General comment. The use of $|G|$ for a (directed) graph G is the usual Garey-Johnson style of measuring the overall size of a graph, and is therefore $O(|V(G)|^2)$. (This differs from the usage in some graph theory books like Diestel's, where $|G|$ is taken as $|V(G)|$.)

Page xiv "The three basic problems refereed to are:", refereed becomes referred

Page 15, Definition 2.1.1 "FTP" should be "FPT" (this one is kind of embarrasing).

Page 28, Proposition 3.3.1 "If S has fewer than $6k+1$ many vertices" should be "If $G-S$ has fewer than $6k+1$ many vertices"

Page 29, Proof of 3.3.2: "Then at each node we delete all of these vertices" should be "Then at each node we delete the associated vertex and its neighbours"

Page 29, 3.4.1: "As we know, a planar graph G we have a degree 5 vertex v should be "As we know, in a planar graph G we have a vertex v of degree 5 or less."

In the following sentences this also should be changed:

"either v or one of its five neighbours" should be "either v or one of its at most five neighbours" "we still keep the five elements of $N[v]$" should be "we still keep the at most five elements of $N[v]$"

Page 30, Annotated Dominating Set "such that for every vertex $u \in B$, there is a vertex $u' \in N[u] \cap V'$?" should be "such that for every vertex $u \in B$, there is a vertex $u' \in N[u] \cap V'$ or $u \in V'$?"

Page 31, Proof of Lemma 3.4.2: "First delete every edge between two black or two white vertices" should be "First delete every edge between two red or two white vertices"

Page 31, Proof of Lemma 3.4.2: "numbered so that the red neighbor u of w occurs between b_d and v_1" should be "numbered so that the red neighbor u of w occurs between b_d and b_1"

Page 33. Proof of Lemma 3.5.1: For a simpler presentation, Oum suggests that we would use the poly-time algorithm to find a shortest cycle (for each edge e, find a shortest path joining 2 ends.) Then the running time can be bounded by $O^*((2k)^k)$.

Page 34, proof of Lemma 3.2.1. Line -13 and line -1, replace $|V|$ by $|G|$.

Page 36 and page 549, Question for Closest String. This should read, "find $s = s_j$, for some $j \in \{1, \ldots, k\}$," Page 37, Theorem 3.6.1. $|G|$ should be $|\Sigma|$.

Page 64, Line 2, Section 5.1.3 should be 5.1.5.

Page 82, Exercise 4.11.5: Bodleander $\rightarrow$ Bodlaender

Page 99, Definition 5.1.3 (ii), should read:

..."with oracle L_2, such that $\Phi^{L_2} = L_1$, and on input $\langle \sigma, k \rangle$, Φ only only makes queries to the oracle L_2 of the form $\langle \tau, k' \rangle$ for $|\tau|, k' \leq g(k)$."

Page 108 Algorithm 6.1.2, Step 3: "$\hat{C} = C \cup \{v\}$" should be "$\hat{C} = C_s \cup \{v\}$".

- Algorithm 6.1.2, Step 3: Throughout "$C_s \cup \{v\}$" could be "$\hat{C}$". (This is not really a correction.)

- Algorithm 6.1.2, Step 3: "$D \sqcup Q$" could be "D and Q", depending on how you define partitions.

- Algorithm 6.1.2, Step 3: "The idea is that that the [...]". Typo: Just one "that".

Page 109

After Algorithm 6.1.2, time "$O(2^k|G|)$" should be "$O(2^k|G|n)$", since each compression needs at most "$O(2^k|G|)$" and we do at most n compression steps.

Page 110, statement of Lemma 6.2.2. 2 should say that Y contains an edge-cut instead of saying that "Y is an edge cut in $G - X$."

Page 110, in the proof of Lemma 6.2.2. G_y should be C_Y It might be clearer to say edges rather than paths in the proof.

also same proof:

(2) implies (1): "$\{u_0v_0, u_1v_1, ..., u_qv_q\}$" should be "$\{u_0v_0, u_1v_1, ..., u_{q-1}v_{q-1}\}$", if we assume q is also the number of edges in this set.

(1) implies (2): "G_Y a two-colouring of $G-Y$" should be "C_Y a two colouring of the bipartite graph $G-Y$". We know (and need) the property of $G-Y$ being bipartite.

(Page 110, line -15) Now define $\hat{\Phi}$, not Φ.

"Thus $C_X(u) = C_Y(v)$" should be "Thus $C_X(u) = C_X(v)$".

"[...], and hence $C_Y(u) \neq C_X(v)$" should be "[...], and hence $C_Y(u) \neq C_Y(v)$".

"Therefore $\hat{\Phi}(v) \neq \hat{\Phi}(v)$." should be "Therefore $\hat{\Phi}(u) \neq \hat{\Phi}(v)$.". (It could also be "$\Phi(u) \neq \Phi(v)$", the restriction to $V(X)$ isn't needed here, since we're only looking at vertices from $V(X)$.

Proof of Theorem 6.2.1: "[...] Lemma 6.2.2 with "$X = X_{i-1}$" [...]", should be "[...] Lemma 6.2.2 with "$X = X_{i-1} \cup \{e_i\}$" [...]". "X_{i-1}" should be "$X_{i-1} \cup \{e_i\}$" throughout.

"[...] in its stead for G_i." should be "[...] in its stead for X_i.".

"If we find a Y, set $G_i = Y$" should be "If we find a Y, set $X_i = Y$"

Page 119, line 1. Lemma 6.6.1 should be Lemma 6.5.1.

Page 165, Exercises 8.8.1 and 8.8.2. The sentence here means that $\mathcal{F}$ is a family of subsets of a set X such that $|\mathcal{F}| = r$, *not* that the subsets have size r. (Oum suggests that probably a better name for this problem is "ANTICHAIN": to find an antichain of size k in a collection of r subsets.)

Page 185, line -4. after *Pascal*, put "are are ≤ 6."

Page 202, line -7, "1966" should be "1996".

Page 215, "quickly decide if $\sigma \in L(M)$." (not $\sigma \in M$)

Page 215, "a little warm up for the next sections, where we will *show* that" ("show" missing)

Page 220, proof of Theorem 12.3.2, describing converted Δ' set. Instead of "to $E(r)$, provided some $q_i \in E(q)$ and some $q_j \in E(r)$, with Δ taking q_i to q_j on input a," replace with "taking $E(q)$ to $E(r)$ provided that there is some $q_i \in E(q)$ with Δ taking q_i to r in input a." (This has a consequential slight change of the diagrams.)

Page 226, Figure 12.7 is cut-off on the right in Step 3, but the missing bit is obvious.

Page 227, line -3, item 2 of the statement of Theorem 12.5.1.

Furthermore any right congruence satisfying (b) and (c) of 1 is a ...

Page 228, line -3. Now we must *show* M ...

Page 235, Exercise 12.5.4. in the hint replace $\sim_L$ by $\approx_L$. Page 275, statement of Theorem 13.4.2. Actually Theorem 13.4.2 (Courcelle and Oum) is still open as stated here. Courcelle and Oum proved only for C2MS1 logic and could only prove a weaker statement with the set predicate for the "even cardinality".

Pages 254-257.

Myhill-Nerode Theorem for Graphs

This is a big error in the proof of this. Here is a correct (and easier) proof. Some property like parsing replacement can likely be extracted.

We prove Theorem 6.77 of [DF98]= Theorem 12.7.2 of the book.

The proof in both places, being the same, has a small error. This is fixed in this note.

The error is in the proof of (iii)→(i).

For neatness we will use the t-boundary operators given, and not worry about a general property making this work.

Now we know that a graph has pathwidth t iff it can be parsed by the above operators without using $\oplus$, and the analogous Parsing Theorem (6.72 of [DF98], 12.7.1 of [DF13]) holds. Now consider the parsing theorem in the context of the small universe.

It is easy to prove by induction of the length of the path, that if G is a pathwidth t graph in the small universe of treewidth t graphs, then G is isomorphic (in the universe forgetting the boundary) to a graph G_1 with a parsing and the boundary vertices in the first bag and also one where the boundary is in the last bag of the parsing of G_1.

Since we can move a boundary over an $\oplus$, it follows that in the small universe of treewidth t graphs, if we consider a parse tree T, then $G(T)$ is isomorphic to a parse tree $\hat{T}$ where the boundary in the underlying tree of bags is at the root, and also one $\hat{T}'$ where the boundary corresponds to a bag corresponding to a given leaf of T.

Now, following the proof of (iii) implies (i), we assume we are given T_k and $T_{i.j}$ such that for all i, j $T_i \cdot_x T_{i,j} \in L$ iff $T_j \cdot_x T_{i,j} \notin L$, where L is the language of trees which are equivalent if the underlying graphs are isomorphic as unlabelled graphs.

The argument above says that we can regard the root of each T_k to correspond in the underlying bags given by the underlying parsing theorem to have the boundary, and the bag corresponding to the leaf of $T_{k,j}$ and x to have the boundary.

In that case, we see that $G(T_i \cdot_x T_{i,j}) \cong G(T_i) \oplus G(T_{i,j})$ since with the boundaries at that placement, there is a $G(T_i \cdot_x T_{i,j})$ corresponds simply to gluing the underlying graphs along that boundary. (Notice that, to do this it might be that we might need to make parts of the boundaty corresponding to (for example) $T_{i,j}$ disjoint. It might be, for instance, that $T_i \cdot_x T_{i,j}$ might correspond to disjoint graphs, but this can be construed as a gluing in any case.)

Then we get a contradiction, since now $G(T_k)$ would witness that $\sim_{\mathcal{F}_t}$ does not have finite index.

Page 381 line 4: delete closing parenthesis.

Page 397. Proof of 21.2.4. "By Theorem 21.2.2 and 21.2.3"

Page 488. Exercise 25.2.3 The exercise asks to prove that Weighted Monotone and Antimonotone Satisfiability are $W[P]$ complete, but this should be $W[SAT]$ complete.

Page 541 section 29.2 line -1: seem → seems

Page 546: Theorem 29.5.1 The d's and k's are mixed up. The occurrence of 2-SAT should be k-Sat. Also 2. should read $p \leq 2^{fracn2}$. And the running time is $2^{\frac{n}{d}}|\phi|^{O(1)}$.

Page 547 line -1: add "holds" as the last word for the sentence.

Page 553 section 29.6 first sentence, delete the fullstop half way through the sentence.

Page 597, 30.10.1, line -5, (i.e. "1") delete "and"

Page 597, 30.10.1, line -3 (i.e. "2") "instance" should be "instances"

Page 636, §31.4.2. There are some genuine mathematical problems in this section.

Theorem 31.4.1 is correct as stated, but does not follow from the proof in the §31.4.2. The *width* of the interval graph is the size of the largest clique in the interval representation, and hence the width used in this section for interval graphs is one higher than the pathwidth. Thus, the algorithm will only colour k-paths with $3k + 1$ many colours. The proof only works for interval graphs and hence for k-paths, whereas graphs of pathwidth k are *partial* k-paths.

In recent work, Askes and Downey showed that the theorem is nevertheless true with a different proof.

Theorem 1 (Askes and Downey (*Online, Computable, and Punctual Structure Theory*, to appear)). *Every online graph G of pathwidth k can be online coloured with at most $3k + 1$ many colours.*

Proof. This is proven by induction on the width k. If $k = 1$ then G is a path and we can use greedy minimization which will use at most 3 colours. So suppose $k > 1$, and let G_n have vertices $\{v_1, \ldots, v_n\}$. The online algorithm A_k will have computed a partition of G, which we denote by $\{D_y \mid y < k\}$. Consider v_{n+1}. We refer to the D_y as *layers.* If the pathwidth of $G_{n+1} = G_n \cup \{v_{n+1}\}$ is $< k$, colour v_{n+1} by A_{k-1}, and put into one of the cells D_y, for $y < k-1$ recursively. We will be colouring using using the set of colours $\{1, \ldots, 3k-2\}$.

If the pathwidth of G_{n+1} is k, consider H_{n+1}, the induced subgraph of $G_{n+1} \setminus D_k$. If the pathwidth of H_{n+1} is $< k$, then again colour v_{n+1} by A_{k-1}, and put into one of the cells D_y, for $y < k-1$, recursively, and colour using the set of colours $\{1, \ldots, 3k-2\}$. If the pathwidth of H_{n+1} is k, then we put v_{n+1} into D_{k-1}.

in this case we will use first fit using colours $3k - 2 < j \leq 3k + 1$.

The validity of this method follows from the fact that the maximum degree of vertices restricted to D_{k-1} is 2, and induction on k. Assume that A_{k-1} is correct and colours using colours $\{1, \ldots, 3k-2\}$. We are assuming that v_{n+1}'s addition to G_n has pathwidth k. Now consider a path decomposition $B_1, \ldots, B_q$ of G_{n+1}. Suppose that the degree of $v = v_{n+1}$ in D_k is ≥ 3. Thus there are x, y, and z in D_k which are each connected to v. Without loss of generality, let's suppose that that they were added at stages $s_x < s_y < s_z \leq n$. Since each is in D_k, when we added them to D_k, we could not have added them to D_y for $y < k-1$. Since they were not added to such D_y it follows that as the stages they were added, they made the pathwidth of the relevant H_s ($s \in \{s_x, s_y, s_z\}$ to be k. Consider s_x. As the pathwidth of H_{s_x} was k, there must be some bag in any path decomposition of G_{s_x}, comsisting of only members of G_{s_x} which has size $k+1$, and containing x. For $t > s_x$, this must still hold. For suppose this was not true at stage t. The pathwidth of G_t is k, and has bags $P_1, \ldots, P_v$, say. Now delete all of the elements of $G_t \setminus G_{s_x}$ from the bags. This is a path decomposition of G_{s_x}, and hence must have pathwidth k, so there must be one of size $k+1$ containing x, and it only consists of elements of G_{s_x}.

Consider s_y. Since the pathwidth of H_{s_y} is k, it follows that s_y must be in a bag of size k in the path decomposition of H_{s_y} containing none of D_{k-1}. In particular, in any path decomposition of G_{s_y}, there x and y must appear in bags Q_x and Q_y, respectively, of size k with $x \notin Q_y$ and $y \notin Q_x$,

and the same holds thereafter. So we can conclude, using the same reasoning, that at stage $n+1$, x, y, z, and v are all in bags of size k, B_x, B_y, B_z, B_v, where $x \notin B_y \cup B_z \cup B_v$, and similarly for y, z and v.

Now consider B_v. Since xv is an edge, x and v lie together in some bag B_{xv}. If B_{xv} is left of B_v but B_{xv} is right of B_v we get a contradiction, since this would put x into B_v, by the interpolation property of pathwidth. So B_{xv}

and B_x both lie, without loss of generality left of B_v. Similarly B_{yv} and B_y must lie on the same side, and this must be right. For if there were both left of B_v, then the interpolation property would make either B_x or B_y contain y of x respectively (considering the relevant orientations of B_x and B_y). But now we get a contradiction, since B_z cannot be wither right or left of B_v without one of the B_x, B_y, or B_z containing a forbidden element. Thus, within D_{k_1} the degree of v is at most 2.

We remark that the proof of the theorem above gives an algorithm which is linear time (as k-PATHWIDTH is linear time FPT), but is inefficient as the constants for the pathwidth algorithm are of the order of 2^{35k^2} which is pretty horrible. We don't know the best complexity for the following (online) promise problem

Input: An online graph G, and a vertex v and a graph H with vertices $V(G) \cup \{v\}$ G a subgraph of H.

Promise: G has pathwidth k.

Parameter: An integer k.

Question: Does H have pathwidth k?

Page 637 after Lemma 31.4.2 add "We leave the proof of this Lemma to the reader.

Page 637, Line -8 "Proof" should be "Proof of 31.4.3"

Page 637, Line -6 "$k = 1$" should be "$k + 1$".

Page 637, Line -6. We define B *at step* p (which is the next vertex to be added) by induction on k at step p.

Page 637, Line -5. B^P should be B^p.

Page 637, Line -4. (Of course in the actual online algorithm we will only have B^p at step p.