Computability and Complexity in Games

K. (Vela) Velupillai
Department of Economics
The Queen's University of Belfast
Belfast BT7 1NN
Northern Ireland
UK

April 1995

"On the first part of my inquiry I soon arrived at a demonstration that every game of skill is susceptible of being played by an automaton."

Charles Babbage

§1. Introduction.

> "………. there are games in which the players who in theory can always win cannot do so in practice because it is impossible to supply him with *effective instructions* regarding how he should play in order to win."
>
> Rabin, 1957, p. 148; italics added.

The above is a statement of the first important result in what I have come to call *computable economics*. The key word in Rabin's quote is *effective*. In this paper I aim to discuss the question of *effective playability* and the *computational and diophantine complexity* of Rabin-type games[1] from various recursion theoretic vantage points.

There is a growing literature on applying recursion theoretic concepts to traditional issues in game theory. Much of this is expertly summarized in Holm's doctoral dissertation (Holm, 1993, especially chapters 4 and 5). I have, myself, been sceptical about the computable structure of many of these models. There are, of course, notable exceptions[2]. On the whole, my view on this literature is best summarized in the following important observation by Megiddo and Wigderson:

> "It has been said that the prisoner's dilemma may be 'resolved' by 'bounded rationality'. One possibility of bounding rationality is to have players play the game through computing machines. ……….. The only restriction is on the number of internal states of the machine ……… [However] it should be emphasized that most of *the results are nonconstructive and we do not believe that [this] approach ……… will lead to a practical resolution of the prisoner's dilemma."*
>
> Megiiddo and Wigderson, 1986, p. 260; italics added.

As a matter of fact it is easy to identify the nonconstructive nature of the results as emanating (not only from the noneffective rules of the underlying games[3] but also) from

[1] In other words, variations of the classic Gale-Stewart games which were perceptively stripped of their noneffective content by Rabin in his brilliant and pioneering contribution - now almost four decades ago. That it is still a fertile framework in which to explore recursion theoretic issues in games is what I intend to demonstrate in this essay.

[2] Above all Binmore (1987).

[3] As Rabin notes in his pioneering paper:

the backward induction proofs. It must be noted that it is customary to use nonconstructive methods in recursion theory too; but existential results come with effective processes to realize them. It is the (universal) undecidable results that are, in general, proved with free swinging nonconstructive methods[4].

I shall, instead, return to the Rabin tradition of seeking recursion theoretic formalisms in game theory (and link it to the classics as well). Indeed, a return to the classics makes it easier to link up with the frontiers in what I call applied recursion theory: computational and diophantine complexity theories and an introduction to the use of new recursion theoretic techniques to simplify proofs in *arithmetical games*.

The paper is organized as follows. In the next section there is a detailed description of the Rabin model and an exhaustive discussion; there is also an introductory discussion and statement of Jones's variation of the Rabin game (Jones, 1974). The reason for the disproportionately long §2 are twofold: one, it gives me a chance to describe the imaginative way Rabin stripped away the noneffective content of the Gale-Stewart game; secondly, via Jones's modified version of Rabin's model I get, eventually, the chance to introduce, in the proofs, *the busy beaver game*. It may be useful, for the uninitiated in recursion theory, to know how an explicit noncomputable function is actually constructed and then to literally *see* the nature of the dimensional monstrosities inherent even in deceptively simple-looking constructions. This background will prepare the sceptical reader to the melancholy fact that most games, even when determined and playable, are intractably complex.

> ".... there should be no sense in worrying about the effective computability of a strategy when the rules of the game in which it should be applied are not effective."
>
> ibid, p. 148, f.n.3;.

In fact, in most - if not all - of the application of recursion theoretic concepts in game theory, the rules of the game, the equilibria sought etc, remain noneffective.

[4] As Davis et. al pointed out:

> ".... these [reductio ad absurdum and other noneffective methods] may well be the best available to us because universal methods for giving all the solutions do not exist."
>
> Davis, Matiyasevic and Robinson, 1976, p. 340.

The basic theme in section 3 is to introduce the idea of *arithmetical games* and to show that they are a natural development of Rabin's *effectivized* Gale-Stewart game. Once we have this at hand it is possible to exploit the elegant results that were by-products of the long saga that led to the negative solution to *Hilbert's 10th problem: the diophantine nature of recursively enumerable sets.* In other words, we reduce the analysis to questions of *arithmetic* (hence *arithmetical games*); the main question that Rabin posed. *effective playability* is answered by looking at the *effective solvability* of diophantine equations by exploiting the above fundamental result.

The concluding section, §4, is a brief introduction to the issue of *diophantine* and *computational complexity in arithmetical games.* The section also includes my suggestions for directions in which the themes of this essay should, or can, be developed as part of a broad research strategy in computable games.

§2. Rabin's Computable Game - and Extensions.

Gale-Stewart Games:

(i) A game in extensive from (a tree);

(ii) Each vertex of the tree corresponds to a particular position of the game;

(iii) Branches from vertices denote possible plays - possible positions at next moves, from a given vertex;

(iv) The root or initial vertex, of a tree corresponds to the initial position of a game;

(v) Each vertex of a tree is assigned to a player, according to whose move it is, at that vertex;

(vi) The Game (Binary Games):

Two-player, infinite games;

Players: I and II;

I and II choose, alternately, binary digits 0 or 1;

Definition: A Play:

A countable number of such moves; i.e. each play consists of a sequence ⇒ s(0), s(1), ... [s(i) = 0 or 1]

and each such sequence determines a (real) number ⇒

$$s = \sum_{n=1}^{\infty} \frac{s(n)}{2^n}$$

Definition: Win-Lose:

(a) Player I wins a $ from player II if s belongs to a certain subset T of the unit interval;

(b) Player II wins a $ from player I if s does not belong to T;

Definition: Result:

A set T can be chosen so that no matter what strategy player I chooses, ∃ a strategy by which player II can win. (Contrariwise: for each strategy player II may elect to use, ∃ a strategy by which player I can win).

(Notational) Definitions.

(a) A game Γ is given by a set X of positions (vertices of a tree).

(Remark 4: given a position $x \varepsilon X$, there must be a way of telling which are the next positions, x', obtainable from x on the next move, where:

x': immediate successor of x

x: immediate predecessor of x').

(b). $x_0 \ \varepsilon \ X$ initial position.

(c). Definition: Immediate predecessor function f, such that:

(i). $f:\{X-x_0\} \rightarrow X$

(ii). $\forall x \in X, \exists n \in \aleph \ (\geq 0)$, s.t $f^n(x) = x_0$

(d). A play: A sequence {s(0), s(1), ---} is a play if:

(i). $s(0) = x_0$

(ii). $\forall\ i \geq 0$, s(i)=f [s(i+1)] and,

S: set of plays of Γ.

(e). X and S are partitioned into two mutually exclusive sets:

(i). $X = X_I \cup X_{II}$

where: X_I (X_{II}) is the set of positions from which player I (II) has the right to move;

(ii). $S = S_I \cup S_{II}$

where: S_I (S_{II}) are the sets of plays in which player I (II) wins a game;

(Win-Lose Game) Definitions (additional):

A win-lose game Γ is a collection of objects:

$\Gamma = (x_0, X, X_I, X_{II}, f, S, S_I, S_{II}, \Phi)$

such that:

(i). $x_0 \varepsilon X$

(ii). $X_I \cap X_{II} = \phi$; $X_I \cap X_{II} = X$

(iii). f is an immediate predecessor function

(iv). S satisfies {(d) - (i) and (ii) above}

(v). $S_I \cap S_{II} = \phi$ and $S_I \cap S_{II} = S$

(vi). Φ is a real-valued function on S (payoff function).

A play of Γ is, therefore, an element of S and:

(vii). A play is a win for player I if it belongs to S_I

(viii). A play is a win for player II if it belongs to S_{II}.

Definition (strategy):

Given a game P, the set of strategies (rules, procedures) for player I is denoted by Σ_I, the set of all functions σ with domain X_I such that:

$$\sigma(x) = f^1 (x)$$

Similarly, for player II (with Σ_{II} and τ replacing Σ_I and σ and X_{II} replacing X_I, respectively).

(ε) Thus, to every pair of strategies, σ and τ, $\exists$ a unique play, s, of the game Γ; denote this as:

$$s = < \sigma, \tau >$$

i.e., this is the play which results when players choose strategies σ and τ respectively. This play satisfies the following inductive definitions:

(a). $s(o) = x_o$

(b). If $s(n) \in X_I$ then $s(n+1) = \sigma(s(n))$;

If $s(n) \in X_{II}$ then $s(n+1) = \tau(s(n)$

Definition of a Strictly Determined Winning strategy:

A strategy (rule, procedure) σ is called a winning strategy for player I if $\tau \in \Sigma_{II}(\Gamma)$ implies $< \sigma, \tau > \in S_I$. Similarly, for player II with corresponding definitions. Thus:

A win-lose game is strictly determined if one of the players has winning strategies.

(c). Rabin's Modifications:

Rabin's concern is the actual playability of a game in *real time*; hence, the first modification would be to do away with the assumption of *infinite plays.*

Rabin's assumptions 1:

(a). Each play terminates with a terminal position;

(b). $\exists$ a 1-1 correspondence between terminal positions and plays;

(c). The set of terminal positions T of Γ is defined by:

$$T = x - f(X - x_0)$$

(d). x is finite or countably infinite;

Remark: It is interesting to note Rabin's motivation for (d):

> "In order to make communication between the players possible, they must have some fixed system of naming or denoting all the different positions of the game. Since any one notational system contains at most a countable number of different symbols we must impose the condition that the set x of positions of Γ is finite or countably infinite"
>
> (ibid, p. 150)

This important constraint is hardly ever considered in standard game theory; indeed not even in ordinary economic analysis where agents routinely face an uncountable infinity of alternatives over which they are able to assign preferences. This basic building block is never, in general, relaxed when agents play, say, the prisoner's dilemma with finite automata or when the algorithmic and computational complexity of Nash equilibria are investigated (cf. f.n., 3, above)..

<u>Rabin's assumptions 2 (Towards Effectiveness):</u>

The following information must be *effectively ascertainable:*

(a). $\forall x \in X$, ***who*** makes the next move;

(b). Given: $x \to x'$ *effective verification* of $f(x') = x$; i.e., effective verification of the validity of a move that is observed.

(c). $\forall x \in X$, whether:

(i). The play terminates at that x;

(ii). if so, who won;

<u>Rabin's assumptions 3:</u>

For X *countably infinite:*

(a). X_I and X_{II} are *recursive sets*;

(b). f is a *computable function* from (X- (0)) to X;

(c). T is recursive; and T_I and T_{II} for S_I and S_{II} are also *recursive sets*;.

<u>Definition: (Actual Games)</u>

$P \equiv \{x_0, X, X_I, X_{II}, f, S, S_I, S_{II}, \Phi\}$ is called an actual game if:

(a). all plays $s \in S$ are finite;

(b). X is finite *or* countably infinite;

Discussion:

(I). If X is finite the game is actually playable. Why? Because the effective instructions to play the game can be given in a finite list which enumerates the sets:

X_I, X_{II}, T, T_I, T_I (recall that these are, now, recursive sets).

(II). If X is countably infinite, too, the game is actually playable. Why? This time, because, the finite list of effective instructions to compute the characteristic functions of X_I, X_{II}, T, T_I, and T_{II} and the computable function, f, can be given.

Whenever the player faces one of the questions (a), (b) or (c) in *"Rabin's assumptions 2 (Torward Effectiveness)"*, an answer can be given

by computing the necessary recursive set or computable function and proceed to play or terminate.

Now, I return to Rabin, again.

Definition: (Effectively) Computable Strategy (Rule):

σ is a(n) (effectively) computable strategy if it is defined by:

$$\sigma'(x) = \sigma(x),\ \forall x \in X_I;$$

and, $$\sigma(x) = 0,\ \forall x \in X - X_I;$$

and is (effectively) computable. Similarly for τ.

Definition: The Actual Effective Game:

For g(x): a computable function; g: N - > N, an actual effective game Γ_g is played as follows:

(i) Player I picks any integer I;

(ii) Player II, knowing I's choice, picks any integer j;

(iii) Player I, knowing i and j, picks an integer k;

(iv) Compute g(k);

(v) if g(k) = i + j, then player I wins the play;

(vi) if g(k) ≠ i + j, then player II wins the play;

And finally:

Rabin's Theorem:

If g(x) enumerates a ***simple set*** G, i.e., the set of values of g(x) is simple, then there is no (effectively) computable winning strategy in Γ_g.

Proof:

Player II has the winning strategies because, given i, II could choose j so that $i + j \notin G$. This is possible because, by the definition of a simple set, *G* is infinite. Therefore, a winning strategy for II is any computable function $\tau(i)$, s.t $\forall i$, $i + \tau(i) \notin G$. In fact, if I picks i and II picks $\tau(I)$, then no matter k player I chooses in the third move, we have: $g(k) \in G$ and $g(k) \neq i + \tau(i) \notin G$. So player II always wins.

Next, assume that player II has the winning strategy $\tau(I)$ and that it is a computable function. Clearly,

$$h(i) = i + \tau(I)$$

is also a computable function[5]. Now, the set of values taken by h(i), call it H, is *recursively enumerable*. H is also infinite. Because $i + \tau(i) \in \overline{G}$, $H \subseteq \overline{G}$. But this means $\overline{G}$ contains an infinite set which is recursively enumerable. This contradicts the assumption that G is simple. Hence $\tau(I)$ cannot be a computable function. Q.E.D.

This whole section may appear arid and antiseptic. But is is never useless to know what the pioneer did - and *why* he did it. More importantly it gave me the chance to describe a research program: on effectivising standard concepts and tools. In the process I was able to introduce important definitions and two techniques customarily used in proving uncomputabilities: *the diagonal argument* and existence of *recursively enumerable nonrecursive sets.*

[5] This is the diagonal method.

Jones's version of Rabin's Gale-Stewart game differs, *ostensibly*, only in the way a win is determined; the play, in its basic first three steps, proceeds identically. Thus, the definition of the way the result is determined is:

> *Player I wins if some (2-symbol) i-state **TM** halts in exactly i + k **shifts**[6] when started on a blank input tape. If not, player II wins.*

Now, in a perfectly intuitive sense, this game should be effectively playable. Why? Because, after I and II have elected i, j and k an impartial umpire should be able to list all possible (2-symbol) i-state ***TMs*** effectively. By a simple counting argument it is easy to show that there are $(4i+4)^{2i}$ such machines. Then, the umpire needs to go down the effective list of $(4i+4)^{2i}$ (2 symbol), i-state ***TMs*** and operate each of them and see if any of them halts after exactly (i+k) shifs. If any does, player I will be declared the winner. If not, player II is the winner.

Moreover, being a (modified) Gale-Stewart game, it is also provable that there is a winning strategy for player II. Even granting all this, it is still remarkable that:

Theorem 1

In Jones's modified Rabin game neither player has an *effective* winning strategy.

Some conceptual machinery must be introduced before the (simple) proof can be given. Jones's original proof uses the formalism and results inherent in the ***busy beaver game***. I will need to outline some of the basics of this game. Tibor Rado's busy beaver game (cf.Rado,1962; Lin and Rado,1965) gives an explicit, intuitively clear, *uncomputable function*. Before defining it, consider, for example, a Rado-type description of a Turing Machine:

---

[6] A definition is given below.

		(scanned symbol)	
		0	1
(current state)	1	1R2	1R0
	2	1L2	0R3
	3	1L3	1L1

Fig 1: A Rado-3-state Turing Machine.

It functions as follows: the entry in the first-row and first-column signifies that the Turing Machine scanning symbol 0, in state 1, writes 1 instead of 0, shifts right one square, and enters state 2.

Now, denote:

$\Lambda(k)$: the set of all k-state Turing Machines which eventually halt, after starting on a blank tape. (note: this # is finite since it is a subset of the # $(4k + 4)^{2k}$);

$\rho(m)$: the number of 1s finally written by a halting Turing Machine m ($\in \Lambda(k)$), which isinitialised on a blank tape; (in fig. 1, $\rho(m) = 5$).

s(m): the number of *shifts* made by Turing Machine m ($\in \Lambda(k)$) before halting (having written $\rho(m)$); (in fig. 1, $s(m) = 21$).

$\Sigma(k)$: the k-th *busy beaver number*, defined by:

$$\Sigma(k) = \max \{\rho(m) \; m \in \Lambda(k)\}$$

S(k): the k-th *busy beaver shift number* defined by:

$$S(k) = \max \{s(m) \mid m \in \Lambda(k)\}$$

<u>Definition: Busy Beaver Game</u>

Suppose we consider the set $\Lambda(k)$. Then the winner of a busy beaver game is that Turing Machine, $m \in \Lambda(k)$, which generates $\Sigma(k)$.

Definition: Busy Beaver Problem.

Find an effective procedure to determine Σ(k).

Theorem 2:

For every computable function f, ∃n, (depending on f) such that Σ(k) > f(k), ∀k > n.

Proof

See Rado (1962) or, for a pedagogically masterful 'constructive' proof, also Boolos and Jeffrey (1989, ch. 4, pp. 34-42).

An immediate

Corollory 1

The busy beaver shift number, S(k), is uncomputable.

Proof (sketch)

If S(k) were computable then Σ(k) could be computed by a straight-forward counting procedure. Simply run each Turing Machine m (∈ Λ(k)) through a maximum of S(k) shifts; tick off the 1s on those Turing Machines (∈ Λ(k)) that halt and choose the one that scores the highest ρ(m).

Remark

Just to get a feel for the orders of magnitude note the following:

(a). There are $[4(2+1)]^4$ = Λ(2), 2-state machines, which is a manageable 20,730; Λ(3), on the other hand, is $[4(3+1)]^6$ = 16,777,216!!

(b). As for the busy beaver number, Σ(k), we have Σ(1) = 1; Σ(2) = 4; Σ(3) = 0; Σ(4) = 13; and then we reach monstrosities: Σ(5) ≥ 4098, ..., Σ(7) ≥ 22,961 and Σ(8) ≥ $3.(7.3^{92} - 1)/2$.

(c). The busy beaver shift number, as can be expected, is properly 'Malthusian':

$S(3) = 21$ (cf. above figure 1); $S(4) = 107$; $S(5) \geq 2{,}358{,}064$.

Note, above all, that these are 2-symbol machines. For example S(2) for a 3-symbol Turing Machine is ≥ 38; and for a 4-symbol Turing Machine it is ≥ 7,195. So far as I know S(k), for $k \geq 3$ and more than 2 symbols, has not ever been seriously calculated.

I can now proceed to give an elementary proof of Theorem 1:

Proof (of *the nonexistence of a computable winning strategy in Jones's variation of Rabin's game*).

Assume that the Turing Machine $m \varepsilon \Lambda(k)$ is initialised in state 1 on a blank tape. Clearly $\Sigma(k) \leq S(k)$. Now player II's strategy would be to try to find an ℓ, as a computable function of k, say $f(k) = \ell$ so that the 1st player's 3rd step is thwarted for any choice of k; i.e., ℓ must be chosen so that $f(k) \geq S(k)$. Thus, there is a winning strategy, S(k), for the 2nd player. However, by theorems 2 and 3, $f(k)<\Sigma(k) = S(k)$. Hence there is no *computable* winning strategy for the 2nd player. Q.E.D.

Remark

The perceptive reader would have noticed that both, in Rabin's game and in Jones's variation of it, there are extractable common elements; the most important of which is:

(a). The player's *alternate* in their moves, to choose;

(b). These alternations can be modelled by alternating *existential* and *universal* quantifiers.

(c). The existential quantifer moves first; if the total number of moves are odd, then, an existential quantifier again fixes the choice of the last integer; if not - i.e., if the total number of moves are even - then the universal quantifier determines the value of the last chosen integer.

(d). One of the players tries to make an expression preceded by these alternating quantifiers TRUE; the other to make it FALSE.

Now, it can immediately be seen why Rabin-type games will be able to exploit the methods and results that were developed to solve Hilbert's 10th problem (negatively) - especially when the expression to be *satisfied*, which comes after the alternating quantifiers, can be cast in *conjunctive normal form*. Recall that Rabin's G is a recursively enumerable set (that is not recursive). Then determining whether i+j belongs to such a set is equivalent to determining whether there is an effective procedure to solve an associated diophantine equation. The negative answer to the latter can then be used to show that there is no computable wining strategy in the game.We will have occasion to return to these heuristic observations in more formal ways in the next and the concluding sections.

§3. Arithmetical Games

Gödel's celebrated result was that there were *arithmetical* problems that were effectively undecidable (or recursively unsolvable). Roughly speaking what he showed was that there exists arithmetical equations, say linking two polynomials, preceded by some finite sequence of existential and universal quantifiers, that are effectively undecidable. Now reflect on the way the moves are made in Rabin's effectivized Gale-Stewart game; the moves 'say': "there exists an integer i, for all possible choices of j, such that there is a g(.)=i+j .....". In other words, a form such as: $\exists i \forall j \exists g(.) \{ \ldots \}$. Inside the brackets the win condition will be stated as a proposition to be *satisfied.* If we can extract an arithmetical form for the win condition we can exploit the full force of the results coming down from Gödel and culminating in Matiyasevic's magisterial demonstration that Hilbert's 10th problem is recursively unsolvable. It is here that we will exploit the fundamental result, i.e., the diophantine nature of recursively enumerable sets. When Rabin stipulated that g enumerates a simple set he was, essentially, stating the win condition in the form of requiring a diophantine equation to be recrsively solvable - an impossibility. This is the potted background to the work initiated by Jones, in a series of important papers, building on the suggestions made by Rabin. This whole line of thought has been almost totally ignored, as I mentioned in the introduction, by traditional game theorists who have attempted *ad-hoc* effectivizing in special classes of games.

Definition: Arithmetical Game of Length q

An arithmetical game of length q between 2 players, labelled I and II, in a game of perfect information and *alternating* choices of integers with, say, player I choosing the 1st and the final q-th integers (i.e., q is odd); I wins iff $P(x_1, x_2, ..., x_q) = 0$, where:

$P(x_1, x_2, ..., x_q)$: a polynomial with integer coefficients;

x_i, $\forall_i = 1, ..., q$: the integers chosen, alternately, by I and II.

q: length of game ≡ total number of 'moves' - i.e., total number of alternate choices made by I and II;

Remark

Note, in accordance with the previous remark, the choices alternate. Indeed, I will be associated with the existential quantifier; II with the universal quantifier. Essentially, what happens is the following: I tries to find an x, $(\exists x_1)$ such that for any choice of x_2 $(\forall x_2)$ by II, I can find an x_3 $(\exists x_3)$ such that .... till, finally, an expression (in the case of arithmetical games, then, the expression is a polynomial) is to be satisfied by I (whilst II tries to make sure it is not satisfied). This, as I pointed out above, is exactly as in Rabin's effectivized Gale-Stewart game. From this it follows immediately that:

Theorem 3 Arithmetical Games are Determined (Jones, p. 64, 1982).

In every arithmetical game either I or II has a winning strategy; either I can ensure that the polynomial is satisfied or II can falsify it.

Proof

Note that the strategy of I or II is a series of alternating quantifiers beginning with the existential one for I and the universal one for II. For example, in a game of length q, with q an odd number, I's strategy would look as follows[7]:

$$\exists x, \forall x_2 \exists x_3 \ldots\ldots\ldots \forall x_{q-1} \exists x_q [P(x_1, \ldots\ldots\ldots\ldots\ldots\ldots\ldots x_q) = 0].$$

Player II's strategy would, of course, be a negation of the above. Then, using the law of the excluded middle[8], one or the other of the forms will be true; i.e., I or II has a winning strategy. Q.E.D.

It is, surely, intuitively obvious that one of the players has a winning strategy. Two obvious questions emerge from the result of the theorem because to prove existence is one thing; to actually implement it is quite another thing. Hence the queries are:

(a). Is the winning strategy <u>easy</u> to implement, i.e., is it tractable?

(b). Even more fundamentally, is it computable?

I will postpone a discussion of the first question to the next section and concentrate on various approaches towards an answer to the second. For this I need two important results derived by Jones (1978, 1981). The first is theorem 2 in Jones (1978, p. 336).

<u>Theorem 4</u>

The recursively enumerable sets $W_1, W_2, \ldots\ldots\ldots\ldots, W_n, \ldots\ldots\ldots\ldots$, can be represented in the *prenex normal form*:

$$x \in W_n \Leftrightarrow \exists ab \forall i \exists swpq \forall jv \exists eg$$

$$\{n + s + 1 - I\}\{\{(s + w)^2 + 3w + s - 2I\}^2 + \{[(j - w)^2 + (v - q)^2]$$
$$[(j - s)^2 + (v - p)^2 ((i - n)^2 + (v - q - x)^2)] [(j - 3I)^2 + (v - p - q)^2]$$
$$[(j - 3i - 1)^2 + (v - pq)^2] - e - 1\}^2 \{[v + e + ejb - a]^2 + [v + g - jb]^2\}\} = 0 \ldots\ldots\ldots (1)$$

The second is lemma 2, p. 407 in Jones (1981):

<u>Theorem 5</u>

[7] Note that this is in *prenex normal form*.

[8] Pace the constructivists; but, of course, these are games of <u>finite</u> length.

For every recursively enumerable set W, there is a polynomial with integer coefficients given by Q (n, x_1 , x_2 , x_3 , x_4) such that, for all natural numbers n:

$$n \varepsilon W \Leftrightarrow \exists x_1 , \forall x_2 , \exists x_3 , \forall x_4 \; [Q (n, x_1 , x_2 , x_3 , x_4) \neq 0] \; .... \; (2)$$

These elegant results are direct, but ingenious, outcomes of methods and results, as mentioned above, developed for resolving Hilbert's 10th problem.
An immediate application of theorem 1 would be to consider a Rabin-type game of length q = 6. This is because we can put x_1, for I's first choice; x_2 for II's choice and the winning condition would be exactly as in Rabin's game: I wins if $x_1+x_2 \in G$. Then analogously to Rabin's proof, II has a winning strategy, but has no computable winning strategy in a game of length q = 6 associated with a polynomial Q (n, x_1 , x_2 , x_3 , x_4) such that for any choice of x_1 and x_2 by I and II, respectively, we have (putting n = x_1+x_2 in theorem 1).

$$x_1 +x_2 \in G \Leftrightarrow \exists x_3 \forall x_4 \exists x_5 \forall x_6 \, [Q (x_1+x_2 , x_3 , x_4, x_5, x_6) \neq 0] \; .... \; (3)$$

This is a recursively undecidable problem and, hence, there is no computable winning strategy in the game.

But, of course, this a a general statement; i.e., without any demonstration of an explicit given polynomial Q. Jones however constructed such an example, which I report, again, in view of the neat way the first of the above two theorems is used in the proof. Jones (1982, pp. 70-1) has shown that the arithmetical game, associated with the following polynomial, has no computable winning strategy for either player.

$$\{n + x_7 + 1 - x_6\} \; \{\{(x_7 + x_9)^2 + 3x_9 + x_7 - 2x_6\}^2 +$$
$$+ \{[(x_{14} - x_9)^2 + (x_{16} - x_{13})^2]$$
$$[(x_{14} - x_7)^2 + (x_{16} - x_{11})^2 ((x_6 - n)^2$$
$$+ (x_{16} - x_{13} - x_1 - x_2)^2)] \; [x_{14} - 3x_6)^2 + (x_{16} - x_{11} - x_{13})^2]$$
$$[(x_{14} - 3x_6 - 1)^2 + (x_{16} - x_{11}x_{13})^2] - x_{17} - 1\}^2 \{[x_{16} + x_{17} + x_{17}x_{14}x_5 - x_3]^2$$

$$+ [x_{16} + x_{19} - x_{14}x_5]^2 \}\}.$$

Since his proof of the above theorem gives the general method I am seeking I will outline it. Consider the above polynomial for a Rabin game: i.e., player I picking i; II picking j; etc. Then relabel, sequentially, the variables in theorem 4 as follows: $i = x_1$; $j= x_2$; $a = x_3$; $b = x_5$; and so on. Replace them in (1) to get the above polynomial. This, then, is an arithmetic game of length 19; but x_4, x_8, x_{10}, x_{12}, x_{15} and x_{18} are dummy variables; hence this is, in essence, an arithmetical game of length 13. Put, for the recursively enumerable set G in the Rabin game, W_n as in theorem 4. Then, the nonexistence of a computable winning strategy for either player follows immediately from the diophantine nature of the above polynomial.

It will be clear that we have exploited results that are based on the diophantine nature of recursively enumerable sets. This means we can easily use these results in conventional (non-Arithmetical) game theory by specifying various relevant functions as appropriate polynomials. The reason I have not done this in this essay is the following. Recall that Rabin first effectivized the whole of the Gale-Stewart game. It is only after that task was accomplished that he investigated the existence of computable winning strategies. To repeat this procedure in more orthodox game theory would necessitate that we go back to basics: essentially reduce rational players to Turing Machines.At that point the games become those played by Turing Machines approximately in the sense investigated by Ann Condon (1989). I refer the reader to this competent work as an excuse for not pursuing a similar exercise in this essay.

It must also be clear that the scepticism I indicated in the opening paragraphs of this essay, regarding the conventional application, originates in reasons related to the remarks in the immediately preceding paragraph. But they are also due to the fact that I believe arithmetical games to be of more relevance to the kind of conflict and cooperative situations economists ought to consider. A vision of an economy as a massively parallel distributed resource allocation system peopled by Turing Machines in conflicting and cooperative situations generating emergent phenomena seems, to me at least, a far more interesting research strategy than pursuing the well-trodden path that appears to have exhausted itself. There are, of course, many who will disagree with this assessment; but there are also some who may agree.

§4. Concluding Notes and Complexity Considerations.

Let me recall, from elementary computational complexity theory, a few standard results.

(a). First of all, the well-known inclusion relation:

$$P \subseteq NP \subset PSPACE.$$

(b). Secondly, that quantified satisfiability (QSAT) is PSPACE-complete.

(c). Thirdly, that every Boolean expression is equivalent to one in conjunctive normal form.

(d). Fourthly, the prenex normal form equivalent with a Boolean expression instead of the polynomial, to theorem 4 is:

Theorem 1a: (Jones, 1978, p. 336, theorem 1).

The r.e. sets, W_1, W_2, ..., may be represented in the form $x \in W_n \Leftrightarrow$

$$\exists ab\ \forall i_{i \leq n} \exists swpq\ \forall jv\ \exists eg\ \{(s + w)^2 + 3w + s = 2i \wedge \langle [j = w \wedge v = q]$$
$$\vee [j = 3i \wedge v = p + q] \vee [j = s \wedge (v = p \vee (I = n \wedge v = q + x))]$$
$$\vee [j = 3i + 1 \wedge v = pq] \rightarrow a = v + e + ejb \wedge v + g = jb\rangle\}.$$

With this background at hand it is easy to see that all the arithmetical games that I have discussed in §3 have the QSAT form and are, hence, PSPACE complete. In particular, Rabin's game is of this category. This goes to show that even through the games are determined in principle, i.e., existence of a winning strategy for one of the players is provable, they are in fact - even in relatively simple cases - intractable. In other words, these arithmetical games would require at least exponential time to determine which player has a winning strategy. This is preliminary to determining whether the winning strategy is recursive! One intuitively illuminating way to formalize the arithmetical games of §3 would be to have them played by alternating Turing Machines. These machines, as the name suggests, allow alternation of existential and universal quantifiers in the execution of a computational process[9]. Thus Jones's modified Rabin game would be an ideal candidate for this reformulation. I leave it for a different exercise (cf. Velupillai, 1995).

Denote, now, the polynomials in theorems 4 and 5 generically as $\mathcal{D}$. Then, recalling the fundamental result that recursively enumerable sets are diophantine, the facts stated in (a)~(d) in the opening paragraphs of this section and the properties of alternating Turing Machines (especially the previous footnote) we can also ask for the <u>diophantine complexity</u> of $\mathcal{D}$ (or W_n or G in Rabin's game) in the following precise sense:

> (i). What is the minimum possible <u>degree</u> with respect to the <u>unknowns</u> of the polynomial $\mathcal{D}$ in the diophantine representation of W_n given in theorem(s) 4 (and 5)?
>
> (ii). What is the minimum possible number of unknowns in $\mathcal{D}$?

For example, in proving that the Arithmetical Game of length 6 (which was a variation of Rabin's game) had no computable winning strategies we used a polynomial with 13 variables and the highest degree was 16. Call the degree condition the <u>order</u> of the set W_n and the condition on the minimum of W_n (cf. Matiyasevic, 1994, chapter 8; and Jones, 1978, pp. 338-9). Then we can define different <u>diophantine complexity</u> measures for W_n, based on combinations of the order and rank conditions for $\mathcal{D}$.

This means we can link up directly with computational complexity measures without going back to quantified Boolean expressions and then coming 'down' through the complexity results for variations on the satisfiability problem.

[9] Recall that in a standard nondeterministic computation one only allows existential quantifiers. Alternating Turing Machines accept exactly the recursively enumerable sets. Hence they are ideal for verifying completeness properties of the Arithmetical Games of the previous section.

Note that in recent years there have been imaginative definitions of nondeterministic diophantine machines in terms of (parametric) diophantine equations like $\mathcal{D}$. Then, its behaviour, for given values of the parameter, is to 'guess' the values of the (integer) variables which will ensure that $\mathcal{D}$=0. It has been shown that nondeterministic diophantine machines are as powerful as nondeterministic Turing Machines. This means, of course, we can also go, fruitfully, in the opposite direction: to give direct number theoretic characterization of standard computational complexity measures. But the obvious next step would be to define *alternating* diophantine machines, in analogy with the alternating Turing Machines. These are issues at the frontiers of applied recursion theory. My aim was to start with a simple example and find a way to get to the frontiers in terms of just this one example: Rabin's effectivized Gale-Stewart game.

Clearly, the way to proceed would be to take a series of traditionally defined, simple, two-person, perfect information games and effectivize them. Next, take the definition of 'solution' and reformulate it, say, in conjunctive normal form - if it is not possible to arithmetise it directly. After that it would be the routine procedure described above: to verify that the game is determined; then, to check that it is effectively playable; finally, to compute its relevant complexities and to see whether it can be simplified, in terms of the complexity measures - computational or diophantine. Eventually, also, to see what type of alternating machines can be simulated to play the effectivized game.

One other point should, perhaps, be made. Alternating Turing Machines are, as mentioned above, genealizations of standard nondeterministic Turing Machines. Now, the feature that distinguishes a nondeterministic Turing Machine from its deterministic counterpart is that transitions are multivalued. This characteristic is, therefore, further generalized in alternating machines in that they can be viewd as simulating a certain type of parallelism in their computing activities. Thus a state in an alternating Turing Machine corresponding to, say, an evaluation at a universal quantifier, will be "cooperating"; i.e., taking into account "all" possible configurations in accordance with "*for all*". Similarly, when the state corresponds to an existential evaluation, it will be in a "conflict" situation; i.e., only one value will be evaluated, and all others ignored. Here I must refer again to the relevance of Condon (op.cit).

This interpretation can be developed in the following direction: to reduce some of the simpler 'many-person', perfect information, win-lose games to two-person games of the same sort. Perhaps we will then travel Convey's path in constructing surreal numbers in the opposite direction.

References

Binmore, K, (1987): "Modelling Rational Players: Pt. I", Economics and Philosophy, Vol. 3, pp. 179-214.

Boolos, G.S., &
Jeffrey, R.C., (1989): Computability and Logic, 3rd edition, Cambridge University Press, Cambridge.

Davis,M, Matijasevic,Y
and Robinson,J (1976):"Hilbert's Tenth Problem. Diophantine Equations: Positive Aspects of a Negative Solution", in: F.E.Browder (ed.), Mathematical Developments Arising from Hilbert Problems, American Mathematical Association, Providence, RI.

Holm, J, (1993): Complexity in Economic Theory - An Automata Theoretical Approach, Lund Economic Studies,No.53, Dept.of Economics, University of Lund.

Jones, J, (1974): "Recursive Undecidability-An Exposition",The American Mathematical Monthly, 81(7): 724-738.

Jones, J, (1978): "Three Universal Representations of Recursively Enumerable Sets", Journal of Symbolic Logic, 43(2): 335-351.

Jones, J (1981): "Classification of Quantifier Prefixes over Diophantine Equations", Zeischrift für Mathematische Logik und Grundlagen der Mathematik,Vol.,27, pp. 403-10.

Condon,A, (1989): Computational Models of Games, The Mit Press, Cambridge, MA.

Lin,S, &
Rado,T (1965): "Computer Studies of Turing Machine Problems", Journal of the Association for Computing Machinery, Vol., 12, No.2, April, pp.196-212.

Matiyasevic,Y, (1994): Hilbert's Tenth Problem, The MIT Press, Cambridge, MA.

Megiddo, N &
Wigderson, A, (1986): "On Play by Means of Computing Machines", Theoretical Aspects of Reasoning About Knowledge, Proceedings of the 1986 Conference, ed. by Joseph Y. Halpern, Morgan Kaufman Publishers, CA.

Rabin, M.O, (1957): "Effective Computability of Winning Strategies", Annals of Mathematics Studies, Vol. 3, No. 39, ed. by M. Dresher, A.W. Tucker and P. Wolfe, Princeton University Press,Princeton,N.J.

Rado, T, (1962): "On Non-Computable Functions", Bell System Technical Journal, pp. 877-884.

Velupillai, K, (1995): Computable Economics: The Fourth Arne Ryde Lectures, Oxford University Press, Oxford (forthcoming).