Available online at www.sciencedirect.com
SCIENCE DIRECT®
Theoretical Computer Science 319 (2004) 455–482
Theoretical Computer Science
www.elsevier.com/locate/tcs

On unique graph 3-colorability and parsimonious reductions in the plane

Régis Barbanchon
GREYC, Université de Caen, 14032 Caen Cedex, France

Abstract

We prove that the Satisfiability (resp. planar Satisfiability) problem is *parsimoniously* P-time reducible to the 3-Colorability (resp. Planar 3-Colorability) problem, that means that the exact number of solutions is preserved by the reduction, provided that 3-colorings are counted modulo their six trivial color permutations. In particular, the uniqueness of solutions is preserved, which implies that Unique 3-Colorability is exactly as hard as Unique Satisfiability in the general case as well as in the planar case. A consequence of our result is the DP-completeness of Unique 3-Colorability and Unique Planar 3-Colorability under random P-time reductions. It also gives a finer and unified proof of the #P-completeness of #3-Colorability that was first obtained by Linial for the general case, and later by Hunt et al. for the planar case. Previous authors' reductions were either *weakly parsimonious* with a multiplication of the numbers of solutions by an exponential factor, or involved #P-complete intermediate counting problems derived from trivial "yes"-decision problems.

Keywords: Computational complexity; Planar combinatorial problems; 3-Colorability; Unique solution problems; Parsimonious equivalence to SAT

1. Introduction

Parsimonious reductions—i.e., P-time reductions that preserve the exact number of solutions of the input problem—are interesting for at least two reasons: (1) such reductions generally preserve the structure of the space of the solutions, since they realize in practice a bijective correspondence between the sets of solutions that is P-time computable; more precisely, linear parsimonious reductions preserve the time complexity of the enumeration of solutions, i.e., the delay between consecutive solutions up to a

E-mail address: regis.barbanchon@info.unicaen.fr (R. Barbanchon).

0304-3975/$ - see front matter
doi:10.1016/j.tcs.2004.02.003

constant factor [4], and (2) such reductions not only allow to prove #P-completeness results for counting problems [15] but also DP-completeness results for decision problems asking about the existence of unique solutions [16]. Many P-time reductions between NP-complete problems are indeed parsimonious. In particular, it is significant to note that the generic reduction that proves the NP-hardness of the *satisfiability* problem (SAT) is easily made parsimonious, which means that SAT accurately captures the structure of any NP problem.

This is not true for all NP-complete problems however, to begin with the ones whose sets of solutions display intrinsic symmetries, e.g., graph *3-Colorability* [1] (3-COL). Typically, each solution of any instance of 3-COL induces six solutions that are *isomorphic* under color permutations. Therefore, the number of 3-colorings of any instance of 3-COL is trivially a multiple of six. Another example is the problem called *Not-All-Equal-In-3-Positive-Sat* (NAE-3-SAT), which asks whether a given conjunction of positive 3-clauses has an assignment for which each 3-clause contains at least one true literal and one false literal. Each solution of NAE-3-SAT induces two solutions that are isomorphic under bitwise negation, so the number of solutions of every instance of NAE-3-SAT is trivially even. Of course, no such symmetry happens for SAT, and for any fixed integer k, it is easy to build SAT instances with exactly k solutions. As an obvious consequence, *no parsimonious* transformation can exist from SAT to 3-COL (or to NAE-3-SAT).

However, one can naturally regard a group of isomorphic solutions as *only one solution*, and count the solutions accordingly. With this new counting convention, the argument does not hold anymore and one can naturally asks whether parsimonious P-time transformations exist from SAT to 3-COL (resp. to NAE-3-SAT). Also, it now makes sense to ask if a given 3-COL or a NAE-3-SAT instance has a unique solution (problems U-3-COL and U-NAE-3-SAT), and exhibiting the reductions above would imply that U-3-COL (resp. NAE-3-SAT) is as hard as deciding if a SAT instance has a unique satisfying assignment (problem U-SAT).

From now on, we shall always consider any group of isomorphic solutions as only *one* solution: interestingly, it is already known that a parsimonious reduction from NAE-3-SAT to SAT does exist under our counting convention, since Creignou and Hermann [5] parsimoniously reduced *1-Exactly-In-Positive-3-Sat* (1/3-SAT) to NAE-3-SAT. The link to SAT itself is done via the following result, whose proof can be found in [9] or alternatively in Appendix B of this paper.

Proposition 1. 1/3-*SAT and SAT are parsimoniously reducible to each other*.

However, we are not aware of a similar result for the more interesting problem 3-COL. Indeed, the classical reductions from SAT to 3-COL, e.g., the one presented by Kozen [11], are not even *weakly parsimonious*, i.e., they do not even establish any precise relation between the number of solutions of the instances, because the 3-colorings are duplicated without any control. However, a *weakly parsimonious* reduction from SAT to 3-COL can be obtained by composing three transformations: the

[1] Terms in italics are formally defined in Section 2.

parsimonious reduction from SAT to 1/3-SAT, Creignou and Hermann's parsimonious reduction from 1/3-SAT to NAE-3-SAT [5], and Dewdney's weakly parsimonious reduction from NAE-3-SAT to 3-COL [6]. While this weak parsimony together with the #P-completeness of #SAT is sufficient to imply the #P-completeness of #3-COL [2] it gives no clue about the expressiveness of U-3-COL compared to the one of U-SAT, because Dewdney's reduction multiplies the solutions by an exponential factor.

A formula φ in CNF is often associated with the bipartite incidence graph $G(\varphi)$ whose set of vertices is made of the set of variables on one hand and the set of clauses on the other hand. Planar formulas are simply the formulas φ such that $G(\varphi)$ is planar and planar versions of SAT, 1/3-SAT and NAE-3-SAT can then be defined as the restrictions of the these problems to planar instances. The questions raised above can naturally be addressed for these restrictions. Note that since the transformations between SAT and 1/3-SAT preserve the planarity of the graphs:

Proposition 2. *PLAN*-1/3-*SAT and PLAN-SAT are parsimoniously reducible to each other.*

Also, it is well known that:

Proposition 3. *PLAN-SAT and SAT are parsimoniously reducible to each other.*

This was established by Lichtenstein by using a well-known parsimonious *crossover-box* eliminating the potential edge crossings [12,9]. Alternatively, one can take advantage of the parsimonious equivalence of SAT and 1/3-SAT on one hand, and of PLAN-SAT and PLAN-1/3-SAT on the other hand, to establish this equivalence by a parsimonious reduction from 1/3-SAT to PLAN-1/3-SAT which is presented in Appendix C for the sake of completeness.

As far as PLAN-NAE-3-SAT and PLAN-3-COL are concerned, the former is a trivial "Yes"-problem as an easy consequence of the Four-Colors Theorem in planar graphs, whereas 3-COL remains NP-complete in the plane. The classical reduction from 3-COL to PLAN-3-COL eliminates the edge crossings by using a well-known non-parsimonious crossover-box [14,11,7]. Hunt et al. modified this crossover-box in [10] to make it weakly parsimonious and hence proved the #P-completeness of #PLAN-3-COL via a weakly parsimonious reduction from 3-COL to PLAN-3-COL, with a multiplication of the number of solutions by an exponential of the square of the size of the input.

Thus, to our knowledge and even with our natural counting convention, no parsimonious reductions are known till now neither from 3-COL to PLAN-3-COL nor from SAT to 3-COL, let alone from PLAN-SAT to PLAN-3-COL. In particular, the hardness

[2] #3-COL was earlier shown to be #P-complete by Linial [13], but not from the #P-completeness of #SAT. This was done under a parsimonious P-time transformation from the #P-complete problem #STABLE in bipartite graphs to #3-COL in bipartite graphs, that are counting problems whose associate decision problems are both trivial "Yes"-problems.

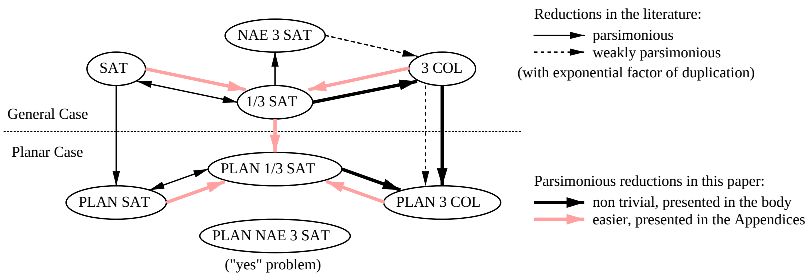


Fig. 1. Reductions towards 3-COL and parsimony.

of U-3-COL and U-PLAN-3-COL are open problems. In this paper, we show that such reductions do exist:

Proposition 4. 3-*COL is parsimoniously reducible to PLAN*-3-*COL in quadratic time*, *and in particular U*-3-*COL is so reducible to U-PLAN*-3-*COL.*

Proposition 5. *SAT is parsimoniously reducible to* 3-*COL in linear time*, *and in particular U-SAT is so reducible to U*-3-*COL.*

Proposition 6. *PLAN-SAT reduces parsimoniously to PLAN*-3-*COL in linear time*, *and in particular U-PLAN-SAT is so reducible to U-PLAN*-3-*COL.*

Fig. 1 sums up the contributions of this paper. Since the easier converse linear reductions from (PLAN-)3-COL to (PLAN-)1/3-SAT also exist, that implies in particular that U-3-COL and U-SAT (resp. U-PLAN-3-COL and U-PLAN-SAT) have exactly the same time complexity up to a constant multiplicative factor. Furthermore, the delays between the output of two consecutive solutions during an enumeration of all the solutions are preserved up to a multiplicative constant. As far as polynomial time complexity classes are concerned, this gives finer and unified proofs of the #P-completeness of #3-COL and #PLAN-3-COL. Also, since U-SAT and U-PLAN-SAT are both known to be complete problems in the *class DP* [16,9] under random P-time reductions,[3] we conclude that:

Corollary 7. 3-*COL*, *PLAN*-3-*COL*, *SAT and PLAN-SAT are equivalent under parsimonious reductions*, *and hence U*-3-*COL and U-PLAN-3-COL are DP-complete under random P-time reductions.*

The paper is organized as follows: Section 2 presents all the definitions of technical terms used in the paper. A number of tools will have to be designed to reach our

[3] A random P-time reduction R from a decision problem A to a decision problem B is a polynomial time Random Turing Machine such that for any instance I of the problem A, it holds (1) $I \in A \Rightarrow R(I) \in B$ with probability $1/p(|I|)$ for some polynomial p, and (2) $I \notin A \Rightarrow R(I) \notin B$.

goal. Section 3 sketches their high-level behavior and explains how we expect them to interact in the big picture. To show the parsimonious equivalence of (PLAN-)SAT and (PLAN-)3-COL, we take advantage of the parsimonious equivalence of SAT, PLAN-SAT, 1/3-SAT and PLAN-1/3-SAT and we only have to parsimoniously reduce (PLAN-)3-COL and (PLAN-)1/3-SAT to each other. Section 4 is devoted to the reduction from (PLAN-)1/3-SAT to (PLAN-)3-COL. The complete proofs of the behaviors of those gadgets are presented in Appendix A. The converse reduction from (PLAN-)3-COL to (PLAN-)1/3-SAT is shown in Appendix D. Finally, we show in Section 5 that we can derive a parsimonious crossover-box for PLAN-1/3-COL from our tools, hence improving the weakly parsimonious crossover-box of [10]. This gives a *direct* parsimonious reduction from 3-COL to PLAN-3-COL (i.e., without using 1/3-SAT and PLAN-1/3-SAT as intermediate problems).

2. Preliminaries and definitions

We now recall the studied satisfiability/colorability problems, the involved complexity classes, and the technical tools and concepts used in the whole paper.

Definition 8 (Problem SAT). *Input:* a CNF formula $\varphi(V,L)$ that is a list L of clauses over the set of variables V. *Question:* does V admit a truth-assignment such that *at least one* literal per clause in L is assigned true?

Definition 9 (Problem NAE-3-SAT). *Input:* a CNF formula $\varphi(V,L)$ that is a list L of *positive* 3-*clauses* (i.e. clauses of length 3 with no negative literals) over the set of variables V. *Question:* does V admit a truth-assignment such that each clause in L contains *at least one* true variable and one false variable, i.e., such that not all variables are equal in any clause?

Definition 10 (Problem 1/3-SAT). *Input:* a CNF formula $\varphi(V,L)$ that is a list L of *positive* 3-*clauses* (i.e. clauses of length 3 with no negative literals) over the set of variables V. *Question*: does V admit a truth-assignment such that *exactly one* variable per clause in L is assigned true?

Definition 11 (Formula-graph and planar formula). The *formula-graph* $G(\varphi)$ of a CNF formula $\varphi(V,L)$, where L is a list of clauses over the set of variables V, is defined as the bipartite graph $G(V\cup L,E)$, with $E=\{(v,c),c\in L,v\in c\}$. If G is planar, then φ is called a *planar formula*.

We now see the SAT-like problems above as vertex 2-coloring problems of the Formula-graphs of their inputs, with the two colors *true* and $false$.

Definition 12 (Vertex k-coloring). A vertex k-coloring of a graph $G(V,E)$ is a function $C: V \longrightarrow P_k$ (where the k-palette P_k is a set of k colors).

Definition 13 (Problem 3-COL). *Input:* a graph $G(V,E)$. *Question:* Does a 3-coloring of the vertices $C: V \longrightarrow P_3$ where $P_3 = \{white, gray, black\}$ exist such that for all $(x,y) \in E$, $C(x) \neq C(y)$?

Definition 14 (Isomorphic colorings). Let $G(V,E)$ be a graph, and C_1, C_2 be two vertex colorings of G with the palette $P_k = \{1,\dots,k\}$. C_1 and C_2 are *isomorphic* if there exists a color permutation $\pi: P_k \longrightarrow P_k$ such that for all $x \in V$, $C_1(x) = \pi(C_2(x))$.

3-COL and NAE-SAT are examples of vertex coloring problems with trivial color isomorphisms. We now define the planar versions, counting versions and "unique" versions of the decision problems cited above.

Definition 15 (Problem PLAN-Π). For any problem Π on graphs, its planar version *PLAN-*Π is defined as the restriction of Π to planar inputs.

Definition 16 (Problems #Π and U-Π associated to a problem Π). For any decision problem Π on input I, the counting version #Π is the problem asking *the number of distinct solutions* of I for Π, and its "unique" version *U-*Π is the problem asking whether I has a *unique solution* for Π. For the coloring problems cited above, a solution is a vertex coloring (in particular, a truth assignment for SAT-like problems). For problems whose sets of solutions have trivial symmetries (e.g., 3-COL and NAE-3-SAT), two isomorphic colorings are counted as *one coloring*.

Definition 17 (Class DP). A property belongs to the *class DP* if it is the conjunction of an NP property and a co-NP property.

In particular, U-SAT and U-3-COL belong to DP since they ask on one hand whether at least one solution exists, and on the other hand whether no two solutions exist.

Definition 18 (Parsimonious and weakly parsimonious reductions). A P-time reduction R from problem Π_1 to problem Π_2 is *weakly parsimonious* if, for each instance $I_1 \in \Pi_1$, its number of solutions $\#I_1$ for problem Π_1 is equal to $f_R(I_1) \times \#I_2$, where $I_2 = R(I_1)$, $\#I_2$ is the number of solutions of I_2 for Π_2, and f_R is a P-time computable function. R is *parsimonious* if and only if $\#I_1 = \#I_2$ (i.e., $f_R(I_1) = 1$).

Our parsimonious reductions will use the following notions:

Definition 19 (Gadget and distinguished vertices). A *gadget* is a connected graph $G(V,E)$ with a specified subset $X = \{x_1,\dots,x_p\}$ of V called the *distinguished vertices* of G, that is usable to build any supergraph $G'(V' \supset V, E' \supset E)$ such that E' has no edge that connects $V \backslash X$ to $V' \backslash V$.

Definition 20 (Planar gadget). A *planar gadget* is a gadget $G(V,E)$ with an ordered list of distinguished vertices $X = (x_1,\dots,x_p)$, that is provided with a fixed embedding of V in the plane such that (1) no two edges in E cross each other, and (2) $x_1,\ldots,x_p$ all lie on the boundary of the outer face of G in the clockwise order specified by the ordered list X.

Definition 21 (Local states and configurations). Let G be a gadget with distinguished vertices $x_1,\ldots,x_p$ and Π be a vertex k-coloring problem. A *local state* of G is a satisfying k-coloring of G, and a *configuration* is the restriction of a local state to $x_1,\ldots,x_p$. Notice that one configuration may generally expand into several distinct local states.

Definition 22 (Parsimonious and weakly parsimonious gadget). Let G be a gadget and Π be a vertex k-coloring problem. G is a *weakly parsimonious* gadget if there exists a constant $s \neq 0$ such that each satisfying configuration C of G for Π expands to exactly s distinct local states. The gadget is *parsimonious* if and only if $s = 1$, i.e., if there is a one-to-one correspondence between its configurations and its local states.

3. Sketch of the reductions and their main tools

Our reductions from (PLAN)-1/3-SAT to (PLAN)-3-COL and from 3-COL to PLAN-3-COL are rather tricky and involve sophisticated gadgets. Therefore, we first present a simplified high-level view of the behaviors of our main gadgets and of the whole reductions. Fortunately the principles of the construction are modular and rather simple.

3.1. From PLAN-1/3-SAT to PLAN-3-COL

We want to design a planarity-preserving and parsimonious P-time reduction R from 1/3-SAT to 3-COL with the palette $P_3 = \{white, gray, black\}$. In the rest of this paper, *black* and *gray* are both called *dark colors*. Similarly, *white* and *gray* are both called *light colors*.

The main task is to design a 3-COL gadget to simulate a 1/3-SAT clause, i.e., a positive clause of length three that constrains *exactly one* of its variables to be assigned true. Each of the three variables of the clause will be represented by one vertex. However, we must find a correspondence between Booleans (2-states objects) and colors (3-states objects). We choose that *dark* colors (i.e., *black* and *gray*) represent *false*, and *white* represents *true*. Furthermore, for the sake of parsimony, the *false* should always be represented by *black* in the “user interface”, i.e., in the distinguished vertices of the 1/3-SAT clause simulator, whereas *gray* may appear in the “implementation side” of the gadget.

So, we need an object such as in Fig. 2 that implements this feature of substitutability between dark colors with respect to the false value. This gadget, called *dark one-way color-converter* binds two distinguished vertices x and y, with x being on the “interface side” (i.e., x cannot be *gray*), and y being on the “implementation side” (i.e., y may be *gray*). For any satisfying 3-coloring C, this converter coerces $C(x)$ and $C(y)$ to

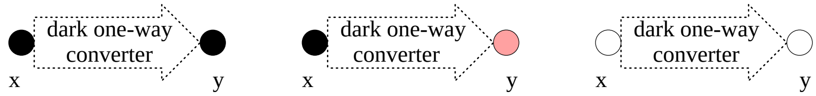


Fig. 2. Mapping Boolean values to colors by substitutability of dark colors.

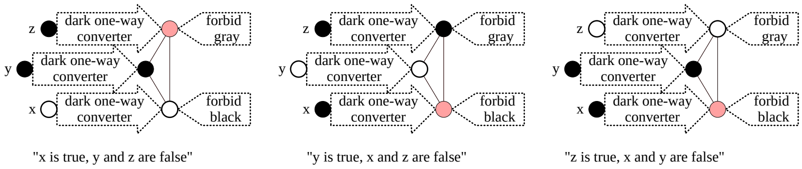


Fig. 3. Simulating a 1/3-SAT clause.

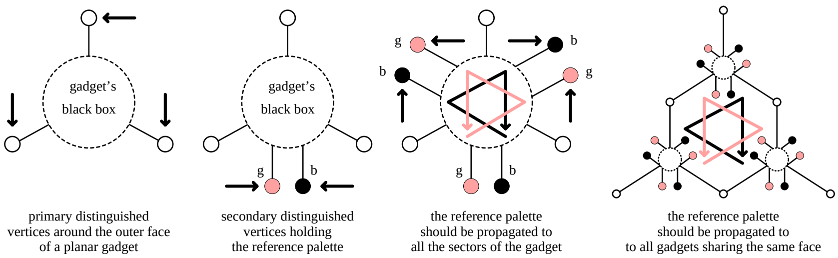


Fig. 4. Propagating the reference palette through and between gadgets.

represent equivalent Boolean values, i.e., the gadget must exactly allow the configurations $(C(x), C(y))$ that are either $(black, black)$ or $(black, gray)$ or $(white, white)$.

A parsimonious implementation of the *dark one-way color-converter* will give us the high-level scheme to implement parsimoniously the 1/3-SAT clause simulator in the way of Fig. 3. The reader can easily check that exactly one of the vertices x, y, and z must be *white* and that the two other vertices must be *black*.

The fact that colors play asymmetrical roles (there are two *dark* colors to represent the *false* and only one *white* color to represent the *true*) requires that the implementations of all the gadgets must use this palette convention. It means that we will have to connect each gadget to two *secondary distinguished vertices* b and g holding resp. the *black* and the *gray* representations of the *false* and lying in the sector formed by two consecutive *primary distinguished vertices*. The additional requirement that planarity be preserved will complicate the design of our gadgets since they will play an additional role beside their primary behaviors: each gadget should propagate the reference palette—i.e., the colors held in the pair (b, g)—to all the other sectors of the gadget (into vertices also named b and g for simplicity), as shown in Fig. 4. This way, gadgets lying in the vicinity of another gadget can use its reference palette if needed and propagate it further themselves. In order that all the gadgets follow the

Fig. 5. Substitutability of light colors.

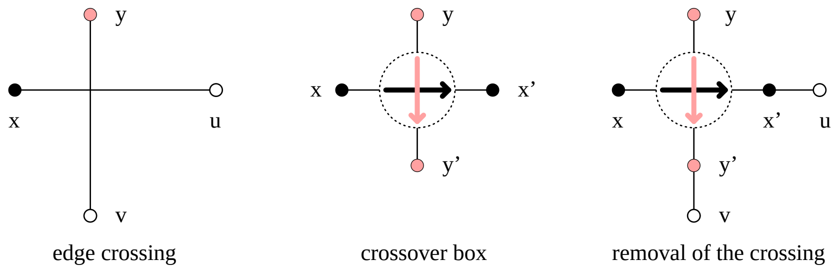


Fig. 6. Resolving edge crossings with crossover-boxes.

same convention, we decide that in each sector of a gadget, g will follow b for the clockwise order around the gadget.

Interestingly, the need of a connection to a reference palette allows to create gadgets with new behaviors at no cost just by changing the content of the palette. This is particularly true for color conversion. Normally, a dark color converter is always connected to a pair (b, g) holding the dark reference palette $(black, gray)$. However, if we decide to store the light colors $(white, gray)$ instead, then we obtain a new gadget, the *light one-way color-converter*, allowing a substitutability between light colors as depicted in Fig. 5 (to be compared to Fig. 2). This behavior will be used in the next reduction.

3.2. From 3-COL to PLAN-3-COL

Finding a parsimonious reduction 3-COL to PLAN-3-COL essentially consists in exhibiting a parsimonious crossover-box to resolve edge crossings as depicted in Fig. 6: a crossing between edges (x, u) and (y, v) is resolved by replacing the two edges by a crossover with distinguished vertices x, y, x', and y' and by creating two edges (x', u) and (y', v). The behavior of a crossover-box can be defined as follows:

Definition 23 (Crossover-box). A *crossover-box* for a vertex k-coloring problem is a planar gadget G with a list of four primary distinguished vertices (x, y, x', y') such that (1) for any local state C, $C(x) = C(x')$ and $C(y) = C(y')$, and (2) for any two colors $C_x, C_y \in P_k$ (possibly equal), there exists a local state C such that $C(x) = C_x$ and $C(y) = C_y$.

Crossover-boxes for 3-COL that exist in the literature are weakly parsimonious at best: the standard crossover-box one finds in the complexity books [14,11,7] is not

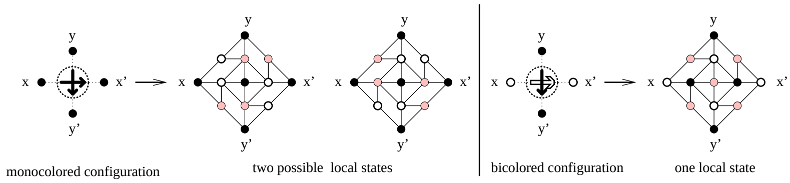


Fig. 7. Standard non-parsimonious crossover-box for PLAN-3-COL.

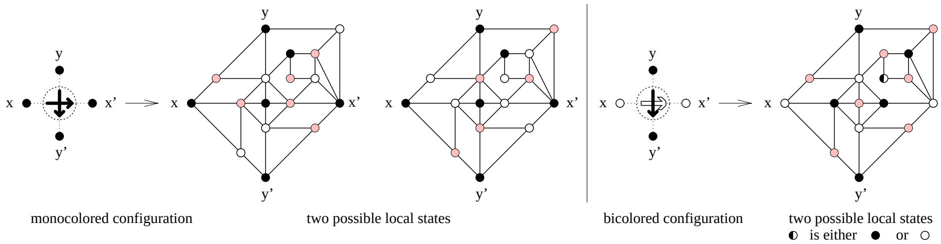


Fig. 8. Hunt et al.'s weakly parsimonious crossover-box for PLAN-3-COL.

parsimonious and not even weakly parsimonious, as shown in Fig. 7: each configuration coloring x and y with the same color expands into two local states whereas each configuration coloring x and y with distinct colors expands into one local state.

This crossover-box was improved by Hunt et al. to make it weakly parsimonious [10]: as shown in Fig. 8, each configuration expands into two local states, whether it colors x and y with the same colors or not. This implies that reducing a 3-COL instance with n vertices and c edge crossings to a PLAN-3-COL instance by using c crossover-boxes will multiply the number of solutions by 2^c where c may be as large as $\Theta(n^2)$.

A parsimonious crossover-box for PLAN-3-COL is hard to construct directly, so we will not propagate the colors $C(x)$ rightwards and $C(y)$ downwards *directly*. Instead, we will proceed in three steps as shown in Fig. 9.

First, using a gadget called the *prism*, we decompose the colors $C(x)$—resp. $C(y)$—into two pure colors, stored in vertices $low(x)$ and $high(x)$—resp. $low(y)$ and $high(y)$. Our two pure colors are *black* and *white*. Therefore,

- *gray* is seen as a composition of *black* and *white*,
- *white* decomposes into *white* and *white*,
- *black* decomposes into *black* and *black*.

By interpreting a *black*/*white* vertex as a bit set on/off, the action of the prism on vertex x can be seen as the writing in binary of its color $C(x)$ as the couple $(C(high(x)), C(low(x)))$. As Fig. 10 shows, a prism is a simple application of the *dark one-way color-converter* to obtain $low(x)$ and the *light one-way color converter* to obtain $high(x)$.

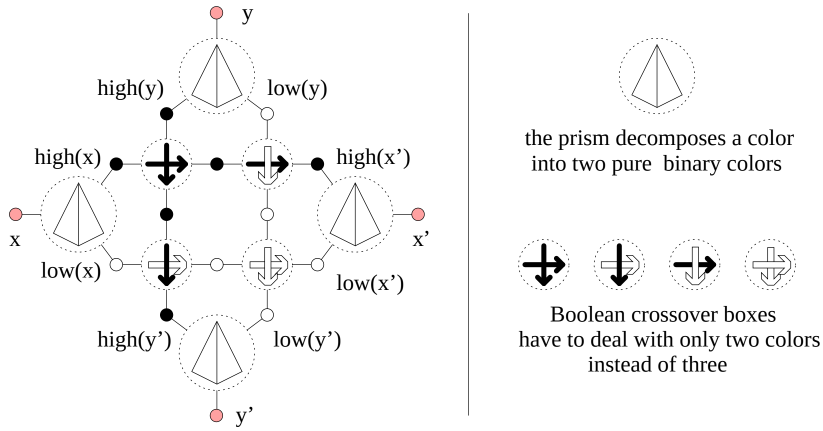


Fig. 9. Implementing a parsimonious crossover-box using prisms.

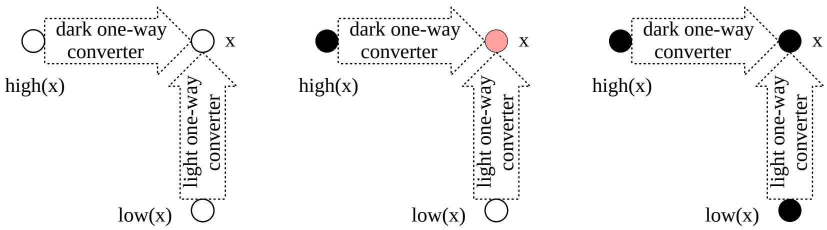


Fig. 10. Implementing a prism with one-way color converters.

Secondly, *high*(x) and *low*(x)—resp *high*(y) and *low*(y)—are the colors we will propagate rightwards—resp. downwards—to recompose them as a third step into $C(x')$ —resp. $C(y')$—by using the prism again. These vertical and horizontal propagations will generate four edge crossings instead of the one we tried to resolve initially, but since the propagated information is now Boolean (one color, *gray*, has temporarily vanished), we expect that a parsimonious resolution of the edge crossings will be easier, by introducing a new object: the Boolean crossover-box.

Definition 24 (Boolean crossover-box). A *Boolean crossover-box* for 3-COL is a planar gadget G with at least one secondary distinguished vertex g and a list of four primary distinguished vertices (x, y, x', y') such that (1) for any local state C, we have $C(x) = C(x') \neq C(g)$ and $C(y) = C(y') \neq C(g)$, (2) for any color C_g and any two colors C_x, C_y that are distinct of C_g, there exists a local state C such that $C(x) = C_x$, $C(y) = C_y$ and $C(g) = C_g$.

It turns out that our Boolean crossover-box will be parsimoniously implemented by using essentially four one-way color converters: two dark ones and two light ones.

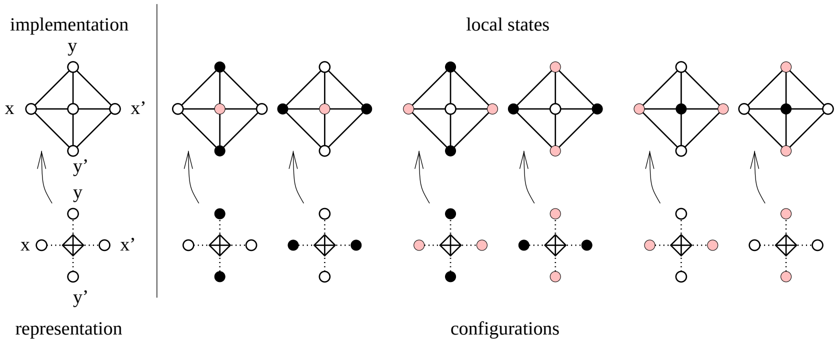


Fig. 11. Exclusive crossover-box.

4. The reduction from PLAN-1/3-SAT to PLAN-3-COL

We now address the details of our reductions and the implementation of our gadgets. We first design the gadget that will propagate the reference palette between the gadgets sharing the same face as explained in the sketch.

4.1. Exclusive crossover-box and pair-duplicator

Recall from Fig. 4 that when propagating the reference palette, the propagation of the gray color crosses the propagation of the black color. However, We do not need a real crossover-box here, because we know that the two colors to propagate are different. This introduces the definition of a new object, namely the *exclusive crossover-box:*

Definition 25 (Exclusive crossover-box). An *exclusive crossover-box* for a vertex k-coloring problem is a planar gagdet G with four distinguished vertices (x, y, x', y') such that (1) for any local state C, $C(a') = C(a) \neq C(b') = C(b)$, and (2) for any two colors C_a, $C_b \in P_k$ such that $C_a \neq C_b$ there exists a local state C such that $C(a) = C_a$ and $C(b) = C_b$.

An exclusive crossover-box is trivially implemented by the diamond depicted in Fig. 11. The reader can easily check that it exactly allows the six configurations drawn, and one configuration corresponds to one local state, i.e., that the gadget is parsimonious. In further figures, the exclusive crossover-box will be symbolized by ⌖) Chaining several exclusive crossover-boxes on a path or a cycle as shown in Fig. 12 will allow us to duplicate a (b, g) pair into as many copies as we need, and thus will allow us to propagate the reference palette along the inner boundary of a face. Such a cycle is called a *pair-duplicator* and will be symbolized by ※ in further figures).

We now address the implementation of color-converters. As a first step, the converters will neither be one-way nor propagate the reference palette from sector to sector.

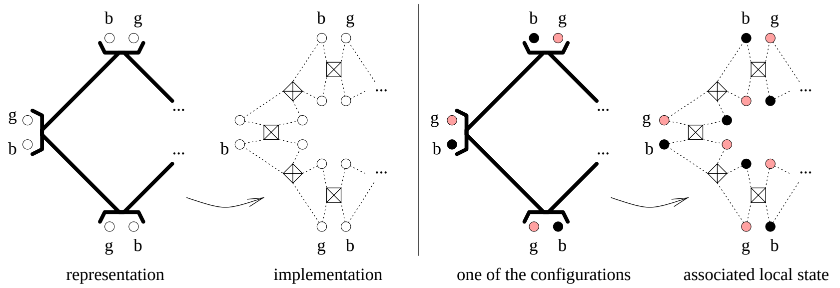


Fig. 12. Pair-duplicator.

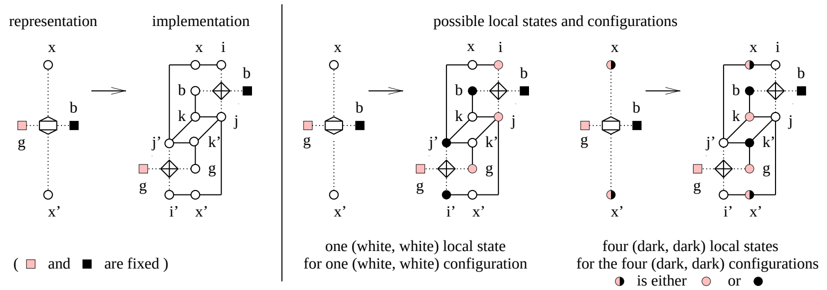


Fig. 13. Two-way (dark) color-converter.

Indeed, each color of the palette will lie in a different sector. This will be corrected as a second step.

4.2. The two-way color-converter

This gadget is depicted in Fig. 13. Its list of distinguished vertices is (x, b, x', g), where b and g are supposed to hold the two distinct colors of the reference palette, i.e., resp. *black* and *gray* if we want a dark converter, or resp. *white* and *gray* if we want a light one, as explained in the sketch. The gadget is parsimonious and its configurations are all the 3-colorings C where $C(x)$ and $C(x')$ are *equivalent* colors with respect to the reference palette. More precisely, if, say, $C(b) = black$ and $C(g) = gray$, the reader can easily check that all the possible configurations $(C(x), C(x'))$ are exactly the $(white, white)$ configuration and the four $(dark, dark)$ configurations, i.e.: $(black, black)$, $(gray, gray)$, $(gray, black)$ and $(black, gray)$. This converter is said *two-way*, because $C(x')$ does not determine $C(x)$ in a $(dark, dark)$ configuration, and conversely. In further figures, it is represented with the notation.

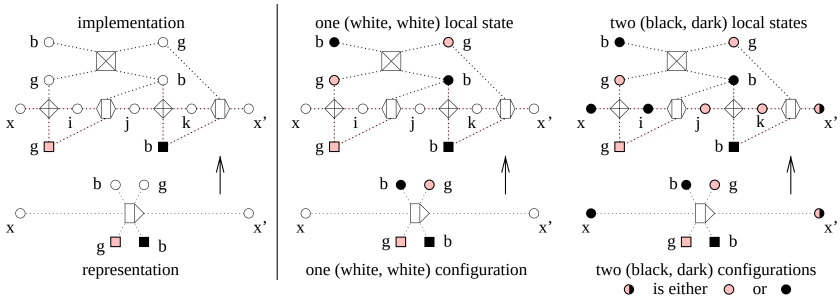


Fig. 14. One-way color-converter.

We now use the two-way color-converter to implement a one-way color-converter that will furthermore propagate the reference palette through the line (x,x').

4.3. The one-way color-converter

This gadget is depicted in Fig. 14. It has two distinguished vertices x,x' plus two pairs (b,g) lying in each of the two sectors defined by the line (x,x'). Note how the reference palette is propagated: w.l.g. assume that the square vertices b and g lying beneath the line (x,x') hold resp. *black* and *gray*. Then the three exclusive crossover-boxes recopy *black*—resp. *gray*—in all other round vertices b—resp. g. All the pairs (b,g) now hold the colors making both two-way colors converters behave as dark color-converters. The reader can then easily check that the gadget is parsimonious and its configurations are all the 3-colorings C such that $C(x)\neq C(g)$ and $C(x)$ is equivalent to $C(x')$ with respect to the reference palette. More precisely, with $C(b)=black$ and $C(g)=gray$, the four possible configurations $(C(x),C(x'))$ are $(white, white)$, $(black, gray)$, and $(black, black)$.

In further figures, this gadget will be represented by the ⊳ notation.

4.4. The 1/3-clause simulator

This gadget is depicted in Fig. 15. It has three distinguished vertices x,y,z embedded clockwise in this order plus three pairs (b,g), each lying in one the three sectors defined by the vertices x, y, z. W.l.g., assume that one of the three pairs (b,g) is colored $(black, gray)$, e.g., the one lying in the bottom sector. Then, these colors are propagated to all the other pairs (b,g) through the pair-duplicators and the one-way color-converters. Therefore, these latter behave as dark color-converters. It is not difficult to check that there are only three ways to color the triangle (i,j,k), each one coloring either i, j, or k in *white* and the two other vertices of the triangle in *dark*. The three dark one-way color-converters ensure that the *gray* vertex of the triangle will be converted into *black* (the converter connecting i to x is useless because i is never gray but is left for the sake of uniformity). Thus, the only configurations are the three 3-colorings C such that $(C(x),C(y),C(z))$ is $(black, white, black)$ or $(black, black, white)$

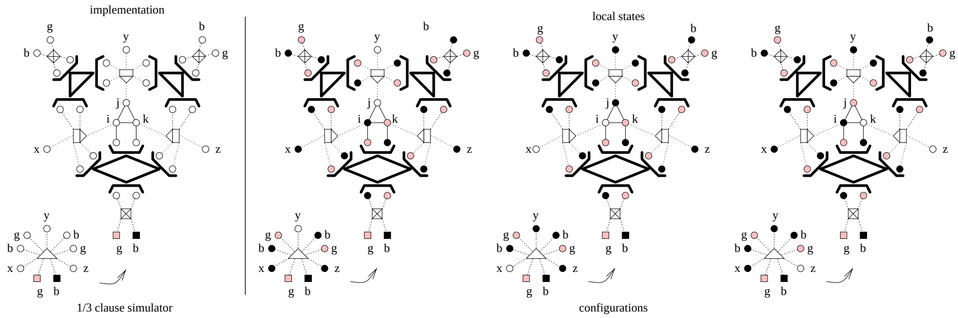


Fig. 15. 1/3-clause simulator.

or $(white, black, black)$, i.e., the gadget parsimoniously simulates a planar 1/3-clause (x, y, z) with the identification $white = true$ and $black = false$. In further figures, the 1/3-clause simulator will be represented by the ⩚ notation.

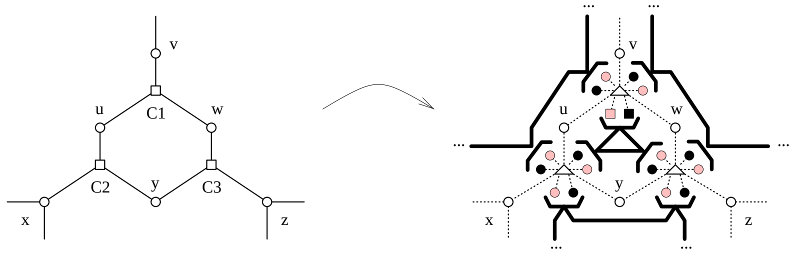
4.5. The reduction itself and its proof

We address the reduction from PLAN-1/3-SAT to PLAN-3-COL. Let $\varphi(V, L)$ be a PLAN-1/3-SAT instance and $G(V \cup L, E)$ be a planar formula-graph associated to φ along with an arbitrary planar embedding (we suppose G is connected for the sake of simplicity). We create a 3-COL instance G' from G that preserves the planarity of G and with the same number of solutions:

(1) for each variable-vertex $v \in V$, create a vertex x_v;
(2) for each clause-vertex $c \in L$ with (c,i), (c,j), $(c,k) \in E$, create a 1/3-clause simulator s_c with distinguished vertices $x = x_i$, $y = x_j$ and $z = x_k$. If i, j, k are in clockwise order around c for the chosen embedding, then x, y, and z should be also in clockwise order around s_c;
(3) There are now a total of $3|L|$ pairs (b, g) among the distinguished vertices of the 1/3-clause simulators. We now want all the simulators to share the same reference palette: let F' be the set of faces of G' corresponding to the set of faces F of G for the chosen embedding. For each face $f \in F'$ create a pair-duplicator embedded in f by chaining all the pairs (b, g) lying in f (see Fig. 16).

(To reduce 1/3-SAT to 3-COL, replace the second step by a simple fusion of the $3|L|$ secondary distinguished vertices b—resp. vertices g—connected to the 1/3-clause simulators into a single vertex b—resp. g.)

The construction is a parsimonious reduction from PLAN-1/3-SAT to PLAN-3-COL: let (b_0, g_0) be an arbitrary pair among the pairs (b, g) connected to one of the 1/3-clause simulators in the graph G'. Since b_0 and g_0 share a common pair-duplicator, we necessarily have $C(b_0) \neq C(g_0)$ for any 3-coloring C. Counting the non-isomorphic 3-colorings C is equivalent to counting the ones that verify $C(b_0) = black$ and $C(g_0) = gray$. Since the input graph G is connected, so is G', and $(black, gray)$ propagates to all the pairs (b, g) via the pair-duplicators inside each face and via the



three 1/3-clauses C1, C2, C3 over variables u, v, w, x, y, z

implementation

Fig. 16. Reduction from PLAN-1/3-SAT to PLAN-3-COL.

1/3-clause simulators across the faces. By the parsimonious behavior of the 1/3-clause simulator in presence of a $(black, gray)$ reference palette, all the vertices x_i, x_j, x_k sharing a 1/3-clause simulator must be either *white* or *black* and exactly one must be *white*. Thus, there is a bijection between the set of assignments I satisfying φ and the set of 3-colorings C such that $C(g_0) = gray$ and $C(b_0) = black$, i.e. a set of non-isomorphic 3-colorings, with the correspondence $I(v) = true \Leftrightarrow C(x_v) = white$ and $I(v) = false \Leftrightarrow C(x_v) = black$. $\square$

5. The reduction from 3-COL to PLAN-3-COL

We first address the implementation of the *prism* that will be used to build our parsimonious crossover-box as explained in the sketch. This will be the first time that we will need a color-converter behaving as a light one. Also, we will need the prism in two symmetric embeddings.

5.1. The prism

This gadget is depicted in Fig. 17. It has three primary distinguished vertices x, $low(x), high(x)$ embedded in this clockwise order around the gadget, plus three pairs of vertices (b, g), each lying in one of the three sectors of the gadget. Let C be a satisfying 3-coloring for the prism and assume w.l.g. that an arbitrary pair among the pairs (b, g) holds $(black, gray)$, e.g., the pair of square vertices lying in the bottom sector. This reference palette propagates as usual except for the Eastern color-converter: this is $(white, gray)$ that is propagated instead by this gadget to the Eastern pair (w, g), hence making it behave as a light color-converter as opposed to the Western dark converter. It follows that: $C(x) = black$ implies $(C(high(x)), C(low(x))) = (black, black)$, $C(x) = gray$ implies $(C(high(x)), C(low(x))) = (black, white)$, $C(x) = white$ implies $(C(high(x)), C(low(x))) = (white, white)$, and the gadget is parsimonious. The prism will be represented by the notation in further figures. A "mirrored" prism where $low(x)$ follows $high(x)$ for the clockwise order is similarly designed and will be represented by the notation.

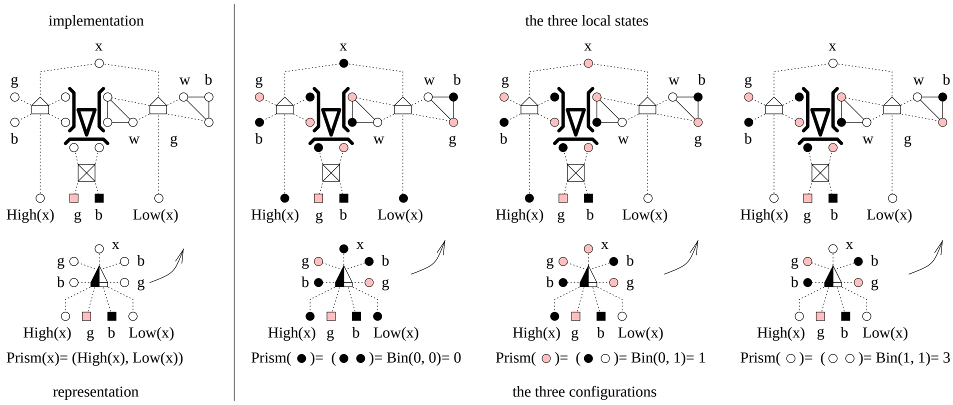


Fig. 17. Prism (here, $high(x)$ follows $low(x)$ for the clockwise order).

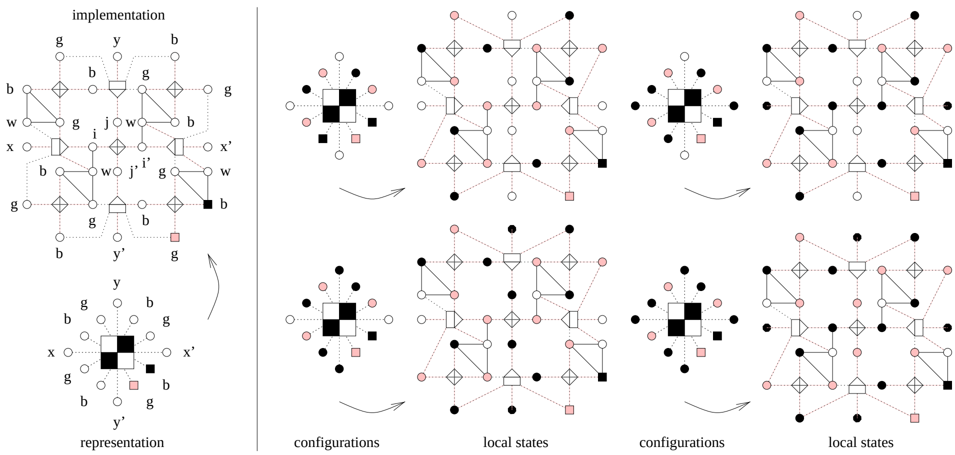


Fig. 18. Boolean crossover-box.

Combining an exclusive crossover-box with two dark color-converters and two light color-converters now gives us the Boolean crossover-box needed to cross each other the binary components of two colors decomposed by the prism. Connected to a pair (b,g) holding $(black, gray)$ for a 3-coloring C, the Boolean crossover-box is able to cross any two (distinct or non distinct) non-gray colors.

5.2. The Boolean crossover-box

A parsimonious implementation of a Boolean crossover-box with palette $P_2 = \{white, black\}$ is depicted in Fig. 18. It has four primary distinguished vertices x, y, x', y' embedded clockwise in this order, plus four pairs (b,g), each one lying in one of the four sectors of the gadget. Let C be a satisfying 3-coloring and assume, w.l.g, that

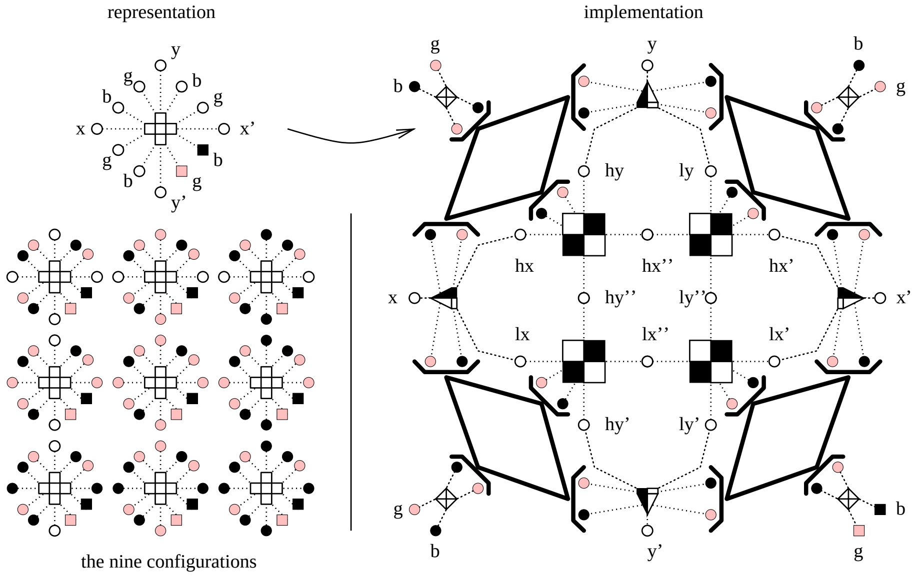


Fig. 19. Unrestricted crossover-box.

one of the pairs (b, g) holds $(black, gray)$, e.g., the two square vertices. The gadget is parsimonious and its behavior matches Definition 24. In further figures, it will be represented by the notation.

We have now all the necessary tools to parsimoniously implement our unrestricted crossover-box crossing any two colors (distinct or not) among three.

5.3. The unrestricted crossover-box

This gadget is depicted in Fig. 19. It has four distinguished vertices x, y, x, y' embedded clockwise in this order plus four pairs (b, g), each one lying in one of the four sectors of the gadget. Let C be a 3-coloring of the gadget, and assume w.l.g., that one of the pairs (b, g) holds $(black, gray)$, e.g., the pair of square vertices. This reference palette is propagated as usual through the exclusive crossover-boxes, the pair-duplicators and the prisms, such that all round pairs (b, g) also hold $(black, gray)$: thus, prisms decompose colors into *black* and *white* and the four central Boolean crossover-boxes cross binary $(black/white)$ colors. Note that the two prisms connected to x and y' have the mirrored embedding so that their *high* and *low* slots face the respective ones of the non-mirrored prisms connected to x' and y. Thus the prism connected to x'—resp. y—correctly recomposes the colors decomposed by the prism attached to x—resp. y'—and propagated through the two Boolean crossover-boxes lying in-between. If follows that for a given reference palette, the built gadget has exactly the 9 expected configurations depicted on the left of Fig. 19, each one corresponding to one local state. In further figures, the unrestricted crossover-box will be represented by the notation.

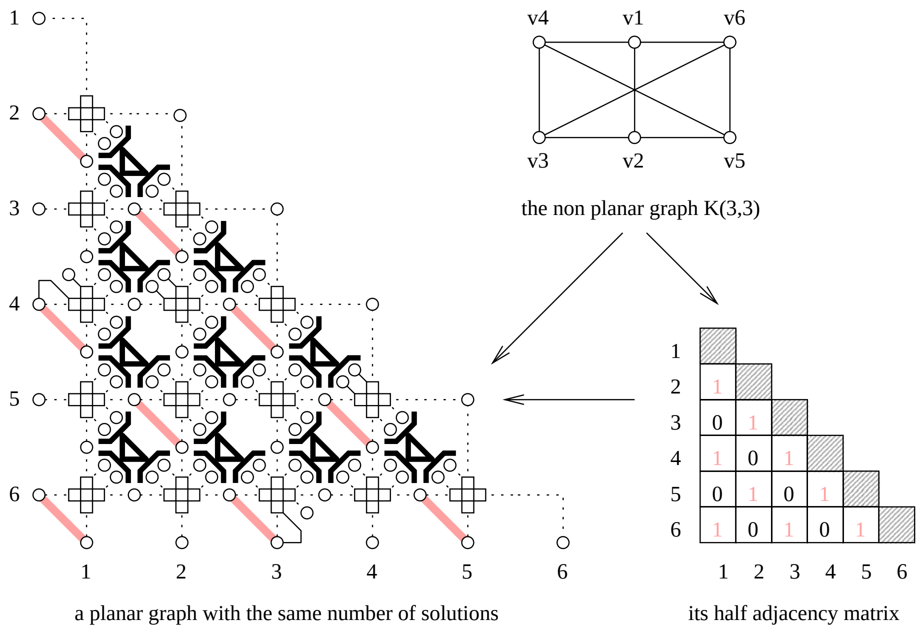


Fig. 20. Parsimonious reduction from 3-COL to PLAN-3-COL.

5.4. *The reduction itself*

Let $G(V=\{v_1,\dots,v_n\},E)$ be a non-planar graph. We want to compute in P-time a planar graph $G'(V',E')$ with the same number of 3-colorings. Let M be the lower-left half adjacency matrix of G. M has $(n(n-1))/2$ entries $M_{i,j}$, $1\leqslant j<i\leqslant n$, with $M_{i,j}=1$ if and only if $(v_i,v_j)\in E$. The embedding of G' will follow the physical grid T of M drawn in the plane (see Fig. 20).

Inside each square $T_{i,j}$ of an entry $M_{i,j}$, create an unrestricted crossover-box $B_{i,j}$, with one primary distinguished vertex embedded on each side of the square. Two crossover-boxes sharing the common side of two squares also share the distinguished vertex lying on this side. Also, any two crossover-boxes $B_{i+1,i}$ and $B_{i,i-1}$, $1<i<n$, share resp. their Northern and Eastern distinguished vertices.

This ensures that the distinguished vertices in the row x_k agree with the ones on the line y_k, so that they are all representants of the vertex v_k. Therefore, for each edge $e=(v_i,v_j)\in E$, $i>j$, one can create an edge (u,v) in G' to simulate e without breaking planarity with u and v being, resp., the Eastern and Southern distinguished vertices of the crossover-box $B_{i,j}$, since u and v are representants of, resp., v_i and v_j.

We now want that those crossover-boxes share the same reference palette: for any three crossover-boxes $B_{i,j}$, $B_{i+1,j}$, $B_{i,j+1}$, connect the north-eastern pair (b,g) of $B_{i,j}$, the south-eastern one of $B_{i+1,j}$, and the south-western one of $B_{i,j+1}$ with a common pair-duplicator.

Now, for a given reference palette, say $(gray, black)$, there are obviously exactly as many 3-colorings in G' as in G. But there are six possible reference palettes and we may choose one independently of the simulated 3-coloring, i.e., the 3-coloring of the primary distinguished vertices. So, G' has six times the required number of solutions. We remove these unwanted duplicates by making the reference palette dependent of the simulated 3-coloring: choose an arbitrary edge $(v_i, v_j) \in E$ (this is (v_4, v_3) in Fig. 20), and let $B_x = B_{n,j}$ and $B_y = B_{i,1}$. Merge the southern distinguished vertex of B_x with the vertex g of its south-eastern pair (b, g), and merge the western distinguished vertex of B_y with the vertex b of its north-western pair (b, g) (these are resp. $i = 4$ and $j = 3$ in Fig. 20). There are now exactly as many solution in G' and in G.

6. Conclusion

In this paper, we have proved that 3-COL and PLAN-3-COL are *parsimoniously* equivalent to the problem SAT, and hence, also capture accurately the structure of the space of the solutions of any problem in NP. This also yields new DP-completeness results under random P-time reductions for 3-COL and PLAN-3-COL.

Finally, it is interesting to note that our parsimonious reductions, from 1/3-SAT (or SAT) to 3-COL on one hand, and from PLAN-1/3-SAT (or PLAN-SAT) to PLAN-3-COL on the other hand, are computed in linear time on RAMs, so they form a sequel to the results of [6,3,8,1,2] on linear reductions.

Appendix A. Proofs of the behaviors of the gadgets

A.1. Proof of the behavior of the two-way color-converter

Let C be a 3-coloring for the two-way color converter and assume w.l.g. that the square vertices b and g hold resp. $black$ and $gray$ (see Fig. 13). Then by the properties of the exclusive crossover-box, the round vertices b and g also hold resp. $black$ and $gray$, furthermore $C(i) = C(j) \neq black$, and $C(i') = C(j') \neq gray$. There are now two cases:

- suppose $C(i) = C(j) = gray$ (leftmost case in Fig. 13). Then, $C(k) = white$ and $C(i') = C(j') = black$, and it follows that $C(x) = C(x') = white$.
- suppose $C(i) = C(j) = white$, (rightmost case in Fig. 13). Then $C(k) = gray$, $C(k') = black$, $C(i') = C(j') = white$, and it follows that $C(x) = dark$ and $C(x') = dark$, i.e., x and x' can be $black$ or $gray$, independently of each other. □

A.2. Proof of the behavior of the one-way color-converter

Let C be a 3-coloring for the one-way color-converter, and assume w.l.g that one of the pairs (b, g) holds ($black$, $gray$), e.g., the pair of square vertices (see Fig. 14). Then, by the properties of the exclusive crossover-box, all other round pairs (b, g) also hold $(black, gray)$ and furthermore $C(j) = C(k) \neq black$ and $C(x) = C(i) \neq gray$.

Thus, both two-way color-converters behave as dark-converters. There are now two cases:

- suppose $C(x) = white$ (leftmost case in Fig. 14). Then $C(x') = C(k) = C(j) = white$, by the properties of the exclusive crossover-box and the two-way dark-converter.
- suppose $C(x) = black$ (rightmost case in Fig. 14). Then $C(i) = black$, $C(j) = C(k) = gray$, and finally $C(x') = dark$ by the property of the two-way dark-converter, i.e., x' can be either black or gray. $\square$

A.3. Proof of the behavior of the PLAN-1/3-SAT clause simulator

Let C be a 3-coloring for the clause simulator, and assume w.l.g that one of the pairs (b, g) holds (*black*, *gray*), e.g., the pair of square vertices (see Fig. 15). These colors propagate through the exclusive crossover-boxes, the pair-duplicators and the one-way color-converters, and finally $C(g) = gray$ and $C(b) = black$, for all round pairs (b, g). Thus, all the one-way converters behave as dark-converters and one of those pairs coerces $C(i) \neq gray$ and $C(k) \neq black$. There are now three cases, depending on the color of j:

- suppose $C(j) = white$ (leftmost case in Fig. 15), then $C(i) = black$, $C(k) = gray$, and finally $C(x) = black$, $C(y) = white$, $C(z) = black$, by the property of the one-way dark-converter.
- suppose $C(j) = black$ (central case in Fig. 15), then $C(i) = white$, $C(k) = gray$, and finally $C(x) = white$, $C(y) = black$, $C(z) = black$, by the property of the one-way dark-converter.
- suppose $C(j) = gray$ (rightmost case in Fig. 15), then $C(k) = white$, $C(i) = black$, and finally $C(x) = black$, $C(y) = black$, and $C(z) = white$, by the property of the one-way dark-converter. $\square$

A.4. Proof of the behavior of the prism

Let C be a 3-coloring for the prism, and assume w.l.g. that one of the pairs (b, g) holds $(black, gray)$, e.g., the pairs of square vertices (see Fig. 17). Then, these colors first propagate in the central part of the gadget through the exclusive crossover-box and the pair-duplicator. It also propagates through the Western one-way color-converter. The central vertex w is then *white* since it shares a triangle with a pair (b, g). Thus, the Eastern one-way color-converter is connected to a pair holding $(white, gray)$, and these colors are propagated through the converter to the Eastern pair (w, g). Thus, the Eastern vertex b sharing a triangle with this pair (w, g) is *black*, and finally, $(black, gray)$ has been propagated to all round pairs (b, g). Also notice that the Western color-converter behaves as dark converter while the Eastern one behaves as a light converter. Both color-converters are one-way and hence $high(x)$ and $low(x)$ cannot be gray. There are now three cases depending on $C(x)$:

- suppose $C(x) = black$ (leftmost case in Fig. 17). Then the dark converter outputs $C(high(x)) = black$, and the light converter outputs $C(low(x)) = black$.
- suppose $C(x) = gray$ (central case in Fig. 17). Then the dark converter outputs $C(high(x)) = black$, and the light converter outputs $C(low(x)) = white$.

- suppose $C(x) = white$ (rightmost case in Fig. 17). Then the dark-converter outputs $C(high(x)) = white$, and the light converter outputs $C(low(x)) = white$. □

A.5. Proof of the behavior of the Boolean crossover-box

Let C be a 3-coloring for the Boolean crossover-box and assume w.l.g, that one of the pairs (b,g) holds $(black, gray)$, e.g., the pairs of square vertices (see Fig. 18). The reference palette is propagated to all pairs (b,g) via exclusive crossover-boxes, pair-duplicators, and one-way color-converters. Moreover, for all vertices named w, $C(w) = white$ since they share a triangle with b and g. Note that both color-converters on the vertical line (y, y') behave as light-converters while both color-converters on the horizontal line (x, x') behave as dark-converters. The four color-converters are one-way, and hence x, y, x', y' cannot be gray. Also, i and i' cannot be white because they are adjacent to w. Now, there are two main cases depending on $C(x)$:

- suppose $C(x) = white$ (leftmost cases in Fig. 18). Then the leftmost light-converter outputs $C(i) = gray$, the exclusive crossover-box outputs $C(i') = gray$, and the rightmost light-converter outputs $C(x') = white$. There are two subcases depending on $C(y)$:
  - suppose $C(y) = white$ (upper leftmost case in Fig. 18). Then the upper dark-converter outputs $C(j) = white$, the exclusive crossover-box outputs $C(j') = white$, and the lower dark-converter outputs $C(y') = white$.
  - suppose $C(y) = black$ (lower leftmost case in Fig. 18). Then the upper dark-converter outputs $C(j) = black$ (gray is excluded by the exclusive crossover-box). The exclusive crossover-box outputs $C(j') = gray$, and the lower dark-converter outputs $C(y') = black$.
- suppose $C(x) = black$ (rightmost cases in Fig. 18). Then the leftmost light-converter outputs $C(i) = black$, the exclusive crossover-box outputs $C(i') = black$, and the rightmost light-converter outputs $C(x') = black$. There are two subcases depending on $C(y)$:
  - suppose $C(y) = white$ (upper rightmost case in Fig. 18). Then the upper dark-converter outputs $C(j) = white$, the exclusive crossover-box outputs $C(j') = white$, and the lower dark-converter outputs $C(y') = white$.
  - suppose $C(y) = black$ (lower rightmost case in Fig. 18). Then the upper dark-converter outputs $C(j) = gray$ (black is excluded by the exclusive crossover-box). The exclusive crossover-box outputs $C(j') = gray$, and the lower dark-converter outputs $C(y') = black$. □

A.6. Proof of the behavior of the unrestricted crossoverbox

Let C be a 3-coloring for the unrestricted crossover-box, and assume w.l.g. that one of the pairs (b,g) holds $(black, gray)$, e.g., the pair of square vertices. Observe that $black$ and $gray$ are resp. propagated to all round vertices b and g via the exclusive crossover-boxes, pair-duplicators, prisms and Boolean crossover-boxes. $C(x)$ is decomposed by the leftmost (mirrored) prism into $C(l_x) = C(low(x))$ and $C(h_x) = C(high(x))$. Then $C(l_x)$ is propagated to l''_x and l'_x through the two lower

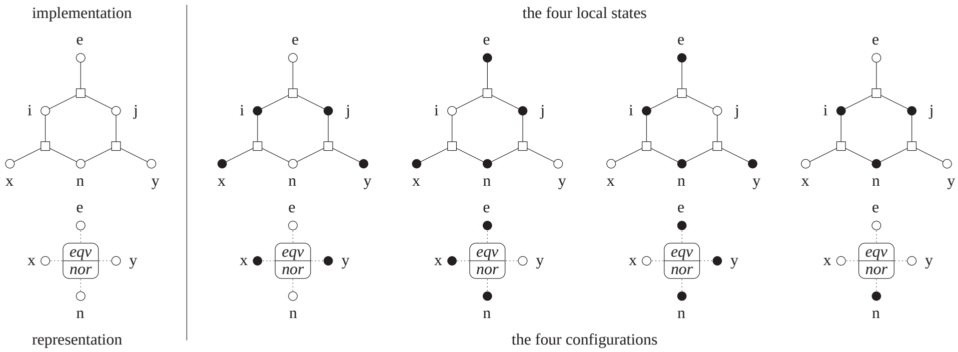


Fig. 21. The NOR–EQV operator.

Boolean crossover-boxes, and similarly $C(h_x)$ is propagated to h''_x and h'_x through the two upper Boolean crossover-boxes. But $l'_x = low(x')$ and $h'_x = high(x')$, the decomposition of $C(x')$ via the rightmost (non-mirrored) prism, and therefore $C(x) = C(x')$. Similarly, $C(low(y)) = C(l_y) = C(l'_y) = C(l''_y) = C(low(y''))$ and $C(high(y)) = C(h_y) = C(h'_y) = C(h''_y) = C(high(y''))$ and finally $C(y) = C(y')$. We conclude that the gadget has nine possible local states resp. corresponding to the nine configurations of Fig. 19. $\square$

Appendix B. The equivalence of (PLAN-)SAT and (PLAN-)1/3-SAT

In this appendix we briefly recall why (PLAN-)SAT and (PLAN-)1/3-SAT are parsimoniously reducible to each other.

Reducing (PLAN-)1/3-SAT to (PLAN-)SAT is trivial since any 1/3-clause (x, y, z) can be parsimoniously simulated with three 2-clauses and one 3-clause: $\neg x \vee \neg y$, $\neg y \vee \neg z$, $\neg z \vee \neg x$ and $x \vee y \vee z$. Moreover the planarity is preserved since to any 3-star of any simulated 1/3-clause corresponds the 3-star of its fourth simulating clause embedded in the hexagon formed by its first three simulating clauses.

In order to reduce (PLAN-)SAT to (PLAN-)1/3-SAT we build gadgets to simulate Boolean operators. Fig. 21 shows the gadget NOR-EQV (with variable vertices as rounds and 1/3-clause vertices as squares): this gadget, which four distinguished vertices x, e, y, n embedded clockwise in this order, is a “two in one”-gadget that simulates both the negated-or operator (NOR) and the equivalence operator (EQV). More precisely, x and y are the input vertices of the operator, e and n are the output vertices, and in any interpretation (1) $e = EQV(x, y)$, i.e. $e \Leftrightarrow (x \Leftrightarrow y)$, and (2) $n = NOR(x, y)$, i.e., $n \Leftrightarrow \neg(x \vee y)$.

Indeed, there are two cases:

- at least one of the three values i, j and n is *true* (three first local states in Fig. 21). The gadget is symmetric, so w.l.g. let n be *true* (first local state in Fig. 21). Then x and i—resp. y and j—are forced to be *false* because of the 1/3-clause (x, i, n)—resp. the 1/3-clause (y, j, n). Now, both i and j being false, it follows that e must be *true* because of the 1/3-clause (e, i, j). The two central local states are rotations of this first local state.

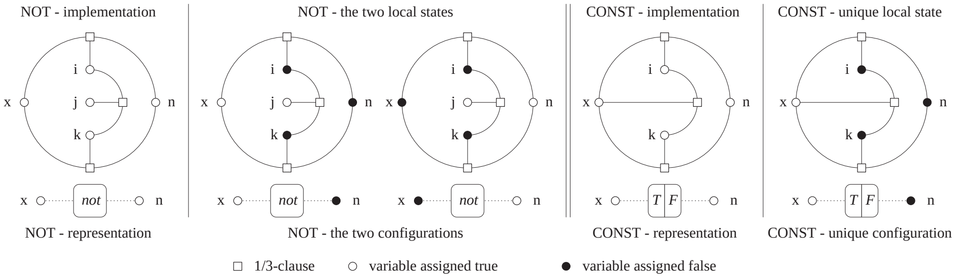


Fig. 22. The NOT operator and the CONST operator.

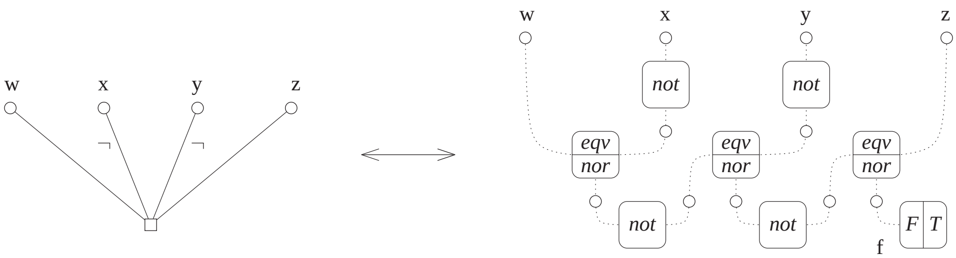


Fig. 23. Simulation of the clause $(w \vee \neg x \vee \neg y \vee z)$ with PLAN-1/3-SAT.

- All of the values i, j and n are $false$ (last local state in Fig. 21). Then the 1/3-clauses (x,i,n), (y,j,n), and (e,i,j) resp. coerce that x, y, and e be all *true*.

One can check that all *true*/$false$ combinations for x and y are possible, so n and e can be seen as functions of x and y with $n = NOR(x,y)$ and $e = EQV(x,y)$ as stated above.

Though the NOR operator is complete for propositional logic, we design two other operators in 1/3-SAT for more convenience. The NOT operator is shown on the left of Fig. 22: it has two distinguished vertices x and n and parsimoniously coerces that $n \Leftrightarrow \neg x$.

There are two cases:

- suppose j is $false$. Then, either i or k is *true* by the 1/3-clause (i,j,k). The gadget being symmetric, assume w.l.g. that i is *true* and k is $false$. By the 1/3-clause (x,i,n), it follows that both x and n are $false$, and finally k must be *true* by the 1/3-clause (x,k,n). A contradiction.
- so, j is always *true*, and i and k are always $false$. And both 1/3-clauses (x,i,n) and (y,j,n) coerces that exactly one of x and n is true, i.e., $n \Leftrightarrow \neg x$.

Note that since j is always true, merging the vertices x and j of the gadget NOT eliminates the configuration where x is $false$. This way, one obtains the gadget CONST shown on the right of Fig. 22. This gadget has two distinguished vertices t and f and only one configuration, with t being the constant *true* and f the constant $false$.

The operator OR is built by chaining the gadget NOR with the gadget NOT, and a clause of arbitrary length is built by chaining several OR operators as in Fig. 23,

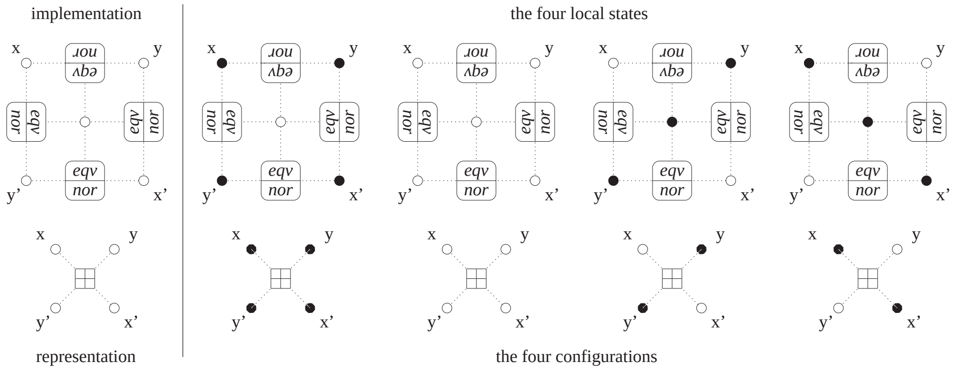


Fig. 24. Crossover-box for PLAN-1/3-SAT.

the terminal output vertex of the chain being connected to a gadget CONST, coercing the clause to be satisfied. The construction is obviously parsimonious and also preserves the embedding of the simulated clause, so the simulation of all the clauses of any (planar) input SAT instance will yield a (planar) 1/3-SAT instance with as many solutions. This completes the reduction from (PLAN-)SAT to (PLAN)-1/3-SAT.

Appendix C. The equivalence of 1/3-SAT and PLAN-1/3-SAT

In this section, we give direct arguments to show that 1/3-SAT and PLAN-1/3-SAT are parsimoniously reducible to each other: the reduction from PLAN-1/3-SAT to 1/3-SAT is the identity, and finding the converse reduction boils down to finding a parsimonious crossover-box. An implementation of this gagdet is obtained by connecting four gadgets NOR–EQV (as defined in the previous section) in the way of Fig. 24. This gadget has four distinguished vertices x, y, x', and x' embedded clockwise in this order. This gadget parsimoniously coerces any two distinguished vertices lying in opposite corners to be equivalent independently of the assignment of the other two distinguished vertices.

Note that the central vertex is connected to the *EQV* slot of all the gadgets NOR–EQV. So, there are two cases:

- the central vertex is *true*: then the vertices x, x', y, y' are all equivalent, i.e., either all $false$ or all *true*. This yields the two leftmost configurations ($false, false, false, false$) and ($true, true, true, true$) on Fig. 24.
- the central vertex is $false$: then x is not equivalent to y which itself is not equivalent to x'. So x and x' turn out to be equivalent. Similarly y and y' are equivalent, and this yields the two rightmost configurations ($false, true, false, true$) and ($true, false, true, false$) on Fig. 24.

Therefore, the gadget is a parsimonious crossover-box for PLAN-1/3-SAT, and can be used in the usual way to reduce parsimoniously 1/3-SAT to PLAN-1/3-SAT in quadratic time.

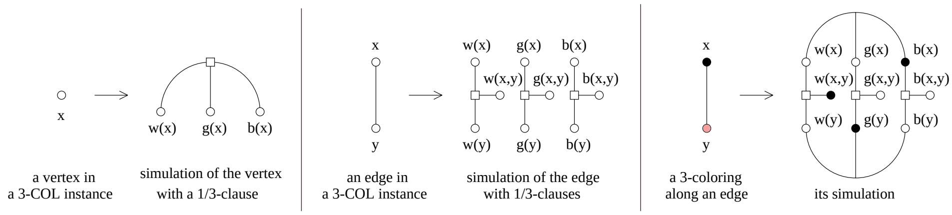


Fig. 25. Reduction from 3-COL to 1/3-SAT.

Appendix D. The reduction of (PLAN-)3-COL to (PLAN-)1/3-SAT

It is easy to reduce 3-COL to 1/3-SAT. The constraint that any vertex of the input graph is colored with exactly one color among three is conveniently simulated with 1/3-clauses: for each vertex x in the input graph, create three variables $w(x)$, $g(x)$ and $b(x)$—meaning that x is resp. colored *white*, *gray* and *black*—and create a 1/3-clause $(w(x), g(x), b(x))$. Now for a given color c and a given edge (x, y), we want that exactly one of the three *exclusive* cases holds:

- x has the color c,
- y has the color c,
- neither x nor y have the color c,

so the constraint that any two adjacent vertices have distinct colors is expressed by three 1/3-clauses per edges: for each edge (x, y) of the input graph, create the vertices $w(x, y)$, $g(x, y)$ and $b(x, y)$—meaning that neither x nor y are colored resp. *white*, *gray* and *black*—and create the 1/3-clauses $(w(x), w(x, y), w(y))$, $(g(x), g(x, y), g(y))$ and $(b(x), b(x, y), b(y))$. See Fig. 25. This nearly ends the parsimonious reduction from 3-COL to 1/3-SAT: we must remove the isomorphic solutions of the 3-COL instance. This is done by choosing an arbitrary edge (u, v) of the input graph and by forcing the 3-coloring of u and v to, say resp. *white* and *black*, which is simulated by connecting $w(u)$ and $b(v)$ to the distinguished vertex t of a gadget CONST, as defined in Appendix B.1.

The planarity is not preserved because of the explosion of each vertex into three variables, which makes the 1/3-clauses simulating distinct edges incident to x overlap. In order to reduce PLAN-3-COL to PLAN-1/3-SAT, one also creates three variables per vertex x—namely $w(x)$, $g(x)$, $b(x)$—connected by a 1/3-clause $(w(x), g(x), b(x))$, but we also duplicate them as many times as the degree $d(x)$ of vertex x into new variables $w_i(x)$, $g_i(x)$, $b_i(x)$, $1 \leqslant i \leqslant d(n)$, the ith 3-uple being denoted $slot_i(x)$ as a whole. The duplication is done by chaining the parsimonious crossover-box for PLAN-1/3-SAT (see Appendix C) as in Fig. 26. Note that the clockwise order for $slot_i(x)$ is $b_i(x)$, $g_i(x)$, and then $b_i(x)$.

Now, each edge (x, y) of the input graph can be associated with a pair of slots $(slot_i(x), slot_j(y))$ with respect to the chosen embedding: for each such edge, we create the three vertices $w(x, y), g(x, y), b(x, y)$. However, one cannot directly create the three 1/3-clauses $(w_i(x), w(x, y), w_j(y))$, $(g_i(x), g(x, y), g_j(y))$, and $(b_i(x), b(x, y), b_j(y))$, to

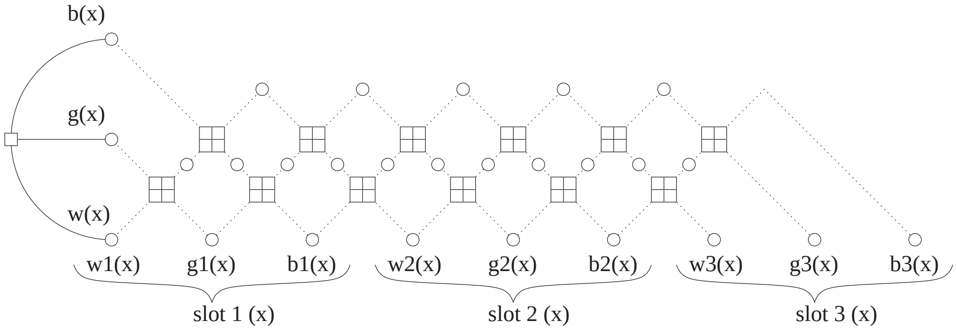


Fig. 26. Duplication of color slots.

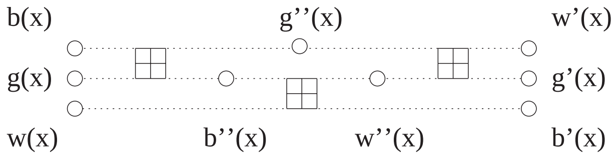


Fig. 27. Reversal of the clockwise order of the color slots.

simulate the edge (x, y) because both slots $slot_i(x)$ and $slot_j(y)$ have the same clockwise order for b, g, w and they face each other. So, one of the two slots, say $slot_i(x)$ must be "twisted" to reorder $b_i(x), g_i(x), w_i(x)$ in counterclockwise order. This is done by using three crossover-boxes as in Fig. 27, where $b_i'(x), g_i'(x), w_i'(x)$ are now in counterclockwise order. We can now create the 1/3-clauses $(w_i'(x), w(x, y), w_j(y))$, $(g_i'(x), g(x, y), g_j(y))$ and $(b_i'(x), b(x, y), b_j(y))$, to simulate any edge (x, y). Finally, choosing an arbitrary edge (u, v) of the input graph and connecting $w(u)$ and $b(u)$ to the distinguished vertex t of a gadget CONST ends the parsimonious reduction from PLAN-3-COL to PLAN-1/3-SAT.

References

[1] R. Barbanchon, Planar Hamiltonian problems and linear parsimonious reductions, Tech. Report, Les Cahiers du GREYC 1, 2001 (postscript available at http://www.info.unicaen.fr/algo/publications).
[2] R. Barbanchon, E. Grandjean, Local problems, planar local problems and linear time, in: Computer Science Logic, Lecture Notes in Computer Science, Vol. 2471, Springer, Berlin, 2002, pp. 397–411.
[3] N. Creignou, The class of problems that are linearly equivalent to satisfiability or a uniform method for proving NP-completeness, Theoret. Comput. Sci. 145 (1–2) (1995) 111–145.
[4] N. Creignou, J.-J. Hébrard, On generating all solutions of generalized satisfiability problems, Inform. Théor. Appl. 31 (6) (1997) 499–511.
[5] N. Creignou, M. Hermann, Complexity of generalized satisfiability counting problems, Inform. Comput. 125 (2) (1996) 1–12.
[6] A.K. Dewdney, Linear time transformations between combinatorial problems, Int. J. Comput. Math. 11 (1982) 91–110.

[7] M.R. Garey, D.S. Johnson, Computers and Intractability, W.H. Freeman and Co., New York, 1979.
[8] E. Grandjean, Sorting, linear time and the satisfiability problem, Ann. Math. Artif. Intell. 16 (1996) 183–236.
[9] H.B. Hunt III, M.V. Marathe, V. Radhakrishnan, R.E. Stearns, The complexity of planar counting problems, SIAM J. Comput. 27 (4) (1998) 1142–1167.
[10] H.B. Hunt III, R.E. Stearns, M.V. Marathe. On the complexity of planar graph and hypergraph coloring (title as cited in the article 'The complexity of planar counting problems'), communicated draft, courtesy of Marathe, 2001.
[11] D.K. Kozen, The Design and Analysis of Algorithms, Springer, Berlin, 1991.
[12] D. Lichtenstein, Planar formulae and their uses, SIAM J. Comput. 11 (2) (1982) 329–343.
[13] N. Linial, Hard enumeration problems in geometry and combinatorics, SIAM J. Algebraic Discr. Meth. 7 (1986) 331–335.
[14] C.H. Papadimitriou, Computational Complexity, Addison-Wesley, Reading, MA, 1994.
[15] L.G. Valiant, The complexity of computing the permanent, Theoret. Comput. Sci. 8 (2) (1979) 189–201.
[16] L.G. Valiant, V.V. Vazirani, NP is as easy as detecting unique solutions, Theoret. Comput. Sci. 47 (1) (1986) 85–93.