Proving Conditional Randomness using the Principle of Deferred Decisions

Alexis C. Kaporis
Lefteris M. Kirousis Yiannis C. Stamatiou

1 INTRODUCTION

In order to prove that a certain property holds asymptotically for a restricted class of objects such as formulas or graphs, one may apply a heuristic on a random element of the class, and then prove by probabilistic analysis that the heuristic succeeds with high probability. This method has been used to establish lower bounds on thresholds for desirable properties such as satisfiability and colorability: lower bounds for the 3-SAT threshold were discussed briefly in the previous chapter. The probabilistic analysis depends on analyzing the mean trajectory of the heuristic—as we have seen in Cocco et al. [3]—and in parallel, showing that in the asymptotic limit the trajectory's properties are strongly concentrated about their mean. However, the mean trajectory analysis requires that certain random characteristics of the heuristic's starting sample are retained throughout the trajectory.

We propose a methodology in this chapter to determine the conditional that should be imposed on a random object, such as a conjunctive normal form (CNF)

Computational Complexity and Statistical Physics,
edited by Allon Percus, Gabriel Istrate, and Cristopher Moore, Oxford University Press.

formula or a graph, so that conditional randomness is retained when we run a given algorithm. The methodology is based on the principle of deferred decisions. The essential idea is to consider information about the object as being stored in "small pieces," in separate registers. The contents of the registers pertaining to the conditional are exposed, while the rest remain unexposed. Having separate registers for different types of information prevents exposing information unnecessarily. We use this methodology to prove various randomness invariance results, one of which answers a question posed by Molloy [8].

2 PRINCIPLE OF DEFERRED DECISIONS

Let $G \in \mathcal{G}_{n,m}$ be a graph chosen uniformly at random, conditional on its number of vertices n and number of edges m. All G with n vertices and m edges are thus equiprobable. Intuitively, if we delete from G a vertex v chosen uniformly at random and also delete all edges incident on v, the new graph should be random conditional on the new number of vertices, $n-1$, and the new number of edges m', where m' is a random variable. In other words, given m', the new graph is equiprobable among all graphs with $n-1$ vertices and m' edges. Note that here and in what follows, "random" will mean "uniformly random," that is, equiprobable, on conditionals that will be either explicit or clear from the context.

Knuth [6, Lecture 3] has introduced a method, known as the *principle of deferred decisions*, by which randomness claims such as the one above can be verified. In the specific example of vertex deletion from a $\mathcal{G}_{n,m}$ graph, it works as follows. Consider $n+m$ cards facing down, or more precisely, $n+m$ registers with unexposed content. The first n of them correspond to the vertices of the graph and the remaining m to its edges. The register of a vertex v contains pointers to the registers of the edges incident on v. The register of an edge e contains pointers to the registers of the two endpoints of e. That the registers are unexposed means that the pointers can be specified randomly. To delete a random vertex, do the following: point randomly to a vertex register; expose its contents; expose all edge registers pointed to by this vertex register; delete the exposed vertex register and the exposed edge registers; nullify pointers in other vertex registers that point to deleted edge registers (without exposing these vertex registers). The registers that have not been deleted remain unexposed and, therefore, they can be filled in randomly. The only conditional, that is, the only exposed information about the graph, is the new number of vertex registers and the new number of edge registers.

The principle of deferred decisions states that conditional randomness is retained as long as no new information about the current contents of unexposed registers can be determined, at any given update step, from information exposed up until that step. The method can be applied in more complicated situations. Consider a random graph conditional on (i) the number of vertices, (ii) the

number of edges, and (iii) for each $i = 0, \ldots, n-1$, the number of vertices of degree i (the degree sequence). We claim that upon deleting a random vertex of degree i (for any i) and its i incident edges, the new graph is random conditional on the same type of information. Indeed, it suffices to augment the argument of the previous paragraph with the additional assumption that for each vertex v there is an *exposed* degree register containing an integer equal to its degree. This degree register needs to be exposed so that the algorithm may choose, at random, a vertex of a given degree. After a deletion step, the contents of the remaining vertices are updated. After the update, no information about the new values of the unexposed registers can be determined from what still is, or previously was, exposed. Therefore, the new graph is random given the number of its vertices, the number of its edges, and its degree sequence.

Notice that keeping a register unexposed is not in itself sufficient to guarantee that its contents stay random. Randomness is destroyed if one could even *implicitly* infer additional information about the current contents of an unexposed register from the combined knowledge of the current and previous contents of exposed registers. Therefore, in all cases, a proof is necessary that no new information about the current values of unexposed registers can be implicitly revealed. On the other hand, it is permissible for a given update step to implicitly reveal information about previous contents—subsequently overwritten—of an unexposed register. This does not destroy randomness, in that it is the *updated* structure that must be proven random. Since revealing past secret information causes no harm as long as no current secret information is revealed, it is convenient to imagine an omniscient "intermediary": an agent independent of the deleting algorithm who updates all necessary registers in total confidence (see, in this respect, the "card model" in Achlioptas [1]). Randomness is retained even if the actions (updates) of the intermediary combined with all exposed information implicitly yield some information about past values of unexposed registers, as long as no information about their current contents is revealed. Of course, this construct of "intermediary" is not a formal notion, but simply a convenient way to describe the updating mechanism.

Notice also that one should not assume that *all* previously unexposed information that is going to be overwritten is necessarily exposed at an update. Doing so might make it possible to infer additional implicit information about the updated contents of unexposed registers. In general, only part of the information to be overwritten needs to be known in order to carry out the update, and thus implicitly revealed. The construct of the omniscient intermediary operating in secrecy frees us from having to make explicit exactly what secret information (to be overwritten) is implicitly revealed at an update. We simply need to make sure that no updated secret information is implicitly revealed after the update.

We illustrate these points by a further example. Consider a random graph conditional on (i) the number of vertices, (ii) the number of edges, (iii) the number of vertices of degree 1, and (iv) the number of vertices of degree 0 (isolated vertices). We claim that upon deleting a random vertex of degree 1

and its incident edge, the new graph is random conditional on the same type of information. This randomness claim is an immediate consequence of a more general theorem proved in Pittel et al. [10] (see also Broder et al. [2]), where the degrees of the vertices to be deleted are allowed to take values up to an arbitrary fixed integer k, assuming that the degree sequence of the graph is given up to k (we have seen in the previous example that this is true if we allow the degrees to range up to $n-1$). The proof in Pittel et al. [10] depends upon counting all possibilities. However, the result can also be proved using the principle of deferred decisions. Assume that for each vertex v there is an *exposed* degree register that contains a three-valued parameter, indicating whether the degree of that vertex is 0, 1, or ≥ 2. In contrast to the case where the whole degree sequence was known, updating these registers after a deletion step presupposes knowledge of *unexposed* information. For instance, to update the degree register of a vertex that had degree at least 2 before the deletion, and which lost an incoming edge because of the deletion, we need to know whether its degree was previously exactly 2 or strictly more. However, it is easy to see that no information about the *updated* value of the unexposed registers is revealed by the combined knowledge of what currently is and previously was exposed: if an updated degree register ends up with the value ≥ 2, beyond this information we still have no knowledge of its actual degree. Therefore, randomness is retained. The omniscient intermediary secretly carries out the updating, using unexposed information. Even though the intermediary might reveal implicit information about the past values of registers, an observer cannot obtain any knowledge about the current contents of any unexposed register from what is and was unexposed.

On the other hand, the fact that additional current information is implicitly revealed is sometimes hard to notice. The subtlety of implicit disclosure can be illustrated by the following example. Let a *B&W graph* be a graph whose *edges* are either black or white. Call a vertex *all-white* if all the edges incident on it are white. Let the w-degree of a vertex v be the number of all-white vertices that v is connected with (see figs. 1 through 3). Notice that a black edge incident on v does not count towards the w-degree of v, while a white edge incident on v may or may not count towards the w-degree of v. Suppose we are given a random B&W graph G conditional on the number of vertices, and for each vertex v, the w-degree of v as well as the number of black edges and the number of white edges incident on v. All other characteristics of G are assumed to be random. Formally, given a fixed integer n and a fixed array of integers $d_{w,i,j}$ where $w, i, j = 0, \ldots, n-1$, then G is chosen with equal probability among all B&W graphs such that $d_{w,i,j}$ is the number of vertices in the graph with w-degree w, i incident white edges and j incident black edges. We assume that the values of the array are such that there is at least one such graph. Suppose now that we delete from G a vertex v, chosen at random among all vertices with a specified w-degree (say 0). Suppose we also delete all edges, black and white, incident on v. Is the new graph random conditional on the same type of information?

Prima facie, one may think that the answer to this question is yes. Indeed, suppose that the exposed registers give for each vertex its w-degree, as well as the number of black edges and the number of white edges incident on it. All other information about the graph is assumed to be unexposed, that iss, random. After the deletion of a vertex v as previously described, and the subsequent deletion of all edges incident on v, all registers are updated. We may be tempted to conclude that the same type of information about the graph is known before and after the deletion, leading to an affirmative answer to the question. Unfortunately, this argument is erroneous. To see why, observe what happens if, after the deletion of v, the exposed w-degree of another vertex u increases. Using the combined knowledge of the current and previous contents of exposed registers, we can infer that in the new graph there exists at least one vertex v' that has just become all-white (as the result of the deletion of a black edge joining v with v'). Additionally, we learn that u is connected with at least one of these newly-all-white vertices. However, this last type of information is not supplied by the currently exposed registers, which give only the w-degree of u and the number of black and white edges incident on it. They do not specify a subset of the all-white vertices connected with u. The fact that we have implicit access to that information means that randomness cannot be retained in the new graph.

We now show a specific case of this. Consider the list of degree parameters (w-degree, number of incident white edges and number of incident black edges) given in figure 1(a) for each vertex of a random B&W graph. Then, by an easy case analysis we may verify that the only graphs having these degree parameters are the two depicted in figure 1(b) and (c). These two graphs are equiprobable, and any information about them other than what is in the upper table is assumed to be stored in unexposed registers. Suppose now that we delete the vertex v_5 from the random graph. Then the resulting graph, depending on which the original one was, will have the degree parameters given either in figure 2(a) or in figure 3(a). Suppose the resulting graph has the degree parameters of figure 2(a), so that the original graph was the one in figure 1(b)—examining this case will be sufficient for the purposes of demonstration. Again, by an easy case analysis we can verify that the only graphs having these degree parameters are the two depicted in figure 2(b) and 2(c). (If the original graph was the one in fig. 1(c), then the only possible graph having the degree parameters of fig. 3(a) is the one depicted in fig. 3(b)—we do not examine that case here.)

If deleting vertex v_5 did not destroy randomness, then both graphs in figure 2(b) and 2(c) should be equiprobable. However, from the *combined* knowledge of the tables in figure 1(a) and figure 2(a), we can easily infer that the graph in figure 2(c) is impossible. This is so because combining the information in the last columns of the tables in figure 1(a) and 2(a) we find that the newly all-white vertex is v_6 (it is the only vertex that previously had, but no longer has, an incident black edge). Also, from the combined information in the third and fourth rows of the second columns of these tables we see that both v_3 and v_4 are adjacent to v_6, as their w-degree has increased. Continuing with an easy case analysis,

(a)

vertices	w-degree	# of white incident edges	# of black incident edges
v_1	0	2	0
v_2	1	1	1
v_3	1	2	2
v_4	0	1	1
v_5	0	0	1
v_6	0	2	1

(b) (c)

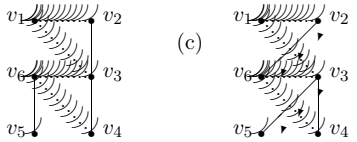

FIGURE 1 Original B&W graph: (a) exposed register values; (b) and (c) the two possible graphs corresponding to these values. Solid lines represent black edges and dashed lines represent white edges.

we conclude that the only graph that has the degree parameters of figure 2(a) and was obtained from a graph that has the degree parameters of figure 1(a) by deleting v_5 is the graph in figure 2(b). In other words, the combined knowledge of the two tables—the one before the deletion and the one after—reveals additional information that cannot be obtained exclusively from the current table, after the deletion. This proves that randomness is not retained. It is instructive to note that if no information were given about the w-degree of the vertices and we dealt only with information about the ordinary degrees (even if they were categorized by the number of incident white edges and the number of incident black edges) then randomness would be retained. That is true because combined knowledge of the two consecutive tables would not then be enough for us to infer additional unexposed information about the resulting graph.

The execution of an algorithmic step on the graph, such as the deletion of a vertex and the edges incident on it, can thus implicitly but subtly expose additional information about the current values of unexposed registers. In section 4, we describe more fully the methodology that is helpful in checking whether any implicit exposure of additional information has taken place as the result of the application of an algorithmic step. As we have seen here, the basic idea is to store information about the random structure in registers, in sufficiently "small pieces." The payoff of doing so is that implicit disclosure of information can be detected easily. Again, we do not require that updates be performed only on the basis of exposed information: unexposed information can be made available to the omniscient "intermediary" doing the updating. But there must be no way for us to infer this information.

(a)

vertices	w-degree	# of white incident edges	# of black incident edges
v_1	0	2	0
v_2	1	1	1
v_3	2	2	2
v_4	1	1	1
–	-	-	-
v_6	0	2	0

(b) (c)

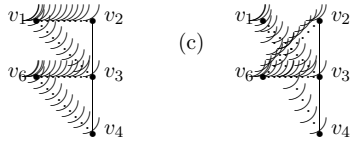

FIGURE 2 B&W graph from figure 1(b) with vertex v_5 deleted: (a) exposed register values; (b) and (c) the two possible graphs corresponding to these values.

(a)

vertices	w-degree	# of white incident edges	# of black incident edges
v_1	0	2	0
v_2	1	1	1
v_3	1	2	1
v_4	0	1	1
–	-	-	-
v_6	0	2	1

(b)

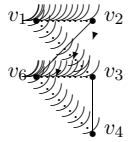

FIGURE 3 B&W graph from figure 1(c) with vertex v_5 deleted: (a) exposed register values; (b) the only possible graph corresponding to these values.

One might say that a safer way to prove conditional randomness claims is by rigorous counting arguments, rather than through the principle of deferred decisions. In complicated situations, however, counting arguments are practically impossible. As we will see from specific applications, our methodology makes it easy to specify what the *a priori* exposed information should be in order to retain randomness throughout the execution of an algorithm, given the type of operations that the algorithm allows. Such considerations have attracted much attention lately, in view of the increased interest in the probabilistic analysis of

heuristics on random Boolean formulas and graphs. This has been discussed in Cocco et al. [3] (see also Molloy [9] for an overview of satisfiability and colorability thresholds). The probabilistic analysis involves analyzing the mean path of the heuristic [11], while showing that randomness is retained throughout the course of the heuristic. It is in situations like this where our methodology is particularly useful. This approach can ultimately be used to obtain lower bounds on threshold locations: indeed, the best lower bound to date on the satisfiability threshold [4, 5], mentioned in the previous chapter, has been proven using the principle of deferred decisions.

The rest of the chapter describes specific applications of this nature. We answer, notably, a question posed by Molloy [8] concerning a Davis-Putnam heuristic acting on a CNF formula comprised of 3- and 2-clauses, when the literals to be satisfied are selected on the basis of how often they appear in each of the two types of clauses. Using the principle of deferred decisions, we show what characteristics must be conditional in order to retain randomness throughout the procedure (theorem 4.2 in section 4), and conjecture that this is the minimal set of conditionals needed.

3 TERMINOLOGY AND NOTATION

Our results can be applied in various contexts related to random graphs or formulas. However, for concreteness, we first present them in the context of random formulas comprised only of 3- and 2-clauses. We introduce below the related terminology and notation.

Let V be a set of variables of cardinality n. Let L be the set of literals of V, that is, elements of V and their negations. A k-clause is a disjunction of exactly k literals from L. Let ϕ be a Boolean formula in conjunctive normal form (CNF), comprised of 3- and 2-clauses. Let m be the total number of clauses of the formula. Let C_3 and C_2 denote the collections of 3-clauses and 2-clauses of ϕ, respectively, and let c_3, c_2, and l be the respective cardinalities of the sets C_3, C_2, and L. Clearly $c_3 + c_2 = m$, and $l = 2n$. (Note that the notation used here is slightly different from that of Cocco et al. [3]: there, C_3 and C_2 were the numbers of 3- and 2-clauses, and c_3 and c_2 were the respective *densities*. Note also that C_3 and C_2 are distinct from $\mathcal{C}_3$ and $\mathcal{C}_2$ from the previous chapter, where they denoted the collections of clauses containing exactly 3 and 2 *positive* literals, respectively.)

For $i = 0, 1, \ldots, 3c_3 + 2c_2$, let D_i be the set of literals in L that have exactly i occurrences in ϕ. The elements of D_i are said to have *degree* i. Literals whose negation is in D_0 are called *pure*. Notice that according to our terminology, a literal in L whose variable does not appear at all in the formula is pure.

Let D_1^3 and D_1^2 be the sets of literals that have exactly one occurrence in ϕ, in a 3-clause and 2-clause, respectively. D_1 is then the disjoint union of D_1^3

and D_1^2. Let also $D_{1\times 1}^2$ be the subset of D_1^2 comprised of literals that appear in a 2-clause whose second literal also belongs to D_1^2.

Let d_i, d_1^3, d_1^2, and $d_{1\times 1}^2$ be the respective cardinalities of D_i, D_1^3, D_1^2, and $D_{1\times 1}^2$. Obviously, $d_1 = d_1^3 + d_1^2$.

Consider the collection of formulas comprised of 3- and 2-clauses that have given, fixed values for the parameters l, c_3, c_2, d_0, d_1^3, d_1^2, and $d_{1\times 1}^2$. Make this collection into a probability space by assigning to each one of its elements the same probability (we assume that the values of the parameters are such that this space is not empty). An element of this space is called a random $\{3,2\}$-CNF formula conditional on the values of l, c_3, c_2, d_0, d_1^3, d_1^2, and $d_{1\times 1}^2$. One could define random graphs similarly, conditional, for instance, on the number of edges and vertices, as we did in the previous section. Such formulas and graphs are called *conditionally random objects*.

We will consider algorithms on random $\{3,2\}$-CNF formulas that only apply steps of the following three types (one step may comprise several constituent substeps):

- *Set a pure literal.* Select at random a pure literal, set it to TRUE and delete all clauses where it appears.
- *Set a degree-one literal from a 3-clause.* Select at random a literal in D_1^3, set it to FALSE, delete it from the 3-clause where it appears and delete all clauses where its negation appears.
- *Set a degree-one literal from a 2-clause.* Select at random a literal in D_1^2, set it to FALSE, delete it from the 2-clause where it appears and delete all clauses where its negation appears. This can create a 1-clause. As long as there are 1-clauses, choose one at random, set its literal to TRUE, delete all clauses where it appears and delete its negation from any clause in which it appears. Ignore (delete) any empty and thus trivially unsatisfiable clause that may occur during this step. This last provision is simply a technicality introduced to study the randomness of the formula independently of its satisfiability. Of course, when such a step is used as a subroutine of an algorithm for satisfiability, the occurrence of an empty clause is an indication to stop immediately and report unsatisfiability.

4 RESULTS

Theorem 4.1. *Let ϕ be a random $\{3,2\}$-CNF formula conditional on the values of the parameters l, c_3, c_2, d_0, d_1^3, d_1^2, and $d_{1\times 1}^2$. If any algorithmic step like the ones described above is applied to ϕ, then the formula obtained is a random $\{3,2\}$-CNF formula conditional on the new values of the parameters l, c_3, c_2, d_0, d_1^3, d_1^2, and $d_{1\times 1}^2$.*

Proof Notice that no algorithmic step differentiates between degree-one literals appearing in 2-clauses on the basis of the degree of the other literal in the 2-clause. Still, according to the statement of the theorem, randomness is preserved if it is conditional not only on d_1^2 but also on $d_{1\times 1}^2$. The reason for this will become clear later in this proof.

We first introduce some general notions, in more formal terms than before. An object such as a formula or a graph can be modeled by a data structure. Let us think of a data structure as a collection of registers containing information about the object. For example, a data structure modeling a graph includes a register for each vertex, with pointers to the registers of the edges incident on the vertex. It also includes a register for each edge, with pointers to the registers of the vertices on which the edge is incident. A data structure modeling a formula includes a register for each literal, with pointers to the registers of the literal appearing in the clause. It also includes a register for each clause (more information about the registers of a formula is given below).

Registers are partitioned into groups. The elements of each group contain various types of information for the same part of the modeled object. For example, for each vertex of a graph, we may have several registers in one group: one with pointers to the edges incident on the vertex, another with the degree of this vertex, etc. For the present purposes, we refer to the registers belonging to the same group as sub-registers of the group. We also imagine, for each group, a head register with pointers to its sub-registers. When a sub-register of a group contains a pointer to another group, it is assumed to point to the head register of that other group. Intuitively, the reason for storing different types of information in separate sub-registers is to avoid exposing all information about a part of the modeled object when it is necessary to expose only a "small piece" of it.

A *data structure with unexposed information* is a data structure whose (sub-)registers are partitioned into two categories, called *unexposed* and *exposed* registers. The partitioning is done according to rules given in the definition of the structure. These rules are based on the type of contents of the registers. The head registers of the groups are always exposed. Intuitively, one may think of such a structure as modeling an object whose characteristics stored in the unexposed registers are random, conditional on the information stored in the exposed registers. The same group may contain both exposed and unexposed sub-registers. For example, although the specific edges where a vertex appears may not be exposed, its degree may be exposed.

In general, given a conditionally random object, we associate with it a data structure as above. An algorithmic step that deletes an element of the object (such as the deletion of a vertex or the assignment of a variable) corresponds to the deletion of the group of registers associated with the deleted element of the object. After the deletion, all registers are updated.

Definition 1. *An algorithmic step is called randomness preserving if, after the corresponding deletions and updates of registers, no information about the contents of unexposed registers can be inferred from what currently is and previously was exposed, beyond what can be inferred from what is currently exposed. In other words, no additional information is implicitly revealed by knowing both past and current exposed information.*

To prove a randomness claim such as the theorem under consideration, it suffices to find a data structure with unexposed information that models the conditionally random object in the claim, and then to show (i) that the algorithmic steps are randomness preserving and (ii) that the information in the conditional is exactly the information that can be extracted from the exposed registers of the structure.

We describe below a structure $\mathcal{S}$, with unexposed information, that models a random $\{3,2\}$-CNF formula conditional on the parameters l, c_3, c_2, d_0, d_1^3, d_1^2, and $d_{1\times 1}^2$.

- For each literal t in L, the structure $\mathcal{S}$ contains a group of sub-registers collectively called literal sub-registers. These contain information about the degree of the literal, its occurrences in the formula and its negation. The information that is assumed exposed is (i) the degree and (ii) the position in the formula of literals with a single occurrence that happens to be in a 2-clause. All other information is unexposed. More formally, one of these sub-registers contains two bits of information indicating whether t belongs to D_0 (t does not appear in the formula), D_1^3 (t has degree 1 and appears in a 3-clause), D_1^2 (t has degree 1 and appears in a 2-clause) or none of these (t has degree at least 2). This sub-register is exposed. Also, we assume that there are sub-registers containing pointers to the positions of all occurrences of t in the formula (to the heads of all clause sub-registers where t appears; see below). These sub-registers are exposed if t is in D_1^2 and unexposed otherwise. The reason for exposing the position in the formula of literals in D_1^2 will become apparent later. Finally, we assume that there is an unexposed sub-register pointing to the head of the literal sub-register of the negation of t. It is important to notice that because the pointer to the negation of a literal is unexposed, each literal is paired with its logical negation randomly.
- For each clause in the formula, the structure $\mathcal{S}$ contains a group of sub-registers collectively called clause sub-registers. These contain information about the type of the clause (3-clause or 2-clause) and pointers to the heads of literal sub-registers corresponding to the literals that appear in the clause. The information about the type of the clause is exposed, while the pointers to the literal registers are unexposed.

It is straightforward to verify that after the application of any of the algorithmic steps, no information about an unexposed register can be deduced from what is and previously was exposed. Under these circumstances, the

randomness of the structure $\mathcal{S}$ is preserved under an algorithmic step. The need for having the positions of literals in D_1^2 exposed can be seen in the event that under an update step, exactly one 3-clause shrinks to a 2-clause and exactly one literal moves from D_1^3 to D_1^2. In that case, the information about the type of each clause and the degree of each literal is sufficient to allow us to infer the position of this literal.

Now the theorem follows because the information that can be extracted from $\mathcal{S}$ consists only of the values of the following parameters: l (the number of groups of literal sub-registers), c_2 (the number of groups of clause sub-registers for 2-clauses), c_3 (the number of groups of clause sub-registers for 3-clauses), d_0, d_1^3, d_1^2, and $d_{1\times 1}^2$. The value of $d_{1\times 1}^2$ can be obtained from $\mathcal{S}$ because the positions in the formula of literals in D_1^2 are exposed. All other information that can be extracted from $\mathcal{S}$ can be expressed in terms of the values of the parameters l, c_3, c_2, d_0, d_1^3, d_1^2, and $d_{1\times 1}^2$, only. (One can immediately see, for instance, that the number of 2-clauses where both positions are filled with literals of degree at least 2 or the number of 2-clauses where one position contains a literal of degree at least two and the other a literal of degree exactly one can be expressed in terms of the values of the parameters l, c_3, c_2, d_0, d_1^3, d_1^2, and $d_{1\times 1}^2$). This completes the proof of Theorem 4.1.

We now come to the generalization of the previous result to arbitrary degrees, where algorithms making use of the overall number of occurrences of literals in 3-clauses and 2-clauses, separately, are allowed. To preserve randomness in this case, a conditional given by a number of integer parameters—as in the previous theorem—is not enough. We have to assume that the positions of all literals appearing in 2-clauses are known, regardless of their degree: this information is revealed when a 3-clause shrinks to a 2-clause and the exposed degree information of literals is updated. However, no information about negations of literals or identification of literals need be revealed, nor does information on the positions of literals appearing in 3-clauses. In other words, we have to assume that the *pattern* in which literals are paired in 2-clauses is conditional, though the pattern need not reveal the pairing of literals of opposite logical sign. This is still a severe restriction on the randomness of the formula. Below, we formalize the notion of pattern.

Fix an even integer $2n$ representing the number of literals of a formula, and an integer c_3 representing the number of 3-clauses in a formula. A *pattern for 2-clauses and degree sets that is transparent with respect to negations* (pattern, in short) is a set of unordered pairs $\underline{C}_2$ of integers from $\{1,\ldots,2n\}$, representing the collection of 2-clauses of the formula, together with a collection of sets $D_i^3 \subseteq \{1,\ldots,2n\}, i=0,\ldots,c_3$, such that $\sum_i i|D_i^3| = 3c_3$, representing the collection of sets of literals whose number of occurrences in 3-clauses is i.

Now fix a pattern P as described above. A random formula ϕ conditional on P is constructed as follows: randomly choose c_3 unordered triplets from $1,\ldots,2n$

so that all integers in each D_i^3 appear in exactly i such triplets; denote this set by $\underline{C}_3$; randomly select a one-to-one and onto mapping neg : $\{1,\ldots,n\} \to \{n+1,\ldots,2n\}$ representing the negations; in the tuples of $\underline{C}_3$ and of $\underline{C}_2$, replace each $k = 1,\ldots,n$ with variable x_k and each $\mathrm{neg}(k), k = 1,\ldots,n$, with its negation $\overline{x_k}$, and denote by C_3 and C_2, respectively, the sets of clauses thus obtained; let the formula ϕ be the one that has as 3-clauses and as 2-clauses the sets C_3 and C_2, respectively. Notice that since the negation function "neg" was random, a literal and its negation may appear in the same clause. If we wish to avoid this, "neg" may instead be a random one-to-one and onto mapping made conditional on the fact that for no $i = 1,\ldots,n$ can both i and $\mathrm{neg}(i)$ appear in the same tuple of either $\underline{C}_3$ or $\underline{C}_2$. Based on the method of proof of the previous theorem, one can obtain the following result that answers an open question posed by Molloy [8].

Theorem 4.2. *Let ϕ be a random $\{3,2\}$-CNF formula conditional on a given pattern P, as described above. For arbitrary i and j, choose at random a literal t with i occurrences in 3-clauses and j occurrences in 2-clauses. Assign to t the value* TRUE *and perform the necessary deletions and shrinking of clauses accompanied by repeated setting to* TRUE *of literals in 1-clauses, as long as 1-clauses exist. The new formula is then random, conditional on its new pattern P'.*

Proof Again, we introduce a structure $\mathcal{S}$ that contains groups of sub-registers corresponding to literals and to clauses. This time, the exposed degree sub-registers of a literal t contain two integers: one giving the number of occurrences of t in 3-clauses and the other giving the number of occurrences of t in 2-clauses. Furthermore, the group of literal sub-registers of t contains information on which 3-clauses and which 2-clauses include t. The information regarding 3-clauses is unexposed. The information regarding 2-clauses, however, must be exposed because after an algorithmic step, it can be inferred from the knowledge of the previous and current values of the registers giving the type of each clause (3-clause or 2-clause) and the degrees of the literals. One may readily confirm that nothing can then be inferred about the unexposed registers after the application of an algorithmic step. It is also immediately apparent that the information that can be extracted from such a structure $\mathcal{S}$ is given by the pattern P.

Note that if the algorithm does not make use of the number of occurrences of literals separately in 3-clauses and 2-clauses, but only needs the total number of occurrences of a literal in the formula, then the conditional does not have to include the pairing of literals in 2-clauses. It is sufficient in this case for the conditional to contain the total degree sequence, the number of 3-clauses, and the number of 2-clauses.

Finally, as a further application, let us see what information must be placed in the conditional for an algorithm deleting vertices of a specified w-degree from a random B&W graph, as discussed in section 2.

Given a B&W-graph G, let $W \subseteq G$ be the subgraph comprised of the vertices of G, with vertices marked according to whether or not they are all-white (in the sense of G), and all white edges with at least one endpoint incident on an all-white vertex. Call W the subgraph of w-degree *witnesses*. Without giving details, one can again define a notion of random B&W graphs conditional on the number of vertices, the total number of edges and the precise subgraph of w-degree witnesses. Note that to construct the rest of the graph from this information, one can arbitrarily place edges between vertices that are not all-white and then arbitrarily color them black or white, taking care that at least one black edge is incident on each vertex that is not all-white.

Then the following theorem holds. We omit its easy proof, as the notion of B&W graphs was introduced only for illustrative purposes.

Theorem 4.3. *If we delete a random vertex of a specified arbitrary w-degree from a B&W graph that is random conditional on the number of vertices, the total number of edges and the subgraph of w-degree witnesses, then the new graph is random conditional on the new number of vertices, the new total number of edges and the new subgraph of w-degree witnesses.*

An analogous result holds if the deleted vertex has specified numbers of white and black edges incident on it (the conditional in the latter case must be augmented to contain the sequence d_{ij} giving the number of vertices with i white and j black edges incident on them).

We conclude this chapter by the following

Informal Conjecture. The conditionals of theorems 4.1, 4.2, and 4.3 contain the least information possible. With weaker conditionals, randomness would not be retained.

ACKNOWLEDGMENTS

This research has been supported by the University of Patras, Research Committee (Project C. Carathéodory no. 2445). We would like to thank an anonymous referee and the editors for their comments on the previous draft that led to substantial improvements. The second author thanks D. Achlioptas and M. Molloy for several discussions about probabilistic arguments, and acknowledges partial support by the EU within the 6th Framework Programme under contract 001907 (DELIS).

REFERENCES

[1] Achlioptas, D. "Lower Bounds for Random 3-SAT via Differential Equations." *Theor. Comp. Sci.* 265(1-2) (2001): 159–185.

[2] Broder, A., A. Frieze, and E. Upfal. "On the Satisfiability and Maximum Satisfiability of Random 3-CNF Formulas." In *Proceedings of the 41st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '93)*, 322–330. New York: ACM Press, 1993.

[3] Cocco, S., R. Monasson, A. Montanari, and G. Semerjian. "Analyzing Search Algorithms with Physical Methods." This volume.

[4] Kaporis, A., L. Kirousis, and E. Lalas. "The Probabilistic Analysis of a Greedy Satisfiability Algorithm." In *Proceedings of the 10th Annual European Symposium on Algorithms (ESA '02)* 574–585. London: Springer-Verlag, 2002

[5] Kaporis, A., L. Kirousis, and E. Lalas. "Selecting Complementary Pairs of Literals." In *Proceedings of the Workshop on Typical Case Complexity and Phase Transitions*, Ottawa, Canada (affiliated with the 18th IEEE Symposium on Logic in Computer Science (LICS '03). To appear in *Electronic Notes in Discrete Mathematics*. http://students/ceid/upatras.gr/∼kaporis/ (accessed on August 9, 2005),

[6] Knuth, D. E. *Stable Marriage and Its Relation to other Combinatorial Problems: An Introduction to the Mathematical Analysis of Algorithms*. (English edition: CRM Proceedings & Lecture Notes 10, American Mathematical Society, 1997; first French edition: Les Presses de l'Université de Montréal, 1976.)

[7] Knuth, D., R. Motwani, and B. Pittel. "Stable Husbands." *Random Structures and Algorithms* 1 (1990): 1–14.

[8] Molloy, M. private communication. [**need date**]

[9] Molloy, M. "Thresholds for Colourability and Satisfiability in Random Graphs and Boolean Formulae." In *Surveys in Combinatorics, 2001*, ed. J. Hirschfeld, 166–200. LMS Lecture Note Series 288. Cambridge University Press, 2001.

[10] Pittel, B., J. Spencer, and N. Wormald. "Sudden Emergence of a Giant k-Core in a Random Graph." *J. Comb. Theor. B* 67 (1996): 111–151.

[11] Wormald, N. "Differential Equations for Random Processes and Random Graphs." *The Ann. App. Prob.* 5(4) (1995): 1217–1235.