Arrow’s General Possibility Theorem

Peter Gammie
peteg42 at gmail.com
May 28, 2015

Contents

1 Overview 2

2 General Lemmas 2
  2.1 Extra Finite-Set Lemmas 2
  2.2 Extra bijection lemmas 2
  2.3 Collections of witnesses: hasw, has 3

3 Preliminaries 5
  3.1 Rational Preference Relations (RPRs) 5
  3.2 Profiles 7
  3.3 Choice Sets, Choice Functions 8
  3.4 Social Choice Functions (SCFs) 8
  3.5 Social Welfare Functions (SWFs) 9
  3.6 General Properties of an SCF 9
  3.7 Decisiveness and Semi-decisiveness 10

4 Arrow’s General Possibility Theorem 11
  4.1 Semi-decisiveness Implies Decisiveness 11
  4.2 The Existence of a Semi-decisive Individual 14
  4.3 Arrow’s General Possibility Theorem 15

5 Sen’s Liberal Paradox 15
  5.1 Social Decision Functions (SDFs) 15
  5.2 Sen’s Liberal Paradox 16

6 May’s Theorem 17
  6.1 May’s Conditions 18
  6.2 The Method of Majority Decision satisfies May’s conditions 19
  6.3 Everything satisfying May’s conditions is the Method of Majority Decision 20
  6.4 The Plurality Rule 22

7 Bibliography 23
1 Overview

This is a fairly literal encoding of some of Armatya Sen’s proofs [Sen70] in Isabelle/HOL. The author initially wrote it while learning to use the proof assistant, and some locutions remain naïve. This work is somewhat complementary to the mechanisation of more recent proofs of Arrow’s Theorem and the Gibbard-Satterthwaite Theorem by Tobias Nipkow [Nip08].

I strongly recommend Sen’s book to anyone interested in social choice theory; his proofs are quite lucid and accessible, and he situates the theory quite well within the broader economic tradition.

2 General Lemmas

2.1 Extra Finite-Set Lemmas

Small variant of Finite-Set finite-subset-induct: also assume \( F \subseteq A \) in the induction hypothesis.

**lemma** finite-subset-induct’ [consumes 2, case-names empty insert]:

**assumes** finite \( F \) and \( F \subseteq A \)

and empty: \( P \{\} \)

and insert: \( \forall a F. \{ \text{finite } F; a \in A; F \subseteq A; a \notin F; P F \} \implies P (\text{insert } a F) \)

**shows** \( P F \)

(proof)

A slight improvement on List.finite-list - add distinct.

**lemma** finite-list: finite \( A \) \( \implies \exists l. \text{set } l = A \land \text{distinct } l \)

(proof)

2.2 Extra bijection lemmas

**lemma** bij-betw-onto: bij-betw \( f \ A B \implies f \ A = B \) (proof)

**lemma** inj-on-UnI: \[ \text{inj-on } f A; \text{inj-on } f B; f \ (A - B) \cap f \ (B - A) = \{\} \] \implies inj-on \( f \ (A \cup B) \)

(proof)

**lemma** card-compose-bij:

**assumes** \( \text{bijf: bij-betw } f \ A \ A \)

**shows** \( \text{card } \{ a \in A. \ P (f a) \} = \text{card } \{ a \in A. \ P a \} \)

(proof)

**lemma** card-eq-bij:

**assumes** \( \text{cardAB: card } A = \text{card } B \)

and \( \text{finiteA: finite } A \) and \( \text{finiteB: finite } B \)

obtains \( f \) where \( \text{bij-betw } f \ A B \)

(proof)

**lemma** bij-combine:

**assumes** \( ABCD: A \subseteq B C \subseteq D \)

and \( \text{bijf: bij-betw } f \ A C \)

and \( \text{bijg: bij-betw } g \ (B - A) \ (D - C) \)
obtains \( h \) where bij-betw \( h \) \( B \) \( D \) and \( \forall x. x \in A \Rightarrow h x = f x \) and \( \forall x. x \in B \setminus A \Rightarrow h x = g x \)
(proof)

lemma bij-complete:
assumes finite\( C \): finite \( C \)
and \( ABC \): \( A \subseteq C \) \( B \subseteq C \)
and bij: bij-betw \( f \) \( A \) \( B \)
obtains \( f' \) where bij-betw \( f' \) \( C \) \( C \)
and \( \forall x. x \in A \Rightarrow f' x = f x \)
and \( \forall x. x \in C \setminus A \Rightarrow f' x \in C \setminus B \)
(proof)

lemma card-greater:
assumes finite\( A \): finite \( A \)
and \( c \) \( \colon \) card \{ \( x \in A. P x \) \} \( > \) card \{ \( x \in A. Q x \) \}
obtains \( C \)
where card \{ \( \{ x \in A. P x \} \setminus C \) \} \( = \) card \{ \( x \in A. Q x \) \}
and \( C \neq \{ \} \)
and \( C \subseteq \{ x \in A. P x \} \)
(proof)

2.3 Collections of witnesses: \( \text{hasw}, \text{has} \)

Given a set of cardinality at least \( n \), we can find up to \( n \) distinct witnesses. The built-in \( \text{card} \) function unfortunately satisfies:

\[
\text{Finite-Set.card-infinite: } \neg \text{finite } A \Rightarrow \text{card } A = 0
\]

These lemmas handle the infinite case uniformly.

Thanks to Gerwin Klein suggesting this approach.

definition \( \text{hasw} \) \( :: \) \( 'a \ \text{list} \Rightarrow 'a \ \text{set} \Rightarrow \text{bool} \) where
\( \text{hasw } xs \ S \equiv \text{set } xs \subseteq S \land \text{distinct } xs \)
definition \( \text{has} \) \( :: \) nat \( \Rightarrow 'a \ \text{set} \Rightarrow \text{bool} \) where
\( \text{has } n \ S \equiv \exists xs. \text{hasw } xs \ S \land \text{length } xs = n \)
declare \( \text{hasw-def\[simp\]} \)

lemma \( \text{hasI\[intro\]} \): \( \text{hasw } xs \ S \Rightarrow \text{has } (\text{length } xs) \ S \) (proof)

lemma \( \text{card-has} \):
assumes \( \text{cardS} \): \( \text{card } S = n \)
shows \( \text{has } n \ S \)
(proof)

lemma \( \text{card-has-rev} \):
assumes \( \text{finiteS} \): finite \( S \)
shows \( \text{has } n \ S \Rightarrow \text{card } S \geq n \) (is \( ?\text{lhs} \Rightarrow ?\text{rhs} \))
(proof)
lemma has-0: has 0 S ⟨proof⟩

lemma has-suc-notempty: has (Suc n) S ⇒ \{\} \neq S ⟨proof⟩

lemma has-suc-subset: has (Suc n) S ⇒ \{\} \subset S ⟨proof⟩

lemma has-notempty-1:
  assumes Sne: \{\} \neq S
  shows has 1 S ⟨proof⟩

lemma has-le-has:
  assumes h: has n S
  and nn’: n’ \leq n
  shows has n’ S ⟨proof⟩

lemma has-ge-has-not:
  assumes h: \neg has n S
  and nn’: n \leq n’
  shows \neg has n’ S ⟨proof⟩

lemma has-eq:
  assumes h: has n S
  and hn’: \neg has (Suc n) S
  shows \card S = n ⟨proof⟩

lemma has-extend-witness:
  assumes h: has n S
  shows [ set xs \subseteq S; length xs < n ] \implies set xs \subset S ⟨proof⟩

lemma has-extend-witness’:
  [ has n S; hasw xs S; length xs < n ] \implies \exists x. hasw (x \# xs) S ⟨proof⟩

lemma has-witness-two:
  assumes hasnS: has n S
  and nn’: 2 \leq n
  shows \exists x y. hasw [x,y] S ⟨proof⟩

lemma has-witness-three:
  assumes hasnS: has n S
  and nn’: 3 \leq n
  shows \exists x y z. hasw [x,y,z] S ⟨proof⟩
lemma finite-set-singleton-contra:
assumes finiteS: finite S
and Sne: S ≠ {} and cardS: card S > 1 ⇒ False
shows ∃ j. S = {j} (proof)

3 Preliminaries

The auxiliary concepts defined here are standard \[Rou79, Sen70, Tay05\]. Throughout we make use of a fixed set A of alternatives, drawn from some arbitrary type 'a of suitable size. Taylor [Tay05] terms this set an agenda. Similarly we have a type 'i of individuals and a population Is.

3.1 Rational Preference Relations (RPRs)

Definitions for rational preference relations (RPRs), which represent indifference or strict preference amongst some set of alternatives. These are also called weak orders or (ambiguously) ballots.

Unfortunately Isabelle's standard ordering operators and lemmas are typeclass-based, and as introducing new types is painful and we need several orders per type, we need to repeat some things.

type-synonym 'a RPR = ('a * 'a) set
abbreviation rpr-eq-syntax :: 'a ⇒ 'a RPR ⇒ 'a ⇒ bool (- - ≈ [50, 1000, 51] 50) where
  x r ≈ y ≡ (x, y) ∈ r
definition indifferent-pref :: 'a ⇒ 'a RPR ⇒ 'a ⇒ bool (- - ⪯ [50, 1000, 51] 50) where
  x r ⪯ y ≡ (x r ⪯ y ∧ y r ⪯ x)
lemma indifferent-prefI[intro]: \[ x r ⪯ y; y r ⪯ x \] ⇒ x r ≈ y (proof)
lemma indifferent-prefD[dest]: x r ≈ y ⇒ x r ⪯ y ∧ y r ⪯ x (proof)
definition strict-pref :: 'a ⇒ 'a RPR ⇒ 'a ⇒ bool (- - ≺ [50, 1000, 51] 50) where
  x r ≺ y ≡ (x r ⪯ y ∧ ¬(y r ⪯ x))
lemma strict-pref-def-irrefl[simp]: ¬ (x r ≺ x) (proof)
lemma strict-prefI[intro]: \[ x r ⪯ y; ¬(y r ⪯ x) \] ⇒ x r ≺ y (proof)

Traditionally, x r ⪯ y would be written x R y, x r ≈ y as x I y and x r ≺ y as x P y, where the relation r is implicit, and profiles are indexed by subscripting.

Complete means that every pair of distinct alternatives is ranked. The ”distinct” part is a matter of taste, as it makes sense to regard an alternative as as good as itself. Here I take
Rational preference relations, also known as weak orders and (I guess) complete pre-orders.

Use the standard reflexivity separately.

Rational preference relations, also known as weak orders and (I guess) complete pre-orders.
lemma rprD: \( rpr \ A \ r \implies \text{complete} \ A \ r \land \text{refl-on} \ A \ r \land \text{trans} \ r \)

lemma rpr-in-set[dest]: \[ rpr \ A \ r; \ x \ r\preceq \ y \] \implies \{x,y\} \subseteq A

lemma rpr-refl[dest]: \[ rpr \ A \ r; \ x \in \ A \] \implies x \ r\preceq x

lemma rpr-less-not: \[ rpr \ A \ r; \ \text{hasw} \ [x,y] \ A; \ \neg \ x \ r\prec \ y \] \implies y \ r\preceq x

lemma rpr-less-imp-le[simp]: \[ x \ r\prec \ y \] \implies x \ r\preceq y

lemma rpr-less-imp-neq[simp]: \[ x \ r\prec \ y \] \implies x \neq y

lemma rpr-less-trans: \[ x \ r\prec \ y; \ y \ r\prec \ z; \ rpr \ A \ r \] \implies x \ r\prec z

lemma rpr-le-trans: \[ x \ r\preceq \ y; \ y \ r\preceq \ z; \ rpr \ A \ r \] \implies x \ r\preceq z

lemma rpr-le-less-trans: \[ x \ r\prec \ y; \ y \ r\preceq \ z; \ rpr \ A \ r \] \implies x \ r\prec z

lemma rpr-less-le-trans: \[ x \ r\prec \ y; \ y \ r\preceq \ z; \ rpr \ A \ r \] \implies x \ r\prec z

lemma rpr-complete: \[ rpr \ A \ r; \ x \in \ A; \ y \in \ A \] \implies x \ r\preceq y \lor y \ r\preceq x

3.2 Profiles

A profile (also termed a collection of ballots) maps each individual to an RPR for that individual.

type-synonym ('a, 'i) Profile = 'i \Rightarrow 'a RPR

definition profile :: 'a set \Rightarrow 'i set \Rightarrow ('a, 'i) Profile \Rightarrow bool where
\( \text{profile} \ A \ \text{Is} \ P \equiv \text{Is} \neq \{\} \land (\forall i \in \text{Is}. \ rpr \ A \ (P \ i)) \)

lemma profileI[intro]: \[ \land i. \ i \in \text{Is} \implies rpr \ A \ (P \ i); \ \text{Is} \neq \{\} \] \implies profile \ A \ Is \ P

lemma profile-rprD[dest]: \[ \text{profile} \ A \ Is \ P; \ i \in \text{Is} \] \implies rpr \ A \ (P \ i)

lemma profile-non-empty: profile \ A \ Is \ P \implies \text{Is} \neq \{\}
3.3 Choice Sets, Choice Functions

A choice set is the subset of $A$ where every element of that subset is (weakly) preferred to every other element of $A$ with respect to a given RPR. A choice function yields a non-empty choice set whenever $A$ is non-empty.

**definition** choiceSet :: 'a set ⇒ 'a RPR ⇒ 'a set

choiceSet $A$ $r$ ≡ \{ $x \in A . \forall y \in A. x \preceq y$ \}

**definition** choiceFn :: 'a set ⇒ 'a RPR ⇒ bool

choiceFn $A$ $r$ ≡ $\forall A' \subseteq A. A' \neq \{\} \rightarrow choiceSet A' r \neq \{\}

**lemma** choiceSetI[intro]:

$[ \forall x \in A; \forall y \in A. x \preceq y ] \rightarrow x \in choiceSet A r$

**lemma** choiceFnI[intro]:

$(\forall A' . [ A' \subseteq A; A' \neq \{\} ] \rightarrow choiceSet A' r \neq \{\}) \rightarrow choiceFn A r$

If a complete and reflexive relation is also quasi-transitive it will yield a choice function.

**definition** quasi-trans :: 'a RPR ⇒ bool

quasi-trans $r$ ≡ $\forall x y z . x \prec y \land y \prec z \rightarrow x \prec z$

**lemma** quasi-transI[intro]:

$(\forall x y z . x \prec y \land y \prec z \rightarrow x \prec z) \rightarrow quasi-trans r$

**lemma** quasi-transD: $[ x \prec y; y \prec z; quasi-trans r ] \rightarrow x \prec z$

**lemma** trans-imp-quasi-trans: $trans r \Rightarrow quasi-trans r$

**lemma** r-c-qt-imp-cf:

assumes finiteA: finite $A$
and c: complete $A$ $r$
and qt: quasi-trans $r$
and r: refl-on $A$ $r$

shows choiceFn $A$ $r$

**lemma** rpr-choiceFn: $[ finite A; rpr A r ] \Rightarrow choiceFn A r$

3.4 Social Choice Functions (SCFs)

A social choice function (SCF), also called a collective choice rule by Sen [Sen70, p28], is a function that somehow aggregates society’s opinions, expressed as a profile, into a preference relation.
The least we require of an SCF is that it be complete and some function of the profile. The latter condition is usually implied by other conditions, such as iia.

definition
SCF :: ('a, 'i) SCF ⇒ 'a set ⇒ 'i set ⇒ ('a set ⇒ 'i set ⇒ ('a, 'i) Profile ⇒ bool) ⇒ bool
where
SCF scf A Is Pcond ≡ (∀ P. Pcond A Is P ➝ (complete A (scf P)))

lemma SCFI[intro]:
  assumes c: ∃ P. Pcond A Is P =⇒ complete A (scf P)
  shows SCF scf A Is Pcond
⟨proof⟩

lemma SCF-completeD[dest]: [ SCF scf A Is Pcond; Pcond A Is P ] =⇒ complete A (scf P)
⟨proof⟩

3.5 Social Welfare Functions (SWFs)

A Social Welfare Function (SWF) is an SCF that expresses the society’s opinion as a single RPR.

In some situations it might make sense to restrict the allowable profiles.

definition
SWF :: ('a, 'i) SCF ⇒ 'a set ⇒ 'i set ⇒ ('a set ⇒ 'i set ⇒ ('a, 'i) Profile ⇒ bool) ⇒ bool
where
SWF swf A Is Pcond ≡ (∀ P x y. Pcond A Is P ∧ x ∈ A ∧ y ∈ A ∧ (∀ i ∈ Is. x (P i) ≺ y) ➝ x (scf P) ≺ y)

lemma SWF-rpr[dest]: [ SWF swf A Is Pcond; Pcond A Is P ] =⇒ rpr A (swf P)
⟨proof⟩

3.6 General Properties of an SCF

An SCF has a universal domain if it works for all profiles.

definition universal-domain :: 'a set ⇒ 'i set ⇒ ('a, 'i) Profile ⇒ bool
where
universal-domain A Is P ≡ profile A Is P

declare universal-domain-def[simp]

An SCF is weakly Pareto-optimal if, whenever everyone strictly prefers x to y, the SCF does too.

definition weak-pareto :: ('a, 'i) SCF ⇒ 'a set ⇒ 'i set ⇒ ('a set ⇒ 'i set ⇒ ('a, 'i) Profile ⇒ bool) ⇒ bool
where
weak-pareto scf A Is Pcond ≡ (∀ P x y. Pcond A Is P ∧ x ∈ A ∧ y ∈ A ∧ (∀ i ∈ Is. x (P i) ≺ y) ➝ x (scf P) ≺ y)

lemma weak-paretoI[intro]:
  (∀ x y. Is A; x ∈ A; y ∈ A; ∃ i ∈ Is. x (P i) ≺ y) =⇒ x (scf P) ≺ y
  =⇒ weak-pareto scf A Is Pcond
⟨proof⟩
lemma weak-pareto\text{D}:
\[
\text{weak-pareto scf A Is Pcond; Pcond A Is P}; \ x \in A; \ y \in A;
\quad (\forall i \in Is \implies x \ (P_i) \prec y) \implies x \ (\text{scf P}) \prec y
\langle \text{proof} \rangle
\]

An SCF satisfies independence of irrelevant alternatives if, for two preference profiles \( P \) and \( P' \) where for all individuals \( i \), alternatives \( x \) and \( y \) drawn from set \( S \) have the same order in \( P \) and \( P' \), then alternatives \( x \) and \( y \) have the same order in \( \text{scf P} \) and \( \text{scf P'} \).

\text{definition} iia :: ('a, 'i) SCF \Rightarrow 'a set \Rightarrow 'i set \Rightarrow \text{bool where}
\quad iia scf S Is P ≡
\quad (\forall P P' x y, \text{profile S Is P} \land \text{profile S Is P'}
\quad \land (\forall i \in \text{Is}, ((x (P_i) \leq y) \leftrightarrow (x (P'_i) \leq y)) \land ((y (P_i) \leq x) \leftrightarrow (y (P'_i) \leq x)))
\quad \implies ((x (\text{scf P}) \leq y) \leftrightarrow (x (\text{scf P'}) \leq y))
\langle \text{proof} \rangle
\]

\text{lemma} iia\text{I}\text{[intro]}:
\[
\forall P P' x y.
\quad \text{profile S Is P; profile S Is P'}
\quad \text{x \in S}; \ y \in S;
\quad (\forall i \in \text{Is}, ((x (P_i) \leq y) \leftrightarrow (x (P'_i) \leq y)) \land ((y (P_i) \leq x) \leftrightarrow (y (P'_i) \leq x)))
\quad \implies ((x (\text{scf P}) \leq y) \leftrightarrow (x (\text{scf P'}) \leq y))
\langle \text{proof} \rangle
\]

\text{lemma} iia\text{E}:
\[
\forall \text{scf S Is P}; \ \{x, y\} \subseteq S;
\quad a \in \{x, y\}; \ b \in \{x, y\}; \ i \in \text{Is}
\quad \implies (a (P'_i) \leq b) \leftrightarrow (a (P_i) \leq b);
\langle \text{proof} \rangle
\]

3.7 Decisiveness and Semi-decisiveness

This notion is the key to Arrow’s Theorem, and hinges on the use of strict preference [Sen70, p42].

A coalition \( C \) of agents is semi-decisive for \( x \) over \( y \) if, whenever the coalition prefers \( x \) to \( y \) and all other agents prefer the converse, the coalition prevails.

\text{definition} semi-decisive :: ('a, 'i) SCF \Rightarrow 'a set \Rightarrow 'i set \Rightarrow 'a \Rightarrow 'a \Rightarrow \text{bool where}
\quad semi-decisive scf A Is C x y ≡
\quad C \subseteq Is \land (\forall P, \text{profile A Is P} \land (\forall i \in C, x (P_i) \prec y) \land (\forall i \in Is - C, y (P_i) \prec x)
\quad \implies x (\text{scf P}) \prec y
\langle \text{proof} \rangle
\]

\text{lemma} semi-decisive\text{I}\text{[intro]}:
\[
\forall \text{profile A Is P}; \quad C \subseteq Is;
\quad (\forall i \in \text{Is \land C} \implies x (P_i) \prec y; \ \text{\land i, i \in Is - C} \implies y (P_i) \prec x)
\quad \implies \text{semi-decisive scf A Is C x y}
\langle \text{proof} \rangle
\]

10
lemma semidecisive-coalitionD[dest]: semidecisive scf A Is C x y \implies C \subseteq Is
\langle proof \rangle

lemma sd-refl: [ C \subseteq Is; C \neq {} ] \implies semidecisive scf A Is C x x
\langle proof \rangle

A coalition C is decisive for x over y if, whenever the coalition prefers x to y, the coalition prevails.

definition decisive :: ('a, 'i) SCF \Rightarrow 'a set \Rightarrow 'i set \Rightarrow 'a \Rightarrow 'a \Rightarrow bool where
decisive scf A Is C x y \equiv
C \subseteq Is \land (\forall P. \text{ profile } A Is P \land (\forall i \in C. x (P i) < y) \implies x (\text{ scf } P) < y)

lemma decisiveI[intro]:
[ C \subseteq Is; \land P. \land \text{ profile } A Is P; \land i. i \in C \implies x (P i) < y ] \implies x (\text{ scf } P) < y
\langle proof \rangle

lemma d-imp-sd: decisive scf A Is C x y \implies semidecisive scf A Is C x y
\langle proof \rangle

lemma decisive-coalitionD[dest]: decisive scf A Is C x y \implies C \subseteq Is
\langle proof \rangle

Anyone is trivially decisive for x against x.

lemma d-refl: [ C \subseteq Is; C \neq {} ] \implies decisive scf A Is C x x
\langle proof \rangle

Agent j is a dictator if her preferences always prevail. This is the same as saying that she is decisive for all x and y.

definition dictator :: ('a, 'i) SCF \Rightarrow 'a set \Rightarrow 'i set \Rightarrow 'i \Rightarrow bool where
dictator scf A Is j \equiv j \in Is \land (\forall x \in A. \forall y \in A. \text{ decisive scf } A Is \{ j \} x y)

lemma dictatorI[intro]:
[ j \in Is; \land x y. \land x \in A; y \in A ] \implies decisive scf A Is \{ j \} x y \implies dictator scf A Is j
\langle proof \rangle

lemma dictator-individual[dest]: dictator scf A Is j \implies j \in Is
\langle proof \rangle

4 Arrow’s General Possibility Theorem

The proof falls into two parts: showing that a semi-decisive individual is in fact a dictator, and that a semi-decisive individual exists. I take them in that order.

It might be good to do some of this in a locale. The complication is untangling where various witnesses need to be quantified over.
4.1 Semi-decisiveness Implies Decisiveness

I follow [Sen70, Chapter 3*] quite closely here. Formalising his appeal to the iia assumption is the main complication here.

The witness for the first lemma: in the profile $P'$, special agent $j$ strictly prefers $x$ to $y$ to $z$, and doesn’t care about the other alternatives. Everyone else strictly prefers $y$ to each of $x$ to $z$, and inherits the relative preferences between $x$ and $z$ from profile $P$.

The model has to be specific about ordering all the other alternatives, but these are immaterial in the proof that uses this witness. Note also that the following lemma is used with different instantiations of $x$, $y$ and $z$, so we need to quantify over them here. This happens implicitly, but in a locale we would have to be more explicit.

This is just tedious.

**Lemma decisive1-witness:**

assumes has3A: hasw $[x,y,z]$ $A$

and profileP: profile $A$ Is $P$

and jIs: $j \in Is$

obtains $P'$$

where profile $A$ Is $P'$

and $x (P' j) \prec y \land y (P' j) \prec z$

and $\bigwedge i. i \neq j \implies y (P' i) \prec x \land y (P' i) \prec z \land ((x (P' i) \preceq z) = (x (P i) \preceq z)) \land ((z (P' i) \preceq x) = (z (P i) \preceq x))$

(proof)

The key lemma: in the presence of Arrow’s assumptions, an individual who is semi-decisive for $x$ and $y$ is actually decisive for $x$ over any other alternative $z$. (This is where the quantification becomes important.)

**Lemma decisive1:**

assumes has3A: hasw $[x,y,z]$ $A$

and iia: iia swf $A$ Is

and swf: SWF swf $A$ Is universal-domain

and wp: weak-pareto swf $A$ Is universal-domain

and sd: semidecisive swf $A$ Is $\{j\}$ $x$ $y$

shows decisive swf $A$ Is $\{j\}$ $x$ $z$

(proof)

The witness for the second lemma: special agent $j$ strictly prefers $z$ to $x$ to $y$, and everyone else strictly prefers $z$ to $x$ and $y$ to $x$. (In some sense the last part is upside-down with respect to the first witness.)

**Lemma decisive2-witness:**

assumes has3A: hasw $[x,y,z]$ $A$

and profileP: profile $A$ Is $P$

and jIs: $j \in Is$

obtains $P'$$

where profile $A$ Is $P'$

and $z (P' j) \prec x \land x (P' j) \prec y$

and $\bigwedge i. i \neq j \implies z (P' i) \prec x \land y (P' i) \prec x \land ((y (P' i) \preceq z) = (y (P i) \preceq z)) \land ((z (P' i) \preceq y) = (z (P i) \preceq y))$

(proof)
lemma decisive2:
assumes has3A: hasw \[x, y, z\] A
    and iia: iia swf A Is
    and swf: SWF swf A Is universal-domain
    and wp: weak-pareto swf A Is universal-domain
    and sd: semidecisive swf A Is \{j\} x y
shows decisive swf A Is \{j\} z y
(proof)

The following results permute \(x\), \(y\) and \(z\) to show how decisiveness can be obtained from semi-decisiveness in all cases. Again, quite tedious.

lemma decisive3:
assumes has3A: hasw \[x, y, z\] A
    and iia: iia swf A Is
    and swf: SWF swf A Is universal-domain
    and wp: weak-pareto swf A Is universal-domain
    and sd: semidecisive swf A Is \{j\} x z
shows decisive swf A Is \{j\} y z
(proof)

lemma decisive4:
assumes has3A: hasw \[x, y, z\] A
    and iia: iia swf A Is
    and swf: SWF swf A Is universal-domain
    and wp: weak-pareto swf A Is universal-domain
    and sd: semidecisive swf A Is \{j\} y z
shows decisive swf A Is \{j\} y x
(proof)

lemma decisive5:
assumes has3A: hasw \[x, y, z\] A
    and iia: iia swf A Is
    and swf: SWF swf A Is universal-domain
    and wp: weak-pareto swf A Is universal-domain
    and sd: semidecisive swf A Is \{j\} x y
shows decisive swf A Is \{j\} y x
(proof)

lemma decisive6:
assumes has3A: hasw \[x, y, z\] A
    and iia: iia swf A Is
    and swf: SWF swf A Is universal-domain
    and wp: weak-pareto swf A Is universal-domain
    and sd: semidecisive swf A Is \{j\} y x
shows decisive swf A Is \{j\} y z decisive swf A Is \{j\} z x decisive swf A Is \{j\} x y
(proof)

lemma decisive7:
assumes has3A: hasw \[x, y, z\] A
    and iia: iia swf A Is
    and swf: SWF swf A Is universal-domain
    and wp: weak-pareto swf A Is universal-domain
    and sd: semidecisive swf A Is \{j\} x y
shows decisive swf A Is \{j\} y z decisive swf A Is \{j\} z x decisive swf A Is \{j\} x y
(proof)

lemma j-decisive-xy:
assumes has3A: hasw [x,y,z] A
and iia: iia swf A Is
and suf: SWF swf A Is universal-domain
and wp: weak-pareto swf A Is universal-domain
and sd: semidecisive swf A Is \{j\} x y
and uv: hasw [u,v] \{x,y,z\}
shows decisive swf A Is \{j\} u v
(proof)

lemma j-decisive:
assumes has3A: has 3 A
and iia: iia swf A Is
and suf: SWF swf A Is universal-domain
and wp: weak-pareto swf A Is universal-domain
and xyA: hasw [x,y] A
and sd: semidecisive swf A Is \{j\} x y
and uv: hasw [u,v] A
shows decisive swf A Is \{j\} u v
(proof)

The first result: if \(j\) is semidecisive for some alternatives \(u\) and \(v\), then they are actually a dictator.

lemma sd-imp-dictator:
assumes has3A: hasw [x,y,z] A
and iia: iia swf A Is
and suf: SWF swf A Is universal-domain
and wp: weak-pareto swf A Is universal-domain
and uv: hasw [u,v] A
and sd: semidecisive swf A Is \{j\} x y
shows dictator swf A Is \(j\)
(proof)

4.2 The Existence of a Semi-decisive Individual

The second half of the proof establishes the existence of a semi-decisive individual. The required witness is essentially an encoding of the Condorcet paradox (aka "the paradox of voting") that shows we get tied up in knots if a certain agent didn’t have dictatorial powers.

lemma sd-exists-witness:
assumes has3A: hasw [x,y,z] A
and V1 = V1 \cup V2 \cup V3
\land V1 \cap V2 = \{\} \land V1 \cap V3 = \{\} \land V2 \cap V3 = \{\}
and Is: Is \neq \{\}
obtains P
where profile A Is P
and \forall i \in V1. x (P i) < y \land y (P i) < z
and \forall i \in V2. z (P i) < x \land x (P i) < y
and \forall i \in V3. y (P i) < z \land z (P i) < x
(proof)
This proof is unfortunately long. Many of the statements rely on a lot of context, making it difficult to split it up.

**Lemma**: sd-exists:
- **Assumes**: has3A: has 3 A
- and finiteIs: finite Is
- and twoIs: has 2 Is
- and iia: iia suf A Is
- and suf: SWF suf A Is universal-domain
- and wp: weak-pareto suf A Is universal-domain
- **Shows**: \( \exists j u v. \text{hasw}[u,v] A \land \text{semidecisive suf A Is} \{j\} u v \)

### 4.3 Arrow’s General Possibility Theorem

Finally we conclude with the celebrated “possibility” result. Note that we assume the set of individuals is finite; [Rou79] relaxes this with some fancier set theory. Having an infinite set of alternatives doesn’t matter, though the result is a bit more plausible if we assume finiteness [Sen70, p54].

**Theorem**: ArrowGeneralPossibility:
- **Assumes**: has3A: has 3 A
- and finiteIs: finite Is
- and has2Is: has 2 Is
- and iia: iia suf A Is
- and suf: SWF suf A Is universal-domain
- and wp: weak-pareto suf A Is universal-domain
- **Obtains**: \( j \text{ where dictator suf A Is} j \)

### 5 Sen’s Liberal Paradox

#### 5.1 Social Decision Functions (SDFs)

To make progress in the face of Arrow’s Theorem, the demands placed on the social choice function need to be weakened. One approach is to only require that the set of alternatives that society ranks highest (and is otherwise indifferent about) be non-empty.

Following [Sen70, Chapter 4*], a **Social Decision Function** (SDF) yields a choice function for every profile.

**Definition**: SDF :: ('a, 'i) SCF \( \Rightarrow \) 'a set \( \Rightarrow \) 'i set \( \Rightarrow \) ('a set \( \Rightarrow \) 'i set \( \Rightarrow \) ('a, 'i) Profile \( \Rightarrow \) bool) \( \Rightarrow \) bool

**Where**
- \( SDF \text{ sdf A Is Pcond} \equiv (\forall P. \text{Pcond A Is P} \rightarrow \text{choiceFn A (sdf P)}) \)

**Lemma**: SDFI[intro]:
- \( (\forall P. \text{Pcond A Is P} \rightarrow \text{choiceFn A (sdf P)}) \rightarrow SDF \text{ sdf A Is Pcond} \)

**Lemma**: SWF-SDF:
- **Assumes**: finiteA: finite A
shows \( \text{SWF scf } A \text{ Is universal-domain } \implies \text{SDF scf } A \text{ Is universal-domain} \)

\[ \langle \text{proof} \rangle \]

In contrast to SWFs, there are SDFs satisfying Arrow’s (relevant) requirements. The lemma uses a witness to show the absence of a dictatorship.

**lemma SDF-nodictator-witness:**

**assumes** has2A: hasw \([x, y]\) \(A\)

and has2Is: hasw \([i, j]\) \(Is\)

**obtains** \(P\)

where profile \(A \text{ Is } P\)

and \(x (P \ i) \prec y\)

and \(y (P \ j) \prec x\)

\(\langle \text{proof} \rangle\)

**lemma SDF-possibility:**

**assumes** finiteA: finite \(A\)

and has2A: has 2 \(A\)

and has2Is: has 2 \(Is\)

**obtains** sdf

where weak-pareto sdf \(A \text{ Is universal-domain}\)

and iia sdf \(A \text{ Is}\)

and \(\neg(\exists j. \text{ dictator sdf } A \text{ Is } j)\)

and SDF sdf \(A \text{ Is universal-domain}\)

\(\langle \text{proof} \rangle\)

Sen makes several other stronger statements about SDFs later in the chapter. I leave these for future work.

### 5.2 Sen’s Liberal Paradox

Having side-stepped Arrow’s Theorem, Sen proceeds to other conditions one may ask of an SCF. His analysis of liberalism, mechanised in this section, has attracted much criticism over the years [AK96].

Following [Sen70, Chapter 6*], a liberal social choice rule is one that, for each individual, there is a pair of alternatives that she is decisive over.

**definition liberal :: \((’a, ’i) \text{ SCF } \Rightarrow ’a \text{ set } \Rightarrow ’i \text{ set } \Rightarrow \text{ bool}\) where**

liberal scf A Is \(\equiv\)

\((\forall i \in Is. \exists x \in A. \exists y \in A. x \neq y\) \wedge \text{decisive scf } A \text{ Is } \{i\} x y \wedge \text{decisive scf } A \text{ Is } \{i\} y x\)

**lemma liberalE:**

\[ [\text{liberal scf } A \text{ Is}; \ i \in Is ] \]

\[ \Rightarrow \exists x \in A. \exists y \in A. x \neq y \]

\[ \wedge \text{decisive scf } A \text{ Is } \{i\} x y \wedge \text{decisive scf } A \text{ Is } \{i\} y x\]

\(\langle \text{proof} \rangle\)

This condition can be weakened to require just two such decisive individuals; if we required just one, we would allow dictatorships, which are clearly not liberal.

**definition minimally-liberal :: \((’a, ’i) \text{ SCF } \Rightarrow ’a \text{ set } \Rightarrow ’i \text{ set } \Rightarrow \text{ bool}\) where**

minimally-liberal scf A Is \(\equiv\)

\((\exists i \in Is. \exists j \in Is. i \neq j)\)
\[ (\exists x \in A. \exists y \in A. x \neq y) \wedge \text{decisive scf} A \text{ Is } \{i\} x y \wedge \text{decisive scf} A \text{ Is } \{i\} y x \]
\[ (\exists x \in A. \exists y \in A. x \neq y) \wedge \text{decisive scf} A \text{ Is } \{j\} x y \wedge \text{decisive scf} A \text{ Is } \{j\} y x \]

**Lemma** liberal-imp-minimally-liberal:

**Assumes** has2Is: has 2 Is
**And** L: liberal scf A Is

**Shows** minimally-liberal scf A Is

(proof)

The key observation is that once we have at least two decisive individuals we can complete the Condorcet (paradox of voting) cycle using the weak Pareto assumption. The details of the proof don’t give more insight.

Firstly we need three types of profile witnesses (one of which we saw previously). The main proof proceeds by case distinctions on which alternatives the two liberal agents are decisive for.

**Lemmas** liberal-witness-two = SDF-nodictator-witness

**Lemma** liberal-witness-three:

**Assumes** threeA: hasw [x,y,v] A
**And** twoIs: hasw [i,j] Is

**Obtains** P
**Where** profile A Is P
**And** x (P i) \prec y
**And** v (P j) \prec x
**And** \( \forall i \in I s. \ y (P i) \prec v \)

(proof)

**Lemma** liberal-witness-four:

**Assumes** fourA: hasw [x,y,u,v] A
**And** twoIs: hasw [i,j] Is

**Obtains** P
**Where** profile A Is P
**And** x (P i) \prec y
**And** u (P j) \prec v
**And** \( \forall i \in I s. \ v (P i) \prec x \wedge y (P i) \prec u \)

(proof)

The Liberal Paradox: having two decisive individuals, an SDF and the weak pareto assumption is inconsistent.

**Theorem** LiberalParadox:

**Assumes** SDF: SDF sdf A Is universal-domain
**And** md: minimally-liberal sdf A Is
**And** wp: weak-pareto sdf A Is universal-domain

**Shows** False

(proof)
6 May’s Theorem

May’s Theorem [May52] provides a characterisation of majority voting in terms of four conditions that appear quite natural for \textit{a priori} unbiased social choice scenarios. It can be seen as a refinement of some earlier work by Arrow [Arr63, Chapter V.1].

The following is a mechanisation of Sen’s generalisation [Sen70, Chapter 5*]; originally Arrow and May consider only two alternatives, whereas Sen’s model maps profiles of full RPRs to a possibly intransitive relation that does at least generate a choice set that satisfies May’s conditions.

6.1 May’s Conditions

The condition of \textit{anonymity} asserts that the individuals’ identities are not considered by the choice rule. Rather than talk about permutations we just assert the result of the SCF is the same when the profile is composed with an arbitrary bijection on the set of individuals.

\textbf{definition anonymous} \(: \langle \langle a, i \rangle \rangle \ SCF \Rightarrow 'a set \Rightarrow 'i set \Rightarrow \text{bool} \ where \\
\text{anonymous scf} \ A \ Is \equiv \\
(\forall P \ f \ x \ y. \ \text{profile} \ A \ Is \ P \land \ \text{bij-betw} \ f \ Is \ Is \land x \in A \land y \in A \\
\rightarrow (x \ (\text{scf} \ P) \leq y) = (x \ (\text{scf} \ (P \circ f)) \leq y))\)

\textbf{lemma anonymousI[intro]}:
\[
(\exists P f x y. \ \text{profile} \ A \ Is \ P; \ \text{bij-betw} \ f \ Is \ Is; \\
x \in A; y \in A) \implies (x \ (\text{scf} \ P) \leq y) = (x \ (\text{scf} \ (P \circ f)) \leq y)
\]
\[\langle\text{proof}\rangle\]

\textbf{lemma anonymousD}:
\[
(\forall \text{scf} A \ Is; \ \text{profile} A \ Is P; \ \text{bij-betw} f \ Is Is; \ x \in A; y \in A) \\
\implies (x \ (\text{scf} \ P) \leq y) = (x \ (\text{scf} \ (P \circ f)) \leq y)
\]
\[\langle\text{proof}\rangle\]

Similarly, an SCF is \textit{neutral} if it is insensitive to the identity of the alternatives. This is Sen’s characterisation [Sen70, p.72].

\textbf{definition neutral} \(: \langle \langle a, i \rangle \rangle \ SCF \Rightarrow 'a set \Rightarrow 'i set \Rightarrow \text{bool} \ where \\
\text{neutral scf} \ A \ Is \equiv \\
(\forall P P' x y z w. \ \text{profile} A \ Is P \land \ \text{profile} A \ Is P' \land x \in A \land y \in A \land z \in A \land w \in A \\
\land (\forall i \in Is. \ x \ (P_i) \leq y \iff z \ (P'_i) \leq w) \land (\forall i \in Is. \ y \ (P_i) \leq x \iff w \ (P'_i) \leq z) \\
\rightarrow ((x \ (\text{scf} P) \leq y \iff z \ (\text{scf} P') \leq w) \land (y \ (\text{scf} P) \leq x \iff w \ (\text{scf} P') \leq z)))\)

\textbf{lemma neutralI[intro]}:
\[
(\forall P P' x y z w. \\
\text{profile} A \ Is P; \ \text{profile} A \ Is P'; \ \{x,y,z,w\} \subseteq A; \\
\forall i. i \in Is \implies x \ (P_i) \leq y \iff z \ (P'_i) \leq w; \\
\forall i. i \in Is \implies y \ (P_i) \leq x \iff w \ (P'_i) \leq z) \\
\implies ((x \ (\text{scf} P) \leq y \iff z \ (\text{scf} P') \leq w) \land (y \ (\text{scf} P) \leq x \iff w \ (\text{scf} P') \leq z))\\n\]
\[\langle\text{proof}\rangle\]

\textbf{lemma neutralD}:
\[
(\exists \text{neutral scf} A \ Is; \)
\[\langle\text{proof}\rangle\]
The method of majority decision (MMD) satisfies May’s conditions.

**Theorem:** The method of majority decision satisfies May’s conditions.

**Proof:**

Neutralarity implies independence of irrelevant alternatives.

**Lemma:** neutral-iia: neutral \( A \) is iia \( A \)

**Proof:**

Positive responsiveness is a bit like non-manipulability: if one individual improves their opinion of \( x \), then the result should shift in favour of \( x \).

**Definition:** positively-responsive :: \( \langle a, i \rangle \) SCF \( \Rightarrow \langle a \rangle \) set \( \Rightarrow \langle i \rangle \) set \( \Rightarrow \) bool where

\[
\left( \forall P P' x y, \text{profile } A \text{ is } P \land \text{profile } A \text{ is } P' \land x \in A \land y \in A \right) \\
\land \left( \forall i \in A, (x (P_i) \prec y) \implies (x (P'_i) \prec y) \right) \\
\land \left( \exists k \in A, (x (P_k) \prec x) \lor (y (P_k) \prec y) \right)
\]

\[
\implies (x (scf P) \preceq y) \implies (x (scf P') \prec y)
\]

**Lemma:** positively-responsiveI[\( \text{intro} \)]

**Assumes:** \( I : \forall P P' x y. \)

\[
\left( \forall i, i \in A, x (P_i) \prec y \right) \implies x (P'_i) \prec y;
\]

\[
\forall i, i \in A, x (P_i) \preceq y \implies x (P'_i) \preceq y;
\]

\[
\exists k \in A, (x (P_k) \preceq x) \lor (y (P_k) \preceq y) \lor (y (P_k) \prec x) \lor (x (P_k) \prec y);
\]

\[
\implies (x (scf P) \preceq y) \implies (x (scf P') \prec y)
\]

**shows:** positively-responsive \( A \)

**Proof:**

**Lemma:** positively-responsiveD:

\[
\left( \forall P P' x y, \text{profile } A \text{ is } P \land \text{profile } A \text{ is } P' \land x \in A \land y \in A \right) \\
\land \left( \forall i, i \in A, x (P_i) \prec y \right) \implies x (P'_i) \prec y;
\]

\[
\forall i, i \in A, x (P_i) \preceq y \implies x (P'_i) \preceq y;
\]

\[
\exists k \in A, (x (P_k) \preceq x) \lor (y (P_k) \preceq x) \lor (y (P_k) \prec x) \lor (x (P_k) \prec y);
\]

\[
\implies (x (scf P) \preceq y) \implies (x (scf P') \prec y)
\]

**Proof:**

6.2 The Method of Majority Decision satisfies May’s conditions

The method of majority decision (MMD) says that if the number of individuals who strictly prefer \( x \) to \( y \) is larger than or equal to those who strictly prefer the converse, then \( x R y \). Note that this definition only makes sense for a finite population.

**Definition:** MMD :: \( \langle a, i \rangle \) SCF where

\[
\text{MMD is } P \equiv \{ (x, y) \cdot \text{card } \{ i \in A, x (P_i) \prec y \} \geq \text{card } \{ i \in A, y (P_i) \prec x \} \}
\]

The first part of May’s Theorem establishes that the conditions are consistent, by showing that they are satisfied by MMD.
lemma MMD-l2r:
fixes A :: 'a set
and Is :: 'i set
assumes finiteIs: finite Is
shows SCF (MMD Is) A Is universal-domain
and anonymous (MMD Is) A Is
and neutral (MMD Is) A Is
and positively-responsive (MMD Is) A Is
(proof)

6.3 Everything satisfying May’s conditions is the Method of Majority Decision

Now show that MMD is the only SCF that satisfies these conditions.

Firstly develop some theory about exchanging alternatives \( x \) and \( y \) in profile \( P \).

definition swapAlts :: 'a ⇒ 'a ⇒ 'a ⇒ 'a where
swapAlts a b u ≡ if u = a then b else if u = b then a else u

lemma swapAlts-in-set-iff: \( \{ a, b \} \subseteq A \implies \text{swapAlts} a b u \in A \iff u \in A \)
(proof)

definition swapAltsP :: ('a, 'i) Profile ⇒ 'a ⇒ 'a ⇒ ('a, 'i) Profile where
swapAltsP P a b ≡ (\lambda i. \{ (u, v) . (swapAlts a b u, swapAlts a b v) \in P i \})

lemma swapAltsP-ab: \( a(P i) \preceq b \iff \text{swapAltsP} P a b i \preceq a \preceq \text{swapAltsP} P a b i \)
(proof)

lemma profile-swapAltsP:
assumes profileP: profile A Is P
and abA: \( \{ a, b \} \subseteq A \)
shows profile A Is (swapAltsP P a b)
(proof)

lemma profile-bij-profile:
assumes profileP: profile A Is P
and bijf: bij-betw f Is Is
shows profile A Is (P ◦ f)
(proof)

The locale keeps the conditions in scope for the next few lemmas. Note how weak the constraints on the sets of alternatives and individuals are; clearly there needs to be at least two alternatives and two individuals for conflict to occur, but it is pleasant that the proof uniformly handles the degenerate cases.

locale May =
fixes A :: 'a set
fixes Is :: 'i set
assumes finiteIs: finite Is
fixes scf :: ('a, 'i) SCF
assumes SCF: SCF scf A Is universal-domain
and anonymous: anonymous scf A Is
and neutral: neutral scf A Is
and positively-responsive: positively-responsive scf A Is

begin

Anonymity implies that, for any pair of alternatives, the social choice rule can only depend on the number of individuals who express any given preference between them. Note we also need iia, implied by neutrality, to restrict attention to alternatives \( x \) and \( y \).

lemma anonymous-card:
assumes profileP: profile A Is P
and profileP': profile A Is P'
ad xzA: basw [x,y] A
and xztally: card \{ i ∈ Is. x (P i)≺ y \} = card \{ i ∈ Is. x (P' i)≺ y \}
ad yztally: card \{ i ∈ Is. y (P i)≺ x \} = card \{ i ∈ Is. y (P' i)≺ x \}

shows \( x (scf P) ≦ y \leftrightarrow x (scf P') ≦ y \)
(proof)

Using the previous result and neutrality, it must be the case that if the tallies are tied for alternatives \( x \) and \( y \) then the social choice function is indifferent between those two alternatives.

lemma anonymous-neutral-indifference:
assumes profileP: profile A Is P
and xzA: basw [x,y] A
and tallyP: card \{ i ∈ Is. x (P i)≺ y \} = card \{ i ∈ Is. y (P i)≺ x \}

shows \( x (scf P) ≈ y \)
(proof)

Finally, if the tallies are not equal then the social choice function must lean towards the one with the higher count due to positive responsiveness.

lemma positively-responsive-prefer-witness:
assumes profileP: profile A Is P
and xzA: basw [x,y] A
and tallyP: card \{ i ∈ Is. x (P i)≺ y \} > card \{ i ∈ Is. y (P i)≺ x \}

obtains \( P' k \)
where profile A Is P'
and \( i. \left[ i ∈ Is; x (P' i)≺ y \right] ⇒ x (P i)≺ y \)
and \( i. \left[ i ∈ Is; x (P' i)≈ y \right] ⇒ x (P i)≺ y \)
and \( k ∈ Is ∧ x (P' k) ≈ y ∧ x (P k) ≺ y \)
and card \{ i ∈ Is. x (P' i)≺ y \} = card \{ i ∈ Is. y (P' i)≺ x \}
(proof)

lemma positively-responsive-prefer:
assumes profileP: profile A Is P
and xzA: basw [x,y] A
and tallyP: card \{ i ∈ Is. x (P i)≺ y \} > card \{ i ∈ Is. y (P i)≺ x \}

shows \( x (scf P) ≺ y \)
(proof)

lemma MMD-r2l:
May’s original paper [May52] goes on to show that the conditions are independent by exhibiting choice rules that differ from MMD and satisfy the conditions remaining after any particular one is removed. I leave this to future work.

May also wrote a later article [May53] where he shows that the conditions are completely independent, i.e. for every partition of the conditions into two sets, there is a voting rule that satisfies one and not the other.

There are many later papers that characterise MMD with different sets of conditions.

### 6.4 The Plurality Rule

Goodin and List [GL06] show that May’s original result can be generalised to characterise plurality voting. The following shows that this result is a short step from Sen’s much earlier generalisation.

**Plurality voting** is a choice function that returns the alternative that receives the most votes, or the set of such alternatives in the case of a tie. Profiles are restricted to those where each individual casts a vote in favour of a single alternative.

**type-synonym** $(\mathit{a}, \mathit{i}) \mathit{SVProfile} = \mathit{i} \Rightarrow \mathit{a}$

**definition** ssvprofile :: \(\mathit{a} \set\Rightarrow \mathit{i} \set\Rightarrow (\mathit{a}, \mathit{i}) \mathit{SVProfile} \Rightarrow \mathit{bool} \) where

\[
\text{ssvprofile } \mathcal{A} \text{ Is } \mathcal{F} \equiv \mathcal{F} \neq \emptyset \land \mathcal{F} \set \subseteq \mathcal{A}
\]

**definition** plurality-rule :: \(\mathit{a} \set\Rightarrow \mathit{i} \set\Rightarrow (\mathit{a}, \mathit{i}) \mathit{SVProfile} \Rightarrow \mathit{a} \set \) where

\[
\text{plurality-rule } \mathcal{A} \text{ Is } \mathcal{F} \equiv \{ x \in \mathcal{A} . \forall y \in \mathcal{A} . \text{card } \{ i \in \mathcal{I} . F i = x \} \geq \text{card } \{ i \in \mathcal{I} . F i = y \} \}
\]

By translating single-vote profiles into RPRs in the obvious way, the choice function arising from MMD coincides with traditional plurality voting.

**definition** MMD-plurality-rule :: \(\mathit{a} \set\Rightarrow \mathit{i} \set\Rightarrow (\mathit{a}, \mathit{i}) \mathit{Profile} \Rightarrow \mathit{a} \set \) where

\[
\text{MMD-plurality-rule } \mathcal{A} \text{ Is } \mathcal{P} \equiv \text{choiceSet } \mathcal{A} (\text{MMD Is } \mathcal{P})
\]

**definition** single-vote-to-RPR :: \(\mathit{a} \set \Rightarrow \mathit{a} \Rightarrow \mathit{a} \mathit{RPR} \) where

\[
\text{single-vote-to-RPR } A \mathit{a} \equiv \{ (a, x) \mid x. x \in A \} \cup (A - \{a\}) \times (A - \{a\})
\]

**lemma** single-vote-to-RPR-iff:

\[
\text{[ } a \in A; x \in A; a \neq x \text{ ] } \Rightarrow (a (\text{single-vote-to-RPR } A b) \prec x) \iff (b = a)
\]

**lemma** plurality-rule-equiv:

\[
\text{plurality-rule } \mathcal{A} \text{ Is } \mathcal{F} = \text{MMD-plurality-rule } \mathcal{A} \text{ Is } (\text{single-vote-to-RPR } A \circ \mathcal{F})
\]

Thus it is clear that Sen’s generalisation of May’s result applies to this case as well.

Their paper goes on to show how strengthening the anonymity condition gives rise to a characterisation of approval voting that strictly generalises May’s original theorem. As this
requires some rearrangement of the proof I leave it to future work.

7 Bibliography

References


