# Computability of simple games: A complete investigation of the sixty-four possibilities

Masahiro Kumabe and H. Reiju Mihara

August 2007

# Computability of simple games: A complete investigation of the sixty-four possibilities

Masahiro Kumabe
Kanagawa Study Center, The University of the Air
2-31-1Ooka, Minami-ku, Yokohama 232-0061, Japan

H. Reiju Mihara*
Graduate School of Management, Kagawa University
Takamatsu 760-8523, Japan

August 2007

**Abstract**

Classify simple games into sixteen "types" in terms of the four conventional axioms: monotonicity, properness, strongness, and nonweakness. Further classify them into sixty-four classes in terms of finiteness (existence of a finite carrier) and algorithmic computability. For each such class, we either show that it is empty or give an example of a game belonging to it. We observe that if a type contains an infinite game, then it contains both computable infinite games and noncomputable ones. This strongly suggests that computability is logically, as well as conceptually, unrelated to the conventional axioms.

*Journal of Economic Literature* Classifications: C71, C69, D71, D90.

*Keywords:* Voting games, axiomatic method, complete independence, Turing computability, legal precedents.

*Corresponding author.
*URL:* http://econpapers.repec.org/RAS/pmi193.htm (H.R. Mihara).

# 1 Introduction

Rules can be classified into two classes according to a given axiom (also called a property or a condition): those satisfying the axiom and those violating it. If there are four axioms, rules can be classified into sixteen ($2^4$) classes. If some of these sixteen classes are empty, it means that the conjunction of some of these axioms implies the disjunction of the others (for example, if there does *not* exist a rule satisfying Axioms 1 and 2 but violating Axioms 3 and 4, then Axioms 1 and 2 together imply Axioms 3 or 4). If, on the other hand, each of the sixteen classes is nonempty, the four axioms are "completely independent" in the sense that there is nothing inconsistent about any combination of their truth values. This is exactly what May (1953) investigated after proposing (May, 1952) four "independent" axioms characterizing simple majority rule: he showed that the four axioms are "completely independent" in the sense above.[1]

We would like to do the same for *(simple) games* (voting games). These are the coalitional games that assign either 0 or 1 to each coalition—those assigned 1 are winning coalitions and those assigned 0 are losing coalitions. We consider *six axioms* for simple games. Four of them are conventional: *monotonicity*, *properness*, *strongness*, and *nonweakness*. Another axiom is *finiteness* (existence of a finite carrier), which distinguishes finite games (those ignoring all except fixed, finitely many players) from infinite ones. The other axiom is "*computability*," which is the focus of this paper.

We start with the four conventional axioms. These axioms classify simple games into sixteen classes, which we call *(conventional) types*. Unfortunately, these axioms are not "completely independent" (though they are "independent"). For example, it is well known that there exist no weak, nonproper games. We therefore cannot hope for the "complete independence" of our six axioms.

Let us start over. Our focus is *computability*, an axiom that we introduce and characterize in a companion paper (Kumabe and Mihara, 2007a). We are interested in the relation of this new axiom to the four existing ones. In other words, we want to investigate how computability restricts the games of each type (some of which are finite and the others are infinite). This suggests the following requirements for independence: If a particular type is empty (i.e., it contains no games), we require nothing—it is not computability or lack of it that is responsible for nonexistence of games of that type. If

[1] A set of consistent (compatible) axioms are "independent" (May, 1952) if for each axiom, there are rules violating the axiom but satisfying the others. As far as the axioms in a characterization are concerned, applying this notion of "independence" is the prevalent practice in the axiomatic method today, as discussed in Thomson (2001). Thomson states that an axiomatic study should offer among other things an analysis of logical relations between axioms. We have chosen to give a complete—rather than partial—analysis, in the spirit of earlier theorists like May (1953) and Arrow (1963, footnote 27, page 102).

a particular type is nonempty, however, it should contain both computable games and noncomputable ones. Put differently, we say that "computability is *independent* of the four axioms (within a class of games)" if for each of the sixteen types, there is a computable game of that type (in that class) if and only if there is a noncomputable game of that type (in that class).

One of our main findings is (Proposition 1) that *computability is independent of the four conventional axioms within the class of* infinite *games*. (The analogue of Proposition 1 does not hold for the class of *finite* games. This is because all finite games are computable.) In fact, we come close to saying that computability is independent of the four conventional axioms (within the class of *all* games). The conditions for the independence are satisfied for fifteen out of the sixteen types. The only exception is the type consisting exclusively of dictatorial (hence computable) games. This strongly suggests that computability is logically, as well as conceptually, unrelated to the conventional axioms.[2] In other words, as far as compatibility with the conventional axioms are concerned, computability is almost nonrestrictive.

We make these findings through complete investigation of the sixty-four ($2^6$) classes with respect to the six axioms. The results of the investigation is summarized in Table 1 in Section 3.

One can make other interesting observations from Table 1. In fact, *the table exhausts all the possible relations among the six axioms*. We discuss in Section 3 observations involving several entries of Table 1. Proposition 1 is an example, since it involves the last two columns of the table. Observations involving a single entry also provide useful information. For example, the last entry on the line corresponding to Type 1 indicates that *one can find a computable infinite game without sacrificing the conventional axioms*, which are voting-theoretically desirable. The companion paper (Kumabe and Mihara, 2007a) only exhibit *non*computable games satisfying the axioms and computable games *not* necessarily satisfying the axioms.

The rest of the Introduction gives a background briefly. The companion paper (Kumabe and Mihara, 2007a) gives further discussion.

One can think of simple games as representing voting methods or multicriterion decision rules. They have been central to the study of social choice (e.g., Peleg, 2002). For this reason, the paper can be viewed as a contribution to the foundations of *computability analysis of social choice*, which studies algorithmic properties of social decision-making.[3]

The importance of computability in social choice theory would be unarguable. First, the use of the language by social choice theorists suggests the importance: for example, Arrow (1963) uses words such as "*process or rule*"

[2]What is behind this terminology is the discussion of logical and conceptual independence by Thomson (2001). We do not define "conceptual independence" mathematically.

[3]This literature includes Kelly (1988), Lewis (1988), Bartholdi et al. (1989a,b), Mihara (1997, 1999, 2004), and Kumabe and Mihara (2007a,b).

or "*procedure*." Second, there is a normative reason: computability of social choice rules formalizes the notion of "due process."[4]

We consider an infinite set of "players." Roughly speaking, a simple game is *computable* if there is a Turing program (finite algorithm) that can decide from a description (by integer) of each coalition whether it is winning or losing. Since each member of a coalition should be describable, we assume that the set $N$ of (the names of) players is countable, say, $N = \mathbb{N} = \{0, 1, 2, \ldots\}$. Also, we describe coalitions by a Turing program that can decide for the name of each player whether she is in the coalition. Since each Turing program has its code number (Gödel number), the coalitions describable in this manner are describable by an integer, as desired. (Such coalitions are called *recursive* coalitions.)

Kumabe and Mihara (2007a) give three interpretations of *countably many players*: (i) generations of people extending into the indefinite future, (ii) finitely many *persons* facing countably many *states* of the world (Mihara, 1997), and (iii) attributes or *criteria* in multi-criterion decision-making.

We discuss interpretation (iii) here because of its versatility. Examples of multi-criterion decisions include (a) forming a team to perform a particular task (Kumabe and Mihara, 2007a),[5] (b) granting tenure to junior faculty members at academic institutions (Al-Najjar et al., 2006), and (c) deciding whether a certain act is legal or not. In these examples, there are potentially infinitely many criteria or contingencies on which the decisions can be based.

There is a small literature on incomplete contracts that deals with countably many criteria explicitly (Anderlini and Felli, 1994; Al-Najjar et al., 2006; Krasa and Williams, 2007). The papers in this literature describe a state as a countable sequence of 0's and 1's—no/yes answers to countably many questions. A connection to our framework should be clear.[6] The underlying set of questions in that literature corresponds to the set of "players" in our paper, implying their "state" corresponds to our "coalition." Similarly, their "computable contract" or a "finite contract" roughly corresponds to our "computable game."

While the definition of computability (or the intuition that we provide to explain it) may sound unrealistic, there is an equivalent definition of computability, according to which an algorithm asks up to finitely many

[4]Richter and Wong (1999) give further justifications for studying computability-based economic theories.

[5]This example illustrates that the desirability of the (conventional) axioms depends on the context. Monotonicity makes sense here, but may be too optimistic (adding a member may turn an acceptable team into an unacceptable one). Properness may be irrelevant or even undesirable (ensuring that a given task can be performed by two non-overlapping teams may be important from the viewpoint of reliability). These observations suggest the importance of finding games that violate some of the axioms.

[6]Put in a general way, their goal is also connected to ours. The papers in that literature investigate, like ours, whether certain desirable properties that a complete contract enjoys can be retained if the contract is replaced by a computable or finite one.

questions of the form "Is player $i$ a member of the coalition?" during a computation (Definition 3, Corollary 1). In fact, a result (Theorem 1) that characterizes computable games provides a "finite approximation" of the idealized notion. We explain that now.

Consider the problem of deciding whether a certain act is legal or not. More specifically, we fix a "clause" and ask for each given legal case, whether the clause applies to the case. We identify a case with a state (coalition) and assume that the clause is a computable game. According to the idealized scenario that we give, a case is truthfully and completely reported (by an "inquirer") to the court ("aggregator"), which in turn decides whether the clause applies to the case, by simply following an algorithm.

Real-world court decisions are not like this scenario, even if we ignore incentive issues. First, a complete description of the case is usually unavailable. Second, there is no algorithm in advance. Court decisions are accumulated bit by bit, forming an algorithm (which can give an answer to every case) only in the limit. In the words of a philosopher of law, "The judicial practice of precedent involves deciding new cases by reference to facts about them that are the same as facts about prior cases that were considered by the courts that decided them as grounds for certain legal consequences."[7] This suggests the following more realistic scenario: The complete description of the case is not presented to the court. Instead, the court asks only finitely many questions and obtains answers to them. It makes the final decision based on those finitely many answers, specifying which set of answers has actually counted.

Theorem 1 provides a model of this practice of "legal precedents."[8] Each court decision extracts, given a case, a finite list of criteria (questions and answers), satisfaction of which either determines that the clause applies to the case or determines it does not. If the *process* of these extractions is algorithmic,[9] we have, in the limit, the idealized legal system that can algorithmically decide the applicability of the clause to every case.

[7]Lyons (1984, page 581). We do not discuss the subtle issue raised by Lyons concerning the relation between (i) the requirement that like cases be treated alike and (ii) the requirement that judicial decision follow prior decisions in similar cases.

[8]The economic literature on legal precedents studies important problems that we ignore, such as the incentives for judges to follow precedents (Rasmusen, 1994) and "depreciation" of precedents (Landes and Posner, 1976).

[9]One could justify the requirement that the process be algorithmic as Anderlini and Felli (1994) do: the courts should be ready to present formal arguments to support their decisions.

# 2 Framework

## 2.1 Simple games

Let $N = \mathbb{N} = \{0, 1, 2, \ldots\}$ be a countable set of (the names of) players. Any **recursive** (algorithmically decidable) subset of $N$ is called a **(recursive) coalition**.

Intuitively, a simple game describes in a crude manner the power distribution among *observable* (or describable) coalitions (subsets of players). We assume that only **recursive** coalitions are observable. According to *Church's thesis* (Soare, 1987; Odifreddi, 1992), the recursive coalitions are the sets of players for which there is an algorithm that can decide for the name of each player whether she is in the set.[10] Note that **the class** REC **of recursive coalitions** forms a **Boolean algebra**; that is, it includes $N$ and is closed under union, intersection, and complementation.

Formally, a **(simple) game** is a collection $\omega \subseteq \mathrm{REC}$ of (recursive) coalitions. We will be explicit when we require that $N \in \omega$. The coalitions in $\omega$ are said to be **winning**. The coalitions not in $\omega$ are said to be **losing**. One can regard a simple game as a function from REC to $\{0, 1\}$, assigning the value 1 or 0 to each coalition, depending on whether it is winning or losing.

We introduce from the theory of cooperative games a few basic notions of simple games (Peleg, 2002; Weber, 1994). A simple game $\omega$ is said to be **monotonic** if for all coalitions $S$ and $T$, the conditions $S \in \omega$ and $T \supseteq S$ imply $T \in \omega$. $\omega$ is **proper** if for all recursive coalitions $S$, $S \in \omega$ implies $S^c := N \setminus S \notin \omega$. $\omega$ is **strong** if for all coalitions $S$, $S \notin \omega$ implies $S^c \in \omega$. $\omega$ is **weak** if $\omega = \emptyset$ or the intersection $\bigcap \omega = \bigcap_{S \in \omega} S$ of the winning coalitions is nonempty. The members of $\bigcap \omega$ are called **veto players**; they are the players that belong to all winning coalitions. (The set $\bigcap \omega$ of veto players may or may not be observable.) $\omega$ is **dictatorial** if there exists some $i_0$ (called a **dictator**) in $N$ such that $\omega = \{\, S \in \mathrm{REC} : i_0 \in S \,\}$. Note that a dictator is a veto player, but a veto player is not necessarily a dictator. It is immediate to prove the following well-known lemmas:

**Lemma 1** *If a simple game is weak, it is proper.*

**Lemma 2** *A simple game is dictatorial if and only if it is strong and weak.*

A **carrier** of a simple game $\omega$ is a coalition $S \subset N$ such that

$$T \in \omega \iff S \cap T \in \omega$$

for all coalitions $T$. We observe that if $S$ is a carrier, then so is any coalition $S' \supseteq S$. Slightly abusing the word, we sometimes say a game is **finite** if it has a finite carrier; otherwise, the game is **infinite**.

[10] Soare (1987) and Odifreddi (1992) give a more precise definition of *recursive sets* as well as detailed discussion of recursion theory. Mihara's papers (Mihara, 1997, 1999) contain short reviews of recursion theory.

### 2.2 The computability notion

To define the notion of computability for simple games, we first introduce an indicator for them. In order to do that, we first represent each recursive coalition by a characteristic index ($\Delta_0$-index). Here, a number $e$ is a **characteristic index** for a coalition $S$ if $\varphi_e$ (the partial function computed by the Turing program with code number $e$) is the characteristic function for $S$. Intuitively, a characteristic index for a coalition describes the coalition by a Turing program that can decide its membership. The indicator then assigns the value 0 or 1 to each number representing a coalition, depending on whether the coalition is winning or losing. When a number does not represent a recursive coalition, the value is undefined.

Given a simple game $\omega$, its **$\delta$-indicator** is the partial function $\delta_\omega$ on $\mathbb{N}$ defined by

$$\delta_\omega(e) = \begin{cases} 1 & \text{if } e \text{ is a characteristic index for a recursive set in } \omega, \\ 0 & \text{if } e \text{ is a characteristic index for a recursive set not in } \omega, \\ \uparrow & \text{if } e \text{ is not a characteristic index for any recursive set.} \end{cases}$$

Note that $\delta_\omega$ is well-defined since each $e \in \mathbb{N}$ can be a characteristic index for at most one set.

We now introduce the notion of *($\delta$)-computable* games. We start by giving an intuition. A number (characteristic index) representing a coalition (equivalently, a Turing program that can decide the membership of the coalition) is presented by an inquirer to the aggregator (planner), who will compute whether the coalition is winning or not. The aggregator cannot know a priori which indices will possibly be presented to her. So, *the aggregator should be ready to give an answer whenever a characteristic index for some recursive set is presented to her.* This intuition justifies the following condition of computability.

**Definition 1** A game $\omega$ is **($\delta$)-computable** if $\delta_\omega$ has an extension to a partial recursive function.

Among various notions of computability that one could conceive of, this notion is the only one that we find (Mihara, 2004) defensible.

## 3 Overview of the Results

This section gives a summary of the results in Sections 5–6.

We first classify games according to the conventional axioms (monotonicity, properness, strongness, and nonweakness) and the axioms concerning simplicity (computability and finiteness).

We identify an axiom (property) for games with the class of games satisfying the axiom. Let $A_1$ denote (the class of games satisfying the axiom of)

monotonicity, $A_2$ properness, $A_3$ strongness, and $A_4$ nonweakness. We can classify games into $2^4 = 16$ classes $C_1 \cap C_2 \cap C_3 \cap C_4$ according to these four axioms, where each $C_j$ is either $A_j$ or $A_j^c$ (where $A_j^c$ is the complement of $A_j$; $A_j^c$ is identified with the negation $\neg A_j$ of $A_j$). Each class $C_1 \cap C_2 \cap C_3 \cap C_4$ is represented by a string (binary word) $b_1b_2b_3b_4$ of length four, consisting of $+$'s (1's) and $-$'s (0's), where each $b_j$ is either $+$ or $-$, depending on $C_j = A_j$ or $C_j = A_j^c$. For example, the class $A_1 \cap A_2^c \cap A_3^c \cap A_4$ of monotonic, nonproper, nonstrong, nonweak games are represented by the string $+--+$. The **(conventional) type** of a simple game is (the string $b_1b_2b_3b_4$ representing) the class $C_1 \cap C_2 \cap C_3 \cap C_4$ to which the game belongs. For ease of reference, we give a label (decimal number) to each type, as shown in the first column of Table 1. (The labels are obtained by first identifying the string with a binary number, next converting the binary number into a decimal number, and finally subtracting the decimal number from 16.)

Similarly, we can classify games into $2^2 = 4$ classes according to $\delta$-computability and finiteness (existence of a finite carrier).

Altogether, we can classify games into $16 \times 4 = 64$ classes according to the six axioms. For each such class, we can ask the question whether there exists a simple game in the class. The answers to those questions are given in Sections 5–6. Table 1 summarizes the answers.[11]

We are mainly interested in the relation of computability to the four conventional axioms. What can we observe from Table 1? For example, we can see from row (2), *there is a computable game of type* $(+++-)$*, but not a noncomputable game of the same type.* (In fact, the type consists of dictatorial games.) This means that computability is not "independent of" the four axioms in the sense to be made precise below. *For each of the other fifteen types, however, there is a computable game of that type if and only if there is a noncomputable game of that type.* Hence, we could almost say that computability is "unrelated to" the four axioms. In fact, if we restrict our attention to the infinite games (games without a finite carrier), we can say this:

**Proposition 1** *Axiom C ($\delta$-computability) is independent of the axioms $A_1$ (monotonicity), $A_2$ (properness), $A_3$ (strongness), and $A_4$ (nonweakness) within the class $I$ of infinite games in the sense that for each of the $2^4 = 16$ types $C_1 \cap C_2 \cap C_3 \cap C_4$, the following condition is satisfied:*

$$C \cap I \cap (C_1 \cap C_2 \cap C_3 \cap C_4) \neq \emptyset \iff C^c \cap I \cap (C_1 \cap C_2 \cap C_3 \cap C_4) \neq \emptyset.$$

The exact sense of the statement "computability is independent of the four axioms" mentioned above can be defined by replacing the class $I$ in the

[11] Some of the games constructed in this paper have the property that an empty coalition is winning. However, one can modify all such computable games so that an empty coalition is losing (Kumabe and Mihara, 2007b).

Table 1: Existence of Games in Different Classes

| | With Finite Carrier | | Without Finite Carrier | |
|---|---|---|---|---|
| Types | Non | Computable | Non | Computable |
| $1(++++)$ | no | yes | yes | yes |
| $2(+++-)$ | no | *yes* | *no* | *no* |
| $3(++-+)$ | no | yes | yes | yes |
| $4(++--)$ | no | yes | yes | yes |
| $5(+-++)$ | no | yes | yes | yes |
| $6(+-+-)$ | no | no | no | no |
| $7(+--+)$ | no | yes | yes | yes |
| $8(+---)$ | no | no | no | no |
| $9(-+++)$ | no | yes | yes | yes |
| $10(-++-)$ | no | no | no | no |
| $11(-+-+)$ | no | yes | yes | yes |
| $12(-+--)$ | no | yes | yes | yes |
| $13(--++)$ | no | yes | yes | yes |
| $14(--+-)$ | no | no | no | no |
| $15(---+)$ | no | yes | yes | yes |
| $16(----)$ | no | no | no | no |

The types are defined by the four conventional axioms: monotonicity, properness, strongness, and nonweakness. For example, row (2) indicates that among the monotonic (+), proper (+), strong (+), weak (−, because not nonweak) games, there exist no finite noncomputable ones, there exist finite computable ones, there exist no infinite noncomputable ones, and there exist no infinite computable ones. Note that except for row (2), the last three columns are identical.

statement of the proposition with the class of all simple games. Therefore, computability is not independent of the four axioms if and only if it is *restrictive* in the sense that some nonempty type consists only of computable games or only of noncomputable games.

We leave this section with two interesting observations involving the last three (instead of two as in Proposition 1) columns of the table: From rows (6), (8), (10), (14), (16), we conclude that *if there does not exist a finite computable game of a particular type, then there does not exist a game of that type.* From the other rows except row (2), we conclude that *if there exists an infinite (non)computable game of a particular type, then there exists a finite computable game of that type.*

# 4 Preliminary Results

This section summarizes and slightly extends the results in the companion paper (Kumabe and Mihara, 2007a). It also introduces notation needed in Sections 5–6.

To give a background, we first mention Theorem 1, which characterizes $\delta$-computable simple games in terms of sets of "determining strings." Roughly speaking, in a computable game, finitely many players determine whether a coalition is winning or losing. Though we cannot tell in advance which finite set of players determines that, we can list such sets in an effective manner.

**Notation**. We identify a natural number $k$ with the finite set $\{0, 1, 2, \ldots, k-1\}$, which is an initial segment of $\mathbb{N}$. Given a coalition $S \subseteq N$, we write $S \cap k$ to represent the coalition $\{i \in S : i < k\}$ consisting of the members of $S$ whose name is less than $k$. We call $S \cap k$ the **$k$-initial segment of** $S$, and view it either as a subset of $\mathbb{N}$ or as the string $S[k]$ of length $k$ of 0's and 1's (representing the restriction of its characteristic function to $\{0, 1, 2, \ldots, k-1\}$). $\|$

**Definition 2** Consider a simple game. A string $\tau$ (of 0's and 1's) of length $k \geq 0$ is **winning determining** if any coalition $G \in \mathrm{REC}$ extending $\tau$ (in the sense that $\tau$ is an initial segment of $G$, i.e., $G \cap k = \tau$) is winning; $\tau$ is **losing determining** if any coalition $G \in \mathrm{REC}$ extending $\tau$ is losing. A string is **determining** if it is either winning determining or losing determining. A string is **nondetermining** if it is not determining.

**Theorem 1 (Kumabe and Mihara (2007a))** *A simple game $\omega$ is $\delta$-computable if and only if there are an r.e. set $T_0$ of losing determining strings and an r.e. set $T_1$ of winning determining strings such that (the characteristic function for) any coalition has an initial segment in $T_0$ or in $T_1$.*

Note that $T_0 \cup T_1$ in the theorem does not necessarily contain all determining strings.

It is easy (Kumabe and Mihara, 2007a) to prove from Theorem 1 that *games having a finite carrier are computable* and that *computable games have both finite winning coalitions and cofinite losing coalitions*. As already seen in Section 3, the property of having a finite carrier is an important criterion for classifying games in this paper.

There are notions of computability (Anderlini and Felli, 1994; Weihrauch, 1995) that describe an input as an *infinite* sequence (of 0's and 1's). Since an algorithm must give an output in a finite number of steps, only finite bits from the sequence is used during the computation. This motivates the following redefinition of computability. (One could give the definition for games defined for all sets of players.)

**Definition 3** A game $\omega$ is **computable with initial segments** if there is an algorithm $M$ (Turing machine) such that for any coalition $S$, if $S$ is input to $M$ as an infinite sequence $S(0)S(1)S(2)\ldots$, then $M$ halts with giving the output $\omega(S) \in \{0,1\}$ reading only an initial segment $S[k]$, for some $k$.

Ordinarily, an input to an algorithm is a finite object. We should therefore justify this alternative notion of computability. The following corollary of Theorem 1 justifies it, as long as games are defined for (recursive) coalitions.

**Corollary 1** *A game $\omega$ is $\delta$-computable if and only if it is computable with initial segments.*

*Proof.* To show the "if" direction, suppose $\omega$ is computable with initial segments. Let $e$ be a characteristic index for a coalition $S$. Effectively obtain $S(0) = \varphi_e(0)$, $S(1) = \varphi_e(1)$, $S(2) = \varphi_e(2)$, ... from $e$ and put them into the algorithm. The computation will halt, giving $\omega(S)$.

To show the "only if" direction, suppose $\omega$ is $\delta$-computable. We are given a coalition $S$. Generate the determining strings in $T_0$ and $T_1$ satisfying the conditions of Theorem 1. Wait until an initial segment $S[k]$ of $S$ is generated. If the initial segment is in $T_0$, then $\omega(S) = 0$; if it is in $T_1$, then $\omega(S) = 1$. ∎

Though the proof of Theorem 1 is rather involved, the full force of this theorem is not needed in this paper.

First, *to construct computable games*, we only need the "if" direction, whose proof is straightforward. The following proposition restates the "if" direction of the theorem in a more convenient form:

**Proposition 2** *Let $T_0$ and $T_1$ be recursively enumerable sets of (nonempty) strings such that any coalition has an initial segment in $T_0$ or in $T_1$ but not both. Let $\omega$ be the simple game defined by $S \in \omega$ if and only if $S$ has an initial segment in $T_1$. Then $T_1$ consists only of winning determining strings, $T_0$ consists only of losing determining strings, and $\omega$ is $\delta$-computable.*

*Proof.* The assertion that $T_1$ consists of winning determining strings is obvious from the definition of $\omega$. To see that $T_0$ consists of losing determining strings, let $\alpha \in T_0$. Suppose a coalition $S$ extends $\alpha$ but $S$ is winning. Thus there is a string $\beta \in T_1$ that $S$ extends. This implies that $S$ has an initial segments in both $T_0$ and $T_1$, contrary to the assumption. Computability of $\omega$ is immediate from the "if" direction of Theorem 1 ∎

Second, *to construct noncomputable games*, we only need the following proposition, whose proof is much easier than that of Theorem 1 (it can also be derived as a corollary of Theorem 1). It states that for $\delta$-computable simple games, (the characteristic function for) every coalition $S$ has an initial segment $S \cap k$ that is determining. (The number $k-1$ may be greater than the greatest element, if any, of $S$.) Recall that $S[k]$ is $S \cap k$ viewed as a string:

**Proposition 3 (Kumabe and Mihara (2007a))** *Suppose that a $\delta$-computable simple game is given.* (i) *If a coalition $S$ is winning, then it has an initial segment $S[k]$ (for some $k \in \mathbb{N}$) that is winning determining.* (ii) *If $S$ is losing, then it has an initial segment $S[k]$ that is losing determining.*

We conclude the section with a summary of notation that the reader should take notice of.

**Notation**. For a partial function $f$, $f(\alpha)\downarrow$ means $f(\alpha)$ is defined; $f(\alpha)\uparrow$ means $f(\alpha)$ is undefined. $\varphi_k(\cdot)$ denotes the $k$th partial recursive function of one variable—it is computed by the Turing program with code (Gödel) number $k$.

Let $\alpha$ and $\beta$ be strings (of 0's and 1's).

Then $\alpha^c$ denotes the string of the length $|\alpha|$ such that $\alpha^c(i) = 1 - \alpha(i)$ for each $i < |\alpha|$; for example, $0110100100^c = 1001011011$. Occasionally, a string $\alpha$ is identified with the set $\{i : \alpha(i) = 1\}$. (Note however that $\alpha^c$ is occasionally identified with the set $\{i : \alpha(i) = 0\}$, but never with the set $\{i : \alpha(i) = 1\}^c$.)

$\alpha\beta$ (or $\alpha * \beta$) denotes the concatenation of $\alpha$ followed by $\beta$.

$\alpha \subseteq \beta$ means that $\alpha$ is an initial segment of $\beta$ ($\beta$ extends $\alpha$); $\alpha \subseteq A$ means that $\alpha$ is an initial segment of a set $A$.

Strings $\alpha$ and $\beta$ are **incompatible** if neither $\alpha \subseteq \beta$ nor $\beta \subseteq \alpha$ (i.e., there is $k < \min\{|\alpha|, |\beta|\}$ such that $\alpha(k) \neq \beta(k)$). ‖

# 5 Games with Finite Carriers

We start with the class of finite games (games having a finite carrier). Any game in this class is $\delta$-computable.

In the following, for each of the sixteen conventional types (with respect to monotonicity, properness, strongness, and nonweakness), we either give

an example of a finite game of that type or give a proof that there exists no such game. We give each example by exhibiting finite sets $T_0$ and $T_1$ satisfying the condition of Proposition 2.

1. $(++++)$ A monotonic, proper, strong, nonweak game. Let $T_0 = \{00, 010, 100\}$ and $T_1 = \{11, 011, 101\}$.

2. $(+++-)$ A monotonic, proper, strong, weak game. Let $T_0 = \{0\}$ and $T_1 = \{1\}$. Player 0 is a dictator.

3. $(++-+)$ A monotonic, proper, nonstrong, nonweak game. Let $T_0 = \{00, 010, 0110, 100, 1010\}$ and $T_1 = \{11, 1011, 0111\}$.

4. $(++--)$ A monotonic, proper, nonstrong, weak game. Let $T_0 = \{0, 10\}$ and $T_1 = \{11\}$.

5. $(+-++)$ A monotonic, nonproper, strong, nonweak game. Let $T_0 = \{00\}$ and $T_1 = \{1, 01\}$.

6. $(+-+-)$ A monotonic, nonproper, strong, weak game. By Lemma 1, there is no such game.

7. $(+--+)$ A monotonic, nonproper, nonstrong, nonweak game. Let $T_0 = \{00, 1000, 1001, 0110, 0100\}$ and $T_1 = \{11, 1011, 1010, 0101, 0111\}$.

8. $(+---)$ A monotonic, nonproper, nonstrong, weak game. By Lemma 1, there is no such game.

9. $(-+++)$ A nonmonotonic, proper, strong, nonweak game. Let $T_0 = \{1\}$ and $T_1 = \{0\}$.

10. $(-++-)$ A nonmonotonic, proper, strong, weak game. By Lemma 2, any such game is dictatorial. But any dictatorial game is monotonic. So there is no such game.

11. $(-+-+)$ A nonmonotonic, proper, nonstrong, nonweak game. Let $T_0 = \{1, 01\}$ and $T_1 = \{00\}$.

12. $(-+--)$ A nonmonotonic, proper, nonstrong, weak game. Let $T_0 = \{1, 00\}$ and $T_1 = \{01\}$.

13. $(--++)$ A nonmonotonic, nonproper, strong, nonweak game. Let $T_0 = \{10\}$ and $T_1 = \{0, 11\}$.

14. $(--+-)$ A nonmonotonic, nonproper, strong, weak game. By Lemma 1, there is no such game.

15. $(---+)$ A nonmonotonic, nonproper, nonstrong, nonweak game. Let $T_0 = \{01, 10\}$ and $T_1 = \{00, 11\}$.

16. $(- - --)$ A nonmonotonic, nonproper, nonstrong, weak game. By Lemma 1, there is no such game.

# 6 Games without Finite Carriers

We consider infinite games (games without finite carriers) in this section.

## 6.1 An assortment of noncomputable games

We first give examples of infinite *noncomputable* simple games. We have noted in Section 4 (immediately after Theorem 1) that all *computable* games belong to the class of games that have both finite winning coalitions and cofinite losing coalitions. To show that variety is not lost even if we restrict our games to this class, all the examples are chosen from the class.

Three examples (namely, types 1, 3, and 9) in this section are based on the following lemma. The construction of other examples is simpler.

**Lemma 3** *Let $A$ be a recursive set. Let $T_0$ and $T_1$ be recursively enumerable, nonempty sets of (nonempty) strings such that any coalition has an initial segment in $T_0$ or in $T_1$ but not both. Let $\omega$ be the simple game defined by $S \in \omega$ if and only if either $S = A$ or [$S \neq A^c$ and $S$ has an initial segment in $T_1$]. Then we have the following:*
(i) *$S \notin \omega$ if and only if either $S = A^c$ or [$S \neq A$ and $S$ has an initial segment in $T_0$].*
(ii) *$\omega$ has a finite winning coalition and a cofinite losing coalition.*
(iii) *Suppose further that either $A$ is infinite and has an initial segment in $T_0$ or $A^c$ is infinite and has an initial segment in $T_1$. Then $\omega$ is $\delta$-noncomputable.*

*Proof.* (i) From the definition of $\omega$ and the assumption that any coalition $S$ has a initial segment in $T_0$ or $T_1$ but not both, we have

$$\begin{aligned} S \notin \omega &\iff S \neq A \text{ and } [S = A^c \text{ or } S \text{ has no initial segment in } T_1] \\ &\iff [S \neq A \text{ and } S = A^c] \text{ or} \\ &\quad\ [S \neq A \text{ and } S \text{ has no initial segment in } T_1] \\ &\iff [S = A^c] \text{ or } [S \neq A \text{ and } S \text{ has an initial segment in } T_0]. \end{aligned}$$

(ii) Choose a string $\alpha$ from the nonempty set $T_1$. Let $\beta = \alpha * A(|\alpha|)$. Then $\beta \neq A^c$ since $\beta(|\alpha|) = A(|\alpha|) \neq A^c(|\alpha|)$. Since $\beta$ has the initial segment $\alpha \in T_1$, $\beta \in \omega$ by the definition of $\omega$. We have obtained a finite winning coalition, namely $\beta$. To obtain a cofinite losing coalition, choose $\alpha \in T_0$ and let $\beta = \alpha * A^c(|\alpha|)$. Then by (i), $B := \{i : \beta(i) = 1 \text{ or } \beta(i) \uparrow\}$ is a cofinite losing set.

(iii) Suppose $A$ is infinite and has an initial segment $A[k]$ in $T_0$. Suppose $\omega$ is $\delta$-computable. Then, by Proposition 3, the winning coalition $A$ has an initial segment $A[k']$ that is a winning determining string. Let $\hat{k} = \max\{k, k'\}$. Then on the one hand, $A[\hat{k}]$, which is different from $A$ and has an initial segment in $T_0$, is losing by (i). On the other hand, $A[\hat{k}]$ is winning since it extends the winning determining string $A[k']$. We have obtained a contradiction. The case where $A^c$ is infinite and has an initial segment in $T_1$ is similar. ∎

In the rest of this section, we give, for each of the sixteen conventional types of simple games, an example of an infinite noncomputable game of that type, if such a game exists.

1. $(+ + ++)$ A monotonic, proper, strong, nonweak game. Let $A = \{0\}^c = \{1, 2, 3, \ldots\} \supseteq 0$, $T_1 = \{1\}$, and $T_0 = \{0\}$. The game $\omega$ defined by Lemma 3 satisfies the properties (Kumabe and Mihara, 2007a, Section 6.1).

2. $(+ + +-)$ A monotonic, proper, strong, weak game. By Lemma 2, such a game has a dictator, violating the property that it has no finite carrier.

3. $(+ + -+)$ A monotonic, proper, nonstrong, nonweak game. Let $A = \{0, 1\}^c = \{2, 3, 4, \ldots\} \supseteq 00$, $T_1 = \{11, 1011, 0111\}$, and $T_0 = \{00, 010, 0110, 100, 1010\}$. Define $\omega$ by $S \in \omega$ if and only if either $S = A$ or $[S \neq A^c$ and $S$ has an initial segment in $T_1]$. Then Lemma 3 applies. Note that all the elements of $T_1 \cup T_0$ except 11 and 00 are determining and 111 is winning determining. To show that $\omega$ is monotonic, suppose $S \in \omega$ and $T \supsetneq S$. If $S = A$, then $T \supseteq 1011$ or $T \supseteq 0111$ or $T = N \supseteq 111$, implying that $T \in \omega$. Otherwise, $S \supseteq \alpha$ for some $\alpha \in T_1$. Then, $T \supseteq 1011$ or $T \supseteq 0111$ or $T = N \supseteq 1111$, implying that $T \in \omega$.

   To show that it is proper, suppose $S$, $S^c \in \omega$. If $S = A$, then contradiction is obtained sine $S^c = A^c \neq A$. Otherwise, $S \neq A^c$ and $S[2] = 11$ or $S[4] = 1011$ or $0111$. This contradicts the condition that $S^c$ has an initial segment in $T_0$.

   It is nonstrong since $\{1, 3\} \supseteq 010$ is losing and $\{1, 3\}^c \supseteq 1010$ is losing. It is nonweak since the winning coalitions $\{0, 1, 4\} \supseteq 11001$, $\{0, 2, 3\} \supseteq 1011$, $\{1, 2, 3\} \supseteq 0111$ have an empty intersection.

4. $(++--)$ A monotonic, proper, nonstrong, weak game. Let $A = \{0, 1\}$, $T_1 = \{11\}$, and $T_0 = \{0, 10\}$. Define $\omega$ by $S \in \omega$ if and only if $S \neq A$ and $S$ has an initial segment in $T_1$. Then, we have $S \notin \omega$ if and only if $S = A$ or $S$ has an initial segment in $T_0$. To show that $\omega$ is noncomputable, suppose it is not. Then, by Proposition 3, the losing coalition

$A$ has an initial segment $A[k]$ that is losing determining, where $k \geq 2$ without loss of generality. But $A[k] * 1$ is winning, contradiction. The proofs of the remaining properties are easy.

5. $(+ - ++)$ A monotonic, nonproper, strong, nonweak game. Let $A = \{0,1\}^c = \{2,3,4,\ldots\}$, $T_1 = \{1, 01\}$ and $T_0 = \{00\}$. Define $\omega$ by $S \in \omega$ if and only if $S = A$ or $S$ has an initial segment in $T_1$. Then, we have $S \notin \omega$ if and only if $S \neq A$ and $S$ has an initial segment in $T_0$. The proofs of all the properties are easy.

6. $(+ - +-)$ A monotonic, nonproper, strong, weak game. There is no such game.

7. $(+ - -+)$ A monotonic, nonproper, nonstrong, nonweak game. Let $A = \{0,1\}^c = \{2,3,4,\ldots\}$, $T_1 = \{11, 1011, 1010, 0101, 0111\}$, and $T_0 = \{00, 1000, 1001, 0110, 0100\}$. Define $\omega$ by $S \in \omega$ if and only if $S = A$ or $S$ has an initial segment in $T_1$. The proofs of all the properties are easy.

8. $(+ - --)$ A monotonic, nonproper, nonstrong, weak game. There is no such game.

9. $(- + ++)$ A nonmonotonic, proper, strong, nonweak game. Let $A = N \supseteq 1$, $T_1 = \{0\}$, and $T_0 = \{1\}$. Define $\omega$ by $S \in \omega$ if and only if either $S = A$ or $[S \neq A^c$ and $S$ has an initial segment in $T_1]$. Then Lemma 3 applies. It is nonmonotonic since $\{1\} \supseteq 0 \in \omega$ but $\{0,1\} \supseteq 1 \notin \omega$. To show that it is proper and strong, note the condition equivalent to $S \notin \omega$ given in Lemma 3 (i): $S = A^c$ or $[S \neq A$ and $S$ has an initial segment in $T_0]$. This condition is in turn equivalent to $S^c \in \omega$ since $S$ has an initial segment in $T_0$ if and only if $S^c$ has an initial segment in $T_1$ in this example. It is nonweak since the disjoint sets $\{1\} \supseteq 01$ and $\{2\} \supseteq 001$ are winning.

10. $(- + +-)$ A nonmonotonic, proper, strong, weak game. There is no such game.

11. $(- + -+)$ A nonmonotonic, proper, nonstrong, nonweak game. Let $A = \{0,1\}^c = \{2,3,4,\ldots\}$, $T_1 = \{00\}$, and $T_0 = \{1, 01\}$. Define $\omega$ by $S \in \omega$ if and only if $S \neq A$ and $S$ has an initial segment in $T_1$. The proofs of all the properties are easy.

12. $(- + --)$ A nonmonotonic, proper, nonstrong, weak game. Let $A = \{1\}$, $T_1 = \{01\}$, $T_0 = \{1, 00\}$. Define $\omega$ by $S \in \omega$ if and only if $S \neq A$ and $S$ has an initial segment in $T_1$. The proofs of all the properties are easy.

13. $(--++)$ A nonmonotonic, nonproper, strong, nonweak game. Let $A = \{1\}^c = \{0, 2, 3, 4, \ldots\}$, $T_1 = \{0, 11\}$, and $T_0 = \{10\}$. Define $\omega$ by $S \in \omega$ if and only if $S = A$ or $S$ has an initial segment in $T_1$. The proofs of all the properties are easy.

14. $(--+-)$ A nonmonotonic, nonproper, strong, weak game. There is no such game.

15. $(---+)$ A nonmonotonic, nonproper, nonstrong, nonweak game. Let $A = N$, $T_1 = \{00, 11\}$, and $T_0 = \{01, 10\}$. Define $\omega$ by $S \in \omega$ if and only if $S \neq A$ and $S$ has an initial segment in $T_1$. The proofs of all the properties are easy.

16. $(----)$ A nonmonotonic, nonproper, nonstrong, weak game. There is no such game.

## 6.2 A class of computable, monotonic, proper, strong, nonweak games without finite carriers

In this section, *we construct for each recursive set A, an infinite, computable, monotonic, proper, strong, nonweak simple game* $\omega[A]$. We do so with a view to constructing examples of various types in Section 6.3. For this reason, the construction is long and elaborate.[12]

Our approach is to construct recursively enumerable sets $T_0$ and $T_1$ of strings (of 0's and 1's) satisfying the conditions of Proposition 2. We first construct certain sets $F_s$ of strings for $s \in \{0, 1, 2, \ldots\}$. We then specify an algorithm for enumerating the elements of $T_0$ and $T_1$ using the sets $F_s$, and construct a simple game $\omega[A]$ according to Proposition 2. We conclude that the game is computable by checking (Lemma 10) that $T_0$ and $T_1$ satisfy the conditions of Proposition 2. Finally, we show (Lemmas 12, 13, and 14) that the game satisfies the desired properties.

Before constructing sets $T_0$ and $T_1$ of determining strings, we introduce the notions of p-strings and d-strings. Roughly speaking, a p-string consists of 10's or 01's; A d-string is a concatenation of a p-string followed by 00 or 11. More formally, a string $\alpha$ is a **p-string** if $|\alpha|$ is even and for each $2k < |\alpha|$, we have $\alpha(2k)\alpha(2k+1) \in \{10, 01\}$ (i.e., $\alpha(2k+1) = 1 - \alpha(2k)$). Examples of a p-string include the empty string, 01, 0101, 0110, and 1001011010.

[12] One reason that the construction is complicated is that we construct a *family* of type 1 games $\omega[A]$, one for each recursive set $A$, while requiring *additional conditions* that would become useful for constructing other types of games in Section 6.3. In Kumabe and Mihara (2007b, Appendix A), we construct just one type 1 game, forgetting about the additional conditions. Some aspects of the construction thus become more apparent in that construction. The construction there extends the one (not requiring the game to be of a particular type) in the companion paper (Kumabe and Mihara, 2007a, Section 6.2). The reader might want to consult these papers first.

Note that any substring of even length of a p-string is a p-string. Denote by $\alpha^-$ the substring $\alpha[|\alpha|-1]$ of $\alpha$ with length $|\alpha|-1$. In other words, $\alpha = \alpha^- * \alpha(|\alpha|-1)$. A string $\alpha$ (of even length) is a **d-string** if $\alpha^{--}$ is a p-string and $\alpha(|\alpha|-2)\alpha(|\alpha|-1) \in \{00, 11\}$ (i.e., $\alpha(|\alpha|-2) = \alpha(|\alpha|-1)$). In other words, a d-string $\alpha$ is of the form $\alpha^{--} * 00$ or $\alpha^{--} * 11$ for some p-string $\alpha^{--}$.

**Lemma 4** (i) *Any string of even length either is a p-string or extends a d-string.* (ii) *Any two distinct d-strings $\alpha$ and $\beta$ are incompatible. That is, we have neither $\alpha \subseteq \beta$ nor $\beta \subseteq \alpha$.*

*Proof.* (i) Let $\alpha$ be a string of even length. Find the least $k$ with $2k+1 < |\alpha|$ such that $\alpha(2k)\alpha(2k+1) \in \{00, 11\}$. If there is not such a $k$, then $\alpha$ is a p-string. If there is such a $k$, then the substring $\alpha[2k+2]$ is a d-string.

(ii) Assume $\alpha$ and $\beta$ are different d-strings. If $|\alpha| = |\beta|$, then clearly they are incompatible (since they are different). Otherwise, assume $|\alpha| > |\beta|$ without loss of generality. Since these length are even, we have $|\alpha| - 2 \geq |\beta|$. Suppose $\alpha$ and $\beta$ are compatible. Then $\alpha \supset \beta$ in this case. In fact, the inequality above implies that $\alpha^{--} \supseteq \beta$. But this is impossible since $\alpha^{--}(|\beta|-2)\alpha^{--}(|\beta|-1) \in \{10, 01\}$ (a pair in a p-string) on the one hand, while $\beta(|\beta|-2)\beta(|\beta|-1) \in \{00, 11\}$ (the tail of a d-string) on the other hand. ∎

Let $\{k_s\}_{s=0}^{\infty}$ be an effective listing (recursive enumeration) of the members of the recursively enumerable set $\{k : \varphi_k(2k) \in \{0,1\}\}$, where $\varphi_k(\cdot)$ is the $k$th partial recursive function of one variable. We can assume without loss of generality that $k_0 \geq 1$ and all the elements $k_s$ are distinct. Thus,

$$\mathrm{CRec} \subset \{k : \varphi_k(2k) \in \{0,1\}\} = \{k_0, k_1, k_2, \ldots\},$$

where CRec is the set of characteristic indices for recursive sets.

Let $l_0 = 2k_0 + 2 \geq 4$ and for $s > 0$, let $l_s = \max\{l_{s-1}, 2k_s + 2\}$. Then $\{l_s\}$ is an nondecreasing sequence of even numbers and $l_s > 2k_s + 1$ for each $s$. Note also that $l_s \geq l_{s-1} > 2k_{s-1} + 1$, $l_s \geq l_{s-2} > 2k_{s-2} + 1$, etc. imply that $l_s > 2k_s + 1$, $2k_{s-1} + 1$, $2k_{s-2} + 1$, $\ldots$, $2k_0 + 1$.

For each $s$, let $F_s$ be the finite set of p-strings $\alpha = \alpha(0)\alpha(1)\cdots\alpha(l_s - 1) \supseteq 10$ of length $l_s \geq 4$ such that

(1) $\alpha(2k_s) = \varphi_{k_s}(2k_s)$ and for each $s' < s$, $\alpha(2k_{s'}) = 1 - \varphi_{k_{s'}}(2k_{s'})$.

Note that (1) imposes no constraints on $\alpha(2k)$ for $k \notin \{k_0, k_1, k_2, \ldots, k_s\}$, while it actually imposes constraints for all $k$ in the set, since $|\alpha| = l_s > 2k_s$, $2k_{s-1}$, $2k_{s-2}$, $\ldots$, $2k_0$. We observe that if $\alpha \in F_s \cap F_{s'}$, then $s = s'$. Let $F = \bigcup_s F_s$. Then $F$ is recursive and we have the following:

**Lemma 5** *Any two distinct elements in $F$ are incompatible.*

*Proof.* Let $\alpha$, $\beta \in F$ such that $|\alpha| \leq |\beta|$, without loss of generality. If $\alpha$ and $\beta$ have the same length, then the conclusion follows since otherwise they become identical strings. If $l_s = |\alpha| < |\beta| = l_{s'}$, then $s < s'$ and by (1), $\alpha(2k_s) = \varphi_{k_s}(2k_s)$ on the one hand, but $\beta(2k_s) = 1 - \varphi_{k_s}(2k_s)$ on the other hand. So $\alpha(2k_s) \neq \beta(2k_s)$. ∎

Let $f$ be a recursive bijection from $F$ onto $\mathbb{N}$ ($f$ can be obtained by enumerating the elements of $F$ one by one, assigning 0 to the first element enumerated, 1 to the second element enumerated, and so on). Regarding $f$ as a partial function on the set of strings, we have $f(\alpha) \downarrow$ (i.e., $f(\alpha)$ is defined) if and only if $\alpha \in F$.

**Lemma 6** *Let $\alpha \supseteq 10$ be a p-string of length $l_s$. Then the following statements are equivalent:* (i) *no substring of $\alpha$ is in $F$;* (ii) *for each $s' \leq s$, $\alpha[l_{s'}] \notin F$;* (iii) *for each $s' \leq s$, $f(\alpha[l_{s'}]) \uparrow$;* (iv) *for each $s' \leq s$, $\alpha(2k_{s'}) = 1 - \varphi_{k_{s'}}(2k_{s'})$.*

*Proof.* The definition of $F$ implies that $\alpha \in F$ only if $|\alpha| = l_s$ for some $s$. Hence the equivalence of (i), (ii), and (iii) is immediate. We next show that (ii) and (iv) are equivalent. The direction from (iv) to (ii) is clear from (1). To see the other direction, suppose that (iv) is not the case; we derive the negation of (ii). For some $s' \leq s$, we have $\alpha(2k_{s'}) = \varphi_{k_{s'}}(2k_{s'})$. Choose the least such $s'$. Then ($s' = 0$ or) for any $s'' < s'$, $\alpha(2k_{s''}) = 1 - \varphi_{k_{s''}}(2k_{s''})$. So $\alpha[l_{s'}] \in F_{s'}$ by (1), since $\alpha[l_{s'}] \supseteq 10$ is a p-string of length $l_{s'}$. Thus (ii) is violated. ∎

Let $A$ be a recursive set. The game $\omega[A]$ will be defined via the sets $T_0 := T_0^A$ and $T_1 := T_1^A$ of strings, constructed by enumerating the elements as follows: For each $s$ and $\alpha \in F_s$ (having a length $l_s$ and extending 10),

(2.i) for each p-string $\alpha'$ that is a proper substring of $\alpha$, if $s = 0$ or $|\alpha'| \geq l_{s-1}$, then enumerate $\alpha' * 11$ in $T_1$ and $\alpha' * 00$ in $T_0$;

(2.ii) if $f(\alpha) \in A$, enumerate $\alpha$ in $T_1$; if $f(\alpha) \notin A$, enumerate $\alpha$ in $T_0$ (note that $f(\alpha) \downarrow$ since $\alpha \in F$);

(3) if a string $\beta$ is enumerated in $T_1$ (or in $T_0$) above, then enumerate $\beta^c$ in $T_0$ (or in $T_1$, respectively).

Clearly, $T_0$ *and* $T_1$ *are recursively enumerable* because of this generating algorithm. We observe that the sets $T_0$ and $T_1$ consist of

- d-strings (11, 00, and those extending 10 enumerated at (2.i) and those extending 01 enumerated at (3) via (2.i)) and

- p-strings (those extending 10 enumerated at (2.ii) and those extending 01 enumerated at (3) via (2.ii)).

We also observe that $11 \in T_1$, $00 \in T_0$, $T_0 \cap T_1 = \emptyset$, and $\alpha \in T_0 \Leftrightarrow \alpha^c \in T_1$.

*Define a game $\omega[A]$ by $S \in \omega[A]$ if and only if $S$ has an initial segment in $T_1$.* Lemma 10 establishes computability of $\omega[A]$ (as well as the assertion that $T_0$ consists of losing determining strings and $T_1$ consists of winning determining strings) by way of Proposition 2.

**Lemma 7** *Let $\alpha$, $\beta$ be distinct strings in $T_0 \cup T_1$. Then $\alpha$ and $\beta$ are incompatible. In particular, if $\alpha \in T_0$ and $\beta \in T_1$, then $\alpha$ and $\beta$ are incompatible.*

*Proof.* Obviously, neither $\alpha$ nor $\beta$ is an empty string. Since $T_0$ and $T_1$ consist of p-strings and d-strings, there are three cases to consider:

*Case* (pp): *Both $\alpha$ and $\beta$ are p-strings.* Then either $\alpha$ or $\alpha^c$ is enumerated at (2.ii) of the generating algorithm and so $\alpha \in F$ or $\alpha^c \in F$. Similarly, $\beta \in F$ or $\beta^c \in F$. If $\alpha \in F$ and $\beta \in F$, then $\alpha$ and $\beta$ are incompatible, since any two distinct elements of $F$ are incompatible by Lemma 5. If $\alpha \in F$ and $\beta^c \in F$, then $\alpha \supset 10$ and $\beta \supset 01$, so they are incompatible. The other two subcases are similar.

*Case* (pd): *one of $\alpha$ or $\beta$ is a p-string and the other is a d-string.* Without a loss of generality, $\alpha$ is a p-string and $\beta$ is a d-string. Suppose $\alpha$ and $\beta$ are compatible. Then, $\beta \supset \alpha$. In fact, $\beta^{--} \supseteq \alpha$. As in (pp) above, either $\alpha \in F$ or $\alpha^c \in F$. Also, since either $\beta$ or $\beta^c$ is enumerated at (2.i) of the algorithm, we have either (pd.i) $\beta^{--} \subset \tilde{\beta}$ for some $\tilde{\beta} \in F$ or (pd.ii) $(\beta^c)^{--} \subset \hat{\beta}$ for some $\hat{\beta} \in F$. *Subcase*: $\alpha \in F$ and (pd.i). $\alpha$ and $\tilde{\beta}$ and both in $F$. So they are incompatible by Lemma 5, contradicting the fact that $\alpha \subseteq \beta^{--} \subset \tilde{\beta}$. *Subcase*: $\alpha \in F$ and (pd.ii). Then $\alpha \supseteq 10$ but $\beta \supset 01$, a contradiction. *Subcase*: $\alpha^c \in F$ and (pd.i). Similar to the second subcase. *Subcase*: $\alpha^c \in F$ and (pd.ii). Similar to the first subcase.

*Case* (dd): *Both $\alpha$ and $\beta$ are d-strings.* Immediate from Lemma 4. ∎

*Notation.* We write $f(\beta)\downarrow \in A$ if $f(\beta) \in A$ (which requires $f(\beta)\downarrow$); we write $f(\beta)\downarrow \notin A$ if $f(\beta)\downarrow$ but $f(\beta) \notin A$.

**Lemma 8** *Let $\alpha \supset 1$ be a string of length $l_s$.*

(i) *$\alpha$ extends a string in $T_1$ if and only if* (i.a) *for some $s' \leq s$, $f(\alpha[l_{s'}])\downarrow \in A$ (in this case, $\alpha[l_{s'}] \in T_1$) or* (i.b) *$\alpha$ extends a d-string $\alpha' = (\alpha')^{--} * 11$ such that no substring of $(\alpha')^{--}$ is in $F$ (in this case, $\alpha' \in T_1$).*

(ii) *$\alpha$ extends a string in $T_0$ if and only if* (ii.a) *for some $s' \leq s$, $f(\alpha[l_{s'}])\downarrow \notin A$ (in this case, $\alpha[l_{s'}] \in T_0$) or* (ii.b) *$\alpha$ extends a d-string $\alpha' = (\alpha')^{--} * 00$ such that no substring of $(\alpha')^{--}$ is in $F$ (in this case, $\alpha' \in T_0$).*

(iii) *$\alpha$ does not extend a string in $T_0 \cup T_1$ if and only if $\alpha$ is a p-string and no substring of $\alpha$ is in $F$.*

*Proof.* (i) ($\Longrightarrow$). Assume $\alpha \supseteq 11$. Then (i.b) is satisfied by letting $\alpha' = 11$.

Assume $\alpha \supseteq 10$ extends a string $\alpha' \in T_1$. Suppose first that $\alpha'$ is enumerated in $T_1$ by applying (2.i) of the generating algorithm. (We show (i.b) holds.) Then $\alpha' = (\alpha')^{--} * 11$ and $(\alpha')^{--}$ is properly extended by some element in $F_s$. Since any two different elements in $F$ are incompatible by Lemma 5, no substring of $(\alpha')^{--}$ is in $F$. So (i.b) holds. Suppose next that $\alpha'$ is enumerated in $T_1$ by applying (2.ii). Then $f(\alpha') \in A$. Since $\alpha' = \alpha[l_{s'}]$ for some $s' \leq s$, we obtain (i.a). Finally, the case where $\alpha' \supseteq 10$ is enumerated in $T_1$ by applying (3) is impossible, since every string enumerated at (3) extends 0.

($\Longleftarrow$). Assume $\alpha \supseteq 11$. Since $11 \in T_1$, the left hand side of (i) holds.

Assume $\alpha \supseteq 10$ and either (i.a) or (i.b) holds.

Suppose (i.a) first. By the definition of $f$, $\alpha[l_{s'}] \in F_{s'}$. Since $f(\alpha[l_{s'}]) \in A$, we have $\alpha[l_{s'}] \in T_1$ by (2.ii). So $\alpha$ extends a string in $T_1$.

Suppose (i.b) next: $\alpha$ extends a d-string $\alpha' = (\alpha')^{--} * 11$ such that no substring of $(\alpha')^{--}$ is in $F$. We show that $\alpha'$ is in $T_1$.

Suppose $(\alpha')^{--} \subset \alpha[l_0]$ first. Since $l_0$ is even and $(\alpha')^{--}$ is a p-string of even length $< l_0$, we have $|(\alpha')^{--}| \leq l_0 - 2$. Since $l_0 := 2k_0 + 2$, we can find a p-string $\beta$ of length $l_0$ that is an extension of $(\alpha')^{--}$ such that $\beta(2k_0) = \varphi_{k_0}(2k_0)$. Then $\beta \in F_0$ and by (2.i) (for $\beta$ and $(\alpha')^{--}$ instead of $\alpha$ and $\alpha'$, respectively), $\alpha' = (\alpha')^{--} * 11 \in T_1$.

Otherwise, there is $s''$ such that $0 < s'' \leq s$ and $\alpha[l_{s''-1}] \subseteq (\alpha')^{--} \subset \alpha[l_{s''}]$. Since $\alpha'$ is a d-string, $(\alpha')^{--}$ is a p-string. As $\alpha[l_{s''-1}] \subseteq (\alpha')^{--}$ and no substring of $(\alpha')^{--}$ is in $F$, $\alpha[l_{s''-1}]$ is a p-string of which no substring is in $F$. By Lemma 6, for each $t \leq s'' - 1$, we have $\alpha[l_{s''-1}](2k_t) = 1 - \varphi_{k_t}(2k_t)$.

Since $\alpha[l_{s''-1}] \subseteq (\alpha')^{--} \subset \alpha[l_{s''}]$, we have $l_{s''-1} < l_{s''}$. Hence $l_{s''} := \max\{l_{s''-1}, 2k_{s''} + 2\} = 2k_{s''} + 2$. Since $|(\alpha')^{--}|$ and $l_{s''}$ are even, $|(\alpha')^{--}| \leq 2k_{s''}$. We can find a p-string $\beta$ of length $l_{s''}$ that is an extension of $(\alpha')^{--}$ such that $\beta(2k_{s''}) = \varphi_{k_{s''}}(2k_{s''})$. Therefore, for each $t \leq s'' - 1$, we have $\beta[l_{s''-1}](2k_t) = (\alpha')^{--}[l_{s''-1}](2k_t) = 1 - \varphi_{k_t}(2k_t)$. So $\beta \in F_{s''}$ by (1). Then since $|(\alpha')^{--}| \geq l_{s''-1}$, we have by (2.i) (for $\beta$ and $(\alpha')^{--}$ instead of $\alpha$ and $\alpha'$, respectively), $\alpha' = (\alpha')^{--} * 11 \in T_1$.

(ii) Similar to (i).

(iii) ($\Longrightarrow$). Suppose that $\alpha$ does not extend a string in $T_0 \cup T_1$. Then the negations of (i.a) and of (ii.a) imply for each $t \leq s$, $f(\alpha[l_t]) \uparrow$, which implies by Lemma 6 that no substring of $\alpha$ is in $F$. Furthermore, (since no substring of $\alpha$ is in $F$) the negations of (i.b) and of (ii.b) imply that $\alpha$ does not extend a d-string. By Lemma 4 (i), $\alpha$ is a p-string.

($\Longleftarrow$). Suppose that $\alpha$ is a p-string and no substring of $\alpha$ is in $F$. Since $\alpha$ is a p-string, no substring of $\alpha$ is a d-string. So $\alpha$ does not satisfy (i.b) or

(ii.b). Since no substring $\alpha'$ of $\alpha$ is in $F$, we have for such $\alpha'$, $f(\alpha')\uparrow$. So $\alpha$ does not satisfy (i.a) or (ii.a). Therefore, $\alpha$ does not extend a string in $T_0 \cup T_1$. ∎

**Lemma 9** *Let $\alpha \supset 1$ be a string of length $l_s$ such that $\alpha(2k_s) = \varphi_{k_s}(2k_s)$. Then $\alpha$ extends a string in $T_0 \cup T_1$.*

*Proof.* If $\alpha \supseteq 11$, the conclusion follows immediately, since $11 \in T_1$.

Suppose $\alpha \supseteq 10$. We prove the lemma by induction on $s$. Assume $s = 0$. If $\alpha$ is a p-string, then $\alpha \in F_0$. By (2.ii) of the generating algorithm for $T_0$ and $T_1$, we obtain $\alpha \in T_0 \cup T_1$. Otherwise, by Lemma 4 (i), $\alpha$ extends a d-string $\beta$. Since $|\beta^{--}| < l_0 \leq l_s$ for all $s$, no substring of $\beta^{--}$ is in $F$ (because $F$ consists of certain strings of length $l_s$ for some $s$). By Lemma 8 (i.b) or (ii.b), $\alpha$ extends a string (namely $\beta$) in $T_0 \cup T_1$.

Assume the lemma holds for $s-1$. If for some $s' < s$, $\alpha(2k_{s'}) = \varphi_{k_{s'}}(2k_{s'})$ then by the induction hypothesis, $\alpha[l_{s'}]$ extends a string in $T_0 \cup T_1$. So $\alpha$ extends a string in $T_0 \cup T_1$. Otherwise, for each $s' < s$, $\alpha(2k_{s'}) = 1 - \varphi_{k_{s'}}(2k_{s'})$. If $\alpha$ is a p-string then $\alpha \in F$ by (1), hence it is in $T_0 \cup T_1$ by (2.ii) of the construction. If $\alpha$ is not a p-string then by Lemma 4 (i), $\alpha$ extends a d-string $\beta$. Then $|\beta^{--}| < l_s$. Since $\beta \subseteq \alpha$ and for each $s' < s$, $\alpha(2k_{s'}) = 1 - \varphi_{k_{s'}}(2k_{s'})$, no substring of $\beta^{--}$ is in $F$ by (1). By Lemma 8 (i.b) or (ii.b), $\alpha$ extends a string (namely $\beta$) in $T_0 \cup T_1$. ∎

**Lemma 10** *Any coalition $S \in$ REC has an initial segment in $T_0$ or in $T_1$, but not both.*

*Proof.* We show that $S$ has an initial segment in $T_0 \cup T_1$. Lemma 7 implies that $S$ does not have initial segments in both $T_0$ and $T_1$. (The assertion following "In particular" in Lemma 7 is sufficient for this, but we can actually show the stronger statement that $S$ has exactly one initial segment in $T_0 \cup T_1$.)

If $S \supseteq 1$, suppose $\varphi_k$ is the characteristic function for $S$. Then $k \in \{k_0, k_1, k_2, \ldots\}$ since this set contains the set CRec of characteristic indices. So $k = k_s$ for some $s$. By Lemma 9, the initial segment $S[l_s]$ (i.e., $\varphi_{k_s}[l_s]$) extends a string in $T_0 \cup T_1$. So, $S$ has an initial segment in $T_0 \cup T_1$.

If $S \supseteq 0$, then $S^c \supseteq 1$ has an initial segment in $T_0 \cup T_1$ by the argument above. So, $S$ has an initial segment in $T_1 \cup T_0$. ∎

Next, we show that the game $\omega[A]$ has the desired properties. Before showing monotonicity, we need the following lemma. For strings $\alpha$ and $\beta$ with $|\alpha| \leq |\beta|$, we say $\beta$ *properly contains* $\alpha$ if for each $k < |\alpha|$, $\alpha(k) \leq \beta(k)$ and for some $k' < |\alpha|$, $\alpha(k') < \beta(k')$; we say $\beta$ *is properly contained by* $\alpha$ if for each $k < |\alpha|$, $\beta(k) \leq \alpha(k)$ and for some $k' < |\alpha|$, $\beta(k') < \alpha(k')$.

**Lemma 11** *Let $\alpha$ and $\beta$ be strings such that $l_s = |\alpha| \leq |\beta|$ for some $s$.* (i) *If $\alpha$ extends a string in $T_1$ and $\beta$ properly contains $\alpha$, then $\beta$ extends a string in $T_1$.* (ii) *If $\alpha$ extends a string in $T_0$ and $\beta$ is properly contained by $\alpha$, then $\beta$ extends a string in $T_0$.*

*Proof.* We only prove (i). The proof for (ii) is similar. Suppose that $\alpha$ extends a string in $T_1$ and that $\beta$ properly contains $\alpha$.

*Case* 1: $\alpha \supseteq 1$. In this case, (i.a) or (i.b) of Lemma 8 holds.

*First assume* (i.a) *is the case*: we can choose an $s' \leq s$ such that $f(\alpha[l_{s'}])\downarrow \in A$ (in this case, $\alpha[l_{s'}] \in T_1$). If $\beta$ extends $\alpha[l_{s'}]$, clearly the conclusion holds. Otherwise, since $|\beta| \geq l_s \geq l_{s'}$, $\alpha[l_{s'}]$ and $\beta$ are incompatible; that is, there exists $k < l_{s'}$ such that $\alpha[l_{s'}](k) \neq \beta(k)$. Choose the least such $k$; since $\beta$ properly contains $\alpha$, we have $\alpha[l_{s'}](k) = 0$ and $\beta(k) = 1$. Let $\beta' = \beta[k](= \alpha[k])$. Note that $f(\alpha[l_{s'}]) \downarrow$ implies $\alpha[l_{s'}] \in F$, which in turn implies $\alpha[l_{s'}]$ is a p-string.

Suppose $k$ is even. We will show that $\beta$ extends $\beta' * 11 \in T_1$. Since $k < l_{s'}$ and $l_{s'}$ is also even, we have $k+1 < l_{s'}$, so that $\alpha[l_{s'}](k+1)\downarrow$. Since $\alpha[l_{s'}]$ is a p-string, $\beta(k+1) \geq \alpha[l_{s'}](k+1) = 1 - \alpha[l_{s'}](k) = 1$. So $\beta(k)\beta(k+1) = 11$. Hence $\beta' * 11 \subseteq \beta[l_s]$. Since $\alpha[l_{s'}] \in F$, no proper substring of $\alpha[l_{s'}]$ is in $F$. As $\beta' \subset \alpha[l_{s'}]$, no substring of $\beta'$ is in $F$. So by Lemma 8 (i.b), $\beta[l_s]$ extends a string (namely, $\beta' * 11$) in $T_1$.

Suppose $k$ is odd. We will show that $\beta$ extends $(\beta')^- * 11 \in T_1$. Since $\alpha[l_{s'}]$ is a p-string, $\beta(k-1) = \alpha[l_{s'}](k-1) = 1 - \alpha[l_{s'}](k) = 1$. So $\beta(k-1)\beta(k) = 11$. Hence $(\beta')^- * 11 \subseteq \beta[l_s]$. Since no proper substring of $\alpha[l_{s'}]$ is in $F$ and $(\beta')^- \subset \alpha[l_{s'}]$, no substring of $(\beta')^-$ is in $F$. So by Lemma 8 (i.b), $\beta[l_s]$ extends a string (namely, $(\beta')^- * 11$) in $T_1$.

*Next assume* (i.b) *is the case*: $\alpha$ extends a d-string $\alpha' = (\alpha')^{--} * 11$ such that no substring of $(\alpha')^{--}$ is in $F$ (in this case, $\alpha' \in T_1$). Choose the least $k \leq |\alpha|$ such that $\alpha(k) \neq \beta(k)$; we have $\alpha(k) = 0$ and $\beta(k) = 1$. Let $\beta' = \beta[k](= \alpha[k])$. Since $\alpha'(|\alpha'| - 2) = \alpha'(|\alpha'| - 1) = 1$, either $k > |\alpha'| - 1$ or $k < |\alpha'| - 2 = |(\alpha')^{--}|$. If $k > |\alpha'| - 1$, we get $\beta' \supseteq \alpha'$. This implies $\beta \supseteq \beta' \supseteq \alpha' \in T_1$; hence $\beta$ extends a string in $T_1$. Otherwise, we have $k < l := |(\alpha')^{--}|$ and $\beta' \subset (\alpha')^{--}$.

Suppose $k$ is even. Since $k < l$ and $l$ is also even, we have $k+1 < l$, so that $(\alpha')^{--}(k+1)\downarrow$. Since $\alpha$ is a p-string, $\beta(k+1) \geq (\alpha')^{--}(k+1) = 1 - (\alpha')^{--}(k) = 1$. So $\beta(k)\beta(k+1) = 11$. Hence $\beta' * 11 \subseteq \beta[l_s]$. Since no substring of $(\alpha')^{--}$ is in $F$ and $\beta' \subset (\alpha')^{--}$, no substring of $\beta'$ is in $F$. So by Lemma 8 (i.b), $\beta[l_s]$ extends a string (namely, $\beta' * 11$) in $T_1$.

Suppose $k$ is odd. Since $(\alpha')^{--}$ is a p-string, $\beta(k-1) = (\alpha')^{--}(k-1) = 1 - (\alpha')^{--}(k) = 1$. So $\beta(k-1)\beta(k) = 11$. Hence $(\beta')^- * 11 \leq \beta[l_s]$. Since no substring of $(\alpha')^{--}$ is in $F$ and $(\beta')^- \subset (\alpha')^{--}$, no substring of $(\beta')^-$ is in $F$. So by Lemma 8 (i.b), $\beta[l_s]$ extends a string (namely, $(\beta')^- * 11$) in $T_1$.

*Case* 2: $\alpha \supseteq 0$. First note that assertion (ii) for Case 1 can be proved by an argument similar to the proof of assertion (i) for Case 1 above (use

Lemma 8 (ii) instead of Lemma 8 (i)). By the construction of $T_1$ and $T_0$, $\alpha^c \supseteq 1$ extends a string in $T_0$ and $\beta^c$ is properly contained by $\alpha^c$. Applying assertion (ii) for Case 1, we obtain that $\beta^c$ extends a string in $T_0$. Hence $\beta$ extends a string in $T_1$. ∎

Note that the preceding proof shows that $\beta$ actually extends a *d-string* unless it extends $\alpha[l_{s'}]$.

**Lemma 12** *The game $\omega[A]$ is monotonic.*

*Proof.* Suppose $B \in \omega[A]$ and $B' \supseteq B$. By the definition of $\omega[A]$, $B$ has an initial segment $\alpha \in T_1$. Choose the least $s$ such that $l_s \geq |\alpha|$. Then the initial segment $B[l_s]$ extends $\alpha \in T_1$. Let $\beta = B'[l_s]$. Then either $\beta = B[l_s]$ or $\beta$ properly contains $B[l_s]$.

If $\beta = B[l_s]$, then clearly $\beta$ extends $\alpha \in T_1$ and so does $B'$. Therefore, $B' \in \omega[A]$. Otherwise, $\beta$ properly contains $B[l_s]$, which extends $\alpha \in T_1$. By Lemma 11 (i), $\beta$ extends a string in $T_1$ and so does $B'$. Therefore, $B' \in \omega[A]$. ∎

**Lemma 13** *The game $\omega[A]$ is proper and strong.*

*Proof.* It suffices to show that $S^c \in \omega \Leftrightarrow S \notin \omega$. From the observations that $T_0$ and $T_1$ consist of determining strings and that $\alpha^c \in T_0 \Leftrightarrow \alpha \in T_1$, we have: $S^c \in \omega$ iff $S^c$ has an initial segment in $T_1$ iff $S$ has an initial segment in $T_0$ iff $S \notin \omega$. ∎

**Lemma 14** *The game $\omega[A]$ is nonweak and does not have a finite carrier.*

*Proof.* We construct a set $B$ such that for infinitely many $l$, the $l$-initial segment $B[l]$ has an extension that is winning and an extension that is losing. Let $B \supseteq 10$ be a set such that for each $k_s$, $B(2k_s) = 1 - \varphi_{k_s}(2k_s)$ and any initial segment of $B$ of even length is a p-string. Let $s$ be such that $l_{s+1} > l_s$.

Then $l_{s+1} := \max\{l_s, 2k_{s+1} + 2\} = 2k_{s+1} + 2$ and $2k_{s+1} + 2 > l_s$ implies (since both sides are even numbers) that $2k_{s+1} \geq l_s$. By the definition of $B$, for each $t \leq s$, we have $B(2k_t) = 1 - \varphi_{k_t}(2k_t)$ and $2k_t < l_s$ (the last inequality from the observation that $l_s > 2k_s + 1$, $2k_{s-1} + 1$, $2k_{s-2} + 1$, …, $2k_0 - 1$). Then since $2k_{s+1} \geq l_s$, there is a p-string $\alpha \supseteq B[l_s]$ of length $l_{s+1}$ such that $\alpha(2k_{s+1}) = \varphi_{k_{s+1}}(2k_{s+1})$ and for each $t \leq s$, $\alpha(2k_t) = 1 - \varphi_{k_t}(2k_t)$. Then by (1), $\alpha \in F_{s+1}$ and $|\alpha^{--}| = |\alpha| - 2 = l_{s+1} - 2 = 2k_{s+1} \geq l_s$. So by (2.i) of the generating algorithm, $\alpha^{--} * 11 \in T_1$ and $\alpha^{--} * 00 \in T_0$.

There are infinitely many such $s$. It follows that any initial segment of $B$ has an extension in $T_1$ and an extension in $T_0$. This means that the game has no finite carrier.

To show nonweakness, we give three (winning) coalitions in $T_1$ whose intersection is empty. First, 10 (in fact any initial segment of the coalition $B \supseteq 10$) has extensions $\alpha$ in $T_1$ and $\beta$ in $T_0$ by the argument above. So 01 has the extension $\beta^c$ in $T_1$. Clearly, the intersection of the winning coalitions $11 \in T_1$, $\alpha \supseteq 10$, and $\beta^c \supseteq 01$ is empty. ∎

Note that the proof that $\omega[A]$ has no finite carrier depends on (2.i), but not (2.ii) or (3), of the generating algorithm.

### 6.3 An assortment of computable games without finite carriers

In this section, we give, for each of the sixteen conventional types of simple games, an example of an infinite computable game of that type, if such a game exists. Most of the examples are based on the game $\omega[A]$ in Section 6.2.

1. $(+\,+\,+\,+)$ A monotonic, proper, strong, nonweak game. $\omega[A]$ is such a game.

2. $(+\,+\,+\,-)$ A monotonic, proper, strong, weak game. By Lemma 2, such a game has a dictator, violating the property that it has no finite carrier.

3. $(+\,+\,-\,+)$ A monotonic, proper, nonstrong, nonweak game. Let $\omega = \omega[\emptyset] \cap \omega[\mathbb{N}]$; that is, $S \in \omega$ if and only if $S \in \omega[\emptyset]$ and $S \in \omega[\mathbb{N}]$.

   To show $\omega$ is proper, suppose $S \in \omega$ and $S^c \in \omega$. Then $S \in \omega[\mathbb{N}]$ and $S^c \in \omega[\mathbb{N}]$, contradicting the properness of $\omega[\mathbb{N}]$.

   To show $\omega$ is nonstrong, let $\alpha \in F$. We show that both $\alpha$ and $\alpha^c$ are losing. On the one hand, we have $\alpha \in T_0^{\emptyset}$ by (2.ii) of the generating algorithm. Since $T_0^{\emptyset}$ consists of losing determining strings, $\alpha \notin \omega[\emptyset]$. Hence $\alpha \notin \omega$. On the other hand, we have $\alpha \in T_1^{\mathbb{N}}$ by (2.ii). Hence $\alpha^c \in T_0^{\mathbb{N}}$. Since $T_0^{\mathbb{N}}$ consists of losing determining strings, $\alpha^c \notin \omega[\mathbb{N}]$. Hence $\alpha^c \notin \omega$, as desired.

   Computability, monotonicity, and nonweakness of $\omega$ are immediate from the corresponding properties of $\omega[A]$. The proof that $\omega$ does not have a finite carrier is similar to the proof for $\omega[A]$.

4. $(+\,+\,-\,-)$ A monotonic, proper, nonstrong, weak game. In the construction of (the sets $T_0$ and $T_1$ for) $\omega[A]$ in Section 6.2, replace (2.i), (2.ii), and (3) by

   (2*.i) for each p-string $\alpha'$ that is a proper substring of $\alpha$, if $s = 0$ or $|\alpha'| \geq l_{s-1}$, then enumerate $1 * \alpha' * 11$ in $T_1$ and $1 * \alpha' * 00$ in $T_0$; furthermore, enumerate 0 in $T_0$;

(2*.ii) if $f(\alpha) \in A$, enumerate $1 * \alpha$ in $T_1$; if $f(\alpha) \notin A$, enumerate $1 * \alpha$ in $T_0$;

(3*) if a string $\beta = 1 * \beta'$ is enumerated in $T_1$ (or in $T_0$) above, then enumerate $1 * (\beta')^c$ in $T_0$ (or in $T_1$, respectively).

Let $T_0'$ and $T_1'$ be the sets $T_0$ and $T_1$ in the original (Section 6.2) construction of $\omega[A]$ renamed. We observe that $\beta = 1 * \beta' \in T_i$ if and only if $\beta' \in T_i'$.

We first show that any coalition $S$ has exactly one initial segment in $T_0 \cup T_1$. This is immediate if $S \supseteq 0$. So, suppose $S \supseteq 1$. Define $S'$ by $S'(k) = S(k+1)$ for all $k$. Then, by the proof of Lemma 10 for $\omega[A]$, $S'$ has exactly one initial segment $S'[k]$ in $T_0' \cup T_1'$. From the observation above, $S[k+1] = 1 * S'[k] \in T_0 \cup T_1$ for a unique $k$, which is what we wanted.

To show the game is monotonic, it suffices to show Lemma 11 (i) holds for the newly defined game. Suppose that $\alpha$, $\beta$ satisfy the assumption of the lemma and that $\alpha$ extends a string $\hat{\alpha}$ in $T_1$ and $\beta$ properly contains $\alpha$. Then, $\hat{\alpha} \supseteq 1$; write $\hat{\alpha} = 1 * \hat{\alpha}'$. Then $\hat{\alpha}' \in T_1'$ from the observation above. We can write $\beta = 1 * \beta'$. Then $\beta'$ either extends or properly contains $\hat{\alpha}' \in T_1'$. If $\beta'$ extends $\hat{\alpha}' \in T_1'$, then $\beta$ extends $1 * \hat{\alpha}' \in T_1$, as desired. Otherwise, $\beta'$ properly contains $\hat{\alpha}' \in T_1'$. By Lemma 11 for the original game $\omega[A]$ (the condition that $l_s = |\alpha|$ can be ignored for our purpose), $\beta'$ extends a string $\hat{\beta} \in T_1'$. So, $\beta = 1 * \beta'$ extends $1 * \hat{\beta} \in T_1$, as desired.

The game is weak (hence proper by Lemma 1) since every winning coalition extends 1; in other words, 0 is a veto player. It is nonstrong since $\{0\} \supseteq 100 \in T_0$ implies $\{0\} \notin \omega$, while $\{0\}^c \supseteq 0 \in T_0$ implies $\{0\}^c \notin \omega$. The proof that the game is computable and has no finite carrier is similar to the proofs for $\omega[A]$.

5. $(+ - ++)$ A monotonic, nonproper, strong, nonweak game. Let $\omega = \omega[\emptyset] \cup \omega[\mathbb{N}]$; that is, $S \in \omega$ if and only if $S \in \omega[\emptyset]$ or $S \in \omega[\mathbb{N}]$.

To show $\omega$ is nonproper, let $\alpha \in F$. We show that both $\alpha$ and $\alpha^c$ are winning. On the one hand, we have $\alpha \in T_1^{\mathbb{N}}$ by (2.ii). So $\alpha \in \omega[\mathbb{N}]$, implying $\alpha \in \omega$. On the other hand, we have $\alpha \in T_0^{\emptyset}$ by (2.ii). Hence $\alpha^c \in T_1^{\emptyset}$. So $\alpha^c \in \omega[\emptyset]$. Hence $\alpha^c \in \omega$, as desired.

To show $\omega$ is strong, suppose $S \notin \omega$ and $S^c \notin \omega$. Then $S \notin \omega[\mathbb{N}]$ and $S^c \notin \omega[\mathbb{N}]$, contradicting the strongness of $\omega[\mathbb{N}]$.

Computability and monotonicity of $\omega$ are immediate from the corresponding properties of $\omega[A]$. Nonweakness is immediate from nonproperness by Lemma 1. The proof that $\omega$ does not have a finite carrier is similar to the proof for $\omega[A]$.

6. $(+\,-\,+-)$ A monotonic, nonproper, strong, weak game. There is no such game.

7. $(+\,-\,-+)$ A monotonic, nonproper, nonstrong, nonweak game. Let $A$ be the set of even numbers. In the construction of $\omega[A]$, replace (2.ii) and (3) by

   (2*.ii) if $f(\alpha) \in A$, enumerate $\alpha$ and $\alpha^c$ in $T_1$; if $f(\alpha) \notin A$, enumerate $\alpha$ and $\alpha^c$ in $T_0$;

   (3*) if a string $\beta$ is enumerated in $T_1$ (or in $T_0$) by applying (2.i), then enumerate $\beta^c$ in $T_0$ (or in $T_1$, respectively).

   To show the game is monotonic, it suffices to show Lemma 11 (i) holds. Suppose that $\alpha$, $\beta$ satisfy the assumption of the lemma and that $\alpha$ extends a string $\alpha'$ in $T_1$ and $\beta$ properly contains $\alpha$. Let $T_0'$ and $T_1'$ be the sets $T_0$ and $T_1$ in the original construction of $\omega[A]$ renamed. Note that the replacement of (2.ii) and (3) by (2*.ii) and (3*) only affects p-strings, but not d-strings; hence the set of d-strings in $T_1$ is the same as the set of d-strings in $T_1'$, the set of d-strings in $T_0$ is the same as the set of d-strings in $T_0'$, and the set of p-strings in $T_0 \cup T_1$ is the same as the set of p-strings in $T_0' \cup T_1'$. If $\alpha'$ is a d-string in $T_1$, it is in $T_1'$. Lemma 11 (i) implies that $\beta$ extends a string in $T_1'$. In fact, an inspection of the proof of Lemma 11 reveals that $\beta$ extends a d-string in $T_1'$, unless $\beta \supseteq \alpha'$, in which case the conclusion is obvious. So assume $\beta \not\supseteq \alpha'$. Then $\beta$ extends a d-string in $T_1'$; hence it extends a d-string in $T_1$, as desired. If $\alpha'$ is a p-string in $T_1$, it is in $T_1' \cup T_0'$. If $\alpha' \in T_1'$, then Lemma 11 (i) implies that $\beta$ extends a string in $T_1'$. So the rest of the proof is similar. If $\alpha' \in T_0'$, then Lemma 11 (ii) implies that $\beta^c$ extends a string in $T_0'$. Assume $\beta \not\supseteq \alpha'$ as before. Then $\beta^c$ extends a d-string in $T_0'$; hence it extends a d-string in $T_0$. By (3*), $\beta$ extends a d-string in $T_1$, as desired.

   The game is nonproper since (2*.ii) implies that there is a string $\alpha \in F$ such that the coalitions $\{i : \alpha(i) = 1\}$ and $\{i : \alpha(i) = 1\}^c$ (which extends $\alpha^c$) are winning. Similarly, it is nonstrong since there is a string $\alpha \in F$ such that the coalitions above are losing. It is nonweak by Lemma 1 since it is nonproper. The proof that the game is computable and has no finite carrier is similar to the proofs for $\omega[A]$.

8. $(+\,-\,--)$ A monotonic, nonproper, nonstrong, weak game. There is no such game.

9. $(-\,+\,++)$ A nonmonotonic, proper, strong, nonweak game. In the construction of $\omega[A]$, replace (2.i) by

(2*.i) for each p-string $\alpha' \neq \emptyset$ that is a proper substring of $\alpha$, if $s = 0$ or $|\alpha'| \geq l_{s-1}$, then enumerate $\alpha' * 11$ in $T_1$ and $\alpha' * 00$ in $T_0$; furthermore, enumerate 00 in $T_1$.

By (3) of the construction, $11 \in T_0$. (In other words, the game is constructed from the sets $T_0 := T_0' \cup \{11\} \setminus \{00\}$ and $T_1 := T_1' \cup \{00\} \setminus \{11\}$, where $T_0'$ and $T_1'$ are $T_0$ and $T_1$ in the original construction of $\omega[A]$ renamed.) Since 00 is winning and 11 is losing, the game is nonmonotonic. It is also nonweak since 00 (or an empty coalition) is winning. For the remaining properties, the proofs are similar to the proofs for $\omega[A]$.

10. $(-++-)$ A nonmonotonic, proper, strong, weak game. There is no such game.

11. $(-+-+)$ A nonmonotonic, proper, nonstrong, nonweak game. In the construction of $\omega[A]$, replace (2.i) and (3) by

(2*.i) for each p-string $\alpha' \neq \emptyset$ that is a proper substring of $\alpha$, if $s = 0$ or $|\alpha'| \geq l_{s-1}$, then enumerate $\alpha' * 11$ in $T_1$ and $\alpha' * 00$ in $T_0$; furthermore, enumerate 00 and 11 in $T_0$;

(3*) if a string $\beta \notin \{00, 11\}$ is enumerated in $T_1$ (or in $T_0$) above, then enumerate $\beta^c$ in $T_0$ (or in $T_1$, respectively).

(In other words, the game is constructed from the sets $T_0 := T_0' \cup \{11\}$ and $T_1 := T_1' \setminus \{11\}$, where $T_0'$ and $T_1'$ are $T_0$ and $T_1$ in the original construction of $\omega[A]$ renamed.)

The game is nonmonotonic since $N$ is losing but there are winning coalitions. It is proper since it is a subset of $\omega[A]$, which is proper. It is nonstrong since $11, 00 \in T_0$ implies that the coalitions $\{0, 1\}$, $\{0, 1\}^c$ are losing.

To show nonweakness, find a $\beta \in T_1$ such that $|\beta| = l_{t+1}$ for some $t$ (e.g., let $\beta = \alpha^{--} * 11$ in the proof of Lemma 14, with $s$ replaced by $t$). Choose an $s$ such that $l_{t+1} < l_s < l_{s+1}$. Following the proof of Lemma 14, we can find $\alpha \in F_{s+1}$ such that $|\alpha^{--}| \geq l_s$, $\alpha^{--} * 11 \in T_1$, and $\alpha^{--} * 00 \in T_0$. Then $(\alpha^c)^{--} * 11 \in T_1$. Nonweakness follows since the intersection of winning coalitions $\beta$ (regarded as the coalition $\{i : \beta(i) = 1\}$), $\alpha^{--} * 11 \in T_1$, and $(\alpha^c)^{--} * 11$ is empty.

The proofs of computability and nonexistence of a finite carrier are similar to the proofs for $\omega[A]$.

12. $(-+--)$ A nonmonotonic, proper, nonstrong, weak game. Let $A = \mathbb{N}$. In the construction of $\omega[A] = \omega[\mathbb{N}]$, replace (2.i) by

(2*.i) for each p-string $\alpha'$ that extends 1010 or 1001 and is a proper substring of $\alpha$, if $s = 0$ or $|\alpha'| \geq l_{s-1}$, then enumerate $\alpha' * 11$ in $T_1$ and $\alpha' * 00$ in $T_0$; furthermore, enumerate d-strings 11 and 1000 in $T_1$ and strings 1011 and 0 in $T_0$.

and remove (3). To show that any coalition $S$ has an initial segment in $T_0 \cup T_1$, suppose that $S$ extends 1010 or 1001. (The other cases are immediate.) Let $T_0'$ and $T_1'$ be $T_0$ and $T_1$ in the original construction of $\omega[\mathbb{N}]$ renamed. Then, by Proposition 10, $S$ has an initial segment $S[k]$ in $T_0' \cup T_1'$, where $k \geq 4$ without loss of generality. If $S[k]$ is enumerated in $T_0' \cup T_1'$ by applying (2.ii), then it is enumerated in $T_0 \cup T_1$ by applying (2.ii). So, the conclusion follows. If $S[k]$ is enumerated in $T_0' \cup T_1'$ by applying (2.i), then $S[k]$ is equal to $\alpha' * 11$ or $\alpha' * 00$ for some p-string $\alpha'$ satisfying the requirements in (2.i). Clearly, $\alpha'$ extends 1010 or 1001. So, $S[k]$ is enumerated in $T_0 \cup T_1$ by applying (2*.i). So the conclusion follows.

To show that no coalition $S$ has initial segments in both $T_0$ and $T_1$, it suffices to show that a string $\alpha$ enumerated in $T_0$ by (2*.i) and a p-string $\beta$ enumerated in $T_1$ by (2.ii) are incompatible. (Note that all $\alpha \in F$ are enumerated in $T_1$ and none in $T_0$ by (2.ii).) Since $\beta \supset 10$, it is incompatible with $0 \in T_1$. All the other strings enumerated by (2*.i) are d-strings, so $\alpha$ and $\beta$ are compatible only if $\alpha$ extends $\beta$, which in turn extends (since $\beta \in F$ is of length $\geq 4$) 1001 or 1010. Then, $\alpha = \alpha' * 00$ for some $\alpha'$, so as above, $\alpha \in T_0'$; similarly, $\beta \in T_1'$. This implies that $\alpha$ and $\beta$ are incompatible.

The game $\omega$ defined above is nonmonotonic since 1000 is winning but 1011 is not. To see $\omega$ is weak (hence proper by Lemma 1), note that any winning coalition extends 1; so the intersection contains a veto player 0. The game is nonstrong because 0, $1011 \in T_0$ imply that the coalitions $\{1\}$ and $\{1\}^c$ are losing. The proofs of computability and nonexistence of a finite carrier are similar to the proofs for $\omega[A]$.

13. $(--++)$ A nonmonotonic, nonproper, strong, nonweak game. In the construction of $\omega[A]$, replace (2.i) and (3) by

(2*.i) for each p-string $\alpha' \neq \emptyset$ that is a proper substring of $\alpha$, if $s = 0$ or $|\alpha'| \geq l_{s-1}$, then enumerate $\alpha' * 11$ in $T_1$ and $\alpha' * 00$ in $T_0$; furthermore, enumerate 00 and 11 in $T_1$;

(3*) if a string $\beta \notin \{00, 11\}$ is enumerated in $T_1$ (or in $T_0$) above, then enumerate $\beta^c$ in $T_0$ (or in $T_1$, respectively).

(In other words, the game is constructed from the sets $T_0 := T_0' \setminus \{00\}$ and $T_1 := T_1' \cup \{00\}$, where $T_0'$ and $T_1'$ are $T_0$ and $T_1$ in the original construction of $\omega[A]$ renamed.)

The game is nonmonotonic since $\emptyset$ is winning but there are losing coalitions. It is nonproper since the coalitions $\{0,1\}$, $\{0,1\}^c$ are winning. It is strong since its subset $\omega[A]$ is strong. It is nonweak by Lemma 1 since it is nonproper. The proofs of computability and nonexistence of a finite carrier are similar to the proofs for $\omega[A]$.

14. $(--+-)$ A nonmonotonic, nonproper, strong, weak game. There is no such game.

15. $(---+)$ A nonmonotonic, nonproper, nonstrong, nonweak game. In the construction of $\omega[A]$, replace (2.i) and (3) by

    (2*.i) for each p-string $\alpha'$ that extends 1010 or 1001 and is a proper substring of $\alpha$, if $s = 0$ or $|\alpha'| \geq l_{s-1}$, then enumerate $\alpha' * 11$ in $T_1$ and $\alpha' * 00$ in $T_0$; furthermore, enumerate d-strings 00, 1000, and 0111 in $T_0$ and d-strings 11, 1011 and 0100 in $T_1$;

    (3*) if a string $\beta \notin \{00, 11, 1000, 0111, 1011, 0100\}$ is enumerated in $T_1$ (or in $T_0$) above, then enumerate $\beta^c$ in $T_0$ (or in $T_1$, respectively).

    The game is nonmonotonic since 0100 is winning but 0111 is not. The game is nonproper since 1011, $0100 \in T_1$ imply that the coalitions $\{1\}$ and $\{1\}^c$ are winning. It is nonstrong since 1000, $0111 \in T_0$ imply $\{0\}$ and $\{0\}^c$ are losing. It is nonweak by Lemma 1 since it is nonproper. The proofs of computability and nonexistence of a finite carrier are similar to the proofs for $\omega[A]$.

16. $(----)$ A nonmonotonic, nonproper, nonstrong, weak game. There is no such game.

## References


Al-Najjar, N. I., Anderlini, L., Felli, L., 2006. Undescribable events. Review of Economic Studies 73, 849–868.

Anderlini, L., Felli, L., 1994. Incomplete written contracts: Undescribable states of nature. Quarterly Journal of Economics 109, 1085–1124.

Arrow, K. J., 1963. Social Choice and Individual Values, 2nd Edition. Yale University Press, New Haven.

Bartholdi, III, J., Tovey, C. A., Trick, M. A., 1989a. Voting schemes for which it can be difficult to tell who won the election. Social Choice and Welfare 6, 157–165.

Bartholdi, III, J. J., Tovey, C. A., Trick, M. A., 1989b. The computational difficulty of manipulating an election. Social Choice and Welfare 6, 227–241.

Kelly, J. S., 1988. Social choice and computational complexity. Journal of Mathematical Economics 17, 1–8.

Krasa, S., Williams, S. R., 2007. Limited observability as a constraint in contract design. Journal of Economic Theory 134, 379–404.

Kumabe, M., Mihara, H. R., 2007a. Computability of simple games: A characterization and application to the core. Journal of Mathematical Economics Doi:10.1016/j.jmateco.2007.05.012.

Kumabe, M., Mihara, H. R., Jun. 2007b. The Nakamura numbers for computable simple games. MPRA Paper 3684, Munich University Library.

Landes, W. M., Posner, R. A., 1976. Legal precedent: A theoretical and empirical analysis. Journal of Law and Economics 19, 249–307.

Lewis, A. A., 1988. An infinite version of Arrow's Theorem in the effective setting. Mathematical Social Sciences 16, 41–48.

Lyons, D., 1984. Formal justice, moral commitment, and judicial precedent. Journal of Philosophy 81, 580–587.

May, K. O., 1952. A set of independent, necessary and sufficient conditions for simple majority decision. Econometrica 20, 680–84.

May, K. O., 1953. A note on the complete independence of the conditions for simple majority decision. Econometrica 21, 172–173.

Mihara, H. R., Aug. 1997. Arrow's Theorem and Turing computability. Economic Theory 10, 257–76.

Mihara, H. R., 1999. Arrow's theorem, countably many agents, and more visible invisible dictators. Journal of Mathematical Economics 32, 267–287.

Mihara, H. R., 2004. Nonanonymity and sensitivity of computable simple games. Mathematical Social Sciences 48, 329–341.

Odifreddi, P., 1992. Classical Recursion Theory: The Theory of Functions and Sets of Natural Numbers. Elsevier, Amsterdam.

Peleg, B., 2002. Game-theoretic analysis of voting in committees. In: Arrow, K. J., Sen, A. K., Suzumura, K. (Eds.), Handbook of Social Choice and Welfare. Vol. 1. Elsevier, Amsterdam, Ch. 8, pp. 395–423.

Rasmusen, E., 1994. Judicial legitimacy as a repeated game. Journal of Law, Economics, and Organization 10, 63–83.

Richter, M. K., Wong, K.-C., 1999. Computable preference and utility. Journal of Mathematical Economics 32, 339–354.

Soare, R. I., 1987. Recursively Enumerable Sets and Degrees: A Study of Computable Functions and Computably Generated Sets. Springer-Verlag, Berlin.

Thomson, W., 2001. On the axiomatic method and its recent applications to game theory and resource allocation. Social Choice and Welfare 18, 327–386.

Weber, R. J., 1994. Games in coalitional form. In: Aumann, R. J., Hart, S. (Eds.), Handbook of Game Theory. Vol. 2. Elsevier, Amsterdam, Ch. 36, pp. 1285–1303.

Weihrauch, K., July 1995. A simple introduction to computable analysis, `http://eccc.hpi-web.de/eccc-local/ECCC-Books/klaus_book_readme.html`