The Nakamura numbers for computable simple games

Kumabe, Masahiro and Mihara, H. Reiju

23 June 2007

The Nakamura numbers for computable simple games

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

H. Reiju Mihara*
http://econpapers.repec.org/RAS/pmi193.htm
Graduate School of Management, Kagawa University
Takamatsu 760-8523, Japan

June 2007

Abstract

The Nakamura number of a simple game plays a critical role in preference aggregation (or multi-criterion ranking): the number of alternatives that the players can always deal with rationally is less than this number. We comprehensively study the restrictions that various properties for a simple game impose on its Nakamura number. We find that a computable game has a finite Nakamura number greater than three only if it is proper, nonstrong, and nonweak, regardless of whether it is monotonic or whether it has a finite carrier. The lack of strongness often results in alternatives that cannot be strictly ranked.

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

Keywords: Nakamura number, voting games, the core, Turing computability, axiomatic method, multi-criterion decision-making.

*The corresponding author; the family name is "Mihara." Phone: +81-87-832-1831. Fax: +81-87-832-1820.

1 Introduction

The *Nakamura number* plays a critical role in the study of preference aggregation rules with *acyclic* social preferences.[1] Consider a *(simple) game* (voting game)—a coalitional game that assigns either 0 or 1 to each coalition: those assigned 1 are winning coalitions and those assigned 0 are losing coalitions. Combining the simple game with a set of alternatives and a profile of individual preferences, we define a *simple game with (ordinal) preferences*. Nakamura's theorem (1979) gives a necessary and sufficient condition for a simple game with preferences to have a nonempty *core* (the set of maximal elements of the social preference produced by the aggregation rule) for all profiles: the number of alternatives is less than a certain number, called the *Nakamura number* of the simple game.

Kumabe and Mihara (2007, Theorem 14) extend Nakamura's theorem to their framework and apply it to *computable* simple games. They show that every (nonweak) computable game has a finite Nakamura number. This implies that under the preference aggregation rule based on a computable game, the number of alternatives that the set of players can deal with rationally is restricted by this number.

We are therefore interested in the question of how large the Nakamura number can be. In fact, Kumabe and Mihara (2007, Proposition 12) show that every integer $k \geq 2$ is the Nakamura number of some computable game. Of course, a large Nakamura number can be attained only by satisfying or violating certain properties for simple games. For example, the Nakamura number of a nonproper game, which admits two complementing winning coalitions, is at most 2 (Lemma 6).

In this paper, we study the restrictions that various properties (axioms) for a simple game impose on its Nakamura number. We restrict our attention to the computable simple games and classify them into thirty-two (2^5) classes in terms of their *types* (with respect to monotonicity, properness, strongness, and nonweakness) and finiteness (existence of a finite carrier). Table 1 summarizes the results. For example, a type 5 ($+-++$) (monotonic, *non*proper, strong, nonweak) computable game has Nakamura number equal to 2, whether it is finite or infinite.[2]

[1]Banks (1995), Truchon (1995), and Andjiga and Mbih (2000) are recent contributions to the literature. Earlier papers on acyclic rules can be found in Truchon (1995) and Austen-Smith and Banks (1999). Note that acyclicity of a preference is necessary and sufficient for the existence of a maximal element on every finite subset of alternatives. When the weak social preferences are required to be *transitive*, we are back in Arrow's difficult setting (1963).

[2]Strictly speaking, we only assert in this paper that the numbers in each entry in the table are *not ruled out*; we are not much interested in asserting that *every* entry not indicated "none" contains a game in which an empty coalition is losing. However, those who accept the results in Kumabe and Mihara (2006) will find the latter assertion acceptable. For most entries, the examples given in the paper cited suffice. For the other

Table 1: Possible Nakamura Numbers for Computable Games

Types	Finite	Infinite	Types	Finite	Infinite
1$(++++)$	3	3	9$(-+++)$	2	2
2$(+++-)$	$+\infty$	none	10$(-++-)$	none	none
3$(++-+)$	≥ 3	≥ 3	11$(-+-+)$	≥ 2	≥ 2
4$(++--)$	$+\infty$	$+\infty$	12$(-+--)$	$+\infty$	$+\infty$
5$(+-++)$	2	2	13$(--++)$	2	2
6$(+-+-)$	none	none	14$(--+-)$	none	none
7$(+--+)$	2	2	15$(---+)$	2	2
8$(+---)$	none	none	16$(----)$	none	none

Possible Nakamura numbers are given in each entry, assuming that an empty coalition is losing (so that the Nakamura number is at least 2). The types are defined by the four conventional axioms: monotonicity, properness, strongness, and nonweakness. For example, the entries corresponding to Type 2 $(+++-)$ indicates that among the computable, monotonic $(+)$, proper $(+)$, strong $(+)$, weak $(-$, because *not* nonweak) games, finite ones have a Nakamura number equal to $+\infty$ and infinite ones do not exist.

We make two observations from Table 1. First, *a computable game has a finite Nakamura number greater than 3 only if it is proper,* non*strong, and nonweak* (i.e., either of type 3 $(++-+)$ or of type 11 $(-+-+)$).[3] In particular, for the players to be always able to choose a maximal element from at least three alternatives, strongness of the game must be forgone (unless the game is dictatorial (type 2)). Second, as far as computable games are concerned, *a number k is the Nakamura number of a* finite *game of a certain type (except type 2) if and only if it is that of an* infinite *game of the same type.* Restricting games to finite ones does not reduce the number of alternatives that the players can deal with rationally.

The rest of the Introduction gives a background briefly. Most of it is fully discussed in Kumabe and Mihara (2007).

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.[4]

entries, we need to modify the examples—which we do, with the exception of a few entries (footnote 9).

[3]Propositions 11 and 19 state that any Nakamura number $k \geq 3$ is attainable by type 3 finite and infinite games. Propositions 12 and 20 state that any Nakamura number $k \geq 2$ is attainable by type 11 finite and infinite games.

[4]This literature includes Kelly (1988), Lewis (1988), Bartholdi et al. (1989a,b), Mihara

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*" or "*procedure.*" Second, there is a normative reason: computability of social choice rules formalizes the notion of "due process."[5]

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 (2007) 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 can naturally re-interpret the preference aggregation problem (which provides motivation for studying the Nakamura number) as a *multi-criterion ranking* problem, for example. In multi-criterion ranking, each criterion ranks finitely many alternatives; we are interested in aggregating those countably many rankings into one (acyclic relation). Assuming that the underlying simple game is computable is intuitively plausible in view of the following consequences: (i) each criterion is treated differently;[6] (ii) whether an alternative has a higher rank than another can be determined by examine finitely many criteria, though how many criteria need to be examined depends on each situation (Proposition 4). The (lack of strongness) observation mentioned above suggests that rational choice from many (at least three) alternatives often involves alternatives that cannot be strictly ranked.

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**.

(1997, 1999, 2004), and Kumabe and Mihara (2007, 2006).

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

[6] Computable simple games violate anonymity (Kumabe and Mihara, 2007, Proposition 10).

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.[7] 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 ω are said to be **winning**. The coalitions not in ω 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 ω is said to be **monotonic** if for all coalitions S and T, the conditions $S \in \omega$ and $T \supseteq S$ imply $T \in \omega$. ω is **proper** if for all recursive coalitions S, $S \in \omega$ implies $S^c := N \setminus S \notin \omega$. ω is **strong** if for all coalitions S, $S \notin \omega$ implies $S^c \in \omega$. ω 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.) ω 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 ω is a coalition $S \subset N$ such that

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

for all coalitions T. When a game ω has a carrier T, we often restrict the game on T and identify ω with $\omega|T := \{S \cap T : S \in \omega\}$. 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**.

The **Nakamura number** $\nu(\omega)$ of a game ω is the size of the smallest collection of winning coalitions having empty intersection

$$\nu(\omega) = \min\{\#\omega' : \omega' \subseteq \omega \text{ and } \bigcap \omega' = \emptyset\}$$

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

if $\bigcap \omega = \emptyset$ (i.e., ω is nonweak); otherwise, set $\nu(\omega) = +\infty$, which is understood to be greater than any cardinal number. In computing the Nakamura number for a game, it suffices to look only at the subfamily of minimal winning coalitions, *provided that the game is finite*. If the game is infinite, we cannot say so since minimal winning coalitions may not exist.

Extending and applying a well-known result by Nakamura (1979), Kumabe and Mihara (2007, Corollary 15) show that that computability of a game entails a restriction on the number of alternatives that the set of players (with the coalition structure described by the game) can deal with rationally. We relegate the details to that paper.

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 (Δ_0-index). Here, a number e is a **characteristic index** for a coalition S if φ_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 ω, its **δ-indicator** is the partial function δ_ω 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 δ_ω 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 *(δ)-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.

(δ)-computability δ_ω has an extension to a partial recursive function.

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

3 Preliminary Results

In this section, we give a sufficient condition and a necessary condition for a game to be computable.

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

The following proposition restates a sufficient condition (Kumabe and Mihara, 2007, the "if" direction of Theorem 3) for a game to be computable. In particular, *finite games are computable.* The proposition can be proved easily:

Proposition 3 (Kumabe and Mihara (2006)) *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 ω 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 ω is δ-computable.*

The following proposition gives a necessary condition for a game to be computable:

Proposition 4 (Kumabe and Mihara (2007, Proposition 2)) *Suppose that a δ-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.*

4 The Main Results

We classify computable games into thirty-two (2^5) classes as shown in Table 1, in terms of their **(conventional) types** (with respect to the conventional axioms of monotonicity, properness, strongness, and nonweakness) and finiteness (existence of a finite carrier). Among the sixteen types, five (types 6, 8, 10, 14, and 16) contain no games; also, the class of type 2 infinite games is empty (since type 2 games are dictatorial).[8]

We therefore have only $(16-5)\times 2-1=21$ classes of games to be checked. For each such class, we find the set of possible Nakamura numbers. We do so, whenever important, by constructing a game in the class having a particular Nakamura number, unless the example given in Kumabe and Mihara (2006) suffices.[9]

We only consider games in which $\emptyset$ is losing. Otherwise, the Nakamura number for the game becomes 1—not a very interesting case. (Also, note that if $\emptyset$ is winning and the game has a losing coalition, then it is *non*monotonic.)

We consider *weak* games first. Among the weak games, types 2, 4, and 12 are nonempty.[10] By definition, their Nakamura number is infinite. We have so far examined all the types whose labels are even numbers.

We henceforth consider *nonweak* (hence nonempty by definition) computable games; they have finite Nakamura numbers:

Lemma 5 (Kumabe and Mihara (2007, Corollary 13)) *Let ω be a computable, nonweak simple game. Then, its Nakamura number $\nu(\omega)$ is finite.*

4.1 Small Nakamura numbers

First, the definition of proper games implies the following:

Lemma 6 *Let ω be a game satisfying $\emptyset \notin \omega$ and $\omega \neq \emptyset$. If ω is nonproper, then ω is nonweak with $\nu(\omega) = 2$.*

Lemma 6 is equivalent to the assertion that a game is proper if its Nakamura number $\nu(\omega)$ is at least 3. It does not rule out the possibility that proper games have Nakamura number equal to 2. Lemma 6 implies that the games of types 5, 7, 13, and 15 have Nakamura number equal to 2.

[8]These results, also found in Kumabe and Mihara (2006), are immediate from Lemmas 1 and 2.

[9]Some examples in Kumabe and Mihara (2006) violate the condition that $\emptyset$ is losing, which we impose in this paper. In this paper, we do not give examples of games with a small Nakamura number when the construction is based on the details of the paper cited. Specifically, we omit examples of a type 9 infinite game and a type 13 infinite game.

[10]These types contain (in fact, consist of) games in which $\emptyset$ is losing. If $\emptyset$ were winning, then the game would not be weak. Kumabe and Mihara (2006) give examples of these types of games.

Example 1 gives examples of type 13 and type 15 finite games.[11]

Example 1. We first give a type 13 finite game. Let $T = \{0, 1, 2\}$ be a carrier and let $\omega|T := \{S \cap T : S \in \omega\}$ consist of $\{0, 1, 2\}$, $\{1, 2\}$, $\{0\}$, $\{1\}$, $\{2\}$. The other three coalitions in T are losing. Then, ω is nonmonotonic, nonproper, strong, and nonweak with $\nu(\omega) = 2$.

We next give a type 15 finite game. Let $T = \{0, 1, 2\}$ be a carrier and let $\omega|T$ consist of $\{0, 1, 2\}$, $\{1, 2\}$, $\{0\}$, $\{1\}$. The other four coalitions in T are losing. Then, ω is nonmonotonic, nonproper, nonstrong, and nonweak with $\nu(\omega) = 2$. $\|$

Next, we consider computable *strong* games that are nonweak. These games have Nakamura numbers not greater than 3:

Lemma 7 *Let ω be a computable, strong nonweak game satisfying $\emptyset \notin \omega$. Then $\nu(\omega) = 2$ or 3.*

Proof. Since ω is computable, by Proposition 4, every winning coalition has a finite subcoalition that is winning, which in turn has a minimal winning subcoalition that is winning. If there is only one minimal winning coalition $S \neq \emptyset$, then the intersection of all winning coalitions is S, which is nonempty; this violates the nonweakness of ω. So there are at least two (distinct) minimal winning coalitions S_1 and S_2 in ω. Let $S = S_1 \cap S_2$. S is losing since it is a proper subcoalition of the minimal winning coalition S_1. Then, since ω is strong, S^c is winning. Since $S_1 \cap S_2 \cap S^c = S \cap S^c = \emptyset$, we have $\nu(\omega) \leq 3$ by the definition of the Nakamura number. The assumption that $\emptyset \notin \omega$ rules out $\nu(\omega) = 1$. ($\nu(\omega) = 2$ if there are distinct minimal winning coalitions S_1 and S_2 such that $S = S_1 \cap S_2 = \emptyset$; otherwise, $\nu(\omega) = 3$.) ∎

Remark 1. The computability condition cannot be dropped from Lemma 7 (a minimal winning coalition may not exist if a winning coalition has no finite, winning subcoalition). A nonprincipal ultrafilter is a counterexample; it has an infinite Nakamura number. (See Kumabe and Mihara (2007, Sections 2.1 and 4.3) for the definition of a *nonprincipal ultrafilter* and the observation that it has no finite winning coalitions and is noncomputable, monotonic, proper, strong, and nonweak.) $\|$

[11] It is easy to show that types 5 and 7 contain games in which $\emptyset$ is losing. If $\emptyset$ were winning, then by monotonicity the game would consist of all coalitions (a type 5 game). Since the examples of types 5 and 7 games in Kumabe and Mihara (2006) all have losing coalitions, $\emptyset$ is losing in those games. The type 15 infinite game in that paper satisfies the condition that $\emptyset$ is losing. To show that type 13 contains an infinite game in which $\emptyset$ is losing is more delicate, but can be done by modifying the example in that paper.

Lemma 8 *Let ω be a monotonic proper game satisfying $\emptyset \notin \omega$ and $\omega \neq \emptyset$. Then $\nu(\omega) \geq 3$.*

Proof. Suppose $\nu(\omega) = 2$. Then, there are winning coalitions S, S' whose intersection is empty. That is $S' \subseteq S^c$. By monotonicity, S^c is winning, implying that ω is not proper. ∎

Lemma 9 *Let ω be a nonmonotonic strong game satisfying $\emptyset \notin \omega$ and $\omega \neq \emptyset$. Then ω is nonweak with $\nu(\omega) = 2$.*

Proof. Since nonempty ω is nonmonotonic, there exist a winning coalition S and a losing coalition S' such that $S \cap S'^c = \emptyset$. This means that the Nakamura number is 2, since S'^c is winning by strongness of ω. ∎

Lemma 7 and Lemma 8 imply that type 1 games have a Nakamura number equal to 3. Lemma 9 implies that type 9 games have a Nakamura number equal to 2. Proposition 10 and Example 2 give examples of these games:[12]

Proposition 10 *There exist finite, type* 1 *(i.e., monotonic proper strong nonweak) games and infinite, computable, type* 1 *games.*

Proof. An example of a type 1 finite game is a majority game with an odd number of (at least three) players. An example of a type 1 infinite game is given in Appendix A. ∎

Example 2. We give a type 9 finite game. Let $T = \{0, 1, 2\}$ be a carrier and let $\omega|T := \{S \cap T : S \in \omega\}$ consist of $\{0, 1, 2\}$, $\{0\}$, $\{1\}$, $\{2\}$. The other four coalitions in T are losing. Then, ω is nonmonotonic, proper, strong, and nonweak with $\nu(\omega) = 2$. $\|$

4.2 Large Nakamura numbers

Having considered all the other types of games, we now turn to types 3 and 11 (i.e., proper nonstrong nonweak games). These are the only types that may have a Nakamura number greater than 3.

First, we consider games with finite carriers. An example of a game having Nakamura number equal to $k \geq 2$ can be defined on the carrier $T = \{0, 1, \ldots, k-1\}$; the game ω consists of the coalitions excluding at most one player in the carrier: $S \in \omega$ if and only if $\#(T \cap S) \geq k - 1$. We extend this example slightly:

[12] We can also give an example of an infinite, computable, type 9 game, but it rests on the details of the construction in Kumabe and Mihara (2006).

Proposition 11 *For any $k \geq 3$, there exists a finite, computable, type* 3 *(i.e., monotonic proper nonstrong nonweak) game ω with Nakamura number $\nu(\omega) = k$.*

Proof. Given $k \geq 2$, let $\{T_0, T_1, \dots, T_{k-1}\}$ be a partition of a finite carrier $T = \bigcup_{l=0}^{k-1} T_l$. Define $S \in \omega$ iff $\#\{T_l : T_l \subseteq S\} \geq k - 1$. Then it is straightforward to show that ω is monotonic and nonweak with $\nu(\omega) = k$. Now, suppose that $k \geq 3$. To show that ω is proper, suppose $S \in \omega$. Then S includes at least $k-1$ of the partition elements T_l, implying that S^c includes at most one of them. To show that ω is nonstrong, suppose that a partition element, say T_l, contains at least two players, one of whom is denoted by t. We then have the following two losing coalitions complementing each other: (i) the union of $k-2$ partition elements $T_{l'}$ and $\{t\}$ and (ii) the union of the other partition element and $T_l \setminus \{t\}$. ∎

Remark 2. Because of Lemma 8, Proposition 11 precludes $k = 2$. Note that the game in the proof is nonproper if and only if $k = 2$. If $k \leq 3$, then it generally fails to be strong, though it is indeed strong if all the partition elements T_l consist of singletons. ‖

Proposition 12 *For any $k \geq 2$, there exists a finite, computable, type* 11 *(i.e., nonmonotonic proper nonstrong nonweak) game ω with Nakamura number $\nu(\omega) = k$.*

Proof. Given $k \geq 3$, let $\{T_0, T_1, \dots, T_{k-1}\}$ be a partition of a finite carrier $T = \bigcup_{l=0}^{k-1} T_l$. Define $S \in \omega$ iff $\#\{T_l : T_l \subseteq S\} = k - 1$. Then ω is nonmonotonic; the rest of the proof is similar to that of Proposition 11.

For $k = 2$, we give the following example: Let $T = \{0, 1, 2\}$ be a carrier and define $\omega|T = \{S \cap T : S \in \omega\} = \{\{0\}, \{1\}\}$. It is nonmonotonic since $\{0\} \in \omega$ but $\{0, 1\} \notin \omega$. It is proper: $S \in \omega$ implies $S \cap T = \{0\}$ or $\{1\}$, which in turn implies $S^c \cap T = \{1, 2\}$ or $\{0, 2\}$, neither of which is in $\omega|T$; hence $S^c \notin \omega$. It is nonstrong since $\{0, 1\}$ and $\{2\}$ are losing. It is nonweak with $\nu(\omega) = 2$ since the intersection of the winning coalitions $\{0\}$ and $\{1\}$ is empty. ∎

Next, we move on to games without finite carriers. We construct them using the notion of the *product* of games. By a **recursive function f on a recursive set $T \subseteq N$** we mean a recursive function restricted to T.

Let (f_1, f_2) be a pair consisting of a one-to-one recursive function f_1 on a (not necessarily finite) recursive set $T \subseteq N$ and a one-to-one recursive function f_2, whose images partition the set of players: $f_1(T) \cap f_2(N) = \emptyset$

and $f_1(T) \cup f_2(N) = N$. Note that f_1^{-1} and f_2^{-1} are recursive functions on recursive sets $f_1(T)$ and $f_2(N)$, respectively.[13]

We define the **disjoint image of coalitions $S_1 \subseteq T$ and $S_2 \subseteq N$ with respect to** (f_1, f_2) as the set

$$S_1 * S_2 = f_1(S_1) \cup f_2(S_2),$$

where $f_1(S_1) = \{f_1(i) : i \in S_1\}$ and $f_2(S_2) = \{f_2(i) : i \in S_2\}$.

Example 3. When $T = N$, an easy example is given by $f_1 : i \mapsto 2i$ and $f_2 : i \mapsto 2i + 1$. In this case, $f_1(T) = 2N := \{2i : i \in N\}$, $f_2(N) = 2N + 1 := \{2i + 1 : i \in N\}$, and $\{0, 2, 3\} * \{1, 2, 4\} = \{0, 4, 6, 3, 5, 9\}$. When $T = \{0, 1, \ldots, k - 1\}$ for some $k \geq 1$, an easy example is given by $f_1 : i \mapsto i$ and $f_2 : i \mapsto i + k$. In this case, if $k = 4$, we have $f_1(T) = T$, $f_2(N) = N \setminus T = \{4, 5, 6, \ldots\}$, and $\{0, 2, 3\} * \{1, 2, 4\} = \{0, 2, 3, 5, 6, 8\}$. $\|$

Lemma 13 *Let* REC *be the class of (recursive) coalitions. Then,*

$$\{S_1 * S_2 : S_1 \subseteq T \text{ and } S_2 \text{ are coalitions}\} = \text{REC}.$$

Proof. ($\subseteq$). If S_1 is recursive, then $f_1(S_1)$ is recursive. (*Details.* Since S_1 and $T \setminus S_1$ are recursive, $f_1(S_1)$ and $f_1(T \setminus S_1)$ are recursively enumerable sets that partition the recursive set $f_1(T)$. $f_1(S_1)$ is therefore recursive.) Similarly $f_2(S_2)$ is recursive. It follows that $f_1(S_1) \cup f_2(S_2)$ is recursive.

($\supseteq$). Let S be recursive. Then

$$\begin{aligned} S &= [S \cap f_1(T)] \cup [S \cap f_2(N)] \\ &= [f_1(f_1^{-1}(S \cap f_1(T)))] \cup [f_2(f_2^{-1}(S \cap f_2(N)))] \end{aligned}$$

∎

Let ω_1 be a game with a carrier included in a set T. (This is without loss of generality since the grand coalition N is a carrier for any game.) Let ω_2 be a game. We define the **product** $\omega_1 \otimes \omega_2$ **of** ω_1 **and** ω_2 **with respect to** (f_1, f_2) by the set

$$\omega_1 \otimes \omega_2 = \{f_1(S_1) \cup f_2(S_2) : S_1 \in \omega_1 \text{ and } S_2 \in \omega_2\}$$

of the disjoint images of winning coalitions.[14] By Lemma 13, $\omega_1 \otimes \omega_2$ is a simple game. We have $S_1 * S_2 \in \omega_1 \otimes \omega_2$ if and only if $S_1 \in \omega_1$ and $S_2 \in \omega_2$.

[13] In general, if f is a recursive function and S is a recursive set, then the image $f(S)$ is recursively enumerable. So $f_1(T)$ and $f_2(N)$ are recursively enumerable. Since they complement each other on the set N, they are in fact both recursive.

[14] The notion of the *product* of games is not new. For example, Shapley (1962) defines it for two games on disjoint subsets of players.

Lemma 14 *If ω_1 and ω_2 are computable, then the product $\omega_1 \otimes \omega_2$ is computable.*

Proof. Let e be a characteristic index for a coalition $S := S_1 * S_2 = f_1(S_1) \cup f_2(S_2)$. It suffices to show that given e, we can effectively obtain a characteristic index for S_1 (and similarly for S_2).

Let t be a characteristic index for $f_1(T)$, a fixed recursive set. Effectively obtain (Soare, 1987, Corollary II.2.3) from e and t a characteristic index e' for $f_1(S_1) = [f_1(S_1) \cup f_2(S_2)] \cap f_1(T)$. Let t' be an index for the recursive function

$$\varphi_{t'}(i) = \begin{cases} f_1(i) & \text{if } i \in T \\ f_2(0) & \text{otherwise.} \end{cases}$$

We claim that $\varphi_{e'} \circ \varphi_{t'}$ is the characteristic function for the recursive set S_1. (*Details.* Suppose $i \in S_1$ first. Then $i \in T$ and $f_1(i) \in f_1(S_1)$. Hence $\varphi_{e'} \circ \varphi_{t'}(i) = \varphi_{e'}(f_1(i)) = 1$. Suppose $i \notin S_1$ next. If $i \in T$, then $f_1(i) \in f_1(T) \setminus f_1(S_1)$. Hence $\varphi_{e'} \circ \varphi_{t'}(i) = \varphi_{e'}(f_1(i)) = 0$. If $i \notin T$, then $\varphi_{e'} \circ \varphi_{t'}(i) = \varphi_{e'}(f_2(0)) = 0$, since $f_2(0) \notin f_1(S_1)$.)

By the Parameter Theorem (Soare, 1987, I.3.5), there is a recursive function g such that $\varphi_{g(e')}(i) = \varphi_{e'} \circ \varphi_{t'}(i)$, implying that $g(e')$ is characteristic index for S_1 that can be obtained effectively. ∎

It turns out that the construction based on the product is very useful for our purpose.

Lemma 15 *ω_1 and ω_2 are monotonic if and only if the product $\omega_1 \otimes \omega_2$ is monotonic.*

Proof. By Lemma 13, any coalition $\hat{S}$ can be written as $\hat{S} = \hat{S}_1 * \hat{S}_2$ for some $\hat{S}_1 \subseteq T$ and $\hat{S}_2$.

($\Longrightarrow$). Suppose $S_1 * S_2 \in \omega_1 \otimes \omega_2$ and $S_1 * S_2 \subseteq S_1' * S_2'$. Then, we have $S_1 \in \omega_1$, $S_2 \in \omega_2$, and $f_1(S_1) \cup f_2(S_2) \subseteq f_1(S_1') \cup f_2(S_2')$. Noting that $f_1(S_1) \subseteq f_1(T)$, $f_1(S_1') \subseteq f_1(T)$, $f_2(S_2) \subseteq f_2(N)$, $f_2(S_2') \subseteq f_2(N)$, and $f_1(T) \cap f_2(N) = \emptyset$, we have $f_1(S_1) \subseteq f_1(S_1')$ and $f_2(S_2) \subseteq f_2(S_2')$. Hence $S_1 \subseteq S_1'$ and $S_2 \subseteq S_2'$. Since $S_1 \in \omega_1$ and $S_2 \in \omega_2$, monotonicity implies that $S_1' \in \omega_1$ and $S_2' \in \omega_2$. That is, $S_1' * S_2' \in \omega_1 \otimes \omega_2$.

($\Longleftarrow$). We suppose that $\omega_1 \otimes \omega_2$ is monotonic and show that ω_1 is monotonic. Suppose $S_1 \in \omega_1$ and $S_1 \subset S_1'$. Choose any $S_2 \in \omega_2$. Then $S_1 * S_2 \in \omega_1 \otimes \omega_2$. By monotonicity, $S_1' * S_2 \in \omega_1 \otimes \omega_2$. Hence $S_1' \in \omega_1$. ∎

Lemma 16 *If ω_1 or ω_2 is proper, then the product $\omega_1 \otimes \omega_2$ is proper.*

Proof. First, we can show that $(S_1 * S_2)^c = S_1^c * S_2^c$, where $S_1^c = T \setminus S_1$ and $S_2^c = N \setminus S_2$. Indeed, $(S_1 * S_2)^c = (f_1(S_1) \cup f_2(S_2))^c = (f_1(S_1))^c \cap (f_2(S_2))^c =$

$[f_1(T)\setminus f_1(S_1)\cup f_2(N)]\cap[f_1(T)\cup f_2(N)\setminus f_2(S_2)] = f_1(T\setminus S_1)\cup f_2(N\setminus S_2) = S_1^c * S_2^c$.

Now suppose $S_1 * S_2 \in \omega_1 \otimes \omega_2$. Then, $S_1 \in \omega_1$ and $S_2 \in \omega_2$. Since ω_1 or ω_2 is proper, we have either $S_1^c \notin \omega_1$ or $S_2^c \notin \omega_2$. It follows that $(S_1 * S_2)^c = S_1^c * S_2^c \notin \omega_1 \otimes \omega_2$. ∎

Lemma 17 *Suppose ω_1 is nonstrong or ω_2 is nonstrong or both ω_1 and ω_2 have losing coalitions. Then the product $\omega_1 \otimes \omega_2$ is nonstrong.*

Proof. We give a proof for the case where each game has a losing coalition: $S_1 \notin \omega_1$ and $S_2^c \notin \omega_2$. Then, $S_1 * S_2 \notin \omega_1 \otimes \omega_2$ and $(S_1 * S_2)^c = S_1^c * S_2^c \notin \omega_1 \otimes \omega_2$. ∎

Lemma 18 *If ω_1 and ω_2 are nonweak, then the product $\omega_1 \otimes \omega_2$ is nonweak. Its Nakamura number is $\nu(\omega_1 \otimes \omega_2) = \max\{\nu(\omega_1), \nu(\omega_2)\}$.*

Proof. If $\bigcap \omega_1 = \bigcap \omega_2 = \emptyset$, then $\bigcap(\omega_1 \otimes \omega_2) = \bigcap_{S_1 * S_2 \in \omega_1 \otimes \omega_2}(S_1 * S_2) = \bigcap_{S_1 \in \omega_1, S_2 \in \omega_2}(f_1(S_1) \cup f_2(S_2)) = (\bigcap_{S_1 \in \omega_1} f_1(S_1)) \cup (\bigcap_{S_2 \in \omega_2} f_2(S_2))$ [because $f_1(S_1) \cap f_2(S_2) = \emptyset$ for all S_1 and S_2] $= f_1(\bigcap_{S_1 \in \omega_1} S_1) \cup f_2(\bigcap_{S_2 \in \omega_2} S_2) = (\bigcap \omega_1) * (\bigcap \omega_2) = \emptyset$. The proof for the Nakamura number is similar. ∎

Propositions 11 and 12 have analogues for infinite games (because of Lemma 8 again, Proposition 19 precludes $k = 2$):

Proposition 19 *For any $k \geq 3$, there exists an infinite, computable, type* 3 *(i.e., monotonic proper nonstrong nonweak) game ω with Nakamura number $\nu(\omega) = k$.*

Proof. For $k \geq 3$, let ω_1 be a finite, computable, type 3 game with $\nu(\omega_1) = k$. (Such a game exists by Proposition 11.) Let ω_2 be an infinite, computable, monotonic nonweak game (which need not be proper or strong or nonstrong) with $\nu(\omega_2) \leq 3$. (Such a game exists by Proposition 10.) Lemmas 14, 15, 16, 17, 18 imply that the product $\omega_1 \otimes \omega_2$ satisfies the conditions. ∎

Proposition 20 *For any $k \geq 2$, there exists an infinite, computable, type* 11 *(i.e., nonmonotonic proper nonstrong nonweak) game ω with Nakamura number $\nu(\omega) = k$.*

Proof. For $k \geq 2$, let ω_1 be a finite, computable, type 11 game with $\nu(\omega_1) = k$. (Such a game exists by Proposition 12.) Let ω_2 be an infinite, computable, nonproper game. (Types 5, 7, 13, and 15 in Kumabe and Mihara (2006) are examples. Alternatively, just for obtaining the results for $k \geq 3$, we can let ω_2 be an infinite, computable, nonweak game with $\nu(\omega_2) = 3$, which exists by Proposition 10.) Then the game is nonweak, with $\nu(\omega_2) = 2$ (if $\emptyset \notin \omega_2$; Lemma 6) or $\nu(\omega_2) = 1$ (otherwise). Lemmas 14, 15, 16, 17, 18 imply that the product $\omega_1 \otimes \omega_2$ satisfies the conditions. ∎

A An Infinite, Computable, Type 1 Game

We exhibit here an infinite, computable, type 1 (i.e., monotonic proper strong nonweak) simple game, thus giving a proof to Proposition 10. Though Kumabe and Mihara (2006) give an example, the readers not comfortable with recursion theory may find it too complicated. In view of the fact that such a game is used in an important result (e.g., Proposition 19) in this paper, it makes sense to give a simpler construction here.[15]

Our approach is to construct recursively enumerable (in fact, recursive) sets T_0 and T_1 of strings (of 0's and 1's) satisfying the conditions of Proposition 3. We first construct certain sets F_s of strings for $s \in \{0, 1, 2, \ldots\}$. We then specify each of T_0 and T_1 using the sets F_s, and construct a simple game ω according to Proposition 3. We conclude that the game is computable by checking (Lemmas 22 and 25) that T_0 and T_1 satisfy the conditions of Proposition 3. Finally, we show (Claims 27, 28, and 29) that the game satisfies the desired properties.

Notation. Let α and β be strings (of 0's and 1's).

Then α^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 α is identified with the set $\{i : \alpha(i) = 1\}$. (Note however that α^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 α followed by β.

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

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

Let $\{k_s\}_{s=0}^{\infty}$ be an effective listing (recursive enumeration) of the members of the recursively enumerable set $\{k : \varphi_k(k) \in \{0, 1\}\}$, where $\varphi_k(\cdot)$

[15] One reason that the construction in Kumabe and Mihara (2006) is complicated is that they construct a *family* of type 1 games $\omega[A]$, one for each recursive set A, while requiring *additional conditions* that would later become useful for constructing other types of games. In this appendix, we construct just one type 1 game, forgetting about the additional conditions. Some aspects of the construction thus become more apparent in this construction. The construction extends the one (not requiring the game to be of a particular type) in Kumabe and Mihara (2007, Section 6.2).

is the kth partial recursive function of one variable (it is computed by the Turing program with code (Gödel) number k). We can assume that $k_0 \geq 2$ and all the elements k_s are distinct. Thus,

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

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

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

For each s, let F_s be the finite set of strings $\alpha = \alpha(0)\alpha(1)\cdots\alpha(l_s - 1)$ of length $l_s \geq 3$ such that

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

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

Lemma 21 *Any two distinct elements α and β in F are* incompatible. *That is, we have neither $\alpha \subseteq \beta$ (α is an initial segment of β) nor $\beta \subseteq \alpha$ (i.e., there is $k < \min\{|\alpha|, |\beta|\}$ such that $\alpha(k) \neq \beta(k)$).*

Proof. Let $|\alpha| \leq |\beta|$, without loss of generality. If α and β 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), $\alpha(k_s) = \varphi_{k_s}(k_s)$ on the one hand, but $\beta(k_s) = 1 - \varphi_{k_s}(k_s)$ on the other hand. So $\alpha(k_s) \neq \beta(k_s)$. ∎

The game ω will be constructed from the sets T_0 and T_1 of strings defined as follows ($10 = 1 * 0$, $00 = 0 * 0$, and $11 = 1 * 1$ below):

$$\begin{aligned}
\alpha \in T_0^0 &\iff \exists s\,[\alpha \in F_s,\ \alpha \supseteq 10, \text{ and } \alpha(k_s)(= \varphi_{k_s}(k_s)) = 0] \\
\alpha \in T_1^0 &\iff \exists s\,[\alpha \in F_s,\ \alpha \supseteq 10, \text{ and } \alpha(k_s)(= \varphi_{k_s}(k_s)) = 1] \\
\alpha \in T_0 &\iff [\alpha \in T_0^0 \text{ or } \alpha^c \in T_1^0 \text{ or } \alpha = 00] \\
\alpha \in T_1 &\iff [\alpha \in T_1^0 \text{ or } \alpha^c \in T_0^0 \text{ or } \alpha = 11].
\end{aligned}$$

We observe that the sets T_0^0, T_1^0, T_0, T_1 consist of strings whose lengths are at least 2, $T_0^0 \subset T_0$, $T_1^0 \subset T_1$, $T_0 \cap T_1 = \emptyset$, and $\alpha \in T_0 \Leftrightarrow \alpha^c \in T_1$.

Define ω by $S \in \omega$ if and only if S has an initial segment in T_1. Lemmas 22 and 25 establish computability of ω (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 3.

Lemma 22 *T_0 and T_1 are recursive.*

Proof. We give an algorithm that can decide for each given string σ with a length of at least 2 whether it is in T_0 or in T_1 or neither.

If $\sigma \supseteq 00$, then $\sigma \notin T_0 \cup T_1$ unless $\sigma = 00 \in T_0$.

If $\sigma \supseteq 11$, then $\sigma \notin T_0 \cup T_1$ unless $\sigma = 11 \in T_1$.

Suppose $\sigma \supseteq 10$. In this case, $\sigma \in T_0 \cup T_1$ iff $\sigma \in T_0^0 \cup T_1^0$. Generate $k_0, k_1, k_2, \ldots$, compute $l_0, l_1, l_2, \ldots$, and determine $F_0, F_1, F_2, \ldots$ until we find the least s such that $l_s \geq |\sigma|$.

If $l_s > |\sigma|$, then $\sigma \notin F_s$. Since l_s is nondecreasing in s and F_s consists of strings of length l_s, it follows that $\sigma \notin F$, implying $\sigma \notin T_0^0 \cup T_1^0$, that is, $\sigma \notin T_0 \cup T_1$.

If $l_s = |\sigma|$, then check whether $\sigma \in F_s$; this can be done since the values of $\varphi_{k_{s'}}(k_{s'})$ for $s' \leq s$ in (1) are available and F_s determined by time s. If $\sigma \notin F_s$ and $l_{s+1} > l_s$, then $\sigma \notin T_0 \cup T_1$ as before. Otherwise check whether $\sigma \in F_{s+1}$. If $\sigma \notin F_{s+1}$ and $l_{s+2} > l_{s+1} = l_s$, then $\sigma \notin T_0 \cup T_1$ as before. Repeating this process, we either get $\sigma \in F_{s'}$ for some s' or $\sigma \notin F_{s'}$ for all $s' \in \{s' : l_{s'} = l_s\}$. In the latter case, we have $\sigma \notin T_0 \cup T_1$. In the former case, if $\sigma(k_{s'}) = \varphi_{k_{s'}}(k_{s'}) = 1$, then $\sigma \in T_1^0 \subset T_1$ by the definitions of T_1^0 and T_1. Otherwise $\sigma(k_{s'}) = \varphi_{k_{s'}}(k_{s'}) = 0$, and we have $\sigma \in T_0^0 \subset T_0$.

Suppose $\sigma \supseteq 01$. Then $\sigma^c \supseteq 10$. In this case the algorithm can decide whether σ^c is in T_0^0 or in T_1^0 or neither. If $\sigma^c \in T_0^0$, then $\sigma \in T_1$. If $\sigma^c \in T_1^0$, then $\sigma \in T_0$. If $\sigma^c \notin T_0^0 \cup T_1^0$, then $\sigma \notin T_0 \cup T_1$. ∎

Lemma 23 *Let α, β be distinct strings in $T_0 \cup T_1$. Then α and β are incompatible. In particular, if $\alpha \in T_0$ and $\beta \in T_1$, then α and β are incompatible.*

Proof. Suppose α and β are compatible. Then there is a coalition S extending α and β.

If $\alpha \supseteq 00$, then $\beta \supseteq 00$. But there is only one string in $T_0 \cup T_1$ that extends 00; namely, 00. So $\alpha = \beta = 00$, contrary to the assumption that they are distinct. The case where $\alpha \supseteq 11$ is similar.

If $\alpha \supseteq 10$, then $\beta \supseteq 10$. So we have α, $\beta \in T_0^0 \cup T_1^0$, which implies that α, $\beta \in F$. By Lemma 21, S cannot extend both α and β, a contradiction.

If $\alpha \supseteq 01$, then $\beta \supseteq 01$. So we have α^c, $\beta^c \in T_1^0 \cup T_1^0$, which implies that α^c, $\beta^c \in F$. By Lemma 21, S^c cannot extend both α^c and β^c, a contradiction. ∎

Lemma 24 *Let $\alpha \supseteq 10$ be a string of length l_s such that $\alpha(k_s) = \varphi_{k_s}(k_s)$. Then for some $t \leq s$, there is a string $\beta \in F_t$ such that $10 \subseteq \beta \subseteq \alpha$.*

Proof. We proceed by induction on s. If $s = 0$, we have $\beta = \alpha \in F_0$ (note that (1) imposes no constraints on $\alpha(0)$ and $\alpha(1)$); hence the lemma holds for $s = 0$. Suppose the lemma holds for $s' < s$. If for some $s' < s$, $\alpha(k_{s'}) = \varphi_{k_{s'}}(k_{s'})$, then by the induction hypothesis, for some $t \leq s'$, the

$l_{s'}$-initial segment $\alpha[l_{s'}]$ of α extends a string $\beta \in F_t$. Hence the conclusion holds for s. Otherwise, we have for each $s' < s$, $\alpha(k_{s'}) = 1 - \varphi_{k_{s'}}(k_{s'})$. Then by (1), $\alpha \in F_s$. Letting $\beta = \alpha$ gives the conclusion. ∎

Lemma 25 *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 23 implies that S does not have initial segments in both T_0 and T_1. (The assertion following "In particular" in Lemma 23 is sufficient for this, but we can actually show the stronger statement that S has exactly one initial segment in $T_0 \cup T_1$.)

The conclusion is obvious if $S \supseteq 00$ or $S \supseteq 11$.

If $S \supseteq 10$, suppose φ_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. Consider the initial segment $S[l_s] := S \cap l_s = \varphi_{k_s}[l_s] \supseteq 10$. By Lemma 24, for some $t \leq s$, there is a string $\beta \in F_t$ such that $10 \subseteq \beta \subseteq S[l_s]$. The conclusion follows since β is an initial segment of S and $\beta \in T_0^0 \cup T_1^0 \subset T_0 \cup T_1$.

If $S \supseteq 01$, then $S^c \supseteq 10$ has an initial segment $\beta \in T_0^0 \cup T_1^0$ by the argument above. So, S has the initial segment $\beta^c \in T_1 \cup T_0$. ∎

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

Lemma 26 *Let α and β be strings such that $|\alpha| \leq |\beta|$.* (i) *If $\alpha \in T_1$ and β properly contains α, then β extends a string in T_1.* (ii) *If $\alpha \in T_0$ and β is properly contained by α, then β extends a string in T_0.*

Proof. (i) Suppose $\alpha \in T_1$ and β properly contains α. We have $\alpha = 11$ or $\alpha \in T_1^0$ or $\alpha^c \in T_0^0$.

If $\alpha = 11$, no β properly contains α.

Suppose $\alpha \in T_1^0$. Then $\alpha \in F_s$ for some s. Since β properly contains $\alpha \supseteq 10$, we have $\beta \supseteq 11$ or $\beta \supseteq 10$. If $\beta \supseteq 11$, the conclusion follows since $11 \in T_1$. Otherwise, $\beta \supseteq 10$; choose the least $s' \leq s$ such that $\beta(k_{s'}) = \varphi_{k_{s'}}(k_{s'}) = 1$. (Such an s' exists since $\alpha \in T_1^0$ implies $\alpha(k_s) = \varphi_{k_s}(k_s) = 1$. Note that $k_{s'} < l_{s'} \leq l_s = |\alpha|$.) Then for each $t < s'$, we have $\beta(k_t) = 1 - \varphi_{k_t}(k_t)$. (*Details.* By the choice of s', for each $t < s'$, either (a) $\beta(k_t) = \varphi_{k_t}(k_t) = 0$ or (b) $\beta(k_t) \neq \varphi_t(k_t)$. Suppose (a) for some $t < s'$. Since $\alpha \in F_s$, we have for each $t < s$, $\alpha(k_t) = 1 - \varphi_{k_t}(k_t)$ by (1). Then we have $\beta(k_t) = 0$ and

$\alpha(k_t) = 1$, contradicting the assumption that β properly contains α.) The conclusion follows since the initial segment $\beta[l_{s'}]$ is in T_1^0.

Suppose $\alpha^c \in T_0^0$. Then $\alpha^c \in F_s$ for some s. Since β^c is properly contained in $\alpha^c \supseteq 10$, we have $\beta^c \supseteq 00$ or $\beta^c \supseteq 10$. If $\beta^c \supseteq 00$, the conclusion follows since $\beta \supseteq 11 \in T_1$. Otherwise, $\beta^c \supseteq 10$; Choose the least $s' \leq s$ such that $\beta^c(k_{s'}) = \varphi_{k_{s'}}(k_{s'}) = 0$. Then for each $t < s'$, we have $\beta^c(k_t) = 1 - \varphi_{k_t}(k_t)$ as before. Therefore, the initial segment $\beta^c[l_{s'}]$ is in T_0^0. The conclusion follows since $\beta[l_{s'}] \in T_1$.

(ii) Suppose $\alpha \in T_0$ and β is properly contained by α. Then $\alpha^c \in T_1$ and β^c properly contains α^c. Assertion (i) then implies that β^c extends a string $\beta^c[l_{s'}]$ in T_1. Therefore, β extends the string $\beta[l_{s'}]$ in T_0. ∎

Claim 27 *The game ω is monotonic.*

Proof. Suppose $A \in \omega$ and $B \supseteq A$. By the definition of ω, A has an initial segment α in T_1. If B extends α, then clearly $B \in \omega$. Otherwise the $|\alpha|$-initial segment $\beta = B[|\alpha|]$ of B properly contains α. By Lemma 26, β extends a string in T_1. Hence B has an initial segment in T_1, implying that $B \in \omega$. ∎

Claim 28 *The game ω 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

$$\begin{aligned} S^c \in \omega &\iff S^c \text{ has an initial segment in } T_1 \\ &\iff S \text{ has an initial segment in } T_0 \\ &\iff S \notin \omega. \end{aligned}$$

∎

Claim 29 *The game ω is nonweak and does not have a finite carrier.*

Proof. To show that the game does not have a finite carrier, we will construct a set A such that for infinitely many l, the l-initial segment $A[l]$ has an extension (as a string) that is winning and for infinitely many l', $A[l']$ has an extension that is losing. This implies that $A[l]$ is not a carrier of ω for any such l. So no subset of $A[l]$ is a carrier. Since there are arbitrarily large such l, this proves that ω has no finite carrier.

Let $A \supseteq 10$ be a set such that for each k_t, $A(k_t) = 1 - \varphi_{k_t}(k_t)$. For any $s' > 0$ and $i \in \{0, 1\}$, there is an $s > s'$ such that $k_s > l_{s'}$ and $\varphi_{k_s}(k_s) = i$.

For a temporarily chosen s', fix i and fix such s. Then choose the greatest s' satisfying these conditions. Since $l_s > k_s > l_{s'}$, there is a string α of length l_s extending $A[l_{s'}]$ such that $\alpha \in F_s$. Since $\alpha \supseteq 10$ and $\alpha(k_s) = \varphi_{k_s}(k_s) = i$, we have $\alpha \in T_i^0$.

There are infinitely many such s, so there are infinitely many such s'. It follows that for infinitely many $l_{s'}$, the initial segment $A[l_{s'}]$ is a substring of some string α in T_1, and for infinitely many $l_{s'}$, $A[l_{s'}]$ is a substring of some (losing) string α in T_0.

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 $A \supseteq 10$) has extensions α in T_1 and β in T_0 by the argument above. So 01 has the extension β^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. ∎

References

Andjiga, N. G. and Mbih, B. (2000). A note on the core of voting games. *Journal of Mathematical Economics*, 33:367–372.

Arrow, K. J. (1963). *Social Choice and Individual Values.* Yale University Press, New Haven, 2nd edition.

Austen-Smith, D. and Banks, J. S. (1999). *Positive Political Theory I: Collective Preference.* University of Michigan Press, Ann Arbor.

Banks, J. S. (1995). Acyclic social choice from finite sets. *Social Choice and Welfare*, 12:293–310.

Bartholdi, III, J., Tovey, C. A., and 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., and 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.

Kumabe, M. and Mihara, H. R. (2006). Computability of simple games: A complete investigation of the sixty-four possibilities. MPRA Paper 440, Munich University Library.

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

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

Mihara, H. R. (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.

Nakamura, K. (1979). The vetoers in a simple game with ordinal preferences. *International Journal of Game Theory*, 8:55–61.

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., and Suzumura, K., editors, *Handbook of Social Choice and Welfare*, volume 1, chapter 8, pages 395–423. Elsevier, Amsterdam.

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

Shapley, L. S. (1962). Simple games: An outline of the descriptive theory. *Behavioral Science*, 7:59–66.

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

Truchon, M. (1995). Voting games and acyclic collective choice rules. *Mathematical Social Sciences*, 29:165–179.

Weber, R. J. (1994). Games in coalitional form. In Aumann, R. J. and Hart, S., editors, *Handbook of Game Theory*, volume 2, chapter 36, pages 1285–1303. Elsevier, Amsterdam.

B Tupe 9 and Type 13 Games (Not to be Published)

In this attachment, we modify the examples of a type 9 game and a type 13 game in Kumabe and Mihara (2006) so that an empty coalition is losing. To do that, modify the infinite, computable, type 1 game $\omega[A]$ in that paper as follows ((2.i) and (3) refer to certain requirements in that paper):

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

   (2*.i) for each p-string $\alpha' \neq 10$ that is a proper substring of α, if $s = 0$ or $|\alpha'| \geq l_{s-1}$, then enumerate $\alpha' * 11$ in T_1 and $\alpha' * 00$ in T_0; furthermore, enumerate 1011 and 1000 in T_0.

   By (3) of the construction of $\omega[A]$, $0100, 0111 \in T_1$. (In other words, the game is constructed from the sets $T_0 := T_0' \cup \{1011\} \setminus \{0100\}$ and $T_1 := T_1' \cup \{0100\} \setminus \{1011\}$, where T_0' and T_1' are T_0 and T_1 in the original construction of $\omega[A]$ renamed. Note that $1011 \in T_1'$, $1000 \in T_0'$, $0100 \in T_0'$, and $0111 \in T_1'$.) Letting $\alpha' = \emptyset$ in (2*.i), we have $00 \in T_0$; so $\emptyset$ is losing. Since either $\alpha' = 1010$ or 1001 is a p-string satisfying the condition in (2*.i), either $101011 \in T_1$ or $100111 \in T_1$. Then by (3), either $010100 \in T_0$ or $011000 \in T_0$. So the game is nonmonotonic, since 0100 is winning. It is also nonweak since 0100 is winning and either 101011 or 100111 is winning. For the remaining properties, the proofs are similar to the proofs for $\omega[A]$.

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

   (2*.i) for each p-string $\alpha' \neq 10$ that is a proper substring of α, if $s = 0$ or $|\alpha'| \geq l_{s-1}$, then enumerate $\alpha' * 11$ in T_1 and $\alpha' * 00$ in T_0; furthermore, enumerate 1011 and 0100 in T_1 and 1000 in T_0;

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

   By (3*), $0111 \in T_1$. (In other words, the game is constructed from the sets $T_0 := T_0' \setminus \{0100\}$ and $T_1 := T_1' \cup \{0100\}$, where T_0' and T_1' are T_0 and T_1 in the original construction of $\omega[A]$ renamed.)

   By an argument similar to that for type 9, $\emptyset$ is losing and the game is nonmonotonic (either $010100 \in T_0$ or $011000 \in T_0$, while 0100 is winning). It is nonproper since the 0100 and 1011 are winning determining. 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 $\omega[A]$.