12. D. E. Knuth, The Art of Computer Programming, vols. 1-3, Addison-Wesley, 1968, 1969, 1973.
13. ______, Computer science and its relation to mathematics, this MONTHLY, 81 (1974) 323–343.
14. L. S. Levy, Discrete Structures of Computer Science, John Wiley, 1980.
15. MAA Committee on the Undergraduate Program in Mathematics, A General Curriculum in Mathematics for Colleges, CUPM, 1965.
16. ______, Recommendations for an Undergraduate Program in Computational Mathematics, CUPM, 1971.
17. ______, Recommendations on Undergraduate Mathematics Programs in Computing, CUPM, 1972.
18. MAA, PRIME 80, Proceedings of a Conference on Prospects in Mathematics Education in the 1980s, 1978.
19. J. E. McKenna, Computers and experimentation in mathematics, this MONTHLY, 79 (1972) 294–295.
20. National Research Council, The Role of Applications in the Undergraduate Mathematics Curriculum, 1979.
21. H. O. Pollak, quoted in The Chronicle of Higher Education, 15 May 1978, p. 3.
22. A. Ralston, Computer Science, Mathematics and the Undergraduate Curricula in Both, Technical Report 161, Department of Computer Science, SUNY at Buffalo, 1980.
23. A. Ralston and M. Shaw, Curriculum 78—Is computer science really that unmathematical?, Comm. ACM, 23 (1980) 67–70.
24. J. B. Rosser, Mathematics courses in 1984, this MONTHLY, 79 (1972) 635–648.
25. D. F. Stanat and D. F. McAlister, Discrete Mathematics in Computer Science, Prentice-Hall, 1977.
26. O. E. Taulbee and S. D. Conte, Production and employment of Ph.D.'s in computer science, Comm. ACM, 20 (1977) 370–372.
27. J. P. Tremblay and R. Manohar, Discrete Mathematical Structures with Applications to Computer Science, McGraw-Hill, 1975.

MACHIAVELLI AND THE GALE-SHAPLEY ALGORITHM

L. E. DUBINS AND D. A. FREEDMAN
Department of Statistics, University of California, Berkeley, CA 94720

Summary. Gale and Shapley have an algorithm for assigning students to universities which gives each student the best university available in a stable system of assignments. The object here is to prove that students cannot improve their fate by lying about their preferences. Indeed, no coalition of students can simultaneously improve the lot of all its members if those outside the coalition state their true preferences.

1. Introduction. The object of this paper is to generalize the following result of Gale and Shapley [1]. For simplicity, suppose first that there are equal numbers of students, denoted

---

This paper describes joint research, but Freedman is mainly responsible for the exposition, including this footnote.

Before he completed his studies for his doctorate in mathematics at the University of Chicago in 1955, Lester Dubins had solved the problem of minimizing the length of a planar curve subject to boundary conditions and a curvature constraint and had settled a conjecture concerning an infinite game with incomplete information. His thesis, written under the guidance of Irving Segal, was of a more abstract nature and concerned a Radon-Nikodým derivative for Banach-space-valued random variables. After spending two years at the Carnegie Institute of Technology, he enjoyed three NSF postdoctoral fellowship years, two of which were spent at the Institute for Advanced Study at Princeton. Since 1960, he has been on the faculty of the University of California, Berkeley. His research, influenced by Leonard J. Savage and Bruno de Finetti, is largely concerned with the mathematics of (subjective and finitely additive) probability with excursions into other areas, particularly geometry, is oriented toward problems which are intuitive and concrete, and is frequently collaborative. He coauthored a book with L. J. Savage which was published by McGraw-Hill in 1965 under the title *How to Gamble if You Must* and reprinted by Dover in 1976 as *Inequalities for Stochastic Processes*.

David Freedman got his Ph.D. at Princeton in 1960; his thesis supervisor was William Feller. He spent a postdoctoral year at Imperial College, London. In 1961, he joined the faculty of the University of California, Berkeley. He has published many articles, some monographs, and an elementary textbook. His theoretical research is mainly in probability and statistics, on representation theorems like de Finetti's, on the sampling behavior of histograms, and on bootstrap methods for confidence intervals or significance levels. His applied work is in the legal area and in the validation of energy data and energy models.—*Editors* generally by S's, and universities, denoted by U's. Suppose that each university is to admit exactly one student. (More realistic assumptions are made in Section 4 below.) Each student rank orders all the universities, and each university rank orders all students. The object is to pair the students and universities off in a *stable* way. By definition, an instability is created by two pairs, S-U and S'-U', where S prefers U' to U, and reciprocally U' prefers S to S'. Nothing is assumed about the preferences of S' and U. If there are no instabilities, the system is said to be stable. Gale and Shapley prove the existence of a stable system of assignments.

$$\begin{tikzcd} S \arrow[r, dash, dashed] \arrow[dr, dash] & U \\ S' \arrow[r, dash, dashed] & U' \end{tikzcd}$$

Each student S has an "available set" $A(S)$ of universities: the ones S can get under some stable assignment. These available sets are nonempty. Consider assigning to S that university in $A(S)$ that S likes best. Gale and Shapley prove that this assignment is one-to-one, and stable.

Here is a sketch of a proof of the Gale-Shapley results which differs from theirs in detail only, but introduces some ideas needed later. Imagine the universities (much reduced in size) lined up in a room, with the students waiting outside in a hall. One student, S, walks into the room and applies to the university S likes best: this completes move #1. Then another student walks in and does likewise; in case both apply to the same university, it keeps the preferred applicant and rejects the other, who goes back outside to the hall: this completes move #2. And so on: student S_j applying to the university S_j likes best—among those that have not previously rejected S_j.

There are two rules to observe.

(1) If there are still students outside in the hall, one, say S_j, goes into the room and applies to that university which S_j likes best, among those which have not previously rejected S_j. This initiates a move.

(2) A university with two applicants keeps the preferred one and rejects the other, who goes back outside to the hall. This completes a move.

Any sequence of moves made in obedience to rules (1) and (2) will be called a "Gale-Shapley algorithm."

(3) THEOREM. *Any Gale-Shapley algorithm terminates. At termination, the students and universities are paired off, one-to-one. This pairing is stable. And, in fact, each student S will be paired with the university S likes best in $A(S)$.*

Theorem (3) will be argued in a moment, but first a statement of the new results. Suppose a student, called Machiavelli, lies, that is, does not apply to the universities in the order of true preference. Can this help Machiavelli? The answer is no, not if the others continue to tell the truth. Similarly for coalitions of student liars. For universities, however, it is another story. These issues will be discussed in Sections 2, 3, and 4 below.

Proof of Theorem (3). Suppose there are n students and n universities. By rule (1) each student applies at most once to each university. Consequently:

(4) A Gale-Shapley algorithm terminates in n^2 moves or less.

Clearly, rules (1) and (2) imply:

(5) Each student applies to successively less desirable universities. For each university, however, the applicants look better and better.

At the end of every move, there are some students in the hall, and an equal number of universities in the room, who have not yet had applications. The remaining students and universities are paired off, one-to-one. After a university gets its first application, it always has one. Furthermore:

(6) The algorithm ends when each university has had at least one application.

Next, it will be argued by induction that:

(7) At the end of every move, the pairing in the room is stable.

Plainly, this is so before move 1. Suppose it is so before move k, and consider the assignment at the end of that move. Now there cannot be two pairs S-U and S'-U', where S prefers U' to U, while U' prefers S to S'. For if S prefers U' to U, then S has already applied to U' and been rejected, by rule (1). Now U' must prefer the current applicant S' to the previous one S, by the fact (5). This completes the proof of (7).

$$\begin{array}{l} S\text{---}U \\ S'\text{---}U' \end{array}$$

The next point, though similar, is a bit trickier.

(8) If a student S is rejected by a university, that university is not in S's available set.

This is vacuous at move 1. Suppose it were so for moves 1 through $k-1$, and S is paired with U at the end of move $k-1$. On move k, suppose S' applies to U. Now U must retain one of these two applicants, say S_1; call the rejected applicant S_2. By way of contradiction, suppose there were a stable assignment in which S_2 got U. Now S_1 has to get some university, call it U'. At the risk of the obvious, S_1 and S_2 are different students; U and U' are different universities.

Case 1: S_1 applied to U' before move k. Then S_1 must have been rejected by U', because S_1 is applying to U on move k. So this system is unstable, by the inductive assumption.

Case 2: S_1 did not apply to U' before move k. Now S_1 prefers U to U', by rule (1). And U prefers S_1 to S_2, the proof being that it rejected S_2. Again an instability.

$$\begin{array}{l} S_1\text{---}U' \\ S_2\text{---}U \end{array}$$

To sum up, the algorithm terminates by (4); the resulting system is stable by (7); and it is optimal for the students by (8). This completes the proof of the theorem. □

2. Enter Machiavelli. One of the students—named M for Machiavelli—will now be treated differently from the rest. M has some true rank ordering on the universities, and if M participates in a Gale-Shapley algorithm following rule (1), M will get some university: the best in M's available set. This is *fair play*. But now permit Machiavelli to lie, that is, to use some false rank ordering. This is *foul play*.

(9) THEOREM. *Suppose M participates in a Gale-Shapley algorithm, but uses a false rank ordering. The university M gets by this foul play is no better—measured by M's true rank ordering—than the one M would have got by fair play.*

For the proof, imagine that M waits outside in the hall until all the others have paired off. This will be called the *prologue*. At the end of the prologue, there will be one university, call it W, which has not yet received an application. M now enters and starts applying in accordance with the rules—but using the false rank ordering. Clearly,

(10) The algorithm terminates when W gets its first application.

No generality is lost by assuming that M does not move until the others are paired off: as Theorem (3) shows, all Gale-Shapley algorithms lead to the same system of assignments. (The algorithm is now being applied with M's false rank ordering in place of M's true one.)

The main step in the proof of Theorem (9) is Lemma (11) below, which requires two definitions. A *scenario* is a sequence of applications for M—an initial segment of a rank ordering. One scenario, for instance, is specified by naming three universities:

$$A \quad B \quad C.$$

The interpretation: M applies first to A; if rejected, M tries B next; if rejected there too, M goes on to C. In general, a scenario is specified by a list of universities; no university appears twice on the list, but the list need not be exhaustive. The action called for by a scenario stops when

- either M is rejected by the last university specified in the scenario (C, in the example);

or

- the whole algorithm stops, W getting its application.

Corresponding to each scenario, there is a script that tells exactly what happens as the action unfolds, after the prologue. A script can be written in standard form as in Table 1.

TABLE 1. Standard script

Question marks indicate that the objects are undefined.

Line	University	rejects	Student	who applies to	University
0	???		$S_0 = M$		$U_0 = A$
1	U_0		S_1		U_1
2	U_1		S_2		U_2
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$k-1$	U_{k-2}		S_{k-1}		U_{k-1}
k	U_{k-1}		S_k		U_k
$\vdots$	$\vdots$		$\vdots$		$\vdots$

The table is interpreted as follows. To fix ideas, suppose again that the scenario is $A \ B \ C$.

Line 0. M enters and applies to A, and so S_0 is M and U_0 is A. Suppose A isn't W.

Line 1. U_0 now has two applicants and must reject one, say S_1. Then S_1 applies to another university; call it U_1. Of course, if S_1 is M, then U_1 must be B, according to the scenario. If S_1 isn't M, then U_1 is determined by S_1's rank order, in accordance with the rules.

Lines $2, 3 \ldots$ are interpreted in a similar way. The last line is special, and there are two cases.

Case 1: M is rejected by the last university in the scenario. Then the last line is:

Line	University	rejects	Student	who applies to	University
k	U_{k-1}		M		???

In our example, the scenario was $A \ B \ C$, so U_{k-1} is C.

Case 2: The last university W gets its application. The last line is

Line	University	rejects	Student	who applies to	University
k	W		???		???

In any case the table has finite length, by (4).

Note: For all $k \geqslant 1$, the first university mentioned in line k is the same as the last university mentioned in line $k - 1$, namely U_{k-1}. In general, the same student will be mentioned several times in the sequence $S_0, S_1, \ldots$; likewise, U_i and U_j can easily be the same, even if $i \neq j$.

One more definition. Consider two scenarios, #1 and #2. Then scenario #1 is *smaller* than #2 if every university mentioned in #1 is also mentioned in #2: order is immaterial. Thus, $A\ B\ C$ is smaller than $E\ B\ D\ C\ A\ F$.

(11) THE SCENARIO LEMMA. *Suppose scenario #1 is smaller than scenario #2, and that*

(12) *M makes every application indicated in the larger scenario.*

Then every rejection and application in the script for the smaller scenario occurs, sooner or later, in the script for the larger scenario.

Proof. The argument is by induction on the line number in the script for the smaller scenario. In line 0, M comes in and applies to U_0; by assumption (12), this application occurs in the script for the larger scenario. Now make the inductive assumption:

(13) All the rejections and applications in lines 0 through $k - 1$ of the script for the smaller scenario occur, sooner or later, in the script for the larger one.

Consider line $k \geqslant 1$ of the script for the smaller scenario. To avoid trivialities, suppose this isn't the last line of the table. It will be shown that the rejection and application in turn occur in the second script as well:

line k U_{k-1} rejects S_k who applies to U_k.

Line k of the script for the smaller scenario begins with university U_{k-1} rejecting student S_k. So S_k must already have applied to U_{k-1}: either in the prologue, or in lines 0 through $k - 1$ of the script. If not in the prologue, this application must occur somewhere in the script for the larger scenario, by inductive assumption (13). Furthermore, according to rule (2), university U_{k-1} must have been applied to by a student preferred to S_k, either in the prologue or in lines 0 through $k - 1$ of the script for the smaller scenario. If not in the prologue, this application too must occur somewhere in the script for the larger scenario. The upshot is that under the larger scenario, poor S_k must again be rejected by U_{k-1}. This event does not occur in the prologue, by assumption: so it must occur in the script.

Line k of the script for the smaller scenario ends by having S_k apply to U_k. There are two cases.

Case 1: S_k is M. This application gets made in the script for the larger scenario, by assumption (12).

Case 2: S_k isn't M. Now U_k in the script for the smaller scenario is identifiable. By rule (1), this is the university ranking after U_{k-1} on S_k's list. As shown above, S_k gets rejected by U_{k-1} in the script for the larger scenario, and must then apply to U_k.

This completes the induction, except for the last line of the table. The argument there is similar, and is omitted. □

Proof of Theorem (9). Suppose that M would get M's ith choice under fair play, where $i \geqslant 2$. By way of contradiction, suppose there is some scenario

(14) $A\ B\ C \ldots U$

that gets M a university U that M ranks ahead of i. Then the corresponding foul play script must terminate with an application to W, while M is paired with U. In particular,

(15) M makes all the applications called for in scenario (14).

There are two cases to consider.

Case 1: M truly prefers all the other universities in scenario (14) to U. To get the contradiction, the foul play scenario (14) will be compared to a fair play scenario in which M applies to M's 1st, 2nd, ..., $(i-1)$th choices in turn. By the assumption defining Case 1, the foul play scenario is smaller than the fair play one, since U ranks ahead of i. And M makes every application called for in this fair play scenario: indeed,

(16) The fair play script ends with M rejected by M's $(i-1)$th choice.

The reason is that, under fair play, M gets M's ith choice.

The Scenario Lemma (11) applies, and shows that every application in the script for foul play, including the one to W, gets made in the script for fair play. In particular, the fair play script has to end with an application to W. This contradicts (16), and disposes of Case 1.

Case 2: M truly prefers U to at least one of the other universities in scenario (14). Delete all such universities, creating a second and smaller foul play scenario. The corresponding script, by Case 1, must end with M ignominiously rejected by U. This rejection must occur in the script for the original foul play scenario (14), by the Scenario Lemma (11): condition (12) is satisfied by (15). This contradiction disposes of Case 2.

REMARK. Two scenarios that are permutations of one another are equivalent, as long as M makes all the applications in both cases.

3. Coalitions. So far, M has acted independently. What happens if M colludes with other students?

(17) THEOREM. *Suppose several students collude in a Gale-Shapley algorithm, each using a false rank ordering. They cannot all get better universities. "Better" is relative to each student's true rank ordering, and indicates strict inequality.*

The proof is an adaptation of the one for (9). Now a scenario indicates separately for each liar the sequence of universities applied to. Imagine the liars to wait outside in the hall until the honest students are all paired off with universities: this defines the prologue. At the end of the prologue, some universities have not yet had applications: their number is equal to the number of liars. Now the liars take turns in any way among themselves applying to the universities, but following the scenario. Each scenario therefore can be expanded into many scripts. To avoid complications, a student who is rejected gets to make the next application, by convention.

The action initiated by a scenario terminates when

- any liar L has been rejected by the last university on L's list

or

- the whole algorithm stops.

If the action ends according to the first possibility, no honest students can be left outside.

Note too that with several liars, and therefore several universities that have not had applications in the prologue, some applications made before the end of the script do not cause rejections. Suppose one such occurs at line k of the script. Since an honest student will be found in the hall only after a rejection, and gets the next turn, line $k+1$ of the script must have an application from a liar.

Line	University rejects	Student who applies to	University
k	U_{k-1}	S_k	U_k
$k+1$	???	S_{k+1}	U_{k+1}

Thus U_k is receiving its first application: S_k may be honest or a liar. However S_{k+1} is necessarily one of the liars.

(18) THE GENERALIZED SCENARIO LEMMA. *Suppose scenario* #1 *is smaller than scenario* #2. *Expand scenario* #1 *into script* #1, *and scenario* #2 *into script* #2. *Suppose*

(19) *In script* #2, *each liar makes every application indicated in scenario* #2.

Then every rejection and application in script #1 *occurs, sooner or later, in script* #2.

Proof. Argue by induction on the line number in script #1, as in the proof of (11). □

Proof of Theorem (17). Number the liars as $L_1, L_2, \ldots$. Suppose that, under fair play, L_j's i_jth choice is what L_j would get. By way of contradiction, suppose there is a foul play script for a scenario in which L_j gets U_{jk_j}, which is strictly better than the university that L_j would get under fair play: L_j ranks U_{jk_j} above i_j. Write the scenario as follows:

$$(20)\qquad \begin{array}{ll} L_1 & U_{11}, U_{12}, \ldots, U_{1k_1} \\ L_2 & U_{21}, U_{22}, \ldots, U_{2k_2} \\ \vdots & \vdots \quad \vdots \quad\quad \vdots \end{array}$$

As before:

(21) All the applications indicated by (20) get made in the foul play script.

Furthermore, by test (6),

(22) The foul play script for (20) ends with all the universities getting applications.

Again, there are two cases to consider.

Case 1: Each L_j really ranks all the universities applied to in scenario (20) as U_{jk_j} or better. This scenario will be compared to a truncated fair play scenario, but some care is needed. To begin with, consider any definite script for fair play. The liars arrive at their final universities in some order or other. Suppose (by renumbering) that L_1 applies to the i_1th choice only after L_j applies to the i_jth choice for all $j \geq 2$. Now consider the truncated fair play scenario in which

L_1 applies to the 1st, 2nd, $\ldots$, $(i_1 - 1)$th choices, in turn;

and for $j > 1$,

L_j applies to the 1st, 2nd, $\ldots$, i_jth choices, in turn.

By the assumption defining Case 1, this truncated fair play scenario is larger than the foul play scenario (20). Furthermore, by definition, in the specific script for fair play under consideration, L_1 gets rejected by $i_1 - 1$ while L_j is paired with the i_jth choice for $j \geq 2$. In other words, all the proposals in the truncated fair play scenario above get made. Thus, condition (19) is satisfied, and the Generalized Scenario Lemma (18) applies. The conclusion is that any application generated under the script for the foul play scenario must also be generated in the script for the truncated fair play scenario. In particular, by (22) the fair play script would have to end with all the universities getting at least one application, rather than L_1 being rejected by the $(i_1 - 1)$th choice. This contradiction disposes of Case 1.

Case 2: Some L_j really ranks at least one of the universities applied to in scenario (20) below U_{jk_j}. Eliminate all such universities from the scenario, for every liar, and expand the reduced scenario into a reduced foul play script. Case 1 applies to this smaller scenario, proving that its script terminates with some liar L_j being rejected by the last university U_{jk_j}. This rejection must also occur in the original foul play script, by the Generalized Scenario Lemma (18). Condition (19) holds by (21). □

We originally thought a stronger result might hold, namely, that if one liar in the coalition does better, another liar must do worse; as stated above, (17) only implies that if one liar does better, another liar must do no better. However, David Gale showed us that the stronger result is false.

(23) EXAMPLE. With three students and three universities, two students can form a coalition and lie: one of the liars will do better, and the other will do no worse.

The students are A, B, C; the universities are U, V, W. The true rank orderings are presented in Table 2 below: W's rank orderings are irrelevant.

TABLE 2. The true rank orderings

	University preferences				Student preferences		
	1st	2nd	3rd		1st	2nd	3rd
U	B	A	C	A	U	V	W
V	A	C	B	B	V	U	W
W	?	?	?	C	V	W	U

One script for fair play is presented in Table 3 below, with a diagram for the positions of the applicants.

TABLE 3. One script for fair play

	Applicants to		
	U	V	W
A applies to U	A		
B applies to V	A	B	
C applies to V	A	B, C	
V rejects B, who applies to U	A, B	C	
U rejects A, who applies to V	B	A, C	
V rejects C, who applies to W	B	A	C
the algorithm ends			

Now suppose B and C form a lying coalition: B's lie coincides with the truth, but C orders the universities as W, V, U. As shown in Table 4 below, C will get the same university W; but B will improve from U to V. It is worth noting that the honest bystander A also does better, going from V to U. The improvement is at the expense of the universities.

TABLE 4. One script for foul play

	Applicants to		
	U	V	W
A applies to U	A		
B applies to V	A	B	
C applies to W	A	B	C

4. Variations and Comments

(24) Theorems (3), (9), and (17) apply even when the numbers of students and universities are unequal.

Suppose, for instance, there are more students than universities. There is a new kind of instability to mention: S is paired with U and S' is not admitted to any university, but U prefers S' to S. The quick fix is to introduce some additional (fictitious) universities, ranking below the real universities in every student's estimation. A similar trick works if there are more universities than students.

Now consider the more realistic case, where universities may admit more than one student apiece. Each university U has a quota $q(U) \geqslant 1$, and may not admit more than $q(U)$ students. In previous sections, $q(U) \equiv 1$. This condition is now dropped. The total number of places is $\Sigma_U q(U)$. If this sum is bigger than the number of students, some universities have unfilled quotas. If the sum is smaller than the number of students, some students do not get assigned to universities.

Rules (1) and (2) require only small modifications to handle this new situation. Students walk in, one at a time, and apply to the university of their choice; rule (1) remains in force. However, a university does not reject any applicants until their number first exceeds its quota: then it rejects the lowest-ranking applicant. This process too will be called a "Gale-Shapley algorithm."

(25) Theorems (3), (9) and (17) hold when each university has a quota.

The trick here is to clone the universities: make $q(U)$ copies of university U, each copy having a quota of 1. Each student rank orders the clones arbitrarily: however, if for instance Harvard is preferred to Yale, then all the Harvard clones must be preferred to all the Yale clones.

What happens if the universities make offers to the students instead of waiting for applications? To be more explicit, line up the students in the room, and make the universities wait outside in the hall. One at a time, the universities walk in and make offers of admission. A university may have more than one offer outstanding; however, the number of offers may not exceed its quota of places. A student who gets two offers rejects the one from the less desirable university, which is then free to make an offer to the next-ranking student. The cloning trick used for (25) proves

(26) Theorems (3), (9), and (17) apply when universities make offers to students; this time, it is the universities that cannot improve their situation by lying.

It may be worth while to state (3) carefully in this new context.

- There is a stable system of assignments of students to universities in which no university admits more than its quota of students. However, if the number of places exceeds the number of students, some universities will have unfilled quotas; if the number of students exceeds the number of places, some students will get assigned to no university.

- For each university U, consider the set $S(U)$ of students admitted to U under some system or other of stable assignments. If card $S(U) \leqslant q(U)$, give U all the students in $S(U)$. If card $S(U) > q(U)$, give U the $q(U)$ students it likes best in $S(U)$. This is a stable system of assignments, and optimal for the universities.

- The Gale-Shapley algorithm terminates in the system of assignments just specified.

When the students do the applying, the algorithm optimizes for students, and no student or coalition of students can all beat the system by lying. When the universities make the offers, the algorithm optimizes for the universities and no university or coalition of universities can all beat the system by lying.

(27) Example. Return to the original rules, with equal numbers of students and universities, each university admitting exactly one student, and the students making the applications. The original algorithm defined by rules (1) and (2) optimized for the students, and no student could beat the system by lying. However, universities can improve their position by lying. There is a situation involving three students A, B, C and three universities U, V, W, in which under honest play U would get its 2nd choice student; but by lying, it gets the 1st choice. The true rank orderings are presented in Table 5 below; W's rank ordering are irrelevant.

TABLE 5. The true rank orderings

	University preferences				Student preferences		
	1st	2nd	3rd		1st	2nd	3rd
U)	A	B	C	A)	V	U	W
V)	B	A	C	B)	U	V	W
W)	?	?	?	C)	U	W	V

One script for fair play is given in Table 6 below, with diagrams for the position of applicants.

TABLE 6. One script for fair play

	Applicants to		
	U	V	W
A applies to V	-	A	-
B applies to U	B	A	-
C applies to U	B, C	A	-
U rejects C, who applies to W	B	A	C
the algorithm ends			

Now in foul play, U rank orders the students as $A\ C\ B$. One script for foul play is given in Table 7 below.

TABLE 7. One script for foul play by university U

	Applicants to		
	U	V	W
A applies to V	-	A	-
B applies to U	B	A	-
C applies to U	B, C	A	-
U lies and rejects B, who applies to V	C	A, B	-
V rejects A, who applies to U	A, C	B	-
U rejects C, who applies to W	A	B	C
the algorithm ends			

(28) POSTSCRIPT THEOREM. *Suppose M would get M's jth choice under fair play. Now M lies. There is no assignment, stable for the lie, under which M would get M's true ith choice, where i is better than j.*

Proof. Suppose there were such an assignment. This assignment would still be stable if M revised the lie to make i the 1st choice. Then M could get into this university by participating in a Gale-Shapley algorithm with the revised lie: for the algorithm gives M the best available university: Theorem (3) applied to the revised lie. Now there is a contradiction to Theorem (9). □

Research for this paper was partially supported by National Science Foundation Grant MCS-77-01665.
We thank David Gale, Donald Knuth, and Jim Pitman for their comments.

References

1. D. Gale and L. S. Shapley, College admissions and the stability of marriage, this MONTHLY, 69 (1962) 9–15.
2. A Gibbard, Manipulation of voting schemes: A general result, Econometrica, 41 (1973) 587–601.
3. D. Knuth, Mariages stables, Les Presses de l'Université de Montréal, 1976.
4. A. E. Roth, The economics of matching: Stability and incentives, Faculty working paper no. 744, College of Commerce and Business Administration, University of Illinois, 1981.
5. M. Satterthwaite, Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions, Journal of Economic Theory, 10 (1975).