Arrow's Theorem, countably many agents, and more visible invisible dictators*

H. Reiju Mihara†
reiju@ec.kagawa-u.ac.jp
Economics, Kagawa University, Takamatsu 760-8523, Japan‡

September, 1998
[Journal of Mathematical Economics (1999) 32: 267–287]

Abstract

For infinite societies, Fishburn (1970), Kirman and Sondermann (1972), and Armstrong (1980) gave a *nonconstructive* proof of the existence of a social welfare function satisfying Arrow's conditions (Unanimity, Independence, and Nondictatorship). This paper improves on their results by (i) giving a *concrete example* of such a function, and (ii) showing how to compute, from a description of a profile on a pair of alternatives, which alternative is socially preferred under the function. The introduction of a certain "oracle" resolves Mihara's impossibility result (1997) about computability of social welfare functions.

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

Keywords: Arrow impossibility theorem, Turing computability, recursion theory, oracle algorithms, free ultrafilters.

*The main result of the paper appeared as Theorem 2 of my thesis (Mihara, 1995) submitted to the University of Minnesota. I am indebted to my thesis adviser Marcel K. Richter and to Wayne Richter for their comments and guidance. Thanks are also due to Leonid Hurwicz and Andrew McLennan. The paper was presented at the Economic Design Workshop (Institute of Social and Economic Research, Osaka) in July 1997 and at the Third Decentralization Conference (Sano Shoin, Hitotsubashi) in September 1997.

†[Fax: +81-87-832-1820. Phone: +81-87-832-1831.]

‡Temporary address (till February 1999): ISER, Osaka University, 6-1 Mihogaoka, Ibaraki, Osaka 567-0047, Japan [Fax: +81-6-878-2766].

1 Introduction

Contrary to Arrow's negative result (1963) for finite societies, Fishburn (1970), Kirman and Sondermann (1972), Armstrong (1980; 1985), and Lauwers and Van Liedekerke (1995) show that when the number of individuals in a society is *infinite*, there exists a social welfare function (which assigns a social preference to each profile of individual preferences) satisfying Arrow's conditions of Unanimity, Independence, and Nondictatorship. Their proofs of the existence, however, is *nonconstructive*, failing to give a concrete example of such a function.[1]

In the context of social choice, where social welfare functions are often understood to be a "process or rule" (Arrow, 1963, p. 23), or "procedure" (Arrow, 1963, p. 2), just giving nonconstructive proofs seems unsatisfactory. To my knowledge, however, a constructive proof of the possibility theorem has not been given. This is not surprising for two reasons.

The first reason is mathematical: the existence of a social welfare function satisfying Arrow's conditions implies (Kirman and Sondermann, 1972, Theorem 1) the existence of a free ultrafilter on the set of individuals; however, as stated in footnote 1, an ultrafilter is a highly nonconstructive mathematical object. This fact gives an impression that the desired social welfare function cannot be constructed without using the axiom of choice or a similar nonconstructive mathematical technique. (Pazner and Wesley (1978) observe a similar point in a paper dealing with social choice functions—which assigns an *alternative* to each profile—instead of social welfare functions.)

The second reason is interpretive: any social welfare function satisfying Arrow's conditions has (Kirman and Sondermann, 1972; Armstrong, 1980; Lauwers and Van Liedekerke, 1995) certain oligarchical characteristics. This means that arbitrarily "small" (though infinite) coalitions can dictate the social preference—in the sense that whenever all individuals in the coalition prefer a certain alternative x to another alternative y, the society prefers x to y. In the words of Kirman and Sondermann (1972), there is an "invisible dictator." One could thus argue that constructing a particular social welfare function satisfying Arrow's conditions would not do much good since the function constructed would have the undesirable property anyway.

I do not regard either of the above reasons as compelling. As for the

[1] The proofs explicitly or implicitly resort to the existence of a free ultrafilter over an arbitrary infinite set. But it is known (Koppelberg, 1989, p. 33) that the existence cannot be derived from the axiom system ZF of Zermelo-Fraenkel set theory without the axiom of choice.

first reason, I may be able to avoid the mathematical problem by modifying the framework slightly—and indeed I do. As for the second reason, while I agree with the assertion (Lauwers and Van Liedekerke, 1995, p. 236) that ultrafilters on which the oligarchical functions are based do not provide a good intuition of "almost all," I do not think that a good social choice must always reflect the preferences of almost all "individuals"—especially if the word "individual" does not refer to an ordinary human being. And in the context of social choice by *infinitely* many individuals, it is natural to interpret (Mihara, 1995; Mihara, 1997) an "individual" as a "person at a particular state of the world." (The social choice is made before the true state is known. There are finitely many persons and infinitely many states; the same *person* at different states are regarded as different *individuals*.) Under this particular interpretation, one can regard an oligarchical social welfare function as simply ignoring the states that are unlikely to be realized. Along this line of reasoning, I suggest elsewhere (Mihara, 1998) that an oligarchical function may be regarded as a component or module of a more appealing superrule. To construct such a superrule, I first need to construct its components.

The main objective of this paper is to construct a concrete example of a social welfare function satisfying Arrow's conditions. In other words, I attempt to "visualize" the "invisible dictator" in this paper. For this purpose, I adopt (Section 2) the framework of my earlier paper (Mihara, 1997), which exploits Armstrong's framework (1980; 1985) in a natural way. In particular, I restrict the domain of a social welfare function to those profiles that are *measurable*: for all alternatives x and y, the coalition that prefers x to y is recursive. (A coalition is *recursive* if there exists an algorithm to decide for each individual whether she is a member of the coalition.) Then a concrete example of a social welfare function satisfying Arrow's conditions can be constructed (as in Mihara (1998)) in this framework. (The fact that the class REC of recursive coalitions is a countable Boolean algebra containing all finite coalitions is used.) The domain restriction above is not an artificial mathematical assumption introduced simply to obtain a positive result. In particular, it does not restrict the preferences of any individual artificially. Instead, it meaningfully restricts the way different individuals may have different preferences. In the above interpretation, the restriction means that each *person* can identify the set of states in which she prefers x to y—a natural epistemological condition.

In this paper, I do more than just give a concrete example of a social welfare function satisfying Arrow's conditions. I also investigate—from the viewpoint of computability analysis of social choice—how complex such functions are. (This viewpoint is absent from my related paper (Mihara, 1998). *Computability analysis of social choice* studies algorithmic properties of social decision-making. The literature includes Kelly (1988), Lewis (1988), and Mihara (1997), who study issues in social choice using *recursion theory* (study of algorithms). More generally, application of recursion theory to economic theory and game theory includes Lewis (1985), Spear (1989), Canning (1992), Anderlini and Felli (1994), and Wong (1994). See also Kalai (1990) and Lipman (1995) for surveys of the literature on information processing in these areas.)

I start investigating the complexity issues with a review (Section 3) of my earlier theorem (Mihara, 1997). The impossibility theorem (Proposition 1) implies that if a social welfare function satisfying Unanimity and Independence is computable in the following sense, then it must be dictatorial. I say that a social welfare function is computable (satisfies *Strong Pairwise Computability*) if there is an algorithm that can decide (for each pair (x, y) of alternatives) whether the society prefers x to y from a description of a profile on $\{x, y\}$. (The domain restriction, or the measurability condition, ensures that a profile restricted to $\{x, y\}$ is describable by a natural number.)

The impossibility theorem suggests relaxing the notion of computability. One can relax ("relativize") the notion by allowing an algorithm to use an "oracle" from a well-known class of successively more complex sets (oracles), $\emptyset$, $\emptyset'$, $\emptyset''$, …, in recursion theory.[2] Then the impossibility theorem can be paraphrased: there is no social welfare function computable relative to $\emptyset$ that satisfies all of Arrow's conditions. It is an open question whether there is such a function computable relative to $\emptyset'$.

I show however the main result (Theorem 2 in Section 4.1) that there is one computable relative to $\emptyset''$; that is, there exists a social welfare function satisfying Arrow's conditions as well as the relaxed notion of *Strong Pairwise Computability 2*. In this possibility result, the oracle $\emptyset''$ that an algorithm (oracle algorithm) is allowed to use can be replaced by a certain oracle called Fin, which can tell for each natural number u whether u is an "index" of a finite set or not. Strong Pairwise Computability 2 therefore means that deciding the social preference is not harder than the problem of deciding

[2] These are the empty set, its "jump," the "second jump," and so forth. The hierarchy consisting of these sets is closely related to the *arithmetical hierarchy*, a different hierarchy of sets according to the quantifier complexity in their syntactic definition (Soare, 1987).

whether a given number is an index of a finite set or not.

Since the main objective of the paper is to constructively resolve Arrow's impossibility (1963), I prove (Appendix B) the main theorem through constructing (Section 4.2) a concrete function satisfying the conditions. Constructing the function would make the "invisible dictator" more "visible" than the existing literature (Fishburn, 1970; Kirman and Sondermann, 1972; Armstrong, 1980; Armstrong, 1985; Lauwers and Van Liedekerke, 1995) did. Furthermore, I sketch (Section 4.3) the process of computing the function. In addition to facilitating an understanding of the proof, the sketch should make the "invisible dictator" even more "visible." I conclude the main body of the paper with a warning that Theorem 2 should not be viewed as a result about absolute computability of social welfare functions.

2 Framework

The framework is the same as that of Mihara (1997; 1995), which is based on Armstrong (1980).

Let $N = \mathbf{N} = \{0, 1, 2, 3, \ldots\}$ be a countable set of individuals (voters), denoted by natural numbers. A countable set of individuals arises naturally in social choice. For example, a finite number of people facing countably many states of the world can be expressed (Mihara, 1997; Mihara, 1995) as a countable number of (state-specific) individuals.

I assume that a *planner* (a human or machine that computes a social welfare function) can observe only a certain class of coalitions (sets of individuals). Since I am concerned with algorithmic properties of a social welfare function, I assume that the observable coalitions consist of the class REC of *recursive* (algorithmically decidable) coalitions. According to Church's thesis (Soare, 1987), these are the coalitions for which there is an algorithm that can decide for (the name of) each individual whether she is in the coalition. (See Appendix A for a more precise definition of recursive sets.) 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.

Let X be a set of alternatives, which has at least three elements. Let $\mathcal{P}$ be the set of (strict) preferences, i.e., asymmetric and negatively transitive binary relations on X. A *profile* is a list $\mathbf{p} = (\succ_i^{\mathbf{p}})_{i \in N} \in \mathcal{P}^N$ of individual preferences $\succ_i^{\mathbf{p}}$, $i \in N$. A weak preference $\succeq_i^{\mathbf{p}}$ is the negation of $\prec_i^{\mathbf{p}}$ (defined from $\succ_i^{\mathbf{p}}$ in the obvious manner), and the indifference relation $\sim_i^{\mathbf{p}}$ is the symmetric part of $\succeq_i^{\mathbf{p}}$. A profile $(\succ_i^{\mathbf{p}})_{i \in N}$ is REC-*measurable* (or simply, *measurable*) if for all x and y in X, the coalition $\{ i : x \succ_i^{\mathbf{p}} y \}$ that prefers x to y is recursive. Intuitively, these are the profiles that the planner can observe. (Though the measurability condition implies correlation of the preferences of different individuals, under a certain interpretation it simply means that each person can identify the set of states in which she prefers an alternative to another. A further justification of the condition is in my dissertation (Mihara, 1995).) Denote by $\mathcal{P}_{\mathrm{REC}}^N$ the set of all measurable profiles.

An REC-*social welfare function* is a function $\succ : \mathcal{P}_{\mathrm{REC}}^N \to \mathcal{P}$ mapping each measurable profile $\mathbf{p} = (\succ_i^{\mathbf{p}})_{i \in N}$ into a social preference $\succ(\mathbf{p}) = \succ^{\mathbf{p}}$. (Using the notation $\succ$ for a function would not cause a confusion since preferences are expressed in the form $\succ_i^{\mathbf{p}}$ or $\succ^{\mathbf{p}}$, with profile $\mathbf{p}$ always present as a superscript.) Social relations $\succeq^{\mathbf{p}}$, $\sim^{\mathbf{p}}$, $\prec^{\mathbf{p}}$, etc., are defined in the obvious manner.

I list Arrow's conditions for REC-social welfare functions:

Unanimity For any x, $y \in X$, and $\mathbf{p} \in \mathcal{P}_{\mathrm{REC}}^N$, if $\{ i : x \succ_i^{\mathbf{p}} y \} = N$, then $x \succ^{\mathbf{p}} y$.

Independence For any x, $y \in X$, and $\mathbf{p}$, $\mathbf{p}' \in \mathcal{P}_{\mathrm{REC}}^N$, if $x \neq y$ and $\succ_i^{\mathbf{p}} \cap \{x, y\}^2 = \succ_i^{\mathbf{p}'} \cap \{x, y\}^2$ for all $i \in N$, then $\succ^{\mathbf{p}} \cap \{x, y\}^2 = \succ^{\mathbf{p}'} \cap \{x, y\}^2$.

Nondictatorship There is no $i \in N$ such that for all x, $y \in X$ and all $\mathbf{p} \in \mathcal{P}_{\mathrm{REC}}^N$, $x \succ_i^{\mathbf{p}} y \Longrightarrow x \succ^{\mathbf{p}} y$.

An REC-social welfare function violating Nondictatorship is called *dictatorial.*

3 Pairwise Computability

According to the main theorem (same as Proposition 1 below) of my earlier paper (Mihara, 1997), no social welfare function satisfies Arrow's conditions of Unanimity, Independence, and Nondictatorship if it also satisfies a certain condition of algorithmic computability, called Pairwise Computability. Before introducing a new notion of relative computability in the next section, I briefly review Pairwise Computability for comparison.

The notion of Pairwise Computability is based on Turing computability. *Turing computability* is (one of several equivalents of) the generally accepted formalization of the intuitive notion of algorithmic computability. Informally, an algorithm is a finite list of instructions that, given a symbolic input, yields after a finite number of steps a symbolic output. According to this intuition, a computation by an algorithm is exact, deterministic and performed in a discrete manner. Also, inputs and outputs are describable by natural numbers. Turing computability meets all these intuitive requirements.

Suppose that $\succ$ is an REC-social welfare function satisfying Independence. Pairwise Computability is a local requirement defined for such a function. It is concerned about how to obtain, for each pair (x, y), the social preference on (x, y) from a description of a measurable profile $\mathbf{p} \in \mathcal{P}_{\text{REC}}^N$ restricted to the set $\{x, y\}$. To describe the restriction $(\succ_i^{\mathbf{p}} \cap \{x, y\}^2)_{i \in N}$ of the profile $\mathbf{p}$ to the set by a natural number e, I use a triple of characteristic indices. Here, a *characteristic index* of a recursive set A is the Gödel number (name) of an algorithm computing the characteristic function of A. Given a characteristic index of A, one can effectively obtain the algorithm, which can determine for any natural number u, whether u is in A. Recall from Appendix A that $e = \langle e_1, e_2, e_3 \rangle$ is the coding (by integer) of a triple (e_1, e_2, e_3) of integers.

Definition 1 A natural number $e = \langle e_1, e_2, e_3 \rangle$ *represents* a profile $\mathbf{p} = (\succ_i^{\mathbf{p}})_{i \in N} \in \mathcal{P}_{\text{REC}}^N$ *at* a pair $(x, y) \in X \times X$ if e_1, e_2, and e_3 are characteristic indices for $\{\, i : x \succ_i^{\mathbf{p}} y \,\}$, $\{\, i : y \succ_i^{\mathbf{p}} x \,\}$, and $\{\, i : x \sim_i^{\mathbf{p}} y \,\}$ respectively.

Note that each natural number represents at most one restricted profile.

The following two definitions of computability require that the process of determining whether $x \succ^{\mathbf{p}} y$ or not (i.e., $\neg x \succ^{\mathbf{p}} y$), be an algorithmic process; they both use as input a representation e of the restricted profile. Pairwise Computability allows different algorithms to be used for different pairs (x, y). Strong Pairwise Computability, a stronger condition introduced for comparison, requires a single algorithm to work for all pairs.

Pairwise Computability For each pair $(x, y) \in X^2$, there exists a partial recursive function γ such that
$(*)$ for each profile $\mathbf{p} \in \mathcal{P}_{\text{REC}}^N$ and for each natural number e, if e represents $\mathbf{p}$ at (x, y), then

$$x \succ^{\mathbf{p}} y \Longrightarrow \gamma(e) = 1, \text{ and}$$
$$\neg x \succ^{\mathbf{p}} y \Longrightarrow \gamma(e) = 0.$$

Strong Pairwise Computability There exists a partial recursive function γ such that for each pair $(x, y) \in X^2$, condition $(*)$ in Pairwise Computability is satisfied.

Proposition 1 (Mihara (1997, Theorem 1)**)** *Let* $\succ : \mathcal{P}_{\mathrm{REC}}^N \to \mathcal{P}$ *be an* REC-*social welfare function satisfying Unanimity and Independence. If* $\succ$ *satisfies Pairwise Computability, then it is dictatorial.*

4 The main result

4.1 Relativized Pairwise Computability

Proposition 1 suggests weakening the notion of Pairwise Computability. Using a certain "oracle," I now introduce a new notion of relative computability for a social welfare function and show (Theorem 2) that there is a social welfare function satisfying the relative computability and Arrow's conditions.

The following notion (Strong Pairwise Computability 2) of relative computability is similar to Strong Pairwise Computability, except that use of a certain oracle and an *oracle algorithm* is allowed instead of an algorithm. To understand the nature of an oracle algorithm and its computation, suppose that we are given the oracle for a set A, which can tell for each number u whether u belongs to A or not. Informally, an *oracle algorithm* is an analogue of an algorithm except that it is allowed to ask questions of the form "Does a number v belong to the oracle?" during computation.

In the following definition, the oracle $\emptyset''$ that I use can be replaced by an oracle called Fin, which is recursively isomorphic (Appendix A.4) to $\emptyset''$. To define Fin, I introduce a few terms. A *recursively enumerable* (r.e.) set is a set of natural numbers whose elements can be listed (generated) by some algorithm. An *r.e. index* of an r.e. set A is the Gödel number of an algorithm that computes a partial function whose domain is A. Formally, the oracle Fin is the set of r.e. indices (descriptions) of finite sets. In the following notion of relative computability, therefore, an oracle algorithm may ask finitely many times during a computation whether a certain number is an r.e. index of a finite set or not. As before, I suppose that $\succ$ is an REC-social welfare function satisfying Independence.

Strong Pairwise Computability 2 (SPC2) There exists a $\emptyset''$-partial recursive function γ such that for each pair $(x, y) \in X^2$, for each profile $\mathbf{p} \in \mathcal{P}_{\mathrm{REC}}^N$, and for each natural number e, if e represents $\mathbf{p}$ at (x, y), then

$$x \succ^{\mathbf{p}} y \Longrightarrow \gamma(e) = 1, \text{ and}$$
$$\neg x \succ^{\mathbf{p}} y \Longrightarrow \gamma(e) = 0.$$

(The number "2" in the name of the condition simply indicates that the *second* jump $\emptyset''$ of the empty set $\emptyset$ is used as an oracle.) Note that SPC2 weakens Strong Pairwise Computability by using the oracle. SPC2 and Pairwise Computability are incomparable.

The following is the main theorem, a positive one. It establishes the existence of a social welfare function satisfying (i) Arrow's conditions and (ii) Strong Pairwise Computability 2. While the *existence* of a social welfare function satisfying only Arrow's conditions can be derived from an earlier result by Armstrong (1980), his result does not give a concrete example of such a function. For this reason, it is significant that the theorem will be proved (Appendix B) for a particular function explicitly constructed. The function is described in Section 4.2. (It is of mathematical interest to know whether Theorem 2 still holds when the oracle $\emptyset''$ is replaced by the less complex oracle $\emptyset'$. The question is left open.)

Theorem 2 *There is an* REC-*social welfare function* $\succ : \mathcal{P}^N_{\mathrm{REC}} \to \mathcal{P}$ *satisfying Unanimity, Independence, Nondictatorship, and Strong Pairwise Computability 2.*

4.2 Construction of the social welfare function

I now construct the social welfare function $\succ$ of Theorem 2. The proof will show that it satisfies the conditions in the theorem. The function is presented here in a more straightforward way than in the proof. Presenting it at this point should give the reader a clearer picture of the function and a better understanding of the proof.

To define the function $\succ$, I first construct a collection $\mathcal{U}$ of recursive coalitions. Each coalition in the collection is understood to be a "majority" of the individuals. First, note that there is (Lemma 2) an enumeration $W_{r(0)}, W_{r(1)}, W_{r(2)}, \ldots$ of all recursive coalitions. (Here, r is the function in the lemma.) Next, define a sequence

$$\begin{aligned} \mathcal{U}_0 &= \emptyset \\ \mathcal{U}_{s+1} &= \begin{cases} \mathcal{U}_s \cup \{W_{r(s)}\} & \text{if this family has an infinite intersection,} \\ \mathcal{U}_s & \text{otherwise.} \end{cases} \end{aligned}$$

Here, $\mathcal{U}_0$ is the empty set; the condition for the first case in the definition of $\mathcal{U}_{s+1}$ means that $(\bigcap \mathcal{U}_s) \cap W_{r(s)}$ is infinite, where $\bigcap \mathcal{U}_s = \bigcap_{W \in \mathcal{U}_s} W$. Finally, let $\mathcal{U} = \bigcup_{s=0}^{\infty} \mathcal{U}_s$.

Given the collection $\mathcal{U}$ of "majorities," I now define a map $\succ$ on $\mathcal{P}_{\mathrm{REC}}^N$ for $\mathbf{p} = (\succ_i^{\mathbf{p}})_{i \in N} \in \mathcal{P}_{\mathrm{REC}}^N$ and x, $y \in X$ by

$$x \succ^{\mathbf{p}} y \iff \{\, i : x \succ_i^{\mathbf{p}} y \,\} \in \mathcal{U}. \tag{1}$$

I will show in the proof that the map $\succ$ is a well-defined REC-social welfare function satisfying Arrow's conditions. Intuitively, the society prefers an alternative x to another alternative y exactly if the coalition that prefers x to y forms a "majority."

4.3 Oracle computation of the function

To show the role of the oracle Fin in computing the social welfare function constructed in the lengthy proof, I now sketch how to decide whether the society prefers an alternative x to another alternative y, from a representation e of a profile $\mathbf{p}$ at (x, y). (Those who are conversant with recursion theory may find this sketch sufficient for establishing Strong Pairwise Computability 2.) Recall that the oracle Fin can tell whether a given number is an r.e. index of an infinite set or not.

In the following, $g(0)$ is a pre-specified r.e. index of N. First, given the representation $e = \langle e_1, e_2, e_3 \rangle$, compute (via a characteristic index e_1 of the recursive set $\{\, i : x \succ_i^{\mathbf{p}} y \,\}$) an r.e. index u of $\{\, i : x \succ_i^{\mathbf{p}} y \,\}$ (this means $W_u = \{\, i : x \succ_i^{\mathbf{p}} y \,\}$). Now suppose that we are in step $s + 1$ and that $g(s)$ has been computed in the previous step. To compute $g(s + 1)$, we ask the oracle whether $W_{r(s)} \cap W_{g(s)}$ is infinite. ($W_{r(s)}$ is the sth recursive set in the above enumeration. By Lemma 2, r can be computed by an oracle algorithm using the oracle.) If the answer is Yes, let $g(s+1)$ be an r.e. index of $W_{r(s)} \cap W_{g(s)}$ (which can be obtained algorithmically). If the answer is No, let $g(s + 1) = g(s)$.

Since $W_u = \{\, i : x \succ_i^{\mathbf{p}} y \,\}$ is recursive and the sequence $r(0)$, $r(1)$, $r(2)$, $\ldots$ lists all r.e. indices of recursive sets, we have $r(s') = u$ at some step s', so that $W_{r(s')} = W_u$. In step $s' + 1$, which is the final step, we are going to determine whether $W_{r(s')} = \{\, i : x \succ_i^{\mathbf{p}} y \,\}$ is "large from the viewpoint of" the infinite $W_{g(s')}$, which has been obtained. More precisely, we ask the oracle whether $W_{r(s')} \cap W_{g(s')}$ is infinite. If Yes (so that $\{\, i : x \succ_i^{\mathbf{p}} y \,\}$ is "large from that point of view"), we decide that the society prefers x to y. If No, we decide that the society does not.

This completes the sketch of the decision problem. Note that we used only e_1 of the representation $e = \langle e_1, e_2, e_3 \rangle$. This means that for the decision problem, the social welfare function $\succ$ uses only the information about the coalition $\{\, i : x \succ_i^{\mathbf{p}} y \,\}$, ignoring the information about the other coalitions $\{\, i : y \succ_i^{\mathbf{p}} x \,\}$ and $\{\, i : x \sim_i^{\mathbf{p}} y \,\}$ that e gives. Furthermore, after we obtained an r.e. index u for $\{\, i : x \succ_i^{\mathbf{p}} y \,\}$, the characteristic index e_1 was not used any more. This suggests that to decide whether to socially prefer x to y using the oracle, the function $\succ$ only needs an r.e. index for the coalition $\{\, i : x \succ_i^{\mathbf{p}} y \,\}$, which index contains less information than a characteristic index for the coalition. (Mihara (1995, Appendix D) observes that computability is generally more difficult to attain with r.e. indices for $\{\, i : x \succ_i^{\mathbf{p}} y \,\}$ than with characteristic indices.) I will record the result, which is of independent interest and gives robustness to Theorem 2, as Proposition 5. The proof of Theorem 2 will be based on the proposition.

The above sketch of how to compute the social preference on a pair shows that the problem is computable *relative to* the oracle Fin. This means that the problem is not harder than deciding whether a given number is an r.e. index of a finite set or not. It does not mean that the problem is computable in the *absolute* sense. For this reason, from the viewpoint of computability analysis of social choice, Theorem 2 is best regarded as a contribution to the comparative study of the complexities of various rules.

For incorporating bounded perception, economic theory sometimes assumes that while an element of a set cannot always be identified, the class to which the element belongs can be. For example, a complete description of an individual's characteristic may not be identifiable, but the age and the sex may be. (In our case, on the contrary, a description—r.e. index—is already identified but not its characteristics.) It might therefore seem harmless to assume that such a seemingly innocent characteristic like finiteness can effectively be determined. However, the fact is that such an assumption is far from being innocent. In particular, Rice's Theorem (Appendix A) tells us that there is no algorithm to do this classification. (This does not necessarily mean that a person or some non-algorithmic machine cannot do the classification. What a person can do is open to philosophical debate. There seems to be no strong case, however, for supposing that one can always do it.) Hence Theorem 2 should be viewed not as a result about absolute computability but as a result comparing the complexity of the social decision problem with the well known mathematical problem.

A Recursion theory

This appendix reviews the definitions and results from recursion theory necessary for understanding the rest of the paper. I mostly follow the notations and terminologies in Soare (1987). My thesis contains an extended version (Mihara, 1995, Appendix A) of the appendix. Other references on recursion theory include Rogers (1987) and Davis and Weyuker (1983).

In this appendix, x, y and z denote nonnegative integers, and $\mathbf{N}$ is the set $\{0, 1, 2, \ldots\}$ of natural numbers. For sets A and B, A^c denotes the complement of A; $A - B$ denotes the set theoretic difference $A \cap B^c$.

A.1 Partial functions

A *partial function* on $\mathbf{N}^n$, where $n \geq 1$ is an integer, is a function (into natural numbers) whose domain is a subset of $\mathbf{N}^n$. If the domain of a partial function on $\mathbf{N}^n$ is $\mathbf{N}^n$, then it is called *total*. For partial functions ϕ and θ, $\phi(x) \downarrow$ denotes that $\phi(x)$ is defined; $\phi(x) \uparrow$ denotes that $\phi(x)$ is undefined; $\phi = \theta$ denotes that for all x, $\phi(x) \downarrow$ iff $\theta(x) \downarrow$ and if $\phi(x) \downarrow$ then $\phi(x) = \theta(x)$; $\operatorname{dom} \phi$ denotes the domain of ϕ.

A.2 Algorithms

Informally, an *algorithm* (for a partial function ϕ on $\mathbf{N}$) is a finite list of instructions that, given an input x, yields an output $y = \phi(x)$ after a finite number of steps if $\phi(x)$ is defined. (It should not yield an output if $\phi(x)$ is undefined.) The algorithm must specify how to obtain each step in the computation from the previous steps and from the input. Informally, if a partial function is computed by an algorithm, it is called *partial recursive*.

We accept *Church's Thesis*, which identifies the informal class of algorithmically computable partial functions with the class of partial functions computable by a Turing program. *Turing programs* can be defined precisely, but for our purpose, it suffices to know that we can list all Turing programs in such a way that for any program we can mechanically find its place (the code number) in the list and conversely. We choose one such algorithmic listing (or coding or *Gödel numbering*) and fix it.

Suppose we were given an "oracle" for a set A, which can tell for each number x whether x belongs to A (in which case, we say that x *belongs to* the oracle) or not. (Of course, for certain sets, the behavior of such an "oracle" cannot be characterized by an algorithm. Formally, an *oracle* is just a set.) Informally, an *oracle algorithm* is an analogue of an algorithm except that it is allowed to ask questions of the form "Does a number y belong to the oracle?" during computation. Then, informally, a partial function is called *partial recursive in A* (A-partial recursive) if there is an oracle algorithm that A-*computes* it, i.e., if there is an oracle algorithm that computes it with the oracle for A.

A.3 Computability theory

Code (Gödel number) all Turing programs. For $e \in \mathbf{N}$, let $\varphi_e^{(n)}$ be the partial function of n variables computed by the eth Turing program. A partial function ϕ of n variable is *partial recursive* if $\phi = \varphi_e^{(n)}$ for some e. A partial recursive function is *recursive* if it is total. Write φ_e for $\varphi_e^{(1)}$.

A set $A \subseteq \mathbf{N}$ is *recursive* ($A \in \mathrm{REC}$) if the characteristic function of A is recursive. e is a *characteristic index* of A if φ_e is the characteristic function of A.

Let $W_e = \operatorname{dom} \varphi_e = \{\, x : \varphi_e(x) \downarrow \,\}$. A set $A \subseteq \mathbf{N}$ is *recursively enumerable* (r.e.) if $A = W_e$ for some e. W_e is the eth r.e. set. For an r.e. set A, $e \in \mathbf{N}$ is said to be an *r.e. index* of A if $A = W_e$. Each r.e. set has countably many r.e. indices.

Let Fin be $\{\, x : W_x \text{ is finite} \,\}$ and let Rec be $\{\, x : W_x \text{ is recursive} \,\}$.

The **Enumeration Theorem** states (Soare, 1987, p. 15) that there is a partial recursive function $\varphi_z^{(2)}$ of two variables such that $\varphi_z^{(2)}(e, x) = \varphi_e(x)$ for all e and x.

The **Parameter Theorem** (*s-m-n* **Theorem**) states (Soare, 1987, p. 16) that for every m, $n \geq 1$, there exists a one-to-one recursive function s_n^m of $m+1$ variables such that for all x, y_1, …, y_m,

$$\varphi^{(n)}_{s_n^m(x, y_1, \ldots, y_m)}(z_1, \ldots, z_n) = \varphi_x^{(m+n)}(y_1, \ldots, y_m, z_1, \ldots, z_n)$$

for all z_1, …, z_n.

The **Graph Theorem** states (Soare, 1987, p. 29) that a partial function is partial recursive iff its graph is r.e.

A set $A \subseteq \mathbf{N}$ is called an *index set* if for all x and y,

$$[x \in A \quad \& \quad \varphi_x = \varphi_y] \Longrightarrow y \in A.$$

Rice's Theorem states that if A is an index set such that $A \neq \emptyset$ and $A \neq \mathbf{N}$, then A is not recursive.

We let $\langle x, y \rangle$ denote the image of (x, y) under the standard pairing function $(x^2 + 2xy + y^2 + 3x + y)/2$, which is a one-to-one recursive function from $\mathbf{N} \times \mathbf{N}$ onto $\mathbf{N}$. Let $\langle x, y, z \rangle$ denote $\langle \langle x, y \rangle, z \rangle$.

We write $\varphi_{e,s}(x) = y$ if $x < s$, $y < s$, $e < s$ and y is the output of $\varphi_e(x)$ in less than s steps of the eth Turing program.

A.4 Relative computability

A partial function ϕ is *partial recursive in* a set $A \subseteq \mathbf{N}$ (*A-partial recursive*) if there is an oracle Turing program that A-computes ϕ. If ϕ is A-computed by the eth oracle Turing program, we write $\phi = \Phi_e^A$. The domain $\operatorname{dom} \Phi_e^A$ is denoted by W_e^A. If ϕ is partial recursive in A and total, it is *recursive in* A. A set B is *recursive in* A if its characteristic function is recursive in A. A set B is *recursively enumerable in* A (*r.e. in* A) if $B = W_e^A$ for some e.

The *jump* of A, denoted by A', is $K^A = \{ x : x \in W_x^A \}$. The *$n$th jump* of A is defined by: $A^{(0)} = A$, $A^{(m+1)} = (A^{(m)})'$. It can be shown that A' is r.e. in A but not recursive in A. In this sense, A' is more complex than A. Also, if A is r.e. in B and B is recursive in C, then A is r.e. in C.

We define the classes Σ_n and Π_n, which will form the *arithmetical hierarchy* of sets in $\mathbf{N}$. A set B is *in* Σ_0 (also, Π_0) if B is recursive. For $n \geq 1$, B is *in* Σ_n if there is a recursive relation R such that

$$x \in B \iff (\exists y_1)(\forall y_2)(\exists y_3) \cdots (Q y_n)\, R(x, y_1, \ldots, y_n),$$

where Q is $\exists$ if n is odd, and $\forall$ if n is even. B is *in* Π_n if $B^c \in \Sigma_n$. B is *in* Δ_n if $B \in \Sigma_n \cap \Pi_n$. B is *arithmetical* if $B \in \bigcup_n (\Sigma_n \cup \Pi_n)$.

Post's Theorem states, among other things, that for every $n \geq 0$, (i) $B \in \Sigma_{n+1}$ iff B is r.e. in $\emptyset^{(n)}$, and (ii) $B \in \Delta_{n+1}$ iff B is recursive in $\emptyset^{(n)}$. Thus, $\emptyset^{(n)} \in \Sigma_n - \Pi_n$ for all $n > 0$.

It is known that $\mathrm{Fin} = \{ x : W_x \text{ is finite} \}$ is *recursively isomorphic* to $\emptyset''$. That is, there is a one-to-one recursive function p from $\mathbf{N}$ onto $\mathbf{N}$ such that $p(\mathrm{Fin}) = \emptyset''$. Hence, Fin is in $\Sigma_2 - \Pi_2$.

A.5 Lemmas

The following three Lemmas will be used in the proof of Proposition 5.

Lemma 1 *There is a recursive function f such that $W_{f(x,y)} = W_x \cap W_y$* (Soare, 1987, II.1.9, p. 30).

Lemma 2 *There is a* $\emptyset''$*-recursive function* r *on* $\mathbf{N}$ *whose range* $\{ r(n) : n \in \mathbf{N} \}$ *is* $\mathrm{Rec} = \{ x : W_x$ *is recursive* $\}$.

Proof. Since $\mathrm{Rec} \in \Sigma_3$, it is r.e. in $\emptyset''$ by Post's Theorem. Hence, Rec is the range of some $\emptyset''$-recursive function r on $\mathbf{N}$. ∎

Lemma 3 *Let* $\mathrm{Inf} = \{ e : W_e$ *is infinite* $\}$. *The characteristic function* $\chi = \chi_{\mathrm{Inf}}$ *of* Inf *is* $\emptyset''$*-recursive.*

Proof. Since $\mathrm{Inf} \in \Pi_2$ (Soare, 1987, IV.3.2, p. 66), $\mathrm{Inf} \in \Delta_3$. Then, by Post's Theorem, Inf is recursive in $\emptyset''$, which means that the characteristic function of Inf is $\emptyset''$-recursive. ∎

B Proof of the main result

B.1 Decisive coalitions and ultrafilters

The proof of Theorem 2 uses Armstrong's results (1980; 1985) discussed in this section.

Let $\mathcal{B}$ be a Boolean algebra on a set I: it includes I and is closed under union, intersection, and complementation. A *filter* $\mathcal{F}$ on $\mathcal{B}$ is a family of sets in $\mathcal{B}$ satisfying: (i) $\emptyset \notin \mathcal{F}$; (ii) if $A \in \mathcal{F}$ and $A \subseteq B$, then $B \in \mathcal{F}$; (iii) if A, $B \in \mathcal{F}$, then $A \cap B \in \mathcal{F}$. We may think of a filter as a family of "large" sets. An *ultrafilter* is a filter $\mathcal{F}$ that satisfies: if $A \notin \mathcal{F}$, then $A^c \in \mathcal{F}$. If $\mathcal{F}$ is an ultrafilter, then $A \cup B \in \mathcal{F}$ implies that $A \in \mathcal{F}$ or $B \in \mathcal{F}$. For $\mathcal{B}$ containing all finite sets of I, we say an ultrafilter $\mathcal{F}$ is *fixed* if it is of the form $\mathcal{F} = \{ A \in \mathcal{B} : i \in A \}$ for some $i \in I$; otherwise, it is called *free* and does not contain any finite sets. Koppelberg (1989) gives an exposition of these notions.

I state the following Propositions for REC-social welfare functions, where REC is the Boolean algebra of recursive coalitions on the set $N = \{0, 1, 2, \ldots\}$ of individuals. The propositions were originally stated for more general social welfare functions.

Proposition 3 (Armstrong (1980, Proposition 3.2)**)** *Suppose an* REC*-social welfare function* $\succ$ *satisfies Unanimity and Independence. Then there is a unique ultrafilter* $\mathcal{U}_\succ$ *on* REC *such that for all* $\mathbf{p} = (\succ_i^{\mathbf{p}})_{i \in N} \in \mathcal{P}_{\mathrm{REC}}^N$ *and* x, $y \in X$,

$$\{ i : x \succ_i^{\mathbf{p}} y \} \in \mathcal{U}_\succ \Longrightarrow x \succ^{\mathbf{p}} y. \tag{2}$$

Proposition 4 (Armstrong (1980, Proposition 3.1)**)** *Suppose* $\mathcal{U}$ *is an ultrafilter on* REC*. Then the map* $\succ$ *on* $\mathcal{P}_{\mathrm{REC}}^N$ *defined for* $\mathbf{p} \in \mathcal{P}_{\mathrm{REC}}^N$ *and* x*,* $y \in X$ *by*

$$x \succ^{\mathbf{p}} y \iff \{\, i : x \succ_i^{\mathbf{p}} y \,\} \in \mathcal{U}$$

is an REC*-social welfare function, satisfying Unanimity and Independence. Furthermore, if* $\mathcal{U}$ *is free, then* $\succ$ *satisfies Nondictatorship.*

Armstrong calls a free ultrafilter on a Boolean algebra of coalitions an *invisible dictator*.

B.2 Deciding decisive coalitions from r.e. indices

Proposition 3 implies that the decisive coalitions of a social welfare function satisfying Unanimity and Independence form an ultrafilter. In my thesis (Mihara, 1995, Appendix D), I investigated various algorithmic methods of deciding from an r.e. index of a recursive coalition whether or not the coalition belongs to the ultrafilter. In the thesis, starting with strong conditions of decidability of coalitions belonging to the ultrafilter, I successively relaxed the conditions to get more positive results. I discuss in this section one of the most positive results there. The result (Proposition 5 below), which is of independent interest, will be used to prove Theorem 2 in the next section.

Let $\succ : \mathcal{P}_{\mathrm{REC}}^N \to \mathcal{P}$ be an REC-social welfare function *satisfying Unanimity and Independence*. Given $\succ$, let $\alpha_\succ$ be the partial function on $N = \mathbf{N}$ defined by

$$\alpha_\succ(e) = \begin{cases} 1 & \text{if } W_e \text{ is recursive and } W_e \in \mathcal{U}_\succ, \\ 0 & \text{if } W_e \text{ is recursive and } W_e \notin \mathcal{U}_\succ, \\ \uparrow & \text{if } W_e \text{ is nonrecursive,} \end{cases} \tag{3}$$

where W_e is the eth r.e. set (i.e., the domain of the eth partial recursive function) and $\mathcal{U}_\succ$ denotes the ultrafilter in Proposition 3. (e is an r.e. index for the set W_e.) Clearly, $\alpha_\succ$ is well-defined. Note that the domain of $\alpha_\succ$ is $\mathrm{Rec} = \{\, e : W_e \text{ is recursive} \,\}$, the index set for the class REC of recursive sets. Also, $\alpha_\succ(e) = \alpha_\succ(e')$ if $W_e = W_{e'}$. I use the indicator $\alpha_\succ$ to define the following condition (DDC2) of decidability of decisive coalitions. (The number "2" in the name of the condition indicates that the *second* jump $\emptyset''$ of $\emptyset$ is used as an oracle. There is no "DDC1" introduced. The condition is called "Computability 5" in my thesis (Mihara, 1995).)

Decidability of Decisive Coalitions 2 (DDC2) $\alpha_\succ$ has an extension to a $\emptyset''$-partial recursive function.

Proposition 5 *There is an* REC*-social welfare function* $\succ : \mathcal{P}^N_{\mathrm{REC}} \to \mathcal{P}$ *satisfying Unanimity, Independence, Nondictatorship, and DDC2.*

Remark 1 Indeed, we actually show below that $\alpha_\succ$ itself is $\emptyset''$-partial recursive. Note that $\alpha_\succ$ cannot be $\emptyset'$-partial recursive. (The domain Rec of $\alpha_\succ$ is known (Soare, 1987, IV.3.6, p. 66) to be Σ_3-complete. In particular, Rec is not in Σ_2. This implies that Rec is not r.e. in $\emptyset'$; if it were, then Rec would be in Σ_2 by Post's Theorem. Since its domain Rec is not r.e. in $\emptyset'$, α cannot be $\emptyset'$-partial recursive.)

$\alpha_\succ$ has no extension α' such that (i) it is recursive in $\emptyset''$, and (ii) $\alpha'(e) = 2$ if $\alpha_\succ(e) \uparrow$. For $(\mathrm{Rec})^c = \{ e : \alpha'(e) = 2 \}$ would be r.e. in $\emptyset''$ if such an α' existed; then $(\mathrm{Rec})^c$ would be Σ_3, implying $\mathrm{Rec} \in \Pi_3$; but $\mathrm{Rec} \notin \Pi_3$. $\diamondsuit$

Proof. We will first construct a free ultrafilter $\mathcal{U}$ on the Boolean algebra REC. Proposition 4 will then imply that the REC-social welfare function $\succ$ corresponding to $\mathcal{U}$ satisfies Unanimity, Independence, and Nondictatorship. We will finally establish that $\succ$ satisfies DDC2 by showing that $\alpha_\succ$ is $\emptyset''$-partial recursive. To show that, note that the indicator $\alpha_\succ$ in DDC2 is defined by letting $\mathcal{U}_\succ = \mathcal{U}$ in (3), since the unique ultrafilter $\mathcal{U}_\succ$ in Proposition 3 corresponding to the function $\succ$ is obviously $\mathcal{U}$.

Before constructing $\mathcal{U}$, let us note Lemmas 1–3. We let f, r, χ be the functions f, r, χ_{Inf} in the Lemmas.

First, we construct sets A, $B \subseteq \mathrm{Rec}$ which are r.e. in $\emptyset''$ as follows:

Stage $s = 0$. Let $A_0 = B_0 = \emptyset$. Choose $g(0)$ arbitrarily such that $W_{g(0)} = N$.

Stage $s + 1$. Suppose at stage $s \geq 0$, finite sets A_s, $B_s \subseteq \mathrm{Rec}$ and an infinite r.e. set $W_{g(s)}$ are defined. We construct finite sets A_{s+1}, B_{s+1} and an infinite r.e. set $W_{g(s+1)}$ at Stage $s+1$ as follows: (At Stage $s+1$, we decide whether to include $r(s)$ in A_{s+1} or B_{s+1}.) Check if $W_{r(s)} \cap W_{g(s)}$ is infinite. (That is, whether $f(r(s), g(s)) \in \mathrm{Inf}$; equivalently, whether $\chi[f(r(s), g(s))] = 1$.) This is effectively decidable with the oracle $\emptyset''$, by Lemmas 1–3. (See the proof of Lemma 6 for further details.) If the answer to this test is Yes, let $A_{s+1} = A_s \cup \{r(s)\}$, $B_{s+1} = B_s$, and $g(s+1) = f(r(s), g(s))$ (so that $W_{g(s+1)} = W_{r(s)} \cap W_{g(s)}$). If the answer is No, let $A_{s+1} = A_s$, $B_{s+1} = B_s \cup \{r(s)\}$, and $g(s+1) = g(s)$ (so that $W_{g(s+1)} =$

$W_{g(s)}$). We can see that $A_s \subseteq A_{s+1}$, $B_s \subseteq B_{s+1}$, $W_{g(s)} \supseteq W_{g(s+1)}$; and $r(s)$ belongs to either A_{s+1} or B_{s+1}, but not both. Also, by induction, we can show the following (assuming $\bigcap_{e\in\emptyset} W_e = N$):

Lemma 4 *For all s,*
(i) $W_{g(s)} = \bigcap_{e\in A_s} W_e$*;*
(ii) $W_{g(s)}$ *is infinite.*

Finally, let $A = \bigcup A_s$ and $B = \bigcup B_s$.

Lemma 5 (i) $A \cup B = \mathrm{Rec}$*;*
(ii) $A \cap B = \emptyset$.

Proof. (i) Obvious.

(ii) Suppose $e \in A \cap B$. Let $e' = e$ and let s', s be the least s', s such that $e' \in A_{s'+1}$ and $e \in B_{s+1}$, so that $e = e'$ is put into $A_{s'+1}$ at Stage $s'+1$ and into B_{s+1} at Stage $s+1$. Then $e' = r(s')$ and $e = r(s)$.

(*Case:* $s' < s$). At Stage $s'+1$, since $e' = r(s')$ is put into $A_{s'+1}$, $W_{r(s')} \cap W_{g(s')}$ must be infinite, and so $W_{g(s'+1)} = W_{r(s')} \cap W_{g(s')}$. Hence, $W_{r(s')} \supseteq W_{g(s'+1)}$. Now at Stage $s+1$, we are given an infinite $W_{g(s)}$ and check whether $W_{r(s)} \cap W_{g(s)}$ is infinite. But since $W_{g(s'+1)} \supseteq W_{g(s)}$ (because $s'+1 \le s$), it follows that $W_{r(s')} \supseteq W_{g(s'+1)} \supseteq W_{g(s)}$. Then,

$$W_{r(s)} \cap W_{g(s)} \supseteq W_{r(s')} \cap W_{g(s)} = W_{g(s)},$$

which is infinite; hence, $W_{r(s)} \cap W_{g(s)}$ is infinite. So $e = r(s)$ is put into A_{s+1}, not into B_{s+1}.

(*Case:* $s' \ge s$). At stage $s+1$, since $e = r(s)$ is put into B_{s+1}, $W_{r(s)} \cap W_{g(s)}$ must be finite. We have $W_{g(s)} \supseteq W_{g(s')}$ since $s \le s'$. At Stage $s'+1$,

$$W_{r(s')} \cap W_{g(s')} \subseteq W_{r(s)} \cap W_{g(s')} \subseteq W_{r(s)} \cap W_{g(s)},$$

which is finite. So, $e' = r(s')$ is not put into $A_{s'+1}$. $\Diamond$

Lemma 6 *A, B are r.e. in* $\emptyset''$*.*

Proof. Let g_A, g_B be defined as follows:

$$g_A(x,0) \equiv 0 \quad \text{and} \quad g_B(x,0) \equiv 0;$$

$$g_A(x,s+1) = \begin{cases} 1 & \text{if} \quad g_A(x,s)=1 \quad \text{or} \quad x = r(s) \;\&\; \chi[f(r(s),g(s))] = 1, \\ 0 & \text{otherwise;} \end{cases}$$

$$g_B(x,s+1) = \begin{cases} 1 & \text{if} \quad g_B(x,s)=1 \quad \text{or} \quad x=r(s) \ \& \ \chi[f(r(s),g(s))]=0, \\ 0 & \text{otherwise.} \end{cases}$$

Then, by mathematical induction on s, we can easily obtain that

$$g_A(x,s)=1 \iff x \in A_s$$

and

$$g_B(x,s)=1 \iff x \in B_s.$$

For easy reference, we rewrite the definition of g:

$$g(0) \text{ is an arbitrary } e \text{ such that } W_e = N,$$

$$g(s+1) = \begin{cases} f(r(s),g(s)) & \text{if } \chi[f(r(s),g(s))]=1, \\ g(s) & \text{otherwise.} \end{cases}$$

Therefore, g is $\emptyset''$-recursive since g is obtained by *primitive recursion* (Soare, 1987, p. 9) from a function composed from $\emptyset''$-recursive functions f, r, and χ. It is then easy to see g_A and g_B are $\emptyset''$-recursive, since each of them is obtained by primitive recursion from $\emptyset''$-recursive functions.

Now we have

$$\begin{aligned} x \in A & \iff x \in A_s \text{ for some } s \\ & \iff \exists s\ [g_A(x,s)=1], \end{aligned}$$

which shows A is r.e. in $\emptyset''$, since $g_A(x,s)=1$ is a $\emptyset''$-recursive relation. Similarly for B. ♢

Now, let $\mathcal{U} = \{ W : W = W_e \text{ for some } e \in A \}$.

Lemma 7 *(i)* $e \in \text{Rec}$ & $W_e \in \mathcal{U} \iff e \in A$*;*
(ii) $e \in \text{Rec}$ & $W_e \notin \mathcal{U} \iff e \in B$*.*

Proof. (i) ($\Longleftarrow$). Trivial.

($\Longrightarrow$). Suppose $e \in \text{Rec}$ and $W_e \in \mathcal{U}$, but $e \notin A$. Then, by Lemma 5, $e \in B$; also, $W_e = W_{e'}$ for some $e' \in A$. Since $e, e' \in \text{Rec}$, $e = r(s)$, $e' = r(s')$ for some s, $s' \in N$. Let s, s' be the least such s, s'. Then $e = r(s)$ is put into B_{s+1} at Stage $s+1$ and $e' = r(s')$ is put into $A_{s'+1}$ at Stage $s'+1$. Follow the cases (*Case:* $s' < s$) and (*Case:* $s' \geq s$) in the proof of Lemma 5(ii).

(ii) ($\Longrightarrow$). Suppose $e \in \text{Rec}$ and $W_e \notin \mathcal{U}$. If $e \in A$, then $W_e \in \mathcal{U}$, by (i). So, $e \notin A$. Then $e \in B$ by Lemma 5.

($\Longleftarrow$). Let $e \in B$. Then $e \in \text{Rec}$ by Lemma 5. If $W_e \in \mathcal{U}$, then $e \in A$ by (i). So, $W_e \notin \mathcal{U}$. ♢

Lemma 8 *$\mathcal{U}$ is a free ultrafilter.*

Proof. (i) ($\mathcal{U}$ is a filter.) (a) To show $\emptyset \notin \mathcal{U}$, observe that if $W_{r(s)} = \emptyset$, then $W_{r(s)} \cap W_{g(s)}$ is finite (being empty). So, $r(s) \notin A$.

(b) Suppose $W_{e'}$, W_e are recursive, $W_{e'} \in \mathcal{U}$ and $W_e \supseteq W_{e'}$. To show $e \in A$ (hence $W_e \in \mathcal{U}$), suppose $e \in B$ for contradiction. Let s, s' be the least s, s' such that $e = r(s)$ and $e' = r(s')$. Then $e = r(s)$ is put into B_{s+1} at Stage $s+1$ and $e' = r(s')$ is put into $A_{s'+1}$ at Stage $s'+1$. Follow the cases (*Case:* $s' < s$) and (*Case:* $s' \geq s$) in the proof of Lemma 5(ii).

(c) Suppose recursive sets W_e, $W_{e'}$ are in $\mathcal{U}$. We show that $W_e \cap W_{e'} \in \mathcal{U}$. Since e, $e' \in A$ by Lemma 7(i), e, $e' \in A_s$ for some s. Choose $\bar{s}$ such that $W_{r(\bar{s})} = W_e \cap W_{e'}$ and $\bar{s} > s$. This is possible since $W = W_e \cap W_{e'}$ is recursive and has infinitely many r.e. indices. Now,

$$W_e, W_{e'} \supseteq \bigcap_{e \in A_s} W_e = W_{g(s)}.$$

But since $s < \bar{s}$, $W_{g(s)} \supseteq W_{g(\bar{s})}$. So,

$$W_{r(\bar{s})} = W_e \cap W_{e'} \supseteq W_{g(s)} \supseteq W_{g(\bar{s})}.$$

At Stage $\bar{s}+1$, given an infinite $W_{g(\bar{s})}$, we check whether $W_{r(\bar{s})} \cap W_{g(\bar{s})}$ is infinite. But it is infinite since it equals $W_{g(\bar{s})}$ because $W_{r(\bar{s})} \supseteq W_{g(\bar{s})}$. Hence $r(\bar{s}) \in A_{\bar{s}+1}$, so $W_{r(\bar{s})} \in \mathcal{U}$. That is, $W_e \cap W_{e'} \in \mathcal{U}$.

(ii) (The filter $\mathcal{U}$ is an ultrafilter.) Suppose W_e, $W_{e'}$ are recursive, they are complements of each other, and W_e, $W_{e'} \notin \mathcal{U}$. Choose s, s' such that $s < s'$ (by symmetry, we can do so without loss of generality) and $e = r(s)$, $e' = r(s')$. Since e, $e' \notin A$, both $W_{r(s)} \cap W_{g(s)}$ and $W_{r(s')} \cap W_{g(s')}$ are finite. Since $s < s'$, $W_{g(s)} \supseteq W_{g(s')}$. Now $W_{r(s)} \cap W_{g(s')} \subseteq W_{r(s)} \cap W_{g(s)}$ is finite, since the right hand side is finite. We then have

$$\begin{aligned} W_{g(s')} &= N \cap W_{g(s')} \\ &= (W_{r(s)} \cup W_{r(s')}) \cap W_{g(s')} \\ &= (W_{r(s)} \cap W_{g(s')}) \cup (W_{r(s')} \cap W_{g(s')}), \end{aligned}$$

which is finite, contradiction to Lemma 4(ii).

(iii) ($\mathcal{U}$ is free.) Suppose $W_e \in \mathcal{U}$ Then by Lemma 7(i), $e \in A_s$ for some s. So, $W_e \supseteq \bigcap_{e \in A_s} W_e = W_{g(s)}$. But $W_{g(s)}$ is infinite by Lemma 4(ii). This shows any element of $\mathcal{U}$ must be infinite. So, $\mathcal{U}$ cannot be principal. $\diamondsuit$

Therefore, the social welfare function $\succ$ corresponding to $\mathcal{U}$ (Proposition 4) satisfies Nondictatorship as well as Unanimity and Independence.

We have, by (3) and Lemma 7,

$$\begin{aligned}\alpha_{\succ}(e) = y \quad &\iff \quad (y = 1 \ \& \ e \in \text{Rec} \ \& \ W_e \in \mathcal{U}) \vee (y = 0 \ \& \ e \in \text{Rec} \ \& \ W_e \notin \mathcal{U}) \\ &\iff \quad (y = 1 \ \& \ e \in A) \vee (y = 0 \ \& \ e \in B).\end{aligned}$$

This shows graph $\alpha_{\succ}$ is r.e. in $\emptyset''$ since A, B are r.e. in $\emptyset''$ by Lemma 6. It follows that $\alpha_{\succ}$ is $\emptyset''$-partial recursive by the Graph Theorem. ∎

B.3 The proof

This section gives the proof of Theorem 2.

Let $\succ$ be the REC-social welfare function constructed in the proof of Proposition 5; the function is the same as that constructed in Section 4.2. It satisfies Unanimity, Independence, Nondictatorship, and DDC2. Let $\alpha = \alpha_{\succ}$, $\mathcal{U} = \mathcal{U}_{\succ}$. ($\alpha_{\succ}$ is defined by (3) in Appendix B.2 and $\mathcal{U}_{\succ}$ is given in Proposition 3.) Note that α is $\emptyset''$-partial recursive. Let CRec be defined by $x \in$ CRec iff x is a characteristic index of a recursive set.

First, using the Parameter Theorem (Appendix A.3), define a recursive f such that

$$\varphi_{f(e)}(u) = \begin{cases} 1 & \text{if } \varphi_e(u) = 1, \\ \uparrow & \text{otherwise.} \end{cases}$$

Proof. The partial function ψ defined by

$$\psi(e, u) = \begin{cases} 1 & \text{if } \varphi_e(u) = 1, \\ \uparrow & \text{otherwise} \end{cases}$$

is partial recursive by the Enumeration Theorem. Hence $\psi = \varphi_z^{(2)}$ for some z. By the Parameter Theorem, there is a one-to-one recursive function s such that

$$\varphi_{s(z,e)}(u) = \varphi_z^{(2)}(e, u) = \psi(e, u).$$

Let $f(e) = s(z, e)$. Then f is recursive. ♢

Lemma 9 *Let* $e \in$ CRec. *Then* $f(e) \in$ Rec *and* e *is a characteristic index for* $W_{f(e)}$.

Proof. Trivial. ♢

Lemma 9 means that if e_1 is a characteristic index for a recursive set A, then $f(e_1)$ is an r.e. index for A.

Define

$$\beta(e) = \begin{cases} 1 & \text{if } e \in \text{CRec} \ \& \ \alpha(f(e)) = 1, \\ 0 & \text{if } e \in \text{CRec} \ \& \ \alpha(f(e)) = 0, \\ 2 & \text{otherwise.} \end{cases}$$

(For the purpose of proving the theorem, totality of β is not essential; we could do as well by replacing the 2 in the third case by $\uparrow$.)

Lemma 10 *β is $\emptyset''$-recursive.*

Proof. Since

$$e \in \text{CRec} \iff (\forall x)(\exists s)[\varphi_{e,s}(x) = 1 \ \vee \varphi_{e,s}(x) = 0],$$

it follows that CRec $\in \Pi_2 \subseteq \Delta_3$. So, CRec is recursive in $\emptyset''$. β can therefore be $\emptyset''$-computed by giving the value 2 when $e \notin$ Crec and the value of $\alpha(f(e))$ (which converges to 1 or 0 by Lemma 9 and the definition of α) when $e \in$ Crec. $\diamondsuit$

By the definition of α,

$$\alpha(f(e)) = \begin{cases} 1 & \text{if } f(e) \in \text{Rec and } W_{f(e)} \in \mathcal{U}, \\ 0 & \text{if } f(e) \in \text{Rec and } W_{f(e)} \notin \mathcal{U}, \\ \uparrow & \text{otherwise.} \end{cases}$$

It follows from Lemma 9 that

$$\beta(e) = \begin{cases} 1 & \text{if } e \text{ is a characteristic index of a recursive set in } \mathcal{U}, \\ 0 & \text{if } e \text{ is a characteristic index of a recursive set not in } \mathcal{U}, \\ 2 & \text{otherwise.} \end{cases} \quad (4)$$

Define a function γ by

$$\gamma(\langle e_1, e_2, e_3 \rangle) = \beta(e_1).$$

Then, for any natural number $e = \langle e_1, e_2, e_3 \rangle$, $\gamma(e) = \beta(\pi(e))$, where $\pi\colon e \mapsto e_1$. Since π is recursive and β is $\emptyset''$-recursive, γ is $\emptyset''$-recursive.

It suffices to show that for all (x, y), γ satisfies the condition in SPC2. Fix (x, y) and suppose $e = \langle e_1, e_2, e_3 \rangle$ represents a $\mathbf{p}$ at (x, y). Then e_1 is a characteristic index of $\{ i : x \succ_i^{\mathbf{p}} y \}$.

- Suppose $x \succ^{\mathbf{p}} y$. Then by the definition of $\succ$ (Proposition 4), $\{ i : x \succ_i^{\mathbf{p}} y \} \in \mathcal{U}$, where $\mathcal{U}$ is constructed in the proof of Proposition 5. This implies by (4) that $\gamma(e) = \beta(e_1) = 1$.
- Similarly, if $\neg x \succ^{\mathbf{p}} y$ then $\gamma(e) = 0$. ∎

References

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

Armstrong, T. E., 1980, Arrow's Theorem with restricted coalition algebras, Journal of Mathematical Economics 7, 55–75.

Armstrong, T. E., 1985, Precisely dictatorial social welfare functions: Erratum and addendum to 'Arrow's Theorem with restricted coalition algebras', Journal of Mathematical Economics 14, 57–59.

Arrow, K. J., 1963, Social choice and individual values, 2nd edition (Yale University Press, New Haven).

Canning, D., 1992, Rationality, computability, and Nash equilibrium, Econometrica 60, 877–888.

Davis, M. D. and E. J. Weyuker, 1983, Computability, complexity, and languages: Fundamentals of theoretical computer science (Academic Press, San Diego).

Fishburn, P. C., 1970, Arrow's Impossibility Theorem: Concise proof and infinite voters, Journal of Economic Theory 2, 103–106.

Kalai, E., 1990, Bounded rationality and strategic complexity in repeated games, in: T. Ichiishi, A. Neyman, and Y. Tauman, eds., Game theory and applications (Academic Press, San Diego) 131–157.

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

Kirman, A. P. and D. Sondermann, 1972, Arrow's Theorem, many agents, and invisible dictators, Journal of Economic Theory 5, 267–277.

Koppelberg, S., 1989, Handbook of Boolean algebras, Vol. 1 (North-Holland, Amsterdam). Edited by J. D. Monk, with the cooperation of R. Bonnet.

Lauwers, L. and L. Van Liedekerke, 1995, Ultraproducts and aggregation, Journal of Mathematical Economics 24, 217–237.

Lewis, A. A., 1985, On effectively computable realizations of choice functions, Mathematical Social Sciences 10, 43–80.

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

Lipman, B. L., 1995, Information processing and bounded rationality: A survey, Canadian Journal of Economics 28, 42–67.

Mihara, H. R., 1995, Arrow's Theorem, Turing computability, and oracles, PhD thesis (University of Minnesota, Minneapolis).

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

Mihara, H. R., 1998, Existence of a coalitionally strategyproof social choice function: A constructive proof, Mimeo. A version available as ewp-pe/9604002 from the EconWPA.

Pazner, E. A. and E. Wesley, 1978, Cheatproofness properties of the plurality rule in large societies, Review of Economic Studies 45, 85–91.

Rogers, Jr., H., 1987, Theory of recursive functions and effective computability, paperback edition (MIT Press, Cambridge).

Soare, R. I., 1987, Recursively enumerable sets and degrees: A study of computable functions and computably generated sets (Springer-Verlag, Berlin).

Spear, S. E., 1989, Learning rational expectations under computability constraints, Econometrica 57, 889–910.

Wong, K., 1994, Economic equilibrium theory: A computability viewpoint, PhD thesis (University of Minnesota, Minneapolis).