ARROW'S THEOREM AND TURING COMPUTABILITY

by

H. Reiju Mihara

Discussion Paper No. 276, August 1994

Center for Economic Research
Department of Economics
University of Minnesota
Minneapolis, MN 55455

Arrow's Theorem and Turing Computability*

H. Reiju Mihara†
Department of Economics
University of Minnesota
Minneapolis, Minnesota 55455
U.S.A.
miha0008@gold.tc.umn.edu

August 1994

Abstract

A social welfare function for an infinite society satisfies *Pairwise Computability* if for each pair (x, y) of alternatives, there exists an algorithm that can decide from a description of a profile on $\{x, y\}$ whether the society prefers x to y. I prove that if a social welfare function satisfying Unanimity and Independence also satisfies Pairwise Computability, then it must be dictatorial. This result severely limits on practical grounds Fishburn's resolution (1970) of Arrow's impossibility. An interpretation of an infinite "society" is also given. *Journal of Economic Literature* Classifications: D71, C69, D89.

Keywords: Arrow impossibility theorem, Hayek's knowledge problem, algorithms, recursion theory, ultrafilters.

*An electronic version (LaTeX file) of this paper is available from the Working Paper Archive (econ-wp) at Washington University. To get the file, send an e-mail to ewp-pe@econwpa.wustl.edu with the command get 9408001 in the subject.

†This paper is based on Theorem 1 of my thesis [16]. I am indebted to my thesis adviser Marcel K. Richter and to Wayne Richter for their comments and guidance. Thanks are also due to the participants of the seminar on economic theory and foundations, spring 1994 at Minnesota, especially to KamChau Wong.

1 Introduction

1.1 Overview

Computability analysis of social choice is a consideration of social choice theory with explicit attention to algorithmic properties of social decision-making. It aims at identifying what social choice rules can be algorithmically executed, and at determining how complex such rules are. This paper reconsiders Arrow's Impossibility Theorem [3] from a viewpoint of computability analysis of social choice.

Arrow's Impossibility Theorem states that there is no social welfare function that satisfies the Unanimity, the Independence and the Nondictatorship axioms. Here, a *social welfare function* maps each profile of individual preferences into a "social preference." The preferences are defined on a set of at least three (social) alternatives and there are no restrictions on the preferences beyond the usual ordering properties. *Unanimity* says that when all individuals prefer an alternative x to another alternative y, then society must "prefer" x to y. *Independence* means that the only information relevant for determining "social preference" on a set $\{x, y\}$ is the individual preferences on the set. *Nondictatorship* rules out an individual such that whenever he prefers x to y, society must "prefer" x to y. Henceforth, I abuse language and apply the word "preferences" and related expressions to society as well as individuals without quoting them.

In this paper, I intend to study feasibility of centralized decision-making such as voting. I will interpret feasibility of executing a social choice rule by a central authority as algorithmic computability of the rule. This is in effect the same as regarding such an authority as an algorithm (or a digital computer) that computes the rule. Support for this comes from several sources. First, well-known schemes, such as simple majority rule, unanimity rule, and the Condorcet and the Borda rules, are all algorithms. Noncomputable social choice rules cannot be carried out systematically no matter how well-specified. (Kelly [10] gives examples of noncomputable social choice rules.) Second, the use of the language by social choice theorists suggests that the social welfare functions they consider are in fact computable ones. For example, Arrow defined a social welfare function to be a "process or rule" which, for each profile of individual preferences, "states" a corresponding social preference [3, p. 23], and called the function a "procedure" [3, p. 2]. Indeed, he later wrote [4, p. S398] in a slightly different context, "The next step in analysis, I would conjecture, is a more consistent assumption of computability in the formulation of economic hypotheses." Finally, there is a normative reason supporting algorithmic computability. Algorithmic social choice rules specify the procedures in such a way that the same results are obtained irrespective of who carries out a computation, leaving no room for personal judgments by the authority. In this sense, computability of social choice rules formalizes the notion of "due process."

However, social choice theory has not traditionally paid much attention to computability issues. This is understandable, for computability of social welfare functions is automatically satisfied by assuming a finite set of alternatives and a finite set of individuals, which is a common assumption in the literature. Computability is satisfied since, in such cases, a finite table can be constructed that expresses the function.

Therefore, as a modification to Arrow's setting, I discard the finite framework. Fishburn [7] and Kirman and Sondermann [11] showed that when there are infinitely many individuals in a society, there is a social welfare function satisfying the axioms of Unanimity, Independence, and Nondictatorship. Armstrong [1, 2] proved that this result is unaffected even when profiles are restricted to those measurable with respect to a Boolean algebra of coalitions.

I apply Armstrong's framework to a particular Boolean algebra of coalitions suitable for considering computability, namely the Boolean algebra of all recursive coalitions. These are coalitions for which there is an algorithm to decide the membership. So, the domain restriction requires that members of a coalition are algorithmically identifiable. We can name recursive coalitions using the Gödel numbers (codes) of these algorithms. We can then describe each measurable profile restricted on a set $\{x, y\}$ by giving names of (i) the coalition that prefers x to y, (ii) the coalition that prefers y to x, and (iii) the coalition that is indifferent between x and y.

Suppose a social welfare function satisfies Independence. The social welfare function for an infinite society satisfies *Pairwise Computability* if for each pair (x, y) of alternatives, there exists an algorithm that can decide, for each measurable profile of individual preferences and for each description of the profile on $\{x, y\}$, whether the society prefers x to y, from the description. I prove (Theorem 1) that if a social welfare function satisfying Unanimity and Independence also satisfies Pairwise Computability, then it must be dictatorial. Though a more desirable notion of computability (*Strong Pairwise Computability*) for a social welfare function satisfying Independence *can be* introduced, the negative content of Theorem 1 implies that strengthening the condition of computability is not very interesting. On the other hand, the proof of Proposition 3 shows by examples that while there are some dictatorial social welfare functions that satisfy Pairwise Computability (even Strong Pairwise Computability), not all do.

These notions of computability require neither finite set of alternatives, nor computable individual preferences. Also, the information necessary to carry out a computation is readily obtainable, being a description of a profile only on a set $\{x, y\}$. Theorem 1 severely limits on practical grounds Fishburn's resolution, re-establishing Arrow's negative result even for infinite societies.

Since speaking of infinitely many people involved in social choice might seem unrealistic, I discuss an example of an infinite "society" derived from only finitely many people. In particular, the example does not assume people extending into the indefinite future. I postpone a concrete example to Section 2.1.3 and give an abstract version here.

Consider a social choice problem in which there are finitely many people and there is uncertainty expressed by a countably infinite (i.e., denumerable) set of states of the world. Assume that the uncertainty cannot be resolved (i.e., it cannot be known which state will be realized) by the time social choice has to be made. Then, it is reasonable to suppose that people express their preferences conditioned on states: "person j prefers social alternative x to y if the state is s"—which is denoted by $x \succ_j^s y$.

An infinite "society" is derived from the society of people facing uncertainty as follows. Regard ***person*** j's preference $\succ_j^s$ in state s as the preference $\succ_{(j,s)}$ of the newly named ***individual*** (j, s). Since there are only finitely many people and denumerably many states, we have a denumerable set of individuals (j, s). This derivation of an infinite "society" as well as the domain restrictions might seem artificial. However, they are in fact natural and even have some advantages. First, inter-state comparisons are avoided, in the same sense that inter-personal comparisons are avoided in Arrow's setting. Second, in this formulation, people can express their preferences without estimating probabilities. Third, the domain restriction requires that each person can identify, for example, the set $\{ s : x \succ_j^s y \}$ of states in which he prefers x to y, for each x and y, which is a natural epistemological condition.

Hayek [9] points out that economic data, or knowledge, is ***dispersed.*** So, assuming data to be given to a single mind is a trivialization of social problems. This paper is still working within this "trivialized" problem, namely, informationally centralized decision-making. It gives an example of a "trivialized" problem that becomes impossible to solve once feasibility of information processing is formally required. In this sense, the paper strengthens Hayek's thesis.

The negative result (Theorem 1) suggests weakening the condition of computability to see if the resulting weak condition can be met by a social welfare function satisfying Arrow's axioms. A positive result, using oracle Turing programs, is discussed in my dissertation [16].

The paper is organized as follows. The rest of the Introduction surveys some related literature. The main results are stated in Section 3, which is preceded by informal discussion in Section 2. The technical reader can go directly to Section 3, which assumes a few terminologies in Appendix B on recursion theory (the study of algorithms). Appendix A discusses two notions of computability not covered in the main body. The proofs of the Propositions and the Theorem are given in Appendix C, where the knowledge of recursion theory covered in Appendix B is assumed.

1.2 Related Literature

The modern paradigm of social choice theory was opened by Arrow's Impossibility Theorem [3] in 1951. Surveys (e.g., Sen [18]) of later developments in social choice reveal that issues of computability in social choice have largely been ignored. Given that social choice theory has had a significant impact on philosophy and economics [8], this lack of interest is surprising: Philosophers have been concerned about algorithmic computability in their study of logical reasoning processes; economists have witnessed [12], among other things, the socialist calculation debate among Mises, Hayek [9], Lange, and Lerner.

Algorithmic (Turing) computability has been studied in related areas by Canning [5] in game theory, by Spear [20] on learning rational expectations, by Wong [21] in general equilibrium theory, and by Lewis [13] in individual choice theory.

In social choice theory, computability is studied from the recursion theoretic point of view by the following authors.

Kelly [10] considered computability of variable-voter social choice rules. He was interested in finding a *non*computable rule satisfying a subset of axioms characterizing the simple majority rule (which is computable), since he wanted to see which properties of the rule lead to computability.

The paper most closely related with the present study is Lewis [14], which is motivated by constructive mathematics. It discusses Arrow's Theorem "within the recursion theoretic setting." Our frameworks are different: although (i) the set of individuals he considers is the same as mine (the set of natural numbers), (ii) the set of alternatives in Lewis is a countable set that contains at least three elements, while my set of alternatives can be uncountable; (iii) his set of preferences is countable, while mine can be uncountable; (iv) Lewis restricts profiles to be recursive (i.e., there is an algorithm to determine for each individual and each pair of alternatives, his preference on the pair)—such profiles form a strict subclass of my REC-measurable profiles to be defined; (v) Lewis assumes that coalitions are recursively enumerable, while in my framework each coalition that prefers one alternative to another is recursive. Lewis states that in his "recursive" setting, there is a "dictator." But, in his theorems, he is using the word "dictator" in a much weaker sense than mine: in essence, he claims that for *each* profile, there is a "dictator" whose preference determines the social preference for that particular profile; by contrast, I prove existence of a single dictator for all profiles. (Although he presents the result without referring to the Unanimity or the Independence axioms, I suppose it is an oversight.)

2 Discussion

This Section is an informal exposition of the framework and the results in Section 3.

In the rest of this paper, I informally use the word person *(*people*) to refer to a person in the ordinary sense, a human being. The word* individual *is used in a technical sense.* An individual may be either a person or a name representing a person at a certain date or state.

2.1 Domain Restrictions

In this Section, I introduce for a social welfare function a domain suitable for consideration of computability issues in the setting of Arrow's Theorem. The treatment is based on Armstrong's extensions [1] of Arrow's Impossibility Theorem [3] and of Fishburn's resolution [7].

2.1.1 Naming Individuals

Since Arrow's impossibility persists for any finite set of individuals (this is a corollary of Proposition 4), I consider as a set I of (the names of) ***individuals*** one of the simplest infinite sets, namely the set $\mathbf{N}$ of nonnegative integers. A countably infinite set of individuals arises naturally in social choice.

For example, a countably infinite set of individuals may arise when a national government is evaluating alternative policies that can affect future generations. We can, for example, assign a name to each person as follows: if a person is a female, she is given a name starting from a nonzero even number; otherwise, an odd number. Likewise, economic, social, political classifications can be coded into a name.

Another example of a denumerable set of individuals is given by the case of finitely many people facing a set X of alternatives and uncertainty expressed by a denumerable set S of states of the world. This was discussed in the Introduction.

2.1.2 Coalitions

I assume that only some *coalitions* (sets of individuals) are observable. Intuitively, I think of the observations as being made by an agent, called a *social planner*, a human or machine that executes a social welfare function. Since an observation is a cognitive activity, it seems natural to introduce a structure for observable coalitions. Following Armstrong [1], I require that a family of observable coalitions form a *Boolean algebra.* Namely, if two coalitions are observable so must be their union, intersection, and complements. For instance, (i) the family of all subsets of $I = \mathbf{N}$ and (ii) the family of all finite or cofinite sets of $I = \mathbf{N}$ each forms a Boolean algebra.

Since I am concerned with algorithmic computability, I restrict coalitions to those which can be recognized by some algorithm. Then, we will see that (i) becomes too broad a family, being uncountable, while (ii) is unnecessarily restrictive, excluding the intuitively "describable" coalition of even numbers, for example. The observable coalitions that I propose are the *recursive* sets of individuals. These are the coalitions whose membership is effectively decidable, i.e., the ones for which there is an algorithm that can decide for each name i, whether individual i is in the coalition. This algorithmically decidable nature of recursive coalitions seems to capture the idea of what we mean by a coalition that is "observable" or "recognizable" or "identifiable." The recursive coalitions form a Boolean algebra.

2.1.3 Social Welfare Functions

A *social welfare function* (formally, an REC-social welfare function) maps a *profile* (list) $\mathbf{p} = (\succ_i^{\mathbf{p}})_{i \in I}$ of individual preferences $\succ_i^{\mathbf{p}}$ (on a set X of alternatives) to a social preference $\succ^{\mathbf{p}}$. I assume that the set I of individuals is $\mathbf{N}$, and that the domain includes all profiles $\mathbf{p}$ such that for any x and y, the *coalition* $\{ i : x \succ_i^{\mathbf{p}} y \}$ *that prefers* x *to* y is recursive. Such profiles are called *measurable* (with respect to the Boolean algebra REC of recursive coalitions). Measurable profiles are understood to be the ones for which the social planner will be required to give a social preference.

Example. Suppose the Administration of Food, Drug, Cosmetics, and Medical Devices (FDCA) is to adopt a usage policy for a newly developed medicine. The FDCA consults some selected set of people, or "experts" about their preferences among alternatives policies such as "prescription only" or "experimental use only on nonhumans." These policies form the set X of alternatives. The experts are not comfortable with giving definite answers since the medicine is new and so there are uncertainties that cannot be resolved by the time a policy has to be adopted. So, the FDCA decides to specify conditions on which experts' opinions might be based. These conditions constitute the set of states of the world that may involve with, for example, (i) the potential benefits s_1 of research for the next dozen years in case only experimental use is allowed or (ii) the cost s_2 in terms of human lives for the next decade in case a unrestrictive policy is adopted. Each state s specifies a value to these variables s_1 and s_2, among others. It is natural to assume that the set S of states is countably infinite. The discussion of finitely many people facing uncertainty (Section 1.1) applies to this example. $\diamondsuit$

2.2 Computability

Proposition 5 implies that if the set I of individuals is $\mathbf{N}$ and a social planner only observes measurable profiles (with respect to REC, the Boolean algebra of recursive coalitions), then a social welfare function (not necessarily computable) exists that satisfies Arrow's conditions. In this section, I introduce some modest conditions for computability of social welfare functions and consider whether there is a *computable* social welfare function satisfying Arrow's axioms.

2.2.1 Naming Restricted Profiles

My notions of computability will be weak in the sense that they are only local requirements: they are concerned about how to obtain, for each pair (x, y), the social preference on (x, y) from a description of a profile restricted to the set $\{x, y\}$. For this purpose, I describe the restriction $(\succ_i^{\mathbf{p}} \cap \{x, y\}^2)_{i \in \mathbf{N}}$ of a measurable profile $\mathbf{p} \in \mathcal{P}_{\mathrm{REC}}^{\mathbf{N}}$ to a set $\{x, y\}$ by a natural number e (as in Section 3.2). When this is done, I say that e ***represents*** $\mathbf{p}$ ***at*** (x, y). A natural number e is ***illegitimate*** if e does not represent any measurable profile at (any) (x, y); e is ***legitimate*** otherwise. (If e represents a profile at (x, y), then it represents all the profiles (at (x, y)) whose restriction to the set $\{x, y\}$ is identical. In this sense, each natural number describes at most one restricted profile.)

2.2.2 Two Notions of Computability

I only consider social welfare functions $\succ$ satisfying Independence (so that social preference on a pair $\{x, y\}$ is determined by the profile restricted to the pair).

I say that a social welfare function $\succ$ satisfies *Pairwise Computability* (PC) if for each pair (x, y), there exists an algorithm that can decide, for each measurable profile $\mathbf{p}$ and for each legitimate representation e of the profile at (x, y), whether the society prefers x to y according to $\succ^{\mathbf{p}}$, from the representation e. If such an algorithm can be chosen so that it works uniformly for all (x, y), then I say that the social welfare function satisfies *Strong Pairwise Computability*. Note that these computability conditions imply that the value given by a deciding algorithm must be invariant over different e that represent the same profile at (x, y). Also, note that PC does not require that a single algorithm work for all pairs, while Strong PC does. In this sense, the latter (which implies the former) is a more intuitively appealing notion of computability for a social welfare function. However, we will see that using even the weaker notion brings about an impossibility result. Proposition 1 gives a condition under which PC is equivalent with Strong PC.

The main result, Theorem 1, states that if a social welfare function satisfying Unanimity and Independence also satisfies PC, then it must be dictatorial. The Introduction interprets this result as strengthening both Arrow's Theorem and Hayek's thesis.

On the other hand, Proposition 3 shows that while there are some dictatorial social welfare functions that satisfy PC (even Strong PC), not all do. *Precisely dictatorial* social welfare functions (where social preference is always identical with the dictator's preference) are examples that do satisfy Strong PC.

3 Theorem

3.1 Framework

I is a set of *individuals*, which is either finite or infinite. An example of I is the set $\mathbf{N}$ of nonnegative integers. X is a set of *alternatives*, which has at least three elements. $\mathcal{P}$ is the set of (strict) *preferences*, i.e., asymmetric and negatively transitive binary relations on X.

A *Boolean algebra* $\mathcal{B}$ consisting of subsets of I satisfies the following: (i) $\emptyset$, $I \in \mathcal{B}$; (ii) $A \cup B$, $A \cap B$, $\overline{A} \in \mathcal{B}$ if A, $B \in \mathcal{B}$ (where $\overline{A}$ denotes the complement of A). If I denotes the set of individuals, then intuitively, an element of a Boolean algebra is a coalition observable by the planner. For example, let REC consist of all recursive subsets of $\mathbf{N}$. Then REC forms a Boolean algebra.

$\mathbf{p} = (\succ_i^{\mathbf{p}})_{i \in I} \in \mathcal{P}^I$ is called a *profile* of *individual preferences* $\succ_i^{\mathbf{p}}$, $i \in I$. 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 $(\succ_i^{\mathbf{p}})_{i \in I}$ is *$\mathcal{B}$-measurable* if $\{\, i \in I : x \succ_i^{\mathbf{p}} y \,\} \in \mathcal{B}$ for all x, $y \in X$. Denote by $\mathcal{P}_{\mathcal{B}}^I$ the set of all $\mathcal{B}$-measurable profiles.

A *$\mathcal{B}$-social welfare function* is a function $\succ : \mathcal{P}_{\mathcal{B}}^I \to \mathcal{P}$ mapping each profile $\mathbf{p} = (\succ_i^{\mathbf{p}})_{i \in I}$ to 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 $\mathcal{B}$-social welfare functions:

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

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

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

A $\mathcal{B}$-social welfare function violating Nondictatorship is called *dictatorial.*

Finally, I list one more condition for a $\mathcal{B}$-social welfare function.

Neutrality For all $x, x', y, y' \in X$, and all $\mathbf{p}, \mathbf{p}' \in \mathcal{P}_{\mathcal{B}}^I$, if $[(x \succ_i^{\mathbf{p}} y \leftrightarrow x' \succ_i^{\mathbf{p}'} y')$ & $(y \succ_i^{\mathbf{p}} x \leftrightarrow y' \succ_i^{\mathbf{p}'} x')]$ for all $i \in I$, then $(x \succ^{\mathbf{p}} y \leftrightarrow x' \succ^{\mathbf{p}'} y')$.

Neutrality requires that if a profile $\mathbf{p}$ treats a pair (x, y) in the same way as a profile $\mathbf{p}'$ treats a pair (x', y'), then the resulting two social preferences must treat the respective pairs in the same way. Clearly, Neutrality implies Independence.

3.2 Computability

I will define computability for social welfare functions using 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; inputs and outputs are finitely describable (equivalently, *describable by natural numbers*); and so on [17, pp. 1–5]. Turing computability meets all these intuitive requirements.

The basic idea of a social welfare function is that it maps each profile $\mathbf{p}$ to a social preference $\succ^{\mathbf{P}}$. So, when one accepts Turing computability, the first approach that one might attempt in introducing a condition of computability for social welfare functions is to represent profiles $\mathbf{p}$ and social preferences by integers, and then, to define computability in terms of these integers. This approach is unsatisfactory in general (for example, it is problematic even when I restrict attention to REC-social welfare functions) unless X is finite. The reason is that when X is infinite, the domain of a social welfare function is not necessarily countable (e.g., $\mathcal{P}_{\mathrm{REC}}^{\mathbf{N}}$ is uncountable [16]), while only countably many profiles $\mathbf{p}$ can be represented by a natural number. This implies a possibility that any algorithm used for obtaining social preferences fails to compute an output for "almost all" profiles in the domain. One way out of this problem would be to consider only countable domains for social welfare functions.

However, there is a different solution, which does not require countable domains. To describe the solution, I henceforth let the set I of individuals be the set $\mathbf{N}$ of nonnegative integers and I let the Boolean algebra $\mathcal{B}$ of coalitions be REC, the Boolean algebra of recursive sets. (So, I am considering only REC-social welfare functions $\succ : \mathcal{P}_{\mathrm{REC}}^{\mathbf{N}} \to \mathcal{P}$.)

A key assumption that I make for my solution is the Independence axiom: I suppose that $\succ$ is an REC-social welfare function satisfying Independence. Corresponding to each profile $\mathbf{p}$ is the social preference $\succ^{\mathbf{P}}$, which determines for each pair (x, y) of alternatives, whether $x \succ^{\mathbf{P}} y$ or not. By Independence, all that is needed to determine that, is the restriction of profile $\mathbf{p}$ to $\{x, y\}$. The Definition below introduces a method of representing such restricted profiles by a natural number e. Such representation (by integers) enables me to apply the notion of Turing computability. Notions of computability for social welfare functions will be introduced afterward.

Remark. Restricting my attention to REC-social welfare functions satisfying Independence is not serious since my main purpose is to determine whether there is a nondictatorial social welfare function among those satisfying Unanimity and Independence. Furthermore, the Independence axiom can be regarded as a part of the computability condition. After all, Independence is a stringent form of an informational viability condition, which requires finiteness in some aspects of information to be processed. $\diamondsuit$

A *characteristic index* of a recursive set A is the Gödel number (name) of an algorithm computing the characteristic function of A. When a characteristic index of A is known, one can effectively recover the algorithm from the Gödel number; using this algorithm, one can decide, for any number $e_1 \in \mathbf{N}$, whether e_1 is in A. Recall from Appendix B 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.

To describe the restriction of a measurable profile $\mathbf{p}$ on a pair $\{x, y\}$, I first describe each of $\{\, i : x \succ_i^{\mathbf{p}} y \,\}$, $\{\, i : y \succ_i^{\mathbf{p}} x \,\}$, and $\{\, i : x \sim_i^{\mathbf{p}} y \,\}$ by its characteristic index and then aggregate the three indices using the above coding for triples. Formally, I have the following definition:

Definition $e = \langle e_1, e_2, e_3 \rangle \in \mathbf{N}$ ***represents*** a profile $\mathbf{p} = (\succ_i^{\mathbf{P}})_{i \in \mathbf{N}} \in \mathcal{P}_{\mathrm{REC}}^{\mathbf{N}}$ ***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.

If e represents $\mathbf{p} \in \mathcal{P}_{\mathrm{REC}}^{\mathbf{N}}$ at (x, y), then e describes the restricted profile $(\succ_i^{\mathbf{P}} \cap \{x, y\}^2)_{i \in \mathbf{N}}$ of $\mathbf{p}$ on $\{x, y\}$ completely (in the sense that one and only one restricted profile corresponds to e). Note that I need at least two of the three characteristic indices above to describe the restricted profile completely.

The following two definitions of computability require that the process of determining whether $x \succ^{\mathbf{P}} y$ holds, 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), while Strong Pairwise Computability requires a single algorithm to work for all pairs.

Pairwise Computability (PC) For each pair $(x, y) \in X^2$, there is a partial recursive function γ such that
(a) for each profile $\mathbf{p} \in \mathcal{P}_{\mathrm{REC}}^{\mathbf{N}}$ and for each integer $e \in \mathbf{N}$, 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 (SPC) There is a partial recursive function γ such that for each pair $(x, y) \in X^2$, condition (a) in Pairwise Computability is satisfied.

Obviously, "$\Longrightarrow$" in (a) above can be replaced by "$\Longleftrightarrow$." But notice that e is restricted to those representing some $\mathbf{p} \in \mathcal{P}_{\mathrm{REC}}^{\mathbf{N}}$ at (x, y).

Remark. In order to appreciate the above notions, it is instructive to consider several other notions of computability. I give two alternative notions of computability in Appendix A. $\diamondsuit$

Note that Strong Pairwise Computability implies Pairwise Computability. The following Proposition, which is proved in Appendix C, shows that they are equivalent under Neutrality.

Proposition 1 *Let* $\succ: \mathcal{P}^{\mathbf{N}}_{\mathrm{REC}} \to \mathcal{P}$ *be an* REC-*social welfare function satisfying Neutrality. If* $\succ$ *satisfies Pairwise Computability, then it also satisfies Strong Pairwise Computability.*

In fact, the following Proposition, proved in Appendix C, shows that Neutrality is necessary for Strong Pairwise Computability, under the assumption of Independence.

Proposition 2 *Let* $\succ: \mathcal{P}^{\mathbf{N}}_{\mathrm{REC}} \to \mathcal{P}$ *be an* REC-*social welfare function satisfying Independence. Then* $\succ$ *satisfies Pairwise Computability and Neutrality if and only if it satisfies Strong Pairwise Computability.*

I now give the main theorem, whose proof appears in Appendix C. It reestablishes Arrow's negative result even for infinite societies.

Theorem 1 *Let* $\succ: \mathcal{P}^{\mathbf{N}}_{\mathrm{REC}} \to \mathcal{P}$ *be an* REC-*social welfare function satisfying Unanimity and Independence. Then* $\succ$ *is dictatorial if it satisfies Pairwise Computability.*

While Theorem 1 asserts necessity of dictatorship for computability, the next proposition shows that it is not sufficient.

Proposition 3 *Among the* REC-*social welfare functions satisfying Unanimity and Independence, there are (i) a dictatorial function satisfying Strong Pairwise Computability (and hence Neutrality), (ii) a dictatorial function satisfying Pairwise Computability but not Strong Pairwise Computability (and hence not Neutrality), and (iii) a dictatorial function not satisfying Pairwise Computability.*

Proof. Items (i), (ii), and (iii) are proved by Examples 1, 2, and 3, respectively. Details are in Appendix C. ∎

Example 1. Let

$$\mathcal{U}_0 = \{ A \in \mathrm{REC} : 0 \in A \}$$

and define by Proposition 5 the social welfare function $\succ: \mathcal{P}^{\mathbf{N}}_{\mathrm{REC}} \to \mathcal{P}$ for $\mathbf{p} \in \mathcal{P}^{\mathbf{N}}_{\mathcal{B}}$, and for $x, y \in X$ by

$$x \succ^{\mathbf{p}} y \iff \{ i : x \succ^{\mathbf{p}}_i y \} \in \mathcal{U}_0.$$

That is, the individual 0 is the "precise dictator." Proposition 5 establishes that $\succ$ satisfies Unanimity and Independence. Appendix C shows that $\succ$ satisfies Strong Pairwise Computability. $\diamondsuit$

Example 2. Let P be an arbitrary nonempty preference in $\mathcal{P}$. Let $\mathcal{U}_0$ be as in Example 1. Define a map $\succ$ from $\mathbf{p} \in \mathcal{P}_{\mathrm{REC}}^{\mathbf{N}}$ into a binary relation $\succ^{\mathbf{p}}$ on X as follows: for $\mathbf{p} \in \mathcal{P}_{\mathcal{B}}^{\mathbf{N}}$, and for $x, y \in X$,

$$x \succ^{\mathbf{p}} y \quad \text{iff} \quad \begin{array}{l} \text{(a) } \{ i : x \succ_i^{\mathbf{p}} y \} \in \mathcal{U}_0 \text{ or} \\ \text{(b) } \{ i : x \sim_i^{\mathbf{p}} y \} \in \mathcal{U}_0 \ \& \ xPy. \end{array}$$

I can show [15] that $\succ$ is a dictatorial REC-social welfare function satisfying Unanimity, Independence, but violates Neutrality. By Proposition 2, $\succ$ violates Strong Pairwise Computability. Appendix C shows $\succ$ satisfies Pairwise Computability. $\diamondsuit$

Example 3. Let $\mathcal{U}_0$ be as in Example 1 and let $\hat{\mathcal{U}}$ be an arbitrary free ultrafilter on REC. Define a map $\succ$ from $\mathbf{p} \in \mathcal{P}_{\mathrm{REC}}^{\mathbf{N}}$ into a binary relation $\succ^{\mathbf{p}}$ on X as follows: for $\mathbf{p} \in \mathcal{P}_{\mathcal{B}}^{\mathbf{N}}$, and for $x, y \in X$,

$$x \succ^{\mathbf{p}} y \quad \text{iff} \quad \begin{array}{l} \text{(a) } \{ i : x \succ_i^{\mathbf{p}} y \} \in \mathcal{U}_0 \text{ or} \\ \text{(b) } \{ i : x \sim_i^{\mathbf{p}} y \} \in \mathcal{U}_0 \ \& \ \{ i : x \succ_i^{\mathbf{p}} y \} \in \hat{\mathcal{U}}. \end{array}$$

It can be shown [2, 15] that $\succ$ is a dictatorial REC-social welfare function satisfying Unanimity and Independence (and in fact, Neutrality). The proof that $\succ$ does not satisfy Pairwise Computability appears in Appendix C. $\diamondsuit$

A Alternative Notions of Computability

A minor problem with the definitions of Pairwise Computability and Strong Pairwise Computability occurs when an input number e for a deciding algorithm (for a partial recursive function γ in (a) in PC or in SPC) for a social welfare function is illegitimate, so that it does not represent any measurable profile at the pair. In this case, application of the algorithm might give a social preference on (x, y) improperly. This problem is minor since I can safely think of a scenario in which a planner only processes inputs whose legitimacy she can prove. While there is no algorithmic procedure to give a proof of legitimacy for *every* input, there is no inconsistency in assuming that only numbers for which legitimacy can be proved are input to the social welfare function.

Having said that, let me consider some ways of avoiding obtaining social preferences for illegitimate inputs, for a planner might incorrectly believe that her input is legitimate. Computability A below requires illegitimate inputs to be indicated by a certain output; Computability B requires that outputs are given *only* for legitimate inputs. Though these notions may be appealing on intuitive grounds, they are both stronger than Pairwise Computability, and therefore, the main Theorem 1 applies *a fortiori*. However, I show here that *no* social welfare function satisfying Independence meets either of these computability conditions. These impossibility results further justify the use of PC or Strong PC as a notion of computability.

Suppose in the following that $\succ : \mathcal{P}^{\mathbf{N}}_{\text{REC}} \rightarrow \mathcal{P}$ is an REC-social welfare function satisfying Independence.

1. In the definition of Pairwise Computability, it is required that an algorithm exists that can decide the restricted social preference given a representation e of a profile at a pair of alternatives. However, there is no requirement as to what the algorithm should do for an integer $e \in \mathbf{N}$ that is illegitimate (i.e., that does not represent any REC-measurable profile at the pair). It would be desirable if the algorithm could decide for each integer e whether e is legitimate or not, in addition to deciding the restricted social preference. This leads to the following definition:

Computability A For each pair $(x, y) \in X^2$, there is a recursive function γ such that the condition (a) in Pairwise Computability is satisfied, and
(b) for each integer $e \in \mathbf{N}$, if e does not represent any $\mathbf{p} \in \mathcal{P}^{\mathbf{N}}_{\text{REC}}$ at (x, y), then $\gamma(e) = 2$.

Unfortunately, Computability A cannot be met by any REC-social welfare function that satisfies Independence. To see why, fix (x, y) and suppose a recursive γ satisfies (a) and (b). Then it must be that $S = \{ e : \gamma(e) = 1 \text{ or } 0 \}$ is r.e. (in fact, recursive). This is because S is the domain of the partial recursive function γ' which is defined by $\gamma'(e) = \gamma(e)$ iff $\gamma(e) = 1$ or 0, and $\gamma'(e) \uparrow$ iff $\gamma(e) = 2$. (That γ' becomes partial recursive is straightforward from the Graph Theorem.) However, Lemma 3 in Appendix C shows that $S = \{ e : e \text{ represents some } \mathbf{p} \in \mathcal{P}^{\mathbf{N}}_{\text{REC}} \text{ at } (x, y) \}$ is not r.e.

2. One of the most obvious conditions of computability that one might think of is the following. It requires existence of a deciding algorithm (for the restricted social preferences) that gives an output only for legitimate representations of a profile at a pair.

Computability B For each pair $(x, y) \in X^2$, there is a partial recursive function γ such that the condition (a) in Pairwise Computability is satisfied, and
(c) for each integer $e \in \mathbf{N}$, if e does not represent any $\mathbf{p} \in \mathcal{P}^{\mathbf{N}}_{\text{REC}}$ at (x, y), then $\gamma(e) \uparrow$.

The same argument (that S is not r.e.) shows that no REC-social welfare function that satisfies Independence can satisfy Computability B.

B Recursion Theory

This appendix reviews the definitions and results from recursion theory necessary for understanding technical sections of the present paper. I mostly follow the notations and terminologies in Soare [19]. Other references on recursion theory include Rogers [17] and Davis and Weyuker [6].

In this appendix, x, y and z denote nonnegative integers. For sets A and B, $\overline{A}$ denotes the complement of A; $A-B$ denotes the set theoretic difference $A\cap\overline{B}$.

B.1 Partial Functions

A *partial function* on $\mathbf{N}^n$ is a function 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.

B.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 computed by a Turing program. *Turing programs* can be defined precisely, but we do not do that here. For our purpose, it suffices to know that we can list all Turing programs in such a way that for any program we can algorithmically 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.

B.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.

The **Enumeration Theorem** states [19, 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 [19, 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 any z_1, ..., z_n.

The **Graph Theorem** states [19, p. 29] that a partial function is partial recursive iff its graph is r.e.

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 $\langle\langle x,y\rangle,z\rangle$.

C Proofs

C.1 Recursion Theory

Lemma 1 *There is a one-to-one recursive function r such that for all e and u,*

$$\varphi_{r(e)}(u) = \begin{cases} 1 & \text{if } \varphi_e(u) = 0, \\ 0 & \text{if } \varphi_e(u)\downarrow \text{ and } \varphi_e(u) \neq 0, \\ \uparrow & \text{if } \varphi_e(u)\uparrow. \end{cases} \tag{1}$$

In particular, if e is a characteristic index of a set A, then $r(e)$ is a characteristic index of its complement $\overline{A}$.

Proof. The right hand side is equal to $\psi(e,u) = 1 - \varphi_e(u)$, where $-$ is the limited subtraction. Since the limited subtraction is recursive, ψ is partial recursive by the Enumeration Theorem. Then by the Parameter Theorem, there is a one-to-one recursive function r such such that $\varphi_{r(e)}(u) = \psi(e,u)$.

Details. Since ψ is partial recursive, $\psi = \varphi_z^{(2)}$ 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 $r(e) = s(z,e)$. Then r is one-to-one and recursive. $\diamondsuit$ ∎

Lemma 2 *Let*

$$\mathrm{CRec} = \{\, e \in \mathbf{N} : e \text{ is a characteristic index of a recursive set} \,\}.$$

Then CRec *is not r.e.*

Proof. (This proof involves deeper recursion theory than that covered in Appendix B.) Fix a Σ_2 set A. Then, by [19, IV.3.2, p. 66], there is a recursive function f such that

$$e \in A \Longrightarrow \varphi_{f(e)}(u)\downarrow \text{ for only finitely many } u$$

and

$$e \notin A \Longrightarrow \varphi_{f(e)}(u) = 0 \text{ for all } u.$$

It follows that

$$e \in A \Longrightarrow f(e) \notin \text{CRec}$$

and

$$e \notin A \Longrightarrow f(e) \in \text{CRec}.$$

This shows that $\overline{A} \leq_m$ CRec, namely, $\overline{A}$ is many-one reducible to CRec.

Now, suppose that CRec is r.e., that is, CRec $\in \Sigma_1$. Then by [19, IV.1.3(v), p. 61], $\overline{A} \in \Sigma_1$, i.e., $A \in \Pi_1$. This means that any Σ_2 set A is Π_1, a contradiction. Hence, CRec is not r.e. ∎

C.2 Social Choice Theory

Proof of Proposition 1. Let $\succ$ satisfy Neutrality and PC. Fix (x', y'). There is a partial recursive γ that satisfies (a) in the PC condition. We show that γ satisfies (a) for all pairs (x, y).

Fix (x, y), $\mathbf{p}$, and e so that e represents $\mathbf{p}$ at (x, y). Then e represents $\mathbf{p}'$ at (x', y') for some $\mathbf{p}'$ such that $\{ i : x' \succ^{\mathbf{p}'} y' \} = \{ i : x \succ^{\mathbf{p}} y \}$ and $\{ i : y' \succ^{\mathbf{p}'} x' \} = \{ i : y \succ^{\mathbf{p}} x \}$. This implies that $(x \succ_i^{\mathbf{p}} y \leftrightarrow x' \succ_i^{\mathbf{p}'} y')$ and $(y \succ_i^{\mathbf{p}} x \leftrightarrow y' \succ_i^{\mathbf{p}'} x')$ for all $i \in I$. By Neutrality, it follows that $(x \succ^{\mathbf{p}} y \leftrightarrow x' \succ^{\mathbf{p}'} y')$. So, if $x \succ^{\mathbf{p}} y$, then $x' \succ^{\mathbf{p}'} y'$, which implies that $\gamma(e) = 1$ by (a) in PC. Similarly, if $\neg x \succ^{\mathbf{p}} y$, then $\gamma(e) = 0$. This shows that (a) in SPC is satisfied for (x, y), which was chosen arbitrarily. ∎

Proof of Proposition 2. ($\Longrightarrow$). By Proposition 1.

($\Longleftarrow$). Suppose that $\succ$ satisfies Independence and SPC but violates Neutrality. Then by the violation of Neutrality, there are x, x', y, y' and $\mathbf{p}$, $\mathbf{p}'$ such that

$$(x \succ_i^{\mathbf{p}} y \leftrightarrow x' \succ_i^{\mathbf{p}'} y') \ \& \ (y \succ_i^{\mathbf{p}} x \leftrightarrow y' \succ_i^{\mathbf{p}'} x') \tag{2}$$

for all $i \in I$, but $x \succ^{\mathbf{p}} y$ and $\neg x' \succ^{\mathbf{p}'} y'$.

Since SPC is satisfied, there is a partial recursive γ that satisfies (a) of SPC. Suppose e represents $\mathbf{p}$ at (x, y). Then, since $x \succ^{\mathbf{p}} y$, it follows by (a) of SPC that $\gamma(e) = 1$. On the other hand, since e represents $\mathbf{p}$ at (x, y), e represents $\mathbf{p}'$ at (x', y') by (2). By (a) of SPC, since $\neg x' \succ^{\mathbf{p}'} y'$, $\gamma(e) = 0$. This contradicts that $\gamma(e) = 1$. ∎

Lemma 3 *Let* $\succ : \mathcal{P}_{\text{REC}}^{\mathbf{N}} \to \mathcal{P}$ *be an* REC-*social welfare function satisfying Independence. Let* $(x, y) \in X^2$. *Then the set*

$$S = \{ e : e \text{ represents some } \mathbf{p} \in \mathcal{P}_{\text{REC}}^{\mathbf{N}} \text{ at } (x, y) \}$$

is not r.e.

Proof. Fix $\succ$ and (x, y). Suppose that S is r.e. Let e_3' be an arbitrary characteristic index for an empty set. Let r be a recursive function satisfying (1) in Lemma 1. Let CRec be the set of characteristic indices of some recursive set.

Claim $\langle e_1, r(e_1), e_3' \rangle \in S$ iff $e_1 \in$ CRec.

Details. ($\Longrightarrow$). Suppose that $\langle e_1, r(e_1), e_3' \rangle \in S$. Then e_1 is a characteristic index for $\{ i : x \succ_i^{\mathbf{P}} y \}$.

($\Longleftarrow$). Suppose that e_1 is characteristic index for an $A \subseteq \mathbf{N}$. Choose a $\mathbf{p} \in \mathcal{P}_{\text{REC}}^{\mathbf{N}}$ such that $A = \{ i : x \succ_i^{\mathbf{P}} y \}$ and $\overline{A} = \{ i : y \succ_i^{\mathbf{P}} x \}$. Then $r(e_1)$ is a characteristic index of $\overline{A}$ by Lemma 1. So, e_1, $r(e_1)$, and e_3' are characteristic indices of $\{ i : x \succ_i^{\mathbf{P}} y \}$, $\{ i : y \succ_i^{\mathbf{P}} x \}$, and $\{ i : x \sim_i^{\mathbf{P}} y \} = \emptyset$ respectively. $\diamondsuit$

Now since S is assumed to be r.e., Claim implies that CRec is r.e.

Details. The function f defined by $f(e_1) = \langle e_1, r(e_1), e_3' \rangle$ is recursive. Since S is r.e., it it the domain of the partial recursive function φ_z for some z. We have, by the above Claim, that $e_1 \in$ CRec iff $f(e_1) \in \text{dom}\varphi_z$. But the latter is equivalent with $e_1 \in \text{dom}(\varphi_z \circ f)$. This means that CRec is the domain of $\varphi_z \circ f$; hence, r.e. $\diamondsuit$

However, this contradicts Lemma 2 which states that CRec is not r.e. ∎

Before proving Theorem 1 and Proposition 3, we must introduce some preliminary results and notions.

Let $\mathcal{B}$ be a Boolean algebra. 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 $\mathcal{U}$ that satisfies: if $A \notin \mathcal{U}$, then $\overline{A} \in \mathcal{U}$. If $\mathcal{U}$ is an ultrafilter, then $A \cup B \in \mathcal{U}$ implies that $A \in \mathcal{U}$ or $B \in \mathcal{U}$. Suppose $\mathcal{B}$ contains 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.

Proposition 4 (Armstrong [1, Proposition 3.2]) *Let $\mathcal{B}$ be a Boolean algebra on I. Suppose a $\mathcal{B}$-social welfare function $\succ$ satisfies Unanimity and Independence. Then there is a unique ultrafilter $\mathcal{U}_\succ$ on $\mathcal{B}$ such that for all $\mathbf{p} = (\succ_i^{\mathbf{P}})_{i \in I} \in \mathcal{P}_{\mathcal{B}}^I$ and x, $y \in X$,*

$$\{ i \in I : x \succ_i^{\mathbf{P}} y \} \in \mathcal{U}_\succ \Longrightarrow x \succ^{\mathbf{P}} y. \tag{3}$$

Remark. The uniqueness follows from Proposition 3.1 of Armstrong [1]. $\diamondsuit$

Remark. Armstrong [2] corrects an error in Proposition 3.2 of his earlier work [1]. Proposition 4 is the corrected version. $\diamondsuit$

Proposition 5 (Armstrong [1, Proposition 3.1]) *Let $\mathcal{B}$ be a Boolean algebra on I. Suppose $\mathcal{U}$ is an ultrafilter on $\mathcal{B}$. Then the map $\succ$ on $\mathcal{P}_{\mathcal{B}}^I$ defined for $\mathbf{p} \in \mathcal{P}_{\mathcal{B}}^I$ and x, $y \in X$ by*

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

is a $\mathcal{B}$-social welfare function, satisfying Unanimity and Independence.

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

$$\beta_\succ(e_1) = \begin{cases} 1 & \text{if } e_1 \text{ is a characteristic index of a recursive set in } \mathcal{U}_\succ, \\ 0 & \text{if } e_1 \text{ is a characteristic index of a recursive set not in } \mathcal{U}_\succ, \\ \uparrow & \text{if } e_1 \text{ is not a characteristic index of a recursive set,} \end{cases} \tag{4}$$

where $\mathcal{U}_\succ$ denotes the ultrafilter in Proposition 4. Note that $\beta_\succ$ is well-defined since each $e_1 \in \mathbf{N}$ can be a characteristic index of at most one set.

A computability condition for an REC-social welfare function satisfying Unanimity and Independence is introduced using the partial function $\beta_\succ$ defined by (4):

Decidability of Decisive Coalitions (DDC) $\beta_\succ$ has an extension to a partial recursive function.

Lemma 4 *Let* $\succ : \mathcal{P}^{\mathbf{N}}_{\mathrm{REC}} \to \mathcal{P}$ *be an* REC-*social welfare function satisfying Unanimity and Independence. Then* $\succ$ *is dictatorial if and only if it satisfies DDC.*

Proof. ($\Longrightarrow$). Suppose $\succ$ is dictatorial. Then the ultrafilter $\mathcal{U}_\succ$ in Proposition 4 corresponding to $\succ$ is principal; namely, for some $i_0 \in \mathbf{N}$, $\mathcal{U}_\succ = \{ W \in \mathrm{REC} : i_0 \in W \}$.

Define β' by

$$\beta'(e) = \begin{cases} \varphi_e(i_0) & \text{if } \varphi_e(i_0) = 0 \text{ or } 1, \\ \uparrow & \text{otherwise.} \end{cases}$$

Then, clearly, β' is an extension of $\beta_\succ$. Also, $\varphi_e(i_0)$ is a partial recursive function of e by the Enumeration Theorem. Hence, β' is partial recursive.

($\Longleftarrow$). Let $\succ$ satisfy the hypothesis and suppose DDC is satisfied but $\succ$ is not dictatorial. In this case, the ultrafilter $\mathcal{U}_\succ$ corresponding to $\succ$ is free: it does not contain any finite sets. Let β' be an extension of $\beta_\succ$ which is partial recursive. Note that $\beta(e) = 1$ if e is a characteristic index of a cofinite set since $\mathcal{U}_\succ$ is a free ultrafilter.

Let $K = \{ e : e \in W_e \}$; K is a nonrecursive r.e. set. Since K is r.e., there is [19, II.1.2, p. 28] a recursive set $R \subseteq \mathbf{N} \times \mathbf{N}$ such that $e \in K \iff \exists z R(e,z)$. Using the Parameter Theorem, define a recursive function f by

$$\varphi_{f(e)}(u) = \begin{cases} 1 & \text{if } \exists z \le u \; R(e,z), \\ 0 & \text{otherwise.} \end{cases}$$

Details. The function h defined by

$$h(e,u) = \begin{cases} 1 & \text{if } \exists z \le u \; R(e,z), \\ 0 & \text{otherwise} \end{cases}$$

is recursive. Hence, for some z, $h = \varphi^{(2)}_z$. By the Parameter Theorem, there is a recursive function s such that

$$\varphi_{s(z,e)}(u) = \varphi^{(2)}_z(e,u) = h(e,u).$$

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

Now,

$$\begin{aligned} e \in K &\implies \varphi_{f(e)}(u) = 1 \text{ except for finitely many } u\text{'s} \\ &\implies f(e) \text{ is a characteristic index of a cofinite set} \\ &\implies \beta'(f(e)) = \beta(f(e)) = 1, \end{aligned}$$

but

$$\begin{aligned} e \notin K &\implies \varphi_{f(e)}(u) = 0 \text{ for all } u \\ &\implies f(e) \text{ is a characteristic index of } \emptyset \\ &\implies \beta'(f(e)) = \beta(f(e)) = 0. \end{aligned}$$

This implies that K is recursive, contradiction. ∎

Proof of Theorem 1. (Preliminaries for this proof begin after the proof of Lemma 3.) Let $\succ$ satisfy the hypotheses and PC. Fix (x, y). There is a partial recursive γ that satisfies the condition (a) in PC. We will show that DDC is satisfied.

Let e_3' be an arbitrary characteristic index for an empty set and let r be a recursive function satisfying (1) in Lemma 1. Then

$$\beta_{\succ}(e_1) = \gamma(\langle e_1, r(e_1), e_3' \rangle) \tag{5}$$

for all $e_1 \in$ CRec. ((5) means that when nobody is indifferent between x and y, then to determine the social preference on (x, y), all the planner need to know is the coalition $\{\, i \in \mathbf{N} : x \succ_i^{\mathbf{P}} y \,\}$ that prefers x to y.)

Details. Suppose $e_1 \in$ CRec, the domain of $\beta_{\succ}$. Then by the Claim in the proof of Lemma 3, $e = \langle e_1, r(e_1), e_3' \rangle$ represents some $\mathbf{p} \in \mathcal{P}_{\mathrm{REC}}^{\mathbf{N}}$ at (x, y). In particular, e_1 and $r(e_1)$ are characteristic indices for $A = \{\, i \in \mathbf{N} : x \succ_i^{\mathbf{P}} y \,\}$ and $\overline{A} = \{\, i : y \succ_i^{\mathbf{P}} x \,\}$ respectively.

(i) Suppose that $\beta_{\succ}(e_1) = 1$. Then by (4), $\{\, i \in \mathbf{N} : x \succ_i^{\mathbf{P}} y \,\} \in \mathcal{U}_{\succ}$. Then (3) in Proposition 4 implies that $x \succ^{\mathbf{P}} y$. So, (a) in PC implies that $\gamma(e) = 1$.

(ii) Suppose $\beta_{\succ}(e_1) = 0$. Then by (4), $A \notin \mathcal{U}_{\succ}$. Since $\mathcal{U}_{\succ}$ is an ultrafilter, it follows that $\overline{A} = \{\, i : y \succ_i^{\mathbf{P}} x \,\} \in \mathcal{U}_{\succ}$. By (3), $y \succ^{\mathbf{P}} x$. By asymmetry, we have $\neg x \succ^{\mathbf{P}} y$. Hence $\gamma(e) = 0$ by (a) in PC. $\diamondsuit$

Now, the partial function ψ defined by $\psi(e_1) = \gamma'(\langle e_1, r(e_1), e_3' \rangle)$ is clearly partial recursive. By (5), $\beta_{\succ}(e_1) = \psi(e_1)$ for all $e_1 \in$ CRec. Hence, the partial recursive function ψ is an extension of $\beta_{\succ}$. So, DDC is satisfied. By Lemma 4, it follows that $\succ$ is dictatorial. ∎

Proof of Proposition 3. (Preliminaries for this proof begin after the proof of Lemma 3.) (i) Let $\succ$ be the social welfare function in Example 1. We show $\succ$ satisfies SPC. Since $\succ$ is dictatorial, Lemma 4 implies that DDC is satisfied.

So, there is a partial recursive function β' that extends $\beta_{\succ}$. Define a partial 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 $\pi: e \mapsto e_1$. Since π and β' are partial recursive, γ is partial recursive.

We show that γ satisfies (a) in SPC for all (x, y). 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 $\{\, i \in \mathbf{N} : x \succ_i^{\mathbf{P}} y \,\}$ and so $\beta_{\succ}(e_1)\downarrow$.

- Suppose $x \succ^{\mathbf{P}} y$. Then by the definition of $\succ$, $\{\, i \in \mathbf{N} : x \succ_i^{\mathbf{P}} y \,\} \in \mathcal{U}_0$. This implies that $\gamma(e) = \beta'(e_1) = \beta_{\succ}(e_1) = 1$.
- Similarly, if $\neg x \succ^{\mathbf{P}} y$ then $\gamma(e) = 0$.

(ii) Let $\succ$ be the social welfare function in Example 2. To show $\succ$ satisfies PC, fix (x, y). Since $\succ$ is dictatorial, Lemma 4 implies that DDC is satisfied. So, there is a partial recursive function β' that extends $\beta_{\succ}$. Define a partial function γ by

$$\gamma(\langle e_1, e_2, e_3\rangle) = \begin{cases} 1 & \text{if } (\beta'(e_1), \beta'(e_2), \beta'(e_3)) = (1, 0, 0), \\ 0 & \text{if } (\beta'(e_1), \beta'(e_2), \beta'(e_3)) = (0, 1, 0), \\ c & \text{if } (\beta'(e_1), \beta'(e_2), \beta'(e_3)) = (0, 0, 1), \\ \uparrow & \text{otherwise,} \end{cases}$$

where $c = 1$ if xPy and $c = 0$ if $\neg xPy$. Then by Church's Thesis (or by the Graph Theorem) γ is partial recursive. To show that γ satisfies (a) in PC, suppose $e = \langle e_1, e_2, e_3\rangle$ represents a $\mathbf{p}$ at (x, y) and argue as before, noting $\beta'(e_1)\downarrow$, $\beta'(e_2)\downarrow$, $\beta'(e_3)\downarrow$, and that exactly one of these three takes the value 1, and the others 0.

(iii) Let $\succ$ be the social welfare function in Example 3. To show $\succ$ does not satisfy PC, suppose otherwise. Choose (x, y) arbitrarily. Then there is a partial recursive γ that satisfies (a) in PC. Let g be a recursive function such that if e is a characteristic index of a set A then $g(e)$ is a characteristic index of $A - \{0\}$. Such a g exists by [19, II.2.3, p. 33]. Let r be a recursive function satisfying (1) in Lemma 1 and let e_2' be an arbitrary characteristic index for an empty set. Define a partial recursive function β' by

$$\beta'(e_1) = \gamma(\langle g(e_1), e_2', r(g(e_1))\rangle).$$

We show that β' extends $\beta_{\hat{\succ}}$, where $\hat{\succ}: \mathcal{P}_{\mathrm{REC}}^{\mathbf{N}} \to \mathcal{P}$ is defined by

$$x \hat{\succ}^{\mathbf{P}} y \iff \{\, i \in \mathbf{N} : x \succ_i^{\mathbf{P}} y \,\} \in \hat{\mathcal{U}}.$$

Notice that $\mathcal{U}_{\hat{\succ}} = \hat{\mathcal{U}}$.

- Suppose e_1 is a characteristic index of a recursive set A in $\hat{\mathcal{U}}$. Then $e = \langle g(e_1), e_2', r(g(e_1))\rangle$ represents a $\mathbf{p}$ at (x, y); in particular, $g(e_1)$ being a characteristic index of $\{ i \in \mathbf{N} : x \succ_i^{\mathbf{P}} y \} = A - \{0\}$. Clearly, $\{ i \in \mathbf{N} : x \succ_i^{\mathbf{P}} y \}$ does not belong to $\mathcal{U}_0$ since it does not contain 0. Hence its complement $\{ i : x \sim_i^{\mathbf{P}} y \}$ belongs to $\mathcal{U}_0$. Now, $\{ i \in \mathbf{N} : x \succ_i^{\mathbf{P}} y \} \in \hat{\mathcal{U}}$ since $\{ i \in \mathbf{N} : x \succ_i^{\mathbf{P}} y \}$ and A are different at most by the finite set $\{0\}$ and A is in the free ultrafilter $\hat{\mathcal{U}}$. Hence, by the definition of $\succ$, it follows that $x \succ^{\mathbf{P}} y$. This implies, by (a) in PC, that $\gamma(e) = 1$. So, $\beta'(e_1) = 1$.
- Similarly, if e_1 is a characteristic index of a recursive set not in $\hat{\mathcal{U}}$, then $\beta'(e_1) = 0$.

We have shown that $\beta_{\hat{\succ}}$ had an extension β' that is partial recursive. This means that $\hat{\succ}$ satisfies DDC, contradicting Lemma 4 since $\hat{\succ}$ is not dictatorial. ∎

References

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

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

[3] K. J. Arrow. *Social Choice and Individual Values*. Yale University Press, New Haven and London, 2nd edition, 1963.

[4] K. J. Arrow. Rationality of self and others in an economic system. *Journal of Business*, 59:S385–S399, 1986.

[5] D. Canning. Rationality, computability, and Nash equilibrium. *Econometrica*, 60(4):877–888, July 1992.

[6] M. D. Davis and E. J. Weyuker. *Computability, Complexity, and Languages: Fundamentals of Theoretical Computer Science*. Computer Science and Applied Mathematics. Academic Press, San Diego, 1983.

[7] P. C. Fishburn. Arrow's Impossibility Theorem: Concise proof and infinite voters. *Journal of Economic Theory*, 2:103–6, 1970.

[8] D. M. Hausman and M. S. McPherson. Taking ethics seriously: Economics and contemporary moral philosophy. *Journal of Economic Literature*, 31:671–731, June 1993.

[9] F. A. Hayek. The use of knowledge in society. *American Economic Review*, 35:519–30, 1945.

[10] J. S. Kelly. Social choice and computational complexity. *Journal of Mathematical Economics*, 17:1–8, 1988.

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

[12] D. Lavoie. *Rivalry and Central Planning: The Socialist Calculation Debate Reconsidered.* Historical Perspectives on Modern Economics. Cambridge University Press, Cambridge, 1985.

[13] A. A. Lewis. On effectively computable realizations of choice functions. *Mathematical Social Sciences*, 10:43–80, 1985.

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

[15] H. R. Mihara. Anonymity and neutrality in Arrow's Theorem with restricted coalition algebras. Mimeo, August 1994.

[16] H. R. Mihara. *Arrow's Theorem, Turing computability, and oracles.* PhD thesis, University of Minnesota, Minneapolis, c. 1995.

[17] H. Rogers, Jr. *Theory of Recursive Functions and Effective Computability.* MIT Press, Cambridge, paperback edition, 1987.

[18] A. K. Sen. Social choice theory. In K. J. Arrow and M. D. Intriligator, editors, *Handbook of Mathematical Economics*, volume III, chapter 22, pages 1073–1181. Elsevier, Amsterdam, 1986.

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

[20] S. E. Spear. Learning rational expectations under computability constraints. *Econometrica*, 57(4):889–910, July 1989.

[21] K. Wong. *General equilibrium theory from a computable point of view.* PhD thesis, University of Minnesota, Minneapolis, c. 1994.