Discussion Paper No. 853

THE COMPLEXITY OF ELIMINATING DOMINATED STRATEGIES

by Itzhak Gilboa*

Ehud Kalai*

and

Eitan Zemel*

September 1989**

---

*Department of Managerial Economics and Decision Sciences, J.L. Kellogg Graduate School of Management, Northwestern University, Evanston, Illinois 60208.

**NSF Grant No. IRI-8814672 is gratefully acknowledged.

Abstract

This paper deals with the computational complexity of some yes/no problems associated with sequential elimination of strategies using three domination relations: strong domination (strict inequalities), weak domination (weak inequalities), and domination (the asymmetric part of weak domination). Classification of various problems as polynomial or NP-Complete seems to suggest that strong domination is a simple notion, whereas weak domination and domination are complicated ones.

1. Introduction

The notion of rationality, being central as it is to game theory, has naturally been interpreted in many ways, ranging from a relatively mild condition of "rationalizability" in the sense of Bernheim (1984) and Pearce (1984) to almost-omniscience in Binmore (1987, 1988). It seems, however, that whatever we mean by "rationality" would have to exclude the possibility of a player choosing a dominated strategy, thus allowing us to eliminate such strategies as a first-step towards "solving" a game. Moreover, assuming that this type of rationality is known to a certain degree among the players--let us say for simplicity that it is common knowledge--we may iteratively eliminate dominated strategies as new domination relationships appear when some of the original strategies have already been deleted.

The validity of this procedure depends, of course, on the precise definition of "domination." At least three notions of domination seem to be natural and have indeed appeared in the literature: a strategy x strongly dominates another strategy y if for whatever strategy combination of the other players x guarantees a strictly higher payoff than does y. It weakly dominates y if the inequality is weak (which allows for the possibility of y weakly dominating x). Finally, x dominates y if it weakly dominates it but the converse is false.

The extent to which game theorists seem to "believe" in these domination relations as analytical tools varies. While strong domination may have a claim to the status of "canon of rationality," domination certainly does not enjoy that status (note that there are Nash equilibria in dominated strategies), and weak domination may be dismissed as rather arbitrary (mainly on account of its failing to be antisymmetric).

In Gilboa-Kalai-Zemel (1989) we studied various properties of abstract domination relations that may guarantee the existence and uniqueness of a reduced game, i.e., a game that is obtained from an original one by iterative elimination of dominated strategies, and in which no further eliminations are possible. It turns out that for finite games, it suffices to require that such a domination relation would be a partial order (when restricted to every subgame of the original one) and hereditary in the following sense: if a strategy x dominates y in a given game, it will also dominate it in any subgame.

Obviously, strong domination is a hereditary partial order; domination is a partial order that is not hereditary; and weak domination is hereditary but is not a partial order. Indeed, the uniqueness of the reduced game is guaranteed only for strong domination, while it is not difficult to show that the other two relations may yield non-unique reduced games. (Note that the reduced game existence is immediate for finite games.)

In this paper we study these three relations from a computational complexity point of view. In a nutshell, the results show that computational problems relating to strong domination tend to be easy (i.e., polynomial), while those involving either domination or weak domination tend to be difficult (namely, NPC). It therefore turns out to be the case that strong domination, which is the most intuitive and most widely accepted relation, and the only one for which the reduced game uniqueness is guaranteed, is also a reasonable notion if one assumes that the players are rational but may have computational restrictions. As for the other two relations (on top of the conceptual objections one may raise and the theoretical flaw of non-uniqueness) one cannot adopt them as viable solution concepts (in general) without implicitly making extremely demanding assumptions on the players' computational abilities.

In a way, this paper may be viewed as another contribution to the literature on classification of game theoretic solution concepts on basis of computational complexity. Works of this nature are by Sahni (1974) (on the existence of Nash equilibrium in pure strategies where the utilities are given as polynomials), Gilboa (1988), Ben-Porath (1988), and Papadimitriou (1989) (on computation of best response automata in repeated games), Knuth-Papadimitriou-Tsitsiklis (1988) (on elimination of dominated strategies from a different viewpoint than ours), and Gilboa-Zemel (1989) (on Nash and correlated equilibria in mixed strategies for general games). The strategy we adopt here is closest to Gilboa-Zemel (1989): we formulate several yes/no problems that try to capture the notion of "solving" the game, and we analyze its complexity with respect to each one of the "competing" solution concepts (domination relations in the case at hand). The analysis stops when we get the impression that there is a certain principle underlying the results, preferably one saying that some notions are intrinsically difficult to compute and other are simple.

The problems we have used in this paper are, loosely, the following:

1. Given a game $G = (N,(S^i)_{i\in N}, (h^i)_{i\in N})$ and a number k_i for each player i, can G be reduced to a game $\hat{G} = (N,(\hat{S}^i)_{i\in N}, (\hat{h}^i)_{i\in N})$ with $|\hat{S}^i| = k_i$?

2. Given two games, G and $\hat{G}$, is $\hat{G}$ a reduction of G?

3. Given a game G and a subset of strategies T^i for each player i, can G be reduced to its subgame $\hat{G}$ defined by these subsets? (Problem 2 is more difficult than Problem 3 since the identities of strategies in $\hat{G}$ are not

given.)

We note here that other YES/NO problems of this type can also be analyzed in a similar way, and it seems that slight modifications of our proofs will establish slightly modified results. In particular, one may ask whether a given strategy (or a subset of strategies) may be eliminated in some legitimate reduction path, and for this case similar results hold.

As mentioned above, the classification of these problems (and some variants) seems to imply that strong domination is an intrinsically simple notion, while domination and weak domination are complex ones.

This paper is organized as follows. In Section 2 we provide formal definitions of the concept of games and the problems discussed. In Section 3 we present the results and some comments on possible extensions and interpretations. The proofs are given in Section 4, and Section 5 contains some remarks regarding the robustness of our results with respect to the way in which a game is presented.

2. Notations and Definitions

We will assume that the description of the game contains a linear order on each player's strategy set. W.l.o.g. (without loss of generality) we may assume that the strategies are natural numbers with the natural order. We therefore define a game to be a triple $G = (N,(S^i)_{i\in N},(h^i)_{i\in N})$ where $N = \{1,2,\ldots,n\}$ is the set of players, S^i is a finite and nonempty set of integers denoting player i's strategies, and h^i: $S \equiv \Pi_{i\in N} S^i \to \mathbb{R}$ is player i's payoff function.

A game $\hat{G} = (N,(\hat{S}^i)_{i\in N},(\hat{h}^i)_{i\in N})$ is a subgame of $G = (N,(S^i)_{i\in N},(h^i)_{i\in N})$ if for every $i \in N$ there is a (strictly) increasing function f^i: $\hat{S}^i \to S^i$ such that

$$\hat{h}^i((s_1,\ldots,s_n)) = h^i((f^1(s_1),\ldots,f^n(s_n))$$

for all $(s_1,\ldots,s_n) \in \Pi_{i\in N} \hat{S}^i$ and all $i \in N$.

Thus, if we consider a one-player game G and identify it with the payoff vector (1,2,3), the vectors (1,2) and (1,3) will correspond to subgames of G, but (3,1) will not.

Two games, G and $\hat{G}$, are said to be <u>almost identical</u> if G is a subgame of $\hat{G}$ and vice versa. Note that "almost identity" is an equivalence relation and that, given two games, one can determine in linear time whether they are almost identical, so that almost identical games may be considered identical.

Given a game $G = (N,(S^i)_{i\in N},(h^i)_{i\in N})$ and a collection $T = (T^i)_{i\in N}$ of strategy subsets $\emptyset \neq T^i \subseteq S^i$, we denote by G(T) the subgame of G corresponding to T, i.e., $\hat{G} = (N,(T^i)_{i\in N},(\hat{h}^i)_{i\in N})$ where $\hat{h}^i$ is the restriction of h^i to T^i.

For every game $G = (N,(S^i)_{i\in N},(h^i)_{i\in N})$ and every $i \in N$ we define three binary (domination) relations, which are strict dominations (>>), domination (>), and weak domination (≥). Since these relations were informally defined in the introduction, we omit their obvious (and tedious) definition.

We now turn to the definition of reduction. Given two games, $G = (N,(S^i)_{i\in N},(h^i)_{i\in N})$ and $\hat{G}$, and a domination relation dom (which may be >>, >, or ≥), we say that $\hat{G}$ is a <u>one-step dom-reduction</u> of G if there exists $T = (T^i)_{i\in N}$ with $\emptyset \neq T^i \subseteq S^i$ (for $i \in N$) such that:

(i) For every i and every $x \in S^i\setminus T^i$ there is a $y \in T^i$ such that y dom x;

(ii) $\hat{G}$ is almost identical to G(T).

Thus, one-step dom-reduction is a binary relation on games, and it is meaningful to define dom-reduction as its transitive closure. We will also use verbal variations such as "G dom-reduces to $\hat{G}$" with their obvious meaning.

A game is said to be dom-irreducible if it is almost identical to every dom-reduction of itself. A game $\hat{G}$ is a maximal dom-reduction of G if it is a dom-irreducible dom-reduction of it.

We may finally turn to the definition of the computational problems. Each of the problems stated above can be posed in two variants: for a general subgame and for a subgame that is restricted to be irreducible. Thus we have six problems, each of them parameterized by the domination relation; since we consider three domination relations we end up with eighteen problems. Formally, the parameterized problems are:

1. SIZE (dom): Given a game $G = (N,(S^i)_{i\in N},(h^i)_{i\in N})$ and positive integers $(k_i)_{i\in N}$, is there a game $\hat{G} = (N,(\hat{S}^i)_{i\in N},(\hat{h}^i)_{i\in N})$ with $|\hat{S}^i| = k_i$ such that G dom-reduces to $\hat{G}$?

2. SIZE' (dom): Given a game $G = (N,(S^i)_{i\in N},(h^i)_{i\in N})$ and positive integers $(k_i)_{i\in N}$, is there a dom-irreducible game $\hat{G} = (N,(\hat{S}^i)_{i\in N},(\hat{h}^i)_{i\in N})$ with $|\hat{S}_i| = k_i$ such that G dom-reduces to $\hat{G}$?

3. SUBGAME (dom): Given a game $G = (N,(S^i)_{i\in N},(h^i)_{i\in N})$ and a game $\hat{G} = (N,(\hat{S}^i)_{i\in N},(\hat{h}^i)_{i\in N})$, does G dom-reduce to $\hat{G}$?

4. SUBGAME' (dom): Given a game $G = (N,(S^i)_{i\in N},(h^i)_{i\in N})$ and a dom-irreducible game $\hat{G} = (N,(\hat{S}^i)_{i\in N},(\hat{h}^i)_{i\in N})$, does G dom-reduce to $\hat{G}$?

5. SUBSETS (dom): Given a game $G = (N,(S^i)_{i\in N},(h^i)_{i\in N})$ and subsets $T = (T^i)_{i\in N}$ $(T^i \subseteq S^i)$, does G dom-reduce to G(T)?

6. <u>SUBSETS'</u> (dom): Given a game $G = (N,(S^i)_{i\in N},(h^i)_{i\in N})$ and subsets $T = (T^i)_{i\in N}$ $(T^i \subseteq S^i)$ such that G(T) is dom-irreducible, does G dom-reduce to G(T)?

3. Results

The results are summarized by the following

<u>Theorem</u>: The problems SIZE (≥), SIZE'(≥), SUBGAME(≥), SUBGAME'(≥), SUBSETS(≥), SUBSETS'(≥), SIZE(>), SIZE'(>), SUBGAME(>), SUBGAME'(>), SUBSETS(>), SUBSETS'(>), SIZE(>>) and SUBGAME(>>) are NPC, while the problems SIZE'(>>), SUBGAME'(>>), SUBSETS(>>) and SUBSETS'(>>) are polynomial.

These results may be presented by the following table (where "P" stand for "polynomial"):

PROBLEM	≥	>	>>
SIZE	NPC (1)	NPC (7)	NPC (13)
SIZE'	NPC (2)	NPC (8)	P (14)
SUBGAME	NPC (3)	NPC (9)	NPC (15)
SUBGAME'	NPC (4)	NPC (10)	P (16)
SUBSETS	NPC (5)	NPC (11)	P (17)
SUBSETS'	NPC (6)	NPC (12)	P (18)

(The numbers in parentheses enumerate the facts we have to prove and will be referred to in the sequel.)

Before continuing to prove the theorem, we should make the following comments:

1. In fact, the theorem may be stated in a more general form. We may define each of these problems without a reference to a specific domination relation dom, but rather with the domination relation given as part of the input data. In general, such a domination relation is simply a sequence of lists of ordered pairs of strategies, but the number of lists we need is as large as the number of subgames of G, which is, of course, exponential in the size of the original game G. Hence a polynomial algorithm in the size of the input may still be exponential in the size of the game.

Trying to circumvent this problem we may assume that the relation dom is indeed given as input, but to compute the complexity with respect to invocations of an "oracle", that is, a procedure which, when invoked, provides an answer regarding the existence of domination between each pair of strategies. Thus we may assume that the dom relation is still a part of the problem's input, but the size of the latter is no more than the size of the game description (and the other data of the specific problem.) We may now wonder whether there exists an algorithm solving any of the problems discussed above in a number of operations which is bounded by a polynomial of the size of the input data, where each oracle invocation is considered to be a single operation.

In this framework we may generalize our results to say that in general the six problems are NPC, though SIZE', SUBGAME', SUBSETS and SUBSETS' are polynomial if the dom relation given as input is restricted to be a hereditary partial order as defined in Gilboa, Kalai and Zemel (1989). Furthermore, the first and second columns of the table above show that all these problems are still NPC if the dom relation is only restricted to be hereditary (but not a partial order, as $\geq$) or only a partial order (but not hereditary, as $>$.) Indeed, the results quoted above are sufficient to prove that the more general problems (with dom given as a part of the input) are difficult. From the proof of facts 14,16-18 in the next subsection it will be clear that the corresponding more general problems are still polynomial, as the uniqueness of the maximal reduction is all that one needs to rely on in these proofs.

2. Coming to interpret the results, it seems to us that reduction problems related to the relations $\geq$ and $>$ are intrinsically difficult, whereas the same problems are easy if we only consider $>>$ as a legitimate domination relation. True, there are two "NPC"'s on strict domination's grade sheet, but they are both referring to "untagged" problems, that is, to problems where the reduced game $\hat{G}$ is not required to be irreducible. Indeed, the proofs of these results (facts 13 and 15) will probably give the reader a feeling of "unfairness", as the difficulty of these problems is not related at all to elimination of dominated strategies. One may even claim that these problems are rather artificial and only the "tagged" problems should be considered for game theoretic applications.

4. Proofs

The time has come to admit that the number of facts stated above exceeds the number of proofs given below by a large margin. As a matter of fact, facts 1-12 all have the same proof which is given in the next sub-section, and so do facts 14,16 and 18.

4.1 Proof of Facts 1-12

We first introduce the satisfiability problem, which will be shown to be reducible to any of the six problems with either $\geq$ or $>$.

A Boolean variable x is a variable assuming values in {0,1}. A Boolean function of s variables is a function $f: \{0,1\}^s \rightarrow \{0,1\}$. For each variable x_i we define two Boolean functions, denoted by x_i (with an obvious abuse of notation) and $\bar{x}_i$. These functions are called Basic functions. The function x_i assumes the value 1 iff the i-th component of its argument is 1. (In other words, it is the projection function with respect to the i-th component.) The function $\bar{x}_i$ equals 1 iff x_i equals 0. Boolean expressions are generated from Boolean functions using the operations + (for logical OR) and * (for AND), and brackets understood in the obvious way. A factor is an expression which is the sum of basic functions. W.l.o.g. (Without Loss of Generality) we will assume that in each factor and for every i no more than one of $\{x_i, \bar{x}_i\}$ is present (otherwise the factor is identically 1). The product of factors is called a conjunctive normal form (CNF) of the function to which it equals. A CNF expression in which every factor has no more than three basic functions is called a 3-CNF expression. W.l.o.g we may assume that in a 3-CNF expression each factor has exactly 3 basic functions (if the need arises, one may always write a certain basic function twice since $x+x=x$ for every Boolean expression x.) It is well known that every Boolean function has a CNF representation and even a 3-CNF one.

The SATISFIABILITY PROBLEM is the following: given a Boolean function f of s variables in a 3-CNF, is there a point $(x_1,\ldots,x_s)$ in its domain for which f assumes the value 1 ? (That is, can the expression be satisfied? Or, equivalently, is it false that f is identically zero?)

The satisfiability problem is known to be NPC. Thus, to prove that a certain problem is NPC it suffices to show that it at least as hard as the satisfiability problem, in the following sense: one should provide an algorithm which translates every input of the satisfiability problem to an input of the problem under consideration, such that: (i) The algorithm completes the translation within a number of operations which is bounded by a polynomial in the size of the satisfiability problem's input; and (ii) The answer to the new problem for the input constructed by the algorithm is "YES" iff that is the answer to the satisfiability problem with its original input (Of course, this is a general procedure for problem reduction and, indeed, we will later on use the same procedure for reducing other problems which are known to be NPC to new problems, thus proving the latter are also NPC.)

Let us define $\hat{G}$ to be the two-players game where each player has a single strategy and the payoff to both of the players is 7. (This seemingly idiosyncratic choice is, naturally, rather arbitrary, and will be clarified in the course of the proof.)

We may finally write:

<u>Lemma</u>: There exists a polynomial algorithm which, given a 3-CNF expression f constructs a two-player game G(f) such that G(f) is $\geq$ -reducible to $\hat{G}$ iff G(f) is > -reducible to $\hat{G}$ iff f is satisfiable.

<u>Proof</u>: Let there be given a 3-CNF expression f. We will describe the game G(f) and it will (hopefully) be clear from this description that it can be constructed in a polynomial time (as a function of the size of f's

description.)

Assume that $f = f_1 * f_2 * \ldots * f_m$ where f_j is a factor with exactly three basic functions in the variables $x_1, \ldots, x_s$. For each j ($1 \le j \le m$) let i_{1j}, i_{2j} and i_{3j} be the indices of the variables appearing in f_j. I.e., $1 \le i_{1j} \le i_{2j} \le i_{3j} \le s$ and x_i or $\bar{x}_i$ appear in f_j iff $i \in \{i_{1j}, i_{2j}, i_{3j}\}$. Next let y_{tj} in $\{0,1\}$ ($t=1,2,3$) denote which of the two basic functions, x_i or $\bar{x}_i$, is in f_j. (Which is a well defined question since we assumed that both of them cannot appear in the same factor.) Suppose, then, that y_{tj} equals 1 iff $x_{i_{tj}}$ (rather than $\bar{x}_{i_{tj}}$) appears in f_j.

In G(f) player 1 will have 2s + 3m + 1 strategies and player 2 will have 3s + 4m + 1 strategies. According to our definition of a game, the strategy sets are thus uniquely defined, but it will prove convenient to provide them with nicknames. Thus we wish to call the first 2s strategies of player 1 by the names $(x_1, \bar{x}_1, \ldots, x_s, \bar{x}_s)$ (henceforth - "the set X"), the next 3m strategies - $(f_1', f_1'', f_1''', \ldots, f_m', f_m'', f_m''')$ ("the set F'") and the last one - B. In a similar way we will use nicknames for player 2's strategies. (The same nickname may refer to different numbers for the two players. Hopefully, this will avoid confusion rather than generate it.) The first 3s strategies of player 2 will be called $(a_1, x_1, \bar{x}_1, \ldots, a_s, x_s, \bar{x}_s)$ ("the set AX"). The next m strategies will be named $(f_1, \ldots, f_m)$ ("the set F"), and the next 3m - $(f_1', f_1'', f_1''', \ldots, f_m', f_m'', f_m''')$ ("the set F'"). Player 2's last strategy will be called R.

We are now about to define the payoff functions h^1 and h^2. Let us agree that payoffs which we do not specify are zero. To simplify matters we will describe the payoff by submatrices (at any rate, the reader is encouraged to consult figure 1):

Insert Figure 1 About Here

X x AX : for each $i \le s$, let

$h^1(x_i, a_i) = h^2(x_i, a_i) = 5$

$h^1(\bar{x}_i, a_i) = h^2(\bar{x}_i, a_i) = 5$

$h^1(x_i, x_i) = h^2(x_i, x_i) = 1$

$h^1(x_i, \bar{x}_i) = h^2(x_i, \bar{x}_i) = -1$

$h^1(\bar{x}_i, \bar{x}_i) = h^2(\bar{x}_i, \bar{x}_i) = 1$

$h^1(\bar{x}_i, x_i) = h^2(\bar{x}_i, x_i) = -1$

F' x AX : -- (Both h^1 and h^2 are identically zero in this submatrix.)

{B} x AX : $h_1 = 6$ (i.e., for all $l \in AX$ $h^1(B,l) = 6$.)

X x F : For $i \le s$, $j \le m$ and $t \le 3$,

if $i_{tj} = i$, then if $y_{tj} = 1$,

then $h^2(x_i, f_j) = 4$;

else (that is, $y_{tj} = 0$),

$h^2(\bar{x}_i, f_j) = 4$.

(Otherwise, i.e., if neither x_i nor $\bar{x}_i$ appears in f_j, h^2 is zero.)

F' x F : For $j \le m$, $h^2(f_j', f_j) = h^2(f_j'', f_j) = h^2(f_j''', f_j) = 2$

{B} x F : $h^1 = -1$

X x F' : For $i \le s$ and $j \le m$,

if $i = i_{tj}$ for some t in {1,2,3}

then if $i \ne i_{1j}$ then $h^2(x_i, f_j') = h^2(\bar{x}_i, f_j') = 4$

and if $i \ne i_{2j}$ then $h^2(x_i, f_j'') = h^2(\bar{x}_i, f_j'') = 4$

and if $i \ne i_{3j}$ then $h^2(x_i, f_j''') = h^2(\bar{x}_i, f_j''') = 4$

$F' \times F'$: For $j \le m$,

$$
\begin{aligned}
h^1(f_j', f_j') &= h^2(f_j', f_j') &= 3 \\
h^1(f_j', f_j'') &= h^2(f_j', f_j'') &= 2 \\
h^1(f_j', f_j''') &= h^2(f_j', f_j''') &= 2 \\
h^1(f_j'', f_j') &= h^2(f_j'', f_j') &= 2 \\
h^1(f_j'', f_j'') &= h^2(f_j'', f_j'') &= 3 \\
h^1(f_j'', f_j''') &= h^2(f_j'', f_j''') &= 2 \\
h^1(f_j''', f_j') &= h^2(f_j''', f_j') &= 2 \\
h^1(f_j''', f_j'') &= h^2(f_j''', f_j'') &= 2 \\
h^1(f_j''', f_j''') &= h^2(f_j''', f_j''') &= 3
\end{aligned}
$$

$\{B\} \times F'$: $h^1 = 5$

$X \times \{R\}$: --

$F' \times \{R\}$: --

$B \times \{R\}$: $h^1(B,R) = h^2(B,R) = 7$.

We claim that $G(f)$ is reducible (using either $\geq$ or $>$) to $\hat{G}$ iff f is satisfiable. Let us first prove the "if" part. It suffices to prove this part for the relation $>$, since any legitimate $>$-reduction is also a legitimate $\geq$ -reduction. Suppose, then, that f is satisfiable. Let $\{b_i\}$ in $\{0,1\}$ ($i \le s$) be the values of x_i for which $f(b_1,\ldots,b_s) = 1$. We shall now describe elimination of strategies from G:

1. For each $i \le s$, if $b_i = 1$ let player 2 eliminate the strategy x_i, which is dominated by a_i. If, on the other hand, $b_i = 0$, let player 2 delete the strategy $\bar{x}_i$, which is also dominated by a_i.

2. For each $i \le s$, let player 1 eliminate the strategy corresponding to the one eliminated by player 2. Note that in the absence of player 2's strategy x_i, player 1's strategy x_i is dominated by $\bar{x}_i$ and vice versa.

3. For each $j \leq m$, consider player 2's strategy f_j. Since the values $\{b_i\}$ satisfy the function f, in particular they also satisfy f_j, which means that (at least) one of the basic functions in f_j equals 1. Suppose it is the function with the minimal index. This means that the first row in which $h^2 = 4$ (in this column) was eliminated by player 1, and now player 2's strategy f_j' dominates f_j. Similarly, if it were the second row - f_j'' would now dominate f_j and the same argument applies to the last case, in which it is the third row.

4. By now all player 2's strategies in the set F are eliminated. Hence player 1's strategy B dominates all others. Let, then, player 1 eliminate all other strategies and be left with B.

5. Finally, when only B is left for player 1, the strategy R dominates all other strategies of player 2 and we are left with $\hat{G}$.

We now wish to show the converse, i.e., that if G(f) is >-reducible or $\geq$ -reducible to $\hat{G}$ then the function f is satisfiable. This time it would suffice to prove the implication for $\geq$ -reducibility. Assume, then that there is a legitimate sequence of eliminations of $\geq$ -dominated strategies which results in $\hat{G}$. Let us consider the last time player 1 has eliminated a strategy. The eliminated strategy must have been dominated by B, which is the only one left. But this is possible only if at this point all player 2's strategies in F had already been eliminated. Let us go back even further and consider the point at which the last of player 2's strategies was still in the game. At this point player 1's strategy B cannot dominate any other strategy, nor could it dominate any prior to this stage. We can therefore conclude that at this stage (when the last of player 2's strategies in F is about to be eliminated) player 1 still has all strategies in F', hence so does player 2. Moreover, since player 1's strategy B cannot dominate any other strategy, out of each pair of player 1's strategies in X (i.e., x_i and $\bar{x}_i$) at least one is still in the game. Let us now define the values $\{b_i\}$ which will satisfy the function f: If at this stage in the iterative elimination of strategies player 1 has only the strategy x_i - define $b_i = 0$; If s/he only has $\bar{x}_i$ - define $b_i = 1$; If both are present the definition of b_i is arbitrary, say 1.

We claim that $f(b_1,...,b_s) = 1$. All we have to show is that for each $j \leq m$ the factor f_j is satisfied. We know that at the critical stage when the last of all the f_j's was about to be deleted by player 2, all of them have been either already eliminated or dominated by other strategies of player 2. We also reached the conclusion that at this point all of player 1's strategies in F' were present, hence player 2's strategies in AX could not have dominated f_j. Moreover, because of the structure of the submatrix F' × F', f_j could be dominated only by one of $\{f_j', f_j'', f_j'''\}$ (which were all present at this stage.) Let us assume that f_j' dominated f_j. This implies that the first row in which the column of f_j has $h^2 = 4$ has been deleted earlier. But this means that f_j is satisfied by $(b_1,...,b_s)$ which were defined by the rows deleted by player 1 up to this stage. A similar argument applies to f_j'' and f_j''', and this completes the proof of the lemma. //

To see that facts 1-12 are indeed proved one needs only notice that the question whether G(f) is reducible to $\hat{G}$ may be presented as any one of the twelve problems: it may be presented as a SIZE problem, asking whether G(f) can be reduced to a game of size 1 × 1 (where there is only one candidate for a reduced game of this size); this question is itself a SUBGAME question; and finally, one may ask whether G(f) can be reduced to {B} × {R}, thus disguising it as a SUBSETS problem. Furthermore, in all three problems the game $\hat{G}$ is irreducible, hence the lemma also proves that the "tagged" problems are NPC.

4.2 Proof of Fact 13

To prove that SIZE(>>) is NPC we will prove that the following problem, which is known to be NPC, is reducible to it:

SET COVER PROBLEM: Given a set $S=\{1,2,...,s\}$, m subsets of S, $S_1,...S_m$ and a number $k \le m$, is there a cover of S consisting of no more than k subsets out of $\{S_1,...,S_m\}$? (I.e., are there indices $j_1,...,j_k$ such that $\cup_{1 \le l \le k} S_{j_l} = S$?)

Given input data for the set cover problem, let us construct a game $G(S,S_1,...,S_m)$ as follows: first, for each $i \in S$, let $M_i=\{j|\ 1\le j\le m$ and $i \in S_j\}$. Define $m_i = |M_i|$, and let M be $\sum_{i=1}^{m} m_i$. $G(S,S_1,...,S_m)$ is a two-player game in which player 1 has s+M strategies and player 2 has s+m+1 strategies. Let us nickname player 1's strategies as follows: the first s strategies will be called $x_1,...,x_s$ and we will refer to them together as S. The next m_1 strategies will be named $\{x_{1j}\}$ for j in M_1, and X_1 as a set. The next m_2 strategies will be $\{x_{2j}\}$ for j in M_2, and X_2 as a set, and so forth. The symbol $X_{i's}$ will refer to the union of $X_1,...,X_s$. Player 2's strategies will be named as follows: the first s strategies, like those of player 1, are $x_1,...,x_s$, and together they will be referred to as S. The next m strategies will be called $S_1,S_2,...,S_m$, and $S_{j's}$ as a set. The last one will be called R.

We will now define the payoff functions. As in the previous proof, we will not mention zero values. As above, we describe each submatrix separately for clarity of exposition, though figure 2 will probably be more helpful:

Insert Figure 2 About Here

$S \times S$: For every $i,k \in S$,

if $i=k$, then $h^1(x_i,x_k) = h^2(x_i,x_k) = 4$

else $(i \neq k)$ $h^1(x_i,x_k) = h^2(x_i,x_k) = -1$

$X_{i's} \times S$: For every $i \in S$ and $j \in M_i$,

$h^1(x_{ij},x_i) = h^2(x_{ij},x_i) = 5$

$S \times S_{j's}$: For every $i \in S$ and $j \leq m$,

if $i \in S_j$, then $h^1(x_i,S_j) = 2$

(otherwise $h^1(x_i,S_j) = 0$.)

$X_{i's} \times S_{j's}$: For every $i \in S$ and $j \in M_i$ and for every $l \leq m$,

If $i \in S_l$ <u>and</u> $j \neq l$, then $h^1(x_{ij},S_l) = 3$

otherwise $(i \notin S_l$ or $j=l)$, $h^1(x_{ij},S_l) = 1$

$S \times \{R\}$: For all $i \in S$, $h^2(x_i,R) = 3$

$X_{i's} \times \{R\}$: For all $i \in S$ and $j \in M_i$, $h^1(x_{ij},R) = 1$,

$h^2(x_{ij},R) = 3$.

To show that SIZE(>>) is NPC it suffices to prove the following:

<u>Claim</u>: For every input to the set cover problem $(S,S_1,\ldots,S_m,k)$, the game

$G(S,S_1,\ldots,S_m)$ is >>-reducible to a game in which player 1 has M strategies and player 2 has $(m - k + n + 1)$ iff there is a cover of S involving no more than k sets out of $\{S_1,\ldots,S_m\}$.

<u>Proof</u>: Let us first note the following facts about the game $G=G(S,S_1,\ldots,S_m)$: for player 2, the strategy R >>-dominates all the strategies in $S_{j's}$, while there are no other dominations. For player 1 there are no dominations. Furthermore, none of player 1's strategies in $X_{i's}$ nor any of player 2's strategies in S could ever (i.e., in any reduced game of G) be dominated by any other strategy (since both h^1 and h^2 attain their maximal payoff on the entries (x_{ij},x_i) for $i \in S$ and $j \in M_i$.) Hence we also conclude that there could be no dominations within player 1's set of strategies S. (In the eternal presence of the S strategies of player 2.) Similarly, since none of player 1's strategies in $X_{i's}$ can ever be eliminated, player 2's strategy R will never be dominated by any other. Taking all these considerations into account we conclude that any reduced game $\hat{G}$ is obtained from G by elimination of player 2's strategies in $S_{j's}$ (dominated by R) and elimination of player 1's strategies in S (dominated by corresponding strategies in $X_{i's}$). Moreover, since R dominates all players 2's strategies in $S_{j's}$ from the very beginning, we could assume w.l.o.g. that first player 2 deletes all strategies not appearing in $\hat{G}$, and then player 1 deletes his.

Indeed, player 2 may delete <u>all</u> of $S_{j's}$ and then player 1 can delete <u>all</u> of S. The (unique) maximal reduction is $X_{i's} \times (S \cup \{R\})$. In this maximal reduction player 1 indeed has exactly M strategies. However, player 2 has only s+1 strategies. The question of whether G can be reduced to a game of size $M^*(m - k + s + 1)$ is therefore equivalent to the question: Are there k strategies of player 2 in $S_{j's}$ such that it suffices that they be deleted by player 2 so that player 1 will be able to eliminate all his strategies in S? It is readily observed that this is so if and only if there are k (or less than k) subsets in $\{S_1,\ldots,S_m\}$ which cover S.

To conclude the proof we only mention that the construction of the game $G(S,S_1,\ldots,S_m)$ can be done in a polynomial time. //

4.3 Proof of Facts 14,16,18

We note that for a given game G, an >>-irreducible >>-reduction of it can be computed in a polynomial time. Indeed, the straightforward algorithm which compares every pair of strategies, eliminates the first dominated one and continues with the reduced game must end up with an >>-irreducible subgame after at most $|S|^4$ operations. (Recall that S was defined to be $\Pi_{i\in N} S^i$, and the size of the input data is at least $|S|$.) Furthermore, if we count an "oracle" invocation as a single operation, this trivial upper bound reduces to $|S|^3$. (Note that this is true for any domination relation dom.) Using the uniqueness result of Gilboa-Kalai-Zemel (1989) it is obvious that SIZE'(>>), SUBGAME'(>>) and SUBSETS'(>>) can be solved in polynomial time.

4.4 Proof of Fact 15

We now wish to show that SUBGAME(>>) is NPC. This time we use the following problem, which is also known to be NPC:

<u>THE CLIQUE PROBLEM</u>: Given an undirected graph $Gr(V,E)$ and a positive integer k, is there a clique of size $k \in Gr$? (I.e., is there a subset V' of V such that $|V'|=k$ and for all $i,j \in V'$, $i \neq j$, $\{i,j\} \in E$?)

Let there be given a graph $Gr(V,E)$ and a number k. W.l.o.g. assume that $V=\{1,2,\ldots,v\}$ (with $v \geq 1$). Define a game $G(Gr)$ as follows: $G(Gr)$ is a two-player game where each of the players has a strategy set $\{1,2,\ldots,v,v+1\}$. Define the payoff functions by:

For $i,j \leq v$

$$h^1(i,j) = h^2(i,j) = \begin{cases} 2 & \text{if } i=j \\ 1 & \text{if } i \neq j \text{ and } \{i,j\} \in E \\ 0 & \text{if } i \neq j \text{ and } \{i,j\} \notin E \end{cases}$$

For $i \leq v$

$$h^1(i,v+1) = h^2(i,v+1) = 3$$

For $j \leq v$

$$h^1(v+1,j) = h^2(v+1,j) = 3$$

and $$h^1(v+1,v+1) = h^2(v+1,v+1) = 4 .$$

Next define $\hat{G}(Gr)$ to be a two-person game where each player has $k+1$ strategies and the payoff functions are

For $i,j \leq k$

$$h'^1(i,j) = h'^2(i,j) = \begin{cases} 2 & \text{if } i=j \\ 1 & \text{if } i \neq j \end{cases}$$

For $i \leq k$

$$h'^1(i,k+1) = h'^2(i,k+1) = 3$$

For $j \leq k$

$$h'^1(k+1,j) = h'^2(k+1,j) = 3$$

and $h'^1(k+1,k+1) = h'^2(k+1,k+1) = 4$.

Observe that in G(Gr), for both players the strategy v+1 >>-dominates all others, but there are no other dominations. Furthermore, whatever dominated strategies are eliminated, no new dominations can be generated. Hence G(Gr) can be reduced to $\hat{G}(Gr)$ iff the latter is a subgame of the former, and that is true iff Gr has a clique of size k. As G(Gr) and $\hat{G}(Gr)$ can be constructed in polynomial time, the proof is complete. //

4.5 Proof of Fact 17

We now have to prove that SUBSETS(>>) is polynomial. Suppose, then, that we are given a game $G = (N,(S^i)_{i\in N},(h^i)_{i\in N})$ and a subset T^i of S^i for each $i \in N$. Define a new domination relation dom on G, as follows: dom is identical to >> when the strategies in T^i are not involved or when they dominate other strategies. However, in all subgames of G (including G itself) no strategy in T^i is dominated by any other.

It is easy to see that dom is a hereditary partial order and that it can be computed in polynomial time. We therefore conclude (again, by Gilboa-Kalai-Zemel (1989)) that G has a unique maximal dom-reduction, and this reduction can be computed in a polynomial time as in 4.3 above.

It only remains to note that G(T) is a reduction of G iff G(T) is the maximal dom-reduction of G, which completes the proof. //

5. Concluding Remarks

The introduction of computational complexity considerations entails some complications which are generally avoided in game theory. In particular, the question of the game representation becomes much more important than it generally is. For instance, two normal-form games which are equivalent up to a renaming of the strategies cannot be assumed to be equivalent for our purposes. Indeed, it may be very hard to determine whether two given games are equivalent in this sense. (Obviously, this problem is at least as hard as the graph equivalence problem, and it is not known whether the latter is polynomial, NPC, neither or both.) For this reason we did not assume that a player's strategies are given as a set; rather, they constitute a sequence, with a well defined linear order. We wish to stress this point which means that those problems which were proved difficult in the sequel have some intrinsic difficulty, unrelated to the complexity of the game-equivalence problem; On the other hand, problems which were proved to be "easy" are easy exactly as stated, and may become difficult if the game data is presented to us in a different form.

Another important presentational question is whether the game is given in its normal or extensive form. We recall that the normal form of a given game can be "easily" (i.e., in polynomial time) translated into an extensive form, but the converse is false. For simplicity, we choose to deal with the normal form and note that those problems which are "difficult" (i.e., NPH or NPC) even with respect to this "wasteful" representation will also be difficult in the extensive form. On the other hand, problems which are proved to be polynomial as a function of the normal form input size are not necessarily polynomial in the size of the more succinct extensive form. We therefore note that regarding both presentational issues our conclusions are similar: those problems which are proved to be NPC may indeed be considered difficult; however, one should be cautious when interpreting the results concerning "easy" problems.

References

Ben-Porath, E. (1988), "The Complexity of Computing Best Response Automata in Repeated Games with Mixed Strategies," forthcoming, Games and Economic Behavior.

Bernheim, B. D. (1984), "Rationalizable Strategic Behavior," Econometrica, 52, 1007-1025.

Binmore, K. (1987), "Modeling Rational Players" I, J. Econ. and Phil., 3, 179-214.

Binmore, K. (1988), "Modeling Rational Players" II, J. Econ. and Phil., 4, 9-55.

Gilboa, I. (1988), "The Complexity of Computing Best-Response Automata in Repeated Games," J. of Econ. Theory, 45, 342-352.

Gilboa, I., E. Kalai and E. Zemel (1989), "On the Order of Eliminating Dominated Strategies," Northwestern University Discussion Paper.

Gilboa, I. and E. Zemel (1989), "Nash and Correlated Equilibria: Some Complexity Considerations," Games and Economic Behavior, 1, 80-93.

Knuth, D. E., C. H. Papadimitriou and J. N. Tsitsiklis (1988), "A Note on Strategy Elimination in Bimatrix Games," O. R. Letters, 7, 103-108.

Papadimitriou, C. H. (1989), "On Players with a Bounded Number of States," manuscript, University of California, San Diego.

Pearce, D. (1984), "Rationalizable Strategic Behavior and the Problem of Perfection," Econometrica, 52, 1029-1050.

Sahni, S. (1974), "Computationally Related Problems," SIAM J. on Computing, 3, 262-279.

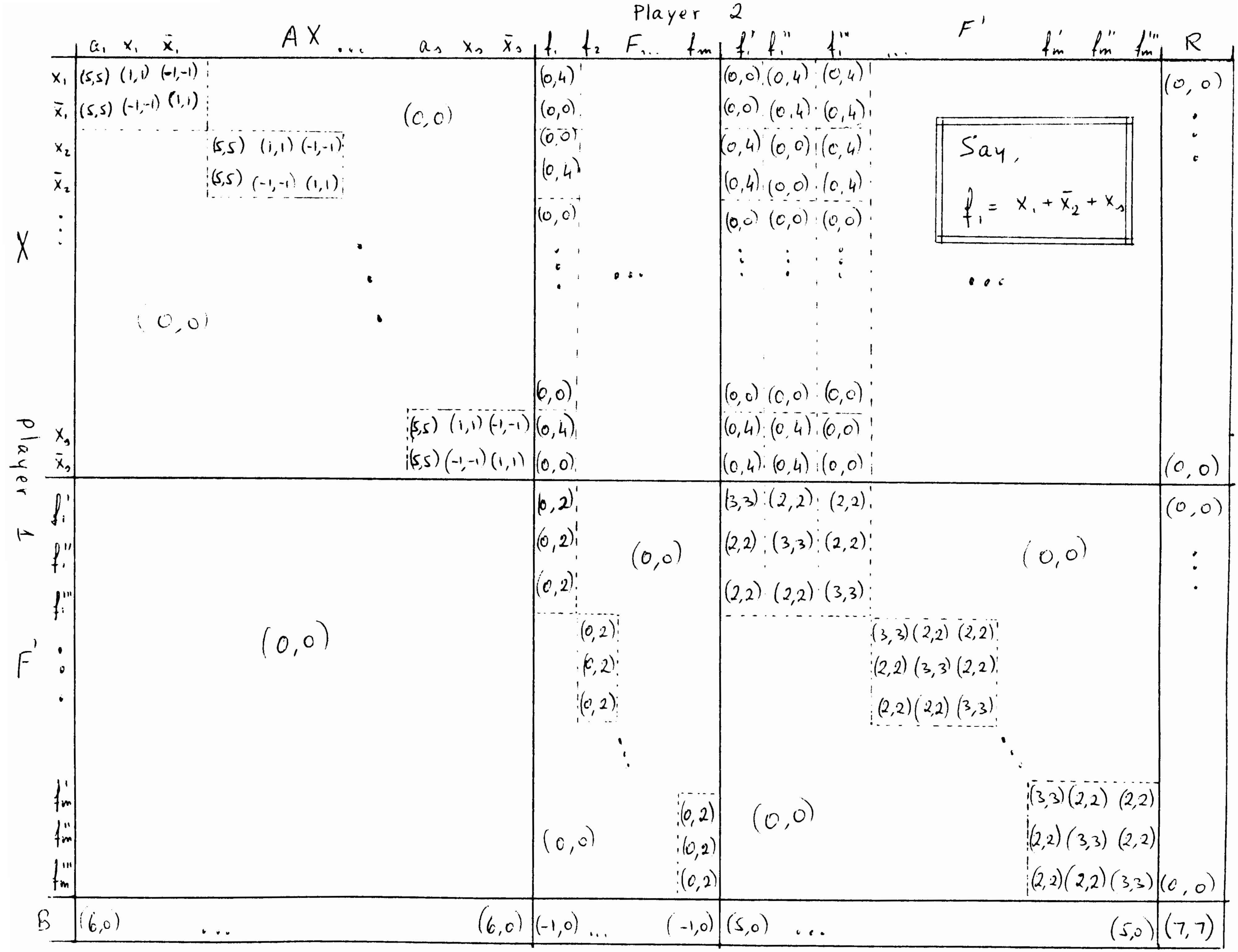
Player 2
Player 1
AX
F
F'
X
R
B
a_1 x_1 $\bar{x}_1$ … a_s x_s $\bar{x}_s$
f_1 f_2 … f_m
f_1' f_1'' f_1''' … f_m' f_m'' f_m'''
x_1 $\bar{x}_1$ x_2 $\bar{x}_2$ … x_s $\bar{x}_s$
(5,5) (1,1) (-1,-1)
(5,5) (-1,-1) (1,1)
(0,0)
(0,4)
(0,2)
(3,3) (2,2) (2,2)
(2,2) (3,3) (2,2)
(2,2) (2,2) (3,3)
(6,0)
(-1,0)
(5,0)
(7,7)
Say,
$f_1 = x_1 + \bar{x}_2 + x_s$

Figure 1

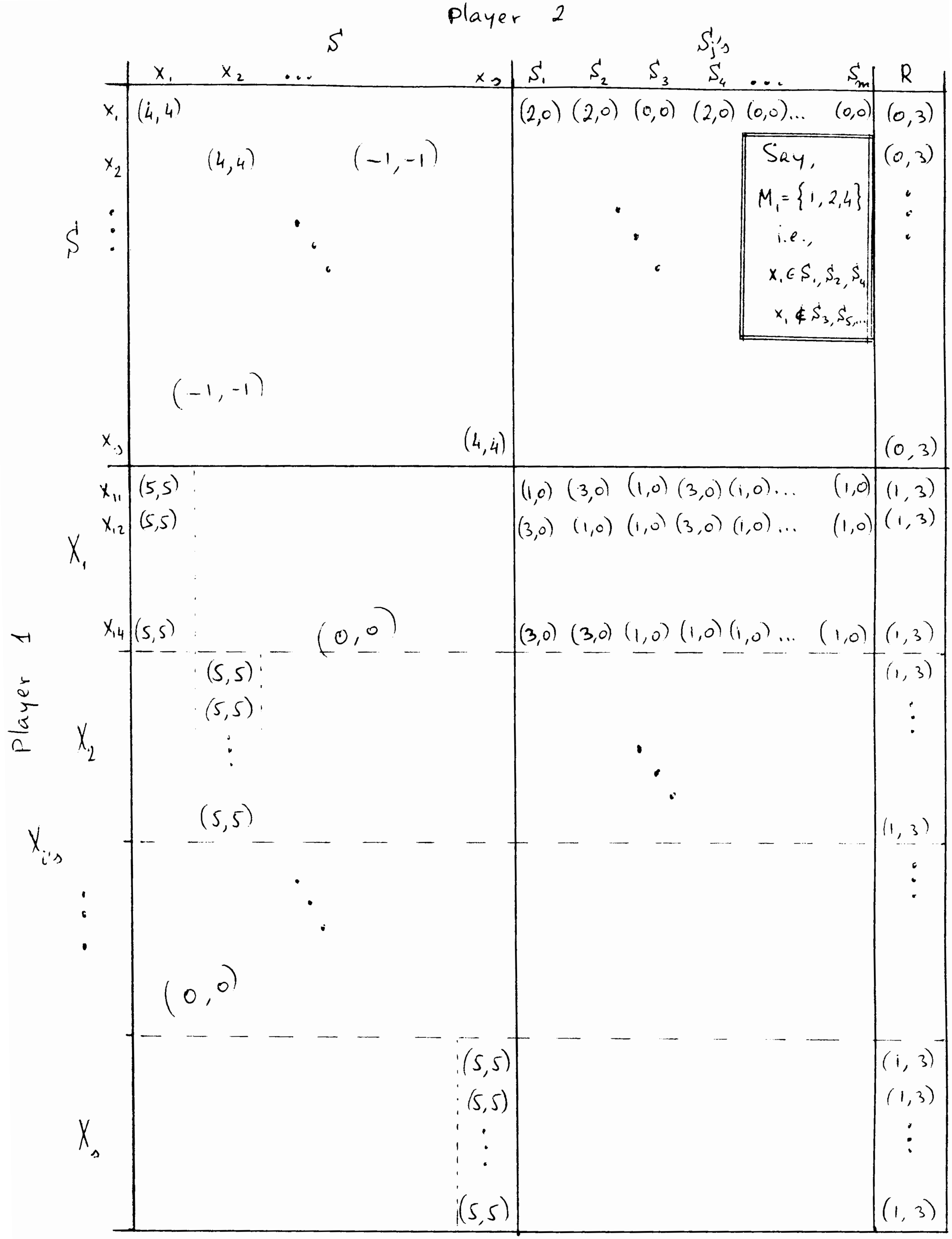
Player 2
S
S_j's
x1 x2 ... xs
S1 S2 S3 S4 ... Sm
R
Player 1
S
X1
X2
Xi's
Xs
x11 x12 x14
(4,4)
(-1,-1)
(2,0) (2,0) (0,0) (2,0) (0,0)... (0,0)
(0,3)
Say,
M1 = {1, 2, 4}
i.e.,
x1 ∈ S1, S2, S4
x1 ∉ S3, S5,...
(5,5)
(0,0)
(1,0) (3,0) (1,0) (3,0) (1,0)... (1,0)
(3,0) (1,0) (1,0) (3,0) (1,0)... (1,0)
(3,0) (3,0) (1,0) (1,0) (1,0)... (1,0)
(1,3)

Figure 2