

Reasoning about Social Choice Functions

Nicolas Troquard, Wiebe van der Hoek, and Michael Wooldridge
Computer Science Department, University of Liverpool, UK

Abstract. We introduce a logic specifically designed to support reasoning about social choice functions. The logic includes operators to capture strategic ability, and operators to capture agent preferences. We establish a correspondence between formulae in the logic and properties of social choice functions, and show that the logic is expressively complete with respect to social choice functions, i.e., that every social choice function can be characterised as a formula of the logic. We prove that the logic is decidable, and give a complete axiomatization. To demonstrate the value of the logic, we show in particular how it can be applied to the problem of determining whether a social choice function is strategy-proof.

1. Introduction

Social choice theory is concerned with collective decision making in situations where the preferences of the decision makers may differ (Arrow et al., 2002). Social choice theorists have developed a range of procedures, such as voting protocols, to support such collective decision making, and have developed a range of criteria with which to characterise the properties of such procedures. Such criteria are usually expressed axiomatically, and a major concern of social choice theory is to study the extent to which decision making procedures do or do not satisfy these axioms (May, 1952; Arrow, 1950; Gibbard, 1973; Satterthwaite, 1975).

In short, the aim of the present paper is to develop a logic that is explicitly intended for reasoning about social choice procedures. We focus on *social choice functions*, a class of social choice procedures that select a single social outcome as a function of individual preferences. *Voting procedures* of the type used in political elections throughout the democratic world are perhaps the best-known examples of social choice functions. A voting procedure determines the winner of an election as a function of the votes cast; votes can be understood as an expression of voter preferences.

One interesting issue that arises in voting procedures is the extent to which voters are incentivised to truthfully report their preferences when voting. For example, suppose we have two voters, 1 and 2, who vote among three candidates, x , y , and z for a role that is currently filled by x . The voting procedure used in this example says that, if there is a unanimously preferred candidate, then that will be chosen, otherwise the candidate x remains. Suppose the true preferences of 1 are given by $z <_1 x <_1 y$ and those of 2 are $x <_2 y <_2 z$. If the social choice function was presented with these true preferences, candidate x would be chosen (since there is no consensus). However, if voter

2 would instead claim his preferences were $x <'_2 z <'_2 y$ while 1 revealed its true preferences, then 2 would be better off, since y would be chosen, rather than x , and agent 2 prefers y over x . This issue suggests the following problem: Can we design a voting procedure that is “immune” to such misrepresentation, i.e., in which a voter can never do any better than by truthfully reporting its preferences? The term *strategy proof* is used to refer to such voting procedures. In fact, fundamental results in social choice theory tell us that there are severe limits to the development of strategy-proof voting procedures (Gibbard, 1973; Satterthwaite, 1975), and for this reason, developing and analysing social choice procedures is a lively and highly active research area.

The long-term aim of our work is to develop formal tools to assist in the analysis and design of social choice procedures. In particular, we hope to develop techniques that will permit the *automated* analysis of social choice procedures. To this end, we aim to develop logics that allow us to formally express the properties of social choice procedures, such that these languages may be automatically processed. Our view is that logic can provide a powerful tool for the analysis of social choice procedures (Pauly, 2001; Wooldridge et al., 2007). Such logics can be used as *query languages* for social choice procedures: given some property P of a social choice procedure, we aim to be able to encode the property P as an expression ρ^P of our language, which we then pose as a query to an automated analysis system. Working towards the long-term goal, the present paper presents a logic for reasoning about social choice procedures, and in particular, for analysing strategy proofness.

The remainder of the paper is structured as follows. In Section 2 we recall the main concepts from game theory and social choice theory that we use throughout the paper. We then introduce our logic in Section 3. The logic is basically a modal logic (Chellas, 1980), which derives inspiration from the Coalition Logic of Propositional Control (CL-PC) (van der Hoek and Wooldridge, 2005). The latter logic includes operators to capture strategic ability. We extend this with operators for capturing agent preferences. The basic idea is to model an agent’s preferences via atomic propositions: a proposition $p_{x>y}^i$ will be used to represent the fact that agent i has reported that he prefers outcome x at least as much as outcome y . The strategic abilities of agents are captured using a CL-PC-like operator: an agent can choose any assignment of values for its preference variables that corresponds to a preference ordering. After presenting the syntax and semantics of the logic, we show how the logic can be used to characterise social choice functions, and show that the logic is expressively complete with respect to social choice functions, i.e., that every social choice function can be characterised as a formula of the logic. We give a complete axiomatization for the logic. To demonstrate the value of the logic, in Section 4 we formalise some properties of social choice functions and in particular, we show how it can be applied to

the problem of determining whether a social choice function is strategy-proof. We conclude in Section 5.

2. Background

In this section, we present the basic definitions of game theory and social choice upon which we construct our framework (Dasgupta et al., 1979; Osborne and Rubinstein, 1994).

We begin with some notation. We assume that game forms and social choice functions (to be defined hereafter) share the same domains of agents and outcomes. We denote by $N = \{1, \dots, n\}$ the finite set of agents (or players) and by K the finite set of social outcomes (outcomes hereafter). We use the letters a, b, c, \dots as constants of K . We use variables i, j, \dots to denote agents, and outcomes will be denoted by the variables x, y, z, \dots . Typically, one can consider that the agents are the voters and the outcomes are the candidates in some election.

We denote by $L(K)$ the set of *linear* orders over K . (A linear order here is a relation that is reflexive, transitive, antisymmetric and total.) By using a linear order, we are assuming the players cannot be indifferent between two *distinct* outcomes. A *preference relation* is a linear order of outcomes. Given K and N , a *preference profile* $<$ is a tuple $(<_i)_{i \in N}$ of preferences, where $<_i \in L(K)$ for every i . The set of preference profiles is denoted by $L(K)^N$. Note that we use the symbol $<_i$ for a preference relation for agents, which in this case happens to be reflexive (and we do not write \leq_i for it). Also, we will use the symbol $>_i$ with the obvious meaning, i.e., $y >_i x$ iff $x <_i y$.

DEFINITION 1. (Social choice function). *Given K and N , a social choice function (SCF) is a single-valued mapping from the set $L(K)^N$ of preference profiles into the set K of outcomes.*

For every preference profile, a social choice function describes the desirable outcome (from the point of view of the designer).

DEFINITION 2. (Strategic game form). *Given the sets N and K , a strategic game form is a tuple $\langle N, (A_i), K, o \rangle$ where:*

- A_i is a finite nonempty set of actions (or strategies) for each player $i \in N$;*
- $o : \times_{i \in N} A_i \rightarrow K$ assigns an outcome for every combination of actions.*

A strategic game form is sometimes called a *mechanism*. It specifies the agents taking part in the game, their available actions, and what outcome results from each combination of actions. We refer to a collection $(a_i)_{i \in N}$, consisting of one action for every agent in N , as an *action profile*. Given an action profile a , we denote by a_i the action of the player i .

REMARK 1. *There is a direct link between strategic game forms and social choice functions. Any social choice function can be viewed as a game form in which the set of actions of every agent is $L(K)$ (think of this as the preference profiles the agent can claim to be his), and the function o represents the social choice function (see (Moulin, 1983)). For any SCF F , we denote its associated game form by g^F .*

A *strategic game* is basically the composition of a strategic game form with a collection of preference relations (one for every agent) over the set of outcomes.

DEFINITION 3. (Strategic game). *A strategic game is a tuple $\langle N, (A_i), K, o, (\prec_i) \rangle$ where $\langle N, (A_i), K, o \rangle$ is a strategic game form, and for each player $i \in N$, \prec_i is a preference relation over K .*

In our context, when the actions A_i in a game $\langle N, (A_i), K, o, (\prec_i) \rangle$ are preference relations themselves, one should think of those as preferences that i can choose to report, whereas \prec_i encodes i 's real preferences.

A solution concept defines for every game a set of action profiles – intuitively, those that may be played through rational action. Exactly which solution concept is used depends upon the application at hand: we will soon introduce a well-celebrated solution concept of *Nash Equilibrium* (see Example 1).

DEFINITION 4. (Solution concept). *A solution concept SC is a function that maps a strategic game form $\langle N, (A_i), K, o \rangle$ and a preference profile over K to a subset of the action profiles.*

We now introduce a simple but fundamentally important solution concept: Nash equilibrium.

DEFINITION 5. (Nash equilibrium). *Given a strategic game form $g = \langle N, (A_i), K, o \rangle$ and a preference profile \prec over K the set of Nash equilibria $NE(g, \prec)$ is given as the set of action profiles in g such that no player would benefit from deviating unilaterally from his current action. More formally, $(a_1, \dots, a_n) \in NE(g, \prec)$ iff for every player k and every $a'_k \in A_k$, we have $o(a_1, \dots, a'_k, \dots, a_n) \prec_k o(a_1, \dots, a_k, \dots, a_n)$.*

We can now introduce the notions of implementation and truthful implementation. The problem of implementation arises because a planner does not know the true preference profile of the players. Given a social choice function F involving a set of players N and a set of outcomes K , the planner only knows that every player $i \in N$ has some preference \prec_i , an element of $L(K)$.

We first define the case of (standard) *implementation*. Assuming a pattern of behaviour – a solution concept SC – the role of the planner is then to design

a mechanism (or game form) g such that for every possible preference profile $\langle \in L(K)^N$, the strategic game $\langle g, \langle \rangle$ admits at least one SC -equilibrium, and every SC -equilibrium leads to the outcome in K which is prescribed by the social choice function for the preference profile at hand, that is, the value of $F(\langle \rangle)$.

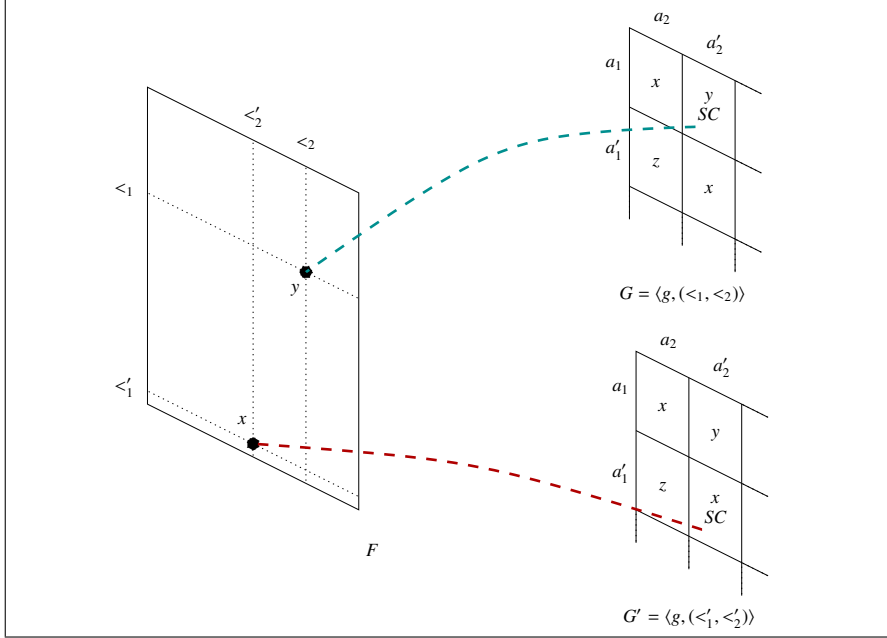


Figure 1. Implementation. The preference profiles \langle and \langle' are two arbitrary members of $L(K)^N$. The left part represents the SCF F . $F(\langle_1, \langle_2) = y$ and $F(\langle'_1, \langle'_2) = x$. The right part represents the strategic game form g instantiated, in the upper part with the preference profile (\langle_1, \langle_2) (game $G = \langle g, (\langle_1, \langle_2) \rangle$) and in the lower part with the preference profile (\langle'_1, \langle'_2) (game $G' = \langle g, (\langle'_1, \langle'_2) \rangle$). All the SC -equilibria of G (and possibly also some others than (a'_1, a'_2)) lead to $F(\langle_1, \langle_2)$. In a like manner, all the SC -equilibria of G' lead to $F(\langle'_1, \langle'_2)$. This has to be verified for every preference profile in $L(K)^N$ and not only \langle and \langle' : if it holds, g is said to SC -implement F .

DEFINITION 6. (Implementation). *Given a solution concept SC , we say that the game form $g = \langle N, (A_i), K, o \rangle$ SC -implements the social choice function F if for every preference profile $\langle \in L(K)^N$ we have that $SC(g, \langle) \neq \emptyset$ and*

$$a^* \in SC(g, \langle) \text{ implies that } o(a^*) = F(\langle)$$

In words: the game form g SC -implements F if for any game form $\langle g, \langle \rangle$ based on g , any outcome associated to a strategy profile in the solution concept SC is the same as what the social choice function would yield for the

preference \prec . Or, more loosely: the game form g implements F if, for every preference profile \prec that we can associate with it, the outcomes in the game $\langle g, \prec \rangle$ and the result of $F(\prec)$ agree at least on those claimed preferences that are in the solution concept of the game.

The problem of implementation is illustrated in Figure 1. We say that the social choice function is *SC-implementable* if there is a game form that *SC-implements* it.

In some situations however, an SCF can be implemented by a strategic game form of which the space of action profiles corresponds to the space of preference profiles, and *telling the truth is an equilibrium*. We call a strategic game form in which the set of strategies of a player i is the set of preferences over K a *direct mechanism*. Hence, each player is asked to report a preference, but not necessarily the true one. An appealing class of direct mechanisms is that in which reporting the true preference profile is an equilibrium of the game consisting of the direct mechanism composed with the true preference profile. That is, for every $\prec \in L(K)^N$, the action profile where every player i reports its true preference \prec_i is an equilibrium of the game $\langle g, \prec \rangle$. We can define this notion for every solution concept *SC*.

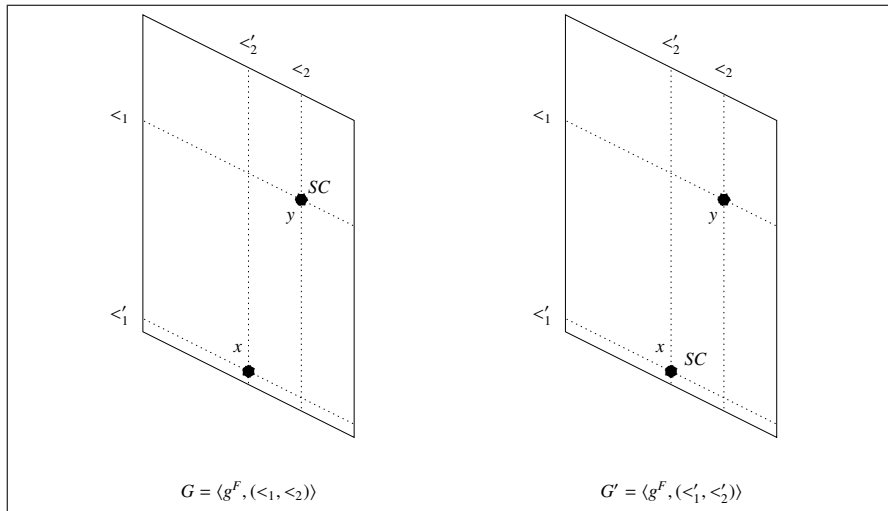


Figure 2. Truthful implementation. The preference profiles \prec and \prec' are two arbitrary members of $L(K)^N$. The left part represents the game form g^F associated to the SCF F when the preferences of the two players are \prec_1 and \prec_2 . The game $G = \langle g^F, (\prec_1, \prec_2) \rangle$ admits an equilibrium at the action profile (\prec_1, \prec_2) . The right part represents g^F when the preferences of the two players are \prec'_1 and \prec'_2 . The game $G' = \langle g^F, (\prec'_1, \prec'_2) \rangle$ admits an equilibrium at the action profile (\prec'_1, \prec'_2) . This has to be verified for every preference profile in $L(K)^N$ and not only \prec and \prec' : if it holds, g^F is said to truthfully *SC-implement* F .

DEFINITION 7. (Truthful implementation). *The direct mechanism $g = \langle N, (A_i), K, o \rangle$ truthfully SC-implements the SCF F if for every true preference profile \prec and reported profile a^* with $a_i^* = \prec_i$ for every i :*

$$a^* \in SC(g, \prec), \text{ and } o(a^*) = F(\prec)$$

In words: g is a truthful SC-implementation of F if, for every profile \prec , whenever the agents declare that to be their real preferences, this a solution concept SC, and the outcome in the game and the function F are the same. The problem of truthful implementation is illustrated on Figure 2. We say that the social choice function is *truthfully SC-implementable* if there is a game form that truthfully SC-implements it. Note that truthful implementations only require that the report of the true preference profile is an equilibrium, but it is not required that this equilibrium is unique. In general, other equilibria could be present that would not lead to the outcome prescribed by the SCF. However, this notion of implementation can be motivated. Indeed, it is assumed that playing a direct mechanism, if casting the real preference is an equilibrium strategy, an agent would be sincere.

We illustrate the differences between the problems of implementation with some simple examples (a ‘minimal’ social choice scenario with only two voters and two alternatives), which demonstrates that the two notions are contingent and independent: a game form g can be both a truthful SC-implementation and an SC-implementation of a social function F , it can be both, and it can be either of them without being the other.

EXAMPLE 1. *In this example we define some simple social choice functions, for all of them we set $N = \{1, 2\}$ and $K = \{a, b\}$. Also, for the sake of comparison between standard and truthful implementations, we only consider direct mechanisms, since truthful implementations are not defined otherwise.*

First, consider the function H for which we claim that its associated game form g^H truthfully NE-implements H but g^H does not NE-implement it. H is the social choice function prescribing the outcome b if and only both agents prefer b over a . We write $[a, b]$ for the individual order of preferences of the outcome a over the outcome b and $[b, a]$ for the individual preference of b over a . Hence, we have:

$$\begin{aligned} H([a, b], [a, b]) &= H([a, b], [b, a]) = H([b, a], [a, b]) = a; \\ H([b, a], [b, a]) &= b. \end{aligned}$$

Figure 3 represents the four possible games $\langle g^H, \prec \rangle$ where $\prec \in L(\{a, b\})^{\{1,2\}}$. In each of them, the circles indicate the action profiles that are Nash equilibria. The outcomes in bold are the outcomes $o(a^)$ for which $a^* = \prec$: in those outcomes, players have revealed their true preferences. So for instance, the outcome **a** in the upper left corner of the game $\langle g^H, ([a, b], [a, b]) \rangle$ reads:*

<table border="1" style="margin: auto;"> <thead> <tr> <th style="padding: 2px;"></th> <th style="padding: 2px;">[a, b]</th> <th style="padding: 2px;">[b, a]</th> </tr> </thead> <tbody> <tr> <th style="padding: 2px;">[a, b]</th> <td style="text-align: center; padding: 2px;">a ✓</td> <td style="text-align: center; padding: 2px;">a</td> </tr> <tr> <th style="padding: 2px;">[b, a]</th> <td style="text-align: center; padding: 2px;">a</td> <td style="text-align: center; padding: 2px;">b</td> </tr> </tbody> </table> <p style="text-align: center;">$\langle g^H, ([a, b], [a, b]) \rangle$</p>		[a, b]	[b, a]	[a, b]	a ✓	a	[b, a]	a	b	<table border="1" style="margin: auto;"> <thead> <tr> <th style="padding: 2px;"></th> <th style="padding: 2px;">[a, b]</th> <th style="padding: 2px;">[b, a]</th> </tr> </thead> <tbody> <tr> <th style="padding: 2px;">[a, b]</th> <td style="text-align: center; padding: 2px;">a</td> <td style="text-align: center; padding: 2px;">a ✓</td> </tr> <tr> <th style="padding: 2px;">[b, a]</th> <td style="text-align: center; padding: 2px;">a</td> <td style="text-align: center; padding: 2px;">b</td> </tr> </tbody> </table> <p style="text-align: center;">$\langle g^H, ([a, b], [b, a]) \rangle$</p>		[a, b]	[b, a]	[a, b]	a	a ✓	[b, a]	a	b
	[a, b]	[b, a]																	
[a, b]	a ✓	a																	
[b, a]	a	b																	
	[a, b]	[b, a]																	
[a, b]	a	a ✓																	
[b, a]	a	b																	
<table border="1" style="margin: auto;"> <thead> <tr> <th style="padding: 2px;"></th> <th style="padding: 2px;">[a, b]</th> <th style="padding: 2px;">[b, a]</th> </tr> </thead> <tbody> <tr> <th style="padding: 2px;">[a, b]</th> <td style="text-align: center; padding: 2px;">a</td> <td style="text-align: center; padding: 2px;">a</td> </tr> <tr> <th style="padding: 2px;">[b, a]</th> <td style="text-align: center; padding: 2px;">a ✓</td> <td style="text-align: center; padding: 2px;">b</td> </tr> </tbody> </table> <p style="text-align: center;">$\langle g^H, ([b, a], [a, b]) \rangle$</p>		[a, b]	[b, a]	[a, b]	a	a	[b, a]	a ✓	b	<table border="1" style="margin: auto;"> <thead> <tr> <th style="padding: 2px;"></th> <th style="padding: 2px;">[a, b]</th> <th style="padding: 2px;">[b, a]</th> </tr> </thead> <tbody> <tr> <th style="padding: 2px;">[a, b]</th> <td style="text-align: center; padding: 2px;">a +</td> <td style="text-align: center; padding: 2px;">a</td> </tr> <tr> <th style="padding: 2px;">[b, a]</th> <td style="text-align: center; padding: 2px;">a</td> <td style="text-align: center; padding: 2px;">b ✓</td> </tr> </tbody> </table> <p style="text-align: center;">$\langle g^H, ([b, a], [b, a]) \rangle$</p>		[a, b]	[b, a]	[a, b]	a +	a	[b, a]	a	b ✓
	[a, b]	[b, a]																	
[a, b]	a	a																	
[b, a]	a ✓	b																	
	[a, b]	[b, a]																	
[a, b]	a +	a																	
[b, a]	a	b ✓																	

Figure 3. g^H does not NE-implement H . But g^H truthfully NE-implements H .

‘the outcome in the game here is a and the voters reveal their true preferences’. For every preference profile $<$, the ticks \checkmark indicate that the action profile $<$ leads to the outcome prescribed by the social choice function H and is a Nash equilibrium in the game $\langle g^H, < \rangle$; Hence the game form g^H truthfully NE-implements H : all the bold outcomes are ticked. The cross $+$ designates a problem with the (standard) implementation of H by g^H : in the game $\langle g^H, ([b, a], [b, a]) \rangle$ the action profile $([a, b], [a, b])$ is a Nash equilibrium and leads to the outcome a , however $H([a, b], [a, b]) = b$. Hence, g^H does not NE-implement H .

Let us next consider the social choice function J which is dictatorial for player 1, i.e., J is defined by

$$\begin{aligned} J([a, b], [a, b]) &= J([a, b], [b, a]) = a; \\ J([b, a], [a, b]) &= J([b, a], [b, a]) = b. \end{aligned}$$

The four possible games $\langle g^J, < \rangle$ for J are depicted in Figure 4. It is easy to see that the circled outcomes in those games are Nash equilibria: they give the preferred outcome for 1 (so 1 cannot improve by deviating) and they are the same in a fixed row (so 2 cannot change the outcome). Moreover, it is also a straightforward check that for all those Nash equilibria, the outcome in the game $\langle g^J, < \rangle$ is the same as $J(<)$ (for instance, in the top left game, both equilibria yield a which coincides with $J([a, b], [a, b])$, and in the bottom left game, both equilibria yield $b = J([b, a], [a, b])$, etc): this justifies the ticks \checkmark .

<table style="margin: auto;"> <thead> <tr> <th style="padding: 5px;"></th> <th style="padding: 5px;">[a, b]</th> <th style="padding: 5px;">[b, a]</th> </tr> </thead> <tbody> <tr> <th style="padding: 5px;">[a, b]</th> <td style="text-align: center; padding: 5px;">Ⓐ ✓</td> <td style="text-align: center; padding: 5px;">Ⓐ ✓</td> </tr> <tr> <th style="padding: 5px;">[b, a]</th> <td style="text-align: center; padding: 5px;">b</td> <td style="text-align: center; padding: 5px;">b</td> </tr> </tbody> </table> <p style="text-align: center;">$\langle g^J, ([a, b], [a, b]) \rangle$</p>		[a, b]	[b, a]	[a, b]	Ⓐ ✓	Ⓐ ✓	[b, a]	b	b	<table style="margin: auto;"> <thead> <tr> <th style="padding: 5px;"></th> <th style="padding: 5px;">[a, b]</th> <th style="padding: 5px;">[b, a]</th> </tr> </thead> <tbody> <tr> <th style="padding: 5px;">[a, b]</th> <td style="text-align: center; padding: 5px;">Ⓐ ✓</td> <td style="text-align: center; padding: 5px;">Ⓐ ✓</td> </tr> <tr> <th style="padding: 5px;">[b, a]</th> <td style="text-align: center; padding: 5px;">b</td> <td style="text-align: center; padding: 5px;">b</td> </tr> </tbody> </table> <p style="text-align: center;">$\langle g^J, ([a, b], [b, a]) \rangle$</p>		[a, b]	[b, a]	[a, b]	Ⓐ ✓	Ⓐ ✓	[b, a]	b	b
	[a, b]	[b, a]																	
[a, b]	Ⓐ ✓	Ⓐ ✓																	
[b, a]	b	b																	
	[a, b]	[b, a]																	
[a, b]	Ⓐ ✓	Ⓐ ✓																	
[b, a]	b	b																	
<table style="margin: auto;"> <thead> <tr> <th style="padding: 5px;"></th> <th style="padding: 5px;">[a, b]</th> <th style="padding: 5px;">[b, a]</th> </tr> </thead> <tbody> <tr> <th style="padding: 5px;">[a, b]</th> <td style="text-align: center; padding: 5px;">a</td> <td style="text-align: center; padding: 5px;">a</td> </tr> <tr> <th style="padding: 5px;">[b, a]</th> <td style="text-align: center; padding: 5px;">Ⓑ ✓</td> <td style="text-align: center; padding: 5px;">Ⓑ ✓</td> </tr> </tbody> </table> <p style="text-align: center;">$\langle g^J, ([b, a], [a, b]) \rangle$</p>		[a, b]	[b, a]	[a, b]	a	a	[b, a]	Ⓑ ✓	Ⓑ ✓	<table style="margin: auto;"> <thead> <tr> <th style="padding: 5px;"></th> <th style="padding: 5px;">[a, b]</th> <th style="padding: 5px;">[b, a]</th> </tr> </thead> <tbody> <tr> <th style="padding: 5px;">[a, b]</th> <td style="text-align: center; padding: 5px;">a</td> <td style="text-align: center; padding: 5px;">a</td> </tr> <tr> <th style="padding: 5px;">[b, a]</th> <td style="text-align: center; padding: 5px;">Ⓑ ✓</td> <td style="text-align: center; padding: 5px;">Ⓑ ✓</td> </tr> </tbody> </table> <p style="text-align: center;">$\langle g^J, ([b, a], [b, a]) \rangle$</p>		[a, b]	[b, a]	[a, b]	a	a	[b, a]	Ⓑ ✓	Ⓑ ✓
	[a, b]	[b, a]																	
[a, b]	a	a																	
[b, a]	Ⓑ ✓	Ⓑ ✓																	
	[a, b]	[b, a]																	
[a, b]	a	a																	
[b, a]	Ⓑ ✓	Ⓑ ✓																	

Figure 4. g^J both NE-implements and truthfully NE-implements J .

So g NE-implements J . To show that g also truthfully NE-implements J , we need to check that all the bold outcomes in Figure 4 are circled and ticked ✓.

Next, to give an example of an NE-implementation that is not a truthful one, consider the game form g_-^J . It is mathematically equivalent to the game form g^J : the outcomes a and b are only inverted. Playing g_-^J , the player 1 would simply play the contrary to his true preference. This always yields a Nash equilibrium and the outcomes are always as prescribed by J . Hence, like g^J , the game form g_-^J is an NE-implementation of the social choice function J . However, since the player 1 needs to trick the game in order to achieve a Nash equilibrium, it is easy to see that g_-^J does not truthfully NE-implement J . The crosses † on Figure 5 mark the action profiles that correspond to the true preferences of the players, and we can see that their respective outcome always fails to be as prescribed by J .

Finally, we argue that it is possible for a game form to be neither a NE-implementation nor a truthful implementation of a given function: take P such that $P(<) = a$ for all profiles $<$. Moreover, for all $<$, let all outcomes in the matrix for $\langle g^P, < \rangle$ be b . For every $<$, every outcome in $\langle g^M, < \rangle$ is a Nash equilibrium (no agent can change the outcome, let alone improve it). At the same time, for all a^* we have $b = o(a^*) \neq P(<) = a$, which shows that g does not NE-implement P . It does also not truthfully NE-implement it: take, for any $<$, a profile a^* in the game $\langle g^M, < \rangle$ such that $a^* = <$. We have already seen that $o(a^*) \neq P(<)$, which proves our claim.

<table style="margin: auto;"> <tr> <td></td> <td style="padding: 2px;">$[a, b]$</td> <td style="padding: 2px;">$[b, a]$</td> </tr> <tr> <td style="padding: 2px;">$[a, b]$</td> <td style="padding: 2px; text-align: center;">b +</td> <td style="padding: 2px; text-align: center;">b</td> </tr> <tr> <td style="padding: 2px;">$[b, a]$</td> <td style="padding: 2px; text-align: center;">\textcircled{a} ✓</td> <td style="padding: 2px; text-align: center;">\textcircled{a} ✓</td> </tr> </table> <p style="text-align: center;">$\langle g^J, ([a, b], [a, b]) \rangle$</p>		$[a, b]$	$[b, a]$	$[a, b]$	b +	b	$[b, a]$	\textcircled{a} ✓	\textcircled{a} ✓	<table style="margin: auto;"> <tr> <td></td> <td style="padding: 2px;">$[a, b]$</td> <td style="padding: 2px;">$[b, a]$</td> </tr> <tr> <td style="padding: 2px;">$[a, b]$</td> <td style="padding: 2px; text-align: center;">b</td> <td style="padding: 2px; text-align: center;">b +</td> </tr> <tr> <td style="padding: 2px;">$[b, a]$</td> <td style="padding: 2px; text-align: center;">\textcircled{a} ✓</td> <td style="padding: 2px; text-align: center;">\textcircled{a} ✓</td> </tr> </table> <p style="text-align: center;">$\langle g^J, ([a, b], [b, a]) \rangle$</p>		$[a, b]$	$[b, a]$	$[a, b]$	b	b +	$[b, a]$	\textcircled{a} ✓	\textcircled{a} ✓
	$[a, b]$	$[b, a]$																	
$[a, b]$	b +	b																	
$[b, a]$	\textcircled{a} ✓	\textcircled{a} ✓																	
	$[a, b]$	$[b, a]$																	
$[a, b]$	b	b +																	
$[b, a]$	\textcircled{a} ✓	\textcircled{a} ✓																	
<table style="margin: auto;"> <tr> <td></td> <td style="padding: 2px;">$[a, b]$</td> <td style="padding: 2px;">$[b, a]$</td> </tr> <tr> <td style="padding: 2px;">$[a, b]$</td> <td style="padding: 2px; text-align: center;">\textcircled{b} ✓</td> <td style="padding: 2px; text-align: center;">\textcircled{b} ✓</td> </tr> <tr> <td style="padding: 2px;">$[b, a]$</td> <td style="padding: 2px; text-align: center;">a +</td> <td style="padding: 2px; text-align: center;">a</td> </tr> </table> <p style="text-align: center;">$\langle g^J, ([b, a], [a, b]) \rangle$</p>		$[a, b]$	$[b, a]$	$[a, b]$	\textcircled{b} ✓	\textcircled{b} ✓	$[b, a]$	a +	a	<table style="margin: auto;"> <tr> <td></td> <td style="padding: 2px;">$[a, b]$</td> <td style="padding: 2px;">$[b, a]$</td> </tr> <tr> <td style="padding: 2px;">$[a, b]$</td> <td style="padding: 2px; text-align: center;">\textcircled{b} ✓</td> <td style="padding: 2px; text-align: center;">\textcircled{b} ✓</td> </tr> <tr> <td style="padding: 2px;">$[b, a]$</td> <td style="padding: 2px; text-align: center;">a</td> <td style="padding: 2px; text-align: center;">a +</td> </tr> </table> <p style="text-align: center;">$\langle g^J, ([b, a], [b, a]) \rangle$</p>		$[a, b]$	$[b, a]$	$[a, b]$	\textcircled{b} ✓	\textcircled{b} ✓	$[b, a]$	a	a +
	$[a, b]$	$[b, a]$																	
$[a, b]$	\textcircled{b} ✓	\textcircled{b} ✓																	
$[b, a]$	a +	a																	
	$[a, b]$	$[b, a]$																	
$[a, b]$	\textcircled{b} ✓	\textcircled{b} ✓																	
$[b, a]$	a	a +																	

Figure 5. g^J NE-implements J but does not truthfully NE-implement it.

3. A Logic of social choice functions

Following the tradition in implementation theory (cf. Remark 1), we model social choice functions as a particular kind of strategic game form. In (Troquard et al., 2009a) we proposed a logic for modelling strategic games on the basis of CL-PC. Every player controls a set of propositional variables and a strategy for a player amounts to choosing a truth value for the variables he controls. We adapt the ideas of (Troquard et al., 2009a) to game forms where the strategies of the players correspond to the reports of preferences.

3.1. SEMANTICS

Let X be an arbitrary set of propositions. We can see a *valuation* of X as a subset $V \subseteq X$ where tt (i.e., true) is assigned to the propositions in V and ff (false) is assigned to the propositions in $X \setminus V$. We denote the set of possible valuations over X by Θ^X .

In the presence of a set of players N and a set of outcomes K , the set of propositions controlled by a player $i \in N$ is defined as $At[i, K] = \{p_{x>y}^i \mid x, y \in K\}$. Every $p_{x>y}^i$ is a proposition controlled by the agent i which means that i reports that it values the outcome x at least as good as y . We also define $At[N, K] = \cup_{i \in N} At[i, K]$, which is then the set of all controlled propositions.

We can ‘encode’ a particular preference (or linear order) of player i as a valuation of the propositions in $At[i, K]$. However, conversely, not all valua-

tions correspond to a linear order preference. A strategy of a player i consists of reporting a valuation of $At[i, K]$ encoding a linear order over K . For every player i , we define $strategies[i, K]$ as a set of valuations $V \in \Theta^{At[i, K]}$ such that: (i) $p_{x>x}^i \in V$, (ii) if $x \neq y$ then $p_{x>y}^i \in V$ iff $p_{y>x}^i \notin V$, and (iii) if $p_{x>y}^i \in V$ and $p_{y>z}^i \in V$ then $p_{x>z}^i \in V$.

REMARK 2. Every $p_{x>y}^i$ could be seen as a predicative expression $p(i, x, y)$ that would read that agent i reported to prefer the outcome x over y . However, since N and K are finite, we look at these expressions as a finite collection of propositions. The constraints of control in Figure 6 will be their propositional theory corresponding to the three preceding constraints on the valuations.

For every coalition $C \subseteq N$, let $strategies[C, K]$ be the set of tuples $v_C = (v_i)_{i \in C}$ where $v_i \in strategies[i, K]$. It is the set of strategies of the coalition C . To put it another way, it corresponds to a valuation of the propositions controlled by the players in C , encoding one preference over K for every player in C .

A *state* (or reported preference profile) is an element of $strategies[N, K]$, that is, a strategy of the coalition containing all the players. We now define the models of social choice functions.

DEFINITION 8. (Model of social choice functions). A model of social choice functions over N and K is a tuple $M = \langle N, K, out, (<_i) \rangle$, such that:

$out : strategies[N, K] \rightarrow K$ maps every state to an outcome;
For every $i \in N$, $<_i \in L(K)$ is the true order of preferences of i .

Hence, every player i has two levels of preferences: (1) a true one, given by $<_i$ and (2) a reported one, given by a valuation in $strategies[i, K]$.

Taking out the true preference profile from a model of SCF, we obtain a mere instantiation of a pre-Boolean game (Bonzon et al., 2007). It is required to assign every variable to one (actual control) and only one (exclusive control) player, but there are some constraints on the possible valuations ('non-full' control). In (Bonzon et al., 2007), actual and exclusive control are grasped by an assignment function (mapping every propositional variable to exactly one player), and the partial control is modelled by a *set of constraints* given as a set of satisfiable propositional formulae.

The language $\mathcal{L}^{scf}[N, K]$ is inductively defined by the following grammar:

$$\varphi ::= \top \mid p \mid x \mid \neg\varphi \mid \varphi \vee \varphi \mid \diamond_C \varphi \mid \blacklozenge_i \varphi$$

where p is atom of $At[N, K]$, x is an atom of K , $i \in N$, and C is a coalition. Given a model M and a state (i.e., a reported profile v), formula $\diamond_C \varphi$ reads that provided that the players outside C hold on to their current strategy v_C , the coalition C has a strategy, i.e., a way to announce their profiles, such that φ holds. Formula $\blacklozenge_i \varphi$ reads that i locally (at the current reported profile) considers a reported profile where φ is true at least as preferable.

DEFINITION 9. (Truth values of $\mathcal{L}^{scf}[N, K]$). *Given a model $M = \langle N, K, out, (<_i) \rangle$, we are going to interpret formulae of $\mathcal{L}^{scf}[N, K]$ in a state of the model. A state $v = (v_1, \dots, v_n)$ in M is a tuple of valuations $v_i \in strategies[i, K]$, one for each agent. The truth definition is inductively given by:*

$$\begin{aligned}
M, v \models p & \quad \text{iff } p \in v_i \text{ for some } i \in N \\
M, v \models x & \quad \text{iff } out(v) = x \\
M, v \models \neg\varphi & \quad \text{iff } M, v \not\models \varphi \\
M, v \models \varphi \vee \psi & \quad \text{iff } M, v \models \varphi \text{ or } M, v \models \psi \\
M, v \models \diamond_C \varphi & \quad \text{iff } \text{there is a state } u \text{ such that} \\
& \quad v_i = u_i \text{ for every } i \notin C \text{ and } M, u \models \varphi \\
M, v \models \blacklozenge_i \varphi & \quad \text{iff } \text{there is a state } u \text{ such that} \\
& \quad out(v) <_i out(u) \text{ and } M, u \models \varphi
\end{aligned}$$

We assume that player i only makes claims or announcements about its own preferences, and i controls nothing else, so the atomic clause could equivalently have read

$$M, v \models p_{x>y}^i \text{ iff } p_{x>y}^i \in v_i$$

The truth of φ in all models over a set of players N and a set of outcomes K is denoted by $\models_{\Lambda^{scf}[N, K]} \varphi$. The classical operators $\wedge, \rightarrow, \leftrightarrow$ can be defined as usual. We also define $\square_C \varphi \triangleq \neg \diamond_C \neg \varphi$ and $\blacksquare_i \varphi \triangleq \neg \blacklozenge_i \neg \varphi$.

THEOREM 1. (Decidability). *The problem of deciding whether a formula $\varphi \in \mathcal{L}^{scf}[N, K]$ is satisfiable is decidable.*

Proof. It suffices to remark that N and K are finite. Hence, we can enumerate every model of SCF over N and K and check whether φ is satisfiable in one state of one model. ■

3.2. BALLOTS

We think of a particular preference of $L(K)$ encoded in the language of the propositions as a *ballot*.

DEFINITION 10. (Ballot). *For every player $i \in N$, we can see every $<_i \in L(K)$ as a permutation $[x_1, x_2, \dots]$ of the elements of K , where the more to the left the outcome is, the more it is preferred by the player i . We can reify in the language the reported preferences, that is, the ballot casted by the player i :*

$$\mathbf{ballot}_i(<) \triangleq p_{x_1>x_2}^i \wedge p_{x_2>x_3}^i \wedge \dots \wedge p_{x_{|K|-1}>x_{|K|}}^i.$$

Then, the formula

$$\mathbf{ballot}(\prec) \triangleq \bigwedge_{i \in N} \mathbf{ballot}_i(\prec)$$

is a reification of the reported preference profile $\prec = (\prec_1, \dots, \prec_n)$, consisting of one ballot for every player $i \in N$.

REMARK 3. Note that for every $\prec \in L(K)$, the formula $\mathbf{ballot}(\prec)$ is true at one and only one state. The reader familiar with Hybrid Logic (Areces and ten Cate, 2006) may think of the formula $\mathbf{ballot}(\prec)$ as a nominal, viz. a state label available in the object language.

EXAMPLE 2. Suppose that $N = \{1, 2\}$ and $K = \{a, b, c\}$. Let a preference profile $(\prec_1^{ex}, \prec_2^{ex}) \in L(K)^N$ given by the data of the two permutations $[a, c, b]$ and $[c, a, b]$ representing respectively the preferences of player 1 and 2. This reported preference profile can be represented in the language $\mathcal{L}^{scf}[\{1, 2\}, \{a, b, c\}]$ by the formula

$$\mathbf{ballot}(\prec^{ex}) \triangleq p_{a>c}^1 \wedge p_{c>b}^1 \wedge p_{c>a}^2 \wedge p_{a>b}^2.$$

It is easy to verify that the constraints on the elements of strategies $[1, K]$ and strategies $[2, K]$ are sufficient for inferring a complete characterisation of the preference profile. The following is valid in the models of social choice functions over $\{1, 2\}$ and $\{a, b, c\}$:

$$\mathbf{ballot}(\prec^{ex}) \leftrightarrow p_{a>a}^1 \wedge p_{b>b}^1 \wedge p_{c>c}^1 \wedge p_{a>c}^1 \wedge p_{c>b}^1 \wedge p_{a>b}^1 \wedge \neg p_{c>a}^1 \wedge \neg p_{b>c}^1 \wedge \neg p_{b>a}^1 \wedge p_{a>a}^2 \wedge p_{b>b}^2 \wedge p_{c>c}^2 \wedge p_{c>a}^2 \wedge p_{a>b}^2 \wedge p_{c>b}^2 \wedge \neg p_{a>c}^2 \wedge \neg p_{b>a}^2 \wedge \neg p_{b>c}^2$$

3.3. CHARACTERISING AN SCF

Recall that a model of social choice functions is a tuple $M = \langle N, K, out, (\prec_i) \rangle$, where \prec_i are the real preferences of the agents and the outcome function o assigns to every valuation an element of K . There is a one-one correspondence between valuations and preference profiles: the preference profile $P(v)$ associated with valuation v is the relation \prec for which $x \succ_i y$ iff $p_{x>y}^i \in v$. Likewise, the valuation $V(\prec)$ associated with \prec is the set $\{p_{x>y}^i \mid x \succ_i y\}$, which collect all the atoms from $\mathbf{ballot}(\prec)$. This makes it possible to relate a model M with a social choice function F as follows.

We say that a model $M = \langle N, K, out, (\prec_i) \rangle$ and social choice function $F : L(K)^N \rightarrow K$ correspond, if for every strategy profile \prec and its associated valuation v (i.e., for which $V(\prec) = v$ and $P(v) = \prec$), we have $o(v) = F(\prec)$.

This correspondence can be syntactically defined in a formula ρ^F :

$$\rho^F = \bigwedge_{\langle \in L(K)^N} \diamond_N(\text{ballot}(\langle) \wedge F(\langle))$$

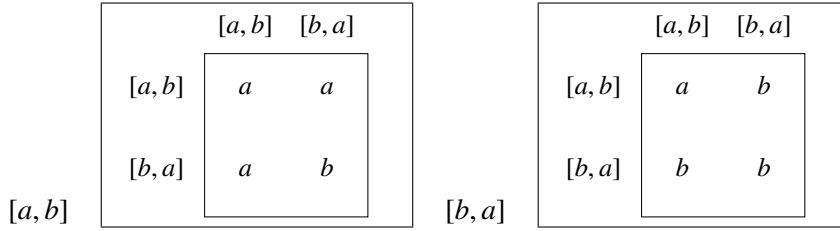
Note that \diamond_N plays the role of the universal/global existential modality often noted E in the literature in modal logic: it allows us to quantify over all the possible valuations in $\Theta^{\text{At}[N,K]}$, or ballots.

Given the outcomes K , the agents N and the social choice function F , formula ρ^F says that every profile \langle together with $F(\langle)$ as an outcome appears in the model. Since the states of a model are all possible profiles in $L(K)^N$, and every profile occurs exactly once, we might as well have defined ρ^F as

$$\rho^F = \bigwedge_{\langle \in L(K)^N} (\text{ballot}(\langle) \rightarrow F(\langle))$$

It is easy to see that the logic is expressively complete wrt. social choice functions. That is, for every SCF F over a set of players N and a set of outcomes K , there exists a formula $\rho^F \in \mathcal{L}^{\text{scf}}[N, K]$ characterising it. Even though it may not be optimal in terms of succinctness, it suffices to consider the conjuncts of formulae $\diamond_N(\text{ballot}(\langle) \wedge x)$, for $\langle \in L(K)$ and $F(\langle) = x$. The next example shows, using a simple scenario, that we can sometimes obtain less naïve and more compact characterisations.

EXAMPLE 3. *Consider the following model of SCF (or game form) where player 1 chooses rows, player 2 chooses columns and player 3 chooses matrices. There are two outcomes a and b . Hence, every player i controls the set of atoms $\{p_{a>a}^i, p_{b>b}^i, p_{a>b}^i, p_{b>a}^i\}$. Every player i has two strategies: $p_{a>a}^i \wedge p_{b>b}^i \wedge p_{a>b}^i \wedge \neg p_{b>a}^i$ and $p_{a>a}^i \wedge p_{b>b}^i \wedge \neg p_{a>b}^i \wedge p_{b>a}^i$, that we denote respectively by $[a, b]$ and $[b, a]$. (In the logic $\Lambda^{\text{scf}}[\{1, 2, 3\}, \{a, b\}]$, they are in fact equivalent to the formulae $p_{a>b}^i$ and $p_{b>a}^i$, respectively.)*



We can represent it in the logic $\Lambda^{\text{scf}}[\{1, 2, 3\}, \{a, b\}]$ of social choice functions by the formula:

$$\rho^F \triangleq a \leftrightarrow (p_{a>b}^1 \wedge p_{a>b}^2) \vee (p_{a>b}^1 \wedge p_{a>b}^3) \vee (p_{a>b}^2 \wedge p_{a>b}^3).$$

Note that since out is functional, in the models of social choice functions with $K = \{a, b\}$ the outcome b will hold whenever a does not.

Going back to the social choice functions of Example 1, we invite the reader to check that

$$\rho^H = b \leftrightarrow (p_{b>a}^1 \wedge p_{b>a}^2)$$

$$\rho^J = a \leftrightarrow p_{a>b}^1$$

$$\rho^P = a$$

3.4. TRUE PREFERENCES

In Section 3.2 we saw how to use the atoms in $At[i, K]$ to encode the reported preference or ballot of a player i . These atoms do not necessarily represent the true preferences of the agents. We handle the true preferences of player i via the \blacklozenge_i modality.

From our basic language $\mathcal{L}^{scf}[N, K]$, we can also define an operator of interest concerning preferences. We can define the *global binary* operator of preferences $\psi \blacktriangleleft_i \varphi$, corresponding to a preference between propositions. It reads “all φ are better than all ψ ”.

$$\psi \blacktriangleleft_i \varphi \triangleq \Box_N \bigvee_{< \in L(K)^N} (\text{ballot}(<) \wedge (\varphi \rightarrow \Box_N(\psi \rightarrow \blacklozenge_i \text{ballot}(<))).$$

Agent i judges the proposition φ at least as good as ψ iff when the reported preference profile is $<$ and φ holds at the state labeled by $\text{ballot}(<)$, then, whenever ψ holds in a state, i would prefer the state labeled by $\text{ballot}(<)$ (cf. Remark 3).

As in Definition 10 for reported preferences, we can now reify the true preferences. Provided that x and y are two possible outcomes, the formula $y \blacktriangleleft_i x$ captures the fact the player i prefers (globally) the outcome y over the outcome x . Hence, from a preference profile $< \in L(K)^N$, we reify the preference $[x_1, x_2 \dots]$ of the player i as follows:

$$\text{true}_i(<) \triangleq (x_{|K|} \blacktriangleleft_i x_{|K|-1}) \wedge \dots \wedge (x_3 \blacktriangleleft_i x_2) \wedge (x_2 \blacktriangleleft_i x_1).$$

Then, the formula

$$\text{true}(<) \triangleq \bigwedge_{i \in N} \text{true}_i(<)$$

is a reification of the true preference profile $\leq = (<_1, \dots, <_n)$.

REMARK 4. *Whenever in a model of social choice function M the true preference of a player i is such that $x <_i y$, then the formula $x \blacktriangleleft_i y$ is true at every state of M . However, the other way around does not hold. Indeed, when either x or y is not a possible outcome of a model, the formula $x \blacktriangleleft_i y$ is always*

true for every i . From the definition, $x \blacktriangleleft_i y \triangleq \Box_N \bigvee_{\langle \in L(K)^N} (\text{ballot}(\langle) \wedge (y \rightarrow \Box_N(x \rightarrow \blacklozenge_i \text{ballot}(\langle))))$. Hence, if y is not a possible outcome, the main implication $y \rightarrow \Box_N(x \rightarrow \blacklozenge_i \text{ballot}(\langle))$ is always true for y being always false. Likewise, if x is not a possible outcome, the implication $x \rightarrow \blacklozenge_i \text{ballot}(\langle)$ is always true for x being always false. In turn, it makes the main implication always true. Also, $\bigvee_{\langle \in L(K)^N} \text{ballot}(\langle)$ will always be satisfied since a state of evaluation represents a ballot by definition.

The object language does not allow to talk about true preferences on impossible outcomes. This observation will have a consequence in the way we prove the completeness of the logic.

3.5. AXIOMATICS

The axiomatization of the models of social choice functions is presented in Figure 6.

Constraints of control (*refl*), (*antisym-total*) and (*trans*) say that every player casts an appropriate valuation of its controlled atoms: a valuation must encode a linear order. (*comp* \cup) defines the local ability of coalitions in terms of local abilities of sub-coalitions. The transitivity of the operator \Box_C is the consequence of (*comp* \cup). Hence, together with (*T(i)*) and (*B(i)*), it makes of \Box_C an *S5* modality. (*empty*) means that the empty coalition has no power. (*comp* \cup) and (*conf* \cup) together make sure that the agents' choices are independent. (*exclu*) means that if an atom is controlled by a player i , the other players cannot change its value. (*ballot*) makes sure that an agent is always locally able to cast any preference. From (*comp-At*), provided that δ_1 and δ_2 do not contain a commonly controlled atom, if a coalition C_1 can locally enforce δ_1 and C_2 can locally enforce δ_2 then they can enforce $\delta_1 \wedge \delta_2$ together.

Axiom (*func1*) forces the fact that for every action profile there is one and only one outcome. (*func2*) ensures that the outcomes are only determined by the valuations. (*incl*) ensures that if something is settled, a player cannot prefer its negation. ($4(\leq_i)$) characterises transitivity. (*antisym'*) and (*total'*) force that the relation of preference over states is antisymmetric and total (and hence, in particular, this relation is reflexive). Finally, (*unifPref*) specifies a fundamental interaction between preferences and the outcomes. If the casted preference profile at hand leads to x and agent i prefers an action profile leading to y , then at every action profile leading to x , agent i will prefer every action profile leading to y , that is, all y are better than all x .

The logic has a clear flavour of normal modal logic (Chellas, 1980). The presence of (*K(i)*) with the necessitation rule (*Nec*(\Box_i)) gives to the operator \Box_i the property of normality. The necessitation rule for the operator \blacksquare_i holds because of (*Nec*(\Box_i)) and the axioms (*comp* \cup) and (*incl*). The normality of the modality \blacksquare_i then follows from (*K*(\langle_i)).

Constraints of control		
(<i>refl</i>)	$p_{x>x}^i$	
(<i>antisym-total</i>)	$p_{x>y}^i \leftrightarrow \neg p_{y>x}^i$, where $x \neq y$
(<i>trans</i>)	$p_{x>y}^i \wedge p_{y>z}^i \rightarrow p_{x>z}^i$	
Propositional control		
(<i>Prop</i>)	φ	, where φ is a propositional tautology
(<i>K(i)</i>)	$\Box_i(\varphi \rightarrow \psi) \rightarrow (\Box_i\varphi \rightarrow \Box_i\psi)$	
(<i>T(i)</i>)	$\Box_i\varphi \rightarrow \varphi$	
(<i>B(i)</i>)	$\varphi \rightarrow \Box_i\Diamond_i\varphi$	
(<i>comp\cup</i>)	$\Box_{C_1}\Box_{C_2}\varphi \leftrightarrow \Box_{C_1\cup C_2}\varphi$	
(<i>confl</i>)	$\Diamond_i\Box_j\varphi \rightarrow \Box_j\Diamond_i\varphi$	
(<i>empty</i>)	$\Box_{\emptyset}\varphi \leftrightarrow \varphi$	
(<i>exclu</i>)	$(\Diamond_i p \wedge \Diamond_i \neg p) \rightarrow (\Box_j p \vee \Box_j \neg p)$, where $j \neq i$
(<i>ballot</i>)	$\Diamond_i \text{ballot}_i(<)$	
(<i>comp-At</i>)	$\Diamond_{C_1}\delta_1 \wedge \Diamond_{C_2}\delta_2 \rightarrow \Diamond_{C_1\cup C_2}(\delta_1 \wedge \delta_2)$	
Outcomes and preferences		
(<i>func1</i>)	$\bigvee_{x \in K}(x \wedge \bigwedge_{y \in K \setminus \{x\}} \neg y)$	
(<i>func2</i>)	$(\text{ballot}(<) \wedge \varphi) \rightarrow \Box_N(\text{ballot}(<) \rightarrow \varphi)$	
(<i>incl</i>)	$\Box_N\varphi \rightarrow \blacksquare_i\varphi$	
(<i>K(<_i)</i>)	$\blacksquare_i(\varphi \rightarrow \psi) \rightarrow (\blacksquare_i\varphi \rightarrow \blacksquare_i\psi)$	
(<i>4(<_i)</i>)	$\blacklozenge_i\blacklozenge_i\varphi \rightarrow \blacklozenge_i\varphi$	
(<i>antisym'</i>)	$(\text{ballot}(<) \wedge \blacklozenge_i\text{ballot}(<')) \rightarrow \Box_N(\text{ballot}(<') \rightarrow \blacksquare_i\neg\text{ballot}(<))$	
(<i>total'</i>)	$(\text{ballot}(<) \wedge \blacklozenge_i\text{ballot}(<')) \vee \Box_N(\text{ballot}(<') \rightarrow \blacklozenge_i\text{ballot}(<))$	
(<i>unifPref</i>)	$(x \wedge \blacklozenge_i y) \rightarrow (x \blacktriangleleft_i y)$	
Rules		
(<i>MP</i>)	from $\vdash \varphi \rightarrow \psi$ and $\vdash \varphi$ infer $\vdash \psi$	
(<i>Nec(\Box_i)</i>)	from $\vdash \varphi$ infer $\vdash \Box_i\varphi$	

Figure 6. Logic of social choice functions $\Lambda^{scf}[N, K]$. i ranges over N , C_1 and C_2 over 2^N , x and y are over K , and $<$ is over $L(K)^N$. δ_1 and δ_2 are two formulae from $\mathcal{L}^{scf}[N, K]$ that do not contain a common atom from $At[N, K]$. φ represents an arbitrary formula of $\mathcal{L}^{scf}[N, K]$, and p an arbitrary atom in $At[N, K]$.

The axiomatics is largely inspired by the axiomatics of the logic of games and propositional control (henceforth LGPC) presented in (Troquard et al., 2009a). The logic LGPC is designed to model strategic games in general. The agents have arbitrary strategies, and preferences allowing for indifference between two different outcomes. On the other hand, in this paper we focus on SCFs and hence on particular strategic games that ‘represent’ an SCF (cf. Remark 1).

While in LGPC we had an axiom saying that every atom was actually controlled by at least one agent, here we are more specific as we know *a priori* which atoms are controlled by a given agent. This is the role of the axiom (*ballot*). Constraints of controls are also specific to the present study. The truth values of the controlled atoms cannot be independent of each other as we use them to encode preferences. In LGPC, all valuations of the controlled atoms were permitted.

THEOREM 2. (Soundness and completeness). $\Lambda^{scf}[N, K]$ is sound and complete with respect to the class of models of social choice functions.

Proof. The proof of completeness first gives an equivalent but more standard semantics to the logic: the Kripke models of SCF. Then we build the canonical model. For every consistent formula φ , we show how to isolate a sub-model M_φ that we prove is a Kripke model of SCF that satisfies φ .

Further details are given in the Appendix. ■

4. Applications

We have already demonstrated that the language allows to completely characterise an SCF. In this section we show how we can express properties of social choice functions in the language and apply the logic to reason about them.

The language can be used to characterise requirements on social choice functions. We first illustrate that with some simple properties, namely citizen sovereignty and non-dictatorship. Next, we will characterise a dominant strategy equilibrium. Finally, we provide a formalisation of monotonicity and strategy-proofness, and use standard results of SCT to show how we can use the logic to check whether an SCF is implementable in a dominant strategy.

4.1. CITIZEN SOVEREIGNTY AND NON DICTATORSHIP

We say that an SCF satisfies *citizen sovereignty* iff every outcome in K is feasible. That is, no outcome is rejected independently of the individual opinions. It is defined as follows.

DEFINITION 11. (Citizen sovereignty). *An SCF F satisfies citizen sovereignty iff for every $x \in K$ there is $a \in L(K)^N$ such that $F(\langle \cdot \rangle) = x$.*

The next formula is a straightforward translation of the definition of citizen sovereignty in the language of social choice functions.

$$\text{CITSOV} \triangleq \bigwedge_{x \in K} \diamond_N x.$$

We say that an SCF satisfies *non dictatorship* iff no player can always impose its favourite outcome.

DEFINITION 12. (Non-dictatorship). *An SCF F is non dictatorial iff for every player $i \in N$ there is a ballot $\langle \in L(K)^N$ such that $F(\langle) \prec_i y$ for some $y \in K \setminus \{F(\langle)\}$.*

This says that for every player, there is a ballot \langle whose outcome is $F(\langle)$, and i prefers an outcome that is not $F(\langle)$.

We can rewrite the definition of non dictatorship into the language of social choice functions as follows.

$$\text{NODICT} \triangleq \bigwedge_{i \in N} \diamond_N \left(\bigvee_{x \in K} \left(x \wedge \bigvee_{y \in K \setminus \{x\}} p_{y>x}^i \right) \right).$$

The following proposition is immediate.

PROPOSITION 1. *Consider a social choice function F and ρ^F a formula characterising it.*

1. *F has the property of citizen sovereignty iff $\models_{\Lambda^{\text{scf}}[N,K]} \rho^F \rightarrow \text{CITSOV}$.*
2. *F is non dictatorial iff $\models_{\Lambda^{\text{scf}}[N,K]} \rho^F \rightarrow \text{NODICT}$.*

4.2. DOMINANT STRATEGY EQUILIBRIUM

Citizen sovereignty and non dictatorship are possible properties of a social choice function: their formulations in logic are globally true (or false) in a model of SCF. However, the logic is also able to formalise solution concepts, which are properties of states. In (Troquard et al., 2009a), we characterised several solution concepts (dominant strategy equilibrium, Nash equilibrium, core membership...) that are directly applicable in the logic of the present work.

In order to formalise strategy-proofness later, we need to characterise a dominant strategy equilibrium. A dominant strategy equilibrium captures a particularly important pattern of behaviour. It arises when every player plays a dominant strategy, that is, a strategy that would represent the best choice whatever the other agents play. We define it directly in our models of SCF.

DEFINITION 13. (Dominant strategy equilibrium). *Let v^* be a state in a model of social choice functions $\langle N, K, \text{out}, (\prec_i) \rangle$. v^* is a dominant strategy equilibrium iff for every player $i \in N$ and every strategy $u_{N \setminus \{i\}} \in \text{strategies}[N \setminus \{i\}, K]$, we have $\text{out}(u_0 \dots u_i' \dots u_n) \prec_i \text{out}(u_0 \dots v_i^* \dots u_n)$ for every $u_i' \in \text{strategies}[i, K]$.*

A dominant strategy equilibrium is a strong solution concept: such an equilibrium does not depend on the knowledge of an agent i about the strategies or preferences of other players.

It is convenient to introduce the notion of best response by an agent i .

$$\text{BR}_i \triangleq \bigvee_{x \in K} (x \wedge \Box_i \blacklozenge_i x).$$

A player i plays a best response in a state if, x being the outcome, for every deviation of i , i prefers x .

We can now define strategy dominance in terms of best response:

$$\text{DOM} \triangleq \bigwedge_{i \in N} \Box_{N \setminus \{i\}} \text{BR}_i.$$

We have a strategy dominant state if the current choice of every player ensures them a best response whatever other agents do.

PROPOSITION 2. *Assume a model of social choice functions M and a state v^* . We have that v^* is a dominant strategy equilibrium iff $M, v^* \models \text{DOM}$.*

4.3. MONOTONICITY AND STRATEGY-PROOFNESS

One important property of SCF is *monotonicity*, as this property can affect the implementability of social choice functions.

DEFINITION 14. (Monotonicity). *An SCF F is monotonic iff for all $\{<, <'\} \subseteq L(K)^N$ and $x \in K$, if $F(<) = x$ and if for all $i \in N$, for all $y \in K$ we have that that $y <_i x$ implies that $y <'_i x$, then, $F(<') = x$.*

We propose to characterise monotonic social choice functions. We define

$$\text{MON} \triangleq \left\{ \begin{array}{l} \bigwedge_{< \in L(K)^N} \bigwedge_{<' \in L(K)^N} \bigwedge_{x \in K} [\Box_N(\text{ballot}(<) \wedge x) \wedge \\ \bigwedge_{i \in N} \bigwedge_{y \in K} (\Box_N(\text{ballot}(<) \wedge p_{x>y}^i) \rightarrow \\ \Box_N(\text{ballot}(<') \wedge p_{x>y}^i)) \rightarrow \Box_N(\text{ballot}(<') \wedge x)] \end{array} \right\}.$$

Although it may appear rather complex, the predicate MON is essentially nothing more than the expression of Definition 14 in our language $\mathcal{L}^{scf}[N, K]$. The following proposition is immediate.

PROPOSITION 3. *Consider a social choice function F and ρ^F a formula characterising it. F is monotonic iff*

$$\models_{\Lambda^{scf}[N, K]} \rho^F \rightarrow \text{MON}.$$

Monotonicity does not depend on the true preference profile of the players. Accordingly, our definition does not involve the modalities of preference $\blacklozenge_i \varphi$ and $\varphi \blacktriangleleft_i \psi$. Capitalising on standard results from social choice theory, we will show that using the full expressivity of our language (that is, using true preference modalities) we can obtain a much simpler formulation.

We say that an SCF is strategy-proof if for every preference profile, telling the truth (reporting the true preference) is a dominant strategy for every player.

DEFINITION 15. (Strategy-proofness). *An SCF F is strategy-proof iff F is truthfully DOM-implementable.*

Hence, a choice function is strategy-proof when it is truthfully implementable in dominant strategy: for every preference profile, reporting their true preference is a dominant strategy for every player.

The *revelation principle* (Gibbard, 1973) is a central result in implementation theory. It states that if an SCF is DOM-implementable, then it is truthfully DOM-implementable. It is true in general even if $L(K)$ is based on weaker orders. The revelation principle tells us that if an SCF F is implementable in dominant strategies then there exists a direct mechanism such that for every preference profile $<$, truth telling (every player i reports $<_i$) is a dominant strategy and the outcome is $F(<)$.

Truthful implementations are rather weak; it is easier in general to implement a choice function truthfully than with ‘standard’ implementations. Indeed, in truthful implementations there might be an equilibrium that leads to an outcome different of the one prescribed by the SCF. But because in this paper we consider linear preferences, and we assume that players cannot be indifferent between two distinct outcomes, such a situation cannot happen. Thus, we can be more specific than the revelation principle.

THEOREM 3. ((Dasgupta et al., 1979, Corollary 4.1.4)). *A direct mechanism g truthfully implements an SCF F in dominant strategies iff g DOM-implements F .*

Hence, when working in dominant strategies with linear preferences, the concepts of implementation and truthful implementation coincide.

We propose to characterise strategy-proof social choice functions as follows:

$$\text{STRPROOF} \triangleq \bigwedge_{< \in L(K)^N} [\text{true}(<) \rightarrow (\text{ballot}(<) \rightarrow \text{DOM})]$$

The formula STRPROOF is an immediate reformulation of the definition of strategy-proofness in our language of social choice functions.

PROPOSITION 4. *Consider a social choice function F and ρ^F a formula characterising it. F is strategy-proof iff*

$$\models_{\Lambda^{\text{scf}}[N,K]} \rho^F \rightarrow \text{STRPROOF}.$$

This Proposition provides us with a general procedure to check whether a social choice function is strategy-proof. Moreover (because of Theorem 3), because we restrict our attention to linear preferences, it allows us to check whether an SCF is DOM-implementable.

EXAMPLE 4. *We can verify for instance that the social choice function characterised in Example 3 is strategy-proof.*

$$\begin{aligned} \models_{\Lambda^{scf}[\{1,2,3\},\{a,b\}]} & (a \leftrightarrow (p_{a>b}^1 \wedge p_{a>b}^2) \vee (p_{a>b}^1 \wedge p_{a>b}^3) \vee (p_{a>b}^2 \wedge p_{a>b}^3)) \\ & \rightarrow \text{STRPROOF.} \end{aligned}$$

Monotonicity sometimes implies implementability and this is actually the case in our setting. Since we are working with *rich domains* of preferences¹ and linear orderings the following result holds.

THEOREM 4. ((Dasgupta et al., 1979, Cor. 3.2.3, Th. 4.3.1)). *An SCF is truthfully implementable in dominant strategies iff it is monotonic.*

This standard result of implementation theory shows that in our setting, the notions of monotonicity and of strategy-proofness match. Trivially we are actually able to substantially simplify the formula MON, our characterisation of monotonicity in the formal language. Indeed, as a consequence of Theorem 4, we have the following.

PROPOSITION 5.

$$\models_{\Lambda^{scf}[N,K]} \text{MON} \leftrightarrow \text{STRPROOF.}$$

5. Discussion and perspectives

We have presented the problem of direct implementation in social choice theory and proposed a logical formalisation of it. We were able to give a sound and complete axiomatization to the logic. We showed how we can characterise social choice functions and properties of social choice functions. And finally, we have demonstrated the value of the logic by proposing a general logical procedure for checking whether a social choice function is strategy-proof.

Our logical language is a formal counterpart of the language of “natural mathematics” that is typically used in social choice theory. There are however two features that make it particularly useful: (i) it is supported by a non ambiguous semantics; and (ii) the resulting logic is decidable.

¹ The notion of a rich domain is some tangential to the purposes of this paper. Briefly, our domain of preferences is rich because we allow every linear order of K . See (Dasgupta et al., 1979, Sec. 3.1)

Section 4 suggests a logical methodology for reasoning about problems of social choice theory with the logic of social choice functions. Let a collection of properties of social choice theory $P_i, i \in \{1, \dots, n\}$ be characterised in the logic $\Lambda^{scf}[N, K]$ by ρ^{P_i} , respectively.

1. *We can use the logic in order to check whether an SCF satisfies a certain property.* An SCF F characterised by ρ^F has the property P_1 iff $\rho^F \rightarrow \rho^{P_1}$ is derivable in $\Lambda^{scf}[N, K]$.
2. *We can use the logic in order to evaluate the strength of constraints in SCT.* P_1 is a property weaker than P_2 iff the formula $\rho^{P_2} \rightarrow \rho^{P_1}$ is derivable in $\Lambda^{scf}[N, K]$. For instance, instead of using a result of SCT to prove Proposition 5, we could actually use the logic to automatically verify that monotonicity and strategy-proofness coincide in the current setting. More interestingly, we could use it to prove new theorems.
3. *We can use the logic for mechanism design.* Building a mechanism that implements a social choice procedure satisfying the properties P_1, P_2, \dots, P_n consists of finding a model for the formula $\rho^{P_1} \wedge \rho^{P_2} \wedge \dots \wedge \rho^{P_n}$.

We believe these are exciting possibilities for social choice theory and logic, and as the logic is decidable, they are in principle possible.

Acknowledgements

An earlier abstract of this paper appeared as (Troquard et al., 2009b). We thank the anonymous reviewers for their suggestions that helped to improve the paper. We are also grateful to the participants of TARK'09. This research is funded by the EPSRC grant EP/E061397/1 *Logic for Automated Mechanism Design and Analysis (LAMDA)*.

References

- Areces, C. and B. ten Cate: 2006, *Hybrid Logics*, Vol. Handbook of Modal Logic, Chapt. 14, pp. 821–868. Elsevier Science Inc.
- Arrow, K.: 1950, 'A Difficulty in the Concept of Social Welfare'. *Journal of Political Economy* **58**(4), 328–346.
- Arrow, K. J., A. K. Sen, and K. Suzumura (eds.): 2002, *Handbook of Social Choice and Welfare*, Vol. 1. Elsevier.
- Bonzon, E., M.-C. Lagasquie-Schiex, and J. Lang: 2007, 'Efficient coalitions in Boolean games'. In: K. Apt and R. van Rooij (eds.): *New Perspectives on Games and Interaction*, Vol. 4 of *Texts in Logic and Games*. Amsterdam University Press, pp. 283–297.
- Chellas, B. F.: 1980, *Modal Logic: an introduction*. Cambridge University Press.

- Dasgupta, P., P. Hammond, and E. Maskin: 1979, ‘The Implementation of Social Choice Rules: Some General Results on Incentive Compatibility’. *Review of Economic Studies* **46**, 185–216.
- Gibbard, A.: 1973, ‘Manipulation of voting schemes: a general result’. *Econometrica* **41**(4), 587–601.
- May, K.: 1952, ‘A set of independent, necessary and sufficient conditions for simple majority decision’. *Econometrica* **20**(4), 680–684.
- Moulin, H.: 1983, *The Strategy of Social Choice*, Advanced Textbooks in Economics. North Holland.
- Osborne, M. J. and A. Rubinstein: 1994, *A Course in Game Theory*. The MIT Press.
- Pauly, M.: 2001, ‘Logic for Social Software’. Ph.D. thesis, University of Amsterdam. ILLC Dissertation Series 2001-10.
- Satterthwaite, M. A.: 1975, ‘Strategy-proofness and Arrow’s Conditions: Existence and Correspondence Theorems for Voting Procedures and Social Welfare Functions’. *Journal of Economic Theory* **10**, 187–217.
- Troquard, N., W. van der Hoek, and M. Wooldridge: 2009a, ‘A Logic of Games and Propositional Control’. In: Decker, Sichman, Sierra, and Castelfranchi (eds.): *8th International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS-09)*, Budapest, Hungary. pp. 961–968.
- Troquard, N., W. van der Hoek, and M. Wooldridge: 2009b, ‘A logic of propositional control for truthful implementations’. In: *TARK ’09: Proceedings of the 12th conference on Theoretical aspects of rationality and knowledge*. pp. 237–246.
- van der Hoek, W. and M. Wooldridge: 2005, ‘On the logic of cooperation and propositional control’. *Artificial Intelligence* **164**(1-2), 81–119.
- Wooldridge, M., T. Ågotnes, P. E. Dunne, and W. van der Hoek: 2007, ‘Logic for Automated Mechanism Design — A Progress Report’. In: *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence (AAAI-2007)*. Vancouver, British Columbia, Canada.

Proof of Theorem 2

$\Lambda^{scf}[N, K]$ is sound and complete with respect to the class of models of social choice functions.

Proof. It is routine to verify that all principles of Figure 6 are valid. We show that if a formula is consistent, it is provable in the system $\Lambda^{scf}[N, K]$.

We first introduce the Kripke models of SCF. A *Kripke model of SCF* is a tuple $M = \langle N, K, S, (R_i), (P_i), V \rangle$ such that:

- N and K are parameters;
- $S = \{V \in \Theta^{At[N, K]} \mid \forall i \in N, \exists V_i \in \text{strategies}[i, K] \text{ s.t. } V = \cup_{i \in N} V_i\}$;
- V is a valuation function of $At[N, K] \cup K$ where for every $v \in S$:
 - $p \in V(v)$ iff $p \in v, p \in At[N, K]$;
 - there is a unique $x \in K$ s.t. $x \in V(v)$; [\Leftarrow we say that the model is based on the outcome function out^M when $out^M(v) = x$ iff $x \in V(v)$].
- $R_i v u$ iff $v_j = u_j$ for all $j \neq i$;
- there is a $<^M \in L(K)^N$ s.t. $P_i v u$ iff (if $x \in V(v)$ and $y \in V(u)$ then $x <_i^M y$); [\Leftarrow we say that the model is based on $<^M$].

Truth values of $\diamond_i\varphi$ and $\blacklozenge_i\varphi$ in a Kripke model of SCF are obtained in the standard way from the relations R_i and P_i , respectively.

Clearly, for every Kripke model M based on out^M and $<^M$, we can construct a model of social choice functions $M^{scf} = \langle N, K, out^M, (<_i^M) \rangle$ and reciprocally.

By construction, there exists a bijection $f : S \rightarrow strategies[N, K]$ that associates a state s in M to a state $v = (v_1 \dots v_n)$ in M^{scf} in such a way that for every $p \in At[i, K]$, we have $p \in V(s)$ iff $p \in v_i$.

The following is easy to see.

CLAIM 1. $M, s \models \varphi$ iff $M^{scf}, f(s) \models \varphi$.

Hence, the proof of the theorem can be reduced to a proof of completeness of the logic wrt. to the class of Kripke models of SCF.

Let Ξ be the set of maximally consistent sets (mcs.) of $\Lambda^{scf}[N, K]$. We define the proper canonical model $M^{can} = \langle N, K, S, (R_i), (P_i), V \rangle$ as follows. N and K are the parameters of the logic. $S = \Xi$. $R_i\Gamma\Delta$ iff $\forall\delta \in \Delta, \diamond_i\delta \in \Gamma$. $P_i\Gamma\Delta$ iff $\forall\delta \in \Delta, \blacklozenge_i\delta \in \Gamma$. $p \in V(\Gamma)$ iff $p \in \Delta$. $x \in V(\Gamma)$ iff $x \in \Delta$.

Given an mcs. Γ_0 we define the set of mcs. ‘describing’ the same SCF and where the players have the same true preferences (modulo the preferences concerning some outcome which is not feasible in the SCF):

$$\begin{aligned} Cluster(\Gamma_0) \triangleq & \{ \Gamma_1 \mid \forall < \in L(K)^N, \forall x \in K, \diamond_N(\text{ballot}(<) \wedge x) \in \\ & \Gamma_1 \text{ iff } \diamond_N(\text{ballot}(<) \wedge x) \in \Gamma_0 \} \cup \{ \Gamma_2 \mid \forall i \in N, \forall \{x, y\} \subseteq \\ & K, x \blacktriangleleft_i y \in \Gamma_2 \text{ iff } x \blacktriangleleft_i y \in \Gamma_0 \} \end{aligned}$$

Let φ be a consistent formula of $\mathcal{L}^{scf}[N, K]$. There is an mcs. Γ_φ s.t. $\varphi \in \Gamma_\varphi$. The proof consists in constructing a model from Γ_φ such that it is indeed a Kripke model of SCF and there is a state satisfying φ .

We define $M_\varphi = \langle N', K', S', R'_i, P'_i, V' \rangle$ from M^{can} as follows:

- $N' = N$ and $K' = K$;
- $S' = \Xi|_{Cluster(\Gamma_\varphi)}$;
- $R'_i = R_i|_{Cluster(\Gamma_\varphi)}$;
- $P'_i = P_i|_{Cluster(\Gamma_\varphi)}$;
- $p \in V'(\Delta)$ iff $p \in V(\Delta)$, $\Delta \in S'$.

It is immediate that the truth lemma holds.

CLAIM 2. $M_\varphi, \Gamma_\varphi \models \delta$ iff $\delta \in \Gamma_\varphi$.

Hence, $M_\varphi, \Gamma_\varphi \models \varphi$.

The set of states in Kripke models of SCF is defined as the set of valuations of $At[N, K]$ encoding a preference profile. We prove that there exists a bijection between S' and $L(K)^N$.

CLAIM 3. *The following statements are true:*

1. $\forall \Delta \in S', \exists! \prec \in L(K)^N$ s.t. $\text{ballot}(\prec) \in \Delta$;
2. $\forall \prec \in L(K)^N, \exists! \Delta \in S'$ s.t. $\text{ballot}(\prec) \in \Delta$.

The first part of the claim follows from the constraints of control (*refl*), (*antisym-total*) and (*trans*). We now argue that for every $\prec \in L(K)^N$, there is exactly one $\Delta \in S'$ such that $\text{ballot}(\prec) \in \Delta$. Let $\prec \in L(K)^N$. We have $\vdash \diamond_i \text{ballot}_i(\prec)$ by (*ballot*). With (*comp-At*), we find that $\vdash \diamond_N \text{ballot}(\prec)$. Hence, $\diamond_N \text{ballot}(\prec) \in \Gamma_\varphi$, and there must be an mcs. Δ s.t. $\text{ballot}(\prec) \in \Delta$. Now suppose that $\Delta' \in S'$ also contains $\text{ballot}(\prec)$. By (*func2*), Δ and Δ' contain the same formulae. Then $\Delta' = \Delta$, which proves the second part of the claim.

As a consequence we will be allowed to use the formulae of the form $\text{ballot}(\prec)$ as world labels in M_φ .

We now prove the main claim of this proof.

CLAIM 4. M_φ is a Kripke model of SCF.

We first prove that for every mcs. Γ and Δ , we have that $R_i \Gamma \Delta$ iff for all $i \neq j$ we have that $\text{ballot}_j(\prec) \in \Gamma$ iff $\text{ballot}_j(\prec) \in \Delta$.

First, observe that for every i , R_i is an equivalence relation because by axioms (*K(i)*), (*T(i)*), (*B(i)*) and (*comp* \cup) all \Box_i are S5 modalities.

(\Rightarrow). Suppose $R_i \Gamma \Delta$. Then by definition $\forall \delta \in \Delta$ we have $\diamond_i \delta \in \Gamma$. For any $\prec \in L(K)^N$ and $j \neq i$, suppose also that $\text{ballot}_j(\prec) \in \Delta$. By (*exclu*), $\Box_i \text{ballot}_j(\prec) \in \Delta$. Then by hypothesis $\diamond_i \Box_i \text{ballot}_j(\prec) \in \Gamma$, which by (*B(i)*) entails that $\text{ballot}_j(\prec) \in \Gamma$. Because $R_i \Gamma \Delta$ is an equivalence relation, the same reasoning can be done to prove that if $\text{ballot}_j(\prec) \in \Gamma$ then $\text{ballot}_j(\prec) \in \Delta$.

(\Leftarrow). Suppose $\forall j \neq i, \forall \prec \in L(K)^N$ we have $\text{ballot}_j(\prec) \in \Gamma$ iff $\text{ballot}_j(\prec) \in \Delta$.

Suppose that $\text{ballot}_i(\prec') \in \Delta$ and $\delta \in \Delta$. Let us note \prec_Δ the preference profile $(\prec_1, \dots, \prec'_i, \dots, \prec_n)$. We hence have $\text{ballot}(\prec_\Delta) \wedge \delta \in \Delta$. Which by (*func2*) means that $\Box_N(\text{ballot}(\prec) \rightarrow \delta) \in \Delta$.

From (*exclu*), $\Box_i \bigwedge_{j \neq i} \text{ballot}_j(\prec) \in \Gamma$. By (*ballot*), we also have that $\diamond_i \text{ballot}_i(\prec') \in \Gamma$. Hence, by S5, $\diamond_i \text{ballot}(\prec_\Delta) \in \Gamma$.

We obtain that $\diamond_i \delta \in \Gamma$.

We now prove that there is a linear order $\prec \in L(K)^N$ such that $P_i \Gamma \Delta$ iff (if $x \in V(\Gamma)$ and $y \in V(\Delta)$ then $x \prec_i y$). For every $i \in N$, we construct an order \prec_i° over the set $K^\circ = \{x \in K \mid \diamond_N x \in \Gamma_\varphi\}$ such that $x \prec_i^\circ y$ iff $x \blacktriangleleft_i y \in \Gamma_\varphi$. (Note that the reason we restrict the preliminary construction of the preference order to the set of possible outcomes is because the language is not strong enough to talk about impossible outcomes. See Remark 4. A careless construction could lead to a relation over K that is not a linear order.)

Capitalising on (*unifPref*), it is immediate that \prec_i° is transitive (4(\prec_i)), antisymmetric (*antisym'*) and total and reflexive (*total'*). Then \prec_i° is a linear order over K° .

It is now easy to obtain a linear order $<_i$ over K such that for all x and y in K° we have $x <_i y$ iff $x <_i^\circ y$.

This completes the proof that M_φ is a Kripke model of SCF.

Then, for every consistent formula φ , there is a Kripke model of SCF in which φ is satisfied. ■

