Postscript to Fully Automated Fragments of Graph Theory.

Siemion Fajtlowicz, University of Houston.

A wise man believes what he sees, not what he knows. (Huang Po.)

ABSTRACT. We shall explain what was meant by attributing conjectures of *Graffiti* to this program, and we shall discuss the relevance of this issue to the dispute about what constitutes a conjecture-making program initiated at the first mathematical meeting devoted to this subject. The same issue is relevant to the so-called AI debate, the Turing test, and the reporting of the performance of some so-called "discovery programs." A considerable part of this discussion is a response to Hansen's GERAD web publication in which he questions the prospects of automated conjectures and, implicitly, what was accomplished in this direction by *Graffiti*.

We will show that the practice of attributing human conjectures to machines does not serve any useful scientific purposes in the framework of discussion of automation. A mathematical conjecture is more than a formula. Usually, it is also an expression of personal opinion concerning the significance, nontriviality, and correctness of this formula. When the act of "making a conjecture" is attributed to a machine, the author of the program should be expected to clearly explain how the program, as opposed to its users, reached these conclusions. In the second part of this paper, we explain how this was done in the case of *Graffiti*. The practice of attributing human conjectures to machines is actually harmful, because it implicitly denies the possibility of genuine conjecture-making programs, and it falls into the same category of misnomers as claims made in the past by some AI researchers, who used to refer to various programs as "discovery systems," when these programs, in fact, had never made any discoveries.

A closely related issue of reporting the performance of so-called intelligent programs is discussed by Dreyfus, and in a context closer to the subject of this paper, by Ritchie and Hanna. Some researchers are notorious for giving undeserved credit to machines. From the very beginnings of the subject, Hubert Dreyfus was an avowed critic of some claims of AI researchers, and watching the debate between him and Herbert Simon, one might have had the impression that the arguments often were not about what was accomplished in AI, but about what tags could be attached to these accomplishments, even in those cases where there were some accomplishments that could not be questioned. One of the programs which had

1991 *Mathematics Subject Classification.* Primary 68T35, 68R10, F82D99; Secondary 03A05.

a considerable impact on the field, at the very least because of the author's afterthoughts, was Weizenbaum's *Eliza*, simulating a Rogerian psychotherapist. This simple program, containing a clear idea germane to the issue of the Turing test, could have had an even greater and more positive impact on the progress of AI, if Weizenbaum had cautioned the reader that impressive conversations with the program, like the one quoted in the introduction to [54], must have been the exception rather than the rule. That, at least, was the impression I had, running a version of *Eliza* at about the time that the book was published.

The discussion of related mathematical and philosophical issues was initiated by Roger Penrose, who argued that digital computers cannot have human mathematical intuitions. In response, in [16], it was proposed that Penrose's claim could be studied, (empirically, as it was suggested by Putnam,) by administering to a program a Turing-like test, in which the machine would simply be asked to present her mathematical conjectures. This is a completely different point of view from Penrose's, according to whom a machine could not form sound mathematical beliefs unless they could be proved, [38]. Needless to say, the test would involve invention of conjectures by a machine, as opposed to making, testing, or confirming human conjectures with computers. The original Turing test was criticized as purely behavioristic, and this is certainly, in my opinion, a drawback. Of greater interest is the nature of human innate language abilities and, closer to the subject of this paper, different, but probably not unrelated, questions concerning the nature of mathematical intuitions. One can study these questions by attempting to write automated conjecture-making programs. In the author's experience, the main difficulty in this activity is understanding what makes some conjectures more significant than others.

The purpose of automation, as it is pursued by the author, is hardly to prove that computers can make conjectures or discoveries without any help from humans, though this is, of course, an appealing goal. Development of conjecture-making systems can be thought of as an attempt to understand mathematical intuitions and the degree to which they can be mechanically implemented, creating a scenario in which purely mathematical, though not necessarily traditionally mathematical, problems and ideas may have an impact going well beyond this subject.

In a sense, it is a pursuit of identification of some intrinsic mathematical values, not unlike the one attempted by Hardy in his "Mathematician's Apology." The difference between the two approaches is that the "algorithmic" attempt is directed at finding specific expressions of these values. From this point of view, again it is of considerable mathematical, scientific, and philosophical interest whether computers can invent mathematical conjectures by themselves. It is not of any conceivable interest whether computers can be of assistance in making conjectures, because to start with, this is a forgone conclusion. As a matter of fact, it was a forgone conclusion from the very first days of computers.

There is, of course, nothing wrong with the usage of computers to study or to test users' conjectures, or with running these programs to make conjectures based on the performance of these programs. However, most mathematical research programs and all data mining programs easily fall into this category, and discussing these in the context of conjecture-making programs clouds the issue, making it useless for mathematical purposes.

It is difficult to overstate this case, particularly in view of the controversial claims of some AI researchers. What would be the point of Turing's discussion whether machines can think, or of Penrose's debate whether machines can have human mathematical intuition, if these questions could be approached as an exercise in semantics, or to be more precise, in verbalism. It could hardly escape their attention, that by feeding a few numbers into a computer and exercising a highly questionable choice of words, one could say that the machine had discovered that

$$E = mc^2.$$

Nevertheless, much more startling claims were made by AI workers. Indeed, an authentic rediscovery of Kepler's laws by a machine might be considerably harder to accomplish than Einstein's $E = mc^2$. It is difficult not to wonder how much the questionable use of the word "discovery" in reference to BACON slowed down the progress of AI.

Apart from this, from what I know about BACON, whose first versions were written by Langley, it became another influential program, not least because of its association with Professor Simon, who was one of the main spokesmen for this idea and a proponent of the so-called strong-AI point of view which stirred a lot of controversy well before Penrose's claims. In a nutshell, the strong-AI viewpoint amounts to attributing mental states to digital machines. Right or wrong (as a point of view), it is regrettable that hurried claims of what was accomplished provided ammunition against some of the realistic goals of AI and practically led to the self-destruction of some of the initial goals of this field. The main tenet of "Scientific Discovery" - the *"how to make discoveries"* book - is that the process of discovery is simply tantamount to problem solving, and that this thesis was proved with BACON, [34].

The distinction between automated conjectures and what Hansen refers to as computer-assisted conjectures is also as obvious as that between discovery and rediscovery. Even computer simulation of the process of rediscovery is of doubtful value without the necessary precautions, because in the very nature of discovery is that it is done for the first time or, of course, that it is done without any previous knowledge of the original discovery. Yet, in defense of his terminology and discussion of *"computer assisted or automated conjecture-making,"* Hansen simply states in [27] that *"automation being hard, it may limit the scope of problems addressed,"* and in support of this position, he quotes an opinion of Langley which indeed can be construed as an admission that, without human assistance, authentic machine discoveries would be difficult, [27, p. 2]. Perhaps so, but this hardly justifies obfuscating the meaning of well established, unambiguous terms like "discovery", "conjecture", or "authorship".

Specifically, Langley wrote: *"Although the term computational discovery suggests an automated process, close inspection of the literature reveals that the human developer or user plays an important role in any successful project. Early scientific discovery downplayed this fact and emphasized the automation aspect in general keeping with goals of artificial intelligence at the time. However, the new climate in*

AI systems that advise humans rather than replace them ... suggests an important role of the developer.", [35].

In my opinion, the "old climate" was admirable, because it fostered many ambitious goals some of which proved to be realistic. Here is an example related to this paper. On page 99 of "Machines Who Think", Pamela McCorduck describes a suggestion of John McCarthy from the mid-fifties: *"to attempt to construct an artificial language which a computer can be programmed to use on problems requiring conjecture and self-reference. "*

I am sure that Langley is more than right in his assessment that the user's role has been downplayed in the discussion of many so-called discovery programs, but in the case of Graffiti, (regardless of whether this program was included in his discussion,) not only was this role not downplayed, but it was very deliberately emphasized in order to not mislead anyone. The same is true about INGRID, as will be shown later in this paper.

The first attempts to use *Graffiti* to make conjectures about fullerenes showed that the program can make conjectures that have an impact going beyond pure mathematics. The sorting pattern of conjectures of the Minuteman version of *Graffiti* suggested that stable fullerenes tend to be good expanders (relatively to other fullerenes with the same number of atoms) [1]. Patrick Fowler - a fullerene expert - disagreed with my interpretation of this conjecture which suggests a causal relationship between the size of the separator of fullerenes and their stability, [14], [16], [17]. Concerning the similar difference of opinions about a possible relation of stability and the independence number of fullerenes, Fowler, after an initial equally skeptical attitude, was later first to present evidence supporting this hypothesis for which now there is very strong statistical evidence, [13]. Questioning what computers can do, and engaging in consideration what computers cannot do, Dreyfus, Chomsky, and Penrose wrote several books and presented a number of thoughtful arguments. By comparison, Hansen does not present a single argument concerning the "limits" of automated conjectures, apart from perhaps suggesting that *Graffiti* "makes conjectures" in the same sense as INGRID, GRAPH, and AGX, and this issue will be discussed later in this paper.

One of the conjectures discussed above suggests that stable fullerenes tend to have large separators, [14], [16], and [17]. The possible significance of this finding, which was first published in [24], was distorted in this paper without my knowledge by incomplete and (in one case alarmingly) biased statistics. The paper informs the readers that one of two stable 84-atom fullerenes is "only fifth", when ranked by its separator, on the list of 24-atom fullerenes with isolated pentagonal faces, but it does not reveal that this molecule ranks twentieth among about 50 000 fullerenes with 84 atoms. Fowler has never explained why this information was ignored.

At the very outset, the program made conjectures which led to a new characterization and representation of buckminsterfullerene, and the proof suggests that this most stable of the new forms of carbon may be related to the Golay code. A derivative of this molecule, the fullerol $C_{60}(OH)_{24}$, which is studied because of its

[1]As pointed out by Alon in [14] and [23], planar graphs cannot be good expanders in the sense in which this term is usually used.

potential as a cure for some forms of cancer, may owe its properties to pentagonal independent sets - the same dominant structure in the new representation of C_{60} that may relate it to the Golay code, [21], [48]. These are outcomes of one of the very first fullerene conjectures of *Graffiti*, [15], conjecture 840 [2].

One of the conjectures of the Minuteman version of *Graffiti* suggests that stable fullerenes tend to minimize their maximum independent sets. Larson and myself reviewed on this occasion other predictors of fullerene stability, and it appears that all of them are far from perfect, which increases the potential significance of the separator as an indicator of stability. In any case, large separators, according to Fiedler, are an algebraic measure of the connectivity of graphs, and intuitively, the same is true about small independence numbers: the smaller these number are, the larger part of the molecule has to be removed so that the rest splits into singletons. It now appears that the independence number of fullerenes is one of the best, perhaps the single best, currently available statistical indicator of stability, [13].

The joint paper with Larson presents compelling statistical evidence that the independence number has a strong bearing on the stability of fullerenes. To obtain some understanding of this phenomenon, the author decided to develop a version of Graffiti for making conjectures about benzenoids. Even before this version made its first conjectures, [23], discussing it with Doug Klein, the author made a simple observation providing a different and new insight into the relation between stability and the independence number: it appears that, like fullerenes, the stable benzenoids (which are their close relatives) also tend to minimize their independence numbers α. However, unlike for fullerenes, this finding is in agreement with accepted theories of stability of benzenoids. It is well known that stable benzenoids tend to maximize their matching numbers μ, and thus (because benzenoids are bipartite) it follows that for n-atom benzenoids

$$\alpha + \mu = n,$$

i.e. that similarly to fullerenes, stable benzenoids also tend to minimize their maximum independent sets.

Thus logically, to minimize the independence number is the same as to maximize the matching number for these molecules, but from the physical point of view, this presents a new perspective of the stability of benzenoids. This new point of view adds credibility to the stability-independence hypothesis for fullerenes. Moreover, since fullerenes always have perfect matchings, the matching number has no bearing on their stability, and thus it is possible, perhaps even likely, that the stability of benzenoids may be also explained in terms of their independence numbers.

[2]The admittedly very speculative possibility of a relation between the Golay code and the immune system, is of course my own, and has no direct relation to Graffiti's conjectures, but this observation is difficult to resist, particularly because of Coxeter's work following a discovery made in the sixties that certain viruses have structures resembling Fuller's geodesic domes. Some of these viruses show icosahedral symmetry. The origin of viruses is still a mystery, but it is well established that a fullerene-like protein, clathrin, which is linked to the origin of eukaryotic cells (i.e., cells with well developed nuclei), plays the dominant role in communication between cells., [49]. The processes of formation of clathrin and of fullerenes are also not known, though recently, a first report of the production of the icosahedral C_{60} by a rational synthesis was published in "Science", [46], increasing the chance of making practical use of *Graffiti's* findings about fullerenes.

Concerning the "limit of scope" of automated conjectures, it should be pointed out that prior to $Graffiti's$ conjectures, there were no studies of the independence number of carbon molecules.

There may also be some mathematical advantages of this approach. Computationally, the independence number is more difficult than the matching number, but conceptually it may be considered simpler, because the matching number of a graph is the independence number of its line graph. Studying the problem from the point of view of the independence number, Ryan Pepper has already identified several molecules as candidates for the least stable benzenoids with respect to a given number of atoms. The most unstable molecules are, at the very least, needed to provide the background for $Graffiti's$ heuristic *Echo*, but the problem of characterizing benzenoids with the minimum matching numbers may also be of purely mathematical interest, [37].

Benzenoids that maximize their matching numbers have perfect matchings, or in chemical terms, Kekule structures. According to another chemical theory, among benzenoids with perfect matchings, those that are the most stable tend to maximize the number of their Kekule structures. This seems to indicate that stable benzenoids tend to be good (relatively to other benzenoids with same number of atoms) expanders, yielding another argument in favor of the stability-expanding hypothesis for fullerenes and possibly suggesting a novel point of view of the stability of benzenoids.

None of these findings and activities were exclusive doings of $Graffiti$, but the program played in all of them the dominating role, providing the ideas in the form of clear-cut conjectures. The "stability patterns" were discovered by running the interactive part of the system for viewing conjectures, but if anything, the program was assisted by its users, not the other way around, though the exact demarcation line would be difficult to draw, as it is usually the case with collaborative processes. Contrary to beliefs of Professor Simon, who insisted (as did Lenat) that successful discovery systems must be based on domain-specific heuristics, there hardly seems to be any doubt now that experiences with $Graffiti$ show that a few simple domain-independent heuristics are enough to get interesting conjectures [3]. So far, it seems that Hansen's notion of "limits of automated conjectures" is also just a dogma.

Ryan Pepper describes in [40] his classroom experiences in learning graph theory Texas style, i.e., by the method developed by Professor R. L. Moore, with the marked difference that the class participants were working exclusively on conjectures of the latest version of $Graffiti$, namely $Little\ Red\ Riding\ Hood$, basically without getting any hints whether conjectures of the program were true. If Pepper's experiences are any indication of possible uses of Graffiti-like programs in learning mathematics, they may eventually prove of some use in education at levels as low as high school. Let α and ω denote, respectively, the independence and the clique number of an n-vertex graph. One of the first conjectures proposed to the class was

$$\alpha + \omega \leq n + 1,$$

[3]Lenat's program Artificial Mathematician is based on about 250 heuristics.

which, as it seems, had not been discussed before it was conjectured by *Graffiti*, in spite of being suitable as an exercise for elementary school students. This and similar conjectures led students to understanding the idea and the proof of the Ramsey theorem before they even had the chance to see the formal statement of this result. Since then, DeLaVina has used her own educational version of *Graffiti* at the undergraduate level, and this and related experiences of her students are discussed in [8], and on DeLaVina's website. Later, her version of the program *Graffiti.Pc* was also used by Gunnar Brinkmann to teach graph theory at the University of Bielefeld.

Concerning the educational versions of *Graffiti*, some of them are completely automated (apart from the invention of concepts) in the mathematical sense, because they present a deterministic algorithm for making conjectures, [16]. Some early versions of these programs were fully automated in practice, and they did not even require any user-supplied examples, [17]. However, proving or refuting conjectures, though related activities as far as mathematics is concerned, are optional in automated conjecture-making processes, [17]. There is no doubt that conjectures of most (though certainly not all) versions of *Graffiti* were influenced by human-invented examples, but the same is true about individuals, who often base their conjectures on examples invented by others. Proving conjectures or finding counterexamples are different (though related) activities, which are as optional in automated conjecture-making as conjectures or refutations are optional in automated theorem proving.

Selection of concepts is, however, a different story. According to [27], I expressed an opinion that it hardly matters what concepts the program makes conjectures about. I never wrote, said, nor even suggested anything of the sort. The conventional wisdom is, of course, that this does matter very much, and actually the above example involving $\alpha + \omega$, not to go very far, well illustrates this point. Indeed, soon after Pepper's *DIMACS* presentation, the case of equality of

$$\alpha + \omega \leq n + 1$$

was discussed in [29], where it plays a pivotal role, (see also [27], p. 30.)

I did write in [16] that I do not know of a single example of a simple mathematical concept that was shown not to be interesting, and this comment was prompted by my reservations about claims involving concept-inventing abilities of computers. These claims originated with Lenat's AM, which supposedly reinvented the concept of primes and then rediscovered the Goldbach conjecture. Lenat's claims were first soberly questioned by Ritchie and Hanna, and later by others, including Wang and Larson.

The second goal of this paper is to explain the motivation, the goals, and the methods of reporting the conjectures of *Graffiti*. It may seem strange to do this after several papers about this program, but Hansen's GERAD publication [27] makes it all but necessary. The original version of Hansen's paper contained close to two dozen false statements, counting only those that were factually false. In spite of that, the paper was displayed for a year on the website of GERAD, including several months after Hansen's own admission that his paper had to be revised. Many, but certainly not all, false claims were indeed removed from Hansen's revised paper. Perhaps the most damaging was Hansen's (eventually removed) claim that readers of my papers were misled by terminology attributing conjectures discussed in these papers to *Graffiti*. Some of the serious false claims were contradictory with Hansen's own description of *Graffiti*, in spite of his assertion that all cited writings about *Graffiti* were reread many times.

Hansen uses interchangeably the terms "conjectures of a program" and "user's conjectures based on the performance of a program" [4], implying that INGRID and GRAPH did not receive due credit in writings about *Graffiti*. INGRID is not a conjecture-making program, and in more than fifteen years since this program was written, nobody made the slightest suggestion that it was [5]. It is self-evident that attributing to a program conjecture-making abilities automatically excludes from this category programs written for the purpose of assisting humans in making conjectures, or for the purpose of verification or confirmation of existing conjectures.

One of Hansen's comments about the Dalmatian version of *Graffiti* was that: *"However it seems that proofs of easy true conjectures are still done by hand, perhaps some selection still takes place"*, implying that the current version of Graffiti still makes a considerable number of trivially true conjectures. The benzenoid version of Grafffiti, whose conjectures are now listed on the author's web-site, contains all those (over eighty) conjectures of the program made in two 10 hour runs which display carcinogenicity patterns, [23]. I could not prove nor refute out of hand a single conjecture from this list, which included 21 conjectures from the first run that were classified by Graffiti as presumably easier to prove or refute. The same was true about almost all of the remaining conjectures from both runs, (backed up for demonstration if needed.) Out of hand, I was able to refute only one conjecture from the second run, and this conjecture also was classified by Graffiti as trivial. The above example may be not typical of the performance of the Dalmatian version, but trivial conjectures do not present a problem any more in the development of the program. Writing about *Graffiti* (as recently as in [26],) Hansen often refers to and points out difficulties with first versions of the program which may have been a concern 15 years ago. For all practical purposes, the problem of trivial conjectures was solved with the Dalmatian version.

The *Dalmatian* heuristic drives the program to make the strongest conjectures to which it does not know a counterexample. The underlying idea is an algorithmic implementation of Popper's Falsifiability criterion. The Falsifiability criterion was invented as an argument against pseudo-scientific practices, and it is a tribute to bold predictions like, for example, some of the startling implications of the relativity theory (Popper's favorite.)

[4] In one instance, there is a reference to a conjecture based on data produced by a run of AutoGraphix (AGX) "combined with human inspection," and later in the same paper, the authors refer to the same proposition as "the AGX conjecture", [25, p. 107 and 110].

[5] The fact that GRAPH is not a conjecture-making program was clearly stated in a recent lecture about this program by one of the authors of [28] in a special session devoted to computers in graph theory organized by Hansen. To the best my knowledge, whenever the expression "for making conjectures" was used in the reference to GRAPH, it was invariably clear from the context that the program was used as a tool to help humans make their conjectures. Needless to say, GRAPH is one of the first and the most accomplished graph-theoretical programs.

Some of the first responses to conjectures of *Graffiti* were papers by Alon, Chung, Erdös, Seymour, Spencer, and Pach. Erdös, Spencer and Pach refuted a conjecture of *Graffiti*, leading to the first publication resulting from a conjecture of a computer program. Still, they proved that a somewhat modified version of the conjecture was correct. Several follow-up questions concerning the relation between the average distance and the sum of reciprocals of the degree sequence are still open, [15].

The conjecture leading to the paper by Alon and Seymour turned out to be somewhat weaker than an earlier and totally overlooked conjecture of Van Neufallen. *Graffiti*'s conjecture was that the chromatic number of any graph is not more than 1 plus its rank. Before it it was refuted, Lovász proposed an application of this conjecture in communication complexity, which considerably increased interest in the problem, leading to results by Razborov, Nissan and Widgerson, Kotlov and Lovász, and eventually by Raz and Spieker.

Last but not least, Chung proved that the average distance in every connected graph is not more than its independence number. Since then, this result, which is often considered one of the most elegant conjectures of *Graffiti*, has inspired several more papers. From the perspective of this paper, more significant than the results themselves, is that for the first time mathematicians worked on conjectures invented by a computer program.

It appears now that Hao Wang, a well-known logician, made the first attempt ever to write what could be considered a conjecture-making program. This contribution to the history of the subject is a discovery of Larson, [33]. As stated in almost all my writings about *Graffiti*, the main, perhaps the only, non-technical difficulty in developing conjecture-making programs is the issue of the significance of mathematical statements - a problem rarely discussed in the literature. Wang, indeed, had addressed this problem, by asking whether a machine can select an interesting conjecture to be proven by mathematicians [6].

The motivation for writing *Graffiti* was to get a first-hand understanding of the potential of AI in mathematics, independent from automated theorem-proving. Bledsoe, one of the pioneers of this branch of computer science, was probably the first researcher to suggest that conjecture-making abilities may prove to be the key to successful theorem-proving programs. This possibility has not materialized

[6] According to a not well documented, but certainly not apocryphal story quoted in [18], Gödel considered the issue of significance of mathematical statements a problem of foundations of mathematics. The exact nature of his conversation with Mostowski (told to me by Ehrenfeucht and Mycielski), where the question came up, may be never known, but it seems reasonable to expect that after showing that not all true statements about natural numbers can be proved, the next obvious question is whether there are interesting mathematical statements with this property. Actually, soon after his Incompleteness Theorem, Gödel did ask for an example of a combinatorial undecidable statement - a problem which was not settled for another forty years. If we had a sensible, not necessarily formal, definition of significance, very likely Gödel would have asked for an example of a significant undecidable statement instead. It seems now that the first person to address the issue of significance in a very specific manner was Wang - a student of Gödel, whose prime interest was in computational aspects of proof theory, incompleteness results, and philosophy. Wang is also the most authoritative historian of Gödel's (very platonic) mathematical and philosophical views.

yet, and automated theorem proving went in a direction completely different from human-invented proofs. However, the ultimate, as far as I am concerned, in automated conjectures would be finding beforehand totally unexpected truths which are nevertheless, post factum, obvious or easy to prove [7].

In the case of the original version of *Graffiti*, the question was whether it was possible to write a program capable of making conjectures, without leaving any doubts that they can be credited to a machine. From the point of view of machine intelligence, crediting a program with the user's contributions defeats the whole idea and purpose of AI. Turing was quite explicit on this point, and he went as far as to write: "we wish to exclude from the machines men born in the usual manner," and fifty years ago, he was concerned enough about the issue of the definition of the machine to discuss the role of clones in his imitation game, [50], p. 56.

There would not be any need for getting into this subject at all, if Hansen's claims did not create unnecessary confusion about what constitutes a conjecture-making program, and if they did not reflect on the performance of *Graffiti*; his statements might be construed as a confirmation of occasionally expressed preconceived notions that I was testing my own conjectures with the program or that I was making my own conjectures based on the performance of *Graffiti*. His discussion of "computer-assisted and automated conjecture-making" only confuses the issue, because the principal feature of automated conjectures is that the programs that invent them are not assisted by their users. Not making a clear distinction between computer-assisted and automated defies the mathematical tradition of clarifying rather than confounding the issues. Programs like GRAPH, and AGX when run in what Hansen calls "computer-assisted mode" (which is most of the time, at least as far as the findings obtained with this program are concerned), compute data or draw pictures, and then users make conjectures on the basis of this output. With *Graffiti*, it is the other way around - all conjectures, unless it is clearly acknowledged, are made by the program, and this includes conjectures whose sorting pattern was later found with the interactive version of *Graffiti*. Finding sorting patterns could be, of course, easily automated, but so far, at least until very recently (in [23],) there was no need for it.

Quite often, the issue of authorship of conjectures of *Graffiti* would culminate in hair splitting debates, going as far as questioning the origin of conjectures, since (after all,) the program was written by a human. This much more subtle point, among others, increased my interest in fully automated versions of the program, discussed in [16] and [17]. However, the insinuation that INGRID and GRAPH did not receive due credit in discussions of conjecture-making programs is of a completely different nature, since with characteristic, admirable, particulary under the circumstances, clarity, the authors of INGRID wrote in the section, "Helping Test Conjectures and The Temporary Theorem Feature", that their program can be a valuable tool for testing conjectures, without leaving even the slightest doubt that the conjectures themselves were proposed by users, ([3], p.170, section 4.4).

[7] An example of this might be a conjecture made by the educational version of Graffiti, namely the statement that 1 plus the number of repetitions in the coding sequence of a planar Eulerian graph is equal to the number of its faces, [16].

Specifically, Brigham and Dutton wrote: *"Another interesting use is testing conjectures. In this situation, a proposed conjecture can be entered in INGRID's database on a temporary basis. Then, values can be set for appropriate invariants. If an inconsistency occurs, i.e., when INGRID determines that some invariant has an empty interval, one can conclude that the proposed conjecture is false."* Further, they add *"Considerable experimentation and analysis eventually led to the following theorem ... Again it should be emphasized that INGRID did not produce the theorem, but it provided a direction for its more conventional derivation."*

Furthermore, in section 4.3, "Helping Test Effectiveness Of New Theorems", describing how they obtained a new proof of a (recent, at the time) result of Stanley, they write: *"When a new inequality relating graph invariants is discovered, INGRID can be employed to determine if the same or better bounds can be obtained from known results".* They explain in this section, that by studying two parameters e and λ in a newly discovered inequality involving these parameters, they found a better bound: *"by varying e in INGRID, observing λ and tracing the cause of λ's change, we found that two theorems currently in INGRID's database ... were usually better, and never worse than the newly discovered bound]".*

A new proof of a result of Stanley, in which a computer played a crucial role, is undoubtedly a rewarding accomplishment, but it is an accomplishment in automated theorem-proving, not conjecture-making.

On page 4 of his paper, Hansen states that the companion publication, [28] explains that some of the functions of INGRID can be viewed as "obtaining particular types of conjectures". One would expect that, given the deliberate description of the process by Brigham and Dutton, leaving no doubts that the critical statements were not invented by the program, one might choose to present a clear and careful argument, explaining the reasoning for a different point of view.

The companion paper does not contain any such explanation. Instead, it simply refers to one of two theorems proved by Brigham and Dutton with the help of the knowledge database of *INGRID*, as a conjecture. These results were described as *rigorously fitting* the definition of the term conjecture quoted in both of Hansen's papers: *"a priori hypothesis on the exactness or falseness of a statement of which one ignores the proof."*.

Exactly the opposite is true. In neither of the two examples was the proof ignored. As a matter of fact, Brigham and Dutton almost went out of their way to clarify possible ambiguities about the origin of the two results, stating: *"INGRID does not of itself find new theorems relating graph invariants, but it can be a valuable tool in aiding researchers to do just that.",* ([3], p. 170, section 4.2. "Helping Derive New Theorems"). The above quote is the opening line of a less than half a page section discussing the two results, which, according to Hansen, rigorously fit the definition of conjecture.

According to one of the authors of [28], before INGRID proved the formula in question, the program must have conjectured it. The authors take for granted the statement from the second page of [27], according to which, "Clearly theorems are first conjectures, possibly known as such only to those who proved them". This is not always true even for theorems proved by humans, but it is conspicuously not true for theorems proved by computers. Consider the following simple problem: find all solutions of the equation $x - 1 = 0$. If Hansen's claim were correct, before proving that $x = 1$, the program would have to first conjecture this fact. A reader who considers this argument artificial should be aware that one can write a program which will eventually print all theorems (and only theorems) of any computable mathematical theory. Indeed, as long as the set of axioms of a formal system T is computable, the theory of T is recursively enumerable.

The point is that human conjecture-making is a mental act; when this activity is attributed to programs, it should be expected of the authors to clearly explain what they mean by this, and that was done in the first writings about $Graffiti$, contrary to Hansen's claim that readers of my papers were misled by this terminology [8].

Hansen implicitly, or sometimes even explicitly (see footnote 4 of this paper), refers to human conjectures that were tested or based on the performance of computer programs as conjectures invented by computers, and he seems to, on a regular basis, refer to statements, most of which were proved before they were announced, as conjectures. Even fewer open statements, perhaps just one or two, were made by AGX run in what Hansen refers to as automated mode. This is hardly ever done by mathematicians. As a rule, nobody refers to statements proved beforehand, as conjectures. Mathematical conjectures, by their very nature, satisfy Popper's Falsifiability criterion, which was invented against practices of making predictions with the benefit of hindsight.

It should be mentioned here that a few statements that had been proved by myself beforehand, were included in *"Written on the Wall"* (WoW) - a list of conjectures of $Graffiti$ - also, but I was always stressing that this was done only because these propositions could be suitable for classroom exercises, [15]. Some of the conjectures from WoW turned out to be also easy to prove or refute, but that was an unavoidable consequence of the method of selecting conjectures for inclusion in WoW. Excluding classroom exercises, I was making a reasonable effort to select conjectures for this list at random, if I could not prove nor refute them out of hand, by which I meant that the solution was not obvious to me at once, after understanding the statement of the conjecture. The sheer number of conjectures of $Graffiti$ makes suggestions that I was testing my own ideas with the program unreasonable. To make this point, conjectures were collected (with approximate dates) in WoW, and for several initial years of development of the program, they were promptly being sent to a few interested colleagues. Until today, I cannot think of a better way of showing that automated conjectures (if such a claim is made) can be bona fide credited to the program rather than to the user, than to announce substantial quantities of formulae printed by the program without working on them. Clearly, in all claims attributing intelligent behavior to computer programs, the onus is on the authors of these programs to dispel any doubts, which these claims are almost certain (and rightly so) to invite.

[8]This is one of the very few claims which were removed from the revised version, but are nevertheless discussed here, because of their relevance to the main issue. Another example is the proportion of trivially true conjectures of the Dalmatian version discussed earlier.

In the past, as far as *Graffiti* was concerned, I did not see any better way of accomplishing this task than by maintaining *WoW*. The dates of this document show that from November '87 to November '89, about 650 conjectures were placed on the list, [15] conj 69 - 715. Probably very few (if any at all) mathematicians (regardless of whether they used computers to assist them in this process) ever made that many conjectures in a two-year period, even if one included in this category every proposition suggested informally to a colleague or a student. In the case of AGX, in a six-year period, the users of the program wrote about 50 (according to Hansen) statements they found with this program. Some of these fifty statements were trivial, and most of them were never announced as conjectures.

The interaction with *Graffiti*, wherever and whenever it was taking place, has always been brought to the attention of the readers. When conjectures made by early versions of the program were reported, the readers were always informed that most of its conjectures were trivially true, and that the selection of possibly less trivial conjectures was done by myself rather than by the program. Similarly, it was always stressed that counterexamples were provided interactively by users. Actually, some of the examples were obtained by the companion program *Autograph*, and sometimes, as for example in the case of number-theoretical conjectures, examples were generated by the program. Automated procedures for building examples were discussed in the very first paper about *Graffiti*, and *Autograph* was discussed in [20]. *Autograph* was a procedure of *Graffiti*, but for clarity I refer to it as a separate program, which of course does not change anything.

Finding counterexamples to some, actually many, conjectures of *Graffiti* was never a problem, and the automation of this part of the program would be totally useless, at least at the current stage of the development of the program, contrary to Hansen's claim that several hundred examples were not enough. Some of the best conjectures of *Graffiti* were obtained in runs involving not more than fifty or even twenty examples. The point is that in order to generate more or perhaps better conjectures, all that *Graffiti* needs (in practice, not in principle,) is barely one counterexample. In any case, the most successful, initially totally ignored in [27], effort of finding counterexamples was accomplished by a team from Los Alamos National Laboratory, who refuted about 40 of about 200 conjectures they tested, using for this purpose simply the catalog of all at most 10-vertex graphs, [2]. The list could be easily incorporated into the program, but there was never any need for it. Discussing examples found with AGX, Hansen explains that no program is capable of finding counterexamples with a million vertices. As far as I know, there are no programs capable of finding all 11-vertex counterexamples to just conjectures of Graffiti. On the other hand, a summary of seven rounds of a run of the Little Red Riding Hood version of Graffiti (see table 1 of [17]) shows how few simple graphs were needed to assure correctness of about half of the conjectures. The largest of these examples has 4 vertices.

The idea of withholding from the program all but the simplest counterexample to one of its conjectures led to *Little Red Riding Hood*, developed initially just for educational purposes and later as a possibility of developing fully automated conjecture-making programs, involving no theorem-proving at all, [16], [17]. It was difficult not to entertain vague ideas of this kind from the very beginning of writing

Graffiti, but my interest in purely conjectural mathematics was reinvigorated when Larson pointed out to me that Putnam had also considered similar possibilities in [42] and [43]. In a sense, so did Erdös, [22]. Putnam referred to his ideas as quasi-empirical mathematics, and in this sense *Little Red Riding Hood* is purely empirical.

For similar reasons, in [19] it was emphasized that the program was not inventing the "needed" concepts. The invention of concepts for their own sake does not present any difficulties. Every false conjecture defines a new concept, and sometimes these concepts were incorporated into *Graffiti* to make conjectures about objects with the corresponding property. Every true inequality defines a class of objects for which the corresponding equality holds true, and these concepts were also used to make conjectures. Actually, two versions of *Graffiti*, (*Forever* and *Whatever*), written jointly with DeLaVina shortly before the Dalmatian version, were generating such concepts automatically, and they were automatically generating background for the heuristic *Echo* for conditional conjectures. The properties were defined as certain collections of objects satisfying formulae generated by *Graffiti*. *Forever* and *Whatever*, particularly the latter, were hardly ever run—because they were generating conditional conjectures at a pace overwhelmingly greater than could be stored in disc files, crashing the system after fairly short runs. Printing, not to even mention reading them, was simply unfeasible.

Hansen's paper contained a number of false statements, and the most notorious among them were those that do not distinguish between the earliest and the later versions of *Graffiti*. Even statements about early versions of the program were sometimes flatly false. Graphs, for example, were never represented in databases of *Graffiti* by adjacency matrices. The program, from the very beginning, had a companion offshoot *Autograph* for finding counterexamples to some of its conjectures. *Graffiti*, for years, had many features that Hansen proposes as novel enhancements; some of his proposed enhancements for AGX included tasks as simple as adding functions for computing the union of two graphs, and some other are even more trivial. Most of the statements which Hansen refers to as conjectures made with AGX seem to be users findings which, in most cases, were proved before they were "announced" as conjectures [9].

Hansen insists that a claim I made about "Little Red Riding Hood" in [16], "does not appear to be true", and he claims to have a counterexample to the statement that in each of its (completed) rounds, the program will make at least one false conjecture, unless $P = NP$. This statement is completely obvious from

[9] I am indebted to Gilles Caporossi for pointing out to me, in the Summer of 2003, that the first time the claim that AGX is a conjecture-making program was made in [55]. AGX was written by Caporossi as a PhD project in '97 under the direction of Hansen, who became the main spokesman for the program. According to [55], AGX rediscovered Kepler's, Newton's and a number of other laws faster than Bacon. Kepler's and other laws were "rediscovered" by finding a basis of affine relations between the variables involved by taking their logarithms. By using the same method, the authors obtained a conjecture, according to which the independence number of trees minimizing their largest eigenvalue with respect to a fixed partition of vertices into two independent sets is a linear combination of the number of edges, the number of endpoints, the radius and the diameter. According to [56], *AGX* has also rediscovered Graffiti's conjecture 146, proved ten years earlier by Shearer.

the definition of bingo conjectures. Suppose that the domain of consideration is the class of all graphs, and after completing a round the program made conjectures

$$\alpha \leq \beta_i, \qquad\qquad i = 1..k$$

The correctness of bingo conjectures corresponding to this round means that each of the conjectured inequalities is correct and moreover, that for every graph there is $i \leq k$ such that $\alpha(G) = \beta_i(G)$. Clearly, the correctness of a bingo conjecture implies the existence of a polynomially computable formula for an NP-hard invariant α, if all β_i are polynomially computable. Hansen's purported counterexample is false, because in the situation described by him the program would have made the conjecture that the independence number

$$\alpha = n - 1$$

not, as he claims, $\alpha \leq n - 1$ [10].

Apart from factually false claims, Hansen's paper contains other highly questionable statements. The main point is however, that as far as I can remember, there was not a single exception to the obvious and crucial distinction between conjectures of my own and conjectures of *Graffiti*. The difference always was (and if there is a single omission, it will be) acknowledged. Any suggestion to the contrary, and in particular, any suggestion, direct or not, to the effect that I was verifying my own conjectures or attributing them to *Graffiti*, is simply not true.

[10] I should add that, contrary to his acknowledgements, I have never discussed Hansen's paper with him. Nevertheless, I did volunteer several times to explain to him why my statement concerning Little Red Riding Hood was correct. All of these offers were rejected.

References

[1] Noga Alon and Paul Seymour, A Counter-Example to the Rank-Coloring Conjecture, Journal of Graph Theory, 13, 523-525, (1989).

[2] Tony Brewster, Michael Dineen, and Vance Faber, Computational Attack on Conjectures of Graffiti, Discrete Mathematics, 147 (1955) 35 - 55.

[3] Brigham and Dutton, *INGRID*, A Graph Invariant Manipulator, J. Symbolic Computation (1989) 7, 163 - 177.

[4] N. Chomsky, Language and Thought, Anshen Interdisciplinary Lectures in Art, Science and the Philosophy of Culture, Moyer Bell, 1993.

[5] Fan Chung, The Average Distance is not more than the Independence Number, Journal of Graph Theory, 12, 229-235, (1988).

[6] H. M. S. Coxeter, Virus Macromolecules and geodesic domes, A Spectrum of Mathematics, Oxford University Press, (1971), 98 - 107.

[7] Cristian DeDuve, A Guided Tour of the Living Cell, Scientific American Library, 1984.

[8] Ermelinda DeLaVina, Graffiti.pc, Graph Theory Notes of New York Academy of Sciences, XLII, 26-30, (2002).

[9] Ermelinda DeLaVina, Some History on the Development of Graffiti, preprint, (2002).

[10] Hubert L. Dreyfus, What Computers Can't Do: A Critique of Artificial Reason. New York, Harper and Row.

[11] Hubert L. Dreyfus, What Computers Still Can't Do: A Critique of Artificial Reason. MIT Press 1992.

[12] Paul Erdos, Joel Spencer and Janos Pach, On the Mean Distance Between Points of a Graph, Ars Combinatoria, 64, 121-124, (1988).

[13] Siemion Fajtlowicz and Craig Larson, Graph-Theoretical Independence as a Predictor of Fullerene Stability, Chemical Physics Letters, 377 (2003) 485 - 490.

[14] Siemion Fajtlowicz, Fullerene Expanders, a list of Conjectures of Minuteman, available from the author.

[15] Siemion Fajtlowicz, Written on the Wall, a list of Conjectures of Graffiti, available from the author.

[16] Siemion Fajtlowicz, Toward Fully Automated Fragments of Graph Theory, Graph Theory II, preprint 2002.

[17] Siemion Fajtlowicz, Toward Fully Automated Fragments of Graph Theory, Graph Theory Notes of NY Academy of Sciences, XLII (2002), 18-25.

[18] Siemion Fajtlowicz, On Conjectures of Graffiti, II, Congeressus Numerantium, 60 (1988) 189-187.

[19] Siemion Fajtlowicz, On Conjectures of Graffiti V, Proceedings of 7th International Quadrennial Conference on Graph Theory, Combinatorics and Applications, Western Michigan University '95, p. 367- 376.

[20] Siemion Fajtlowicz, On Conjectures and Methods of Graffiti, 4th Proceedings of Clemson Miniconference, (1989).

[21] Siemion Fajtlowicz, On Representation and Characterisation of Fullerene C_{60}, Graphs and Discovery,

[22] Siemion Fajtlowicz, Examples are Forever, available from the autor's website.

[23] Siemion Fajtlowicz, Pony Express, a list of benzenoid conjectures of Graffiti, http://www.math.uh.edu/ siemion/pony.html

[24] P. W. Fowler, K. M. Rogers, S. Fajtlowicz and P. Hansen, Facts and Conjectures about Fullerene Graphs: Leapfrog, Cylinder and Ramanujan Fullerenes", ALCOMA 2000 conference, 134 -146.

[25] P. W. Fowler, P. Hansen, G. Caporossi, A. Soncini, Polyenes with maximum HOMO-LUMO gap, Chemical Physics Letters, 342(2001) 105 - 112.

[26] Pierre Hansen, Computers in Graph Theory, Graph Theory, Graph Theory Notes of New York Academy of Sciences, XLIII, 20 - 34.

[27] Pierre Hansen, How Far Should, Is AND Could Be Conjecture-Making Automated in Graph Theory, Les Cahiers du GERAD, G-2002-44, August 2002.

[28] Pierre Hansen, A. Aouchiche, G. Caporossi, D. Stevanovic, What Forms Do Interesting Conjectures Have in Graph Theory, Les Cahiers du GERAD, G-2002-46, August 2002.

[29] P. Hansen and H. Melot, Variable Neighborhood Search for extremal graphs 9. Bounding the irregularity of a graph, Les Cahiers de GERAD, preprint 2002.
[30] V. W. Marek and Jan Mycielski, Foundation of Mathematics in Twenthiest Century, http://spot.colorado.edu/ jmyciel/
[31] Pamela McCorduck, Machines Who Think. W. H. Freeman, 1981.
[32] Craig Larson, Intelligent Machinery and Mathematical Discovery, Graph Theory Notes of New York Academy of Sciences, XLII, 26-30, (2002).
[33] Craig Larson, An Updated Survey of Research in Automated Mathematical Conjecture-Making, preprint 2002.
[34] P. Langley, H. A. Simon, G. L. Bradshaw, and J. Zytkow, Scientific Discovery, The MIT Press, '1987.
[35] P. Langley, The Computer Aided Discovery of Scientific Knowledge, Discovery Science, Proceedings of the First International Conference on Discovery Science, Lecture Notes in Artificial Intelligence, 25 - 39, (1998).
[36] Arnold L. Levine, Viruses, Scientific American Library, 1992.
[37] L. Lovasz and M. Plummer, Matching Theory, North-Holland, 1986,
[38] Roger Penrose, Shadows of the Mind, Oxford University Press, 1994.
[39] Roger Penrose, The Large, the Small and the Human Mind, Cambridge University Press, 1997.
[40] Ryan Pepper, On New Didactics of Mathematics-Learning Graph Theory via Graffiti, preprint, (2002).
[41] Hillary Putnam, The Best of All Possible Brains?, The New York Times Book Review, Nov.20,1994, p.7.
[42] Hillary Putnam, Mathematics Witout Foundations, Mathematics, Matter and Method, Cambridge University Press, 1975
[43] Hillary Putnam, What is Mathematical Truth, Mathematics, Matter and Method, Cambridge University Press, 1975.
[44] Ran Raz and Boris Spieker. On the "log rank"- Conjecture in Communication Complexity, Combinatorica, 15(4)(1995) 567-588.
[45] G. Ritchie and F. Hanna, AM: A Case Study in AI Methodology, Artificial Intelligence 23(3), 1984, 340-268.
[46] L. Scott, M. Boorum, B. J. McMahon, S.Hagen, J. Mack, J. Blank, H. Wegner, A. de Meijere, A Rational Synthesis of C_{60}, Science 295 (22 February 2002).
[47] J. R. Searle, The Rediscovery of Mind, MIT Press, Cambridge, Massachusetts, 1992.
[48] Simic-Kristic, Effects of $C_{60}(OH)_{24}$ on Microtubule Assembly, Journal of Oncology, 5, 143, (1997)
[49] Clifford J. Steer and John Hanover, Intracellular Trafficking of Proteins, Cambridge University Press, 1991.
[50] Allan M. Turing, Computing Machinery and Intelligence, in "Mind's I", Basic Books 1981.
[51] Hao Wang, Toward Mechanical Mathematics, in: Classical Papers on Computational Logic, 1957 - 1966, J. Siekmann and G. Wrightson, eds, Spriger-Verlag, 1983, 244-264.
[52] Hao Wang, Reflections on Gödel, MIT Press, Cambridge, Massachusetts.
[53] Hao Wang, Computer Theorem Proving and Artificial Intelligence, in "25 years of Automated Theorem Proving", Contemporary Mathematics, AMS, 1984.
[54] Joseph Weizenbaum, Computer Power and Human Reason, ¿From Judgement to Calculation, W. H. Freeman and Company, 1976.
[55] Gilles Caporossi and Pierre Hansen, Finding Relations in Polynomial Time,Proceedings of the Joint International Joint Conference on Artificial Intelligence, Stockholm Sweden, July 31 - August 6, 780 - 785.
[56] Gilles Caporossi, Dragos Cvetkovic, Ivan Gutman, and Pierre Hansen, Variable Neighborhood Search for Extremal Graphs 2. Finding Graphs with Extremal Energy, J. Chem. Inf. Comput. Sci 199, 39, 984 - 996.

Novemer 30, 2003

Department of Mathematics, University of Houston, Houston, Texas 77204
E-mail address: math0@bayou.uh.edu