MUTUAL COOPERATION AND UNILATERAL ALTRUISM IN A ONE-SHOT PRISONER'S DILEMMA
- A COMPUTABILITY APPROACH -

MIKIO NAKAYAMA

Abstract. We consider the one-shot Prisoner's Dilemma played by programs or machines, and show that the mutual cooperation is rather an ordinary event under the bounded rationality expressed by the computability. The *kin recognition player* (KRP, for short) is a program with the ability to recognize the opponent, and cooperate if and only if the opponent is *kin* to itself. We prove the existence of the KRP, and also of altruistic players which unilaterally self-sacrifice to the opponents that are kin to a reference KRP. It turns out that while any KRP is evolutionary stable, the self-sacrificing altruistic player is not.

Keywords: Prisoner's dilemma, computability, self-recognition, kin relation, altruism, self-sacrifice

1. Introduction

The possibility of cooperation in the Prisoner's Dilemma has been well-studied, and now seems to be well-understood. The classical folk theorem describes rational cooperation in a repeated play under appropriate punishment mechanisms; and bounded rationality as modeled by automata or machines with computational constraints can also induce cooperation between players. The bounded rationality approach can be found, for example, in Rubinstein [11], Abreu and Rubinstein [1], Neyman [8], Megiddo and Wigderson [7], and others. Howard [6] also presented machine players, and argued even more drastically that cooperation is possible in the one-shot Prisoner's Dilemma. Tennenholtz [12] has considered the machine program with essentially the same ability to that of Howard [6]. With imperfect information, Harrington [5] also showed that cooperation can be obtained in the one-shot play.

Among others, the argument of Howard [6] is remarkable in that it is based on the logical feasibility of recognizing the opponent players. The *self-recognition player* (the SRP, for short) has the ability to recognize the *type* of the opponent, and cooperates if and only if the opponent is identical to itself. Under the assumption that players are drawn from a program pool and matched to play the Prisoner's Dilemma, such an ability of the SRP leads to mutual cooperation between the same SRPs in the one-shot play.

Department of Economics, Ryutsu Keizai University, 120 Ryugasaki, Ibaragi 301-8555. (e-mail: mnakayama@rku.ac.jp) .

The *secret handshake mutant* due to Robson [10], too, has a similar ability to recognize opponents through signaling.

As is also mentioned in Howard [6], however, a drawback of the SRP would be that it cannot by definition cooperate with an opponent that is different from itself yet behaves identically. In other words, the SRP *cannot* cooperate with its kin, relatives, friends, or fellows, leaving considerable inefficiency in the achievement of mutual cooperation.

In this paper, we shall first extend this ability of cooperation to a wider class of players which might be interpreted as brothers and sisters, a family, relatives and kin, thereby obtaining the cooperation as rather an ordinary behavior in the one-shot Prisoner's Dilemma. Howard [6] discusses several extensions of the SRP, but here we present an extended model via the computability approach. Two players will be called *kin* to each other if they are in a *kin relation* in the sense that they have an *ancestor* in common. This will turn out to be a *recursive equivalence relation*; that is, an equivalence relation that can be decidable by a fixed algorithm in finite steps. We will call the player with the ability to cooperate with its kin *the kin-recognition player* (KRP, for short).

One of the interesting consequences of considering in the computability setting is the logical existence of a highly altruistic player associated with a KRP. This player unilaterally *sacrifices* itself to any opponent that is kin to the KRP, being certainly exploited by the opponent. Such a player, though not a KRP, necessarily exists along with any KRP. Therefore, the altruism may be attributed to the bounded rationality as expressed by the computability.

We then discuss the stability of a KRP and other players in an evolutionary environment. It turns out that while any KRP is evolutionary stable, the altruistic player is not: the unilateral altruism is hard to prevail in a population. This is in accord with the fact that mutual cooperation is more frequently observed compared to unilaterally altruistic behavior in real life situations.

A crucial structure of the KRP is the self-reference that *a* KRP *is a player that recognizes the opponent as a* KRP. Howard [6] presented the SRP by directly constructing an algorithm, both in English and in a programming language, dissolving the self-reference. We will prove the existence of a KRP by a *recursion theorem* in computability theory, which enables us to treat the self-referential property of a KRP.

Finally, we conclude with some remarks.

In Appendix, some of the elements of computability theory necessary for our results is summarized.

2. The Self-Recognition Player

Let us consider the following Prisoner's Dilemma with c denoting cooperation, and d, defection.

	c	d
c	$3,3$	$0,4$
d	$4,0$	$1,1$

A Nash equilibrium is a pair of strategies, each of which is a best reply to the other. Thus, (d,d) is the only Nash equilibrium in this game.

The *self-recognition player* (SRP) introduced by Howard [6] is a strategy that may be interpreted to have acquired, in the evolutionary process, the ability to recognize the opponent and cooperate if and only if the opponent is identical to itself. The *secret handshake mutant* (Robson [10]) would be an example of such players acting to the same effect through mutually recognizable signaling.

Denoting SRP by s, the Prisoner's Dilemma is augmented as follows.

	c	d	s
c	$3,3$	$0,4$	$0,4$
d	$4,0$	$1,1$	$1,1$
s	$4,0$	$1,1$	$3,3$

There are now two Nash equilibria, (d,d) and (s,s). In the evolutionary interpretation, however, only the latter equilibrium can survive the evolutionary process. To see this, let us recall the definition of the *evolutionary stable strategy* (ESS, for short): a strategy x is said to be an ESS if (x,x) is a Nash equilibrium, and if y is also a best reply to x then x is a better reply to y than y is to itself. Thus, the strategy s is the only ESS, and mutual cooperation will prevail in the population.

3. The Kin-Recognition Player

In order to discuss the extension of SRP and its general existence, let us treat SRP in a more rigorous framework.

In this paper, a program is a finite algorithm that computes a unary *partial* function f from $N = \{0,1,2,\dots\}$ to N, i.e., a function $f : D(f) \subseteq N \to N$, where $D(f) = \{x \mid f(x) \text{ is defined}\}$ is the domain of f (see Appendix (1)). Such a function f is called *computable*. Any such program can be coded into a natural number by a fixed coding system, so that N is also the set of the code numbers or *indices* of all such programs, and that there are only countable numbers of computable functions (Appendix (2)).

Let x be now the index of a program computing the function φ_x. Program x is then a player of the Prisoner's Dilemma if the range of the function φ_x is $\{c,d\} \subsetneq N$, where the numbers c and d ($c \neq d$) represent cooperation and defection, respectively.

We will assume that *every program is fed as an input a natural number, the index of the opponent program*. Or, as Binmore [3] metaphorically suggested, every program may have its index labeled on its 'forehead' and have the ability each other to read it. Other abilities such as lying or cheating (e.g., the 'sucker punch mutant' due to Robson [10]) could also be treated in the computability setting, but here we confine ourselves to 'honest players' only.

Program x follows the procedure according to its own instructions: it may decode the input and simulate the behavior of the opponent to determine its output, or may simply ignore it and produce an output, or may produce nothing. Since the function φ_x is partial, φ_x may be undefined for some inputs. The ability of recognition in the Prisoner's Dilemma, however, requires a player to compute a *total* function, i.e., a function with domain N as defined below.

Let us now consider, for each $x \in N$ and $y \in N$, a binary relation $K(x, y)$. We will later define a binary relation that *x and y are kin to each other*. For this purpose, let us introduce a basic property of $K(x, y)$. The relation $K(x, y)$ is said to be *recursive* if there exists a computable function $f(x, y)$ satisfying

$$f(x,y) = \begin{cases} 1 & if\ K(x,y), \\ 0 & if\ \neg K(x,y). \end{cases}$$

Thus, if $K(x, y)$ is recursive, whether or not x and y are in this relation is *decidable* by a finite algorithm (see Appendix (6) and (3)). Then, the following lemma is basic to our results.

Lemma 1. *Let $K(x, y)$ be a recursive relation. Then, there exists a program x such that for all $y \in N$,*

$$\varphi_x(y) = \begin{cases} c & if\ K(x,y) \\ d & if\ \neg K(x,y). \end{cases}$$

Proof. Since the relation $K(x, y)$ is recursive by assumption, the function h defined by

$$h(x,y) = \begin{cases} cf(x,y) & if\ K(x,y) \\ d + f(x,y) & if\ \neg K(x,y) \end{cases}$$

is computable. The *second recursion theorem* then guarantees the existence of a *fixed point*, i.e., an index x such that

$$\varphi_x(y) = h(x,y).$$

(Appendix (8)). Hence, there exists an index x such that

$$\varphi_x(y) = \begin{cases} c & if\ K(x,y) \\ d & if\ \neg K(x,y). \end{cases}$$

□

The recursive relation assures the existence of a player x that cooperates if and only if the opponent y is in the relation $K(x, y)$. Further, the recursiveness of $K(x, y)$ alone provides potentially a wide domain of cooperation in the one-shot Prisoner's Dilemma. Since there are infinitely many recursive relations, the cooperating player x will not be an exception in the environment of machine players. However, the cooperation may not be mutual; and, the opponent y of x with $K(x, y)$ may not be a player of the Prisoner's

Dilemma. To obtain mutual cooperation, therefore, the recursive relation should have an appropriate structure.

By the way, the *self-recognition player* x is one that is given by the following:

Definition 1. *Program x is said to be a self-recognition player* (SRP) *if for all $y \in N$,*

$$\varphi_x(y) = \begin{cases} c & if\ y = x \\ d & if\ y \neq x. \end{cases}$$

The existence of SRPs is a direct consequence of the above lemma, since the equality relation is recursive.

Proposition 1. *There exists a self-recognition player.*

4. The Kin Relation with A Common Ancestor

For each $e \in N$, let us consider the set $I_e = \{z \mid \varphi_z = \varphi_e\}$. This is the set of indices of all programs that compute the same function φ_e, that is, programs that output the same action. Then, it is clear that the relation $y \in I_x$ is an equivalence relation. But, the set I_x is not recursive due to *Rice's Theorem* (Appendix (9)). Intuitively, this can be seen by observing that $y \in I_x$ iff $\varphi_y = \varphi_x$ and that the latter relation is not decidable because the equality of functions cannot be assured in any finite number of steps.

The non-recursiveness of I_e makes it impossible for any member $x \in I_e$ to decide whether the opponent y is also a member of I_e or not, i.e., whether to cooperate or not. Therefore, it is at least necessary to have a set of 'fellow' programs as a recursive set.

For each $e \in N$, therefore, consider the subset I_e^* of I_e; namely,

$$e \in I_e^* \subsetneq I_e.$$

The set I_e^* is intended to mean a set of all *descendants* of e by the following assumptions.

Assumption 1: I_e^* is a recursive set.
Assumption 2: $x \in I_e^*$ implies $e \leq x$.
Assumption 3: $I_x^* \cap I_y^* \neq \emptyset$ implies $I_x^* \subseteq I_y^*$ or $I_y^* \subseteq I_x^*$.

An immediate example of I_e^* satisfying the assumptions 1,2 and 3 would be obtained if the members of I_e^* are programs generated by adding to e any finite number of redundant instructions in a recursive way. Consider, for example, the simple case in which there are two different redundant instructions. Then, the same function is computed by 2^n different programs with n redundant instructions added to e allowing repetitions. Letting I_e^* be the set of all such programs for $n = 1, 2, \ldots$, the set I_e^* can be made recursive by the *Padding Lemma* (e.g., Proposition II.1.6 in Odifreddi [9]).

Or, we may resort to a biological analogy that descendants as living organisms generally have acquired more complexity than the ancestor in the evolutionary process. The greater program-sizes of descendants might, therefore, be viewed as reflecting such complexity of subroutines which are *irrelevant to* the main part playing the Prisoner's Dilemma.

In this way, we may call the members of I^*_e the descendants of e, which is also justified by the following remark.

Remark 1. *For any $x, e \in N$, $x \in I^*_e$ iff $I^*_x \subseteq I^*_e$. That is, x is a descendant of e iff the descendants of x are also the descendants of e.*

This is so because by Assumption 2 we have that $\neg(I^*_e \subsetneq I^*_x)$, since $I^*_e \subsetneq I^*_x$ would lead to the contradiction that $x < e$; and then, Assumption 3 implies that $I^*_x \subseteq I^*_e$. The converse is clear by $x \in I^*_x$. Thus, due to the recursively nested structure, the set I^*_e can be represented as a *tree*.

The singleton set $\{e\}$ is a degenerate example of I^*_e. Here, we allow a slight abuse of the use of the word: any program is a descendant and an ancestor of itself.

We can now define the *kin relation* $K^*(x, y)$ as follows : For all x and y,

$$K^*(x, y) \Leftrightarrow \exists w \geq 0 \textit{ such that } x \in I^*_w \wedge y \in I^*_w \ .$$

The kin relation $K^*(x, y)$ can be read as stating that x and y are *kin* to each other if and only if they have an *ancestor* in common.

Remark 2. *If we take $I^*_e = \{e\}$ for each $e \in N$, then the relation $K^*(x, y)$ reduces to the equality relation $x = y$.*

The relation $K^*(x, y)$ has the desired property as shown below.

Lemma 2. *The kin relation $K^*(x, y)$ is a recursive equivalence relation.*

Proof. First, we show that it is an equivalence relation. It will be enough to check the transitivity. Assume that $K^*(x, y)$ and $K^*(y, z)$. Then, there are w and v such that

$$x \in I^*_w \wedge y \in I^*_w \text{ and } y \in I^*_v \wedge z \in I^*_v.$$

Hence, $y \in I^*_w \cap I^*_v \neq \emptyset$, so that by Assumption 3, $I^*_w \subseteq I^*_v$ or $I^*_v \subseteq I^*_w$. Then, w and v have a common ancestor o; that is, there exists an o such that $I^*_w \subseteq I^*_v \subseteq I^*_o$, or $I^*_v \subseteq I^*_w \subseteq I^*_o$. Hence,

$$x \in I^*_o \wedge z \in I^*_o,$$

which shows that $K^*(x, z)$, i.e., the transitivity.

To show that $K^*(x, y)$ is recursive, first note that $K^*(x, y)$ has a *bounded* search for a number w, that is,

$$K^*(x, y) \quad \Leftrightarrow \quad \exists w \leq z \ \ s.t. \ \ x \in I^*_w \wedge y \in I^*_w,$$

where $z = \min\{x, y\}$. This must be so, because by Assumption 2, $0 \leq w \leq x$ and $0 \leq w \leq y$ whenever the common ancestor w of x and y exists. By Assumption 1, the two relations $x \in I^*_w$ and $y \in I^*_w$ are recursive. Conjunction of two recursive relations is recursive, and a bounded search for a number satisfying a recursive relation again defines a recursive relation (Appendix (5) and (6)). Hence, $K^*(x, y)$ is recursive. □

We are now ready to define the *kin recognition player*.

Definition 2. *Let $K^*(x, y)$ be the kin relation. Then, program x is said to be a kin-recognition player* (KRP) *if for all $y \in N$,*

$$\varphi_x(y) = \begin{cases} c & if\ K^*(x, y) \\ d & if\ \neg K^*(x, y). \end{cases}$$

The kin-recognition player is thus a program that cooperates if and only if the opponent is kin to itself.

Proposition 2. *Under assumptions 1, 2 and 3:*

(1) *There exists a* KRP.
(2) *If x is a* KRP *and $K^*(x, y)$, then y is also a* KRP.

Proof. Existence follows from lemmas 2 and 1. Result 2 follows by the fact that $K^*(x, y)$ is an equivalence relation. □

If x is a KRP, the members of $\{y \mid K^*(x, y)\}$, the equivalence class of x, are all KRPs cooperating with each other. Since $\{y \mid K^*(x, y)\}$ is generally an infinite set, the domain of mutual cooperation is much broader than that of the SRP.

5. Unilateral Altruism

The fact that any KRP x cooperates with y if and only if $K^*(x, y)$, i.e., y is kin to x just implies that x regards the opponent z with $\neg K^*(x, z)$ as a stranger. This is so even if the stranger z computes the same function $\varphi_z = \varphi_x$. In the pool of programs that are strangers to KRP x, there are players of Prisoner's Dilemma that behave in fact strangely. We show that there exists a program sacrificing itself unilaterally to all players kin to x.

Given x, let us define $D(x) := \{y \mid \varphi_y(x) = d\}$. This is the set of programs not cooperating with x.

Definition 3. *Program z is said to be a self-sacrificing player if there is a recursive set $D^* \subsetneq D(z)$ such that $\varphi_z(y) = c\ \forall y \in D^*$*

The self-sacrificing player is a player who cooperates in spite of being *certainly* exploited.

Proposition 3. *Let x be a* KRP. *Then, there exists a self-sacrificing player z such that $\neg K^*(z, x)$ and*

$$\varphi_z(y) = \begin{cases} c & if\ K^*(y, x) \\ d & if\ \neg K^*(y, x) \end{cases}$$

with $D^* = \{y \mid K^*(y, x)\}$.

Proof. Take a KRP x, and consider the set $\{y \mid K^*(y, x)\}$. Then, by construction, we have $\{y \mid K^*(y, x)\} \subsetneq I_x$. The inclusion is proper, since $\{y \mid K^*(y, x)\}$ is a recursive set, whereas I_x is not. Then, there exists z such that

$$z \in I_x \setminus \{y \mid K^*(y, x)\}.$$

Hence, $\varphi_z = \varphi_x$ and $\neg K^*(z, x)$. Moreover, $\varphi_y(z) = d$ for all y with $K^*(y, x)$, since $\neg K^*(z, x)$ is equivalent to $\neg K^*(z, y)$ whenever $K^*(y, x)$. Hence, z is self-sacrificing with $D^* = \{y \mid K^*(y, x)\}$. □

The self-sacrificing player z might be called a *kin-to-x-altruistic* player, and an *x-altruistic* player in the special case where the KRP is just an SRP. The player z self-sacrifices just for any opponent y that is kin to x, but defects otherwise even when the opponent is identical to itself. The altruism is never reciprocal, since by definition the opponent players that are kin-to-x will not cooperate with the player z.

It is somewhat surprising that the very existence of a KRP should entail the existence of such a self-sacrificing, altruistic player. If the rationality of players were *perfect* so that $\{y \mid y \text{ is kin to } x\} = I_x$, then such an altruistic player could not exist at all. In this sense, the altruistic behavior can be ascribed to the bounded rationality as embodied by the computability.

6. Evolutionary Stability

A legitimate question to be posed then would be whether or not such cooperation and the unilateral altruism can prevail in a population. Since any SRP is an ESS, any KRP can be expected to prevail as well, which is in fact the case as shown below.

Let $J \subsetneq N$ be a nonempty subset of players of the Prisoner's Dilemma. We say J is *homogeneous* if there is a number v such that for all $x, y \in J$, the pair (x, y) generates the unique identical payoff v to each. Now, let the Prisoner's Dilemma be played by any $x, y \in N$ drawn from the population.

Definition 4. *Let $J \subsetneq N$. Then, any member of J is said to be a collectively evolutionarily stable strategy* (CESS) *if*

(1) *J is homogeneous.*
(2) *For any $x, y \in J$, (x, y) is a Nash equilibrium.*
(3) *For any $x \in J$, if there exists $z \notin J$ such that z is also a best reply to x, then x is a better reply to z than z is to itself.*

The set J satisfying conditions (1), (2) and (3) is a special case of the *evolutionarily stable set* defined by Thomas [13], and is a straightforward extension of the ESS to a set of strategies yielding a unique identical payoff against any member of the set. The set of kin to x^* for any given KRP x^* is a set of CESSs as can be seen from the following result.

Proposition 4. *Let $K^*(x,y)$ be the kin relation, and let $x \in J_{x^*} = \{y \mid K^*(x^*, y)\}$ for some KRP x^*. Then x is a* CESS.

Proof. It will be sufficient to check condition 3 in the definition of CESS. Let $z \notin J_{x^*}$. Since every member of J_{x^*} defects against z, the payoff to z is at most 1. Hence, z cannot be a best reply to any $x \in J_{x^*}$, and condition 3 is vacuously satisfied. □

As for the stability of the unilaterally altruistic player, the situation is opposite: it will not become dominant, for the altruistic behavior would become more and more hard to take because the matching would tend more and more to be the one defecting each other. In fact, any such kin-to-x-altruistic player z cannot survive the evolutionary process as indicated in the payoff matrix below.

	c	d	x	z
c	$3, 3$	$0, 4$	$0, 4$	$0, 4$
d	$4, 0$	$1, 1$	$1, 1$	$1, 1$
x	$4, 0$	$1, 1$	$3, 3$	$4, 0$
z	$4, 0$	$1, 1$	$0, 4$	$1, 1$

The KRP x is the only ESS in this game, and the kin-to-x-altruistic player z is not an ESS as long as a KRP is in the population.

Thus, while cooperation among a family, relatives and kin can evolve in the population, the altruism would become extinct, which would explain why such a self-sacrificing, unilateral altruism is not so widely observed in real life situations.

7. Concluding Remarks

Assuming players of Prisoner's Dilemma as programs (finite algorithms), we have shown that the self-recognition player (SRP) can be extended to the kin-recognition player (KRP) cooperating with much larger class of opponent players. The kin relation is defined as having an ancestor in common, which led to a recursive equivalence relation guaranteeing the existence of KRPs.

It was shown that a KRP entails the existence of an altruistic player cooperating with any opponent in spite of being certainly exploited. The existence of such a player turned out intrincically dependent upon the bounded rationality in terms of computability as applied in this paper. In the evolutionary interpretation, such an altruism is to be extinct, whereas KRP was shown to be evolutionary stable, which would not contradict the theory of kin selection in evolutionary biology.

Due to the property that the kin relation is an equivalence relation, any KRP does not cooperate across different equivalence classes of KRPs. Since we are concerned with cooperation based on the kin relation, this would be a natural consequence rather than a limitation of KRP: real players not in the *same* kin relation may not necessarily cooperate with each other.

But, can we have any general relation that admits a broader domain of cooperation across different equivalence classes of KRPs? Consider the set

$$I := \left\{ x \in N \;\middle|\; \forall y \in N,\ \varphi_x(y) = \begin{cases} c & if\ K^*(x, y) \\ d & if\ \neg K^*(x, y) \end{cases} \right\}.$$

I is the set of all KRPs; namely, any member $x \in I$ cooperates if and only if the opponent is in the Kin relation with x. If this set were recursive, we could obtain a program x such that

$$\varphi_x(y) = \begin{cases} c & if\ x \in I \wedge y \in I \\ d & if\ x \notin I \vee y \notin I. \end{cases}$$

That is, any member $x \in I$ cooperates with *any* member of I, i.e., with any player in *any* equivalence class of KRPs. But, here again, Rice's Theorem stands in the way: I is a set of indices of programs computing unary functions which constitute a *nonempty proper subset* of all unary computable functions. Hence, the relation $x \in I$ is undecidable.

Rice's Theorem is indeed a source of many negative results in computability theory, though we do not regard the above property as a defect of KRP. Rather, we may conclude that KRP is one of logically *maximal*, as well as behaviorally reasonable extensions of SRP in achieving cooperation in the one-shot Prisoner's Dilemma.

Appendix

Here, some of the elements of computability theory is summarized. For formal treatments, the reader may refer to Cutland [4] , or Odifreddi [9].

(1). Intuitively, a partial function f from $N = \{0, 1, 2, ...\}$ to N is said to be *computable* if there exists a finite algorithm such as a *Turing machine* or a *unlimited register machine* to compute f. The definition is similar for n-ary functions. There are several formalizations of the intuitive concept of *effective computability*, all of which have turned out to be equivalent to the *Turing-machine computability*, giving rise to the well-defined class of all *partial recursive functions*. Thus, the partial recursive functions are considered as the formalization of the functions which are effectively computable in the intuitive sense (*Church's thesis*).

(2). Any algorithm or program computing a unary function is a finite sequence of well-defined instructions. Let $\mathcal{P}$ be a set of all such programs. Then a bijection $\gamma : \mathcal{P} \to G \subset N$ can be defined and is called a *coding* or *Gödel numbering* if γ and γ^{-1} are both computable in the following sense:

a: Given a particular program $P \in \mathcal{P}$, we can effectively find the code number $\gamma(P) \in G$;

b: Given a number $n \in G$, we can effectively find the program $P = \gamma^{-1}(n)$.

There are several established ways to code finite objects. Fixing on one coding, every computable (unary) function appears in the enumeration:

$$\varphi_0,\ \varphi_1,\ \varphi_2,\ \varphi_3,\ \cdots$$

where, for each φ_e, the number e is the *index* (code number) of a program computing the function φ_e. Thus, a natural number can be identified with the program with that number as its index.

(3). An n-ary *relation* or *predicate* $K(x_1, \ldots, x_n)$ is said to be *decidable*, *recursive* or *computable* if its *characteristic function* $c_R(x_1, \ldots, x_n)$ is computable, i.e., if the total function

$$c_R(x_1, \ldots, x_n) = \begin{cases} 1 & \text{if } R(x_1, \ldots, x_n) \\ 0 & \text{if } \neg R(x_1, \ldots, x_n) \end{cases}$$

is computable.

(4). We say that an n-ary relation $Q(x_1, \dots, x_n)$ is *partially decidable* if its *partial characteristic function* $f(x_1, \dots, x_n)$ is computable, i.e., if the partial function

$$f(x_1, \dots, x_n) = \begin{cases} 1 & \text{if } Q(x_1, \dots, x_n), \\ \text{undefined} & \text{if } \neg Q(x_1, \dots, x_n) \end{cases}$$

is computable.

(5). It can be shown that an n-ary relation $Q(x_1, \dots, x_n)$ is partially decidable iff there is a decidable n+1-ary relation $R(x_1, \dots, x_n, y)$ such that

$$Q(x_1, \dots, x_n) \text{ iff } \exists y R(x_1, \dots, x_n, y).$$

The relation in the right-hand side involves the *unbounded search* for a number y satisfying the decidable relation $R(x_1, \dots, x_n, y)$. Checking successively for $y = 0, 1, 2, \dots$ whether or not y satisfies the relation R, the search procedure stops if it finds such a y; otherwise the search goes on for ever.

If the above search procedure is *bounded*, i.e.,

$$Q(x_1, \dots, x_n) \text{ iff } \exists y \leq z R(x_1, \dots, x_n, y),$$

then the relation $Q(x_1, \dots, x_n)$ is said to be decidable, for only a finite number of checking is needed to decide whether or not $R(x_1, \dots, x_n, y)$.

(6) . A subset A of N is said to be *recursive* if the membership relation $x \in A$ is decidable. The set of primes, the set of odd numbers, the set N, the empty set and finite sets are immediate examples of recursive sets. A finite union of recursive sets are also recursive.

(7). A subset A of N is called *recursively enumerable* (r.e. for short) if the membership relation $x \in A$ is partially decidable. Recursive sets are recursively enumerable, since the partial characteristic function for the relation $x \in A$, where A is recursive, can be always obtained by having the computation of the characteristic function for the relation $x \in A$ enter a loop whenever $x \notin A$.

An important r.e. set that is *not recursive* is $\{x \mid \varphi_x(x) \text{ is defined}\}$. The set $I(K^*)$ appeared in Concluding Remarks is not recursively enumerable due to the theorem of Rice and Shapiro (see, Cutland [4, Theorem 7-2.16, p.130]); and the complement $I(K^*)^c$ in N is also not recursively enumerable (Cutland [4, Theorem 7-3.4, p.135]). Hence, the decision problem whether or not y is a KRP is not just undecidable in the same sense as the problem whether or not $\varphi_x(x)$ is defined, but far more difficult than this problem.

(8) **The Second Recursion Theorem**.

> Let f be a 2-ary computable function. Then, there exists an integer e such that $\varphi_e(x) \simeq f(e, x)$.

Here, the symbol $\simeq$ means that respective values of both sides are either undefined or defined with the same value. The number e is called a *fixed point*. When f is a total function, there are infinitely many fixed points (see e.g., Odifreddi [9]). A fixed point e is the index of a program that computes the function *defined by using e itself*; therefore, it is widely useful in showing the existence of programs defined in a *self-referential* way.

This theorem is true for $x = (x_1, ..., x_n)$ and (n+1)-ary computable function f.

(9) **Rice's Theorem** .

> Suppose that B is a nonempty proper subset of all unary computable functions. Then the problem $\varphi_e \in B$ is undecidable.

That is, wheher or not a given function e has a certain non-trivial property is generally undecidable. This theorem is a source of many impossibility results in computability theory.

For example, the set $I_e = \{z \mid \varphi_z = \varphi_e\}$ is not recursive, since the set $\{\varphi_z \mid z \in I_e\}$ is a nonempty proper subset of all unary computable functions.

References

[1] Abreu, D., and A. Rubinstein, ”The Structure of Nash Equilibrium in Repeated Games with Finite Automata,” *Econometrica*, **56** (1988), 1259–1281.

[2] Axelrod, R., *The Evolution of Cooperation*, Basic Books, New York 1984.

[3] Binmore, K. G., ”Modeling Rational Players, Part I,” *Economics and Philosophy*, **3** (1987), 179–214.

[4] Cutland, N. J., *Computability*, Cambridge University Press, Cambridge, 1980.

[5] Harrington, Jr, J.E., ”Cooperation in a One-Shot Prisoner’s Dilemma,” *Games and Economic Behavior*, **8** (1995), 304-377.

[6] Howard, J. V., ”Cooperation in the Prisoner’s Dilemma,” *Theory and Decision*, **24** (1988), 203–213.

[7] Megiddo,N and A.Wigderson, ”On Play by Means of Computing Machines,” in *Theoretical Aspects of Reasoning About Knowledge*, Poceedings of the 1986 Conference, J.Y.Halpern ed. Los Altos:Kaufmann, 1986.

[8] Neyman, A., ”Bounded Complexity Justifies Co-Operation in the Finitely Repeated Prisoner’s Dilemma,” *Economics Letters*, **19** (1985), 227–229.

[9] Odifreddi, P., *Classical Recursion Theory*, North-Holland, Amsterdam, 1992.

[10] Robson, A.J., ”Efficiency in Evolutionary Games: Darwin, Nash, and the Secret Handshake,” *Journal of Theoretical Biology*, **144** (1990), 379–396.

[11] Rubinstein,A., ”Finite Automata Play a Repeated Prisoner’s Dilemma,” *Journal of Economic Theory*, **39** (1986), 83–96.

[12] Tennenholtz, M., ”Program Equilibrium,” *Games and Economic Behavior*, **49** (2004), 363-373.

[13] Thomas, B., ”On Evolutionarily Stable Sets,” *Journal of Mathematical Biology*, **22** (1985), 105-115.