might have the same associated literal as an L-vertex of some other G_1. Finally, the vertices of $\{c_{i,j} : i,j = 1,2,\ldots,n\}$ are called *c-vertices*, the vertices of $\{d_{i,j} : i,j = 1,2,\ldots,n\}$ are called *d-vertices* and the vertices of $\{e_{i,j}^1, e_{i,j}^2 : i,j = 1,2,\ldots,n\}$ are called *e-vertices*.

(d) Include a disjoint copy of K_4, whose vertices are $\{y,z,w,x\}$ and join vertices y, z and w to every a-vertex. Include the vertex s as an independent vertex.

Our partial ordering P is defined as follows. First, order the Boolean variables $\{X_{i,j} : i,j = 1,2,\ldots,n\}$ lexicographically as

$$X_{1,1}, X_{1,2}, X_{1,3}, \ldots, X_{1,n}, X_{2,1}, X_{2,2}, \ldots, X_{n,n}$$

and denote this ordering by $<_X$; so $X_{1,1} <_X X_{1,2} <_X X_{1,3} <_X \ldots$. Next, consider the L-vertices. We obtain the notions of a positive L-vertex, where the vertex has an associated positive literal, and a negative L-vertex, where the vertex has an associated negative literal. Order the positive L-vertices so that if vertex λ_i is less than vertex λ_j in this ordering then the associated literal of λ_i is less than or equal to the associated literal of λ_j with respect to the ordering $<_X$ (note that there may be a number of such orderings on the positive L-vertices: it does not matter which of them we use). We obtain an analogous ordering of the negative L-vertices by taking complements (note that for every positive L-vertex $L^m_{i,j}$ or $\bar{L}^m_{i,j}$ with label l, the vertex $\bar{L}^m_{i,j}$ or $L^m_{i,j}$, respectively, is a negative L-vertex with label $\neg l$; and vice versa). As we walk down these two orderings in a synchronous fashion, the pairs of L-vertices are always complementary as is the pair of associated literals. Denote these orderings as

$$\lambda_1 < \lambda_2 < \ldots < \lambda_k \text{ and } \mu_1 < \mu_2 < \ldots < \mu_k,$$

respectively, where $\{\lambda_i, \mu_i : i = 1, 2, \ldots, k\} = \{L^1_{i,j}, L^2_{i,j}, \bar{L}^1_{i,j}, \bar{L}^2_{i,j} : (i,j) \in F, \text{ where } i < j\}$.

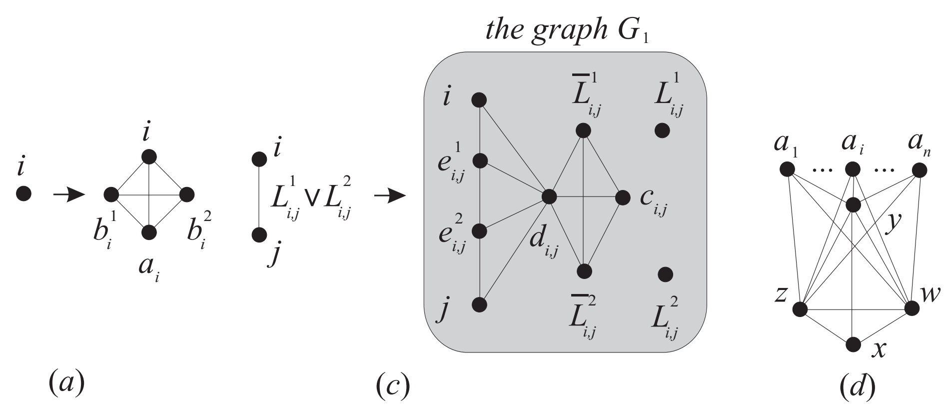

Figure 1. Phases (a), (c) and (d) of constructing G from H.

Our partial ordering P begins as follows. The vertex s is less than both λ_1 and μ_1; and then we have the orderings $\lambda_1 < \lambda_2 < \ldots < \lambda_k$ and $\mu_1 < \mu_2 < \ldots < \mu_k$. Also, for any index $i \in \{1, 2, \ldots, k-1\}$, if the associated literal of λ_i is different from the associated literal of λ_{i+1} then additionally $\lambda_i < \mu_{i+1}$ and $\mu_i < \lambda_{i+1}$. In order to complete P, choose any linear ordering of the c-vertices, followed by any linear ordering of the d-vertices, followed by any linear ordering of the e-vertices, followed by the ordering $1, 2, \ldots, n$ of the H-vertices, followed by any linear ordering of the b-vertices, followed by any linear ordering of the a-vertices, followed by the ordering w, y, z, x; and additionally define that both λ_k and μ_k are less than the least c-vertex (if there are no L-vertices then just concatenate the linear ordering of the c-vertices after the vertex s).

The construction of (G, P, s, x) from H is illustrated in Fig. 2 (note that to avoid cluttering the figure, not all vertices are named; and the bold edges correspond to the structure of H). Clearly, this construction can be completed using logspace.

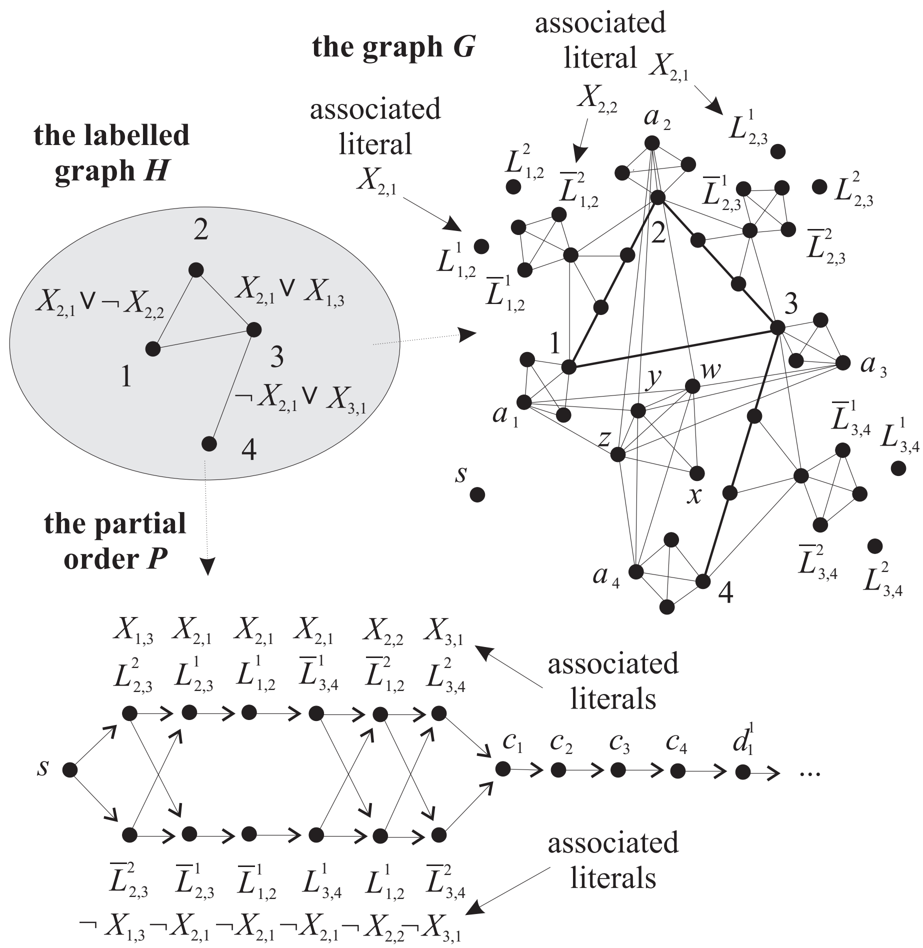

Figure 2. The construction of (G, P, s, x) from H.

Suppose that H is a yes-instance of problem $\mathcal{N}$. Hence, there exists a truth assignment t such that $t(H)$ is not 3-colourable. Consider the execution of the algorithm GREEDY(3-colourable) on (G, P, s, x) where the chosen linear order in P is that induced by the truth assignment t; that is, an L-vertex is chosen if, and only if, its associated Boolean literal is set at true by t. The first point to note is that s and every L-vertex chosen is output by GREEDY(3-colourable), as is every c-vertex. Let us freeze the execution at this point. Note that if the truth assignment t makes the label of some edge (i, j) of F true then at our freeze-point, the vertex $d_{i,j}$ is adjacent to at most 2 vertices of S, and so this vertex $d_{i,j}$ is subsequently output by GREEDY(3-colourable).

Conversely, if the truth assignment t makes the label of some edge (i, j) of F false then at our freeze-point, the vertex $d_{i,j}$ is adjacent to 3 mutually adjacent vertices of S and so this vertex $d_{i,j}$ is not subsequently output by GREEDY(3-colourable).

Unroll the execution of GREEDY(3-colourable) until every d-vertex and e-vertex has been considered. Note that every e-vertex is output regardless. Let us freeze the execution for a second time at this point.

Our next task in the execution is to consider the H-vertices as to whether they are output or not. Let (i, j) be some edge of F which is either unlabelled or whose label has been made true by t. It may or may not be the case that the vertices i and j are output; but if they are both output then at the point after the second of these vertices is output, the subgraph induced by the vertices of S can be 3-coloured but not so that i and j have the same colour. This is so because each of the vertices $d_{i,j}$, $e^1_{i,j}$ and $e^2_{i,j}$ is in S. Hence, as we know that $t(H)$ cannot be 3-coloured, there must be some H-vertex that is not output; and, consequently, there is at least one a-vertex output. Having an a-vertex output means that not all of $\{y, z, w\}$ are output which in turn means that x is output. Hence, (G, P, s, x) is a yes-instance of problem $\mathcal{G}$.

Conversely, suppose that (G, P, s, x) is a yes-instance of problem $\mathcal{G}$. Fix an accepting execution of the algorithm GREEDY(3-colourable) on input (G, P, s, x) and denote the linear order chosen within P by π. This execution gives rise to a truth assignment t on the literals labelling the edges of the graph H: if π is such that a positive L-vertex, with associated literal $X_{i,j}$, say, is chosen then set $t(X_{i,j})$ to be true; and if π is such that a negative L-vertex, with associated literal $\neg X_{i,j}$, say, is chosen then set $t(X_{i,j})$ to be false (note that this truth assignment is well-defined). As before, every L-vertex on π is output by GREEDY(3-colourable); and, by arguing as we did earlier, for any $i, j \in \{1, 2, \ldots, n\}$ with $i < j$ and where (i, j) is a labelled edge of H, the truth assignment t makes $L^1_{i,j} \vee L^2_{i,j}$ true if, and only if, the vertices $d_{i,j}$, $e^1_{i,j}$ and $e^2_{i,j}$ are output.

At various points in the execution of GREEDY(3-colourable), a check is made to see whether the vertices of S induce a 3-colourable graph. Consider such a check and suppose that the vertices of $\{d_{i,j}, e^1_{i,j}, e^2_{i,j}\}$ have been placed in S. Consider the subgraph K of G induced by those vertices that are both in S and in the copy of G_1 pertaining to the labelled edge (i, j) of H. In particular, consider the role of K when it comes to attempting to colour the subgraph of G induced by the vertices of S. A simple combinatorial verification yields that the role of the vertices of K is to allow i and j to be coloured with any pair of distinct colours but not with identical colours. Hence, any check to see whether the subgraph of G induced by the vertices of S can be 3-coloured is equivalent to a check of whether the subgraph of $t(H)$ induced by (vertices corresponding to) the H-vertices of S can be 3-coloured. We know that our accepting computation on (G, P, s, x) outputs x. This can only happen if not all of $\{y, z, w\}$ are output, *i.e.*, if at least one a-vertex, a_m, say, is output, *i.e.*, if the H-vertex m is not output, *i.e.*, if the graph $t(H)$ can not be 3-coloured. The result follows. □

4 The construction

We now prove our main result using the techniques originating with Hell and Nešetřil. Of course, these techniques have to be adapted to our scenario.

Theorem 2 *The problem GREEDY(partial order, undirected graph, H-colourable) is* **NP***-complete, if H is bipartite, and* $\mathbf{\Sigma}_2^p$*-complete, if H is non-bipartite.*

Proof Throughout the proof we shall denote the problem GREEDY(partial order, undirected graphs, H-colourable) by $\mathcal{G}_H$. Clearly, $\mathcal{G}_H$ can be solved in $\mathbf{\Sigma}_2^p$, if H is non-bipartite, and in **NP**, if H is bipartite (the latter because the H-colourability problem, for H-bipartite, can be solved in polynomial-time [6]). Moreover, because the property of being H-colourable, for H bipartite, is non-trivial on graphs, hereditary, satisfied by all sets of independent edges and polynomial-time testable, by [11] we have that $\mathcal{G}_H$ is **NP**-complete if H is bipartite[1]. Actually, note that if H is bipartite then $\mathcal{G}_H$ and the problem GREEDY(partial order, undirected graphs, bipartite) are one and the same.

To prove that for any non-bipartite graph H, the problem $\mathcal{G}_H$ is $\mathbf{\Sigma}_2^p$-complete, we will modify the proof of Theorem 1 of [6] which states that: '*If H is bipartite then the H-colouring problem is in* **P**. *If H is non-bipartite then the H-colouring problem is* **NP***-complete.*' The proof begins by detailing three ways of constructing a graph H' from a graph H such that if the H'-colouring problem is **NP**-complete then the H-colouring problem is **NP**-complete as well. We will show that such constructions can be used to prove that the problem $\mathcal{G}_H$ is $\mathbf{\Sigma}_2^p$-complete.

Construction A: The indicator construction.

Let I be a fixed graph and let i and j be distinct vertices of I such that some automorphism of I maps i to j and j to i. The indicator construction (with respect to (I, i, j)) transforms a given graph H into a graph H^* defined to be the subgraph of H induced by all edges (h, h') for which there is a homomorphism of I to H mapping i to h and j to h'. Because of our assumptions on I, the edges of H^* will be undirected. The construction is illustrated in Fig. 3.

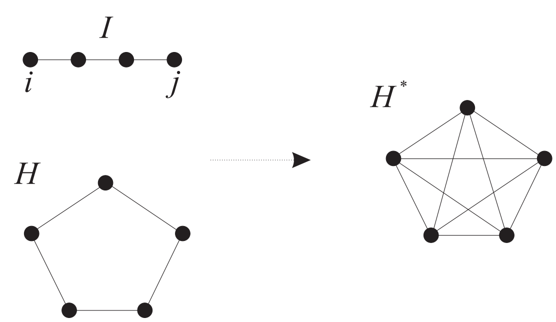

Figure 3. The indicator construction.

[1]Actually, the result proven in [11] insists that the property should be non-trivial on planar bipartite graphs, but it is straight-forward to weaken this assumption and still obtain our application.

Lemma 3 *If the problem $\mathcal{G}_{H^*}$ is $\mathbf{\Sigma}_2^p$-complete then so is $\mathcal{G}_H$.*

Proof Assume that $\mathcal{G}_{H^*}$ is $\mathbf{\Sigma}_2^p$-complete; and so, in particular, H^* has at least one edge (otherwise H^* would be the empty graph and $\mathcal{G}_{H^*}$ would not be $\mathbf{\Sigma}_2^p$-complete). We will reduce $\mathcal{G}_{H^*}$ to $\mathcal{G}_H$ (via a logspace reduction). Let (G^*, P^*, s^*, x^*) be an instance of $\mathcal{G}_{H^*}$. From it, we shall construct an instance (G, P, s, x) of $\mathcal{G}_H$.

Graph G is obtained from G^* as follows. For any vertex i of G^*, there is a corresponding vertex i of G: we will refer to such vertices of G as G^*-*vertices* (note how we consider the G^*-vertices of G and the vertices of G^* as being identically named). For any edge (u, v) of G^*, we add a copy of graph I to G by identifying the G^*-vertex u with vertex i in I and the G^*-vertex v with vertex j in I (all added copies of I are disjoint).

The partial order P consists of a linear order L (any one will do) on the vertices of G which are not G^*-vertices, and we concatenate on to this linear order the partial order P^* (of the G^*-vertices). Vertex s is the first vertex of the linear order L and vertex x is the G^*-vertex x^*. An illustration of this construction is depicted in Fig. 4 (where the graphs I, H and H^* are as in Fig. 3).

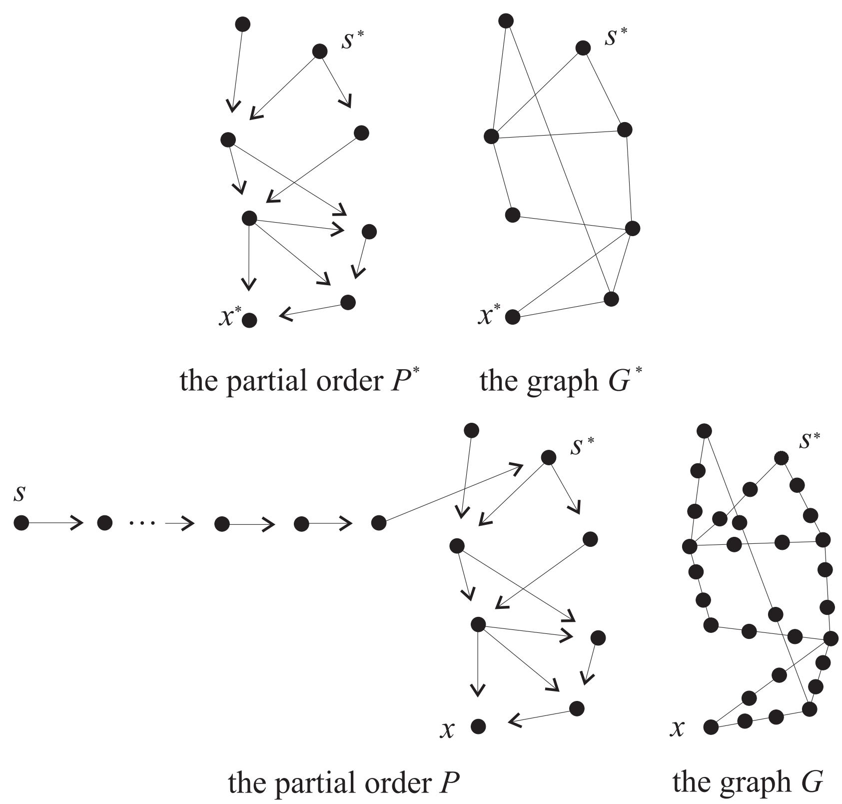

Figure 4. Building (G, P, s, x) from (G^*, P^*, s^*, x^*).

Consider the algorithm GREEDY(H-colourable) on the input (G, P, s). As H^* contains at least one edge, there is a homomorphism from I to H. Hence, as the linear order L consists of disjoint copies of $I \setminus \{i, j\}$, GREEDY(H-colourable) outputs every vertex of L. After consideration of the vertices of L, GREEDY(H-colourable) is working with essentially the same partial order as is the algorithm GREEDY(H^*-colourable) initially on input (G^*, P^*, s^*); so consider executions of these algorithms with respect to the same subsequent linear order.

Our induction hypothesis is as follows: 'The current-vertex in both executions is s_0; GREEDY(H-colourable) has so far output the vertices of $L \cup \{s_1, s_2, \ldots, s_m\}$, where vertex s_i is a G^*-vertex, for $i = 1, 2, \ldots, m$; and GREEDY(H^*-colourable) has so far output the vertices of $\{s_1, s_2, \ldots, s_m\}$.'

Suppose that the induction hypothesis holds at some point (it certainly holds when $s_0 = s^*$).

Suppose that GREEDY(H^*-colouring) outputs the vertex s_0. This means that there exists an homomorphism $f^* : \langle\{s_0, s_1, \ldots, s_m\}\rangle_{G^*} \to H^*$. By construction of H^*, there must exist a homomorphism $f : \langle L \cup \{s_0, s_1, \ldots, s_m\}\rangle_G \to H$, where $f(s_i) = f^*(s_i)$, for $i = 0, 1, \ldots, m$, and $f(v)$ is the 'natural' map for $v \in L$ (derived from the definition of H^* from H). Hence, GREEDY(H-colourable) outputs the vertex s_0.

Conversely, suppose that GREEDY(H-colourable) outputs the vertex s_0. This means that there exists a homomorphism $f : \langle L \cup \{s_0, s_1, \ldots, s_m\}\rangle_G \to H$. Again by construction of H^*, there must exist a homomorphism $f^* : \langle\{s_0, s_1, \ldots, s_m\}\rangle_{G^*} \to H_*$, where $f^*(s_i) = f(s_i)$, for $i = 0, 1, \ldots, m$. Hence, GREEDY(H^*-colouring) outputs the vertex s_0. The result follows by induction. □

Construction B: *The sub-indicator construction.*

Let J be a fixed graph with specified (distinct) vertices j and $k_1, k_2, \ldots, k_t$, for some $t \geq 1$. The sub-indicator construction (with respect to $J, j, k_1, k_2, \ldots, k_t$) transforms a given graph H with t (distinct) specified vertices $h_1, h_2, \ldots, h_t$ to its subgraph $\tilde{H}$ induced by the vertex set $\tilde{V}$ defined as follows. A vertex v of H belongs to $\tilde{V}$ just if there exists a homomorphism of J to H taking k_i to h_i, for $i = 1, 2, \ldots, t$, and taking j to v. An illustration of this construction is depicted in Fig. 5 (where, for clarity, we have shown the vertices of H excluded from $\tilde{H}$).

Lemma 4 *If the problem $\mathcal{G}_{\tilde{H}}$ is $\mathbf{\Sigma}_2^p$-complete then so is $\mathcal{G}_H$.*

Proof Assume that $\mathcal{G}_{\tilde{H}}$ is $\mathbf{\Sigma}_2^p$-complete; and so, in particular, $\tilde{H}$ has at least one vertex. We will reduce $\mathcal{G}_{\tilde{H}}$ to $\mathcal{G}_H$ (via a logspace reduction). Let $(\tilde{G}, \tilde{P}, \tilde{s}, \tilde{x})$ be an instance of $\mathcal{G}_{\tilde{H}}$. From it, we shall construct an instance (G, P, s, x) of $\mathcal{G}_H$.

The graph G is built from: a copy of $\tilde{G}$, of size n; a copy of H; and n copies of J (with J and H prior to the statement of the lemma), by identifying the vertex k_i in any copy of J with the vertex h_i of H, for $i = 1, 2, \ldots, t$, and identifying the vertex j in the i^{th} copy of J with the i^{th} vertex of $\tilde{G}$, for $i = 1, 2, \ldots, n$. The vertices of G corresponding to the vertices of $\tilde{G}$ (and the vertices j of the copies of J) are called *$\tilde{G}$-vertices*, the vertices of G corresponding to the vertices of the copies of J but different from $j, k_1, k_2, \ldots, k_t$ are called *J-vertices*, and the vertices of G corresponding to the vertices of H are called *H-vertices*.

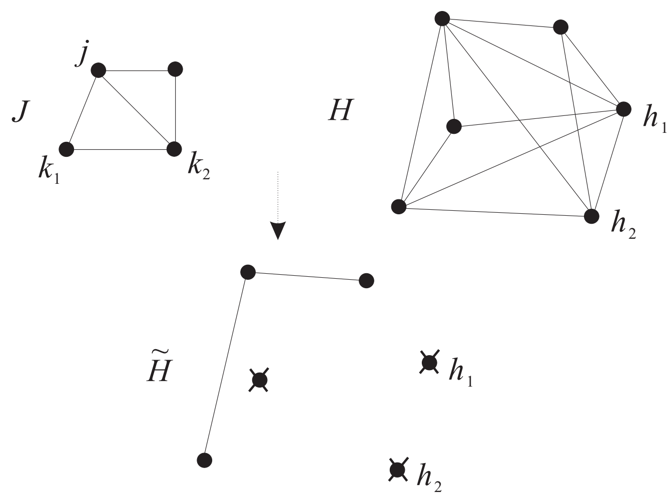

Figure 5. Building $\tilde{H}$ from H and J.

The partial order P consists of any linear ordering of the H-vertices, concatenated onto any linear ordering of the J-vertices concatenated onto the ordering $\tilde{P}$ of the $\tilde{G}$-vertices. The vertex s is the first H-vertex in the ordering P and the vertex x is the vertex $\tilde{x}$ of $\tilde{P}$. The whole construction can be pictured in Fig. 6. Clearly, this construction can be undertaken using logspace.

We begin by showing that any execution of GREEDY(H-colourable) on input (G, P, s) outputs every H-vertex and J-vertex of G. Clearly every H-vertex is output. Consider some copy of J (used in the formation of G). As $\tilde{H}$ has at least one vertex, there is a homomorphism from J to H taking k_i to h_i, for $i = 1, 2, \ldots, t$. Hence, every J-vertex is output. Denote the set of H-vertices and J-vertices of G by L.

Consider the algorithm GREEDY(H-colourable) on the input (G, P, s), where the current-vertex is $\tilde{s}$ (with the vertices of L having been output so far), and the algorithm GREEDY($\tilde{H}$-colourable) on the input $(\tilde{G}, \tilde{P}, \tilde{s})$ where the current-vertex is $\tilde{s}$ (note how we consider the $\tilde{G}$-vertices of G and the vertices of $\tilde{G}$ as being identically named). Essentially, these two algorithms work with the same partial order; so consider executions of these algorithms with respect to the same subsequent linear order.

Our induction hypothesis is as follows: 'The current-vertex in both executions is s_0; GREEDY(H-colourable) has so far output the vertices of $L \cup \{s_1, s_2, \ldots, s_m\}$, where each s_i is a $\tilde{G}$-vertex, for $i = 1, 2, \ldots, m$; and GREEDY($\tilde{H}$-colourable) has so far output the vertices of $\{s_1, s_2, \ldots, s_m\}$.'

Suppose that the induction hypothesis holds at some point (it certainly holds when $s_0 = \tilde{s}$).

Suppose that s_0 is output by GREEDY(H-colourable). That is, there is a homomorphism $f : \langle L \cup \{s_0, s_1, \ldots, s_m\}\rangle_G \to H$. In particular: $f(s_i)$ is a vertex of $\tilde{H}$, for $i = 0, 1, \ldots, m$; and if (s_i, s_j) is an edge of $\tilde{G}$ then $(f(s_i), f(s_j))$ is an edge of $\tilde{H}$, for $i, j = 0, 1, \ldots, m$. Hence, we have a homomorphism $\tilde{f} : \langle\{s_0, s_1, \ldots, s_m\}\rangle_{\tilde{G}} \to \tilde{H}$, and so s_0 is output by GREEDY($\tilde{H}$-colourable).

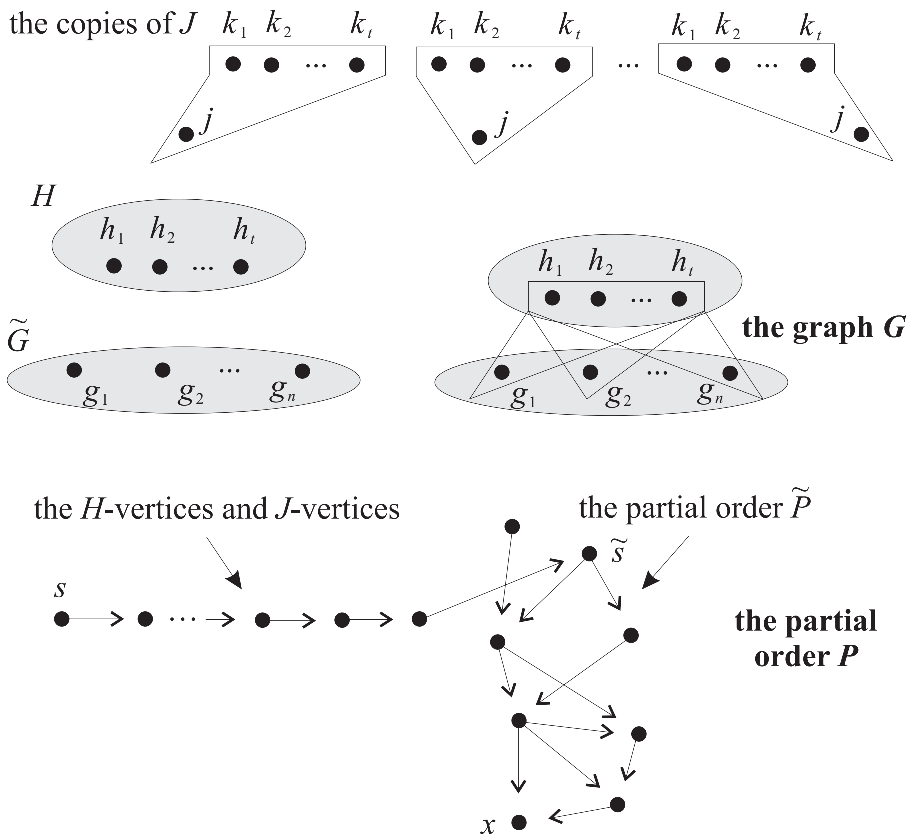

Figure 6. Building G from H, copies of J and $\tilde{G}$.

Conversely, suppose that s_0 is output by GREEDY($\tilde{H}$-colourable). That is, there is a homomorphism $\tilde{f} : \langle\{s_0, s_1, \ldots, s_m\}\rangle_{\tilde{G}} \to \tilde{H}$. Consider the copy of J corresponding to the $\tilde{G}$-vertex s_i of G. As $\tilde{f}(s_i)$ is a vertex of $\tilde{H}$, $\tilde{f}$ can be extended to a homomorphism $f : \langle L \cup \{s_0, s_1, \ldots, s_m\}\rangle_G \to H$. Hence, s_0 is output by GREEDY(H-colourable). The result follows by induction. □

Construction C: The edge-sub-indicator construction.

Let J be a fixed graph with a specified edge (j, j') and t specified vertices $k_1, k_2, \ldots, k_t$, such that all vertices $j, j', k_1, k_2, \ldots, k_t$ are distinct and some automorphism of J keeps $k_1, k_2, \ldots, k_t$ fixed while exchanging the vertices j and j'. The edge-sub-indicator construction transforms a given graph H with t (distinct) specified vertices $h_1, h_2, \ldots, h_t$ into its subgraph $\hat{H}$ induced by those edges (h, h') of H for which there is a homomorphism of J to H taking k_i to h_i, for $i = 1, 2, \ldots, t$, and j to h and j' to h'. The construction can be visualised as in Fig. 7.

Lemma 5 *If the problem $\mathcal{G}_{\hat{H}}$ is $\mathbf{\Sigma}_2^p$-complete then so is $\mathcal{G}_H$.*

Proof Assume that $\mathcal{G}_{\hat{H}}$ is $\mathbf{\Sigma}_2^p$-complete; and so, in particular, $\hat{H}$ has at least one edge. We will reduce $\mathcal{G}_{\hat{H}}$ to $\mathcal{G}_H$ (via a logspace reduction). Let $(\hat{G}, \hat{P}, \hat{s}, \hat{x})$ be an instance of $\mathcal{G}_{\hat{H}}$. From it, we shall construct an instance (G, P, s, x) of $\mathcal{G}_H$.

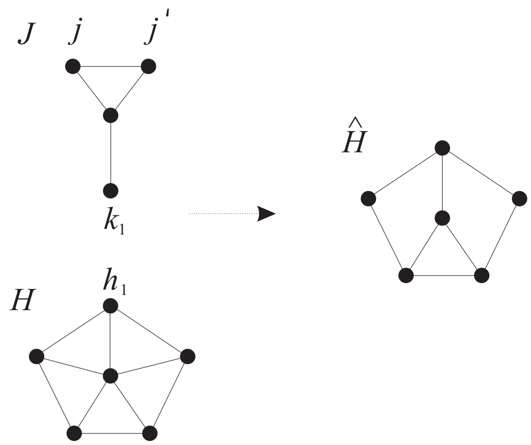

Figure 7. Building $\hat{H}$ from H and J.

The graph G is constructed from: a copy of $\hat{G}$, with e edges; a copy of H; and e copies of J (with H and J as prior to the statement of this lemma), by identifying every vertex k_i in any copy of J with the vertex h_i of H, for $i = 1, 2, \ldots, t$, and each edge e of $\hat{G}$ with the edge (j, j') of a unique copy of J. The vertices of G corresponding to the vertices of $\hat{G}$ (and the vertices j and j' of the copies of J) are called *$\hat{G}$-vertices*, the vertices of G corresponding to the vertices of the copies of J but different from $j, k_1, k_2, \ldots, k_t$ are called *J-vertices*, and the vertices of G corresponding to the vertices of H are called *H-vertices*.

The partial order P consists of any linear ordering of the H-vertices, concatenated onto any linear ordering of the J-vertices concatenated onto the ordering $\hat{P}$ of the $\hat{G}$-vertices. The vertex s is the first H-vertex in the ordering P and the vertex x is the vertex $\hat{x}$ of $\hat{P}$. The whole construction can be pictured in Fig. 8. Clearly, this construction can be undertaken using logspace.

We begin by showing that any execution of GREEDY(H-colourable) on input (G, P, s) outputs every H-vertex and J-vertex of G. Clearly every H-vertex is output. Consider some copy of J (used in the formation of G). As $\hat{H}$ has at least one edge, there is a homomorphism from J to H taking k_i to h_i, for $i = 1, 2, \ldots, t$. Hence, every J-vertex is output. Denote the set of H-vertices and J-vertices of G by L.

Consider the algorithm GREEDY(H-colourable) on the input (G, P, s), where the current-vertex is $\hat{s}$ (with the vertices of L having been output so far), and the algorithm GREEDY($\hat{H}$-colourable) on the input $(\hat{G}, \hat{P}, \hat{s})$ where the current-vertex is $\hat{s}$ (note how we consider the $\hat{G}$-vertices of G and the vertices of $\tilde{G}$ as being identically named). Essentially, these two algorithms work with the same partial order; so consider executions of these algorithms with respect to the same subsequent linear order.

Our induction hypothesis is as follows: 'The current-vertex in both executions is s_0; GREEDY(H-colourable) has so far output the vertices of $L \cup \{s_1, s_2, \ldots, s_m\}$, where each s_i is a $\hat{G}$-vertex, for $i = 1, 2, \ldots, m$; and GREEDY($\hat{H}$-colourable) has so far output the vertices of $\{s_1, s_2, \ldots, s_m\}$.'

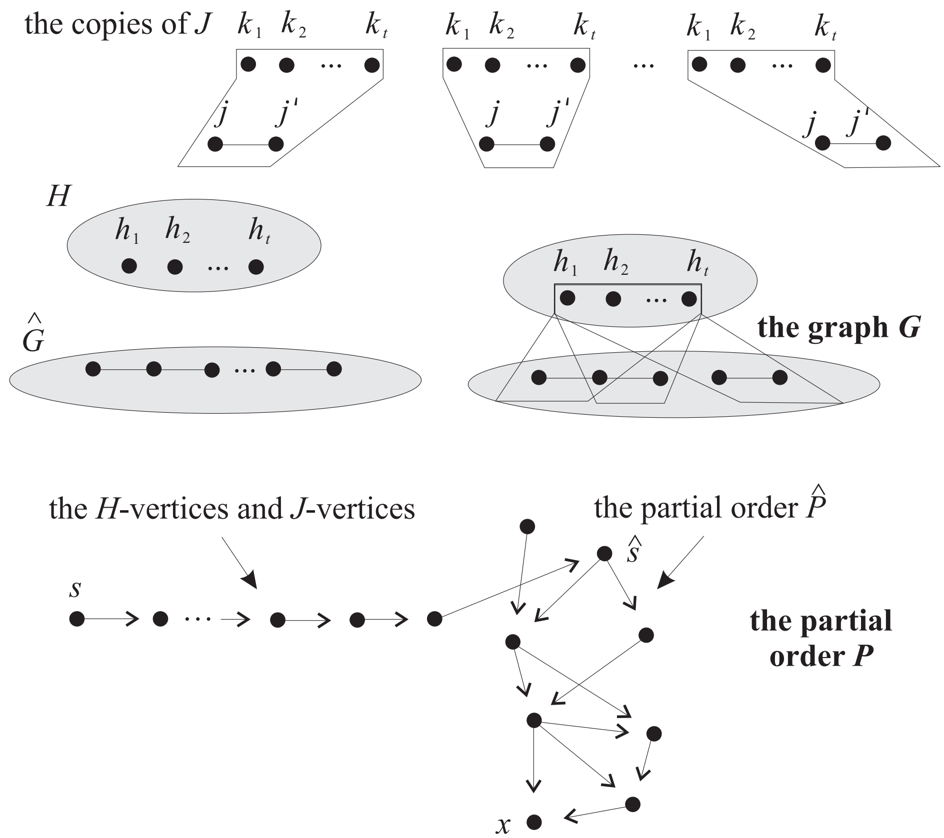

Figure 8. Building G from H, copies of J and $\hat{G}$.

Suppose that the induction hypothesis holds at some point (it certainly holds when $s_0 = \hat{s}$).

Suppose that s_0 is output by GREEDY(H-colourable). That is, there is a homomorphism $f : \langle L \cup \{s_0, s_1, \ldots, s_m\}\rangle_G \to H$. In particular, if (s_i, s_j) is an edge of $\hat{G}$ then $(f(s_i), f(s_j))$ is an edge of $\hat{H}$, for $i, j = 0, 1, \ldots, m$. Hence, we have a homomorphism $\hat{f} : \langle\{s_0, s_1, \ldots, s_m\}\rangle_{\hat{G}} \to \hat{H}$, and so s_0 is output by GREEDY($\hat{H}$-colourable).

Conversely, suppose that s_0 is output by GREEDY($\hat{H}$-colourable). That is, there is a homomorphism $\hat{f} : \langle\{s_0, s_1, \ldots, s_m\}\rangle_{\hat{G}} \to \hat{H}$. Consider the copy of J corresponding to the $\hat{G}$-vertex s_i of G. As $\hat{f}(s_i)$ is a vertex of $\hat{H}$, there must be a $\hat{G}$-vertex s_j of G such that $(\hat{f}(s_i), \hat{f}(s_j))$ is an edge of $\hat{H}$, and so $\hat{f}$ can be extended to a homomorphism $f : \langle L \cup \{s_0, s_1, \ldots, s_m\}\rangle_G \to H$. Hence, s_0 is output by GREEDY(H-colourable). The result follows by induction. □

Now we can proceed as Hell and Nešetřil did in [6]. Assume that there exists a non-bipartite graph H for which the problem $\mathcal{G}_H$ is not $\mathbf{\Sigma}_2^p$-complete. Choose H so that it is non-bipartite and the problem $\mathcal{G}_{H'}$ is $\mathbf{\Sigma}_2^p$-complete for any non-bipartite graph H':

(i) with fewer vertices than H; or

(*ii*) with the same number of vertices as H but with more edges.

It is straightforward to see that, under the assumption above, such an H must exist.

In [6], working from a similar hypothesis and graph H, the proof proceeds by using the indicator, sub-indicator and edge-sub-indicator constructions, in tandem with lemmas analogous to Lemmas 3, 4 and 5, to show that H must be a 3-clique; and hence that the 3-colouring problem is not **NP**-complete, thus yielding a contradiction. The sections of the proof of the main theorem of [6] entitled 'The structure of triangles' and 'The structure of squares' can be applied verbatim to our graph H (as the constructions we use are identical and we have our analogous Lemmas 3, 4 and 5). Hence, we may assume that H is 3-colourable, *i.e.*, that H is a 3-clique. However, Theorem 1 yields a contradiction as the problem GREEDY(partial order, undirected graphs, H-colourable) is none other than $\mathcal{G}_H$ when H is a 3-clique, and the result follows. □

5 Conclusion

In this paper, we have exhibited a complexity-theoretic dichotomy result concerning the non-deterministic computation of lexicographically first maximal H-colourable subgraphs of graphs. Our dichotomy result is different from other dichotomy results in that it is concerned with **NP**-completeness and $\mathbf{\Sigma}_2^p$-completeness, as opposed to computability in polynomial-time and **NP**-completeness as is more often the case. There are natural directions in which to extend this research.

> *Can we obtain a constructive proof of our main result?*

> *Can we obtain a similar result in the case of directed graphs or other structures?*

Of course, it is open as to whether there is a constructive proof of Hell and Nešetřil's result and also whether it can be extended to directed graphs; but it may be the case that these questions might be easier in our scenario.

> *What is the complexity of counting the number of distinct sets of vertices output by GREEDY*(π) (*on a given instance and for some appropriate property* π) *that contain a given vertex* v?

This question is motivated by the results of Dyer and Greenhill [3].

> *What is the complexity of the analogously defined lexicographically last maximal subgraph problem* (*again, with respect to an appropriate property* π)*, in the cases when a graph is linearly ordered and partially ordered?*

The only result we know of as regards computing lexicographically last subgraphs is that of [7] where it is proven that deciding whether a given set of vertices of a given linearly ordered graph is the lexicographically last such maximal independent set is **co-NP**-complete.

References

[1] C. Berge, *Graphs and Hypergraphs*, North-Holland (1973).

[2] J. Díaz, M. Serna and D. M. Thilikos, (H, C, K)-colorings: fast, easy and hard cases, *Proceedings of 26th International Symposium on Mathematical Foundations of Computer Science*, Lecture Notes in Computer Science 2136, Springer-Verlag, Berlin (2001) 304–315.

[3] M. Dyer and C. Greenhill, The complexity of counting graph homomorphisms, *Random Structures and Algorithms* **17** (2000) 260–289.

[4] T. Feder and P. Hell, List homomorphisms to reflexive graphs, *Journal of Combinatorial Theory Series B* **72** (1998) 236–250.

[5] T. Feder, P. Hell and J. Huang, List homomorphisms and circular arc graphs, *Combinatorica* **19** (1999) 487–505.

[6] P. Hell and J. Nesetril, On the complexity of H-colouring. *Journal of Combinatorial Theory Series B* **48** (1990) 92–110.

[7] D.S. Johnson, M. Yannakakis and C.H. Papadimitriou, On generating all maximal independent sets, *Information Processing Letters* **27** (1988) 119–123.

[8] S. Miyano, The lexicographically first maximal subgraph problems: P-completeness and NC algorithms, *Mathematical Systems Theory* **22** (1989) 47–73.

[9] S. Miyano, Δ_2^p-complete lexicographically first maximal subgraph problems, *Theoretical Computer Science* **88** (1991) 33–57.

[10] C.H. Papadimitriou, *Computational Complexity*, Addison-Wesley (1994).

[11] A. Puricella and I.A. Stewart, A generic greedy algorithm, partially-ordered graphs and NP-completeness, *Proceedings of 27th International Workshop on Graph-Theoretic Concepts in Computer Science*, Lecture Notes in Computer Science 2204, Springer-Verlag, Berlin (2001) 306–316.

[12] I.A. Stewart, Complete problems involving Boolean labelled structures and projection translations, *Journal of Logic and Computation* **1** (1991) 861–882.