GAMES AND ECONOMIC BEHAVIOR 1, 144–169 (1989)

On Computable Beliefs of Rational Machines

NIMROD MEGIDDO

IBM Research Division, Almaden Research Center, San Jose, California 95120-6099, and School of Mathematical Sciences, Tel Aviv University, Tel Aviv, Israel

Traditional decision theory has assumed that agents have complete, consistent, and readily available beliefs and preferences. Obviously, even if an expert system has complete and consistent beliefs, it cannot have them readily available. Moreover, some beliefs about beliefs are not even approximately computable. It is shown that if all players have complete and consistent beliefs, they can compute approximate beliefs about beliefs of any order by considering events arbitrarily close in some well-defined sense to those in question.

1. INTRODUCTION

In traditional decision sciences (see, for example, Luce and Raiffa, 1957) decision makers are usually not assumed to be restricted in their thinking in any way. They have consistent beliefs and preferences which are available throughout the decision making process. The widely accepted Bayesian approach to decision making under uncertainty maintains that whenever an agent lacks information about the value of a certain variable, s/he still has some "subjective" probability distribution (i.e., beliefs) with respect to such values. The sense of the word "has" in the preceding sentence is that all the probabilities are readily available. Of course, the beliefs are subject to Bayesian updating whenever some new information is received.

There has recently been interest in modeling players as computing machines (see, for example, Binmore, 1987). If the decision maker is a computer program (an "expert system"), rather than an ideal player as in the traditional theory, its beliefs are not readily available. The program may have consistent beliefs which it can only approximate with arbitrary precision. Moreover, for some events with complicated descriptions, the

0899-8256/89 $3.00

beliefs may be determined by the basic beliefs but the program cannot even approximate them. It should be noted that in this paper we do not deal with the question of computational complexity at all but rather with the more basic notion of computability. Thus, we are interested here in what expert systems can do in principle and not necessarily in practice.

It is useful to consider beliefs as computable and noncomputable real numbers. A real number a is said to be computable if there exists a computer program A such that, given any rational $\varepsilon > 0$, A outputs a rational number $\tilde{a}(\varepsilon)$ such that $|\tilde{a}(\varepsilon) - a| < \varepsilon$. In this sense the program A "knows" the real number a. However, the program can only tell us approximations to a. We believe that a "rational" program should have consistent beliefs in this asymptotic sense, namely, its exact beliefs should be consistent even though the program can only work with approximate beliefs.

Game theory is concerned with situations where more than one decision maker is involved. Players must reason about one another's behavior. In a game of incomplete information, players do not know exactly who the other players are; i.e., they do not know exactly what the other players know about other players. However, Bayesian players have beliefs (in the form of probability distributions) about other players. The foundations for a theory of games with incomplete information played by Bayesian players were given in Harsanyi (1967–1968) (see also Mertens and Zamir, 1985). However, the questions of computability have not been addressed in this context.

In this paper players assume the form of programs residing in computers. We are concerned with the issue of beliefs of programs about other programs and their beliefs. Beliefs about the state of "nature" (as well as beliefs about beliefs about the state of nature, and so on) are suppressed for simplicity of presentation. In other words, states of the world (or "possible worlds") correspond here to combinations of beliefs of players about players. Our discussion here should be considered an extension to the foundations of games with incomplete information played by Bayesian players. Beliefs about beliefs have also been considered by philosophers, mathematicians, and computer scientists. Some references on beliefs can be found in Gaifman (1986), Fagin *et al.* (1988), Fagin and Halpern (1988), and Konolige (1986).

2. On Levels of Belief

For the benefit of readers who have not been exposed to the issues of beliefs, we first demonstrate the complications involved in beliefs of players about each other. To simplify the discussion consider henceforth only

2-person games. The extension to any finite number of players is straightforward.

Suppose none of the players knows exactly who his/her opponent is. For example, suppose none of the players knows whether his/her opponent is a male or a female. However, each player has some belief about the sex of his/her opponent. Let P_i ($i = 1, 2$) denote the probability with which player i believes his/her opponent is a male. Suppose player $j = 3 - i$ does not know the precise value of P_i. Thus, s/he considers P_i to be a random variable with some probability distribution $F_j^{(1)}$. Here the superscript (1) indicates that this is a belief of level 1. Similarly, player i has a probability distribution $F_i^{(1)}$ with respect to P_j which s/he views as a random variable.

If the players are not restricted in any way then $F_i^{(1)}$ and $F_j^{(1)}$ may already be quite complicated mathematical objects. Note that, in general, the player may be concerned not only with the question of whether the opponent is a male or a female but also with the question of what the opponent believes about the sex of the player him/herself. In principle, each player should have a probability measure on the space of possible opponents. In particular, this space must have a measurable structure relevant to the game. Thus, this structure should reflect not only the sex of the opponent but also the opponent's beliefs about the first player's sex, the beliefs of the second player about the beliefs of the first player about the sex of the second player, and so on. In general, this already raises the need to consider infinite spaces of possible opponents. More specifically, already at the first level player 1, say, must characterize possible opponents not just according to being M (male) or F (female) but also according to types (M, P_2) or (F, P_2) where P_2 (which may be any number between 0 and 1) is the probability which player 2 ascribes to the event that player 1 is a male. Of course, the space of possible opponents must be considered together with a *measurable* structure.

At the next level, player i does not know what $F_j^{(1)}$ is, so s/he has some probability distribution $F_i^{(2)}$ with respect to it. (The superscript (2) indicates a second level of beliefs.) This is a distribution on a class of possible probability distributions of a single random variable. In general, the complication of the possible distributions grows quickly with the level of belief. Special care must be given to the problem of measurability. We sometimes talk about levels of *events* which are the objects of belief. Essentially, the level of an event is the number of times we include a reference to a player in the definition of the event.

We consider below the restricted case where players are identified with finite programs. The set of all possible programs is of course enumerable. This implies severe restrictions on the type of beliefs of players about each other.

3. An Overview

Every program has a finite size, yet it reacts to an infinite number of possible inputs. In other words, the behavior of the program in an infinite number of situations is described (implicitly) using finite space. A similar observation applies to the "beliefs" of the program about an infinite number of events. Since the program is finite, it cannot have all its beliefs readily available. Thus it may have to compute some of its beliefs during the decision making process. Of course, the computation is invoked by some signal from the outside, and there are infinitely many possible signals.

The players in our model are programs residing in computers. Recall that we restrict attention to games with two players. Denote by M^1, M^2, . . . the sequence of all possible programs. These programs do not have to exist in the physical sense of the word. They are merely strings of characters. It is easy to construct a one-to-one mapping from programs to natural numbers. Gödel constructed such a mapping (for a different purpose), so it is quite common to talk about the "Gödel number" of a program (or, equivalently, a Turing machine). The details of the mapping are not relevant. However, the important property is that there exists an effective procedure for translating numbers into programs and vice versa.

In our model there are two computers C_1, C_2. In the beginning these computers are empty (like the "empty shells" in Aumann, 1985). They are then loaded with programs X_1, X_2, respectively. The symbols X_1, X_2 should be interpreted as random variables whose values are names of programs, or Gödel numbers. The latter interpretation is appealing since it makes X_1 and X_2 random variables in the usual sense. Note that we allow for $X_1 = X_2$ since there is no limit on the number of copies of the same program which may be involved in a game.

We do not impose any restriction on the "thinking power" of our players beyond the fact that they are finite programs. We will always assume they are sufficiently smart to compute whatever is needed and computable. Thus, we are aiming at a definition of a class **S** of those programs which qualify as "smart." In particular, smart programs have complete beliefs about their opponents. If the class **S** is finite then questions about computability become trivial, so we assume **S** is infinite. Note that for every program there are infinitely many programs that are equivalent to it in the sense that they react in the same way to any input.

The measurable space underlying our discussion is therefore as follows. The points of the space are pairs (M^i, M^j) of programs where M^i and M^j are members of a certain subset **S** of the set of all possible programs. Since the space is enumerable there is no problem in assuming that *all* the subsets of $\mathbf{S} \times \mathbf{S}$ are measurable. Each program in **S** has well-defined

beliefs, so a pair (M^i, M^j) entails a complete description of the state of the world.

4. An Example

To explain what we mean by computation of beliefs, consider a simple example where $\mathbf{S} = \{M^1, M^2\}$ and, furthermore, suppose M^1 and M^2 have the same prior beliefs about the pair (X_1, X_2). Thus, each of them contains a certain joint probability distribution for the random variables X_1, X_2. Since each of the variables has two possible values, M^1 and M^2, this distribution is given by four numbers $p_{11}, p_{12}, p_{21}, p_{22} \geq 0$ such that $\Sigma p_{ij} = 1$, where p_{ij} is the probability of the event $\{X_1 = M^i\} \cap \{X_2 = M^j\}$. It is not difficult to see what ought to be the beliefs of such programs. For example, consider the query: "Given you are residing in C_1, what are your beliefs about the program residing in C_2?" It is easy to see that the answer must be a probability of $p_{11}/(p_{11} + p_{12})$ for the event $\{X_2 = M^1\}$ and a probability of $p_{12}/(p_{11} + p_{12})$ for the event $\{X_2 = M^2\}$.

We denote by $F(X)$ the probability distribution which each of the programs has with respect to a random variable X. As we shall see in a moment, the variable X may attain values which are themselves possible probability distribution functions of another random variable. We have already computed $F(X_2/X_1 = M^1)$. The unconditional distribution $F(X_2)$ obviously gives a probability of $p_{11} + p_{21}$ to the event $\{X_2 = M^1\}$ and a probability of $p_{12} + p_{22}$ to the event $\{X_2 = M^2\}$. Another example is $F(X_1|X_2 = M^2)$ which gives $p_{12}/(p_{12} + p_{22})$ to $\{X_1 = M^1\}$ and $p_{22}/(p_{12} + p_{22})$ to $\{X_2 = M^2\}$.

In general, denote by $F_{M^i}(X)$ the distribution which the program M^i has with respect to a random variable X. If we write $F_{X_1}(X)$ we get a random variable whose values are probability distributions, namely, it is the probability distribution which the program residing in the computer C_1 has with respect to the random variable X. For example, $F_{X_1}(X_2)$ is the distribution which X_1 has with respect to X_2, which is computed as follows. With probability $p_{11} + p_{12}$, we have $X_1 = M^1$, in which case the distribution of X_2 gives $p_{11}/(p_{11} + p_{12})$ to $\{X_2 = M^1\}$ and $p_{12}/(p_{12} + p_{22})$ to $\{X_2 = M^2\}$; with probability $p_{21} + p_{22}$, we have $\{X_1 = M^2\}$, in which case the distribution of X_2 gives $p_{21}/(p_{21} + p_{22})$ to $\{X_2 = M^1\}$ and $p_{22}/(p_{21} + p_{22})$ to $\{X_2 = M^2\}$. It is quite obvious to see how higher levels of beliefs of the programs about each other can be extracted from the numbers p_{ij}.

5. The Model

As noted above, we eventually would like to have defined a class $\mathbf{S}$ of programs which would include only programs of a certain degree of so-

phistication. These programs would be considered "rational players." The set **S** would be countable, and we expect it to be infinite.

Our main assumption is that rational players have consistent beliefs. Thus, we assume the following:

A1. Each of the programs in **S** contains an implicit description of a probability distribution over the "states of the world" (or "possible worlds"), i.e., a joint probability distribution of the random variables X_1, X_2, signifying the programs residing in the computers C_1, C_2.

The implicit presence of a distribution is considered one of the axioms that would characterize programs in the class **S.** This distribution is an inherent part of the program. It reflects the program's prior probabilities before it is informed of the computer in which it resides. For any program $M^k \in \mathbf{S}$, denote by p_{ij}^k the probability which M^k ascribes to the event $\{X_1 = M^i\} \cap \{X_2 = M^j\}$. This probability may be viewed as a function of two variables, i, j.

In traditional game theory it is informally assumed to be common knowledge among the players that they are all rational. Accordingly, we assume:

A2. Each program in **S** ascribes probability zero to any event in which any of the computers C_1, C_2 stores a program which is not in **S.**

We do not assume that programs in **S** can decide whether a given program belongs to the class **S.**

The objects of belief are events. Recall that the events are precisely the subsets of $\mathbf{S} \times \mathbf{S}$. Such a subset E represents all instances in which $(X_1, X_2) \in E$. However, events are usually described without specifying the sets E directly. In order for a program to "understand" what the event is, there must be an effective procedure which tells for each pair (i, j) whether, say, $(M^i, M^j) \in E$. In such a case we say that the event E is "computable." Obviously there cannot exist more than $\aleph_0$ computable events.

When an event is described verbally, we can attach to the description a "level number" as follows. First, direct descriptions of the set E will be considered of level 1. On the other hand, a description in the form of a sentence such as "X_1 ascribes probability greater than 50% to the event that X_2 believes with probability greater than 90% that $X_1 = M_{17}$" is to be considered of level 3. Essentially, descriptions of level $\nu + 1$ are stated in terms of beliefs of players about events with descriptions of level less than or equal to ν. Note that the level numbers are associated with descriptions of events rather than the events themselves. An event may have descriptions of different levels.

It is trivial to see that the prior distributions $\{p_{ij}^k\}$ determine the beliefs

of the programs with respect to any event. Obviously, for every S_1, $S_2 \subseteq \mathbf{S}$,

$$p^k(S_1 \times S_2) = \sum_{\substack{i \in S_1 \\ j \in S_2}} p_{ij}^k.$$

The latter may constitute an infinite series, convergent, of course. By cardinality arguments, not all the beliefs are computable.

An interesting question is the relation between the description of an event and the computability of its probability. Consider the following example. Denote by E an even in which player 1, say, believes with probability greater than π_0 that a certain event E' with a description of level ν has occurred. Let p_i denote the probability which M^k ascribes to $\{X_1 = M^i\}$, and let $\pi_i = p^i(E'/X_1 = M^i)$; i.e., π_i is the conditional probability which M^i ascribes to the event E', given that $X_1 = M^i$. Let S_1 denote the set of all indices i such that $\pi_i > \pi_0$. Then, obviously,

$$p^k(E) = \sum_{i \in S_1} p_i.$$

The quantities p_i and π_i are determined by the p_{ij}^k's but there may not exist programs which compute their exact values.

We prefer not to restrict the probabilities p_{ij}^k to be rational numbers. However, since our players are finite programs, we must assume the probabilities are computable. One can distinguish two approaches to computation of beliefs of programs, namely, exact and approximate computation. However, there is a difficulty with the exact computation approach. We might insist that the p_{ij}^k's be rational numbers but that does not imply that numbers of the form $\Sigma_j\, p_{ij}^k p_{hl}^j$, which are typically involved in the computation of beliefs, will also be rational. It seems unjustified to require that the probabilities of all events be rational numbers, so we adopt the approximate computation approach, which means that the program computes its beliefs with any prescribed precision.

More formally, we assume that given i, j and a rational $\varepsilon > 0$, the program M^k computes a nonnegative rational number $\tilde{p}_{ij}^k(\varepsilon)$ such that

$$|\tilde{p}_{ij}^k(\varepsilon) - p_{ij}^k| < \varepsilon.$$

Thus the exact belief p_{ij}^k is the limit (as ε tends to zero) of the approximate beliefs $\tilde{p}_{ij}^k(\varepsilon)$ which M^k computes given the prescribed precision. We of course assume that $\Sigma_{i,j}\, p_{ij}^k = 1$. A stronger assumption, namely, $\Sigma_{i,j}\, \tilde{p}_{ij}^k = 1$, is reasonable yet not necessary.

6. Facts about Computable Numbers

In this section we present some elementary facts about computable numbers.

Definition 6.1. A real number a is said to be *computable* if there is a program A such that, given any rational number $\varepsilon > 0$, A computes a rational number $\tilde{a} = \tilde{a}(\varepsilon)$ such that

$$|\tilde{a}(\varepsilon) - a| < \varepsilon.$$

Proposition 6.2. *The computable real numbers constitute a field.*

Proof. The theme in what follows is that the quality of the required approximation can be computed in advance. First note that if a is computable then so is $-a$. Suppose there exist programs which compute rational ε-approximations $\tilde{a}(\varepsilon)$ and $\tilde{a}(\varepsilon)$ for a and b, respectively, for any rational $\varepsilon > 0$. Thus $|\tilde{a}(\varepsilon) - a| < \varepsilon$ and $|\tilde{b}(\varepsilon) - b| < \varepsilon$. To approximate $a + b$, consider the estimate

$$|(\tilde{a}(\delta) + \tilde{b}(\delta)) - (a + b)| \leq |\tilde{a}(\delta) - a| + |\tilde{b}(\delta) - b| < 2\delta.$$

It implies that a rational ε-approximation for $a + b$ can be computed by adding up $\varepsilon/2$-approximations of a and b. To approximate ab, consider the estimate

$$\begin{aligned} |\tilde{a}(\delta)\tilde{b}(\delta) - ab| &\leq |\tilde{a}(\delta)\tilde{b}(\delta) - \tilde{a}(\delta)b| + |\tilde{a}(\delta)b - ab| \\ &\leq |\tilde{a}(\delta)| \cdot |\tilde{b}(\delta) - b| + |\tilde{a}(\delta) - a| \cdot |b| \\ &\leq (|\tilde{a}(\delta)| + |\tilde{b}(\delta)| + \delta)\delta. \end{aligned}$$

As δ tends to zero, this computable upper bound tends to zero. Thus, an ε-approximation of $a + b$ can be computed by adding up $\tilde{a}(\delta) + \tilde{b}(\delta)$, where $\delta = 1/n$ and n is the first integer such that

$$(|\tilde{a}(\delta)| + |\tilde{b}(\delta)| + \delta)\delta < \varepsilon.$$

Suppose $a \neq 0$. To approximate a^{-1}, assume without loss of generality that for any ε, $\tilde{a}(\varepsilon) \neq 0$ and consider the estimate

$$\left|\frac{1}{\tilde{a}(\delta)} - \frac{1}{a}\right| \leq \frac{|\tilde{a}(\delta) - a|}{|a| \cdot |\tilde{a}(\delta)|}.$$

For δ sufficiently small, since $a \neq 0$ and $\tilde{a}(\delta)$ tends to a, we have

$$\left|\frac{1}{\tilde{a}(\delta)} - \frac{1}{a}\right| \leq \frac{\delta}{|\tilde{a}(\delta)| \cdot (|\tilde{a}(\delta)| - \delta)}$$

which implies that a^{-1} is computable. We have thus shown that the set of the computable numbers is closed under the arithmetic operations. ■

DEFINITION 6.3. Let $a = a(i)$ $(i = 1, 2, \ldots)$ be a function which assigns to every positive integer i, a real number $a(i)$. The function a is called *computable* if there exists a program A which approximates $a(i)$ with arbitrary precision. Specifically, when the program A receives i and any rational $\varepsilon > 0$, it computes a rational number $\tilde{a}(i, \varepsilon)$ such that

$$|\tilde{a}(i, \varepsilon) - a(i)| < \varepsilon.$$

PROPOSITION 6.4. *If $a = a(i)$ and $b = b(i)$ $(i = 1, 2, \ldots)$ are computable nonnegative functions such that*

$$\sum_{i=1}^{\infty} a(i) = \sum_{i=1}^{\infty} b(i) = 1$$

then the number

$$c = \sum_{i=1}^{\infty} a(i)b(i)$$

is computable.

Proof. Suppose A and B are approximation programs for a and b, and let $\tilde{a}(i, \delta)$ and $\tilde{b}(i, \delta)$ denote the approximate values which they compute for $a(i)$ and $b(i)$, respectively, given the requirement δ on the approximation:

$$|\tilde{a}(i, \delta) - a(i)|, |\tilde{b}(i, \delta) - b(i)| < \delta.$$

We sketch an approximation program for the number c. Given any ε, we must compute a number $\tilde{c}(\varepsilon)$ such that

$$|\tilde{c}(\varepsilon) - c| < \varepsilon.$$

Let n denote any positive integer and let $\delta = o(n^{-1})$. Obviously,

$$\left|\sum_{i=1}^{n} \tilde{a}(i, \delta) - \sum_{i=1}^{n} a(i)\right| < \delta n = o(1)$$

and

$$\left|\sum_{i=1}^{n} \tilde{b}(i, \delta) - \sum_{i=1}^{n} b(i)\right| < \delta n = o(1).$$

Moreover,

$$\begin{aligned}
&\left|\sum_{i=1}^{n} \tilde{a}(i, \delta)\tilde{b}(i, \delta) - \sum_{i=1}^{n} a(i)b(i)\right| \\
&\quad \leq \left|\sum_{i=1}^{n} \tilde{a}(i, \delta)b(i) - \sum_{i=1}^{n} a(i)b(i)\right| + \left|\sum_{i=1}^{n} \tilde{a}(i, \delta)b(i) - \sum_{i=1}^{n} \tilde{a}(i, \delta)\tilde{b}(i,\delta)\right| \\
&\quad \leq \delta \sum_{i=1}^{n} b(i) + \delta \sum_{i=1}^{n} \tilde{a}(i, \delta) \leq \delta(2 + \delta n) = o(n^{-1}).
\end{aligned}$$

Also,

$$\begin{aligned}
&\left|\sum_{i=1}^{\infty} a(i)b(i) - \sum_{i=1}^{n} \tilde{a}(i, \delta)\tilde{b}(i, \delta)\right| \\
&\quad \leq \sum_{i=n+1}^{\infty} a(i)b(i) + \left|\sum_{i=1}^{n} a(i)b(i) - \sum_{i=1}^{n} \tilde{a}(i, \delta)\tilde{b}(i, \delta)\right| \\
&\quad \leq \left(\sum_{i=n+1}^{\infty} a(i)\right)\left(\sum_{i=n+1}^{\infty} b(i)\right) + o(n^{-1}) \\
&\quad = \left(1 - \sum_{i=1}^{n} a(i)\right)\left(1 - \sum_{i=1}^{n} b(i)\right) + o(n^{-1}) \\
&\quad \leq \left(1 - \sum_{i=1}^{n} \tilde{a}(i, \delta) + n^{-1}\right)\left(1 - \sum_{i=1}^{n} \tilde{b}(i, \delta) + n^{-1}\right) + o(n^{-1}) \\
&\quad \leq \left(1 - \sum_{i=1}^{n} \tilde{a}(i, \delta)\right)\left(1 - \sum_{i=1}^{n} \tilde{b}(i, \delta)\right) + 2\delta n + \delta^2 n^2 + o(n^{-1}).
\end{aligned}$$

It is obvious that when n tends to infinity, the error

$$\left|\sum_{i=1}^{n} \tilde{a}(i, \delta)\tilde{b}(i, \delta) - \sum_{i=1}^{n} a(i)b(i)\right|$$

tends to zero. This means that c can be approximated by

$$\sum_{i=1}^{n} \tilde{a}(i, \delta)\tilde{b}(i, \delta)$$

with arbitrary precision. An effective procedure for achieving a bound of ε on the error is to compute for increasing values of n the upper bound

$$\left(1 - \sum_{i=1}^{n} \tilde{a}(i, n^{-2})\right)\left(1 - \sum_{i=1}^{n} \tilde{b}(i, n^{-2})\right) + 2n^{-1} + n^{-2} + n^{-2}(2 + n^{-1}),$$

and as soon as a value of n is found such that the latter is less than ε, the corresponding approximation for c is guaranteed to be sufficient. ■

7. Examples of What a Program Computes As Beliefs

In this section we give some examples of computable beliefs.

Proposition 7.1. *The probability which M^k ascribes to an event $E = \{X_1 = M^i\}$ is computable.*

Proof. Obviously, the probability stated in the proposition is

$$p_{i\cdot}^k = p^k(\{X_1 = M^i\}) = \sum_{j=1}^{\infty} p_{ij}^k.$$

Recall that for any rational $\varepsilon > 0$, the program M^k computes a rational number $\tilde{p}_{ij}^k(\varepsilon)$ such that

$$|\tilde{p}_{ij}^k(\varepsilon) - p_{ij}^k| < \varepsilon.$$

For any positive integer n, let $\delta = o(n^{-2})$. We have

$$\left|\sum_{j=1}^{n} \tilde{p}_{ij}^k(\delta) - p_{i\cdot}^k\right| \leq \left|\sum_{j=1}^{n} \tilde{p}_{ij}^k(\delta) - \sum_{j=1}^{n} p_{ij}^k\right| + \left(p_{i\cdot}^k - \sum_{j=1}^{n} p_{ij}^k\right) \leq \delta n + \left(p_{i\cdot}^k - \sum_{j=1}^{n} p_{ij}^k\right).$$

It follows that when we let n tend to infinity, $\sum_{j=1}^{n} \tilde{p}_{ij}^k(\delta)$ tends to $p_{i\cdot}^k$. Now, in order to compute an ε-approximation, note that

$$\left|\sum_{l=1}^{n}\sum_{j=1}^{n}\tilde{p}_{lj}^{k}(\delta) - \sum_{l=1}^{n}\sum_{j=1}^{n}p_{lj}^{k}\right| \leq \delta n^2 = o(1)$$

so the sum $\sum_{l=1}^{n}\sum_{j=1}^{n}\tilde{p}_{lj}^{k}(\delta)$ tends to 1 as n tends to infinity. We thus have

$$\begin{aligned}\left|\sum_{j=1}^{n}\tilde{p}_{ij}^{k}(\delta) - p_{i\cdot}^{k}\right| &\leq o(n^{-1}) + \left(p_{i\cdot}^{k} - \sum_{j=1}^{n}p_{ij}^{k}\right)\\ &= o(n^{-1}) + \left(1 - \sum_{j=1}^{n}p_{ij}^{k} - \sum_{\substack{l=1\\ l\neq i}}^{\infty}\sum_{j=1}^{\infty}p_{lj}^{k}\right)\\ &\leq o(n^{-1}) + \left(1 - \sum_{l=1}^{n}\sum_{j=1}^{n}p_{lj}^{k}\right)\\ &\leq o(n^{-1}) + \left(1 - \sum_{l=1}^{n}\sum_{j=1}^{n}\tilde{p}_{lj}^{k}(\delta)\right) + \delta n^2.\end{aligned}$$

Thus, by evaluating all the $\tilde{p}_{lj}^{k}(\delta)$ $(1 \leq l, j \leq n)$ for increasing values of n, the program can actually compute an upper bound on the error in this approximation, which tends to zero with n. Thus, $p_{i\cdot}^{k}$ can be approximated with arbitrary precision. ■

COROLLARY 7.2. *A program $M^k \in \mathbf{S}$ can approximate its prior belief for the event that it will be loaded into C_l $(l = 1, 2)$, e.g.,*

$$p^k(\{X_2 = M^k\}) = \sum_{i=1}^{\infty}p_{ik}^{k}.$$

PROPOSITION 7.3. *For every set $S_1 \subseteq \mathbf{S}$, if there is an effective procedure for deciding whether $M^i \in S_1$, then the probability $p^k(\{X_1 \in S_1\})$ is computable.*

Proof. By Proposition 7.1, for every i and every rational $\varepsilon > 0$, the program M^k computes an approximation $\tilde{p}_{i\cdot}^{k}(\varepsilon)$ such that

$$|\tilde{p}_{i\cdot}^{k}(\varepsilon) - p_{i\cdot}^{k}| < \varepsilon.$$

Suppose M^k is provided with a decision procedure for testing for any i whether $M^i \in S_1$. Thus the program can compute for any n the sum

$$\sum_{\substack{i=1\\ M^i\in S_1}}^{n}\tilde{p}_{i\cdot}^{k},$$

and also estimate the difference between the latter and the exact probability. It is easy to see how this implies that the exact probability is computable. ■

If the program M^k itself is involved in the game, it computes conditional probabilities. These also turn out to be computable:

PROPOSITION 7.4. *The conditional probability which M^k ascribes to the event that the program residing in computer C_1 is M^i, given that M^k is residing in C_2, is computable.*

Proof. This conditional probability is given by

$$p^k(\{X_1 = M^i\}|\{X_2 = M^k\}) = \frac{p_{ik}^k}{\sum_{l=1}^{\infty} p_{lk}^k}.$$

Thus our claim follows from Propositions 7.1 and 6.2. ■

It is interesting to note that some expected values are computable:

PROPOSITION 7.5. *Let Y denote a random variable whose value is the conditional probability which the program residing in X_1 ascribes to the event $\{X_2 = M^k\}$, given that it knows it is residing in C_1. Let μ denote the conditional expected value of Y, relative to M^k's beliefs, given that M^k knows that $X_2 = M^k$. Under these conditions, μ is computable.*

Proof. Let

$$\beta_k'(i) = p^k(\{X_1 = M^i\}|\{X_2 = M^k\})$$

and

$$\beta_i''(k) = p^i(\{X_2 = M^k\}|\{X_1 = M^i\}).$$

By Proposition 7.4 both $\beta_k'(i)$ and $\beta_i''(k)$ are computable. Now,

$$\mu = \sum_{i=1}^{\infty} \beta_k'(i)\beta_i''(k)$$

so by arguments similar to those of Proposition 6.4 it is computable. We note that the quantity $\beta_i''(k)$ is not well defined if $M^i \notin \mathbf{S}$. However, since in this case $\beta_k'(i) = 0$ there is no problem. ■

Remark 7.6. Although the expected value of the variable Y of Proposition 7.5 is computable, some other numbers related to its distribution are not computable. This is due to the fact that, for example, there is no general effective procedure for deciding whether a computable real num-

ber a is greater than 1. We can compute ε-approximations $\tilde{a}(\varepsilon)$ of a for any rational $\varepsilon > 0$. The case $a \neq 1$ is decidable since, if we let $\varepsilon = 1/n$ for $n = 1, 2, \ldots$, when we reach $\varepsilon < \frac{1}{2}|a - 1|$, we observe that $|\tilde{a}(\varepsilon) - 1| > \varepsilon$ and we then know that $a > 1$ if and only if $\tilde{a}(\varepsilon) > 1$. On the other hand, if $a = 1$ we may never be able to conclude anything. In fact, there does not exist a general program which can decide, given the description of a program which computes a number a (in the asymptotic sense, with no input), whether $a = 1$. The proof of this claim is standard and proceeds as follows. Suppose, to the contrary, there exists such a program. Then there exists a program that decides for any program x and any input y whether the number $x(y)$ computed by x in the asymptotic sense, given the input number y, is 1. Furthermore, there exists a program z which computes the number $z(x) = 1$ given the input x if $x(x) \neq 1$, and $z(x) = 0$ otherwise. It turns out that if $z(z) = 1$ then $z(z) \neq 1$ and if $z(z) \neq 1$ then $z(z) = 1$.

The difficulty pointed out in Remark 7.6 is interesting in its own right. In the traditional theory, when a person wants to find out what his/her subjective probability $p(E)$ is for some event, s/he tries to compare $p(E)$ in a binary search fashion with numbers such as 0.5, 0.75, 0.675, and so on, so as to get a good approximation. However, a program can compute approximations but cannot in general perform a single comparison.

Note that it is still possible to perform comparisons in an approximate sense as follows.

PROPOSITION 7.7. *If a and b are computable then there exists a program which recognizes for any given rational $\varepsilon > 0$ either that $a \leq b + \varepsilon$ or that $a \geq b - \varepsilon$.*

Proof. Let $c = b - a$ and suppose C is a program which computes for any rational $\varepsilon > 0$ a rational number $\tilde{c}(\varepsilon)$ such that $|\tilde{c}(\varepsilon) - c| < \varepsilon$. Obviously, if $\tilde{c}(\varepsilon) \geq 0$ then $c > -\varepsilon$, and if $\tilde{c}(\varepsilon) \leq 0$ then $c < \varepsilon$. ■

Remark 7.8. Despite the positive tone of Proposition 7.7, there is still a difficulty in computing probabilities as in the following example. Let α_k denote the conditional probability (given that $X_2 = M^k$) which M^k ascribes to the event that X_1 ascribes probability of at least 90% to the event that $X_2 = M^k$. The conditional probability

$$\beta_i''(k) = p^i(\{X_2 = M^k\} | \{X_1 = M^i\})$$

is computable. Let S_k denote the set of indices i such that $\beta_i''(k) \geq 0.9$. Then

$$\alpha_k = \sum_{i \in S_k} p^k(\{X_1 = M^i\} | \{X_2 = M^k\}).$$

However, we do not have an effective procedure for deciding whether $i \in S_k$. So, it seems that α_k cannot in general be approximated with an arbitrarily small error.

8. Computable Random Variables and Their Distributions

In this section we present some facts about the computability of the probability distribution of certain random variables.

Definition 8.1. Consider a discrete random variable Y which attains values y_i with respective probabilities $p_i \geq 0$ ($i = 1, 2, \ldots$). Thus, $\sum_{i=1}^{\infty} p_i = 1$. We say that Y is *computable* if there exists a program A that computes ε-approximations $\tilde{p}_i(\varepsilon)$ and $\tilde{y}_i(\varepsilon)$ of p_i and y_i, respectively, for any i and any rational $\varepsilon > 0$.

Denote

$$S(t) = \{i : y_i \leq t\}.$$

The cumulative distribution function (c.d.f.) of Y is given by

$$F(t) = \sum_{i \in S(t)} p_i.$$

We now consider the problem of approximating the c.d.f. of a computable random variable. For any rational $\delta > 0$ and any positive integer n, denote

$$S_n^+(\delta, t) = \{i : \tilde{y}_i(\delta) \leq t + \delta,\ 1 \leq i \leq n\}$$

and

$$S_n^-(\delta, t) = \{i : \tilde{y}_i(\delta) \leq t - \delta,\ 1 \leq i \leq n\}.$$

Let

$$F^+(t; \delta, n) = \sum_{i \in S_n^+(\delta, t)} \tilde{p}_i(\delta)$$

and

$$F^-(t; \delta, n) = \sum_{i \in S_n^-(\delta, t)} \tilde{p}_i(\delta).$$

Fact 8.2. $F^-(t; \delta, n) - \delta n \leq F(t) \leq F^+(t; \delta, n) + (1 - \sum_{i=1}^{n} \tilde{p}_i(\delta)) + 2\delta n$.

Proof. The lower bound follows from

$$F(t) \geq \sum_{\substack{i=1 \\ y_i \leq t}}^{n} p_i.$$

The upper bound follows from

$$F(t) \leq \sum_{\substack{i=1 \\ y_i \leq t}}^{n} p_i + \sum_{i=n+1}^{\infty} p_i \leq F^+(t; \delta, n) + \delta n + \left(1 - \sum_{i=1}^{n} \tilde{p}_i(\delta)\right) + \delta n. \quad \blacksquare$$

COROLLARY 8.3. *If* $\delta = o(n^{-1})$ *then*

$$\lim_{n \to \infty} F^-(t; \delta, n) \leq F(t) \leq \lim_{n \to \infty} F^+(t; \delta, n).$$

PROPOSITION 8.4. *For every computable* t *such that* $t \neq y_i$ *for all* i, *the value* $F(t)$ *is computable.*

Proof. Given t such that $t \neq y_i$ for all i, denote

$$S_n^0(\delta, t) = \{i : t - \delta < \tilde{y}_i(\delta) \leq t + \delta, 1 \leq i \leq n\}.$$

Thus,

$$\Delta \equiv F^+(t; \delta, n) - F^-(t; \delta, n) = \sum_{i \in S_n^0(\delta, t)} \tilde{p}_i(\delta).$$

Obviously, as δ tends to zero, the sum

$$\sum_{t - \delta < y_i \leq t + \delta} p_i$$

tends to zero. It is thus easy to see that Δ tends to zero if δ does, and hence the difference between the bounds stated in Fact 8.2 tends to zero when n tends to infinity and $\delta = o(n^{-1})$. Since these bounds are computed exactly by a program, the program can approximate $F(t)$ with any prescribed precision. $\blacksquare$

DEFINITION 8.5. We say that the c.d.f. $F(t)$ of a random variable is *computable in the weak sense* if the following is true. There exists a program A such that, given any computable t and any rational $\varepsilon > 0$, A finds values t^-, t^+ such that

$$t - \varepsilon \leq t^- \leq t \leq t^+ \leq t + \varepsilon$$

and values $\tilde{F}(t^-, \varepsilon)$ and $\tilde{F}(t^+, \varepsilon)$ such that

$$|\tilde{F}(t^-, \varepsilon) - F(t^-)| \leq \varepsilon$$

and

$$|\tilde{F}(t^+, \varepsilon) - F(t^+)| \leq \varepsilon.$$

PROPOSITION 8.6. *If Y is a computable random variable then its c.d.f. is computable in the weak sense.*

Proof. Consider the computation of t^- and $\tilde{F}(t^-, \varepsilon)$; the computation of t^+ and $\tilde{F}(t^+, \varepsilon)$ is analogous. Since t is computable, the program can compute a sequence $\{t_j\}$ of distinct rational numbers which converges to t from below. For any n let $\delta = \delta(n) > 0$ be smaller than half the minimum distance between any $t_i \neq t_j$ such that $i, j \leq n$. Thus the intervals $(t_i - \delta, t_i + \delta)$ $(i = 1, \ldots, n)$ are disjoint. It follows that

$$\sum_{i=1}^{n} (F^+(t_i; \delta, n) - F^-(t_i; \delta, n)) \leq \sum_{i=1}^{n} \tilde{p}_i(\delta) \leq 1 + \delta n.$$

This means that the minimum

$$\min_{1 \leq i \leq n} (F^+(t_i; \delta, n) - F^-(t_i; \delta, n))$$

tends to zero as n tends to infinity. By choosing t^- to be a minimizer t_i (for sufficiently large n), we get an ε-approximation for $F(t^-)$ for a value t^- arbitrarily close to t. ■

COROLLARY 8.7. *If Y is a computable random variable, then there exists a program A such that for every computable number y and every rational $\varepsilon > 0$, A computes an interval I of length $|I| < \varepsilon$, which contains y, and an ε-approximation to the probability that Y is in I.*

Remark 8.8. If $\{I_\varepsilon\}$ is a family of intervals containing y, such that $|I_\varepsilon| < \varepsilon$, then for any random variable Y and any probability measure p,

$$\lim_{\varepsilon \to 0} p(\{Y \in I_\varepsilon\}) = p(\{Y = y\}).$$

Thus, the ε-approximations claimed in Corollary 8.7 converge to the

probability of $\{Y = y\}$. Nevertheless, the program cannot compute ε-approximations to the latter with a prescribed ε.

Remark 8.9. As noted above, the beliefs of programs with respect to certain random variables may be determined by some consistency requirements even though the programs cannot compute them. Thus we may denote by $p^k(\{Y = y\})$ the probability ascribed by M^k to the event $\{Y = y\}$ whenever this value is determined by probabilities ascribed by M^k to some other events. We have seen examples of such cases where $p^k(\{Y = y\})$ is the sum of a well-defined infinite series. The conclusion of Corollary 8.7 suggests that we might relax the definition of computability of a random variable as follows. Let us say that a random variable Y is pseudo-computable for M^k if the probability distribution ascribed to Y by M^k is well defined and discrete and has the following property. Given any computable y and rational $\varepsilon > 0$, M^k computes an interval I, $|I| < \varepsilon$, which contains y, and an ε-approximation to the probability $p^k(\{Y \in I\})$. Unfortunately, it seems that this notion is yet too restrictive. To clarify this point, suppose Y is pseudo-computable for every M^k and let y be any computable number. Denote by Z the probability ascribed by X_i (i.e., the program residing in C_i) to the event $\{Y = y\}$. Here Z is not even pseudo-computable since we must replace not only values z of Z by small intervals but also values y of Y by such intervals. We propose below a weaker notion of computability which seems more fit.

9. Computable Beliefs

We first introduce some notation for discussing more general computable beliefs. Let E be any computable event. For any interval I (which may consist of a single point t) denote by $E[I; i]$ the event in which the probability ascribed by X_i to the event E lies in the interval I. Inductively, let

$$E[I_1, \ldots, I_l; i_1, \ldots, i_l] = \{p^{X_{i_l}}(E[I_1, \ldots, I_{l-1}; i_1, \ldots, i_{l-1}]) \in I_l\}.$$

Definition 9.1. If

$$\underline{\lim}_{x_1 \to 0} \cdots \underline{\lim}_{x_n \to 0} f(x_1, \ldots, x_n) = \overline{\lim}_{x_1 \to 0} \cdots \overline{\lim}_{x_n \to 0} f(x_1, \ldots, x_n)$$

then we denote the common value of these limits by

$$\operatorname{Lim}_{x_1, \ldots, x_n \to 0} f(x_1, \ldots, x_n).$$

To simplify notation, we omit the indices $i_1, \ldots, i_l$. Also, let $\check{I}$ and $\bar{I}$ denote, respectively, the interior and the closure of an interval I. The following proposition is an extension of Remark 8.8.

PROPOSITION 9.2. *For any family of intervals,* $I_j(\varepsilon)$ $(j = 1, \ldots, l, \varepsilon > 0)$, *if* $t_j \in \check{I}_j(\varepsilon)$ *and* $|I_j(\varepsilon)| < \varepsilon$ *then*

$$\operatorname*{Lim}_{\varepsilon_l,\ldots,\varepsilon_1\to 0} p^k(E[I_1(\varepsilon_1), \ldots, I_l(\varepsilon_l)]) = p^k(E[t_1, \ldots, t_l]).$$

Proof. The proof goes by induction on l. The case $l = 1$ was already mentioned in Remark 8.8. For the inductive step, note that

$$\begin{aligned} p^k(E[I_1, \ldots, I_l]) &= p^k(\{p^{X_{i_l}}(E[I_1, \ldots, I_{l-1}]) \in I_l\}) \\ &= \sum_{m\in R[I_1,\ldots,I_l]} p^k(\{X_{i_l} = M^m\}), \end{aligned}$$

where

$$R[I_1, \ldots, I_l] = \{m: p^m(E[I_1, \ldots, I_{l-1}]) \in I_l\}.$$

By the induction hypothesis,

$$\operatorname*{Lim}_{\varepsilon_{l-1},\ldots,\varepsilon_1\to 0} p^m(E[I_1(\varepsilon_1), \ldots, I_l(\varepsilon_l)]) = p^m(E[t_1, \ldots, t_l]).$$

It follows that

$$\overline{\lim_{\varepsilon_{l-1}\to 0}} \cdots \overline{\lim_{\varepsilon_1\to 0}} p^k(E[I_1(\varepsilon_1), \ldots, I_l(\varepsilon_l)]) \le \sum_{m\in R[t_1,\ldots,t_{l-1},\bar{I}_l]} p^k(\{X_{i_l} = M^m\})$$

and

$$\overline{\lim_{\varepsilon_{l-1}\to 0}} \cdots \overline{\lim_{\varepsilon_1\to 0}} p^k(E[I_1(\varepsilon_1), \ldots, I_l(\varepsilon_l)]) \ge \sum_{m\in R[t_1,\ldots,t_{l-1},\check{I}_l]} p^k(\{X_{i_l} = M^m\}).$$

It is easy to see that, as ε_l tends to zero, the right-hand sides of the latter inequalities tend to the sums taken over

$$R[t_1, \ldots, t_{l-1}, t_l]$$

and this implies our claim. ∎

Remark 9.3. It is interesting to note the complications associated with the limits discussed in Proposition 9.2. It seems that the limit would

behave more regularly if we replaced the general families of intervals $I_j(\varepsilon)$ (a family for each j, satisfying $t_j \in I_j(\varepsilon)$ and $|I_j(\varepsilon)| < \varepsilon$) by sequences of the form $I_j(\varepsilon) = (t_j - \delta, t_j = \delta)$. However, this simplification implies a limit in the usual sense ("simultaneous") only if $l < 2$. More specifically, first recall that for $l = 1$ we always have

$$\lim_{\varepsilon_1 \to 0} p^k(E[I_1(\varepsilon_1)]) = p^k(E[t_1]),$$

since

$$p^k(E[I_1(\varepsilon_1)]) = p^k(\{p^{X_{i_1}}(E) \in I_1\}).$$

Moreover, if $\{I_1(\varepsilon_1)\}$ is a nested family of intervals, then for every k, the function

$$f^k(\varepsilon_1) = p^k(E[I_1(\varepsilon_1)])$$

decreases monotonically to $p^k(E[t_1])$ as ε_1 tends to 0. Now, consider the case $l = 2$. We know that

$$p^k(E[I_1(\varepsilon_1), I_2(\varepsilon_2)] = \sum_{m \in R[I_1, I_2]} p^k\{X_{i_2} = M^m\},$$

where

$$R[I_1, I_2] = \{m: p^m(E[I_1]) \in I_2\}.$$

For any fixed I_2, let ε_1 tend to zero, and consider the varying set $R[I_1, I_2]$. Obviously, in this process every m enters this set at most once and leaves it at most once. The contribution of m to $p^k(E[I_1(\varepsilon_1), I_2(\varepsilon_2)])$ is $p^k(\{X_{i_2} = M^m\})$ and the sum of all these values is of course bounded. Thus, this contribution tends to zero as m tends to infinity. It follows that, as ε_1 tends to zero, the size of the jumps in the value of $p^k(E[I_1(\varepsilon_1), I_2(\varepsilon_2)])$ tends to zero. This means that the limit exists. However, monotonicity is not guaranteed since there can be infinitely many values of m entering and leaving the set $R[I_1, I_2]$ in the limit process, during which I_2 is fixed, and this may happen for infinitely many intervals I_2. Since monotonicity is not guaranteed, it may happen that in the case $l = 2$, a limit in the usual sense will not exist.

Proposition 9.2 provides the justification for an approximate computation of $p^k(E[t_1, \ldots, t_l])$ in a sense defined below. We first consider the case $l = 1$.

PROPOSITION 9.4. *There exists a program A which does the following. It receives a program M^k, a computable event E, an index i, and rational numbers t and $\varepsilon > 0$. It then computes an interval I such that $t \in I$ and $|I| < \varepsilon$, and an ε-approximation to the probability $p^k(E[I; i])$.*

Proof. First, note that

$$p^k(E[I; i]) = \sum_{m: p^m(E) \in I} p^k(\{X_i = M^m\}).$$

Consider intervals of the form $I(\delta) = (t - 2\delta, t + 2\delta)$. Denote by $\tilde{p}^m(E, \delta)$ the δ-approximation computed by M^m for $p^m(E)$. Let $U(\delta)$ denote the set of m's such that

$$|\tilde{p}^m(E, \delta) - t| \leq \delta.$$

Obviously, if $m \in U(\delta)$ then $p^m(E) \in I(\delta)$. Let $W(\delta)$ denote the set of m's such that either

$$\tilde{p}^m(E, \delta) \geq t + 3\delta$$

or

$$\tilde{p}^m(E, \delta) \leq t - 3\delta.$$

Similarly, if $m \in W(\delta)$ then $p^m(E) \notin I(\delta)$. The remaining values of m are those for which either

$$t + \delta < \tilde{p}^m(E, \delta) < t + 3\delta$$

or

$$t - 3\delta < \tilde{p}^m(E, \delta) < t - \delta.$$

Denote the set of these m's by $V(\delta)$. We claim that for every m, there exists $\delta^* = \delta^*(m)$ such that for all $\delta < \delta^*$, $m \notin V(\delta)$. For if $p^m(E) = t$ then $m \in U(\delta)$ and if $p^m(E) \neq t$ then for all δ sufficiently small $m \in W(\delta)$. Now, for every n

$$\sum_{\substack{m=1 \\ m \in U(\delta)}}^{n} \tilde{p}^k(\{X_1 = M^m\}, \delta) \leq p^k(E[I; i])$$

$$\leq \sum_{\substack{m=1 \\ m \in U(\delta) \cup V(\delta)}}^{n} \tilde{p}^k(\{X_i = M^m\}, \delta) + \left(1 - \sum_{m=1}^{n} \tilde{p}^k(\{X_i = M^m\}, \delta)\right) + \delta n.$$

These estimates suggest how to effectively choose n and δ so as to compute the approximations as required. Specifically, if $\delta = o(n^{-1})$ the difference between the lower and upper bounds on $p^k(E[I; i])$ tends to zero as n tends to infinity. Note that both these bounds can be computed exactly. ■

We now consider the general case.

PROPOSITION 9.5. *There exists a program A which does the following. It receives a program M^k, a computable event E, indices $i_1, \ldots, i_l$, rational numbers $t_1, \ldots, t_l$, and $\varepsilon > 0$. It then computes open intervals $I_1, \ldots, I_l$ such that $t_j \in I_j$ and $|I_j| < \varepsilon$ $(j = 1, \ldots, l)$, and an ε-approximation to the probability*

$$p^k(E[I_1, \ldots, I_l]) = p^k(E[I_1, \ldots, I_l; i_1, \ldots, i_l]).$$

Proof. We sketch a program which recurses on the value of l. The case $l = 1$ was proven in Proposition 9.4. Suppose $l > 1$ and let the inputs M^k, E, i_j, t_j $(j = 1, \ldots, l)$, and ε be given. Recall from the proof of Proposition 9.2 that

$$p^k(E[I_1, \ldots, I_l]) = \sum_{m \in R[I_1, \ldots, I_l]} p^k(\{X_{i_l} = M^m\}),$$

where

$$R[I_1, \ldots, I_l] = \{m: p^m(E[I_1, \ldots, I_{l-1}]) \in I_l\}.$$

Our program works by recursing to problems of approximating $p^m(E[I_1, \ldots, I_{l-1}])$ for $m = 1, \ldots, n$, where n is determined by the program. The complete algorithm therefore computes approximations for $p^{m_j}(E[I_1, \ldots, I_j])$ for $m_j = 1, \ldots, n_j$ (where n_j is determined by the program) for $j = 1, \ldots, l - 1$; the intervals I_j turn out to be the same for all values of m_j, depending on j. We compute intervals of the form

$$I_j = (t_j - 2\delta_j, t_j + 2\delta_j).$$

Actually, the value of δ_{l-1} is determined with respect to δ_l, the value of δ_{l-2} is determined with respect to δ_{l-1}, and so on. Thus we actually prove the following:

Claim. There exists a program that computes positive rationals $\delta_1, \ldots, \delta_l$ (where $\delta_l < \varepsilon$), positive integers $n_1, \ldots, n_l$, intervals I_j as defined above, and approximations as follows:

(i) a δ_{j+1}-approximation for $p^{m_{j+1}}(\{X_{i_j} = M^{m_j}\})$ $(m_j = 1, \ldots, n_j)$,

$$\tilde{p}^{m_{j+1}}(\{X_{i_j} = M^{m_j}\}, \delta_{j+1});$$

(ii) a δ_{j+1}-approximation for $p^{m_{j+1}}(E[I_1, \ldots, I_j])$,

$$\tilde{p}^{m_{j+1}}(E[I_1, \ldots, I_j], \delta_{j+1}) = \sum_{\substack{m_j=1 \\ m_j \in U[\delta_1, \ldots, \delta_j]}}^{n_j} \tilde{p}^{m_{j+1}}(\{X_{i_j} = M^{m_j}\}, \delta_j),$$

where

$$U[\delta_1, \ldots, \delta_j] = \{m_j : |\tilde{p}^{m_j}(E[I_1, \ldots, I_{j-1}], \delta_j) - t_j| < \delta_j\}.$$

To prove the claim, suppose we have established the existence of a program which does all the above for the values $1, \ldots, j-1$, and consider the case of the value j. Note that only part (ii) of the claim must be proven. We rely on estimates similar to those made in the proof of Proposition 9.4. First, note that if $m_j \in U[\delta_1, \ldots, \delta_j]$ then $p^{m_j}(E[I_1, \ldots, I_{j-1}]) \in I_j$. Now, let $W[\delta_1, \ldots, \delta_j]$ denote the set of values of m_j such that either

$$\tilde{p}^{m_j}(E[I_1, \ldots, I_{j-1}], \delta_j) \geq t_j + 3\delta_j$$

or

$$\tilde{p}^{m_j}(E[I_1, \ldots, I_{j-1}], \delta_j) \leq t_j - 3\delta_j.$$

It follows that if $m_j \in W[\delta_1, \ldots, \delta_j]$ then $p^{m_j}(E[I_1, \ldots, I_{j-1}]) \notin I_j$. The remaining values of m_j are those for which either

$$t_j + \delta_j < \tilde{p}^{m_j}(E[I_1, \ldots, I_{j-1}], \delta_j) < t_j + 3\delta_j$$

or

$$t_j - 3\delta_j < \tilde{p}^{m_j}(E[I_1, \ldots, I_{j-1}], \delta_j) < t_j - \delta_j.$$

We denote the set of these values of m_j by $V[\delta_1, \ldots, \delta_j]$. Given a set of values $\delta_1, \ldots, \delta_{j-1}$, for every value of m_j, there exists $\delta_j^* = \delta_j^*(m_j; \delta_1, \ldots, \delta_{j-1})$ such that for all $\delta_j < \delta^*$,

$$m_j \notin V[\delta_1, \ldots, \delta_{j-1}, \delta_j].$$

Now, for every n_j we have

$$\tilde{p}^{m_{j+1}}(E[I_1, \ldots, I_j]) \geq \sum_{\substack{m_j=1 \\ m_j \in U[\delta_1,\ldots,\delta_j]}}^{n_j} \tilde{p}^{m_{j+1}}(\{X_{i_j} = M^{m_j}\}, \delta_j)$$

and

$$p^{m_{j+1}}(E[I_1, \ldots, I_j]) \geq \sum_{\substack{m_j=1 \\ m_j \in U \cup V}}^{n_j} \tilde{p}^{m_{j+1}}(\{X_{i_j} = M^{m_j}\}, \delta_j) + \left(1 - \sum_{m_j=1}^{n_j} \tilde{p}^{m_{j+1}}(\{X_{i_j} = M^{m_j}\}, \delta_j)\right) + \delta_j n_j$$

(where $U = U[\delta_1, \ldots, \delta_j]$ and $V = V[\delta_1, \ldots, \delta_j]$). These estimates suggest how to effectively choose n_j and δ_j so as to compute the approximations as required. Specifically, given the requirement δ_{j+1}, we run over values of n_j, taking $\delta_j = o(n_j^{-1})$. For every δ_j we recurse and find the approximations and δ's from smaller problems. We then observe the difference between the upper bound and the lower bound derived above. When the latter becomes less than the given δ_{j+1}, an approximation as required in (ii) has been found. ■

Remark 9.6. It is clear that a stronger result can be proven as follows. Instead of the rational numbers $t_1, \ldots, t_l$ in Proposition 9.5, we could use sets $T_1, \ldots, T_l$ which are computable in some obviously defined sense. For every j, the interval I_j would then be interpreted as a δ_j-neighborhood of the set T_j.

As pointed out earlier, beliefs about computable events are themselves computable. On the other hand, there exist noncomputable events. Among the noncomputable events, we are especially interested in events defined in terms of beliefs about beliefs (and so on) about computable events. The above results indicate that these can be approximated in a natural well-defined sense. A class **E** of such events is defined as follows. Recall that the sample space (or the space of "states of the world" or "possible worlds") consists of combinations of programs. Thus, we consider a pair of random variables (X_1, X_2), specifying the programs residing in the two computers which play the game.

We start with the set $\mathbf{E}_0$ of computable events. Recall that a subset E of the sample space is called a computable event if there exists a program A which decides for any point x of the space whether $x \in E$. The program A may be considered the description of the event E. It was shown in Proposition 7.3 that the probability ascribed by M^k to a computable event is a computable real number. The set of computable events is of course closed under finite union and complementation. Since every subset of the sample

space is a countable union of computable events (namely, singleton sets), it follows by a cardinality argument that there exist noncomputable events which are themselves countable unions of computable ones.

Next, we define a set $\mathbf{E}_1$ as follows. A basic event $E' \in \mathbf{E}_1$ is a set of points defined by an inequality of the form

$$p^{X_i}(E) \geq \pi$$

(where $E \in \mathbf{E}_0$) which reads: "The probability ascribed to the event E by the program residing in the computer C_i is at least π." The set $\mathbf{E}_1$ is the algebra spanned by the basic events E' (by finite unions and complementations). We could define $\mathbf{E}_1$ to be larger by allowing E' to be defined by more general predicates than the inequality given above, but we prefer, for simplicity, not to do so.

As indicated above, events in $\mathbf{E}_1$ are in general not computable. Moreover, the probability which a program must ascribe (in order to be consistent) to an event of the type E' may be noncomputable. However, as pointed out in Proposition 9.5, the program can approximate its belief with respect to some event $\tilde{E}$ "close" to E' (for example, $\tilde{E} = \{p^{X_i}(E) \geq \tilde{\pi}\}$ where $\tilde{\pi}$ is arbitrarily close to π). Inductively, $\mathbf{E}_{j+1}$ is the algebra spanned by events of the form $p^{X_i}(E) \geq \pi$ where $E \in \mathbf{E}_j$. Finally, $\mathbf{E} = \bigcup_{j=1}^{\infty} \mathbf{E}_j$.

It is easy to see that every event $E \in \mathbf{E}$ can be represented by some computable events, some logical connectives, and some numerical parameters $\pi_1, \ldots, \pi_q$. The sense of the approximate computation is that, given any $\varepsilon > 0$, the program computes an event $\tilde{E}$ which is close to E in the sense that the numerical parameters are changed by amounts up to ε. Furthermore, it also computes an ε-approximation to the belief with respect to $\tilde{E}$. Replacing E by $\tilde{E}$ can be easily justified in practice. Note that the computable events involved in the definitions of E and $\tilde{E}$ are the same. Only the numerical parameters which signify probabilities are different. It seems that in every practical situation there exists an $\varepsilon > 0$ such that changes in probabilities within ε do not really matter.

10. Conclusion

Since a computer program is necessarily limited in what it can do, there must be some freedom in defining a class of rational programs. It is expected that different classes of programs could serve as candidates. We have considered two properties A1 and A2 of what we see as candidates. Specifically, in a candidate set **S** each member has a joint probability distribution with respect to the identities of the other players, ascribing probability zero to the event that any player is not in **S**.

In view of the results proven in this paper, we can say that if a class **S** has these properties then there exists a class **S*** as follows. For every member M of **S**, there exists a member M^* of **S*** which "emulates" M and is also capable of computing its beliefs with respect to events in **E** in the approximate sense discussed above. This is true because we have proven the existence of a program for carrying out these computations so this program can be "added" to each member of **S**. Thus, we might add a third requirement which would say that only classes of the form **S*** qualify as rational. Our results indicate that this third requirement is not too restrictive. To narrow the set of candidate classes even further, one would have to introduce more axioms. Our proposal should be considered a first step away from the classical abstract assumption that all players are rational and that this fact is common knowledge. We have not considered here situations where some players are allowed to be "irrational." Such situations could be difficult for programs to handle since sometimes a "rational" program M^i might ascribe a positive probability to an event where another program M^j does not halt when it attempts to calculate its beliefs. In such a case, the "rational" program would have to compute its belief about whether another program halts in a certain computation.

REFERENCES

AUMANN, R. J. (1985). *Correlated Equilibrium As an Expression of Bayesian Rationality*. Institute of Mathematics, Jerusalem: The Hebrew University.

BINMORE, K. G. (1987). *Remodeled Rational Players*. London School of Economics.

FAGIN, R., AND HALPERN, J. Y. (1988). "Reasoning about Knowledge and Probability," *in Proc. 2nd Conf. on Theoretical Aspects of Reasoning about Knowledge* (M. Y. Vardi, Ed.), pp. 227–293. Los Altos, CA: Kaufmann.

FAGIN, R., HALPERN, J. Y., AND MEGIDDO, N. (1988). "Logic for Reasoning about Probabilities," *in Proc. 3rd IEEE Symp. on Logic in Computer Science*, pp. 277–291. Los Angeles, CA: IEEE.

GAIFMAN, H. (1986). "A Theory of Higher Order Probabilities," *in Reasoning about Knowledge* (J. Y. Halpern, Ed.). Los Altos, CA: Kaufmann.

HARSANYI, J. C. (1967–1968). "Games with Incomplete Information Played by Bayesian Players, I, II, III," *Manage. Sci.* **14.**

KONOLIGE, K. (1986). *A Deduction Model of Belief*. London: Pitman.

LUCE, R. D., AND RAIFFA, H. (1957). *Games and Decisions: Introduction and Critical Survey*. New York: Wiley.

MEGIDDO, N. (1986). *Remarks on Bounded Rationality*, Research Report RJ 5270. San Jose, CA: IBM Almaden Research Center.

MERTENS, J.-F., AND ZAMIR, S. (1985). "Formulation of Bayesian Analysis for Games with Incomplete Information," *Int. J. Game Theory* **14**, 1–29.