Towards a Logic of Social Welfare[1]

Thomas Ågotnes, Wiebe van der Hoek, and Michael Wooldridge

Abstract

We present a formal logic of social welfare functions. The logical language is syntactically simple, but expressive enough to express interesting and complicated properties of social welfare functions involving, e.g., quantification over both preference relations and over individual alternatives, such as Arrow's thorem.

1 Introduction

In the recent years there has been a great deal of interest in the logical aspects of *societies*. For example, Alternating-time Temporal Logic (ATL) [1] and Coalition Logic (CL) [11] can be used to reason about the strategic abilities of individual agents and of coalitions. There is a close connection between these logics and game theory. A related field which, like game theory, also is concerned with social interaction, is social choice theory. A key issue in the latter field is the construction of *social welfare functions, (SWFs)*, mapping individual preferences into "social preferences". Many of the most well known results in social choice theory are impossibility results such as Arrow's theorem [3]: there is no SWF that meets all of a certain number of reasonable conditions. Formal logics related to social choice have focused mostly on the logical representation of preferences when the set of alternatives is large and on the computational properties of computing aggregated preferences for a given representation [7, 8, 9].

In this paper, we present a formal logic which makes it possible to explicitly represent and reason about individual preferences and social preferences. The main differences to the logics mentioned above are as follows. First, the logical language is interpreted directly by social welfare functions and thus that formulae can be read as properties of such functions; second, that preferences are represented in a more abstract way; and, third, that the expressive power is sufficient for interesting problems as discussed below.

Motivations for modeling social choice using logic are manyfold. In particular, logic enables formal *knowledge representation* and *reasoning*. For example, in multiagent systems [13], agents must be able to represent and reason about propositions involving other agents' preferences and preference aggregation. For social choice theory, logic can enable tools for, e.g., mechanically generating proofs, checking the soundness of proofs, mechanically generating possibly in-

[1] An almost identical version of this paper was presented at *the 7th conference on Logic and the Foundations of Game and Decision Theory (LOFT 06)*.

teresting theorems, checkin gproperties of particular social welfare functions, etc.

An example of a property of (some) social welfare functions is so-called *independence of irrelevant alternatives (IIA)*: given two preference profiles and two alternatives, if for each agent the two alternatives have the same order in the two preference profiles, then the two alternatives must have the same order in the two preference relations resulting from applying the SWF to the two preference profiles, respectively. From this example it seems that a formal language about SWFs should be able to express:

- Quantification on several levels: over alternatives; over preference profiles, i.e., over relations over alternatives (second-order quantification); and over agents.
- Properties of preference relations for different agents, and properties of several different preference relations for the same agent in the same formula.
- Comparison of different preference relations.
- The preference relation resulting from applying a SWF to other preference relations.

From these points it seems that such a language would be complex (in particular, they seem to rule out a "standard" propositional modal logic). However, perhaps surprisingly, the language we present in this paper is syntactically and semantically rather simple; and yet the language is, nevertheless, expressive enough to give an elegant and succinct expression of properties such as IIA.

In the next section, we introduce preference relations and social welfare functions. We formally define certain well known potential properties of SWFs, and give a statement of Arrow's theorem. In Section 3 we present the syntax and semantics of our logic, and discuss the complexity of the model checking problem. We show how the mentioned properties can be expressed in the logical language in Section 4. In particular, we show that we can express the statement of Arrow's theorem as a formula – as a result of the theorem, this formula is valid in our logic. In Section 5 we discuss some other valid properties of the logic, and briefly discuss how some of the properties can be expressed in the modal logic *arrow logic* (which originally is about arrows and not about Arrow!). We conclude in Section 6.

2 Social Welfare Functions

Social welfare functions (SWFs) are usually defined in terms of ordinal preference structures, rather than cardinal structures such as utility functions. An

SWF takes as input a preference relation, a binary relation over some set of alternatives, for each agent, and outputs another preference relation representing the aggregated preferences.

The most well known result about SWFs is Arrow's theorem [3]. Many variants of the theorem appears in the literature, differing in assumptions about the preference relations. In this paper, we take the assumption that all preference relations are linear orders, i.e., that neither agents nor the aggregated preference can be indifferent between distinct alternatives. This gives one of the simplest formulations of Arrow's theorem (Theorem 1 below). Cf., e.g., [4] for a discussion and more general formulations.

Formally, let A be a set of *alternatives*. We henceforth implicitly assume that there is always at least two alternatives. A *preference relation* (over A) is, here, a total (linear) order on A, i.e., a relation R over A which is antisymmetric (i.e., $(a,b) \in R$ and $(b,a) \in R$ implies that $a = b$), transitive (i.e., $(a,b) \in R$ and $(b,c) \in R$ implies that $(a,c) \in R$), and total (i.e., either $(a,b) \in R$ or $(b,a) \in R$ for every pair of alternatives a and b). We sometimes use the infix notation aRb for $(a,b) \in R$. The set of preference relations over alternatives A is denoted $L(A)$. Alternatively, we can view $L(A)$ as the set of all permutations of A. Thus, we shall sometimes use a permutation of A to denote a member of $L(A)$. For example, when $A = \{a,b,c\}$, we will sometimes use the expression acb to denote the relation $\{(a,c),(a,b),(c,b),(a,a),(b,b),(c,c)\}$. aRb means that b is preferred over a if a and b are different. R^s denotes the non-reflexive version of R, i.e., $R^s = R \setminus \{(a,a) : a \in A\}$. aR^sb means that b is preferred over a and that $a \neq b$.

Let n be a number of *agents*; we write Σ for the set $\{1,\ldots,n\}$. A *preference profile* for Σ over alternatives A is a tuple $(R_1,\ldots,R_n) \in L(A)^n$.

A *social welfare function* (SWF) is a function

$$F : L(A)^n \to L(A)$$

mapping each preference profile to an aggregated preference relation. The class of all SWFs over alternatives A is denoted $\mathcal{F}(A)$.

Commonly discussed properties a SWF F can have include:

PO $\forall_{(R_1,\ldots,R_n)\in L(A)^n}\forall_{a\in A}\forall_{b\in A}((\forall_{i\in\Sigma}aR_i^sb) \Rightarrow aF(R_1,\ldots,R_n)^sb)$ (pareto optimality)

ND $\neg\exists_{i\in\Sigma}\forall_{(R_1,\ldots,R_n)\in L(A)^n}F(R_1,\ldots,R_n) = R_i$ (non-dictatorship)

IIA $\forall_{(R_1,\ldots,R_n)\in L(A)^n}\forall_{(S_1,\ldots,S_n)\in L(A)^n}\forall_{a\in A}\forall_{b\in A}((\forall_{i\in\Sigma}(aR_ib \Leftrightarrow aS_ib)) \Rightarrow (aF(R_1,\ldots,R_n)b \Leftrightarrow aF(S_1,\ldots,S_n)b))$ (independence of irrelevant alternatives)

Arrow's theorem says that the three properties above are inconsistent if there are more than two alternatives.

Theorem 1 (Arrow)**.** *If there are more than two alternatives, no SWF has all the properties **PO**, **ND** and **IIA**.*

We now introduce a formal language in which properties such the above can be expressed.

3 The Logic

We now present a logical language and its interpretation in SWFs. The language is syntactically simple, but the representation of preferences is unconventional and we will therefore discuss the main points before giving formal definitions.

An example of a formula is

$$\Diamond \boxdot (r_1 \leftrightarrow r) \tag{1}$$

A formula denotes a property of a SWF. The formula (1) says that there exist ($\Diamond$) preferences for the agents such that for all ($\boxdot$) pairs of alternatives, agent 1 (r_1) and the aggregated preferences (r) agree on the relative ranking of the two alternatives (i.e., on which of the two is better than the other).

While a formula is interpreted in a SWF, a subformula may be interpreted in additional structures depending on which quantifiers ($\Diamond$,$\Box$,$\diamondsuit$,$\boxdot$) the subformula is in the scope of. Here is a detailed description of the intended meaning of the parts of the formula (1):

r_1 : A statement about the combination of a SWF F, a preference profile $(R_1, \ldots, R_n)$ and a pair of alternatives (a, b). It says that according to the preference profile, agent 1 prefers b (the last element in the pair) over a (the first element in the pair).

r : A statement about the combination of a SWF F, a preference profile $(R_1, \ldots, R_n)$ and a pair of alternatives (a, b). It says that according to the preference relation resulting from applying the SWF to the preference profile, b is preferred over a.

$\boxdot(r_1 \leftrightarrow r)$: A statement about the combination of a SWF F and a preference profile $(R_1, \ldots, R_n)$. It says that for every pairs of alternatives, $(r_1 \leftrightarrow r)$ holds wrt. the SWF, preference profile, and pair of alternatives.

$\Diamond \boxdot (r_1 \leftrightarrow r)$: A statement about a SWF F. It says that there exists a preference profile such that for all pairs (a, b) of alternatives, b is preferred over a in the aggregation (by the SWF) of the preference profile if and only if agent 1 prefers b over a.

3.1 Syntax

The logical language is parameterised by the number of agents n, in addition to a stock of symbols $\Pi = \{r, s, \ldots\}$. A symbol $r \in \Pi$ will be used to refer to a preference profile $R \in L(A)^n$. In the example above, formula (1), we only used one symbol r, but as we shall see it is useful to be able to reason about several different preference profiles at the same time. Formally, we define three languages: $\mathcal{L}$ expresses properties of SWFs and is the language we are ultimately interested in. $\mathcal{L}$ is defined in terms of $\mathcal{L}_2$. $\mathcal{L}_2$ expresses properties of preference profiles (one for each member of Π) relative to a SWF, and is again

defined in terms of $\mathcal{L}_3$. $\mathcal{L}_3$ expresses properties of a pair $(a, b) \in A^2$ relative to a SWF and some preference profiles.

$\mathcal{L}$: $\phi ::= \Box\psi \mid \neg\phi \mid \phi_1 \wedge \phi_2$

$\mathcal{L}_2$: $\psi ::= \boxdot\gamma \mid \neg\psi \mid \psi_1 \wedge \psi_2$

$\mathcal{L}_3$: $\gamma ::= r_i \mid r \mid \neg\gamma \mid \gamma_1 \wedge \gamma_2$ where $i \in \Sigma$ and $r \in \Pi$

We use the duals: $\Diamond\psi \equiv \neg\Box\neg\psi$ and $\diamondsuit\gamma \equiv \neg \boxdot \neg\gamma$, in addition to the usual derived propositional connectives.

Note that we do not allow arbitrary nesting of the quantifiers.

3.2 Semantics

A *profile function*

$$\delta : \Pi \to L(A)^n$$

associates a preference profile $\delta(r) = (R_1, \ldots, R_n)$ with each symbol $r \in \Pi$. If $\delta(r) = (R_1, \ldots, R_n)$, we write $\delta_i(r)$ for R_i. The set of all profile functions over A and Π is denoted $\Delta(A, \Pi)$ (or just Δ). $\mathcal{L}$ is interpreted in an SWF $F \in \mathcal{F}(A)$ as follows:

$$\begin{array}{lcl} (A,F) \models \Box\psi & \Leftrightarrow & \forall_{\delta\in\Delta}(A,F,\delta) \models \psi \\ (A,F) \models \neg\phi & \Leftrightarrow & (A,F) \not\models \phi \\ (A,F) \models \phi_1 \wedge \phi_2 & \Leftrightarrow & (A,F) \models \phi_1 \text{ and } (A,F) \models \phi_2 \end{array}$$

$\mathcal{L}_2$ is interpreted in an SWF F and a profile function δ as follows:

$$\begin{array}{lcl} (A,F,\delta) \models \boxdot\gamma & \Leftrightarrow & (\forall_{(a,b)\in A\times A} a \neq b \Rightarrow (A,F,\delta,(a,b)) \models \gamma) \\ (A,F,\delta) \models \neg\psi & \Leftrightarrow & (A,F,\delta) \not\models \psi \\ (A,F,\delta) \models \psi_1 \wedge \psi_2 & \Leftrightarrow & (A,F,\delta) \models \psi_1 \text{ and } (A,F,\delta) \models \psi_2 \end{array}$$

$\mathcal{L}_3$ is interpreted in an SWF F, a profile function δ and a pair of distinct alternatives (a, b) as follows:

$$\begin{array}{lcl} (A,F,\delta,(a,b)) \models r_i & \Leftrightarrow & (a,b) \in \delta_i(r) \\ (A,F,\delta,(a,b)) \models r & \Leftrightarrow & (a,b) \in F(\delta(r)) \\ (A,F,\delta,(a,b)) \models \neg\gamma & \Leftrightarrow & (A,F,\delta,(a,b)) \not\models \gamma \\ (A,F,\delta,(a,b)) \models \gamma_1 \wedge \gamma_2 & \Leftrightarrow & (A,F,\delta,(a,b)) \models \gamma_1 \text{ and } (A,F,\delta,(a,b)) \models \gamma_2 \end{array}$$

Given a set of alternatives A, as formula is *valid on* A if $A, F \models \phi$ for all $F \in \mathcal{F}(A)$. A formula ϕ is *valid*, written $\models \phi$, if $A \models \phi$ for all A.

3.3 Model Checking

Most implemented systems for reasoning about cooperation are based on *model checking* [6, 2]. Roughly speaking, the model checking problem for a given logic is as follows: Given a formula ϕ of the logic, and a model/interpretation M for the logic, is it the case that $M \models \phi$? For our logic, we have three model checking problems, for the languages $\mathcal{L}$, $\mathcal{L}_2$, and $\mathcal{L}_3$ respectively. For example, the $\mathcal{L}$ model checking problem is as follows:

Given a set A of alternatives, a social welfare function $F \in \mathcal{F}(A)$, and a formula ϕ of $\mathcal{L}$, is it the case that $(A, F) \models \phi$?

The model checking problems for $\mathcal{L}_2$ and $\mathcal{L}_3$ may be derived similarly. The model checking problem for $\mathcal{L}$ can be understood as asking whether the property of social welfare functions expressed by the formula ϕ is true of the given social welfare function F. For example, given the formula PO discussed in the next section, checking whether $(A, F) \models PO$, is exactly the problem of checking whether F has the Pareto Optimality property.

The *complexity* of the model checking problem for $\mathcal{L}$ depends upon the representation chosen for the function F. The simplest representation will be an *extensive* one, where the function is enumerated as the set of all pairs of the form (i, o), where i is an input to F and $o = F(i)$ is the corresponding output. The obvious "catch" is that this representation of F must list the value of F for every input: and there will be exponentially many (in the number of alternatives) possible inputs. So, an alternative is to assume a *succinct* representation for F. We consider one such alternative, where F is represented as a polynomially bounded deterministic two-tape Turing machine. Roughly, this can be understood as representing F as a program computing the social welfare function which is guaranteed to terminate with an output in polynomial time. (Of course, it may be the case that there are F's which cannot be so represented.)

Now, it is easy to see that, assuming the extensive representation, the model checking problems for $\mathcal{L}$, $\mathcal{L}_2$, and $\mathcal{L}_3$ may be solved in deterministic polynomial time. However, since the inputs are exponentially large, this result is perhaps misleading. We can show the following.

Proposition 1. *For the succinct representation of SWFs, the model checking problem for $\mathcal{L}$ is NP-hard even for formulae of the form $\Box\psi$.*

Proof. We reduce SAT, the problem of determining whether a given formula ξ of propositional logic over variables $x_1, \ldots, x_k$ is satisfied by some assignment of truth/falsity to its Boolean variables $x_1, \ldots, x_k$ [10]. Given an instance $\xi(x_1, \ldots, x_k)$ of SAT, we create an instance of model checking for $\mathcal{L}$ as follows. First, we create just two alternatives, $A = \{a, b\}$; for each Boolean variable x_i we create an agent, and define an $\mathcal{L}_2$ variable r_i . We then define F so that it produces the ranking (a, b). Next, we define $\xi^{\#}$ to be the formula obtained from ξ by systematically replacing the variable x_i by r_i . We then define the formula ζ that is input to the $\mathcal{L}$ model checking problem to be:

$$\zeta = \Diamond\Diamond\xi^{\#}.$$

That the formula ζ is true given F and A as defined iff ξ is satisfiable is now straightforward. □

Notice that for the succinct representation, the model checking problems for $\mathcal{L}_2$ and $\mathcal{L}_3$ are easily seen to be solvable in deterministic polynomial time. The general model checking problem for $\mathcal{L}$ for succinct representations is also easily

seen to be in Δ_2^p (the class of problems solvable in polynomial time assuming an oracle for problems in NP).

4 Examples

The proofs of the following propositions are straightforward.

Pareto optimality can be expressed as follows:

$$PO = \Box \boxdot ((r_1 \wedge \cdots \wedge r_n) \rightarrow r) \tag{2}$$

Proposition 2. *Let $F \in \mathcal{F}(A)$. $(A, F) \models PO$ iff F has the property **PO**.*

Non-dictatorship can be expressed as follows:

$$ND = \bigwedge_{i \in \Sigma} \Diamond \circledcirc \neg (r \leftrightarrow r_i) \tag{3}$$

Proposition 3. *Let $F \in \mathcal{F}(A)$. $(A, F) \models ND$ iff F has the property **ND**.*

Independence of irrelevant alternatives can be expressed as follows:

$$IIA = \Box \boxdot ((r_1 \leftrightarrow s_1 \wedge \cdots \wedge r_n \leftrightarrow s_n) \rightarrow (r \leftrightarrow s)) \tag{4}$$

Proposition 4. *Let $F \in \mathcal{F}(A)$. $(A, F) \models IIA$ iff F has the property **IIA**.*

4.1 Cardinality of Alternatives

The properties expressed above are properties of social welfare functions. We turn to look now at which properties of the set of *alternatives* A we can express. Note that we cannot refer to particular alternatives directly in the logical language. Properties involving *cardinality* is often of interest, for example in Arrow's theorem. Let:

$$MT2 = \Diamond (\circledcirc (r_1 \wedge s_1) \wedge \circledcirc (r_1 \wedge \neg s_1))$$

Proposition 5. *Let $F \in \mathcal{F}(A)$. $|A| > 2$ iff $(A, F) \models MT2$.*

Proof. For the direction to the left, let $(A, F) \models MT2$. Thus, there is a δ such that there exists $(a^1, b^1), (a^2, b^2) \in A \times A$, where $a^1 \neq b^1$, and $a^2 \neq b^2$, such that (i) $(a^1, b^1) \in \delta_1(r)$, (ii) $(a^1, b^1) \in \delta_1(s)$, (iii) $(a^2, b^2) \in \delta_1(r)$ and (iv) $(a^2, b^2) \notin \delta_1(s)$. From (ii) and (iv) we get that $(a^1, b^1) \neq (a^2, b^2)$, and from that and (i) and (iii) it follows that $\delta_1(r)$ contains two different pairs each having two different elements. But that is not possible if $|A| = 2$, because if $A = \{a, b\}$ then $L(A) = \{ab, ba\} = \{ \{(a, b), (a, a), (b, b)\}, \{(b, a), (a, a), (b, b) \}$, so it cannot be that $\delta_1(r) \in L(A)$.

For the direction to the right, let $|A| > 2$; let a, b, c be three different elements of A. Let $\delta_1(r) = abc$ and $\delta_1(s) = acb$. Now, for any F, $(A, F, \delta, (a, b)) \models r_1 \wedge s_1$ and $(A, F, \delta, (b, c)) \models r_1 \wedge \neg s_1$. Thus, $(A, F) \models MT2$, for any F. □

Other interesting properties hold when the cardinality of the set of alternatives is finite and fixed:

Example 1. *Consider the case when $\Pi = \{r\}$, there are two agents, and three alternatives. Then the following holds (for every A with $|A| = 3$):*

$$A \models \Box(\diamond(r \wedge r_1 \wedge r_2) \wedge \diamond(r \wedge \neg r_1 \wedge r_2) \wedge \diamond(r \wedge r_1 \wedge \neg r_2) \rightarrow \boxdot(r \rightarrow (r_1 \vee r_2))$$

This validity says that, for any SWF and any preferences, if there exist pairs of alternatives on which (i) both agents agree with the SWF, (ii) only agent 1 agrees with the SWF and (iii) only agent 2 agrees with the SWF, then for every pair at least one of the agents must agree with the SWF.

Here is a justification. There are eight "descriptors" of the form $r_1 \wedge r_2 \wedge r$, $\neg r_1 \wedge r_2 \wedge r$, etc., i.e. conjunctions of literals completely describing preferences over a pair. But, given a SWF F and a profile function δ, a $\mathcal{L}_3$ formula on the form $\diamond d$ where d is a descriptor holds for exactly six of the eight descriptors. To see this, observe that with three alternatives, there are only six distinct pairs, and two different descriptors cannot be true in the same pair. Furthermore, these six descriptors consists of three pairs of complementary descriptors, where the complement of a descriptor is obtained by changing the sign of each literal: if d is true in a pair (a, b), then the complement of d is true in the pair (b, a). So $\diamond d$ can be true in a given SWF and profile function for only three different non-complimentary descriptors d at the same time. In the example formula above, the three descriptors in the antecedent of the implications are non-complimentary, and the fourth descriptor in the consequent is non-complimentary to these three as well, so the latter cannot be true at the same time as all the three former.

4.2 Arrow's Theorem

We now have everything we need to express Arrow's statement as a formula. It follows from his theorem that the formula is valid.

Theorem 2.

$$\models MT2 \rightarrow \neg(PO \wedge ND \wedge IIA)$$

Proof. Let A be a set of alternatives, $F \in \mathcal{F}(A)$, and $(A, F) \models MT2$. By Proposition 5, A has more than two alternatives. By Arrow's theorem, F cannot have all the properties **PO**, **ND** and **IIA**. By Propositions 2, 3 and 4, $(A, F) \models \neg PO \vee \neg ND \vee \neg IIA$. □

5 Logical Properties

We here take a closer look at additional universal properties of SWFs expressible in the logic: which $\mathcal{L}$ formulae are valid?

First – trivially – we have that

$$\models \phi \qquad \phi \text{ instance of prop. tautology} \qquad (Prop_1)$$
$$\models \Box\psi \qquad \psi \text{ instance of prop. tautology} \qquad (Prop_2)$$
$$\models \Box\boxdot\gamma \qquad \gamma \text{ instance of prop. tautology} \qquad (Prop_3)$$

It is also easy to see that we have the K axiom, on both "level" $\mathcal{L}$ and $\mathcal{L}_2$:

$$\models \Box(\psi_1 \to \psi_2) \to (\Box\psi_1 \to \Box\psi_2) \qquad (K_1)$$
$$\models \Box\,(\boxdot(\psi_1 \to \psi_2) \to (\boxdot\psi_1 \to \boxdot\psi_2)) \qquad (K_2)$$

However, the remainding principle of normal modal logics (cf., e.g., [5]), *uniform substitution*, does *not* hold for our logic. A counter example is the fact that the following is valid:

$$\Box\Diamond r \tag{5}$$

– no matter what preferences the agents have, the SWF will always rank some alternative over another – while this is not valid:

$$\Box\Diamond(r \wedge r_1) \tag{6}$$

– the SWF will not necessarily rank any two alternatives in the same order as agent 1.

The formuale in (5) and (6) have the same pattern of quantifiers ($\Box\Diamond$), and a natural question is then for which γ the formula $\Box\Diamond\gamma$ is valid. Theorem 3 below partly answers that question (both claims above about validity and non-validity of (5) and (6), respectivelly, thus follow from that theorem). First some definitions and an intermediate result.

We shall sometimes treat $\mathcal{L}_3$ as the language of propositional logic, with atomic propositions

$$Atoms(\Pi, \Sigma) = \{r_i, r : r \in \Pi, i \in \Sigma\}$$

(or just *Atoms* when Π and Σ are clear from context). A propositional valuation will simply be represented as a subset V of *Atoms*. We reuse the $\models$ symbol (no confusion can occur), and write $V \models \gamma$ when V is a valuation satisfying (in the classical truth-functional sense) a formula $\gamma \in \mathcal{L}_3$, as well as $\models \gamma$ when $V \models \gamma$ for all $V \subseteq Atoms$. We use $Lit(\Pi, \Sigma)$ (or just *Lit*) to denote the set of literals: $Lit(\Pi, \Sigma) = Atoms(\Pi, \Sigma) \cup \{\neg q : q \in Atoms(\Pi, \Sigma)\}$. When $\gamma \in \mathcal{L}_3$, we use $\overline{\gamma}$ to denote the result of negating every occurrence of an atom in γ.[2] Formally: $\overline{q} = \neg q$ when $q \in Atoms$; $\overline{\neg\gamma} = \neg\overline{\gamma}$; $\overline{\gamma_1 \wedge \gamma_2} = \overline{\gamma_1} \wedge \overline{\gamma_2}$.

The proof of the following Lemma is straightforward.

Lemma 1. *For any A, F, δ, any pair $a, b \in A$, $a \neq b$, and any $\mathcal{L}_3$ formula γ:*

$$(A, F, \delta, (a, b)) \models \overline{\gamma} \Leftrightarrow (A, F, \delta, (b, a)) \models \gamma$$

[2]The "overline" notation is sometimes used to denote negation, note that our use is different.

Theorem 3. *For any $k \geq 1$, and any $\gamma_1, \ldots, \gamma_k \in \mathcal{L}_3$:*

$$\models \Box(\Diamond\gamma_1 \vee \cdots \vee \Diamond\gamma_k) \Leftrightarrow \models \gamma_1 \vee \overline{\gamma_1} \vee \cdots \vee \gamma_k \vee \overline{\gamma_k}$$

Proof. Let $\gamma_1, \ldots, \gamma_k \in \mathcal{L}_3$.

For the direction to the left, let A be a set of alternatives, F an SWF, and $\delta \in \Delta$. Note that $\overline{\gamma_1 \vee \cdots \vee \gamma_k} = \overline{\gamma_1} \vee \cdots \vee \overline{\gamma_k}$. Let $a, b \in A$, $a \neq b$. $(A, F, \delta, (a, b))$ can be seen as a valuation (over *Atoms*), so by the right hand side, $(A, F, \delta, (a, b)) \models (\gamma_1 \vee \cdots \vee \gamma_k) \vee (\overline{\gamma_1 \vee \cdots \vee \gamma_k})$, so either $(A, F, \delta, (a, b)) \models \gamma_1 \vee \cdots \vee \gamma_k$ or $(A, F, \delta, (a, b)) \models \overline{\gamma_1 \vee \cdots \vee \gamma_k}$ (or both). By Lemma 1, either $(A, F, \delta, (a, b)) \models \gamma_1 \vee \cdots \vee \gamma_k$ or $(A, F, \delta, (b, a)) \models \gamma_1 \vee \cdots \vee \gamma_k$ (or both). Thus, there is a j such that either $(A, F, \delta, (a, b)) \models \gamma_j$ or $(A, F, \delta, (b, a)) \models \gamma_j$. It follows that $(A, F, \delta) \models \Diamond\gamma_j$, and thus that $(A, F, \delta) \models \Diamond\gamma_1 \vee \cdots \vee \Diamond\gamma_k$. Since A, F, δ were arbitrary, we have that $\models \Box(\Diamond\gamma_1 \vee \cdots \vee \Diamond\gamma_k)$.

For the direction to the right, we show the contrapositive. Assume that there is a propositional valuation V such that $V \not\models \gamma_1 \vee \overline{\gamma_1} \vee \cdots \vee \gamma_k \vee \overline{\gamma_k}$. Then $V \models \neg(\gamma_1 \vee \cdots \vee \gamma_k)$ and $V \models \neg(\overline{\gamma_1} \vee \cdots \vee \overline{\gamma_k})$. The latter is equivalent to $V \models \overline{\neg(\gamma_1 \vee \cdots \vee \gamma_k)}$. Now, let $A = \{a, b\}$ $(a \neq b)$, and let F and δ be defined as follows:

$$\delta_i(r) = \begin{cases} ab & r_i \in V \\ ba & \text{otherwise} \end{cases} \qquad F(\delta(r)) = \begin{cases} ab & r \in V \\ ba & \text{otherwise} \end{cases}$$

It can easily be seen, by induction over the formula, that V and (a, b) agrees on every $\mathcal{L}_3$ formula, i.e., that for every $\gamma \in \mathcal{L}_3$

$$V \models \gamma \Leftrightarrow (A, F, \delta, (a, b)) \models \gamma \tag{7}$$

Thus, we have that $(A, F, \delta, (a, b)) \models \neg(\gamma_1 \vee \cdots \vee \gamma_k)$. But since $V \models \overline{\neg(\gamma_1 \vee \cdots \vee \gamma_k)}$, we also get $(A, F, \delta, (a, b)) \models \overline{\neg(\gamma_1 \vee \cdots \vee \gamma_k)}$ from (7), and thus that $(A, F, \delta, (b, a)) \models \neg(\gamma_1 \vee \cdots \vee \gamma_k)$ from Lemma 1. Since (a, b) and (b, a) are the only pairs of distinct elements from A, we have that $(A, F, \delta) \models \boxdot\neg(\gamma_1 \vee \cdots \vee \gamma_k)$. From K_2 and $Prop_2$ and $Prop_3$ we get that $(A, F, \delta) \models \boxdot\neg\gamma_1 \wedge \cdots \wedge \boxdot\neg\gamma_k$. This is, again by propositional reasoning, the same as $(A, F, \delta) \models \neg(\Diamond\gamma_1 \vee \cdots \vee \Diamond\gamma_k)$. Thus, we have established that $\not\models \Box(\Diamond\gamma_1 \vee \cdots \vee \Diamond\gamma_k)$. □

Some applications showing both directions of Theorem 3:

$\models \Box\Diamond q$ **for any** $q \in Lit$**:** Both the individual agents and the SWF will always rank some alternative above another and, conversely, some alternative below some other. (5) above is an instance. Justification: if $q \in Lit$, then $\overline{q} = \neg q$, so $\models q \vee \overline{q}$ holds.

$\not\models \Box\Diamond(q_1 \wedge q_2)$ **when** $q_1 \neq q_2 \in Lit$**:** we are not guaranteed that there is a pair of alternatives ranked in the same order by two agents and/or the SWF. (6) above is an instance. Justification: if $q_1 \neq q_2 \in Lit$, then $\overline{q_1 \wedge q_2} = \neg q_1 \wedge \neg q_2$. But it is not the case that $(q_1 \wedge q_2) \vee (\neg q_1 \wedge \neg q_2)$ is a propositional tautology.

$\models \Box(\boxdot(r_1 \vee r_2) \rightarrow \diamondsuit(r_1 \wedge \neg r_2))$**:** if, given preferences of agents and a SWF, for any two alternatives it is always the case that either agent 1 or agent 2 prefers the second alternative over the first, then there must exist a pair of alternatives for which the two agents disagree. Justification: the formula in question is equivalent to $\Box(\diamondsuit\gamma_1 \vee \diamondsuit\gamma_2)$, where $\gamma_1 = \neg r_1 \wedge \neg r_2$ and $\gamma_2 = r_1 \wedge \neg r_2$. $\overline{\gamma_1} = \neg\neg r_1 \wedge \neg\neg r_2$ and $\overline{\gamma_2} = \neg r_1 \wedge \neg\neg r_2$, so $\gamma_1 \vee \gamma_2 \vee \overline{\gamma_1} \vee \overline{\gamma_2}$ is a propositional tautology.

The following theorem characterises all valid formulae of the form $\Box\boxdot\gamma$: γ is a propositional tautology. The proof is straightforward.

Theorem 4.

$$\models \Box\boxdot\gamma \Leftrightarrow \models \gamma$$

Properties involving other combinations of quantifiers include:

$\models \diamondsuit\diamondsuit(r_1 \wedge r_2)$**:** There exist preference relations such that agents 1 and 2 agree on some pair of alternatives.

$\not\models \diamondsuit\diamondsuit(r_1 \wedge r)$**:** There does not necessarily exist preference relations such that agent 1 and the SWF agree on some pair of alternatives.

$\models \diamondsuit\boxdot(r_1 \leftrightarrow r_2)$**:** There exist preference relations such that agents 1 and 2 always agree.

$\not\models \diamondsuit\boxdot(r_1 \leftrightarrow r)$**:** There does not necessarily exist preference relations such that agent 1 and the SWF always agree.

5.1 Arrow Logic for Arrow's logic

The modal logic *arrow logic* is designed to reason about any object that can be graphically represented as an arrow [12]. Arrows typically represent a transition triggered by the execution of an action or a computer program, or even the dynamic meaning of a discourse, which explains the popularity of arrow logic among computer scientists, philosophers, and linguists. However, arrows can also be thought of as representing a *preference*, which justifies using arrow logic for our study as well. In this section, we only describe how the language and semantics of arrow logic can be used to represent properties of language $\mathcal{L}_3$: all definitions and notation used in this section are taken from [12].

An *arrow frame* is a tuple $\mathcal{F} = \langle W, \mathcal{R}\rangle$ where W, the universe of $\mathcal{F}$, is a set of *arrows*. Sometimes, it is convenient to think about an arrow a as having as start a_0 and end a_1. Moreover, $\mathcal{R}$ is a set of relations on W, which we will discuss shortly. Given a set of atomic propositions P denoting basic properties, in line with standard modal logic, we can then base a model $\mathcal{M} = \langle\mathcal{F}, V\rangle$ on a frame $\mathcal{F}$ by adding a valuation function $V : P \rightarrow 2^W$, with the meaning that $V(p)$ collects those arrows that satisfy property p. For our purposes, we will take $P = Atoms$, representing the agents' preferences r_i and the collective preference

r, where $\mathcal{M}, a \models r_i$ is meant to mean that according to agent i, alternative a_1 is preferred over a_0 . And similarly $\mathcal{M}, a \models r$ denotes that the welfare function has decided upon judging a_1 better than a_0.

In "basic" arrow logic, there are three relations in $\mathcal{R}$. We follow the notation of [12] and denote them by $C \subseteq W \times W \times W$, and $R \subseteq W \times W$ and $I \subseteq W$, respectively. For three arrows a, b and c, when $Cabc$, we say that a is the composition of b and c. Putting it a bit more formal: $Cabc$ iff $a_0 = b_0$, $b_1 = c_0$ & $c_1 = a_1$. The relation R holds between a and b if b is the inverse of a: Rab iff $a_0 = b_1$ & $b_0 = a_1$. Finally, Ia denotes that a is a reflexive arrow: Ia iff $a_0 = a_1$.

Naturally, in the language for basic arrow logic, we have an operator for each of these relations:

$$\varphi := p \mid \delta \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \circ \varphi \mid \otimes\varphi$$

We now immediately give the truth definition of a formula in an arrow:

$$\begin{array}{lll}
\mathcal{M}, a \models p & \text{iff} & a \in V(p) \\
\mathcal{M}, a \models \delta & \text{iff} & Ia \\
\mathcal{M}, a \models \neg\varphi & \text{iff} & not\mathcal{M}, a \models \varphi \\
\mathcal{M}, a \models \varphi \vee \psi & \text{iff} & \mathcal{M}, a \models \varphi \text{ or } \mathcal{M}, a \models \psi \\
\mathcal{M}, a \models \varphi \circ \psi & \text{iff} & \text{for some } b, c\ (Cabc \& \mathcal{M}, b \models \varphi \& \mathcal{M}, c \models \psi) \\
\mathcal{M}, a \models \otimes\varphi & \text{iff} & \text{for some } b\ (Rab \& \mathcal{M}, b \models \varphi)
\end{array}$$

Recall that $P = \{r_1, r_2, \ldots, r_n, r, \ldots\}$, and that $\mathcal{M}, a \models p$ means that according to p, alternative a_1 is better than a_0, where p either refers to one of the agents, or to the aglomorated result.

Properties of Preferences It appears that most properties we used for preferences have an straightforward translation in arrow logic. We list the following:

1. *transitivity*. This property is expressed by $(p \circ p) \rightarrow p$
2. *asymmetry*. This is $p \rightarrow \otimes\neg p$
3. *linearity*. This becomes $p \vee \otimes p$.
4. *irreflexivity*. This is $\neg\delta$
5. *pareto optimality*. $(\bigwedge r_i \leq nr_i) \rightarrow r$
6. *at most* $n + 1$ *alternatives*. This is $\neg \underbrace{(\top \circ (\top \circ (\cdots \circ \top \cdots)))}_{n \times \top}$

Arrow logics are ususally proven complete wrt. an *algebra*. This would mean, in our context, that it might be possible to use algebras as the underlying structures to represent individual and collective preferences. Then, δ is used to take us from one algebra to another, and F determines the collective preference, in each of the algebras.

6 Conclusions

We have presented a logic of social welfare functions, which is syntactically simple but which can express interesting and complicated properties, involving quantification on several levels, such as Arrow's theorem.

In Section 5 we discussed in depth several properties of the logic. These seem to be a good starting point for a complete axiomatisation of the logic, which remains to be found. Also of importance is to investigate the complexity of the satisfiability problem. Further possibilities for future work include the expression of additional results from social choice theory in general, and in particular relaxing the assumptions about linear orders for the preference relations and the expression of more general variants of Arrow's theorem.

It is interesting to observe that the logic can also be easily used to reason about *judgment aggregation*, i.e., about *judgment aggregation rules* which aggregate consistent sets of propositional formulae, each representing the judgments of an individual agent, into a single consistent set of formulae representing the collective judgments. We are currently working on this interpretation, which we feel can help shed light on the relationship between preference aggregation and judgment aggregation by allowing us to compare the logical principles of each.

The relationship between our logic and arrow logic could also be investigated further.

Acknowledgements The research reported in this paper was carried out when the first author was visiting the Department of Computer Science, University of Liverpool. The first author's work was funded by grant 166525/V30 from the Norwegian Research Council.

References

[1] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. In *Proceedings of the 38th IEEE Symposium on Foundations of Computer Science*, pages 100–109, Florida, October 1997.

[2] R. Alur, T. A. Henzinger, F. Y. C. Mang, S. Qadeer, S. K. Rajamani, and S. Taşiran. Mocha: Modularity in model checking. In *CAV 1998: Tenth International Conference on Computer-aided Verification, (LNCS Volume 1427)*, pages 521–525. Springer-Verlag, 1998.

[3] K. J. Arrow. *Social Choice and Individual Values*. Wiley, 1951.

[4] K. J. Arrow, Amartya K. Sen, and Kotaro Suzumura, editors. *Handbook of Social Choice and Welfare*, volume 1. North-Holland, 2002.

[5] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, Cambridge University Press, 2001.

[6] E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking*. MIT Press, 2000.

[7] Celine Lafage and Jérôme Lang. Logical representation of preferences for group decision making. In Anthony G. Cohn, Fausto Giunchiglia, and Bart Selman, editors, *Proceedings of the Conference on Principles of Knowledge Representation and Reasoning (KR-00)*, pages 457–470, S.F., April 11–15 2000. Morgan Kaufman Publishers.

[8] Jérôme Lang. From preference representation to combinatorial vote. In Dieter Fensel, Fausto Giunchiglia, Deborah L. McGuinness, and Mary-Anne Williams, editors, *Proceedings of the Eights International Conference on Principles and Knowledge Representation and Reasoning (KR-02), Toulouse, France, April 22-25, 2002*, pages 277–290. Morgan Kaufmann, 2002.

[9] Jérôme Lang. Logical preference representation and combinatorial vote. *Ann. Math. Artif. Intell*, 42(1-3):37–71, 2004.

[10] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[11] M. Pauly. A modal logic for coalitional power in games. *Journal of Logic and Computation*, 12(1):149–166, 2002.

[12] Y. Venema. A crash course in arrow logic. In M. Marx, M. Masuch, and L. Pólos, editors, *Arrow Logic and Multi-Modal Logic*, pages 3–34. CSLI Publications, Stanford, 1996.

Thomas Ågotnes
Department of Computer Engineering, Bergen University College
P.O.Box 7030, N-5020 Bergen, Norway
Email: tag@hib.no

Wiebe van der Hoek
Department of Computer Science, University of Liverpool
Ashton Building, Ashton Street, Liverpool L69 3BX, UK
Email: wiebe@csc.liv.ac.uk

Michael Wooldridge
Department of Computer Science, University of Liverpool
Ashton Building, Ashton Street, Liverpool L69 3BX, UK
Email: mjw@csc.liv.ac.uk