Expressing Properties of Social Choice Functions Using Modal Logic

by

Erik Parmann

MSc Thesis (Masteroppgave)
for the degree
Master of Science in Informatics

(Master i Informatikk, Programutvikling)

Faculty of Mathematics and Natural Sciences
University of Bergen

November 21, 2011

Det matematisk- naturvitenskapelige fakultet
Universitetet i Bergen

Abstract

In this thesis we explore the problem of expressing properties of social choice functions using modal logic. The strategy is to create a modal logic that uses social choice functions as models, and where true formulae express properties of the social choice function in which they are evaluated. We will especially try to model the properties needed to express the Gibbard-Satterthwaite theorem: dictatorship, strategyproofness and that there are at least three alternatives that can win.

We will start with a positive result, showing that the models chosen correspond nicely with social choice functions. We will then show several negative results, in that even a very simple modal logics on these models is undecidable. We then give a method for expressing properties of certain families of social choice functions, and we look at possible extensions of the basic language to increase expressivity.

Acknowledgment

This thesis would not have been possible without my supervisor, Thomas Ågotnes. I also want to thank him for helping me travel to ILLC in Amsterdam twice, first as an exchange student, and secondly as an independent visitor. He was much helpful, reading emails and providing supervision over Skype.

In Amsterdam there is a long range of people that deserves thanks. Among the staff I want to thank Ulle Endriss for pushing me to hold a seminar when it was most needed, and the whole Computational Social Choice group for providing feedback. I also want to thank Yde Venema for a short, but insightful conversation, providing deeper insight (an a section in this thesis).

There is a large number of students at the ILLC that should also be thanked for making my stays there extra-ordinary. I want to thank Paula Henk for letting me bike her around (and recursion theory), Marta Sznajder for making me cakes, Peter Fritz for many rounds of pool (and some whiskey) and Kasper Christensen for having such a weak right arm (and for just being Danish). In addition Ilan Frank for awesome naps (and your mother's cookies), Gabriela Aslı Rino Nesin for many rounds of booth ping-pong and backgammon (and for the conversations in between), Maja Jaakson for delicious breakfasts (and dinners and Portal 2), and finally, Johannes Marti for showing me how logic is done. You have all made those two semesters some of the bests of my life.

Some of you deserve extra thanks, both Maja and Gabriela have done more than expected in reading my drafts, and helping me with my English. If it were not for you, the rest of the thesis would be as badly written as this. In addition, you were the two most important people for me in Amsterdam, and I thank you both for existing.

In Bergen I want to thank Øyvind Døskeland and Øystein Rolland, for staying my friends and dragging me out, even when I was deep in my own world. I want to thank my family, Øystein, Nina and Anders for always being there. I also want to thank everyone in Bergen logic group, you help keep logic fun.

Finally, my deepest gratitude towards Pål Grønås Drange. I can not imagine being a student without you. I thank you for coffee in the morning, chicken baguette for lunch, bread for dinner, and IPA for supper. I thank you for conversations and for silence. I thank you for the past five years.

Contents

Chapter 1

Introduction

> "There's definitely, definitely, definitely, no logic to human behaviour."
> —*Human Behaviour, Björk Guðmundsdóttir*

To explain the title of the thesis we need to understand several concepts. In this chapter we will take a broad overview of the two subjects of this thesis, voting theory and modal logic. The first part about voting theory will enable the reader to understand what the phrase "properties of social choice functions" means, while the second part will give an overview of modal logic. This chapter will not be very technical; those endeavours are for later chapters. The purpose of this chapter is to provide the reader with intuitions concerning the problems which we will define more formally in the later chapters.

1.1 Voting theory

Voting theory is a branch of game theory with direct applications in society. It concerns the aggregation of preferences for a group of agents, i.e. how to select a winner from a poll, and the different properties inherent in different aggregations. The most prominent real-world example is maybe the many different election systems, where votes are aggregated into winner(s). Studying the properties of voting, we quickly notice that there are many different ways in which such a process can take place. A voter might be able to vote on one item or several; there may be one or several winners; and although sometimes all voters have the same number of votes, this does not hold in all cases (e.g. when the number of votes is relative to the number of stocks held). And we still have not touched upon the properties of the actual aggregation function, the procedure followed in order to determine the winner(s).

A *voting system* needs to specify all the above aspects. That is, it needs to specify what the voters can submit as their votes. The vote is usually called a *ballot*. It also needs to specify the exact procedure by which the winner(s) are found. When we have specified a voting system, we can examine its features. This is the object of study in voting theory.

In this thesis we will consider a special class of voting systems, where the aggregation function is a so called *social choice function* (SCF). These are functions where the voters give a preference ordering over a set of *alternatives*, and the social choice function returns one winning alternative.

Knowing what a social choice function is, we still need to know what a property of such a function might be. An example of a much-studied property of social choice functions is *dictatorship*, or its converse, non-dictatorship. Intuitively, a voter is a dictator if he can choose what the winning alternative will be. An SCF is dictatorial if there is a dictator, and non-dictatorial if there is no dictator. Another property is *strategyproofness*. Roughly speaking, an SCF is strategyproof when every voter is always best off if he is honest about her preferences; that is, there is never any incentive to play "strategically". The reader might be aware that many elective systems in use are not strategyproof. There are several features of a voting system we could be interested in studying, ranging from abstract properties of the aggregation function to its computational complexity. In this thesis, we will focus on the former.

The most famous result in voting theory is Arrow's impossibility theorem[3]. This theorem is not about social choice functions, but about social welfare functions. While both social choice and social welfare functions take in linear orders for each voter, latter functions do not return a single, winning alternative, but a whole linear order. Intuitively, the social welfare function returns, as it were, the society's ranking of *everything*. Arrow's theorem states that a certain combination of "nice" features of a voting system are incompatible, so there can be no voting system incorporating them all.

Closely related to, and inspired by, Arrow's impossibility theorem is the Gibbard-Satterthwaite theorem[14, 8], proved independently by Allan Gibbard and Mark Satterthwaite. The Gibbard-Satterthwaite theorem is about social choice functions. It states that, if the social choice function has more than three alternatives that can win and is it is strategyproof, then there must be a dictator. This claim holds for all voting systems in which the aggregation function is a social choice function, and is thus a very broad claim. These two famous impossibility results are closely connected. This is shown in e.g. [13], which gives side-by-side proofs of both theorems; these demonstrate that the same proof can, in essence, be given for either theorem.

In this thesis we will focus on the properties needed to express the Gibbard-Satterthwaite theorem. The reason for this is twofold. Firstly, there has never been a fully-automated proof of either the Gibbard-Satterthwaite or Arrow's theorem; constructing a logic capable of expressing the properties related to these theorems could be a step in this direction. Secondly, the Gibbard-Satterthwaite theorem is about rather fundamental properties of social choice functions. Consequently, being able to express these properties provides a good "bench test" for a logic. Incidentally, making a logic which can express all three properties relevant to the Gibbard-Satterthwaite theorem is significantly more difficult than constructing a logic capable of expressing only two of those three properties.

1.2 Modal logic

In this section we will give an explanation of what modal logic is. This is a harder feat than it may first appear to be. One of the definitive guides to modal logic, [5], begins its preface as follows:

> Ask three modal logicians what modal logic is, and you are likely to get at least three different answers. The authors of this book are no

exception, so we won't try to start off with a neat definition.

Nonetheless, we will try to give the reader an intuition of what logic is. The reader is warned, however, that as the quote above suggests, others may give a different answer to the question of what modal logic is.

1.2.1 Logic in general

Before we start with modal logic, it is useful to understand what a logic is. And before understanding what a logic is, we need to understand what a language is.

In its most general form, a language is just a set of strings, and a string is part of the language exactly when it is an element of that set. When a string is part of a language that is used for a logic, the string is usually called a *formula*. The reader should be aware that this definition of a language is deceptive in its simplicity; it allows for *any* set to be a language. Furthermore, as the reader might be aware, there are many sets, most of which are difficult to fathom. As we will build complicated things on top of our languages, we prefer them to be manageable. Logicians are therefore mostly interested in languages which are not only well-defined, but also *decidable*. A language is decidable if, whenever we are given a string and asked whether it is part of the language or not, we can give an answer. We will discuss the notion of decidability in more depth later.

We will now give an example that should be familiar to most people. If we get the strings $4+5+4$ and $45++4$, and are asked which are meaningful arithmetic formulae, most people will be able to provide a correct answer. The former is an element in the language of arithmetic, while the latter is not. When we are taught at school which strings constitute meaningful expressions in arithmetic and which are not, we are actually being taught the language of arithmetic. It is important to note that something being part of the language does not mean it is "true" in any sense of the word. To illustrate this, we consider another example from arithmetic: even though the string $4 + 5 = 5 + 6$ is a formula, it is false. Truth is left for the logic to handle.

In its most general form, a logic is just a set of sentences in a formal language, and any set of sentences can be seen as a logic. However, we are mostly interested in logics in which the set of sentences capture the "truth" of something. One such logic is *propositional logic*, which captures exactly the valid statements expressible in its language. Often, a logic is defined relative to some class of structures called *models*. Then we have mathematically defined structures, and we tell how to interpret any formula in our language on these structures. Then we define the logic to be exactly those formulae which are true in all the structures of interest. The classes of all graphs or of all finite trees serve as good examples of classes of potentially interesting structures. As these examples indicate, the class of structures need not be finite; indeed, it usually is not.

We may, in some circumstances, choose to define a logic by some syntactic rule, without talking about structure at all. At other times, we may do both and be interested in whether or not the two definitions of our logic coincide.

Let us pause to take stock. A language tells us which strings are meaningful, those we call formulae. A logic is an interesting subset of those formulae; these formulae are often interesting because we see them as the "true" formulae expressible in the language.

We are interested in making a language capable of expressing properties of social choice functions and in building a logic on top of this language. The logic will be the formulae true of any social choice function. If this logic is decidable (see below), we will be able to prove how different properties of social choice functions relate to each other. If the logic is capable of expressing strategyproofness, dictatorship, and that there are more than three alternatives that can win (*3-winning*), we can use the logic to prove the Gibbard-Satterthwaite theorem.

1.2.2 Modal logic

Modal logic started out as the logic of modality; that is, of possibility and necessity. Modal logic has since been extended to a whole family of different languages and logics, with many different properties. In the spirit of the quote starting this section, we will not try to define what modal logic is; instead, we try to give the reader an intuition. We will do this by giving the *basic modal language* and some formulae in it, and an explanation as to how we evaluate the truth of such formulae. The reader will see similarities between the basic modal logic and the logics defined later in this thesis.

The basic modal language is defined relative to a set of propositional letters P, whose elements are usually denoted $p, q, r, \ldots$. We usually assume that it is countably infinite. The basic modal language is defined by the following Backus–Naur Form (BNF):

Definition 1.1 (Basic modal language on P)**.**

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \Box\varphi$$

where $p \in \mathsf{P}$.

The language is defined in terms of a BNF. A BNF is a recursive definition of the language, which says which strings are legal formulae. The BNF above says that the "basic" building blocks of the language are propositional letters like p. If we have those, we can construct larger formulae by either putting a "$\neg$" in front of a formula, putting a "$\Box$" in front of a formula, or combining two formulae to create a new one with "$\vee$". It is worth mentioning that the character "$|$" is a meta-character used to describe the language; it is not a part of the language itself. Thus, no formula contains a "$|$". We also often use parentheses to clarify the structure, without specifying them as part of the language. Some examples of formulae are:

$$p \vee q$$
$$p \vee \neg p$$
$$\Box(p \vee (q \vee \neg q))$$
$$\neg((\Box p \vee \Box\neg p) \vee \neg(q \vee \neg\Box q))$$

Notice that one can also use q, not only p. This is because the p in the BNF refers to any element of Φ and, as mentioned previously, we usually refer to such elements by using letters $p, q, r, \ldots$. Some examples of non-formulae strings are:

$$\neg\vee$$
$$q\neg$$
$$p|q$$

The intended reading is as follows: It is thought that each propositional letter in Φ refers to some proposition, that is, something that can be true or false. The formula p claims that the proposition p holds, and the formula holds if the proposition it expresses holds. If the formula is $\neg p$ the formula claims that p does not hold, so $\neg$ negates the formula it precedes. In the same way, if φ is some formula, then $\neg\varphi$ claims that the formula φ is false. $\neg\neg p$ claims that it is false that it is false that p holds, i.e. that p holds. $\vee$ is used to combine two formulae into a disjunction. That is, $\varphi \vee \psi$ claims that φ or ψ holds (and maybe both of them). Thus, the formula $p \vee \neg p$ claims that p holds or it does not.

So far we have described what is called propositional logic. The new addition modal logic brings to propositional logic is the box($\Box$). $\Box\varphi$ is to be read as "it is necessary that φ". One can now combine the parts of the language and make more complicated formulae. One such formula is $\neg\Box\neg\varphi$, which says "it is not necessary that not φ". This is usually read as "φ is possible".

As in propositional logic, we introduce some extra symbols to represent certain combinations which we often write. Instead of writing $\neg\Box\neg\varphi$, we write $\Diamond\varphi$; and instead of $\neg\varphi \vee \psi$, we write $\varphi \to \psi$, read as "φ implies ψ". We also use $\psi \wedge \varphi$ for $\neg(\neg\psi \vee \neg\psi)$, read as "$\psi$ and φ".

Now we have the language and a basic idea about what its formulae are supposed to mean. However, in order to reason clearly about formulae in this language, it helps to formally define what the formulae "mean"; that is, when they are true. This is the formal semantics. The models for the basic modal language are *pointed, labelled, directed graphs.* Depending on what we are trying to model, these graphs may also be called Kripke structures, epistemic models or labelled transition systems. By "pointed, directed graph", we mean a directed graph with a designated node in it, which is the "point". Formulae are interpreted in a node of the graph. A directed graph is usually called a *frame* and a labelled frame is usually called a *model.* A model with a point it is called a *pointed model.*

The labelling assigns to each node in the frame a set of propositional letters, which we will interpret as the true propositional letters in that node. A proposition letter p holds in a node in the frame if it is an element of the label for that node.

The connectives from propositional logic work as explained above, while $\Box\varphi$ is true in some point in the model if φ holds for all the points to which it is connected. Then $\Diamond\varphi$, which is an abbreviation for $\neg\Box\neg\varphi$, becomes true if not all neighbouring points are such that $\neg\varphi$. In other words, $\neg\Box\neg\varphi$ is true if φ holds for some neighbouring point. Thus, from one point in the frame, one can "look" around in the frame by making formulae with $\Box$.

A usual interpretation of the graph is to see it as modelling possible worlds. Each node in the graph represents a possible state in which the world can be, and one point relates to another if the second is seen as a possibility from the first. The formula $\Box\varphi$ then means that φ is true in all possible worlds, which fits well with our reading that φ is necessary.

When we know how to evaluate a formula in a pointed model, we can easily extend it to saying that a formula is globally true in a model if it is true in all nodes on that model.

It is also clear that, given some frame and a set of propositional letters Φ, it is possible to construct many models based on that frame. That is, there are many possible labellings of the nodes. If some formula φ is globally true in all models built from some frame, we say that φ is valid on the frame. With this notion we can use modal logic as a language to define graph classes. It is a fine exercise for a beginning modal logician to prove e.g. that $\Box p \rightarrow p$ is true in exactly the reflexive frames; that is, on frames where every point is related to itself.

A modal logic is often defined as the set of all formulae true in some specific class of frames. One can, for instance, define the modal logic of all formulae true in reflexive frames, transitive frames, or in frames with an equivalence relation.

One may also define a modal logic as all those formulae which are deducible by following some rules. One normally does this by giving a set of formulae called *axioms* and rules for transforming formulae into other formulae. For example, a rule that is often used is Modus Ponens, which says that if we have $\varphi \rightarrow \psi$ and φ, we can infer ψ. One can define a logic as the set of all formulae into which the axioms can be transformed.

An obvious question is now whether we can define the same logic both semantically, as the logic of a class of frames, and syntactically, as the logic of some deduction rules and axioms. That is, we can ask whether we can exactly capture the truths of a class of frames by using some set of rules of formulae. This is the question of soundness and completeness of a deduction system. A system is sound if all the deduced formulae hold in the frame class, while it is complete if all the formulae valid in the frame class are deducible in the deduction system.

1.3 Decidability

A concept of importance to us is that of a *decidable* decision problem. By a decision problem, we mean a yes-no question raised within a formal system. A decision problem is decidable if it is theoretically possible for a computer to solve it. Conversely, a decision problem is undecidable if it is impossible to construct a computer program which solves the problem—no matter how smart the programmer or how fast the computer. In 1928, David Hilbert put forth one of his mathematical challenges, the "Entscheidungsproblem", by posing the question as to whether every decision problem in formal systems is decidable. This problem was solved independently by Alonzo Chruch in 1935-36 and by Alan Turing in 1936-37, both of whom gave negative answers. They both showed that there exist well-defined problems in formal systems such that no algorithmic procedure can solve them.

Turing's proof is the most approachable, and it can be described in a few lines. Owing to its general beauty and the fact that undecidability will play such an important role for us later on, we include it here. The proof asks whether there exists some computer program which can take *any* computer program as its input, and tell whether that computer program will either terminate or run forever. Assume that it does, and call that function $h(i, x)$. $h(i, x)$ returns **Yes** or **No**, depending on whether the program i halts with x as input. Now, define $g(i, x)$ such that g halts if and only if h answers **No** on (i, x). This is clearly possible if h exists. Now, we ask:what happens when we run $g(g, x)$? We get that g halts if and only if g does not halt, which is obviously a contradiction. Thus we must reject our assumption that there exist such a program h.

Undecidable problems, then, do exist; but how do they concern us? We are working toward creating a logic for social choice functions partly because we wish to reason about them. Several decision problems about social choice functions which may interest us; we can ask whether a property holds for all social choice functions, or whether a specific social choice function has some property. It is of interest to us whether decision problems such as these are decidable or not.

It is clear that decidability depends on the power of our language. If the language is very expressive, we can express difficult problems in it—perhaps even undecidable problems. On the other hand, a highly-expressive language might also enable us to express interesting properties. When we create our language, then, we will have to accept a trade-off between expressibility and computational complexity.

1.4 Modal logic and voting theory

Proposing the connection between modal logic and voting theory raises a few questions. One is why we are interested in a logic in the first place; a second is why we would then choose a modal logic.

We tackle the former question first. As described above, a logic consists of a formal language and a way of interpreting the formulae in the language. The obvious advantage of using a formal language is that it allows for the precise communication of ideas. Another advantage is that the level of precision needed to formalise a notion can reveal subtle nuances, providing deeper insight into the notion of study.

Now, this does not give any reason for why we would want to develop yet another logic. Why not formalise the properties in the general language of mathematics, using sets, relations and functions? There are many reasons to avoid doing this. Firstly, some notions are just more easily communicated in a dedicated formal language. We can illustrate this point by considering basic modal logic, with the language defined in Definition 1.1. Using $\Box p$ to mean "p is necessary" makes it easier to communicate the latter; $\Box p$ is clearly easier to read than the equivalent definition of necessity which uses first-order quantifiers and relations in the model. Secondly, to define the simplest possible language, we need to find the essential properties of the notions we want to express. Looking for these properties can provide us with further insight into the notions we want to formalise.

We are not the first to promote the simplification of formal languages. In [12], Pauly defends what he calls *formal minimalism*, arguing that, when axiomatizing properties, one should aim for the smallest possible "reasonable" language. He claims that this gives us a way to compare different notions in terms of the complexity of the languages in which they are formalised. In this thesis, we will strive for formal minimalism; and insofar as modal logics are often relatively simple, making use of them is especially well-suited to our goals.

A relevant issue here concerns what constitutes a "reasonable" language. Clearly, if one wants to express the notion of dictatorship, a simple language would be one with only one symbol, d, meaning "dictatorial". In the same spirit one could add s and 3, meaning "strategyproof" and 3-winning, and express the Gibbard-Satterthwaite with the formula $s \wedge 3 \rightarrow d$. Or, to make it even more extreme, one could have just one symbol, g, with the interpretation "the Gibbard-Satterthwaite is true". It is clear that this is not a good choice, as such a language is too simple to provide any insight into the underlying structure of SCFs. It also tells us nothing about the interaction between different properties. We will not try to define what is meant by a "reasonable" language, we will return to this problem later on.

Another third reason for constructing a new logic concerns decidability. The general language of mathematics is not decidable; but if we constrain the language, we may be able to find a decidable fragment. If we can express interesting properties in a logic, we might also be able to use it to express well-known theorems like the Gibbard-Satterthwaite theorem or Arrow's theorem, and then use the decidability to make a fully automated proof. In [11], Arrow's and the Gibbard-Satterthwaite theorems are formalised in the logic HOL (Isabelle). With this approach the proof is human-made, but verified by a computer. None of these theorems have, to the author's knowledge, a fully-automated proof.

Closely related to the problem of decidability is that of *function synthesis*. This is the problem of synthesising (creating) a function from a specification. In our case, a natural problem is to construct a social choice function from a specification. Having made a logic expressing properties of social choice functions, we can use this to specify properties of functions, and then try to synthesise social choice functions with those properties.

There are arguments regarding decidability (and computational complexity in general) for why we would want to use *modal* logic as a basis. Many modal logics are decidable, and by making our logic a modal logic, we hope to inherit this property. In general, the complexity hierarchy of different modal logics is well understood. If the modal logic we make to express some property is similar enough to a well-known modal logic, we may be able to extend complexity results about the latter logic to our new one.

In addition, there are some similarities between notions used in voting theory and notions often expressed in modal logic. Recall that modal logic is often used to model a possible-world scenario, in which a state is connected to other states which are seen as possible from the original state. We can, and typically do, add one relation per agent to model the agents', i.e. voters, differing views of the state in which the world is. Voting theory, on the other hand, concerns how voting systems react to the different behaviours of the agents. Strategyproofness, for example, constrains how the SCF can react to an agent lying about her preferences. On the other hand, from the agent's perspective, strategyproofness is a statement about what the outcome will going to be in all possibly accessible worlds (accessible by lying, that is) for that agent.

When used to describe interaction of agents, modal logics often tend toward expressing "local" properties for each agent. That is, they tend to express properties which say something about the relation between a state and neighbouring states. Strategyproofness is an example of a local property.

Other properties, such as dictatorship, are of a more "global" nature. In order to express dictatorship, we need to say that, in any possible situation, the outcome is better for the agent than any of the other possible outcomes. Once again, we see the notion of "possible worlds", but in this case the possibility relation is different than regarding strategyproofness. When modelling strategyproofness, it is natural for an agent to perceive another world as possible if he can get there by lying about his preference. When modelling dictatorship, however, the agent needs to reflect on all possible outcomes. Finding a good definition for what a possible world should be, is one of the challenges one faces when making a modal logic for voting theory.

We see that the Gibbard-Satterthwaite theorem expresses the interaction between a local property, strategyproofness, and a more global one, dictatorship, making it an interesting use case for a modal logic.

There has been previous work done on modal logic and voting theory. This thesis is strongly influenced by two papers [17, 18], both modelling voting theory with modal logics.

To the author's knowledge, the first of these papers is the first publication on a modal logic designed to express properties of social welfare functions (see Section 1.1 on the difference between social welfare functions and social choice functions). The language is syntactically easy, yet still able to express interesting properties. It is, in fact, strong enough to express Arrow's impossibility theorem. The language is simple, but has some "non-modal" features. Mainly it consists of three levels of languages, the lowest being a propositional Boolean logic with a limited set of propositional letters. The two next levels introduce one box each, with no stacking at each level. Consequently, each formula in the final language has a modal depth of exactly two. Every formula is essentially a Boolean combination of quantification over profiles and pairs of alternatives.

It is interesting that this language is able to express rather complicated properties, and as such it finds a simple shape of quantification, which is still quite expressive. But it also has some disadvantages. The approach makes the language non-elegant, and in distancing itself from general themes in modal logic it becomes harder to transfer results to it from other logics. This is one of the motivations of the author of this thesis to try to find a more "modal" language capable of expressing similar notions.

In [18] the focus is shifted towards judgement aggregation, in which the agents no longer submit linear orders over alternatives, but submit no/yes judgements to a set of propositions. In this logic, the formulae are evaluated in triples, consisting of a judgement aggregation rule, a judgement profile, and an agenda item. The language contains a special symbol per agenda item, a special symbol per agent, and a special symbol meaning that the agenda item is in the judgement of the aggregation rule on the judgement profile. The language also contains two boxes: one changing the judgement profile and another changing the agenda item. One of its features is that it parametrises the language not only on the agents, but on the agenda (alternatives) as well. This means that extending results proved in this logic to more general cases requires extra work, and the logic cannot capture properties of infinite sets of alternatives. In this thesis we will have as a goal not to include the set of alternatives into the language, giving it expressivity over arbitrary set of alternatives.

There is also work being done at using other logics than modal logic, [9] uses first-order logic to express properties of voting systems, in this case social welfare functions. They are able to express Arrow's theorem in first-order logic, and start working towards a fully-automated proof.

In making a modal logic for social choice functions, or voting theory in general, we need to pose several questions. First, in making the language, we must decide what to include in it and what to abstract over. Most of the aforementioned approaches parametrise the language on the agents, some on the alternatives. The disadvantage here is that results proved inside these logics will only hold for the special parameters with which they are instantiated. The advantage, however, is that it is easier to express certain properties. Without having the agents as a part of the language, it becomes quite difficult to enforce that the number of agents are finite, a requirement for the Gibbard-Satterthwaite theorem. Secondly, in making a language, we must also decide on the essential properties to quantify over and on how this quantification will take place. In a modal logic, it is reasonable to encode this, in some way, as relations between different states, where each state represents a possible value quantified over.

The main goal of this thesis is to find a "reasonable" modal language capable of expressing the Gibbard-Satterthwaite theorem. We keep the set of alternatives out of the language to make it easier to generalise results. We want to allow for the arbitrary stacking of boxes; this makes the language more "modal", thus easing the transfer of results. Finally, we aim toward constructing a decidable logic.

We adhere to formal minimalism as a design principle. We will also investigate the relationship between minimalism and expressive power, trying to strike a good balance.

The remainder of thesis is organised as follows. In Chapter 2, we define the basic concepts from both voting theory and modal logic. We will also introduce ATL, which we will see again in Chapter 5, and some complexity theory which will form the background of a model-checking section. In Chapter 3, we present the main contribution of this thesis. We start by defining models defined by SCFs and prove that they capture the class of SCFs adequately. We then proceed with the language $\mathcal{L}^{a,b}_{n,U}$ and the logic $\mathbf{VL}_{n,U}$, and we see how the language is capable of expressing properties of SCFs.

Chapter 4 explores a long range of meta logical results. This involves exploring a sub-language of $\mathcal{L}^{a,b}_{n,U}$, $\mathcal{L}^{a,b}_{n}$, and proving several results about this, some which carry over to $\mathcal{L}^{a,b}_{n,U}$. We prove both undecidability, decidability, completeness and incompleteness[1] for $\mathcal{L}^{a,b}_{n}$, depending on the number of agents. We also take a look at the complexity of model-checking of $\mathcal{L}^{a,b}_{n,U}$. While exploring an even smaller language than $\mathcal{L}^{a,b}_{n}$, we discover a general result giving an "unstacked" normal form for $\mathbf{S5}^n$. In Chapter 5 we describe a method of merging ATL and CGS with $\mathcal{L}^{a,b}_{n,U}$, enabling us to talk about properties of certain families of SCFs. We extend this to work for languages other than $\mathcal{L}^{a,b}_{n,U}$, providing a general framework. We proceed to see how we can use ATL and CGS without any dedicated logic for SCFs to express properties of a single SCF.

We reflect on future work in Chapter 6. We look at how we could extend $\mathcal{L}^{a,b}_{n,U}$ to be more expressive, making it a more useful tool in model checking. We will pose several open questions related to both $\mathcal{L}^{a,b}_{n}$ and $\mathcal{L}^{a,b}_{n,U}$. Finally we conclude with a review of the progress made in this thesis.

[1]The incompleteness is relative to a specific deduction system.

Chapter 2

Background

> "Read the directions and directly you will be directed in the right direction."
> —*Doorknob, Alice in Wonderland*

In this chapter we will introduce all the basic concepts needed for this thesis. We define concepts from voting theory, basic modal logic, ATL and computational complexity.

Most of the definitions and notions are standard in their fields, but even the well-versed logician or voting theorist should read this chapter to establish notation. It will have a more technical character than the previous chapter.

2.1 Voting theory

In this section we will define the concepts we need from voting theory. We will start by reminding the reader what social choice functions are. They are functions taking one preference order over the alternatives for each agent, and returning one alternative. As the reader might notice we are now talking about "agents" and not "voters". We choose this terminology as it fits better with the usual terminology used in modal logic. We will call a set of agents a "coalition".

We fix the following notation.

- We are given a finite set of agents $I = \{1, \ldots, n\}$, and a non-empty set of alternatives A.
- By $L(A)$ we denote the set of all linear orders over A, and by $L(A)^n$ the set of all n-tuples of linear orders.
- An element of $L(A)^n$ will be called a profile.

Definition 2.1. A *linear order* is a binary relation $\leq$ on A which is antisymmetric, transitive and total. Formally, for all $a, b, c \in A$:

- If $a \leq b$ and $b \leq a$ then $a = b$ (antisymmetry).
- If $a \leq b$ and $b \leq c$ then $a \leq c$ (transitivity).
- $a \leq b$ or $b \leq a$ (totality).

We introduce the following notation, allowing us to pick out a specific linear order for each agent from a profile.

Notation 2.2. For $D \in L(A)^n$ we will, by D_i, denote the linear order of agent i in D. We denote that i orders x less than or equal to y in D by xD_iy.

When we have several profiles we will often call them D and P. If $C \subseteq I$ is a coalition of agents such that $|C| \leq n$ we will often write D_C for a $|C|$-tuple of linear orders, and D_{-C} for a $n - |C|$-tuple. We will generally assume that the agents are named $1, \ldots, n$ where $n = |I|$.

We define the binary relation $\sim_i$ on $L(A)^n$ in the following way:

Definition 2.3 ($\sim_i$)**.** $\sim_i$ is a binary relation on $L(A)^n$, such that for all $D, P \in L(A)^n$ we have $D \sim_i P$ if and only iff $D_j = P_j$ for all $j \neq i$, $j \leq n$.

In other words, $\sim_i$ relates two profiles that differ only in the preference for agent i. Similarly we let $\sim_{-i}$ on $L(A)^n$ be such that $P \sim_{-i} P'$ if and only if $P_i = P'_i$, so it relates two profiles that might differ in any other preferences than agent i's. In the definition of $\sim_i$ the number of agents n is implicit, as it is always clear from the context which n it is.

We define social choice functions as follows.

Definition 2.4 (SCF)**.** A social choice function (SCF) F is a total function of the form $F\colon L(A)^n \to A$.

There are a couple of things to notice about this definition. First, we will only concern ourselves with finite numbers of agents. The set of alternatives on the other hand can be countably infinite. Also note that the requirement for something to be a social choice function concerns only its domain and co-domain. Specifically, there is no requirement that it is “fair”, “reasonable”, or has any other particular “social” features.

Definition 2.5 (Winning and possibly-winning)**.** For an SCF $F\colon L(A)^n \to A$, profile $D \in L(A)^n$ and alternative $a \in A$, a is *winning* if $F(D) = a$, and a is possibly-winning of there is some $P \in L(A)^n$ s.t. $F(P) = a$.

Let us also set up some conventions for later. As an SCF F is defined as a total function from $L(A)^n$, we can assume that if we have F then we have both n and A. Consequently, when needed, we will assume that we have both A and n whenever we are referring to a SCF F. We will also need to talk about all social choice functions on A and n.

Definition 2.6 ($\mathbf{SCF}(A, n)$)**.** When A is a set and n a natural number, $\mathbf{SCF}(A, n)$ is the set of all total functions of the form $L(A)^n \to A$. $\mathbf{SCF}(n)$ is the union of $\mathbf{SCF}(A, n)$ for all countable sets A[1], and $\mathbf{SCF}$ is the union of $\mathbf{SCF}(n)$ for all $n \in \mathbb{N}$.

2.1.1 Properties of SCFs

Given a number of agents n and some set of alternatives A, we can now talk about the different properties which different social choice functions $F\colon L(A)^n \to$

[1] The requirement of countability is just to avoid making **SCF** a class.

A may have. We will define the three properties which are important for the Gibbard-Satterthwaite theorem, and which we will later model using a modal logic.

Recall that the image of a function $F\colon L(A)^n \to A$ is exactly the set $F[L(A)^n] = \{F(D) \mid D \in L(A)^n\}$.

Definition 2.7 (k-winning SCF)**.** An SCF F is k-winning when the image of F has at least k elements, i.e. $|\mathrm{Img}(F)| \geq k$.

Observe that our definition of an SCF does not require it to be onto.[2] Consequently, a function can avoid being k-winning even if A is greater than k.

Definition 2.8 (Strategyproof)**.** An SCF F is strategyproof (SP) if for all agents i, profiles P it holds that $F(P')P_iF(P)$ for all $P \sim_i P'$.

In other words, a social choice function is strategyproof if an agent never has an incentive to lie about her preference.

Finally we define when a function is i-dictatorial. The basic idea is that there is some agent i who gets to choose what element will be the winner. When we try to formalise this idea, we run into some issues. One concerns how the dictator gets to choose his element, and the other one concerns the limit of his power. In general, the linear order does not need to have a top element, so it does not help to say that the top element of the dictator's preferences must be the chosen item. Furthermore, even if one concerns oneself with bounded linear orders, the problem is not solved if the SCF is not onto. Another problem concerns deciding exactly how much power the dictator should have. One way of formalising dictatorship, as Gibbard[8] did originally, is that for every alternative x, the dictator has some linear order such that, no matter what the other agents do, x will be the winner. This is elegant as it does not depend in any way on the actual linear orders provided. The disadvantage, however, is that it implies that the SCF is onto. Another consequence of this definition is that the *anti-dictator* —the agent whose **least** preferred item always wins—becomes a dictator. One way of handling this is, instead, to say that some agent i is a dictator if the other agents' ballots do not matter. For any profile, then, if you change the ballots of every agent other than i, the outcome is still the same. One consequence of this definition is that it makes constant functions dictatorial for all agents.

Tweaking the "top element" method provides us with another way of formalising the notion of dictatorship. For instance, we may say that, if the linear order provides some top element in the image of the SCF, this element must be the winning alternative. This is the formalisation that we will be using; it is also the definition used in [4].

Definition 2.9 (i-dictatorial)**.** An agent i is dictatorial for an SCF $F\colon L(A)^n \to A$ when $\forall D, P \in L(A)^n : F(P)D_iF(D)$.

In other words, an agent i is dictatorial for an SCF $F\colon L(A)^n \to A$ if, for every profile $D \in L(A)^n$, there is no other profile $P \in L(A)^n$ such that agent i prefers $F(P)$ to $F(D)$ in D. We will now give some examples which illuminate the differences between the competing definitions of dictatorship. Imagine there are two agents, $I = \{1, 2\}, n = 2$, giving linear orders over how

[2] f is onto, or surjective, if the co-domain is equal to the image.

much money/utility agent 1 will receive. Their set of alternatives is $\mathbb{N}$, and P_1 prefers getting as much money as possible. So his ordering over the set of alternatives is the usual ordering over $\mathbb{N}$, with no top element. Assume some SCF F such that, for every natural number n, the agent i can submit a linear order such that n becomes the outcome. Then he is a dictator by Gibbard's original definition, but not by ours. On the other hand, assume some SCF F' such that the winner is always the number in the range $1 \dots 10$ which agent i has ranks highest. This makes i a dictator by our definition, but not by Gibbard's.

Definition 2.10 (Dictatorial)**.** A SCF $F \colon L(A)^n \to A$ is dictatorial when is i-dictatorial for one of the agents $i \leq n$.

We can now state the classical result of Gibbard and Satterthwaite:

Theorem 2.11 (Gibbard-Satterthwaite Theorem)**.** *For any social choice function F, if F is 3-winning and strategyproof, then it is dictatorial.*

This was originally proved independently by Gibbard [8] and Satterthwaite [14]. Contrary to the original proofs, many newer proofs of the Gibbard-Satterthwaite theorem [15, 13] demand that the SCF is onto (and that there are more than 3 alternatives), as this makes for elegant proofs. As our definition of dictatorship is slightly different from the ones in Gibbard's and Satterthwaite's original proofs, we refer to [4] for a proof that is compatible with our definition of dictatorship.

2.2 Modal logic

In this section we will introduce general concepts for modal logic. We will go through general modal languages, their interpretations in frames and models, and define what is a logic.

The well versed modal logician will find little new here, besides notational conventions.

See Definition 1.1 for the basic modal language. We often need slightly different languages than the basic modal language, often containing more modalities. To accommodate this we use the a *modal similarity type*, the following definition is from [5].

Definition 2.12 (Modal similarity type)**.** A modal similarity type is a pair $\tau = (O, \rho)$ where O is a nonempty set, and ρ is a function $O \to \mathbb{N}$. The elements of O are called *modal operators*; we use ∇, sometimes subscripted $\nabla_0, \nabla_1, \dots$ to denote elements of O. The function ρ assigns to each operator $\nabla \in O$ a finite *arity*, giving the number of arguments ∇ can be applied to.

To sum up, the modal similarity type gives all the modalities and their arities. As an example the modal similarity type τ of the basic modal language is $O = \{\Box\}$ and $\rho(\Box) = 1$. We will often not distinguish between O and τ, and talk about modal operators in τ.

Having a modal similarity type we can extend it to a modal language in the following way.

Definition 2.13 (Modal language $ML(\tau, \Phi)$)**.** A modal language $ML(\tau, \Phi)$ is built up using a modal similarity type τ as above and a non-empty set of propositional letters Φ. The set $ML(\tau, \Phi)$ of *modal formulae* over τ and Φ is given by the BNF

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \nabla(\varphi_1, \ldots, \varphi_{\rho(\nabla)})$$

where p ranges over elements of Φ.

The reader should note that with the similarity type given above for the basic modal language Definition 2.13 and Definition 1.1 defines the same language. There are a couple of standard abbreviations that we will use, and these are as follows

Definition 2.14 (Standard syntactic abbreviations)**.**

$$\begin{aligned} \varphi \wedge \psi &\stackrel{def}{=} \neg(\neg\varphi \vee \neg\psi) \\ \varphi \to \psi &\stackrel{def}{=} \neg\varphi \vee \psi \\ \Diamond\varphi &\stackrel{def}{=} \neg\Box\neg\varphi \end{aligned}$$

We assume that there is a fixed countable infinite set of propositional letters denoted by P, and unless otherwise noticed our modal languages are defined on this. This compared to Φ, which is used as a letter over sets of propositional letters.

We also need the notion of subformula, defined inductively on formulae in $ML(\tau, \Phi)$.

Definition 2.15 (Sub(φ))**.**

$$\begin{aligned} \text{Sub}(p) =& \{p\} \\ \text{Sub}(\neg\varphi) =& \{\neg\varphi\} \cup \text{Sub}(\varphi) \\ \text{Sub}(\varphi \vee \psi) =& \{\varphi \vee \psi\} \cup \text{Sub}(\varphi) \cup Sub(\psi) \\ \text{Sub}(\nabla(\varphi_1, \ldots, \varphi_{\rho(\nabla)})) =& \{\nabla(\varphi_1, \ldots, \varphi_{\rho(\nabla)})\} \cup \bigcup_{1 \leq i \leq \rho(\nabla)} (\text{Sub}(\varphi_i)) \end{aligned}$$

We will mostly work inside a fragment of the modal languages where we have one box modality per agent, and possibly a restricted set of propositional letters. We will therefore define the following special similarity type, and then define notations denoting modal languages that will be of interest to us.

Definition 2.16 (The τ_n and $\tau_{n,U}$ similarity type)**.** We define $\tau_n = (O_n, \rho^1)$ to be such that $O_n = \{\Box_1, \ldots, \Box_n\}$ and $\rho^1(\Box_i) = 1$ for $1 \leq i \leq n$, that is the signature of n unary boxes. And $\tau_{n,U} = (O_{n,U}, \rho^1)$ such that $O_{n,U} = \{\Box_1, \ldots, \Box_n, \Box^U\}$, that is, τ_n with an extra unary box $\Box^U$.

Definition 2.17 ($\mathcal{L}_n^\Phi$)**.** We let $\mathcal{L}_n^\Phi \stackrel{def}{=} ML(\tau_n, \Phi)$ be the modal language of n unary boxes built on the propositional letters in Φ. We also let $\mathcal{L}_n \stackrel{def}{=} \mathcal{L}_n^{\mathsf{P}}$ be the language built on the countable set of standard propositional letters. With $\mathcal{L}_n^{a,b}$ we denote the language $ML(\tau_n, \Phi_n^{a,b})$ with $\Phi_n^{a,b} = \bigcup_{i \leq n}\{a_i, b_i\}$, a finite number of propositional letters distinct from P.

Modal formulae are evaluated in models, and models are built on frames which we will now define.

Definition 2.18 (τ-frame)**.** Given a modal similarity type $\tau = (O, \rho)$, a τ-frame is a tuple $\mathcal{F} = (W, R_1, \ldots, R_{\nabla_{|O|}}$, where

- W is a non empty domain, and
- for each $n \geq 0$, and each n-ary modal operator ∇ in τ a $(n+1)$-ary relation R_∇.

That is, a frame is a relational structure $\mathcal{F} = \{W, R_{\nabla_1}, \ldots, R_{\nabla_{|O|}}\}$, with a domain and a number of relations over the domain. The relations will often be called "accessibility relations". In our case we will as noted usually talk about unary modalities, so in which case they correspond to binary relations on the domain. In this case we can view the frame as a directed graph with multiple edge sets. When we have finite number n of unary relations (that is, if $\tau = \tau_n$) we usually just write $\mathcal{F} = \{W, R_1, \ldots, R_n\}$.

On top of a frame we build models in the following fashion.

Definition 2.19 ((τ, Φ)-models)**.** Given a τ frame $\mathcal{F}$ and a set of propositional letters Φ we can extend it to a model $\mathcal{M} = (\mathcal{F}, V)$ by providing a *valuation function* $V \colon \Phi \mapsto \mathcal{P}(W)$ which assign each propositional letter to a set of states in the frame.

We will sometimes denote a model as just $\mathcal{M}$ and not $\mathcal{M} = (\mathcal{F}, V)$, and still talk about the underlying frame $\mathcal{F}$ and valuation function V when it is clear what model they are from.

The valuation function maps each propositional letter to a set of states. Alternatively we can see it as a function labelling each state with propositional letters. We will sometimes need $V^\vee \colon W \mapsto \mathcal{P}(\Phi)$ induced by some $V \colon \Phi \mapsto \mathcal{P}(W)$ in the following way: $V^\vee(w) = \{p \mid w \in V(p)\}$. Sometimes we will use $V^\vee$ without explicitly mentioning it by talking about "propositional letters holding in w" for some $w \in W$.

If we have a frame $\mathcal{F}$ we will sometimes talk about "all models on $\mathcal{F}$". With this we mean all possible models that are constructed on the frame $\mathcal{F}$.

Definition 2.20 (Pointed (τ, Φ)-models)**.** Given a (τ, Φ)-model $\mathcal{M}$, a pointed model $(\mathcal{M}, w)$ is a pair consisting of the model $\mathcal{M}$ and a state $w \in W$. When talking about all pointed models on a model $\mathcal{M}$ we talk about the set $\{(M, w) \mid w \in W\}$, that is all pointed models that can be made on the domain of the model.

We will several places in Chapter 4 need to construct a new model from an old one, and a useful method for this is to generate a submodel.

Definition 2.21 ($R_1, \ldots, R_n$-generated submodel)**.** Let $\mathcal{M} = (W, R_1, \ldots, R_m, V)$ be a τ_m-model. When $n \leq m$ and $w \in W$, we can form the $R_1, \ldots, R_n$ generated submodel of $(\mathcal{M}, w)$ as

$$\mathcal{M}_{|R_1,\ldots,R_n} = (W_{|R_1,\ldots,R_n}, R'_1, \ldots, R'_n, V'),$$

where

- $w \in W_{|R_1,\dots,R_n}$
- if $u \in W_{|R_1,\dots,R_n}$ and $R_i uv$ then $v \in W_{|R_1,\dots,R_n}$ for all $i \leq n$ and
- $R'_i = R_i \cap (W_{|R_1,\dots,R_n} \times W_{|R_1,\dots,R_n})$ for all $i \leq n$.

The valuation V' is the restriction of V to $W_{|R_1,\dots,R_n}$, that is, for all $p \in \mathsf{P}$, $V'(p) = V(p) \cap W_{|R_1,\dots,R_n}$. We then call $(\mathcal{M}_{|R_1,\dots,R_n}, w)$ the *pointed* $R_1, \dots, R_n$ *generated submodel* of $(\mathcal{M}, w)$.

Definition 2.22 ($\mathbf{Mod}(\tau, \Phi)$ and $\mathbf{Mod}^p(\tau, \Phi)$)**.** Given a modal similarity type τ and a set of propositional letters Φ, $\mathbf{Mod}(\tau, \Phi)$ is the class of all (τ, Φ)-models, and $\mathbf{Mod}^p(\tau, \Phi)$ is the class of all pointed (τ, Φ)-models. $\mathbf{Mod}(\tau)$ and $\mathbf{Mod}^p(\tau)$ is the same, but with $\Phi = \mathsf{P}$.

Definition 2.23 ($\mathbf{Mod}(C, \Phi)$ and $\mathbf{Mod}^p(C, \Phi)$)**.** If C is a class of τ-frames and Φ is a set of propositional letters, then $\mathbf{Mod}(C, \Phi)$ is the class of all (τ, Φ)-models built on frames in C, and $\mathbf{Mod}^p(C, \Phi)$ is the class of all pointed (τ, Φ)-models built on frames in C. Again we use $\mathbf{Mod}(C)$ and $\mathbf{Mod}^p(C)$ for the case when $\Phi = \mathsf{P}$.

A formula φ in the modal language $ML(\tau, \Phi)$ is evaluated in a pointed (τ, Φ)-model $\mathcal{M}, w$ in the following way.

Definition 2.24 (Truth of $ML(\tau, \Phi)$ formulae in a pointed (τ, Φ)-model)**.**

$$\begin{array}{ll}
\mathcal{M}, w \models p & \text{iff } w \in V(p) \\
\mathcal{M}, w \models \neg\varphi & \text{iff not } \mathcal{M}, w \models \varphi \\
\mathcal{M}, w \models \varphi \vee \psi & \text{iff } \mathcal{M}, w \models \varphi \text{ or } \mathcal{M}, w \models \psi \\
\mathcal{M}, w \models \nabla(\varphi_1, \dots, \varphi_n) & \text{iff for all } v_1, \dots, v_n \text{ with } R_\nabla w v_1, \dots, v_n \\
& \quad \text{we have, for each i, } \mathcal{M}, v_i \models \varphi_i
\end{array}$$

We will use $\mathcal{M}, w \not\models \varphi$ to mean that it is not the case that $\mathcal{M}, w \models \varphi$. As is common in the literature, we extend the use of the symbol $\models$ to denote several other relations, depending on its context:

$$\begin{array}{rl}
\mathcal{M} \models \varphi & \text{iff } \mathcal{M}, w \models \varphi \text{ for all } w \in W, \text{ and} \\
\mathcal{F} \models \varphi & \text{iff } \mathcal{M} \models \varphi \text{ for all models } \mathcal{M} \text{ constructed from } \mathcal{F}
\end{array}$$

When we have $\mathcal{M} \models \varphi$ we say that φ is globally true in $\mathcal{M}$, or a validity in $\mathcal{M}$. When we have $\mathcal{F} \models \varphi$ we say that φ is valid in $\mathcal{F}$. If Λ is a set of formulae $\mathcal{F} \models \Lambda$ denotes that all formulae in Λ are validities in $\mathcal{F}$. Furthermore, when C is a class of frames and Λ is a set of formulae, then $C \models \Lambda$ means that $\mathcal{F} \models \Lambda$ for all $\mathcal{F} \in C$. Often we will use the notation $\models \varphi$ to denote that $C \models \varphi$ for a default class of frames C.

When we have $\mathcal{M}, w \models \varphi$ we say that M, w satisfies φ. $\mathcal{M}$ satisfies φ if $\mathcal{M}, w \models \varphi$ for some $w \in W$. In the same manner we say that a frame $\mathcal{F}$ satisfies φ if some models on $\mathcal{F}$ satisfies φ. And when C is a class of frames we say that it satisfies φ if some frame $\mathcal{F} \in C$ satisfies it.

Sometimes we will say that a formula φ *defines* a certain property. With this we will mean that all models that have this property validates the formula, and all models that validates the formula have this property.

Definition 2.25 (A logic)**.** A logic is a set of formula in a formal language.

If Λ is a logic, we will by a "Λ-formula φ" mean a $\varphi \in \Lambda$. A logic can be specified in different ways, and we will be focused on two. One way is to fix a language and a class of frames, and let the logic be all the formulae holding on all the frames. That is, if we let C be the class of frames we are interested in we let the logic be the largest set Γ of formulae in our language such that $C \models \Gamma$. This is usually called a *semantically defined* logic. If C is a class of frames $\mathbf{Log}(C)$ is the logic of all the formulae valid in C. It is usually clear from context in what language the logic is in, and by default it is taken to be $\mathcal{L}_m$ when C is a class of τ_m frames, or just $\mathcal{L}_1$ otherwise. Before we continue with an example of a semantically defined logic we define two classes of frames.

Definition 2.26 (The frame-classes EQ and U)**.** Let EQ be the class of τ_1-frames where the relation is an equivalence relation (that is reflexive, symmetric and transitive), and U be the class of τ_1-frames where the relation is the universal relation (that it, it relates all points).

An example of a semantically defined logic is e.g. $\mathbf{Log}(EQ)$, the logic of all $\mathcal{L}_1$ formula valid in EQ.

Another way is to define some deduction system that syntactically specifies a set of formulae of some language. The following deduction system defines the logic **S5** in a syntactic matter.

Definition 2.27 (**S5**)**.** MP: From $\varphi \to \psi$ and φ get ψ

NEC: If φ then $\Box\varphi$

US: From φ prove φ' where φ' is obtained from φ' by uniformly replacing proposition letters in φ by formulae from $\mathcal{L}_1$.

TAUT: The set of all propositional tautologies

K: $\Box(p \to q) \to (\Box p \to \Box q)$

S5: T $\Box p \to p$

4 $\Box p \to \Box\Box p$

B $p \to \Box\Diamond p$

The set **S5** is then formed by taking the formulae above (K,T,4,B and all of TAUT), and then closing it under the rules given (MP, NEC and US). For a further reading we refer the reader to Section 1.6 and 4.1 of [5], or Chapter 1 of [7].

An common question is how the two ways of defining a logic relates to each other, and whether they define the same logic. If we have a syntactically specified logic Λ and a semantically specified logic Γ we say that Λ is *sound* with respect to Γ if $\Lambda \subseteq \Gamma$, and *complete* with respect to Γ if $\Lambda \supseteq \Gamma$.

If we talk about a class of frames C instead of the logic Γ, we take Γ to be $\mathbf{Log}(C)$. When Λ is both sound and complete with respect to $Log(C)$ we say that C *determines* Λ. So saying that the frame class EQ determines the syntactically specified logic **S5** defined in Definition 2.27, means that $\mathbf{S5} = \mathbf{Log}(EQ)$.

When Λ is a logic we say that a formula φ is Λ-consistent if $(\varphi \to \bot) \notin \Lambda$. We will later use the following theorem, proved in e.g. proposition 4.12 in [5].

Theorem 2.28. *A logic Λ is complete with respect to a class of structures S if and only if every Λ-consistent formula is satisfiable in some $\mathcal{M} \in S$.*

It is a standard result, proved e.g. in [5] that this is the case. Another folklore result is that $\mathbf{Log}(EQ) = \mathbf{Log}(U)$, a result that essentially states that $\mathcal{L}_1$ can not distinguish between generally equivalence classes and the universal relation specifically. So to sum up we have the following equation:

$$\mathbf{S5} = \mathbf{Log}(EQ) = \mathbf{Log}(U).$$

2.3 $\mathbf{S5}^m$

In this section we introduce the logic $\mathbf{S5}^m$. The basic definitions can also be found in [7], one of the seminal works in this field, and a major inspiration for many of the later results in this thesis. First we start by defining what the product of frames are in general, and then we use this to define what $\mathbf{S5}^m$ is.

We will only need to take the product of frames with one relation in each(τ_1-frames), and not on general τ-frames. The definition below are for this specific case.

Definition 2.29 (Product frames)**.** The product $\mathcal{F}_1 \times \cdots \times \mathcal{F}_m$ of τ_1-frames $\mathcal{F}_i = (W_i, R_i), i = 1, \ldots, m$, is the τ_m-frame

$$\mathcal{F}_1 \times \cdots \times \mathcal{F}_m \stackrel{def}{=} (W_1 \times \cdots \times W_m, \overline{R}_1, \ldots, \overline{R}_m)$$

where, for each $i = 1, \ldots, m, \overline{R}_i$ is the binary relation on $W_1 \times \cdots \times W_m$ such that

$$(u_1, \ldots, u_m)\overline{R}_i(v_1, \ldots, v_m) \text{ iff } u_i R_i v_i \text{ and } u_k = v_k, \text{ for } k \neq i.$$

That is, the product of m frames of one relation is a m-frame where the domain is the product of all the original domains, and each of the m relations relate two points that (possibly) differ in only that specific coordinate. The following clarifies what we mean with the product of frame classes. To simplify notation we only do products of frames from the same class, but the definition can be extended if needed.

Definition 2.30 (C^m)**.** Given some class of τ_1-frames C we define C^m to be the class of all m-products of frames from C.

Specifically, EQ^m is the class of all m-products of frames from EQ, and U^m is the class of all m-products of frames from U. Elements of U are called universal product frames. We use the new notion to define a semantically defined logic $\mathbf{S5}^m$.

Definition 2.31 ($\mathbf{S5}^m$)**.** $\mathbf{S5}^m \stackrel{def}{=} \mathbf{Log}(EQ^m)$. That is, $\mathbf{S5}^m$ is the set of all $\mathcal{L}_m$ formula valid in all m-product frames $\mathcal{F}_1 \times, \ldots, \times \mathcal{F}_m$ where for each $\mathcal{F}_1, \ldots, \mathcal{F}_m$, the accessibility relation is an equivalence relation.

As EQ is determined by the logic $\mathbf{S5}$, we could also have defined $\mathbf{S5}^m$ to be the logic of m-product frames $\mathcal{F}_1 \times, \ldots, \times \mathcal{F}_m$ where for each $\mathcal{F}_1, \ldots, \mathcal{F}_m$ we have $\mathcal{F}_i \models \mathbf{S5}$. This is a normal way of defining it, and explains the name.

We will later need the following result, which is proved in proposition 3.12 in [7].

Theorem 2.32. $\mathbf{S5}^m$ *is determined by product of frames where the relation is the universal relation on its domain.*

This says that the logic $\mathbf{S5}^m$ is the same as the logic of a much smaller class of frames than in the original definition. It reflects the fact that $\mathbf{Log}(EQ) = \mathbf{Log}(U)$, but shows that this result generalises to product frames. So $\mathcal{L}_m$ is too weak to distinguish between product of universal relations, and general equivalence classes. Summing up we have the following equality:

$$\mathbf{S5}^m \stackrel{def}{=} \mathbf{Log}(EQ^m) = \mathbf{Log}(U^m).$$

We will use the following facts later.

Theorem 2.33. *For the logic* $\mathbf{S5}^m$ *we have that*

- $\mathbf{S5}^2$ *is decidable and has a complete, finite, axiomatization,*
- $\mathbf{S5}^m, m \geq 3$ *is undecidable and lacks the finite model property,*
- $\mathbf{S5}^m, m \geq 3$ *has no finite axiomatization.*

Proof. See [7, p. 379]

We will by $\mathbf{Mod}^p(U^m)$ denote all pointed (τ_m, P)-models built on frames in U^m, and similar with $\mathbf{Mod}^p(E^m)$.

2.4 Alternating-time temporal logic

In the following section we will give basic definitions of alternating-time temporal logic (ATL) and concurrent game systems (CGS). We do this because we will later merge a logic for social choice functions with ATL. First, however, we will learn the basics of ATL and CGS. ATL is evaluated in a CGS, which we will define now.

ATL was introduced by Alur et al. in a series of papers, and over different structures. Here we will be interested in concurrent game structures, introduced in [2]. In this setting, ATL is a logic describing coalitional power.

We will simplify the structures and call them concurrent game systems to separate them from their inspiration.

Definition 2.34 (Concurrent game systems)**.** A CGS is a tuple $S = (I, \mathcal{Q}, \Phi, \pi, \Sigma, \delta)$, where I is a finite set of agents, $\mathcal{Q}$ is a non-empty set of states, Φ is a non-empty set of propositional letters, $\pi : \Phi \to \mathcal{P}(\mathcal{Q})$ is a valuation function, Σ a set of possible actions, and $\delta\colon \Sigma^{|I|} \times \mathcal{Q} \to \mathcal{Q}$ a transition function. A pointed CGS is a pair S, q where S is a CGS, and $q \in \mathcal{Q}$.

There are a few ways in which the above definition differs from the definition of a concurrent game structure given in [2]. We do not demand that $\mathcal{Q}$ or Φ are finite. We also have a global set of actions, instead of separate possible actions for each state in $\mathcal{Q}$. These are simplifications that make the presentation easier and they fit better with the use we will have for these structures later.

The structure consists of a set of states ($\mathcal{Q}$) with valuations on them (π), as usual in modal logic. The major difference is in the representation of the accessibility relation between the states. Instead of having one relation per agent, the accessibility relation is given as a transition function which yields outputs depending on the actions (Σ) of each agent.

The conceptual idea is that, at each point in the graph, the agents play a strategic game; they all pick a action at the same time and the result of the game is the next state. In these models we evaluate formulae in the language ATL_I^Φ.

Definition 2.35 (ATL_I^Φ). ATL_I^Φ is the following language, where I is a finite set of agents and Φ a countable set of propositional letters, and we have $p \in \Phi$ and $C \subseteq I$.

$$\varphi := p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\langle C\rangle\rangle \mathcal{X}\varphi \mid \langle\langle C\rangle\rangle \mathcal{G}\varphi \mid \langle\langle C\rangle\rangle \varphi\mathcal{U}\varphi$$

Before moving on to the formal semantics, we will first elicit some intuitions and give some examples of formulae. Formulae are evaluated in a pointed CGS; that is, in a CGS and a point $q \in \mathcal{Q}$. The truth value of propositional letters depends on the valuation function as usual; the interesting news concerns the modalities. We see that, for each of the three modalities, they exist in one version per subset of agents, called "coalitions". $\langle\langle C\rangle\rangle \mathcal{X}\varphi$ expresses that the coalition C of agents can force that the outcome of the current strategic game becomes a state in which φ holds. $\langle\langle C\rangle\rangle \mathcal{G}\varphi$ expresses that the coalition C can force that φ will always be the case. This is an interesting addition, as it talks not only about the next available state, but about all states reachable as the game continues. The last modality is our first instance of a binary modality, taking two formulae as arguments. It expresses that the coalition C has a strategy such that they can enforce that φ will hold until at some point ψ holds.

Before we continue with the truth definitions we need to define what is meant by a strategy. We start by defining *individual strategies*. Consider a CGS $S = (I, \mathcal{Q}, \Phi, \pi, \Sigma, \delta)$ as in Definition 2.34. A strategy for an agent $i \leq n$ is a function $f_i \colon \mathcal{Q} \to \Sigma$, mapping every state to an action. So a strategy determines for every state what the agent will do next. By F_C, we denote a set of individual strategies for all $i \in C$, and call this a *coalitional strategy*, and by F_{-C} the same for all $i \in I \setminus C$. From two coalitional strategies F_C and F_{-C} we can form one *grand strategy* F, giving individual strategies for every agent in I. A joint action for a coalition C is an element of $\Sigma^{|C|}$, and a joint action is an element in $\Sigma^{|I|}$, giving an action for every agent. A grand strategy determines a joint action for every point $q \in \mathcal{Q}$, and we will write $F(q)$ for this element of $\Sigma^{|I|}$. We say that some state $q' \in \mathcal{Q}$ is compatible with F_C in q when $\delta(F(q), q) = q'$ for all coalitional strategies F_{-C} for the complement-group.

Given some CGS S, a state $q \in \mathcal{Q}$ and a coalitional strategy F_C for some $C \subseteq I$, we define $o(q, F_C)$ to be the set of all states that are compatible with F_C in q. We denote sequences of states as λ, and $\lambda[i]$ is the ith element in the sequence, starting with 0. From o we define the function $out(q, F_C)$ to be the set of all sequences λ of states starting in q which is compatible with F_C. That is, $out(q, F_C)$ contains all $\lambda \in \mathcal{Q}^+$ s.t. $\lambda[0] = q$ and for all $i \in \mathbb{N}$, $\lambda[i+1] \in o(\lambda[i], F_C)$.

The satisfiability conditions are defined as follows, skipping the trivial Boolean ones. We will use the usual abbreviations given in Definition 2.14.

Definition 2.36 (Semantics of ATL)**.**

$S, q \models p$ iff $p \in \pi(q)$

$S, q \models \langle\langle C\rangle\rangle \mathcal{X}\varphi$ iff there exists a collective strategy F_C such that for every sequence $\lambda \in out(q, F_C)$, we have that $S, \lambda[1] \models \varphi$

$S, q \models \langle\langle C\rangle\rangle \mathcal{G}\varphi$ iff there exists a collective strategy F_C such that for every sequence $\lambda \in out(q, F_C)$, we have that $S, \lambda[i] \models \varphi$ for all i

$S, q \models \langle\langle C\rangle\rangle \varphi\mathcal{U}\psi$ iff there exists a collective strategy F_C such that for every sequence $\lambda \in out(q, F_C)$, there exist a position $j \geq 0$ s.t. $S, \lambda[j] \models \psi$, and for all $0 \leq i \leq j$ we have that $S, \lambda[i] \models \varphi$

2.5 Complexity theory

In Section 4.3 we will consider the model checking problem (to be defined there), and in that regard we are interested in its complexity. We will therefore give a short introduction here, just enough to understand that section.

We have already seen some notion of complexity in Section 1.3 about decidability. Few would argue against a claim that an undecidable problem is more complex than a decidable one[3]. Within the class of decidable problems, however, we are interested in drawing distinctions based on the time it takes to compute the solution to a problem. In this section, we begin by giving two examples of problems; the hope, here, is that the reader will intuitively see that they are of different complexities. We will then proceed to the formal definitions.

Definition 2.37 (Max element)**.** Given a finite list L of whole numbers (in $\mathbb{Z}$), find the largest number in the list.

Definition 2.38 (Subset sum)**.** Given a finite list L of whole numbers (in $\mathbb{Z}$), find if any combination of the numbers sums to 0.

If we let n be the size of L, we can now consider how much time these two problems take to solve, depending on n. Max element 1 is "easy" in the sense that one need only to go through the list L one time, and at each step remember the largest number so far. If L gets twice as large, the algorithm must do twice as much. We see that the running-time of the algorithm is linear in n. Subset sum, on the other hand, seems less trivial. The obvious solution is to check every subset of the list L, an algorithm which takes 2^n time to run. Here, adding just one more element to the input list L will make the algorithm take twice as long.

In both of the above problems we measured complexity in terms of the length of the input list L. We generally measure complexity as a function of the size of the input. It is important what we mean by the size of the input. We are referring to the number of bits needed to represent the problem, not the numeric value of the input. As should be clear, when using binary representation, the numeric value grows exponentially with the space needed to represent it. We will meet the importance of representation later as well, when talking about different ways of representing social choice functions.

[3]Note that we call the study of (un)decidability "computability theory", not complexity theory.

That was the intuition; we now move on to more formal matters. We are interested in two important classes of decision problems, **P** and **NP**. P is the class of decision problems that are solvable by some algorithm in time that runs polynomial in the size of the input. This include problem 1 above, but also less obviously "easy" problems like determining whether a number is a prime[1]. The problem of primes is not obvious, exactly because of the way in which we speak of input sizes. If the number was written in unary, it would take as much time reading the number as it was large, and in the same time you could check all numbers below it if it divides it. But since the input is in binary, this means it has size $log(n)$. Thus, for an algorithm to be polynomial, it must be polynomial in the logarithm of the numeric value of the input.

NP is the class of decision problems that are solvable by a non-deterministic Turing machine in polynomial time. A non deterministic machine can be seen as a machine which, every time it needs to make a decision in its algorithm, chooses all options at the same time (or alternatively, it always guesses right). There is an alternative description of this class: NP is the class of problems such that, if one proposes a solution, the solution can be verified in polynomial time (polynomial in the size of the problem instance). As an example, consider problem 2 above. If someone gives you a subset of the numbers of L, you can, in polynomial time, check if it sums to 0; so the problem is in NP.

For both of the classes (and complexity classes in general) we use the notion of a P/NP - *hard* problem. An NP-hard problem is a problem such that any problem in NP can be reduced to this problem in polynomial time[4]. Here, "reduced" means that solutions are transferable back. Problems which are both in NP and are NP-hard, are called NP-*complete* problems. Problem 2 above is NP-complete, as is the **SAT** problem. We will meet this problem again in Section 4.3, so I will define it now.

Definition 2.39 (SAT)**.** Given a formula φ in propositional logic, is there an assignment to its propositional letters such that it makes φ true.

Propositional logic uses the same language as basic modal logic (Definition 1.1), except it does not have the $\Box\varphi$. The semantics for the remaining language are the same.

If we have a problem and wish to show it is NP-hard, we may proceed as follows. We take a problem already known to be NP-hard (like SAT) and find a way to translate any instance of that problem into some instance of our problem in polynomial time. The translation needs to be such that the original NP-hard problem has a solution if and only if the new problem has a solution. Now, any problem in NP can be translated to the original problem, and then translated again to our problem. As solutions are preserved, our problem is NP-hard as well. We will use this technique later.

[4]It should be noted that NP-hard problems does not need to be in NP themselves.

Chapter 3

$\mathbf{VL}_{n,U}$

"And since it is beautiful, it is truly useful."
—*The Little Prince*

In this version chapter we will build a logic on top of the (A, n) models and see how we can express the Gibbard-Satterthwaite theorem in this logic. In preparation for this, we shall first establish the degree to which the class of (A, n) models reflects social choice functions.

3.1 (A, n) frames and models

In this section we will define (A, n) frames and models, which will later serve as the basis of the modal logic expressing properties of social choice functions. The structure depends on a set of alternatives A and a finite set of agents I with cardinality n. The structure is a τ_{n+1}-frame which consists of a set W, the domain, and $n + 1$ binary relations.

Formally, we define an (A, n) frame in the following way, where $i \in I$ is an agent:

Definition 3.1 ((A, n)-frame)**.**

$$\begin{aligned} W &= L(A)^n \times L(A)^n \\ R_i &= \{((D, P), (D', P')) \mid D = D' \;\&\; P' \sim_i P\} \\ R_U &= \{((D, P), (D', P')) \mid D' = P'\} \end{aligned}$$

Notice that this is a standard $\tau_{n,U}$-frame with the set of points being W, one relation per agent, and the relation U relating every point to all the points with identical first and second coordinates. The R_i relation relates a point to all the other points sharing the first coordinate, and having the second such that it differs only in agent i's linear order.

On top of the frame we construct a $(\tau, \Phi_n^{a,b})$ model from a social choice function $F\colon L(A)^n \to A$ by letting it induce the valuation function in the following way:

Definition 3.2 ((A, n)-models)**.** Given an (A, n)-frame and an SCF $F : L(A)^n \to A$ we define $V : \Phi_n^{a,b} \to \mathcal{P}(W)$ in the following way for all $(D, P) \in W$.

$$(D, P) \in V(a_i) \text{ iff } F(P) D_i F(D).$$
$$(D, P) \in V(b_i) \text{ iff } F(D) D_i F(P).$$

Thus, some pair of profiles are in $V(a_i)$ exactly when the winner of the first profile is above-or-equal to the winner in the second profile for agent i in the first profile. Or, in the language of voting theory, in the first profile, the agent i prefers the winner in the first profile to the winner in the second profile. Symmetrical for b_i.

It is clear that, given a social choice function F, we can uniquely construct a model as above, since F determines both the frame and the valuation. We make this clear by defining the function $\mathbf{M}$, converting social choice functions into its induced model.

Definition 3.3 ($\mathbf{M}$)**.** Given an SCF $F : L(A)^n \to A$, $\mathbf{M}(F)$ returns the (A, n) model it induces.

We also use the following to pinpoint all the *pointed* models

Definition 3.4 ($\mathbf{M}^p$)**.** $\mathbf{M}^p(F)$ is all pointed versions of the model $\mathbf{M}(F)$.

We overload the notation to also work on sets of social choice functions by returning the set of individual applications: $\mathbf{M}(\Phi) = \{\mathbf{M}(F) \mid F \in \Phi\}$, and similar for $\mathbf{M}^p(\Phi)$.

Some examples of uses are $\mathbf{M}(\mathbf{SCF})$, the set of all models built by some SCF, $\mathbf{M}^p(\mathbf{SCF}(n))$, the set of all pointed models built on social choice functions of n agents, and $\mathbf{M}(\mathbf{SCF}(A, n))$, the set of all models built on social choice functions on the set of alternatives A and with n agents.

Note that $\mathbf{M}(\mathbf{SCF}(n))$ is a strict subset of all $(\tau_{n,U}, \Phi_n^{a,b})$-models that can be built on the same frame, as there are many valuations that are not induced by any SCF F. One example is that, in our models, every point must be a member of $V(b_i)$ or $V(a_i)$ for every agent i. This holds due to our assumption that the models are constructed from some SCF and all SCFs are total.

3.2 Model theory

We have above defined models induced by SCFs. These models have properties, both by them self and in combination with a language. The most prominent question about the models is which social choice functions that induce the same models. The interesting thing about these models is that even if we add new, powerful modalities, we will still not be able to distinguish between these models. This is in contrast to social choice functions that give logically equivalent models only relative to some language. That is models that are not the same, but which the language can not distinguish. In these cases one might still hope to find a more powerful language that can distinguish between these model.

We will below examine the former question, which social choice functions that induce the same models.

Theorem 3.5 (A, n adequacy)**.** *For all non-empty sets A and $n \in N$, any two different social choice functions $F : L(A)^n \to A$ and $F' : L(A)^n \to A$ will induce the same model if and only if they are both constant.*

Proof. Note that two social choice functions F and F' of the same signature induce the same frame, so the only thing that can separate them is the valuation functions they induce. Denote by V^F the valuation function induced by the social choice function F, and likewise $V^{F'}$ for F'.

($\Leftarrow$) Given an arbitrary constant SCF F it will for any profiles D and P induce the same winner in D as in P. So for all points (D,P) we will have that $F(P)D_iF(D)$ and $F(D)D_iF(P)$. So we will have that $V^F(a_i) = L(A)^n \times L(A)^n$ and $V^F(b_i) = L(A)^n \times L(A)^n$. As this was an arbitrary constant function, this holds for them all. And as all there is to separate two different models built on the same A and n is the valuation of a_i and b_i in the points, we are done.

($\Rightarrow$) Assume some set A and $n \in \mathbb{N}$. Assume two different SCFs $F\colon L(A)^n \to A$ and $F'\colon L(A)^n \to A$ such that $\mathbf{M}(F) = \mathbf{M}(F')$. As they are different there is some profile P s.t. $F(P) \neq F'(P)$. Denote $F(P) = p$ and $F'(P) = p'$. We will use the following claim, proved below.

Claim 3.6. *All profiles D that for some $i \leq n$ have p on top of D_i and p' on bottom of D_i must have $F(D) = p$ and $F'(D) = p'$.*

Following we will assume that D is as above. Notice that as $L(A)$ is the set of all linear orders, there must be some linear order as needed above, with p on top and p' on bottom. Also notice that our models consist of all pairs over $L(A)^n$, so there are points with profiles like D both as their first and second coordinate.

Assume towards a contradiction that there is some other profile Q such that $F(Q) = q \neq p$. Note that for (D,Q) we must then have that $(D,Q) \in V^F(a_i)$ and $(D,Q) \notin V^F(b_i)$ (as if we have both then we get that $F(Q) = p$). But also note that under F', as $F'(D) = p'$, which is at the bottom of D_i we must have $(D,Q) \in V^{F'}(b_i)$. But we assumed that F and F' induced the same model, so we have a contradiction. Wee that we can do the symmetric argument if we assume that $F'(Q) = q \neq p$. So the only way for both functions to induce the same valuation is if $F(Q) = p$ and $F'(Q) = p'$. And as Q was arbitrary we get that they are both constant functions. ∎

Proof of Claim 3.6. Pick any such profile D, and we will focus on the point (D,P), where P is as above, a point such that $F(P) \neq F'(P)$. Assume towards a contradiction that $F'(D) = q' \neq p'$. As $F'(P) = p'$, and p' is on the bottom of D_i that means that $(D,P) \in V^{F'}(a_i)$ but also $(D,P) \notin V^{F'}(b_i)$. But the only way F can induce the same valuation is if it can give a winner in D that is above p in D_i, but this is not possible as p was on top. So our assumption is wrong, and $F'(D) = p'$. Also see that we can do the symmetric argument to show that $F(D) = p$. ∎

3.3 $\mathcal{L}^{a,b}_{n,U}$ and $\mathbf{VL}_{n,U}$

Having defined adequate models constructed by social choice functions, we need to define a logic on top. To this end, we will soon define the language we need. After this we give the semantics of the language; that is, how to interpret a formula in a model. The set of all formulae that are true in all models representing social choice functions will form the logic $\mathbf{VL}_{n,U}$.

We will now explain the basic idea of the logic, making it easier to grasp the intuition behind the formal definitions. The inspiration is taken from the properties related to the Gibbard-Satterthwaite, and especially strategyproofness. Strategyproofness, defined in Definition 2.8, concerns whether agents can be better off lying. More formally, it asks whether there are profiles such that some agent prefers to submit a linear order other than the one to which he actually adheres. It is clear that we will need some way to model agents' "real" preferences, the preferences they submit, and how they interact. We also need to give the agents the power to change their ballots. We can do this by evaluating the formulae in pairs of profiles, where one is interpreted as the "real" world/preferences, while the other coordinate is the profile the agents can submit by lying in a certain way. We add one box per agent, reaching all those profiles that differ only in his ballot. We then add two primitives, describing which of the possibilities the agent prefers, the real of the fake world. In addition we notice that we are interested in global properties of our model. That is, we are interested in saying that if some formula holds everywhere, then some other formula holds everywhere. We therefore add a semi-global modality to our language, enabling us to talk about properties of the whole social choice function. The modality is semi-global in the sense that it only reaches all points where the agents are truthful about their preferences.

Below are two examples of formulae and their intended meanings. Formal syntax and semantics follows.

$$\Box_1 a_1 \tag{3.1}$$

Formula 3.1 says that agent 1 prefers the winner in the actual world above all alternatives that can win if only he lies.

$$\neg\Diamond_1 b_1 \wedge \Diamond_2\Diamond_3(b_1 \wedge \neg a_1) \tag{3.2}$$

Formula 3.2 say that agent 1 cannot lie about his preference to get some alternative he prefers, but if agent 2 and 3 lie together, they can force a winning alternative such that agent 1 strictly prefers it to his "honest" winning alternative. We will later consider more example formulae and see what the language can express once we have gone through the formal semantics. The curious reader may skip ahead to Section 3.3.3 for more examples of properties $\mathcal{L}^{a,b}_{n,U}$ can express.

3.3.1 Syntax

Given some finite set of agents $I = \{1, \ldots, n\}$, we define the language $\mathcal{L}^{a,b}_{n,U}$ by the following BNF, with $1 \leq i \leq n$.

$$\varphi ::= a_i \mid b_i \mid \neg\varphi \mid \varphi \wedge \varphi \mid \Box_i\varphi \mid \Box^U\varphi$$

Notice that this language is $ML(\tau_{n,U}, \Phi^{a,b}_n)$, we use $\mathcal{L}^{a,b}_{n,U}$ for simplicity. Also notice that it is actually a family of languages, with one language per natural number n. We will later see some of the properties of the members of this family, depending on the n's. Notice that, insofar as it contains only a finite set of propositional letters—more precisely, two letters per agent—this language differs from those usually used for modal logics. These propositional letters are supposed to reflect agent i's preference in a given point, in a manner which will become clearer when we formally define the semantics below.

We will use the usual abbreviations given in Definition 2.14, in addition to the following:

Definition 3.7. Given $i, n \in \mathbb{N}$

$$=_i \stackrel{def}{=} a_i \wedge b_i$$

$$\Box_{\dots} \stackrel{def}{=} \Box_1 \dots \Box_n$$

When using $\Box_{\dots}$ it is always clear from context which n we are referring to.

3.3.2 Semantics

We will evaluate formula of $\mathcal{L}^{a,b}_{n,U}$ in pointed $(\tau_{n,U}, \Phi^{a,b})$-models, and we will be interested in those which are true on all such models extracted from social choice functions. The truth evaluation of a formula is as in Definition 2.24; we write it up here for convenience.

We define the family of binary truth-relations $\models_n$ between $\mathbf{Mod}^p(\tau_{n,U}, \Phi^{a,b})$ and $\mathcal{L}^{a,b}_{n,U}$, one for each n. It is usually clear what n we are using; in such cases, we simply write $\models$.

Note that the class $\mathbf{Mod}^p(\tau_{n,U}, \Phi^{a,b})$ contains many more models than those in $\mathbf{M}^p(\mathbf{SCF}(n))$, and the use of (D, P) as the point instead of w is slightly unconventional. We do this for ease of reading as we are mostly interested in the case where the models are actually from $\mathbf{M}^p(\mathbf{SCF}(n))$, and this makes the semantics more approachable for the case we are mostly interested in.

Definition 3.8 ($\models_n : \mathbf{Mod}^p(\tau_{n,U}, \Phi^{a,b}) \times \mathcal{L}^{a,b}_{n,U}$)**.** Given some pointed model $(\mathcal{M}, (D, P))$, we interpret the formula in the following way, letting V be the valuation in $\mathcal{M}$, W its domain, and $R_1, \dots R_n, R_U$ the relations.

- $\mathcal{M}, (D, P) \models_n b_i$ iff $(D, P) \in V(b_i)$
- $\mathcal{M}, (D, P) \models_n a_i$ iff $(D, P) \in V(a_i)$.
- $\mathcal{M}, (D, P) \models_n \Box_i \varphi$ iff $\mathcal{M}, (D, P') \models_n \varphi$ for all (D, P') s.t. $(D, P) R_i (D, P')$
- $\mathcal{M}, (D, P) \models_n \Box^U \varphi$ iff $F, V, (D', D') \models_n \varphi$ for all (D', D') s.t. $(D, P) R_U (D', D')$
- $\mathcal{M}, (D, P) \models_n \varphi \wedge \psi$ iff $\mathcal{M}, (D, P) \models_n \varphi$ and $\mathcal{M}, (D, P) \models_n \psi$
- $\mathcal{M}, (D, P) \models_n \neg\varphi$ iff $\mathcal{M}, (D, P) \not\models_n \varphi$

We then extend $\models_n$ in the usual way to global truth in a model, to validity in a class of models, and to validity over all models.

- $\mathcal{M} \models_n \varphi$ iff $\mathcal{M}, (D, P) \models_n \varphi$ for all $(D, P) \in W$
- $C \models_n \varphi$ iff $\mathcal{M} \models_n \varphi$ for all $\mathcal{M} \in C$
- $\models_n \varphi$ iff $\mathcal{M} \models_n \varphi$ for all $\mathcal{M} \in \mathbf{Mod}(\tau_{n,U}, \Phi^{a,b})$.

We will identify the following logic as all $\mathcal{L}^{a,b}_{n,U}$ formulae true on all models extracted from social choice functions.

Figure 3.1: A padlock

Definition 3.9 ($\mathbf{VL}_{n,U}$).

$$\mathbf{VL}_{n,U} \stackrel{def}{=} \{\varphi \in \mathcal{L}^{a,b}_{n,U} \mid \quad \mathbf{M}(\mathbf{SCF}(n)) \models_n \varphi\}$$

On one way of interpreting the semantics when interpreted in $\mathbf{M}(\mathbf{SCF})$, we see the pair (D, P) as a pair of an actual world (D) and an imaginary world (P) where some agents lie about their preferences. The R_i relation relates two pairs, (D, P) and (D, P'), such that agent i can change P into P' by lying about his preferred ballot. The relation R_U moves between different real worlds where the agents are honest about their preferences. We see that R_U moves between different $R_1, \dots, R_n$ clusters, and we will say that R_u moves between different real worlds, while the $R_1, \dots, R_n$ moves between different lies. We usually call the $R_1, \dots, R_n$ for agent relations/boxes, and the R_U for the semi-global relation/box.

To visualise how the agent boxes interact, the reader is asked to imagine a code padlock, as in Figure 3.1. Each agent get to turn his own wheel, and the profile in the second coordinate is the result of all agents' choices.

We will use a slightly unconventional notation in saying that a model validates a formula, instead of saying that it makes it globally true. It is slightly unconventional talking about something being valid in a model, the most common use of the term "valid" is if something is globally true in all models built by some frame. For us this would correspond to something holding for all social choice functions on a certain set of alternatives A and number of agents n. As we do not focus on this we will use the word in the way described above.

As an SCF determines a unique model, we will also use the same notions for social choice functions instead of models. Thus, we can say, for instance, that an SCF F validates φ, meaning that the model extracted from F validates φ.

Notice also that the semi-global modality is not really global; it does not hit all points, only those with the two coordinates being equal. It is still global in the sense that it hits all the "padlocks"—that is, it hits all the clusters defined by $R_1 \dots R_n$. If one wants a true global modality, one can combine $\Box^U$ with $\Box_{\dots}$ to get $\Box^U \Box_{\dots} \varphi$, which holds if and only if φ is true everywhere in the model.

Validities

The following list reflects basic properties of the models in $\mathbf{M}(\mathbf{SCF}(n))$. In all of them φ and ψ are arbitrary formulae of $\mathcal{L}^{a,b}_{n,U}$.

In what follows, we will use $\Box_i$ to mean that the validity holds for any $i \leq n$.

K $\Box_i(\varphi \to \psi) \to (\Box_i\varphi \to \Box_i\psi)$

S5-i:
T $\Box_i\varphi \to \varphi$
4 $\Box_i\varphi \to \Box_i\Box_i\varphi$
B $\varphi \to \Box_i\Diamond_i\varphi$

The names of the formulae are standard names. T defines reflexivity, 4 defines transitivity and B defines symmetry. These reflects the fact that, for each of the agents, there is an equivalence class of different ballots they can choose. Note that none of the axioms above reflect the interaction between different agents, they just reflect the equivalence classes of points for each agent.

COMM $\Box_j\Box_i\varphi \leftrightarrow \Box_i\Box_j\varphi$

CR $\Diamond_j\Box_i\varphi \to \Box_i\Diamond_j\varphi$

Here we see some validities that reflect how the agents interact, and they both reflect the grid-like structure of the models. The former says that the boxes for different agents commute: if, for all i-accessible profiles, for all j-accessible profiles φ holds, then for all j-accessible profiles for all i-accessible profiles φ holds. The latter expresses that, if j has a ballot x such that, no matter what ballot i submits, φ will be the case, then no matter what i submits, j can submit the ballot x and make φ hold.

The reader might want to recall the padlock where the box for each agent moves that agent's wheel.

KD45-U:
D $\Box^U\varphi \to \Diamond^U\varphi$
4 $\Box^U\varphi \to \Box^U\Box^U\varphi$
5 $\Diamond^U\varphi \to \Box^U\Diamond^U\varphi$

1. $\Box_i\Box^U\varphi \leftrightarrow \Box^U\varphi$
2. $\Box_i\neg\Box^U\varphi \leftrightarrow \neg\Box_i\Box^U\varphi$

Observe that the validities $KD45 - U$ reflect that R_U is not an equivalence relation, as it lacks both symmetry and reflexivity. This as a "lying" point sees all the "honest" points, but the converse does not hold. The validity D defines seriality (i.e. , that all points has at least one R_U-neighbour), 4 defines transitivity, and 5 defines that the relation is euclidean[1]. Together they reflect the fact that all points see into an equivalence class through the R_U relation.

COMP $a_i \vee b_i$

EQ $(=_i \leftrightarrow =_j)$

REACH $\Diamond_1 \ldots \Diamond_n =_i$

The last three validities are interesting, in that they reflect the fact that $\mathcal{L}^{a,b}_{n,U}$ is evaluated in models in $\mathbf{M}^p(\mathbf{SCF})$. COMP reflect that every SCF are total and must provide a winning alternative for each profile, and that each linear order of the agents are total, so they must rank the two winning alternatives relative

[1] $\forall x, y, z \in W(xR_Uy \wedge xR_Uz) \to yR_Uz$

to teach other. EQ reflect the fact that the two linear orders are antisymmetric, so if a agent has $=_i$ this implies that the two winning alternatives in the two profiles must be the same alternative. The axiom REACH reflects that the stacking of all the agent-boxes is the global modality on the second coordinate, giving that by stacking them all the agents can ensure that the point consist of the same profile in the first and second coordinate.

3.3.3 The properties

All the properties below exist in one version per n. They are not explicitly parametrised on this n, however, as it is always clear that the n is the n of the language $\mathcal{L}^{a,b}_{n,U}$ the formulae is written in. The n is part of $\Box_{\ldots}$, but also in conjunctions over all the agents.

i-dict and SP

$$\textit{i-dict} \stackrel{def}{=} \Box^U \Box_{\ldots} a_i$$

This expresses that, of all alternatives which can win (possibly-winning), whatever is topmost for agent i always wins. From the semantics, it is easily seen that this holds at any point in $\mathbf{M}(F)$ if and only if F is i-dictatorial as in Definition 2.9.

$$SP \stackrel{def}{=} \Box^U (\bigwedge_{i \leq n} \Box_i a_i)$$

This says that, for all possible profiles, no single agent is better off by saying he prefers something he does not. In this formula, it is essential that the $\Box^U$ relation takes us to a different real world in which all agents are honest, as we want to speak about what happens if only agent i lies, and nobody else.

2p3a

$$\textit{2p3a} \stackrel{def}{=} \Diamond^U ((\Diamond_{\ldots} \neg a_1 \wedge \Diamond_{\ldots} \neg b_1) \vee (\Diamond_{\ldots} \neg a_2 \wedge \Diamond_{\ldots} \neg b_2))$$

The above formula expresses that there must be some profile where one of the agents has the winning alternative below some other alternative which can win, and above another possibly-winning alternative.

This formula always implies at least three possibly-winning alternatives, but without SP it is not necessarily valid only given three possibly-winning alternatives and two agents. To see that the above formula implies that there must be at least three distinct possibly-winning alternatives, note the following: the formula says there exists some profile P such that an agent i has one other profile with a winner that is better than the winner in P, and one other profile with a winner that is worse. Furthermore, a middle alternative plus one above and one below gives three alternatives.

Note that this formula is **not** a validity on all social choice functions on two agents with three possibly-winning alternatives; we can construct a counterexample. See that, for all profiles, there must be at least one of the alternatives

Agent 1	Agent 2
A	B
B	A
C	C

Table 3.1: An example profile over $|A| = 3, n = 2$

which none of the agents place in the middle. Table 3.1 shows a possible profile over two agents and three alternatives. It should be easy to see that, for any profile, there must be at least one alternative which neither of the agents place in the middle, as there are three alternatives and only two possible middle slots.

If this alternative (or an arbitrary of them if there are two such alternatives) always wins, the inner part of the formula is not satisfiable, so the whole formula is false.

Even though the formula fails to define the property that we want, not everything is lost. Strictly speaking, we do not need to be able to define 3-winning itself; we need only to be able to say that a function is strategyproof **and** 3-winning. As one should see, the above counterexample defines an SCF F that is not strategyproof; as an agent can "veto" away an alternative by putting it in the middle, he will want to put the least preferred alternative in the middle, instead of at the bottom.

The following theorem shows us that the formula *2p3a* is adequate in the sense that it defines 3-winning inside the class of strategyproof models.

Theorem 3.10 (*SP* $\rightarrow$ *2p3a* adequacy)**.** *2p3a is valid in all 3-winning social choice functions validating SP where there are at least two agents.*

Proof. Assume some arbitrary, strategyproof, 3-winning SCF F. For *2p3a* to be valid in $\mathbf{M}(F)$, we need to find some point (D, D) such that

$$(\mathbf{M}(F)(D, D)) \models ((\Diamond_{\ldots}\neg a_1 \wedge \Diamond_{\ldots}\neg b_1) \vee (\Diamond_{\ldots}\neg a_2 \wedge \Diamond_{\ldots}\neg b_2)).$$

Notice that each $\Diamond_{\ldots}$ is universal in the second coordinate. Thus, this formula expresses that, for one of the agents i, there must be some profile P' such that $F(P')$ is not less than or equal $F(D)(\neg a_i)$; and there must be some profile P'' such that $F(P'')$ is not above or equal $F(D)$ $(\neg b_i)$. In other words, the formula tells us that, for some agent i, $F(D)$ must be below some other possibly-winning alternative, and above yet another (in D_i).

We now consider the profile where both agents least prefer the same possibly-winning alternative, but differ on their most-preferred of the possibly-winning alternatives. Without loss of generality, we can image the profile like in Table 3.2, where A, B, C are alternatives that can win. For this point to not be a

Agent 1	Agent 2
A	B
B	A
C	C

Table 3.2: The profile D

witness, the least-preferred alternative (C) must win. Now, assume that one of the agents swaps one of the alternatives with the bottom one. If the winning alternative is still not C, then the function is not *SP*, as the agent managed to get one of his more-preferred alternatives to win by lying about his preference about C. And since the winning alternative is C then we have a witnessing profile D. ∎

3.3.4 Closing remarks

Now we have the tools to express the Gibbard-Satterthwaite theorem in our logic.

$$GS \stackrel{def}{=} SP \wedge 2p3a \rightarrow \bigvee_{i \leq n} \Box^U \Box_{\ldots} a_i$$

It is interesting to note the role of the semi-global modality $\Box^U$ in the formulae expressing the different properties. In all of them the semi-global modality is all the way on the outside, with only other modalities inside. We can say that the inner part of the formulae expresses "local" properties of each of the different real-world clusters, and then each of the properties express that some property must hold locally in all/some real world.

Chapter 4

Metalogical results

> ""It's too late to correct it," said the Red Queen: "when you've once said a thing, that fixes it, and you must take the consequences"."
> —*Through the Looking-Glass*

In this chapter we will focus on some metalogical results, with an eye towards decidability. We will carve out a sub-language of $\mathcal{L}^{a,b}_{n,U}$ that we will call $\mathcal{L}^{a,b}_{n}$. For the associated logic $\mathbf{VL}_n$ we obtain translations both to and from $\mathbf{S5}^m$, which will prove undecidability of $\mathbf{VL}_n$, and by extension $\mathbf{VL}_{n,U}$.

It should be clear that the models $\mathbf{M}(\mathbf{SCF}(n))$ and the class U^n of frames share many structural similarities. If we consider only the second coordinate of the points in $\mathbf{M}(\mathbf{SCF}(n))$-models, and the other points reachable through the $R_1, \ldots, R_n$ relations, it is clear that this is exactly the product of n equivalence classes, where each of the R_i relations is an equivalence relation over all possible linear orders over the set of alternatives. And as $\mathbf{S5}^m$ is the logic of $\mathcal{L}_m$ formulae true on all frames in U^m, and $\mathbf{VL}_{n,U}$ is the logic of all $\mathcal{L}^{a,b}_{n,U}$ formulae true on all models in $\mathbf{M}(\mathbf{SCF}(n))$, it is not unreasonable that these logics have tight connections, although they are in different languages.

4.1 $\mathcal{L}^{a,b}_{n}$

This is a restricted version of the language $\mathcal{L}^{a,b}_{n,U}$, without the $\Box^U$ modality.

4.1.1 Syntax

The language is $\mathcal{L}^{a,b}_{n}$, defined in Definition 2.17, we repeat it here for convenience.

$$\varphi ::= a_i \mid b_i \mid \neg\varphi \mid \varphi \wedge \varphi \mid \Box_i \varphi$$

Notice that this is the same as for the larger version, but without the semi-global modality.

4.1.2 Semantics

A formula in $\mathcal{L}^{a,b}_{n}$ is evaluated in a model in $\mathbf{Mod}^p(\tau_{n,U}, \Phi^{a,b})$, in the same way as in Definition 3.8, except it omits the clause for $\Box^U$. We will by $\mathbf{VL}_n$ denote all formula of $\mathcal{L}_n^{a,b}$ true on every model in $\mathbf{M}(\mathbf{SCF}(n))$, that is, $\mathbf{VL}_n \stackrel{def}{=} \{\varphi \in \mathcal{L}_n^{a,b} \mid \mathbf{M}(\mathbf{SCF}(n)) \models_n \varphi\}$. Notice that in this language we can not move the first coordinate of a point, but we can change the second coordinate freely by stacking all the boxes.

We will not show examples of properties expressible in $\mathcal{L}_n^{a,b}$, its primary feature of interest is not its expressiveness, but its simplicity.

4.2 Translating from $\mathbf{S5}^m$ to $\mathbf{VL}_n$

We will start by providing a translation $\dagger$ from $\mathcal{L}_m$ to $\mathcal{L}_n^{a,b}$, such that for any $\varphi \in \mathcal{L}_m : \varphi \in \mathbf{S5}^m$ if and only if $\varphi^\dagger \in \mathbf{VL}_n$. This translation is twofold; first we provide a formula translation, and then a corresponding model translation which preserves satisfiability through the formula translation. Notice here that there is a dimensional change, from m to n, and the reason will be clear later.

A major difference between the two languages is that $\mathcal{L}_m$ is associated with a countably infinite set of propositional letters, while in $\mathcal{L}_n^{a,b}$ there are only $2 \times n$ propositional letters available. Still, every $\mathcal{L}_m$-formula only has a finite number of propositional letters in use, and thus there is some $n \in \mathbb{N}$ such that $\mathcal{L}_n^{a,b}$ has the same number of propositional letters available.

We are not only interested in any formula translation, but one that preserves satisfiability. Note that in $\mathbf{VL}_n$ there are dependencies between the different propositional letters. For each agent we have two propositional letters, a_i and b_i, but these are not independent in that if we have $a_i \wedge b_i$ at some point for some agent, we must have $a_j \wedge b_j$ for all agents at that point. We also have that for all i we must have $a_i \vee b_i$ at every point. So this means that if we want to use a_i and b_i as the propositional letters we have only two distinct possible valuations, $a_i \wedge \neg b_i$ or $\neg a_i \wedge b_i$.

It should be clear from this that when translating from $\mathcal{L}_m$ we will often need to translate into $\mathcal{L}_n^{a,b}$ with $n > m$. We will make sure that the formula translation has the property that the translated formulae only contain modalities for the first m agents, making the model translation easier.

4.2.1 Formula translation

Given an $\mathcal{L}_m$-formula φ denote its finite set of propositional letters by $\Phi(\varphi)$. As is clear from the discussion above, we will use at least as many agents as $|\Phi(\varphi)|$, and as each agent contributes to one dimension, we need at least m agents as well.

We will assume an enumeration $h \colon \Phi(\varphi) \to \mathbb{N}$ of the propositional letters in $\Phi(\varphi)$. It should be clear that h is a injection, and that it is a bijection between Φ and $\{1, \ldots, |\Phi(\varphi)|\}$.

We let $n = 1 + max(m, |\Phi(\varphi)|)$, and we will now translate φ into $\varphi^\dagger \in \mathcal{L}_n^{a,b}$ as follows:

Definition 4.1 ($\dagger\colon \mathcal{L}_m^{\Phi(\varphi)} \to \mathcal{L}_n^{a,b}$).

$$\begin{aligned} p^\dagger &= a_{h(p)} \\ (\varphi \vee \psi)^\dagger &= \varphi^\dagger \vee \psi^\dagger \\ (\Box_i \varphi)^\dagger &= \Box_i \varphi^\dagger \\ (\neg\varphi)^\dagger &= \neg\varphi^\dagger \end{aligned}$$

for $p \in \Phi(\varphi)$.

Essentially, the translation translates the only thing which is handled differently between the two languages: the propositional letters.

4.2.2 Model translation

The next step is to translate models of $\mathbf{S5}^m$ into models in $\mathbf{M}^p(\mathbf{SCF}(n))$, such that satisfiability of formulae translated with $\dagger$ is preserved. The challenge is that models in $\mathbf{M}^p(\mathbf{SCF}(n))$ must all be induced by some SCF. The model translation problem is essentially a function construction problem, such that the model induced by the new function preserves satisfiability.

Note that an $\mathbf{S5}^m$-model has valuations for all of the possible propositional letters, not only for a finite set. On the other hand, as we can only have a finite number of agents, we only have a finite number of propositional letters. Therefore we will translate a $\mathbf{S5}^m$-model, modulo some finite set of propositional letters. Our goal is that all formulae using only those propositional letters will have satisfiability preserved. Note that since $\mathbf{S5}^m$ is determined by U^m, we will assume that the $\mathbf{S5}^m$ model is in $\mathbf{Mod}^p(U^m)$.

For the translation we need to choose an appropriate set A of alternatives. We also need to find some point (D, P), and finally we must construct an SCF F that will induce the valuation in the model.

The essential insight is that we are free in choosing A, and we can choose A to be as big as the model satisfying φ. We will then make a profile D which will be the first coordinate of the designated point of our designated model, and can thus not be changed in the language $\mathcal{L}_n^{a,b}$. In D we will place the alternatives depending on the associated state in the satisfying $\mathbf{S5}^m$ model. Letting $\mathcal{P}_{fin}(\mathsf{P})$ denote the finite subsets of the set P, we define the model translation as follows.

Definition 4.2 ($\theta\colon \mathbf{Mod}^p(U^m) \times \mathcal{P}_{fin}(\mathsf{P}) \to \mathbf{M}^p(\mathbf{SCF})$). We are given a pointed $S5^m$ model $(\mathcal{M}, w)$ with frame $\mathcal{F} = (W, R_1, \ldots, R_m)$, and some **finite** set Φ of propositional letters. By Theorem 2.32 we can assume that $\mathcal{F}$ is an universal product frame $(W_1, \ldots, W_m)$, so each point $w \in W$ is of the form $(w_1, \ldots, w_m)$, where $w_i \in W_i$, and the relations R_i are universal on W_i. Now we need to build a model in $\mathbf{M}(\mathbf{SCF})$.

(1) We let $n = 1 + max\{m, |\Phi|\}$ and $A = W \cup \{l\}$ where l is a new element not in W. This determines the frame as per Definition 3.1, it remains to define a point (D, P), and construct the SCF F that will induce the valuation.

(2) We will now define a profile D. We will later define F such that $F(D) = l$, but first we place the other alternatives in A relative to l in D_i according to the valuation of the propositional letters in $\mathcal{M}$. Recall that in the formula translation we associated each propositional letter with an agent, and this is reflected now. For each alternative $a \in A \setminus \{l\}$ and for each $i \leq |\Phi|$ we place a below l in D_i iff $a \in V(h^{-1}(i))$, and above otherwise:

$$\forall a \in A \setminus \{l\} : aD_il \text{ iff } a \in V(h^{-1}(i)).$$

Note that $h\colon \Phi \to \mathbb{N}$ is the natural enumeration assumed of the propositional letters, and that $h^{-1}(i) \in \Phi$. And as $A \setminus \{l\}$ is exactly the domain W, the test $a \in V(h^{-1}(i))$ is well formed.

This defines for every D_i two "bags", one of alternatives above l, and one of them below. This allows for many different profiles $D \in L(A)^n$, we choose one and fix it as D for the remaining of the construction.

(3) Fix a family of surjective mappings $(s_i)_{i \leq m}$ such that each $s_i\colon L(A) \to W_i$ maps all the different linear orders over A onto *all* the elements of W_i. These mappings exist, as $L(A)$ is exponentially larger than W, which itself is larger than each W_i. Given $(s_i)_{i \leq m}$, define $s\colon L(A)^n \to W$ in the following way:

$$s(P) = (s_1(P_1), \dots, s_m(P_m)).$$

Note that the domain of s is $L(A)^n$, but its definition uses only the first m linear orders. This is natural as W contains m-tuples. As $n > m$ there will be several distinct $P, P'L(A)^n$ such that $P \neq P'$, but $s(P) = s(P')$.

(4) Pick a $P \in L(A)^n$ s.t. $D_k \neq P_k$ for some $k > m$, while satisfying $s(P) = w$, where w is the distinguished point of the pointed $\mathbf{S5}^m$ model $\mathcal{M}$. As each s_i is onto with respect to W_i, there is a P such that $s(P) = w$, and as s is indifferent about any differences in the coordinates above m, there are P such that $D_k \neq P_k$.

Note that as D and P differ in some coordinates, the SCF F does not need to assign to them the same winning alternative, and this will hold for all profiles P' accessible from P by changing only the first m coordinates.

(5) Define F by letting $F(D) = l$, and then define F for all P' such that $P_j = P'_j, j > m$ in the following way:

$$F(P') = s(P').$$

By construction of P we know that for all of these P' we have $P' \neq D$, so the construction is not immediately inconsistent. Also note that as $W \subset A$, $s(P')$ is an alternative. Observe that s is surjective, and that for every element $a \in W$ there is a P' such that $P_j = P'_j, j > m$ and $s(P') = a$, resulting in F being onto A. F can assign arbitrary alternatives to other profiles, as they are not reachable by any converted $\mathbf{S5}^m$ formula.

(6) From the SCF F constructed we extract a model $\mathbf{M}(F)$ as per Definition 3.2, and we let θ return $(\mathbf{M}(F), (D, P))$. This ends the construction. ⊣

To sum up, we used the fact that we are free in choosing the set A of alternatives, and then we associated every point in the $\mathbf{S5}^m$ model with an alternative. We then mapped profiles to points in the $\mathbf{S5}^m$, such that the profiles had the associated alternative as its winning alternative.

The following theorem describes the interaction between the formula translation $\dagger$ and the model translation θ.

Theorem 4.3. *For all* $(M, w) \in \mathbf{Mod}^p(U^m), \Phi \in \mathcal{P}_{fin}(\mathsf{P}), \varphi \in \mathcal{L}_m^\Phi$:

$$M, w \models \varphi \textit{ iff } \theta((M, w), \Phi) \models \varphi^\dagger.$$

Proof. The proof is by structural induction on φ. From the function s we define $s^\vee : W \to \mathcal{P}(L(A)^n)$,

$$s^\vee(w) = \{P \in L(A)^n \mid s(P) = w\},$$

as all the profiles mapping to some state w. We proceed to prove that for all $w \in W$ and $P \in s^\vee(w)$ we have $(\mathcal{M}, w) \models \varphi$ iff $(\mathbf{M}(F), (D, P)) \models \varphi^\dagger$, where $(\mathbf{M}(F), (D, P)) = \theta((M, w), \Phi)$.

Base Case, $\varphi = p$. The formula is in the shape of a propositional letter $p \in \Phi$. As $p^\dagger = a_{h(p)}$ we need to show that for all w and all $P \in s^\vee(w)$ we have $(\mathcal{M}, w) \models p$ iff $(\mathbf{M}(F), (D, P)) \models a_{h(p)}$. Let $i = h(p)$. Note that as $P \in s^\vee(w)$ we have $F(P) = w$.

We have $(\mathcal{M}, w) \models p$ iff $w \in V(p)$ iff $w \in V(h^{-1}(i))$ iff $wD_i l$, the latter coming from the construction of θ. This is equivalent with $F(P)D_iF(D)$, which holds iff $(\mathbf{M}(F), (D, P) \models a_i$.

As $w \in W$ and $P \in s^\vee(w)$ was arbitrary we have for all $w \in W$ and all $P \in s^\vee(w)$ that $(\mathcal{M}, w) \models p$ iff $(\mathbf{M}(F), (D, P)) \models a_{h(p)}$.

Induction case, $\varphi = \Box_i(\psi)$. Assuming the induction hypothesis we have for any w' and $P \in s^\vee(w')$ that $(\mathcal{M}, w') \models \psi$ iff $(\mathbf{M}(F), (D, P) \models \psi^\dagger$. Now assume that we have $(\mathcal{M}, w) \models \Box_i\psi$, so we have $(\mathcal{M}, w' \models \psi$ for all wR_iw'.

Note that as R_i is universal, we know that it holds for *all* w' s.t. $w \sim_i w'$, meaning all w's possibly differing from w only in their i'th coordinate.

Now assume an arbitrary P' s.t. $P \sim_i P'$, possibly differing from P in only the i'th coordinate. By the construction of s, any such P' has the property that $s(P') = w'$, for some w' s.t. $w \sim_i w'$. By assumption we have for any such w' that $(\mathcal{M}, w' \models \psi$. By induction hypothesis we then get $(\mathbf{M}(F), (D, P')) \models \psi^\dagger$, and as P' was an arbitrary profile such that $P \sim_i P'$, we get $(\mathbf{M}(F), (D, P)) \models \Box_i\psi^\dagger$ which is the same as $(\mathbf{M}(F), (D, P)) \models (\Box_i\psi)^\dagger$.

For the other direction, note that for any $w' : w \sim_i w'$ there exists some $P' : P \sim_i P'$ such that $s(P') = w'$. This follows from the surjectivity of each s_i, and by the coordinate-wise construction of s. By the induction hypothesis we are done. ■

Induction case, $\varphi = \chi \vee \psi$. Assuming an arbitrary $w \in W$ and from the induction hypothesis we have
for all $P \in s^\vee(w)$: $(\mathcal{M}, w) \models \psi$ iff $(\mathbf{M}(F), (D, P)) \models \psi^\dagger$, and
for all $P \in s^\vee(w)$: $(\mathcal{M}, w) \models \chi$ iff $(\mathbf{M}(F), (D, P)) \models \chi^\dagger$.
This yields
$\exists P \in s^\vee(w)$ such that $(\mathbf{M}(F), (D, P)) \models \psi^\dagger$ iff
$\forall P \in s^\vee(w)$: $(\mathbf{M}(F), (D, P)) \models \psi^\dagger$, and similar for χ.

Using this we proceed to see that
$(\mathcal{M}, w) \models \varphi$ iff $((\mathcal{M}, w) \models \chi$ or $(\mathcal{M}, w) \models \psi)$,
and by the induction hypothesis on each disjunct this holds iff
$(\forall P \in s^\vee(w)$: $(\mathbf{M}(F), (D, P) \models \chi^\dagger)$ or $(\forall P \in s^\vee(w)$: $(\mathbf{M}(F), (D, P) \models \psi^\dagger)$.
By the fact noted above this holds if and only if

$\forall P \in s^{\vee}(w) : (\ (\mathbf{M}(F), (D, P)) \models \chi^{\dagger}$ or $(\mathbf{M}(F), (D, P)) \models \psi^{\dagger})$, which by the semantics of $\vee$ holds iff $\forall P \in s^{\vee}(w)$: $(\mathbf{M}(F), (D, P) \models \chi^{\dagger} \vee \psi^{\dagger}$ which by the definition of $\dagger$ is the same as $(\mathbf{M}(F), (D, P) \models \varphi^{\dagger}$.

Induction case, $\varphi = \neg\psi$. Assuming an arbitrary $w \in W$ and $P \in s^{\vee}(w)$, we see that $(\mathcal{M}, w) \models \varphi$ is the same as $(\mathcal{M}, w) \models \neg\psi$ which by definition of $\neg$ holds iff $(\mathcal{M}, w) \not\models \psi$ which by the induction hypothesis holds iff $(\mathbf{M}(F), (D, P) \not\models \psi^{\dagger}$ which by definition of $\neg$ holds iff $(\mathbf{M}(F), (D, P) \models \neg\psi^{\dagger}$ which by the definition of $\dagger$ is the same as $(\mathbf{M}(F), (D, P) \models (\neg\psi)^{\dagger}$.

As $w \in W$ and $P \in s^{\vee}(w)$ were arbitrary, we have that for all $w \in W$ and $P \in s^{\vee}(w)$, $(\mathcal{M}, w) \models \neg\psi$ iff $(\mathbf{M}(F), (D, P) \models (\neg\psi)^{\dagger}$.

As a corollary we have the following special case, in which the translated model is built exactly for the propositional letters in φ (remember that $\Phi(\varphi)$ is the set of propositional letters occurring in φ).

Corollary 4.4. *For all $\varphi \in \mathcal{L}_m, M \in \mathbf{Mod}^p(U^m) : M \models \varphi$ iff $\theta(M, \Phi(\varphi)) \models \varphi^{\dagger}$*

To sum up, we have that if some $\varphi \in \mathcal{L}_m$ is satisfiable then $\varphi^{\dagger} \in \mathcal{L}_n^{a,b}$ is satisfiable. As a formula is satisfiable if and only if its negation is not a validity, we can rephrase it as: if $\varphi^{\dagger} \in \mathbf{VL}_n$ then $\varphi \in \mathbf{S5}^m$. We proceed to prove the converse, in which the translations interact in a slightly different way. Now we assume that we have a model satisfying the translated formula, and we will show that we can construct a model satisfying the original formula.

Theorem 4.5. *For any $\varphi \in L_m$, with its translation $\varphi^{\dagger}$ being in $\mathcal{L}_n^{a,b}$, if there is an $(\mathcal{M}, (D, P)) \in \mathbf{M}^p(\boldsymbol{SCF}(n))$ s.t. $(\mathcal{M}, (D, P)) \models \varphi^{\dagger}$ then there is a $(M, w) \in \mathbf{Mod}^p(U^m)$ such that $(M, w) \models \varphi$.*

Proof. We start with the model $\mathcal{M}$ s.t. $(\mathcal{M}, (D, P)) \models \varphi^{\dagger}$, and we assume it has the form of $\mathcal{M} = (W, R_1, \ldots, R_n, V)$. We construct the pointed $R_1 \ldots R_m$-generated submodel of $(\mathcal{M}, (D, P))$, denoted as $(\mathcal{M}_{|R_1 \ldots R_m}, (D, P)$, and note that this is the model of a U^m frame. We construct a fresh $V' \colon \mathsf{P} \to W_{|R_1 \ldots R_m}$ from V such that for all $p \in \Phi(\varphi)$: $V'(p) = V(a_{h(p)}) \cap W_{|R_1 \ldots R_m}$, and denote this model $\mathcal{M}'_{|R_1 \ldots R_m}$. It is clear that $(\mathcal{M}'_{|R_1 \ldots R_m}, (D, P)) \in \mathbf{Mod}^p(U^m)$.

We will show by induction that for all $(D, P) \in W_{|R_1 \ldots R_m}$: $\mathcal{M}, (D, P) \models \varphi^{\dagger}$ iff $\mathcal{M}'_{|R_1 \ldots R_m}, (D, P) \models \varphi$.

The proof is by structural induction on φ, leaving out the trivial cases for disjunction and negation.

Base Case, $\varphi = p$. The formula is in the shape of a propositional letter p, translated into $a_{h(p)} = a_i$. By construction of V' we have that for any point (D, P) in $\mathcal{M}_{|R_1 \ldots R_m}$, $(D, P) \in V'(p)$ iff $(D, P) \in V(a_i)$.

Induction case, $\varphi = \Box_i\psi$. Assuming the induction hypothesis for the subformula ψ we have for all points $(D, P) \in W_{|R_1 \ldots R_m}$:

$$\mathcal{M}, (D, P) \models \psi^{\dagger} \text{ iff } \mathcal{M}'_{|R_1 \ldots R_m}, (D, P) \models \psi.$$

As the points reachable by R_i in $\mathcal{M}, (D, P)$ are the same as the points reachable by R_i in $(\mathcal{M}'_{|R_1 \ldots R_m}, (D, P))$ we get by the induction hypothesis that $\mathcal{M}, (D, P) \models \Box_i\psi^{\dagger}$ iff $\mathcal{M}'_{|R_1 \ldots R_m}, (D, P) \models \Box_i\psi$, which because of the way $\dagger$ translates $\Box_i$ is the same as $\mathcal{M}, (D, P) \models (\Box_i\psi)^{\dagger}$ iff $\mathcal{M}'_{|R_1 \ldots R_m}, (D, P) \models \Box_i\psi$. ■

Theorem 4.5 gives that for $\varphi \in \mathcal{L}_m$ we have that if $\varphi \in \mathbf{S5}^m$ then $\varphi^\dagger \in \mathbf{VL}_n$, and combining this with Corollary 4.4 we get that for all $\varphi \in \mathcal{L}_m$ there is a $(\mathcal{M}, w) \in \mathbf{Mod}^p(U^m)$ s.t. $(\mathcal{M}, w) \models \varphi$ iff there is a $(\mathcal{M}', (D, P)) \in \mathbf{M}^p(\mathbf{SCF}(n))$ $\varphi^\dagger$ s.t. $(\mathcal{M}', (D, P)) \models \varphi^\dagger$, where $n = 1 + max\{m, |\Phi(\varphi)|\}$.

Corollary 4.6 (Equisatisfiability). *$\varphi \in \mathbf{S5}^m$ iff $\varphi^\dagger \in \mathbf{VL}_n$, where $n = 1 + max\{m, |\Phi(\varphi)|\}$.*

Corollary 4.7 (Undecidability). *The family of logics $\mathbf{VL}_n$ is udecidable.*

From Corollary 4.6 we note that if the family of logics $\mathbf{VL}_n$ were decidable then $\mathbf{S5}^3$ would be decidable as well. From Theorem 2.33 we know that $\mathbf{S5}^3$ is undecidable.

Unfortunately we can not say at which $n \in \mathbb{N}$ the logic $\mathbf{VL}_n$ becomes undecidable. This depends on the smallest number of propositional letters one can restrict $\mathbf{S5}^3$ to use for it to still be undecidable.

4.3 Model checking of $\mathcal{L}^{a,b}_{n,U}$

In this section we will apply the formula translation † in investigating the complexity of model checking for $\mathcal{L}^{a,b}_{n,U}$. Intuitively, model checking is checking whether a given formula is true in a given model. As our models are social choice functions, the question becomes whether the property expressed by the formula holds for the social choice function. Formally, we define the model checking problem as follows:

Definition 4.8 (Model checking). Given a set A of alternatives, n agents and an SCF $F\colon L(A)^n \to A$, a point $(D, P) \in L(A)^n \times L(A)^n$, and a formula $\varphi \in \mathcal{L}^{a,b}_{n,U}$, check if $(\mathbf{M}(F), (D, P)) \models \varphi$.

The complexity of the model checking problem depends on the representation of its input. Complexity is measured as a function in the size of the input, which in our case essentially consists of an SCF F and a formula φ. See Section 2.5 for background in complexity theory.

There are several ways of representing the F, the simplest being an *extensive* one. That is, representing it as a set of input/output pairs. The problem with this approach is that for a given pair A, n there are $|A|! \times n$ different profiles on this pair, giving exponentially (in $|A|$) many possible inputs to the model checking algorithm. Representing the SCF F in this way makes the potential input to the model checking algorithm very large, even for small As. This has the potential effect of making the model checking too "easy", as the complexity is measured relative to the size of the input.

We are often interested in social choice functions with some structure, which follow some algorithm in their decisions. If that is the case, we can often represent the algorithm in a more succinct way than as a set of input/output pairs. We will here focus on those SCFs which can be represented as a polynomially bounded deterministic two-tape Turing machine. A Turing machine can be seen as a computer program, and by polynomially bounded we mean that it is guaranteed to produce the output in polynomial time relative to the input (which is a profile of size $|A| \times n$).

We will now show that model checking as in Definition 4.8 where the SCF is given as a polynomially bounded deterministic two-tape Turing machine is NP-hard.

We do this by assuming that we have an algorithm doing model checking, and then we use it to solve a problem that we already know is NP-hard (in our case SAT).

Theorem 4.9. *Model checking for $\mathcal{L}^{a,b}_{n,U}$ is NP-hard.*

Proof. We reduce SAT, the problem of determining whether a given formula φ of propositional logic over letters $x_1, \ldots, x_k$ is satisfiable by some assignment of true/false to its letters. We translate φ into φ' such that the satisfiability problem of φ corresponds to a model checking problem of φ'.

For this we will construct an SCF F, a set of alternatives A, agents n, and a pair $D', P' \in L(A)^n$ s.t. $(\mathbf{M}(F), (D, P)) \models \varphi'$ iff φ is satisfiable.

Given an instance $\varphi(x_1, \ldots, x_k)$ of SAT, we create an instance of model checking for $\mathcal{L}^{a,b}_{k,U}$ as follows. First we add two elements to the set of alternatives, $A = \{a, b\}$. We then add one agent per Boolean letter (k), and one additional agent d. The intuition is that we let d be a dictator, by letting $F(D)$ be the element on top of D_d.[1] The point (D', P') can be any. This finishes the model construction, and we proceed to translate φ.

We will now apply † from Definition 4.1, but first note that the domain of † is $\mathcal{L}_m$, not propositional logic. But as propositional logic is a sub-language of $\mathcal{L}_m$, it is clear that † is defined on propositional logic as well. Let $\varphi' = \Diamond^U \Diamond_{\ldots} \varphi^\dagger$. Now note that the translation † translates each of the propositional letters x_i into a_i, and no letters are translated into a_d, as d was the last of the $k+1$ agents.

Now we claim that φ' holds in the point (D', P') if and only if φ is satisfiable, proved below.

Claim 4.10. $(\mathbf{M}(F), (D', P')) \models \varphi'$ *iff* φ *is satisfiable.*

It should be clear that both F and A are of constant size, independently of the size of the input formula φ, while n is linear in the size of φ. It is also clear that the translation † takes linear time in φ. Thus we have a linear-time translation preserving size such that if we could solve the translated instance in polynomial time we could solve SAT in polynomial time. ■

Proof of Claim 4.10. Assuming $(\mathbf{M}(F), (D', P')) \models \varphi'$, we get that there is a point (D', P') s.t. $(\mathbf{M}(F), (D, P)) \models \varphi^\dagger$. Denote by V the valuation in $\mathbf{M}(F)$, and define the valuation $V^{PL} \colon \{x_1, \ldots x_k\} \to 2$ as:

$$V^{PL}(x_i) = \begin{cases} \text{true} & (D, P) \in V(a_i) \\ \text{false} & \text{o.w.} \end{cases}$$

See that in propositional language the translation † is identity on the Boolean connectives, and a bijection between x_i and a_i, $1 \leq i \leq k$. Thus φ holds under the valuation V^{PL}.

For the converse, we assume that the formula φ holds under a valuation $V^{PL} \colon \{x_1, \ldots x_k\} \to 2$, and we will show that $(\mathbf{M}(F), (D', P')) \models \varphi'$. This

[1] As A is finite there is always a top element.

holds iff there is some point (D, P) such that $(\mathbf{M}(F), (D, P)) \models \varphi^\dagger$. We proceed to create the profiles D and P such that $\varphi^\dagger$ holds in (D, P). This means that we need to place the two alternatives a, b relative to each other in both D and P for each of the agents $1, \ldots, k, d$. We start by letting the dictator d prefer a in D, and the alternative b in P, and by construction of the SCF F these will also be the outcomes.

For each of other the agents $1 \ldots k$ we let D_i be such that bD_ia (b is ranked less than a) iff $V^{PL}(x_i) =$ true, otherwise aD_ib. P can be any, as long as agent d prefers alternative b. Now note that with $\dagger$ each of the propositional letters x_i is translated into a_i, holding in (D, P) iff agent i prefers the winner in D to the winner in P (bD_ia). By construction of D, this holds if and only if $V^{PL}(x_i) =$ true.

This shows that the truth value of x_i and a_i $1 \leq i \leq k$ coincide, and as the translation is the identity on the Boolean connectives, we get that $\varphi^\dagger$ holds in (D, P), making it the witnessing point of $\Diamond^U \Diamond_{\ldots} \varphi^\dagger$. ■

4.4 Translating $\mathbf{VL}_n$ to $\mathbf{S5}^n$

> "After a fall such as this, I shall think nothing of tumbling down stairs."
> —*Alice, Alice in Wonderland*

In Section 4.2 we translated from $\mathcal{L}_m$ to $\mathcal{L}_n^{a,b}$, but it provides us with no tools for going the other way. In this section we provide this translation from $\mathcal{L}_n^{a,b}$ to $\mathcal{L}_n$.

This allow us to translate formulae of $\mathcal{L}_2^{a,b}$ into formulae of $\mathcal{L}_2$, and we will again construct the translation in such a way that it preserves satisfiability. This will provide decidability of $\mathbf{VL}_2$, as $\mathbf{S5}^2$ is decidable. We will also use the translation in a completeness proof of $\mathbf{VL}_2$, by tying it to a complete axiomatic system for $\mathbf{S5}^2$.

It is clear that to achieve these goals we do not have the same freedom as earlier in changing the dimensions, so we will aim at translating from $\mathcal{L}_n^{a,b}$ into $\mathcal{L}_n$ while preserving the dimension.

We will provide a formula translation much as we did earlier. But in order to be able to translate the model $\mathcal{M} \in \mathbf{Mod}^p(U^n)$ into a model in $\mathbf{M}^p(\mathbf{SCF}(n))$ we need to ensure that the model $\mathcal{M}$ evaluates the propositional letters in such a way that we can construct a SCF satisfying the original formula. We will do this by providing a formula $\delta^n \in \mathcal{L}_n$ which enforces the right kind of models.

4.4.1 Formula translation

We start with the mapping from $\mathcal{L}_n^{a,b}$ formulae to $\mathcal{L}_n$ formulae, named $\star$. We will assume that the set of propositional letters P contains two designated propositional letters per agent, p_i^a and p_i^b.

Definition 4.11 ($\star\colon \mathcal{L}_n^{a,b} \to \mathcal{L}_n$).

$$\begin{aligned}(a_i)^\star &= p_i^a\\ (b_i)^\star &= p_i^b\\ (\varphi \vee \psi)^\star &= (\varphi^\star \vee \psi^\star)\\ (\Box_i \varphi)^\star &= \Box_i \varphi^\star\\ (\neg\varphi)^\star &= \neg\varphi^\star\end{aligned}$$

In addition to the formula translation we provide the formula δ^n for n agents, which the attentive reader might recognise as the conjunction of the global versions of COMP, EQ and REACH for each of the agents, given in Section 3.3.2.

Definition 4.12.

$$\begin{aligned}\delta^n = &\bigwedge_{i\in I} \Box_1 \ldots \Box_n (p_i^a \vee p_i^b) \wedge\\ &\bigwedge_{i,j\in I} \Box_1 \ldots \Box_n ((p_i^a \wedge p_i^b) \leftrightarrow (p_j^a \wedge p_j^b)) \wedge\\ &\bigwedge_{i\in I} \Box_1 \ldots \Box_n (\Diamond_1 \ldots \Diamond_n (p_i^a \wedge p_i^b))\end{aligned}$$

4.4.2 Model translation

We identify a subclass of pointed models on U^n which we will translate from and into, by letting Δ^n be all the pointed U^n models satisfying δ^n.

Definition 4.13. $\Delta^n = \{(M, w) \in \mathbf{Mod}^p(U^n) \mid (M, w) \models \delta^n\}$.

We define the model translation from this subset of $\mathbf{S5}^m$ models in much the same way as in Definition 4.2. This translation differs in that it must preserve the dimension of the models, and that it can assume that the original model satisfies δ^n.

Definition 4.14 ($\circ\colon \Delta^n \to \mathbf{M}^p(\mathbf{SCF}(n))$). Assume we have some pointed model $(\mathcal{M}, w)$ s.t. $(\mathcal{M}, w) \in \Delta^n$, and each of the relations R_i is universal on its domain. Also assume that we have propositional letters p_i^a and p_i^b in P for each of the agents $i \leq n$.

We translate this into a model in $\mathbf{M}(\mathbf{SCF}(n))$. We let $A = W \cup \{l\}$, and proceed to define a function $f\colon W \to A$. By the REACH axiom we know that there is at least one point $v \in W$ such that $\mathcal{M}, v \models p_1^a \wedge p_1^b$, and by EQ we know that then $\mathcal{M}, v \models p_i^a \wedge p_i^b$ for $1 \leq i \leq n$. For all these points v we let $f(v) = l$, while for all other $w \in W$ we let $f(w) = w$.

Now we construct a profile D. As in Definition 4.2 we will let D be defined relative to the values of the propositional letters in $\mathcal{M}$. For any world w such that $\mathcal{M}, w \not\models p_1^a \wedge p_1^b$ we let lD_iw iff $\mathcal{M}, w \models p_i^a \wedge \neg p_i^b$ and wD_il iff $\mathcal{M}, w \models p_i^b \wedge \neg p_i^a$. As $W \subset A$ we know that each such w is in A. We also know that exactly one of the clauses is true for each such point w, as the translation of the COMP axiom gives us that $p_i^a \vee p_i^b$ must hold for all i at all points. For each state w in $\mathcal{M}$ we have the alternative $f(w)$ placed in each D_i relative to the value of p_i^a and p_i^b in w, which concludes the construction of D.

Let $v \in W$ be one of the points such that $\mathcal{M}, v \models p_1^a \wedge p_1^b$. Fix some family of surjective mappings $(s_i)_{i \leq n}$ $s_i : L(A) \to W_i$ such that $s_i(D_i) = v_i$ and each s_i map $L(A)$ onto W_i.

Given s_i we define $s : L(A)^n \to W$ as before: $s(P) = (s_1(P_1), \ldots, s_n(P_n))$, and use it to define the SCF $F : L(A)^n \to A$ as

$$F(P) \mapsto f(s(P)).$$

Denote by P one of the elements in $L(A)$ s.t. $s(P) = w$, where w is the designated point in the pointed model $(\mathcal{M}, w)$, and we know that one exist as s is surjective. Let the return value be the pointed model $(\mathbf{M}(F), (D, P))$, and this concludes the construction. $\dashv$

The following theorem ties together the formula and model translation. Its proof is closely related to the one of Theorem 4.3, in that it uses a model in $\mathbf{Mod}^p(U^m)$ as a basis, and shows that the corresponding model translation is adequate. It also relates to the proof of Theorem 4.5 in that in both of them we have a model and a translated formula which we will prove to be equisatisfiable to a translated model and the original formula.

Theorem 4.15. *For all models $(\mathcal{M}, w) \in \Delta^n$ and $\varphi \in \mathcal{L}_n^{a,b}$:*

$$\circ(\mathcal{M}, w) \models \varphi \text{ if and only if } (\mathcal{M}, w) \models \varphi^\star.$$

Proof. The proof is by structural induction on φ. We give only the base case in full detail, as the induction steps are almost identical to the steps in Proof 4.3. From the function s we again define $s^\vee : W \to \mathcal{P}(L(A)^n)$

$$s^\vee(w) = \{P \in L(A)^n \mid s(P) = w\},$$

to be all the profiles mapping to some state w. Letting W be the domain of $\mathcal{M}$, we proceed to prove that for all $w \in W$ and $P \in s^\vee(w)$ we have $\mathbf{M}(F), (D, P) \models \varphi$ iff $\mathcal{M}, w \models \varphi^\star$, $\mathbf{M}(F), (D, P) = \circ((M, w))$.

Base Case, $\varphi = a_i$. The formula is in the shape of a propositional letter, and without loss of generality assume it is a_i, giving $\varphi^\star = p_i^a$. Assume an arbitrary $w \in W$ and $P \in s^\vee(w)$.

$(\mathcal{M}, w) \models p_i^a$ iff $w \in V(p_i^a)$. Now we distinguish between the two cases $f(w) = l$ and $f(w) \neq l$.

If $f(w) = l$ then $f(s(P)) = l$, so $w \in V(p_i^a)$ and $lD_i f(s(P))$, so $w \in V(p_i^a)$ iff $lD_i f(s(P))$.

If $f(w) \neq l$ then $w \in V(p_i^a)$ iff $lD_i w$ iff $lD_i s(P)$ iff $lD_i f(s(P))$.

So we have $w \in V(p_i^a)$ iff $lD_i f(s(P))$ iff $lD_i F(P)$ iff $F(D)D_i F(P)$ iff $(F, (D, P)) \models a_i$.

As w and $P \in s^\vee(w)$ was arbitrary we have for all $w \in W$ and $P \in s^\vee(w)$: $\mathcal{M}, w \models \varphi^\star$ iff $(\mathbf{M}(F), (D, P)) \models \varphi$.

Induction steps for $\Box_i\psi$, $\neg\psi$ and $\psi_1 \vee \psi_2$. These are basically identical to the proof of Theorem 4.3, so we will only outline them here. If $\varphi = \Box_i\psi$ we assume the induction hypothesis and get that for any w' and $P' \in s^\vee(w')$, $\mathbf{M}(F), (D, P) \models \psi$ iff $\mathcal{M}, w \models \psi^\star$. As the R_i relation in $\mathcal{M}$ is universal we have again that for all $P' \sim_i P$ there is some $w' \sim_i w$ s.t. $s(P') = w$, and that for all $w' \sim_i w$ there is some $P' \sim_i P$ s.t. $s(P') = w$, and we can apply induction hypotesis to get the needed correspondence for ψ on these states.

For the negation we can use induction hypotesis to get $\mathbf{M}(F), (D, P) \not\models \psi$ iff $\mathcal{M}, w \not\models \psi^\star$, and the disjunction goes through by again showing that for both disjuncts ψ we have ($\exists P \in s^\vee(w)$ s.t. $(\mathbf{M}(F), (D, P)) \models \psi$ iff $\forall P \in s^\vee(w)$: $(\mathbf{M}(F), (D, P)) \models \psi$). ∎

From this we deduce the following corollary:

Corollary 4.16. *For any formula $\varphi \in \mathcal{L}_n^{a,b}$, if there is a model $(\mathcal{M}', w) \in \Delta^n$ s.t. $(\mathcal{M}', w) \models \varphi^\star$ then there is a model $(\mathbf{M}(F), (D, P)) \in \mathbf{M}^p(\boldsymbol{SCF}(n))$ s.t. $(\mathbf{M}(F), (D, P)) \models \varphi$.*

This yields satisfiability in one direction, and we proceed to prove the other direction. The following theorem is reminiscent of Theorem 4.5 in that it constructs a generated submodel, and shows that this generated submodel is adequate. It is similar to Theorem 4.3 in that the newly generated model has to satisfy the translated formula.

Theorem 4.17. *For any formula $\varphi \in \mathcal{L}_n^{a,b}$, if there is a model $(\mathcal{M}, (D, P)) \in \mathbf{M}^p(\boldsymbol{SCF}(n))$ s.t. $(\mathcal{M}, (D, P)) \models \varphi$ then there is a model $(\mathcal{M}', w) \in \Delta^n$ s.t. $(\mathcal{M}', w) \models \varphi^\star$.*

Proof. As in Theorem 4.5 we construct the pointed $R_1 \ldots R_n$ generated submodel $\mathcal{M}_{|R_1 \ldots R_n}$, and note that $\mathcal{M}_{|R_1 \ldots R_n}$ is a model on a U^n frame. Then we define V' from V by $V'(p_i^a) = V(a_i)$ and $V'(p_i^b) = V(b_i)$.

The fact that $\mathcal{M}'_{|R_1 \ldots R_n} \in \Delta^n$ follows from the construction of V' and the fact that δ^n expresses validities on $\mathbf{M}^p(\mathbf{SCF}(n))$.

We will show by induction that for all $(D, P) \in W_{|R_1 \ldots R_m}$: $\mathcal{M}, (D, P) \models \varphi$ iff $\mathcal{M}'_{|R_1 \ldots R_m}, (D, P) \models \varphi^\star$.

The proof is by structural induction on φ, leaving out the induction steps as they are the same as in Proof 4.5

Base Case, $\varphi = a_i$. The formula is in the shape of a propositional letter, and wlog we can assume that it is a_i; the translation becomes p_i^a.

By construction of V' we have that for any point (D, P) in $\mathcal{M}_{|R_1 \ldots R_m}$, $(D, P) \in V'(p_i^a)$ iff $(D, P) \in V(a_i)$.

Induction case, $\varphi = \Box_i \psi$. Assuming the induction hypothesis for the subformula ψ we have for all points $(D, P) \in W_{|R_1 \ldots R_m}$: $\mathcal{M}, (D, P) \models \psi$ iff $\mathcal{M}'_{|R_1 \ldots R_m}, (D, P) \models \psi^\star$.

As the points reachable by R_i in $\mathcal{M}, (D, P)$ are the same as the points reachable by R_i in $(\mathcal{M}'_{|R_1 \ldots R_m}, (D, P))$ we get by induction hypotesis that $\mathcal{M}, (D, P) \models \Box_i \psi$ iff $\mathcal{M}'_{|R_1 \ldots R_m}, (D, P) \models \Box_i \psi^\star$, which because of the way $\star$ translates $\Box_i$ is the same as $\mathcal{M}'_{|R_1 \ldots R_m}, (D, P) \models (\Box_i \psi)^\star$. ∎

From the two theorems above we get the following corollary, stating the equisatisfiability of the original $\mathbf{VL}_n$ formula and the translated $\mathbf{S5}^n$ formula.

Corollary 4.18 (Equisatisfiability). *We can translate any $\varphi \in \mathcal{L}_n^{a,b}$ into an $\varphi^\star \in \mathcal{L}_n$ such that φ is satisfiable in $\mathbf{Mod}^p(U^n)$ iff $\varphi^\star$ is satisfiable in Δ^n.*

Recalling from Theorem 2.33 that $\mathbf{S5}^2$ is decidable, we form the following decidability result for $\mathbf{VL}_2$.

Corollary 4.19 (Decidability of $\mathbf{VL}_2$). *It is decidable whether a $\mathcal{L}_2^{a,b}$ formula is an element of $\mathbf{VL}_2$.*

In addition to decidability, we also see that in the process we provided sufficient conditions for a model in $\mathbf{Mod}^p(U^n)$ to be convertible into an SCF F; it is enough for it to be in Δ^n. We will use this fact later, in relation to completeness.

4.5 Investigating completeness

In this section we will investigate completeness of $\mathbf{VL}_n$ for different n's. We already know that $\mathbf{S5}^n, n \geq 3$ has no finite axiomatization, and that $\mathbf{S5}^2$ has a natural axiomatization (see Definition 4.20). We will study whether we can use any of those results to provide similar results for $\mathbf{VL}_n$

We will first examine the two-agent case.

4.5.1 $\mathbf{VL}_2$

Our goal is to construct a deduction system for $\mathcal{L}_2^{a,b}$ whose deducible formulae are exactly the valid formulae over $\mathbf{M}(\mathbf{SCF}(2))$, that is, $\mathbf{VL}_2$.

We will use the fact that we have a complete axiomatic system for $\mathbf{S5}^2$, and we will first show this. We provide the complete axiomatic system given in [7], and we will modify this system, though maintaining an equivalent system. We will then use this last system to prove that the axiomatization of $\mathbf{VL}_2$ given in Definition 4.24 is complete.

Definition 4.20. First we define the following formulae in $\mathcal{L}_2$, for any $1 \leq i, j \leq 2$, and $p, q \in \mathsf{P}$. The reader should compare this with the definition of the set $\mathbf{S5}$ from Definition 2.27.

K$_i$ $\Box_i(p \to q) \to (\Box_i p \to \Box_i q)$

$\mathbf{S5}_i$ T$_i$ $\Box_i p \to p$

4$_i$ $\Box_i p \to \Box_i \Box_i p$

B$_i$ $p \to \Box_i \Diamond_i p$

COMM$_{i,j}$ $\Box_j \Box_i p \leftrightarrow \Box_i \Box_j p$

CR$_{i,j}$ $\Diamond_j \Box_i p \to \Box_i \Diamond_j p$

In addition we add to the set above all tautologies of propositional logic, written with propositional letters from P. We then provide the following three usual deduction rules, modus ponens (MP), necessitation (NEC) and uniform substitution (US), recapitulated below.

MP: From $\varphi \to \psi$ and φ infer ψ.

NEC: From φ infer $\Box_i \varphi$ for any $1 \leq i \leq n$.

US: From φ infer φ' where φ' is obtained from φ' by uniformly replacing proposition letters in φ by formulae from $\mathcal{L}_2$.

We call a deducible formula any formula that is either an axiom, or the result of using a finite number of applications of the rules above.

If φ is deducible we will write $\vdash_{US} \varphi$.

By corollary 5.10 in [7] we have that the set of deducable formula is exactly $\mathbf{S5}^2$, that is, it is sound and complete with respect to EQ^2 and U^2. For a more detailed proof we refer to Section 2.2 of [10].

Now we will define another axiomatic system, and show that it defines exactly the same formulae. First we define the following formula schemata, for any $1 \leq i, j \leq 2$, where $\varphi, \psi \in \mathcal{L}_2$.

Definition 4.21. K_i $\Box_i(\varphi \to \psi) \to (\Box_i\varphi \to \Box_i\psi)$

$\mathbf{S5}_i$ T_i $\Box_i\varphi \to \varphi$

4_i $\Box_i\varphi \to \Box_i\Box_i\varphi$

B_i $\varphi \to \Box_i\Diamond_i\varphi$

$\mathrm{COMM}_{i,j}$ $\Box_j\Box_i\varphi \leftrightarrow \Box_i\Box_j\varphi$

$\mathrm{CR}_{i,j}$ $\Diamond_j\Box_i\varphi \to \Box_i\Diamond_j\varphi$

As axioms we allow any formula in $\mathcal{L}_2$ which is a substitution instance of one of the above formulae. Alternatively, the reader can see it as allowing any formula of $\mathcal{L}_2$ which fits one of the above schemata.

In addition we add all substitution instances of propositional tautologies as axioms, and we provide the deduction rules MP and NEC, but not US. Again we say that formula is deducible if it is either a axiom, or the result of a finite number of applications of the rules on deducible formulae.

If φ is deducible we will write $\vdash_{SC} \varphi$

The reader should notice that the difference between the two systems is in the role of the uniform substitution. The first system, given in Definition 4.20, is a classical one, allowing for uniform substitution in any theorem, and anywhere in the proof. In the second one, we are essentially only allowed to perform uniform substitution on the axioms when we introduce them.

We will now show the equivalence of the two systems. This is folklore, but the the author is not aware of any published proofs on the two axiomatic systems above.

Theorem 4.22. *For all $\varphi \in \mathcal{L}_2$, $\vdash_{US} \varphi$ if and only if $\vdash_{SC} \varphi$.*

First, it is clear that any formula deducible in $\vdash_{SC}$ is deducible in $\vdash_{US}$, as any of the formulae that can be added as axioms in $\vdash_{SC}$ are just a substitution instances of axioms, so in $\vdash_{US}$ one can infer the same formula by introducing the axiom, and using US once. The interesting part is the converse, that for any formula $\varphi \in \mathcal{L}_2$ such that $\vdash_{US} \varphi$ we have that $\vdash_{SC} \varphi$.

Claim 4.23. *For all $\varphi \in \mathcal{L}_2$, if $\vdash_{US} \varphi$ then $\vdash_{SC} \varphi$.*

Proof. Assume an arbitrary such φ where $\vdash_{US} \varphi$. We will now show that for any proof sequence $\varphi_1, \ldots, \varphi_n$, $\varphi_n = \varphi$, all uses of uniform substitution can be emulated by pushing the use of US "down" to an axiom, and such fitting the scheme in $\vdash_{SC}$.

First we need the notion of a "schematic" formula. Any axiom is a schematic formula, and applying any US to a schematic formula yields a schematic formula. We will write "[p/ψ]" for the function which uniformly replaces every occurrence of p with ψ in the formula it is applied to, yielding that [p/ψ] applied on φ results in the formula φ' where p is uniformly replaced with ψ.

Now we will prove that for any of the $\varphi_i \in (\varphi_1, \ldots, \varphi_n)$, if one performs a US on it, and it is not a schematic formula, one can infer the same result by only applying US on some earlier formulae in the sequence, and possibly using MP or NEC. This enables us to push down all the applications of US to the beginning, making sure that all instances of US are on schematic formulae.

We do this by induction on $\varphi_i \in (\varphi_1, \ldots, \varphi_n)$, and the base case is trivial, as φ_1 must be an axiom.

For the induction step assume some $\varphi_i \in (\varphi_1, \ldots, \varphi_n)$, and that we have the result for previous $\varphi_k, k < i$. Now consider any φ' which is the result of applying some [p/ψ] on φ_i. Note that φ' does not need to be one of the $\varphi_1, \ldots, \varphi_n$, but it can be. We will not distinguish between the different ways the formulae φ_i may have been introduced into the proof, and show the induction hypothesis for it: "if one performs a US on φ_i, and it is not a schematic formula, then one can infer the same result by only applying US on some earlier formulae in the sequence, and then MP or NEC."

If φ_i is an axiom we are done, as it is then schematic.

If φ_i was the result of applying MP we know that earlier in the proof we must have had $\varphi_x \to \varphi_i$ and φ_x. Then we can push the use of [p/ψ] to $\varphi_x \to \varphi_i$ and φ_x, and get $\varphi_x[p/\psi] \to \varphi'$ and $\varphi_x[p/\psi]$, where $x[p/\psi]$ is result of applying [p/ψ] on φ_x. By one application of MP we infer φ', and we have pushed the US earlier in the proof.

If φ_i was the result of applying NEC on some formula φ_x, so if $\varphi_i = \Box_k \varphi_x$, we have that $\varphi' = (\Box_k \varphi_x)[p/\psi]$, which is the same as $\Box_k(\varphi_x[p/\psi])$.

The last possibility is if φ_i was itself introduced by applying [q/χ] on some earlier formula φ_x, where $\varphi_x \in (\varphi_1, \ldots, \varphi_n)$. So $\varphi_i = \varphi_x[q/\chi]$ and $\varphi' = (\varphi_x[q/\chi])[p/\psi]$. If φ_x is either an axiom, or the result of applying a finite number of uniform substitutions on an axiom, we are done, as we then just do one more US. Otherwise we know by induction hypothesis, as φ_x is an earlier formula in the deduction, that we can move the US $\varphi_x[q/\chi]$ "down" through the deduction of φ_x, and infer φ_i by applying NEC or MP on some resulting formula(e).

To finally prove the claim, we need to see that for any schematic formula φ_i such that $\vdash_{US} \varphi_i$, we also have $\vdash_{SC} \varphi_i$ by making all the substitutions in one go when we introduce the axiom. Also note that as MP and NEC are the same in the two systems, the rest of the deduction steps go without change. ∎

Deduction system for $\mathbf{VL}_2$

We will now define a deduction system which generates exactly the formulae in $\mathbf{VL}_2$. First we define the following axiom schemata, for $1 \leq i, j \leq 2$:

Definition 4.24 ($\vdash_{\mathbf{SCF}}$). K$_i$ $\Box_i(\varphi \to \psi) \to (\Box_i\varphi \to \Box_i\psi)$

S5$_i$ T$_i$ $\Box_i\varphi \to \varphi$

4$_i$ $\Box_i\varphi \to \Box_i\Box_i\varphi$

B$_i$ $\varphi \to \Box_i\Diamond_i\varphi$

COMM$_{i,j}$ $\Box_j\Box_i\varphi \leftrightarrow \Box_i\Box_j\varphi$

CR$_{i,j}$ $\Diamond_j\Box_i\varphi \to \Box_i\Diamond_j\varphi$

We let every well-formed formula in $\mathcal{L}_2^{a,b}$ which is of one of the forms above be an axiom. In addition we add all substitution instances of propositional tautologies, that is, all propositional tautologies where all the propositional letters are substituted with formulae in $\mathcal{L}_2^{a,b}$. We also add the three following axioms

COMP$_i$ $\Box_{\dots}(a_i \vee b_i)$

EQ $\Box_{\dots}(=_i \leftrightarrow =_j)$

REACH$_i$ $\Diamond_{\dots} =_i$

It should be noted that all of these sets are computable, and that finite unions of computable sets are computable, so we have a computable set of axioms.

We then add the proof rules MP and NEC.

MP: From $\varphi \to \psi$ and φ infer ψ.

NEC: If φ then $\Box_i\varphi$ for any $1 \leq i \leq n$.

We say that a formula is deducible if it is either an axiom, or the result of using a finite number of applications of the rules on deducible formulae.

If φ is deducible we will write $\vdash_{\mathbf{SCF}} \varphi$.

The set of all formulae $\varphi \in \mathcal{L}_2^{a,b}$ such that $\vdash_{\mathbf{SCF}} \varphi$, will be denoted $sLog^2$.

We will now show that this set is sound and complete for $\mathbf{VL}_2$, the logic of all formulae true on all models in $\mathbf{M}(\mathbf{SCF}(2))$.

Recall from Theorem 2.28 that completeness of a logic is the same as satisfiability of consistent formulae. So we proceed to show that any formula φ in $\mathcal{L}_2^{a,b}$ that is consistent with $sLog^2$ has a satisfying model in $\mathbf{M}^p(\mathbf{SCF})$, thus proving completeness of $sLog^2$

In the following we say that a formula $\varphi \in \mathcal{L}_2^{a,b}$ is consistent if it is $sLog^2$-consistent, and we say that $\psi \in \mathcal{L}_2$ is consistent if it is $\mathbf{S5}^2$-consistent.

Theorem 4.25 (Completeness of $sLog^2$)**.** *For any $sLog^2$-consistent formula $\varphi \in \mathcal{L}_2^{a,b}$ we have a pointed model in $\mathbf{M}(\boldsymbol{SCF})$ satisfying φ.*

Proof. Assume some arbitrary consistent $\mathcal{L}_2^{a,b}$ formula φ. As it is consistent we have that $\varphi \to \bot \notin sLog^2$.

Define COMP$^2 \stackrel{def}{=}$ COMP$_1 \wedge$ COMP$_2$, and likewise for REACH^2. Now see that as COMP$^2 \wedge$ EQ $\wedge$ REACH$^2 \in sLog^2$, we have that the formula, conjoined with the axioms COMP$^2 \wedge$ EQ $\wedge$ REACH^2 does not imply $\bot$ either, i.e. $((\varphi \wedge$ COMP$^2 \wedge$ EQ $\wedge$ REACH$^2) \to \bot) \notin sLog^2$. Because otherwise we would have that $(\mathrm{COMP}^2 \wedge \mathrm{EQ} \wedge \mathrm{REACH}^2 \rightarrow (\varphi \rightarrow \bot)) \in sLog^2$ by simple propositional reasoning, which then would mean that $(\varphi \rightarrow \bot) \in sLog^2$ by MP.

Now we translate the conjoined formula into a formula $\mathcal{L}_2$ by using the translation $\star$ from Definition 4.11, and the result is $\varphi^\star \wedge \delta^2$, where δ^2 is as in Definition 4.12.

Now we will prove that $\varphi^\star \wedge \delta^2$ is consistent with $\mathbf{S5}^2$, by showing that if it were not, then $\varphi \wedge \mathrm{COMP}^2 \wedge \mathrm{EQ} \wedge \mathrm{REACH}^2$ would not have been consistent with $sLog^2$. Assume towards a contradiction that $(\varphi^\star \wedge \delta^2 \rightarrow \bot) \in \mathbf{S5}^2$. This means that it is deducible in the $\mathbf{S5}^2$ system given in Definition 4.21, i.e. $\vdash_{SC} \varphi^\star \wedge \delta^2 \rightarrow \bot$. Now it should be clear that we can reproduce the same deduction in $\vdash_{\mathbf{SCF}}$. For any axiom ψ introduced in $\vdash_{SC}$ we can introduce $\psi^{-\star}$ in $\vdash_{\mathbf{SCF}}$, that is, replace back from p_i^a, p_i^b into a_i, b_i. Also note that the formula $(\varphi^\star \wedge \delta^2 \rightarrow \bot)$ contains at most four propositional letters, and the deduction can happen using only that fragment of the language $\mathcal{L}_2$.[2]

Also see that the rules MP and NEC are the same for the two systems, so after the axioms are introduced the rest of the deduction is the same. As $(\varphi^\star)^{-\star} = \varphi$ we have that $\vdash_{\mathbf{SCF}} ((\varphi \wedge \mathrm{COMP}^2 \wedge \mathrm{EQ} \wedge \mathrm{REACH}^2) \rightarrow \bot)$, which contradicts the assumption.

As $\varphi^\star \wedge \delta^2$ is consistent with $\mathbf{S5}^2$, and $\mathbf{S5}^2$ is complete with respect to $Log(U^2)$ we have there is some model in $\mathbf{Mod}^p(U^2)$ satisfying $\varphi^\star \wedge \delta^2$. Now we use the model translation given in Definition 4.14 to get a pointed model in $\mathbf{M}^p(\mathbf{SCF}(2))$, and from Theorem 4.15 we get that the model satisfies $\varphi \wedge \mathrm{COMP}^2 \wedge \mathrm{EQ} \wedge \mathrm{REACH}^2$. Then it must also satisfy φ, and we are done. ■

Theorem 4.26 (Soundness of $sLog^2$). **$\mathbf{M}(\boldsymbol{SCF}(2)) \models sLog^2$**

Proof. As all the axiom schemata are validities on $\mathbf{M}(\mathbf{SCF}(2)$ (see Section 3.3.2), and all the rules preserve validity on models, $sLog^2$ is sound. ■

4.5.2 $\mathbf{VL}_n$

We now advance from $\mathbf{VL}_2$ to $\mathbf{VL}_n$, $n \geq 3$. There is a general result, proved in [7], that there is no finite axiomatization for any logic L s.t.

$$K^m \subseteq L \subseteq \mathbf{S5}^m$$

for $m \geq 3$. [3] The logic $\mathbf{VL}_n$ does not fit in this interval, but it hints that finding a complete system for $\mathbf{VL}_n$ might be harder than for $\mathbf{VL}_2$.

We will fist investigate whether we can construct a simple extension of the axiomatic system given in Definition 4.24, and achieve completeness for $\mathbf{VL}_{3,U}$. The extension is the natural extension, where the system is the same, but parametrised on $1 \leq i, j \leq 3$ instead of $1 \leq i, j \leq 2$, and formulae substituted in are from $\mathcal{L}_3^{a,b}$. This logic will be called $sLog^3$.

In [7, p. 380] they provide a proof of why the same extension when applied to Definition 4.20 yields a deduction system which is not complete for $\mathbf{S5}^3$. We will here convert this proof to show that $sLog^3$ is incomplete for $\mathbf{VL}_3$, and the proof naturally extends to larger n.

[2] This is a consequence of the completeness proofs, but not explicitly stated in neither of them.

[3] K^m is here the logic of the product of m τ_1-frames.

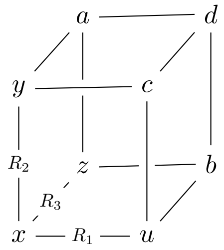

Figure 4.1: A cubic model on 3 dimensions

The proof goes in two steps. First we give a formula, $cube_{123}$, which we will show is a validity on all frames which are products of equivalence classes. We will then give a model refuting $cube_{123}$, while satisfying all the axiom schemata of the proposed deduction system. This gives that $cube_{123}$ can not be deducible.

First we define the formula

$$\begin{aligned} cube_{123} \stackrel{def}{=} & \Diamond_1 u \wedge \Diamond_2 y \wedge \Diamond_3 z \rightarrow \\ & \Diamond_1\Diamond_2\Diamond_3(\Diamond_3(\Diamond_2 u \wedge \Diamond_1 y) \wedge \Diamond_2(\Diamond_3 u \wedge \Diamond_1 z) \wedge \Diamond_1(\Diamond_3 y \wedge \Diamond_2 z)) \end{aligned}$$

Now we see that it holds in any cubic model.

Claim 4.27. *The formula $cube_{123}$ holds in all models in* $\mathbf{Mod}^p(U^n), n \geq 3$.

We will sketch the proof below by showing it for $n = 3$.

Proof. Wlog assume that the model is as pictured in Figure 4.1. In this model I have named the points as the propositional letters which are true in them, and we assume that we are evaluating the formula in x. Note that the antecedent of the formula $cube_{123}$ holds in x, so we need to find a point satisfying $\Diamond_1\Diamond_2\Diamond_3(\Diamond_3(\Diamond_2 u \wedge \Diamond_1 y) \wedge \Diamond_2(\Diamond_3 u \wedge \Diamond_1 z) \wedge \Diamond_1(\Diamond_3 y \wedge \Diamond_2 z))$ accessible through the $\Diamond_1\Diamond_2\Diamond_3$ relation. One such point is d, which the reader can verify makes the formula $(\Diamond_3(\Diamond_2 u \wedge \Diamond_1 y) \wedge \Diamond_2(\Diamond_3 u \wedge \Diamond_1 z) \wedge \Diamond_1(\Diamond_3 y \wedge \Diamond_2 z))$ true by following the R_3 relation to c, the R_2 relation to b and the R_1 relation to a.

For the case where $n \geq 3$, see that the formula speaks only about 3 dimensions, so one can take the submodel generated by R_1, R_2, R_3, and it will be as in Figure 4.1. ■

Now, consider the model in Figure 4.2[4], from now on called $\mathcal{M}$. We will see that $\mathcal{M}$ validates the axiom schemata, but not $cube_{123}$.

First, we need to explain the figure. The horizontal red lines represent R_1, the vertical green ones represent R_2, and all "inwards going" black ones represent R_3. Also note that the figure lacks some transitive lines, omitted in order to keep the figure relatively understandable. Consider for example the point at top left corner, and follow the R_1 relation to the right. From there one can follow

[4]This picture is a slight modification of a picture by Dr. Szabolcs Mikulas, found at `http://www.dcs.bbk.ac.uk/~szabolcs/permu.html`

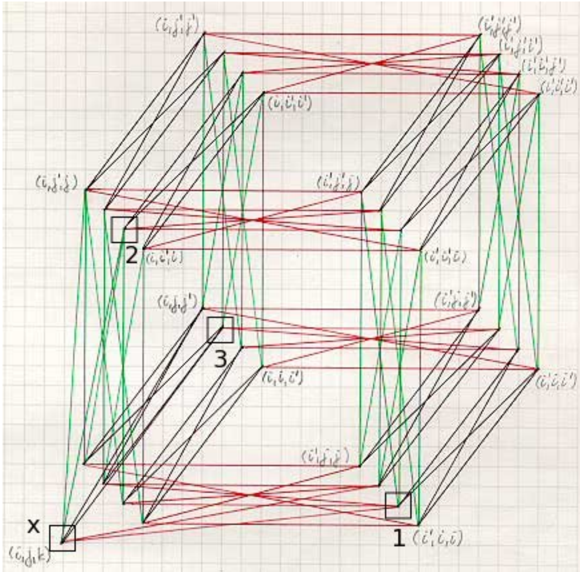

Figure 4.2: Permutation-model

the R_1 relation back left, but down one level. So by transitivity there should be a line between them, but it is not drawn.

We have added boxes to a couple of the points, as we need to distinguish these, in addition to naming the lower left point as x. Initially assume that all points satisfy $\neg a_i \wedge b_i$ for all agents i. We modify the model by letting boxed 1 satisfy $a_1 \wedge \neg b_1$, boxed 2 satisfy $a_2 \wedge \neg b_2$, and boxed 3 satisfy $a_3 \wedge \neg b_3$. The point x satisfies $=_i$ for $1 \leq i \leq 3$.

Now, the reader can verify for himself/herself that this model does in fact validate all instances of the $\mathbf{S5} - i$ axiom schemata for each of the relations. That is, each of the relations is itself an equivalence relation. Also note that $\Box_j\Box_i\varphi \leftrightarrow \Box_i\Box_j\varphi$ and $\Diamond_j\Box_i\varphi \rightarrow \Box_i\Diamond_j\varphi$ holds for all $i, j \leq 3$. We repeat the valuation-axioms

COMP$_i$ $\Box_{\dots}(a_i \vee b_i)$

EQ $\Box_{\dots}(=_i \leftrightarrow =_j)$

REACH$_i$ $\Diamond_{\dots} =_i$

Note that by construction of the valuation in the model, all of these hold in all points, as every point can access the point x through $\Diamond_1\Diamond_2\Diamond_3$, and this is the only point such that $=_i$ for any and all i.

Now we will see that $\mathcal{M}, x$ does not satisfy $cube_{123}$. First we change $cube_{123}$ into the language $\mathcal{L}_n^{a,b}$, by uniformly substituting u, y, z by a_1, a_2, a_3, then we get

$$\begin{aligned} cube_{123} \overset{def}{=} & \Diamond_1 a_1 \wedge \Diamond_2 a_2 \wedge \Diamond_3 a_3 \rightarrow \\ & \Diamond_1\Diamond_2\Diamond_3(\Diamond_3(\Diamond_2 a_1 \wedge \Diamond_1 a_2) \wedge \Diamond_2(\Diamond_3 a_1 \wedge \Diamond_1 a_3) \wedge \Diamond_1(\Diamond_3 a_2 \wedge \Diamond_2 a_3)) \end{aligned}$$

Theorem 4.28. $\mathcal{M}, x \not\models cube_{123}$

Proof. First note that $\mathcal{M}, x \models \Diamond_1 a_1 \wedge \Diamond_2 a_2 \wedge \Diamond_3 a_3$, the antecedent of $cube_{123}$. For the consequent to be false we need to negate its inner part for all points accessible through the $R_1 R_2 R_3$ relation, which the reader can verify consists all points. So to falsify the consequent we need to prove that for all points $v \in \mathcal{M}$ we have $\mathcal{M}, v \not\models (\Diamond_3(\Diamond_2 a_1 \wedge \Diamond_1 a_2) \wedge \Diamond_2(\Diamond_3 a_1 \wedge \Diamond_1 a_3) \wedge \Diamond_1(\Diamond_3 a_2 \wedge \Diamond_2 a_3))$.

Now see that all points on the lower level falsify the formula $\Diamond_3(\Diamond_2 a_1 \wedge \Diamond_1 a_2)$, and the same with the 8 highest points. For the remaining 8 points we must separate between back/front left/right (BL/BR/FL/FR). Both BL and FL falsify $\Diamond_3(\Diamond_2 a_1 \wedge \Diamond_1 a_2)$ as well. FR falsifies $\Diamond_2(\Diamond_3 a_1 \wedge \Diamond_1 a_3)$. In BR there are two points, upper and lower. Upper falsifies $\Diamond_1(\Diamond_3 a_2 \wedge \Diamond_2 a_3)$, and lower falsifies $\Diamond_2(\Diamond_3 a_1 \wedge \Diamond_1 a_3)$. Finally, x falsifies all of the conjuncts. ∎

To wrap up we see that any $R_1, \ldots, R_n$ generated submodel of $\mathbf{M}^p(\mathbf{SCF}(n))$ is a model in $\mathbf{Mod}^p(U^n)$, thus we have that $cube_{123}$ is globally true on all such models.

It is easy to see that the deduction rules NEC and MP preserve global truth on all τ_3-models. That is, when they are applied to globally true formulae then the resulting formulae is globally true as well. We have above noted that all instances of the axiom schemata are globally true on $\mathcal{M}$, but that $cube_{123}$ is not, as $\mathcal{M}, w \not\models cube_{123}$. This shows that $cube_{123}$ can not be deducable from the axioms, as the deduction rules preserve global truth.

It should be noted that this does not show that there does not exist any complete axiomatization, just that the proposed one above is not complete.

4.6 Minimising the language

In this section we will present a result providing a normal form for $\mathcal{L}_n$, while preserving equisatisfiability inside $\mathbf{Mod}(U^n)$. This result was an attempt to minimise $\mathcal{L}_n^{a,b}$, a line of work which has not yet been completed. We still present both the background and the result, as the latter is interesting in itself. Finally we will see why the same method does not immediately give a minimisation result for $\mathcal{L}_n^{a,b}$, even though we have proved tight connections between $\mathbf{VL}_n$ and $\mathbf{S5}^m$.

4.6.1 Background

Having shown the undecidability of $\mathbf{VL}_{n,U}$, it is natural to ask if there are some simple modifications we can apply to $\mathcal{L}_{n,U}^{a,b}$ to make it decidable, while still being able to express the properties we are interested in.

As noted in Section 3.3.4, the formulae expressing the properties we are interested in all share a common pattern: they have the semi-global modality as the outermost modality, and the inner formulae are part of $\mathcal{L}_n^{a,b}$. As $\mathbf{VL}_n$ is undecidable, it is natural to attempt to change $\mathcal{L}_n^{a,b}$ while maintaining expressibility of the inner parts of the property-formulae.

One feature of $\mathcal{L}_n^{a,b}$, and most modal languages, is that it allows for arbitrary stacking of the boxes. This is in fact something the author sees as a virtue, and as noted in Section 1.4, a feature the author wants from a modal logic. But relative to expressing the Gibbard-Satterthwaite theorem, stacking gives us more power than we need. The stacking of the agents' boxes corresponds to the agents cooperating in their lies, and arbitrary stacking of boxes allows the language to talk about the power of arbitrary coalitions. As an example the $\mathcal{L}_n^{a,b}$ formula

$$\neg\Box_1 b_1 \wedge \neg\Box_2 b_2 \wedge \Box_1\Box_2(b_1 \wedge b_2)$$

expresses that neither of the agents 1 or 2 can manipulate the SCF by themselves, but together they can make both of them happier.

The Gibbard-Satterthwaite theorem is not about the power of coalitions, but about the power of single agents. It is possible that this extra expressibility, the ability to speak about coalitional power, is what tips the logic from decidable to undecidable.

We will therefore investigate a simpler language than $\mathcal{L}_n^{a,b}$, one speaking only about single agents. We will not allow any stacking of the agent boxes, thus speaking only about properties of single agents.

A problem is that if we allow no stacking of the agent boxes, we can not express the inner parts of the property-formulae. Examining them more closely, we see that we need the agent-stacking both to express dictatorship, and to say that there are more than three alternatives. In both cases we stack all the boxes, in order to express properties about all alternatives that can win. It seems that to be able to express the Gibbard-Satterthwaite theorem we need to be able to speak about one coalition, the grand coalition of all agents, in addition to the single agents.

Keeping with formal minimalism, we will add a feature as small as possible to the non-agent-stacking language. We extend the language with one more box $\Box^A$, intended to let us speak about the grand coalition, while allowing no stacking of the single agent boxes. The box $\Box^A$ will have the same semantics as $\Box_{\dots}$. It is included solely to allow for the cooperation of the grand coalition, while disallowing cooperation between arbitrary subsets of agents.

As we have already established a strong connection between $\mathbf{VL}_n$ and $\mathbf{S5}^n$, we will start by making a restricted version of $\mathbf{S5}^n$, exploring which properties this language has. We will then see if potential results can be transfered back to $\mathcal{L}_n^{a,b}$.

4.6.2 The unstackable $\mathbf{S5}^n$

In this section we provide a "unstacked" version of $\mathcal{L}_n$ containing a universal modality, and show that this fragment is equally satisfiable to the $\mathcal{L}_n$ with arbitrary stacking. This is based on a conversation with Prof. Venema, and this spurred the underlying idea.

We define $L'_{n,A}$, the modal language with n agent-boxes in addition to $\Box^A$, where one cannot stack any of the agent-boxes, but one can stack the $\Box^A$ in front of them. Following is a *BNF* defining the unstackable language $L'_{n,A}$, consisting all formulae φ.

Definition 4.29 ($L'_{n,A}$)**.**

$$\begin{aligned}
\psi &::= p \mid \psi \wedge \psi \mid \neg\psi \\
\xi &::= \psi \mid \Box_1\psi \mid \cdots \mid \Box_n\psi \\
\varphi &::= \xi \mid \varphi \wedge \varphi \mid \neg\varphi \mid \Box^A\varphi
\end{aligned}$$

The semantics of $\Box^A$ is that of the universal modality, the other modalities work as usual.

$$(\mathcal{M}, w) \models \Box^A \varphi \text{ iff for all } v \in W : (\mathcal{M}, v) \models \varphi$$

We will first provide an example, before proceeding with the actual translation.

Example 4.30. Assuming that the propositional letters $p_1 \ldots$ are fresh, the formulae on the left are equisatisfiable with the formulae on the right.

$$\begin{aligned} \Box_1\Box_2\varphi &\Rightarrow \Box^A(p_1 \leftrightarrow \Box_2\varphi) \wedge \Box_1 p_1 \\ \Box_1\Box_2\Box_3\varphi &\Rightarrow \Box^A(p_1 \leftrightarrow \Box_3\varphi) \wedge \Box^A(p_2 \leftrightarrow \Box_2 p_1) \wedge \Box_1 p_2 \end{aligned}$$

We see that for a boxed subformula φ we can use a fresh propositional letter p and enforce with $\Box^A$ that p is true exactly where φ is true.

We define a translation $\sigma\colon \mathcal{L}_n \to L'_{n,A}$ such that satisfiability is preserved inside $\mathbf{Mod}(\tau_n)$

Definition 4.31 ($\sigma\colon \mathcal{L}_n \to L'_{n,A}$)**.** Assume some $\varphi \in \mathcal{L}_n$. Recall from Definition 2.15 that by $\mathrm{Sub}(\varphi)$ we denote all subformulae of φ. With $\Phi^{-\varphi}$ we denote $\mathsf{P} \setminus \Phi(\varphi)$, the set of propositional letters distinct from the propositional letters occurring in φ. As $\mathsf{P} \setminus \Phi(\varphi)$ is infinite and $\mathrm{Sub}(\varphi)$ is finite, there is a injective function from $\mathrm{Sub}(\varphi)$ into $\Phi^{-\varphi}$, and we will write p_ψ for $\psi \in \mathrm{Sub}(\varphi)$ to denote the propositional letter mapped to by ψ.

We will proceed to do two things. We first need to set up the global requirements, that $\Box_i\psi$ holds iff $p_{\Box_i\psi}$ holds. This needs a bit of care, as ψ might contain boxes itself. In addition we need to translate φ into a formula in $L'_{n,A}$.

We first define the helper function f translating subformulae of φ into formulae using only Boolean combinations of propositional letters.

$$\begin{aligned} f(q) &= q \\ f(\Box_i\psi) &= p_{\Box_i\psi} \\ f(\neg\psi) &= \neg f(\psi) \\ f(\psi_1 \vee \psi_2) &= f(\psi_1) \vee f(\psi_2) \end{aligned}$$

Define the set $\Sigma(\varphi) = \{\Box^A(p_{\Box_i\psi} \leftrightarrow \Box_i f(\psi)) \mid \Box_i\psi \in \mathrm{Sub}(\varphi)\}$, and note that as $\mathrm{Sub}(\varphi)$ is finite, $\Sigma(\varphi)$ is finite as well. Now we let $\sigma(\varphi) = \bigwedge\Sigma(\varphi) \wedge f(\varphi)$, and we are done with the translation. ⊣

To make the translation easier to understand, we provide two examples, each containing the original formula φ, the set $\Sigma(\varphi)$ made during the construction, and the final translated formula.

Example 4.32.

$$\begin{aligned} \varphi &= \Box_1\Box_2 p \\ \Sigma(\varphi) &= \{\Box^A(p_{\Box_2 p} \leftrightarrow \Box_2 p), \Box^A(p_{\Box_1\Box_2 p} \leftrightarrow \Box_1 p_{\Box_2 p})\} \\ \sigma(\varphi) &= \Box^A(p_{\Box_2 p} \leftrightarrow \Box_2 p) \wedge \Box^A(p_{\Box_1\Box_2 p} \leftrightarrow \Box_1 p_{\Box_2 p}) \wedge p_{\Box_1\Box_2 p} \end{aligned}$$

Example 4.33.

$$\begin{aligned} \varphi &= \Box_1(\Box_2 p \wedge q) \\ \Sigma(\varphi) &= \{\Box^A(p_{\Box_2 p} \leftrightarrow \Box_2 p), \Box^A(p_{\Box_1(\Box_2 p \wedge q)} \leftrightarrow \Box_1(p_{\Box_2 p} \wedge q)) \\ \sigma(\varphi) &= \Box^A(p_{\Box_2 p} \leftrightarrow \Box_2 p) \wedge \Box^A(p_{\Box_1(\Box_2 p \wedge q)} \leftrightarrow \Box_1(p_{\Box_2 p} \wedge q)) \wedge p_{\Box_1(\Box_2 p \wedge q)} \end{aligned}$$

It should be clear from the construction of σ that it does actually translate into $L'_{n,A}$, and we proceed to show that it preserves satisfiability.

Theorem 4.34 (Equisatisfiability)**.** *For all formulae $\varphi \in L_n$ there is a $(\mathcal{M}, w) \in \mathbf{Mod}^p(\tau_n)$ s.t. $(\mathcal{M}, w) \models \varphi$ if and only if there is a $(\mathcal{M}', w') \in \mathbf{Mod}^p(\tau_n)$ s.t. $(\mathcal{M}', w') \models \sigma(\varphi)$.*

Proof. ($\Rightarrow$) Assume some $\varphi \in \mathcal{L}_n$ and a pointed model $(\mathcal{M}, w)$ s.t. $(\mathcal{M}, w) \models \varphi$. Now we extend $\mathcal{M}$ to a new model $\mathcal{M}'$, such that it satisfies $\sigma(\varphi)$. We do this by keeping the domain W and the relations $R_1, \dots R_n$ unchanged, but changing the valuation of the propositional letters from $\Phi^{-\varphi}$ in $\mathcal{M}'$.

Letting $V^{\mathcal{M}}$ be the valuation function in $\mathcal{M}$, we define $V^{\mathcal{M}'}$, the valuation function in $\mathcal{M}'$. We initialise $V^{\mathcal{M}'}$ by setting $V^{\mathcal{M}'} = V^{\mathcal{M}}$, and then for all $\psi \in \mathrm{Sub}(\varphi)$ we let $v \in V^{\mathcal{M}'}(p_\psi)$ iff $\mathcal{M}, v \models \psi$.

We will need the following lemma, proved below.

Lemma 4.35. *For all $\psi \in Sub(\varphi)$ and $v \in W$: $\mathcal{M}, v \models \psi$ iff $\mathcal{M}', v \models f(\psi)$.*

We also need to show that $\bigwedge \Sigma(\varphi)$ holds in $(\mathcal{M}', w)$, and we do this by showing that $\bigwedge \Sigma(\varphi)$ is true everywhere in $\mathcal{M}'$.

Proof. First, the conjunction over an empty set is true, so assume $\Sigma(\varphi)$ is non-empty. Note that as $\mathcal{M}'$ is identical to $\mathcal{M}$ except in valuation of propositional letters in $\Phi^{-\varphi}$ we have that for all $\psi \in \mathrm{Sub}(\varphi) : (\mathcal{M}', v) \models \psi$ iff $(\mathcal{M}, v) \models \psi$, as these contain no propositional letters in $\Phi^{-\varphi}$.

From the construction of $\mathcal{M}'$ we have that for all $\Box_i\psi \in \mathrm{Sub}(\varphi)$ and $v \in W$ that $(\mathcal{M}', v) \models p_{\Box_i\psi}$ iff $(\mathcal{M}, v) \models \Box_i\psi$, and as noted above this holds iff $(\mathcal{M}', v) \models \Box_i\psi$, giving $\mathcal{M}' \models p_{\Box_i\psi} \leftrightarrow \Box_i\psi$, that the bi-implication is globally true in $\mathcal{M}'$.

By the proof above we have that $\mathcal{M}' \models \psi \leftrightarrow f(\psi)$, giving $\mathcal{M}' \models \Box_i\psi \leftrightarrow \Box_i f(\psi)$. Therefore $\mathcal{M}' \models p_{\Box_i\psi} \leftrightarrow \Box_i f(\psi)$, and as $\Box^A$ is the global modality, we have $\mathcal{M}' \models \Box^A(p_{\Box_i\psi} \leftrightarrow \Box_i f(\psi))$. And $\Sigma(\varphi)$ consists exactly of $\Box^A(p_{\Box_i\psi} \leftrightarrow \Box_i f(\psi))$ for each $\Box_i\psi \in \mathrm{Sub}(\varphi)$, we have that $\mathcal{M}' \models \bigwedge \Sigma(\varphi)$.

As we have $(\mathcal{M}, w) \models \varphi$, by the two results above we get $(\mathcal{M}', w) \models \bigwedge \Sigma(\varphi) \wedge f(\varphi)$, which is exactly $(\mathcal{M}', w) \models \sigma(\varphi)$.

($\Leftarrow$) Assume a pointed model $(\mathcal{M}, w)$ such that $(\mathcal{M}, w) \models \sigma(\varphi)$. We will now prove that in that case $(\mathcal{M}, w) \models \varphi$. We proceed by induction on $\psi \in \mathrm{Sub}(\varphi)$ and show that $\mathcal{M}, w \models \bigwedge \Sigma(\varphi)$ implies $\mathcal{M} \models f(\psi) \leftrightarrow \psi$. That is, the truth of $\bigwedge \Sigma(\varphi)$ at one point gives the global truth of $f(\psi) \leftrightarrow \psi$.

Base case, $\psi = q$. When $\psi = q$ then $f(\psi) = q$, and $\mathcal{M} \models f(q) \leftrightarrow q$ is obvious.

Induction step, $\psi = \Box_j\psi_1$. By the fact that $(\mathcal{M}, w) \models \bigwedge \Sigma(\varphi)$ and that $\Box_j\psi_1 \in \mathrm{Sub}(\varphi)$, we have that $\mathcal{M} \models p_{\Box_j\psi_1} \leftrightarrow \Box_j f(\psi_1)$. Note that $p_{\Box_j\psi_1} = f(\Box_j\psi_1) = f(\psi)$, so we have $\mathcal{M} \models f(\psi) \leftrightarrow \Box_j f(\psi_1)$. By the induction hypothesis we have that $\mathcal{M} \models \psi_1 \leftrightarrow f(\psi_1)$, yielding by substitution of equivalent terms $\mathcal{M} \models f(\psi) \leftrightarrow \Box_j\psi_1$, and this is exactly $\mathcal{M} \models f(\psi) \leftrightarrow \psi$.

Induction step, $\psi = \psi_1 \wedge \psi_2$. See that $f(\psi) = f(\psi_1 \wedge \psi_2) = f(\psi_1) \wedge f(\psi_2)$ by definition of f. By the induction hypothesis we have that $\mathcal{M} \models f(\psi_1) \leftrightarrow \psi_1$ and $\mathcal{M} \models f(\psi_2) \leftrightarrow \psi_2$, so we obtain $\mathcal{M} \models f(\psi_1) \wedge f(\psi_2) \leftrightarrow \psi_1 \wedge \psi_2$, which is exactly $\mathcal{M} \models f(\psi) \leftrightarrow \psi$.

Induction step, $\psi = \neg\psi_1$. See that $f(\psi) = f(\neg\psi_1) = \neg f(\psi_1)$. By the induction hypothesis we have that $\mathcal{M} \models \psi_1 \leftrightarrow f(\psi_1)$, which is equivalent with $\mathcal{M} \models \neg\psi_1 \leftrightarrow \neg f(\psi_1)$, which is exactly $\mathcal{M} \models \psi \leftrightarrow f(\psi)$.

From $(\mathcal{M}, w) \models \sigma(\varphi)$ we get that $(\mathcal{M}, w) \models f(\varphi)$. As we have $\mathcal{M} \models f(\psi) \leftrightarrow \psi$ we have $(\mathcal{M}, w) \models f(\psi) \leftrightarrow \psi$, and as $\varphi \in \mathrm{Sub}(\varphi)$ we have $(\mathcal{M}, w) \models f(\varphi) \leftrightarrow \varphi$, resulting in $(\mathcal{M}, w) \models \varphi$. ■

Proof of Lemma 4.35. We prove this by induction on the complexity of $f(\psi)$.

Base case, $f(\psi) = q$. If $f(\psi) = q$ then $\psi = q$. We need only see that the models $\mathcal{M}$ and $\mathcal{M}'$ coincide on all letters in $\Phi(\varphi)$.

Base case, $f(\psi) = p_{\Box_i\chi}$. As $f(\psi) = p_{\Box_i\chi}$ we have that $\psi = \Box_i\chi$. By construction of $V^{\mathcal{M}'}$ we have $p_{\Box_i\chi} \in V^{\mathcal{M}'}(v)$ iff $\mathcal{M}, v \models \Box_i\chi$, giving $\mathcal{M}, v \models \Box_i\chi$ iff $\mathcal{M}', v \models p_{\Box_i\chi}$.

Induction step, $f(\psi) = f(\chi) \vee f(\xi)$. As $f(\psi) = f(\chi) \vee f(\xi)$ we have that $\psi = \chi \vee \xi$. By the induction hypothesis we have for all $v \in W$: $(\mathcal{M}, v) \models \chi$ iff $(\mathcal{M}', v) \models f(\chi)$ and $(\mathcal{M}, v) \models \xi$ iff $(\mathcal{M}', v) \models f(\xi)$.

By definition of $\vee$ we have $(\mathcal{M}, v) \models \psi$ iff $((\mathcal{M}, v) \models \chi$ or $(\mathcal{M}, v) \models \xi)$ by the induction hypotesis iff $((\mathcal{M}', v) \models f(\chi)$ or $(\mathcal{M}', v) \models f(\xi))$ by definition of $\vee$ iff $(\mathcal{M}', v) \models f(\chi) \vee f(\xi)$ which is the same as $(\mathcal{M}', v) \models f(\psi)$

Induction step, $f(\psi) = \neg f(\chi)$. As $f(\psi) = \neg f(\chi)$ we know that $\psi = \neg\chi$. By the induction hypothesis we have $(\mathcal{M}, v) \models \chi$ iff $(\mathcal{M}', v) \models f(\chi)$. Then we have $(\mathcal{M}, v) \models \neg\chi$ iff $(\mathcal{M}, v) \not\models \chi$ iff $(\mathcal{M}', v) \not\models f(\chi)$ iff $(\mathcal{M}', v) \models \neg f(\chi)$ being the same as $(\mathcal{M}', v) \models f(\psi)$

■

Theorem 4.34 tells us that the language with n boxes is equally undecidable as one with a global modality where there is no stacking of the agent-boxes.

Denote with $\mathcal{L}'_n$ the language $L'_{n,A}$, where $\Box^A$ is replaced with $\Box_{\dots}$. This is the fragment of $\mathcal{L}_n$ where every formula has the property that the only possible ways to stack boxes are either to stack them all, or to have only one box.

If we restrict ourselves to $\mathbf{Mod}(U^n)$ where we have $\Box_{\dots}$ as the universal modality, we can translate back from $L'_{n,A}$ into the fragment L'_n of $\mathcal{L}_n$, by replacing $\Box^A$ with $\Box_{\dots}$. Denote by $L'_n(\varphi)$ the formula resulting from uniformly replacing $\Box^A$ with $\Box_{\dots}$ in $\sigma(\varphi)$.

As both of the constructions in the proof above keep the relational structure of the models intact, we know that a formula φ is in $S5^n$ iff its unstacked version is.

Corollary 4.36 (Unstacking $\mathbf{S5}^n$)**.** *$\varphi \in \mathbf{S5}^n$ iff $L'_n(\varphi) \in \mathbf{S5}^n$.*

4.6.3 Relating $L'_{n,A}$ and $\mathcal{L}_n^{a,b}$

Our goal with the translation above was to find a similar result to Corollary 4.36 for $\mathcal{L}_n^{a,b}$ in $\mathbf{M}^p(\mathbf{SCF}(n))$. This would show us that a version of $\mathcal{L}_n^{a,b}$ without arbitrary stacking would be just as undecidable as $\mathcal{L}_n^{a,b}$.

One seemingly straightforward way to do this would be to translate a formula in $\mathcal{L}_n^{a,b}$ into a $\mathbf{S5}^n$ formula φ, and then translate $L'_n(\varphi)$ back into $\mathcal{L}_m^{a,b}$. The problem is that in translating back we have to increase the dimensions to accommodate for the new propositional letters introduced in the unstacking procedure, meaning that $\Box_1 \dots \Box_n$ is no longer the universal modality. Consequently, we can not use this to show that every $\mathbf{VL}_n$ formula is equivalent with a unstacked one, yielding undecidability.

The strategy used for $\mathbf{S5}^m$ hinges on the fact that we can introduce new propositional letters, playing the role of subformulae of the original formula. The problem is that inside $\mathbf{M}(\mathbf{SCF})$ the values of the propositional letters are not independent.

We will now see why the method used for $\mathbf{S5}^m$ is not immediately transferable to $\mathbf{VL}_n$. Recall that we have the following validities in $\mathbf{M}(\mathbf{SCF}(m))$.

EQ $(=_i \leftrightarrow =_j)$

REACH $\Diamond_1 \dots \Diamond_m =_i$

From the validities we have that there must be some point accessible through the $R_1 \dots R_n$ relation (the composition of $R_1, \dots, R_n$) such that we have $a_3 \wedge a_1$. Now consider the formula

$$\Box^A(a_3 \leftrightarrow \Box_1 \neg a_1), \tag{4.1}$$

and recall that the relations underlying the agent boxes are reflexive, giving that Formula 4.1 implies $\Box^A(a_3 \rightarrow \neg a_1)$. As $\Box^A$ is seen as the universal modality this formula contradicts the existence of a point such that $a_3 \wedge a_1$. This gives us that Formula 4.1 is a contradiction inside $\mathbf{M}(\mathbf{SCF})$, but it is a formula that we could have obtained as the result if we had used the same technique as in Definition 4.31.

This shows that we are not able to use the same strategy for $\mathcal{L}_n^{a,b}$ to show that every stacked formula is equivalent with an unstacked one. It is thus possible that a version of $\mathcal{L}_n^{a,b}$ with a global modality and no stacking of the boxes is decidable.

4.7 Closing remarks on decidability

We have shown that $\mathbf{VL}_n$ is undecidable, despite the relative inexpressiveness of $\mathcal{L}_n^{a,b}$. In terms of properties of SCFs, it seems like $\mathcal{L}_n^{a,b}$ expresses properties which any SCF-describing language would have to describe, the possibility of lying, and how the lie relates to the actual preference. This is the essence of strategyproofness, and thus also of the Gibbard-Satterthwaite theorem.

The only "unnecessary" feature of $\mathcal{L}_n^{a,b}$ is the stacking of boxes, allowing arbitrary coalitions to cooperate. Not allowing arbitrary stacking might result in a decidable logic, but it is not very elegant. As the results above show, the propositional letters will then play a vital role, as allowing both a global modality and arbitrary propositional letters quickly allows us to express the arbitrary coalitions even in a restricted language. Allowing only a finite number of propositional letters, in addition to restricted stacking, makes for a modal language on the verge of non-reasonableness. When that is said, it is still interesting to explore exactly how weak we must make the language to achieve decidability, even though we are then forced to consider inelegant languages.

The results of $\mathbf{VL}_n$ should be considered when looking for other approaches to modelling the Gibbard-Satterthwaite theorem. In designing a logic for social choice functions we must take care that we can not embed $\mathbf{VL}_n$ into it, thus inheriting its undecidability, which it earns from the way it models the agents' lies. The essential feature is that each profile is a point in the product of several equivalence classes, reflecting that a profile is the result of all the agents' choices, and each agent determines an equivalence relation over possible ballots. This is a rather fundamental property of SCFs, and it is not clear to the author how one could model interesting properties of SCFs, especially strategyproofness, without a logic reflecting this.

Chapter 5

Extending with ATL

> "The beanstalk grew up quite close past Jack's window, so all he had to do was to open it and give a jump on to the beanstalk which ran up just like a big ladder. So Jack climbed, and he climbed and he climbed and he climbed and he climbed and he climbed and he climbed till at last he reached the sky."
> —*Jack and the Beanstalk*

We now investigate an application of a logic for social choice functions. The logic for social choice functions can be any such logic, but we will use the language $\mathcal{L}_{n,U}^{a,b}$ with the semantics given in Definition 3.8. We will later discuss how it can be extended to a more general case.

The underlying principle is to see certain families of social choice functions as concurrent game systems (CGS). On top of this we will combine ATL and the SCF-describing language, in our case $\mathcal{L}_{n,U}^{a,b}$. This will enable us to use ATL to express global properties of the family of functions, while we will use $\mathcal{L}_{n,U}^{a,b}$ to talk about local properties of single SCFs.

This differs from what we have studied earlier, when we only considered single social choice functions. There are several reasons why we would want to do this.

In the first place, it is a natural way to model certain situations. Imagine that some group of agents (e.g. a nation) is trying to determine what kind of governing system it wants. When the agents vote, the outcome might be a different system with a different social choice function associated. And this result might not be final, there might be new election in this state, resulting in a further change of SCF. When analysing this kind of situation, it is natural to ask questions that concern not only each of the social choice functions individually, but the whole family of possible SCFs. We might ask if it is possible to end up in stable situations where there is no more change, and possibly, which coalitions need to cooperate for this to happen. Alternatively, if we see the question from the perspective of some coalition of agents, we might want to ask if there is a way to get to an SCF where this coalition is dictatorial. Some example of formulae, and what they express, can be found in Section 5.1.4.

Another motivation is that forming these kind of families might enable us to express properties about single functions, from the way they interact with the other elements, bringing us back to the original topic of this thesis. We will use parts of this chapter to explore exactly this, and find that by constructing a "family" consisting of many copies of the same SCF, we can express interesting properties about that single SCF as a property of the resulting system.

5.1 Basic definition

We will see how we can see a family of SCF as a CGS. Recall that the basics of ATL and CGS were described in Section 2.4.

There are many ways one can construct a CGS from families of SCFs. The simplest, but least flexible one, will be considered first. The lack of flexibility stems from the fact that it can only model very specific such families, where the family is exactly as large as the set of alternatives of the functions it contains. We will call these self-describing families. We later investigate different methods of remedying the lack of flexibility, but to simplify presentation we start with the more restrictive approach.

5.1.1 Constructing a CGS from SCFs

We will construct a CGS as per Definition 2.34 from a self-describing family of social choice functions. First we give a definition of self-describing SCFs.

Definition 5.1 (Self-describing families of SCFs). A family $(H_i), 1 \leq i \leq m$ of SCFs is self-describing when for each $1 \leq i \leq m$: $H_i\colon L(\{1,\ldots,m\})^n \to \{1,\ldots,m\}$.

Let $(H_i), 1 \leq i \leq m$ be a self-describing family of total functions. Recall that by $L(A)$ we denote the set of all linear orders over A. We will denote the family $(H_i), 1 \leq i \leq m$ by H, and a specific element of it by H_i. We will also denote the set $\{1,\ldots,m\}$ by $[m]$.

Recall that a CGS is a tuple $S = (I, \mathcal{Q}, \Phi, \pi, \Sigma, \delta)$. We let $I = [n]$, the set of agents being the same as the ones for the SCF functions. We let the set of states consist of the social choice functions combined with a pair of profiles: $\mathcal{Q} = H \times L([m])^n \times L([m])^n$. We will be discussing Φ and π later in this section.

For the set of actions we set $\Sigma = L([m])$, that is, the full set of linear orders over the set of alternatives (which is the set indexing H).

We construct the transition function in a slightly unconventional way, by letting $\delta((q, D, P), \overline{a}) = (H_{q(\overline{a})}, D, P)$. That is, we let the transition function use the SCF associated with the point it is in to make the transition to another point. This is well defined, as H is indexed by $[m]$, and q is a function of signature $L([m])^n \to [m]$. And $\overline{a}$, being a collective action over the set of actions $L([m])$, is exactly a member of $L([m])^n$. This fits with the example described in the introduction, where the outcome of the current vote is a new voting system.

We have several options regarding Φ and π, depending on what we want to model. We will here keep matters simple by taking $\Phi = \emptyset$ as standard, but we will keep the language and semantics parametrised on Φ and π. We will discuss this more in Section 5.3.1.

We will later consider other ways of generating the CGS, and with different semantics. This version will be referred to as the model-fusing method. The name is chosen as the CGS is constructed by "fusing" together the family of

SCFs, with the models constructed by applying the function $\mathbf{M}$ on them. In Section 5.2 we will look at ways of constructing a CGS where the states do not contain any traces of the models in $\mathbf{M}(\mathbf{SCF})$.

5.1.2 Combining the languages

Given a set of agents I and a set of propositional letters Φ, we define the language $ATL^{a,b}_{n,U}$ generated by the following rule:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\!\langle C\rangle\!\rangle\mathcal{X}\varphi \mid \langle\!\langle C\rangle\!\rangle\mathcal{G}\varphi \mid \langle\!\langle C\rangle\!\rangle\varphi\mathcal{U}\varphi \mid a_i \mid b_i \mid \Box_i\varphi \mid \Box^U\varphi$$

where $i \in I$, $C \subseteq I$ and $p \in \Phi$. As the reader can see, this is nothing but a combination of the two languages ATL and $\mathcal{L}^{a,b}_{n,U}$, and the intended semantics is simple. The underlying idea is that any part of the formula that is a member of $\mathcal{L}^{a,b}_{n,U}$ will be true or false at a point depending on whether the SCF associated to that point has the properties described by the formula, and the ATL modalities will be evaluated as in Definition 2.36. This is written out in further detail below.

5.1.3 Semantics

We are given a self-describing family of SCFs $(H_j)_{j\leq m}$, and from this we extract a CGS $S = (I, \mathcal{Q}, \Phi, \pi, \Sigma, \delta)$ as described in Section 5.1.1.

Formulae are evaluated at a point (q, D, P) of $\mathcal{Q}$ in the following way. Recall that $D, P \in L([m][])^n$, and that $\sim_i$ is a binary relation on $L([m])^n$ given in Definition 2.3.

Definition 5.2 (Semantics of $ATL^{a,b}_{n,U}$).

$S, (q, D, P) \vDash \langle\!\langle C\rangle\!\rangle\mathcal{X}\varphi$ iff there exists a collective strategy F_C such that
for every sequence $\lambda \in out(q, F_C)$, we have that $S, (\lambda[1], D, P) \vDash \varphi$

$S, (q, D, P) \vDash \langle\!\langle C\rangle\!\rangle\mathcal{G}\varphi$ iff there exists a collective strategy F_C such that
for every sequence $\lambda \in out(q, F_C)$, we have that $S, (\lambda[i], D, P) \vDash \varphi$ for all i

$S, (q, D, P) \vDash \langle\!\langle C\rangle\!\rangle\varphi\mathcal{U}\psi$ iff there exists a collective strategy F_C such that
for every sequence $\lambda \in out(q, F_C)$, there exist a position $j \geq 0$ s.t.
$S, (\lambda(j), D, P) \models \psi$, and for all $0 \leq i \leq j$ we have that $S, (\lambda[i], D, P) \models \varphi$

$S, (q, D, P) \models \Box_i\varphi$ iff $S, (q, D, P') \models \varphi$ for all (D, P') s.t. $(D, P) \sim_i (D, P')$

$S, (q, D, P) \models \Box^U\varphi$ iff $S, (q, D', P') \models \varphi$ for all $D', P' \in L([m])^n$ s.t. $D' = P'$

$S, (q, D, P) \models p$ iff $(q, D, P) \in \pi(p)$

$S, (q, D, P) \models a_i\varphi$ iff $q(D)D_iq(P)$

$S, (q, D, P) \models b_i\varphi$ iff $q(P)D_iq(D)$

$S, (q, D, P) \models \varphi \vee \psi$ iff $S, (q, D, P) \models \varphi$ and $S, (q, D, P) \models \psi$

$S, (q, D, P) \models \neg\varphi$ iff not $S, (q, D, P) \models \varphi$

With the construction of the transition function δ, we see that $S, (q, D, P) \vDash \langle\!\langle C\rangle\!\rangle\mathcal{X}\varphi$ iff the coalition C can choose some ballots K_C such that no matter what other ballots K_{-C} the other agents choose, the outcome of q on the resulting combined profile K is a SCF satisfying φ, that is $S, (q(K), D, P) \models \varphi$.

5.1.4 Example formulae

Below are some examples of formulae, and the properties they express. First we repeat a couple of formulae in $\mathcal{L}^{a,b}_{n,U}$.

$$i\text{-}dict \overset{def}{=} \Box^U \Box_{\ldots} a_i$$

$$SP \overset{def}{=} \Box^U (\bigwedge_{i \in I} \Box_i a_i)$$

$$2p3a \overset{def}{=} \Diamond^U ((\Diamond_{\ldots} \neg a_1 \wedge \Diamond_{\ldots} \neg b_1) \vee (\Diamond_{\ldots} \neg a_2 \wedge \Diamond_{\ldots} \neg b_2))$$

Examples of some possible new formulae, and their meanings, are the following:

- $i\text{-}dict \to \langle\langle i \rangle\rangle \mathcal{G} i\text{-}dict$ says that if i is a dictator, he can make sure she always stays a dictator.
- $\langle\langle i \rangle\rangle \mathcal{G} i\text{-}dict \to i\text{-}dict$ says that if i has a strategy guaranteeing that he can become a dictator, he is already a dictator.
- $\langle\langle \emptyset \rangle\rangle \top \mathcal{U} SP \to \neg 2p3a$ says that if all possible outcomes result in a strategyproof SCF, there must be fewer than 3 alternatives.
- $\langle\langle A \rangle\rangle a_i$ says that in all possible reachable SCFs, agent i prefers the outcome of his real preference to that of the lie he is considering.
- $\langle\langle A \rangle\rangle a_i \to i\text{-}dict$ says that in such a case he is a dictator.
- $\Diamond_i(\langle\langle A \rangle\rangle b_i)$ says that there is some alternative profile accessible for agent i such that for all accessible SCFs, agent i prefers the outcome of the alternative profile.

We see that we are able to express many different properties of the family, and statements about the relative power of coalitions. There is a natural interplay between the two languages, where one expresses properties of the individual social choice functions, and the other expresses how one transitions between the different functions.

We will now consider a slightly different approach, where we have isolated the two logics, focusing on letting each play their different part, and restricting the interplay.

5.2 Allowing other logics

The previous method were tightly connected to $\mathcal{L}^{a,b}_{n,U}$. In this section we will try to take a more abstract approach, and assume only that we have some modal logic capable of expressing properties of SCFs. This enables us to use the same method for other logics, and provides a framework for comparing different SCF-describing logics, in terms of their expressive power over a self-describing family of social choice functions.

5.2.1 Basic setup

In the construction given in Section 5.1.1 we tie together ATL and $\mathcal{L}^{a,b}_{n,U}$ already in the construction of the CGS, by letting $\mathcal{Q} = H \times L([m])^n \times L([m])^n$, but there are other possible approaches. We can make the setup more general by only assuming that we have some modal language $\mathcal{L}$ with a relation $\models_{\mathcal{L}}$ between **SCF** and $\mathcal{L}$. That is, we have some concept of global truth/validity on a social choice function. This is indeed the case for $\mathcal{L}^{a,b}_{n,U}$, so we will use this as a case study for that variant as well. It should be mentioned that the semantics for $\mathcal{L}^{a,b}_{n,U}$ defines a relation between $\mathbf{M}(\mathbf{SCF})$ and $\mathcal{L}^{a,b}_{n,U}$, not between **SCF** and $\mathcal{L}^{a,b}_{n,U}$, but as each element of **SCF** translates to an element in $\mathbf{M}(\mathbf{SCF})$ precisely through the function $\mathbf{M}$, this is a trivial matter.

In constructing the merged language we have to make a decision regarding how closely the two languages will be mixed. The easiest method to generalise is to define the language on two levels, where we can take a "pure" $\mathcal{L}$ formula and use it in an ATL formula, but we can not stack a $\mathcal{L}$ box on top of this formula again. An alternative is to mix the languages as we did above. We will now go down the route of separation.

Before constructing the language there is one pitfall we need to be aware of; that with the intended semantics, shown below, we need to know whether the disjunctions and negation are from $\mathcal{L}$ or from ATL. For many logics there is a difference between "$\neg\varphi$ is globally true on F", and "it is not true that φ is globally true on F". Therefore we will use $\neg^A$ for the ATL negation and $\neg^{\mathcal{L}}$ for the $\mathcal{L}$ negation, and similarly for $\vee$. We then let $ATL^{\mathcal{L}}$ be the language defined below, where $\psi \in \mathcal{L}$.

$$\varphi ::= \psi \mid p \mid \neg^A\varphi \mid \varphi \vee^A \varphi \mid \langle\langle C\rangle\rangle\mathcal{X}\varphi \mid \langle\langle C\rangle\rangle\mathcal{G}\varphi \mid \langle\langle C\rangle\rangle\varphi\mathcal{U}\varphi$$

Constructing the CGS we no longer use points that are members of $H \times L([m])^n \times L([m])^n$, but we let $\mathcal{Q} \stackrel{def}{=} H$. The transition function gets the natural modification

$$\delta(q, \overline{a}) = H_{q(\overline{a})}.$$

The rest of the CGS is constructed as in Section 5.1.1.

Letting $\models_{\mathcal{L}}$ be the binary relation between **SCF** and $\mathcal{L}$, we define the following truth relation:

$S, q \models \psi$ where $\psi \in \mathcal{L}$ iff $q \models_{\mathcal{L}} \psi$

$S, q \vDash \langle\langle C\rangle\rangle\mathcal{X}\varphi$ iff there exists a collective strategy F_C such that for every sequence $\lambda \in out(q, F_C)$, we have that $S, \lambda[1] \vDash \varphi$

$S, q \vDash \langle\langle C\rangle\rangle\mathcal{G}\varphi$ iff there exists a collective strategy F_C such that for every sequence $\lambda \in out(q, F_C)$, we have that $S, \lambda[i] \vDash \varphi$ for all i

$S, q \vDash \langle\langle C\rangle\rangle\varphi\mathcal{U}\psi$ iff there exists a collective strategy F_C such that for every sequence $\lambda \in out(q, F_C)$, we have that φ until ψ

$Sq \models \varphi \vee^A \psi$ iff $S, q \models \varphi$ and $S, q \models \psi$

$S, q \models \neg^A\varphi$ iff not $S, q \models \varphi$

$S, q \models p$ iff $q \in \pi(p)$

The difference relative to the earlier version is that the formula φ must be a *validity* on the SCF that the path goes through, and not only true on the same pair of profiles. We will now compare the two versions in more depth, and determine whether there are differences in their expressive power.

5.2.2 Comparison of the two versions

The version using global truth is the more general in the sense that we can plug into it any SCF-logic, as long as it has a notion of global truth in an SCF. It should be noted that the original version is still relatively easy to generalise, as it does not use the internal structure of the points in any way and just fixes the point while moving between SCFs. We should be able to generalise this version to other logics as well, but the method differs depending on how they choose to extract a model from an SCF.

Comparisons of the global-truth and model-fusing versions depend on the language of the SCF-logic used. If we use $\mathcal{L}_{n,U}^{a,b}$ for both versions it is clear that anything that can be expressed in the global-truth version can also be expressed in the model-fusing setup, as we have a way of expressing global truth on a SCF in the language itself. On the other hand, if we restrict ourselves to $\mathcal{L}_n^{a,b}$, removing the box $\Box^U$, the picture is not that clear. Properties like strategyproofness are global properties of SCFs, and are naturally expressible in the global-truth version, but using $\mathcal{L}_n^{a,b}$, not expressible from a single point (D, P).

In the case where we take $\mathcal{L}_{n,U}^{a,b}$ as the SCF-logic, we can see that there are properties that we can only express in the first version, but not the later one. As an example, consider the formula

$$T, q, (D, P) \models \neg\langle\!\langle N \rangle\!\rangle \top \mathcal{U} \neg a_i,$$

saying that for all strategies of the grand coalition, the formula a_i must hold for all the possible SCFs that we visit. As we never change the pair (D, P), this is true if and only if all the SCFs in the family yield a winner in D that agent i prefers to the winner in P.

The general theme is that the first method enables greater expressivity about how the different SCFs handle specific profiles, while the latter enables greater flexibility, and are arguably more elegant. The latter makes it easy to plug in other SCF-logics, as it abstracts away the semantics, and only demands a truth relation between social choice functions and formulae in the language.

5.3 Future work and ideas

The setup above is rather strict, and works only for the class of self-describing families. We now look at natural variations of the framework, and see if we can extend it to work on larger classes of SCFs.

5.3.1 Propositional letters

In the definition of the induced CGS we left the set of propositional letters unspecified, but taking the empty set as default. In this respect there are several variants. We can for instance introduce special "name" letters, one for each SCF in the model.

We can also use P, the countable set of propositional letters, and investigate what is true for all possible valuations π. This enables us to express modally definable frame properties such as reflexivity, transitivity, and symmetry, but now as properties of the family of SCFs. Reflexivity would correspond to a form of "vetoing-power", that for every point the coalition C can force that they stay in the same SCF, and symmetry expresses that the coalition can always "regret" their choice, and get back the previous SCF.

5.3.2 Family size

Until now we have only been using restricted families of SCFs, namely the self-describing ones. We will look at several ways to relax this restriction.

Firstly, we can relax the requirement of A being $[m]$, as long as $|A| = m$, so we have the situation $(H_i), 1 \leq i \leq m$ where for each $1 \leq i \leq m : H_i \colon L(A)^n \to \{A\}$, and $|A| = m$. As A is as large as H, it is clear that we have a bijection g between them, and we can use this bijection in the transition function δ in the following way

$$\delta(q, \overline{a}) = H_{g(q(\overline{a}))}.$$

We can use a similar technique to model the situation where we have a family of SCFs where the set of alternatives is larger than H. That is, $(H)_j, j \in J$ such that $H_j \colon L(A)^n \to A$ where $A > J$. We then let g be a surjective function from A to J. If there is no clear choice of a suitable g, we can evaluate truth over all possible surjections $g \colon A \to J$.

1-family

As a special case we can have a "family" of only one function. If we use the same strategy as above, defining a constant function g, there seems to be little use for the CGS structure, as every action would fix the point. Instead we opt for the introduction of $|A|$ copies of the same function, combined with the introduction of propositional letters, as discussed in Section 5.3.1. We will then look at some validities on the CGS constructed, where by validities we mean formulae that are globally true under all valuations π.

Let an SCF $F \colon L(A)^n \to A$ be given. Make $|A|$ copies of F, associating each copy with an alternative, and denote this family by $(H_i), 1 \leq i \leq |A|$. Let the transition function be

$$\delta(q, \overline{a}) = H_{q(\overline{a})}.$$

We have the propositional letters P, and we consider all valuation functions $\pi \colon \mathsf{P} \to H$. Assume that the formula

$$p \to \langle\langle i \rangle\rangle \mathcal{X} p$$

is a validity on the CGS constructed from F. If this is the case, we have that the agent i can always force staying in the current point (reflexivity). This again means that i can force any alternative $a \in A$ to win in the SCF F. It should be noted that this is not the definition of i-dictatorial that we have chosen in

Definition 2.9, but it *is* dictatorial in the way i-dictatorial is defined in [8].[1] It is unclear whether we can express i-dictatorship as in Definition 2.9 without resolving to the SCF-describing logic.

We did not use the SCF-describing logic in any way. It might be possible to construct a simple logic capable of expressing only strategyproofness and 3-winning, and then combine this logic with ATL on the structure just described.

Non-deterministic actions

We discussed above the situation where the family is smaller than the set of alternatives, but we can also imagine situations where the opposite holds, i.e. where the family of SCFs is larger than the set of alternatives. The family could for example be all the social choice functions on some set of alternatives and n agents. The construction does not support this properly, as only $|A|$ many social choice functions would be reachable through the transition function δ.

One way of handling this is to introduce a non-deterministic transition function. That is, instead of going to another state, a joint action leads to a set of outcome states, and a coalition can force φ if φ holds in all outcome states for all joint actions compatible with the coalition's action. There are, to the author's knowledge, no papers investigating non-deterministic outcomes in ATL; the closest being one of the approaches in [6]. Here the non-deterministic outcomes are modelled as agent ignorance.

There is still considerable choice involved in choosing the sets we would let the transition function land non-deterministically in; the author can not see any clear best choice.

5.3.3 Letting the inner logic do the actions

In the current framework we have that n-profiles over A play many roles, both as parts of the internal points in the SCF-describing logic, but also as possible actions for the coalitions.

The formula $\langle\langle C\rangle\rangle\mathcal{X}\varphi$ expresses that the coalition C can choose some linear orders P_C such that no matter what other linear orders P_{-C} the rest chooses, φ holds on the outcome of the resulting profile P. We could make the choice of this action more explicit, for example by making the second profile in the internal point be the action for the coalition C. More explicitly, we can define the modality $[C]$ such that

$$(q, D, P) \models [C]\varphi \text{ iff } (q(P'), D, P') \models \varphi \text{ for all } P' \sim_{-C} P.$$

$\langle\langle C\rangle\rangle\mathcal{X}\varphi$ can be expressed as $\Diamond_C[C]\varphi$.

An example of a property we can express in this new language is "for all profiles where we (coalition C) don't prefer the winner, we force a move to an SCF where that same profile gives a winner we prefer", expressed by the formula

$$\Box_{\dots}(a_C \to [C]b_C).$$

[1] See Section 2.1.1 for an introduction to the difference between the different definitions of dictatorship.

5.3.4 Allowing different alternatives

In the definition of a CGS given in Definition 2.34 we have a global set of actions the agents can perform, and in the constructions given in this chapter this set is $L(A)$. In the more common version of concurrent game structures, given in e.g. [2], there is a function d, which for each point $q \in \mathcal{Q}$ gives a set of legal actions in that point. We can adopt this technique to allow families of SCFs where the SCFs do not all work on the same set of alternatives, by letting $d(q)$ give as the set of action $L(A)$, where A is the set of alternatives for the SCF q.

This seems like a natural addition when modeling real-world cases. It allows us to model situations where some voting systems are not directly accessible from others, but might be accessible through some chain of other systems. It might not be possible to vote for a dictatorship in a perfect democracy[2], but it might be possible to get to a dictatorship non the less if we pass through some SCFs that are "coalitional dictatorial". With coalitional dictatorial we mean that there is some coalition which decides the outcome. There might be a path of SCFs such that we start in a democratic one, move through coalitional dictatorial ones for smaller and smaller coalitions, before ending in a dictatorship.

[2]Here, a perfect democracy is one where every agents vote is equally important for the outcome.

Chapter 6

Discussion

> "Whereof one cannot speak, thereof one must be silent."
> —*Ludwig Wittgenstein, Tractatus Logico-Philosophicus*

We discuss some natural extensions of $\mathcal{L}^{a,b}_{n,U}$, and we summarise the open questions raised in this thesis. We conclude with the main lessons learnt in the process of trying to construct a modal logic for social choice functions.

6.1 Future work

6.1.1 Proposed extensions of the language

In this thesis we have kept a strong focus on minimising the language, making it as weak as possible while still keeping it powerful enough so it can express interesting properties. In this section we will go the other way, and consider possible extensions of $\mathcal{L}^{a,b}_{n,U}$. Some of them will be left as little more than research ideas, while others will be discussed more extensively.

There are several reasons why we would want to explore a larger language. For one, making the language more expressive might enable us to make more powerful axioms, facilitating completeness. This is a double edged sword, as the axioms must also be complete for the features of the new language.

Another reason would be to facilitate model checking. A larger language might enable us to check for more interesting properties, but could also have the disadvantage of increasing the complexity of the model checking problem.

All of the proposed extensions here have a truth relation $\models$ between **SCF** and formulae of their language, and can as such be used as a SCF-describing logic together with ATL as described in Section 5.2.1.

Switch and profileEQ

We propose to extend $\mathcal{L}^{a,b}_{n}$ with two new modalities, S and E. The intuition is that S switches the two profiles at a point, while E is a nullary modality which is true exactly when the two profiles are identical. We will see that this is not only an extension of $\mathcal{L}^{a,b}_{n}$, but it also subsumes $\mathcal{L}^{a,b}_{n,U}$, in the sense that $\Box^U$ can be expressed in this new language. The new language will be called $\mathcal{L}^{a,b}_{n,S,E}$.

The semantics is a modification of the version for $\mathcal{L}^{a,b}_{n,U}$ given in Definition 3.8, with the case for $\Box^U$ replaced with the cases for S and E. It is also specified solely for models in $\mathbf{M}(\mathbf{SCF}(n))$ in contrast to the whole of $\mathbf{Mod}^p(\tau_{n,U}, \Phi^{a,b})$.

Definition 6.1 ($\models_n$: $\mathbf{M}(\mathbf{SCF}(n)) \times \mathcal{L}^{a,b}_{n,S,E}$)**.** Given a pointed model $(\mathcal{M}, (D, P))$ we interpret the formulae in the following way.

- $\mathcal{M}, (D, P) \models_n S\varphi$ iff $\mathcal{M}, (P, D) \models_n \varphi$
- $\mathcal{M}, (D, P) \models_n E$ iff $D = P$

First note that we can emulate the semi-global modality $\Box^U$ with these two new modalities in the following way:

$$\Box^U \varphi \overset{def}{=} \Box_{\dots} S(\Box_{\dots}(E \to \varphi))$$

To see that this is the case, assume that we evaluate it at some arbitrary point (D, P). The outermost $\Box_{\dots}$ loops over all possible (D, P') s.t. $P' \in L(A)^n$. The first S switches the point to (P', D) for all of these P'. The second $\Box_{\dots}$ loops over all (P', D') s.t. $D' \in L(A)^n$. The result is that the sub-formula $(E \to \varphi)$ is evaluated for all (P', D') s.t. $D', P' \in L(A)^n$. As E holds at exactly those pairs (P', D') s.t. $P' = D'$ we have that φ must hold at all those pairs, which is exactly the semantics of $\Box^U$.

We note a few validities in this extended language, reflecting the semantics of the new modalities. As usual they hold for all $i \leq n$.

K: $S(\varphi \wedge \psi) \leftrightarrow (S(\varphi) \wedge S(\psi))$

- $E \leftrightarrow SE$
- $E \to b_i \wedge a_i$
- $b_i \wedge a_i \leftrightarrow S(b_i \wedge a_i)$

We are also able to express some new properties of SCFs, e.g. we see that the formula scheme

$$a_i \to Sb_i \wedge b_i \to Sa_i$$

expresses that in the pair (P, D), P_i and D_i rank $F(P)$ and $F(D)$ in the same way relative to each other. That is, $F(P)P_iF(D)$ iff $F(P)D_iF(D)$.

We can use these new language features to talk about strategyproofness in greater detail, not only can we express that an can agent can lie to get a preferred alternative, we can also say something about how he must lie. The formula

$$\Box_i(b_i \to Sb_i)$$

expresses that if i wants to lie to get a preferred winning alternative, he must lie by moving the current winner above the new winner he prefers.

We also have the dual, saying that he can get to a better alternative by being honest about the relative ranking of the current winning alternative and the new one:

$$\Diamond_i(b_i \wedge Sa_i)$$

Cooperate-function

We propose to investigate the introduction of a function in the language, which given an agent at a point, returns the single other agent that he can cooperate with to get the best possible outcome. The natural way would be to use this function inside the modalities, in the form $\Box_{f(i)} a_i$, expressing that no matter which agent lies (alone), agent i will not be better off.

A natural extension is to make the function binary, accepting an agent and a coalition size. $\Diamond_{f(i,k)}\varphi$ is true iff the coalition of size k which can get the most preferred outcome for agent i, can lie together and make φ true.

This enables us to express "agent i needs a majority to lie in order to get a better alternative" ($\bigwedge_{k \leq n/2} \Box_{f(i,k)} a_i$), and "agent i is as well off alone as with any other single agent" ($\Box_{f(i,1)} b_i \leftrightarrow \Box_i b_i$).

Proximity-based accessibility relation

In the previous systems considered, the agents have had an equivalence relation on the possible alternative profiles the can choose. For some applications it can be useful to have a different accessibility relation, such that the agent can not choose between all possible lies, at least not in one step.

The change we propose is to let an agent i move to an alternative profile which is "close" to the current one. There are several possible ways to define closeness; one way could be that you can get to the new profile by swapping two alternatives, another even more restrictive way could be that you can get to the new profile by only swapping two incident items. This allows for a measure of "truthfulness", whether the agent can get to a better alternative by just lying "a bit", or that she must lie "a lot" to get to a preferred winning alternative.

The resulting relation would be reflexive and symmetric, but not transitive. It should be noted that the modal logics of products of $n \geq 3$ such relations (reflexive and symmetric) are not decidable (see e.g. Theorem 8.28 in [7]), so this does not provide a short-cut to decidability.

Extending the language with the alternatives

All the languages considered here have been parametrised on the number of agents, but never on the alternatives. This has its roots in our attempt to model the Gibbard-Satterthwaite theorem, which holds for arbitrary sets of alternatives of size larger than two, but only for a finite set of agents. However, depending on what we want to do with the extended language, we might want to add the set of alternatives to it. Especially in the case of model checking, where we know the set of alternatives, there seems to be few reasons not to add the alternatives to the language.

There are several possible ways to add the alternatives to the language. One is to add each alternative as a nullary modality, true in only those profiles where that alternative is the winning alternative. This is reminiscent of way this is done in e.g. [18], where there is one propositional letter per agenda item.

Another approach is to add a nullary modality per triple of two alternatives and one agent, saying that agent i prefers alternative a above b in the current world, or possibly in the lying world.

If we have a finite number of agents, this allow us to express properties such as "if alternative a wins, then for every other alternative b, more agents prefer

a than b"(i.e. a is a Condorcet winner), or "alternative a wins iff agent i has a ranked above every other alternative".

6.1.2 Other future work

There are many open questions concerning $\mathcal{L}^{a,b}_{n,U}$ and $\mathcal{L}^{a,b}_{n}$. We would like to investigate whether we can use the fact that $\mathcal{L}^{a,b}_{2}$ is decidable, to show a similar result about $\mathcal{L}^{a,b}_{2,U}$. This would enable us to prove the two-agent case of the Gibbard-Satterthwaite theorem. In [16], P. Tang and F. Lin are able to reduce the Gibbard-Satterthwaite theorem to the two-agent/three-alternative case. They then proceed to give a computer program that verifies the claim by iterating over, and checking, all strategyproof SCFs on two agents and three alternatives. If we proved decidability of $\mathcal{L}^{a,b}_{2,U}$ it would provide a more elegant alternative to this iteration.

In investigating the computational complexity of $\mathcal{L}^{a,b}_{n,U}$, we would like to have a tighter bound on the model checking problem. Currently, we know that this is NP-hard, but we do not know if the problem is in NP, rendering it NP-complete.

It is the author's strong intuition, given $\mathbf{VL}_n$'s closeness to $\mathbf{S5}^n$, that $\mathbf{VL}_n$ for $n \geq 3$ is not finitely axiomatizable, nor does it have the finite model property. Establishing these results, or alternatively, finding an axiomatization, would provide deeper understanding of this logic. Extending these results to $\mathbf{VL}_{n,U}$ would then be natural.

In [10], Venema provide a complete axiomatization for $\mathbf{S5}^n, n \geq 3$, by allowing a non-conventional deduction rule, the irreflexivity rule. It is natural to try to extend this to $\mathbf{VL}_n$, and then see if it is possible to extend this further to $\mathbf{VL}_{n,U}$.

We would like to establish the decidability or undecidability of an unstackable version of $\mathcal{L}^{a,b}_{n}$, as discussed in Section 4.7. An undecidability result would strengthen the thesis that there are fundamental properties of the Gibbard-Satterthwaite theorem which naturally induce undecidability of the associated logic. Continuing this line of work could lead to important insights into the inherit complexity of the Gibbard-Satterthwaite theorem.

There is also much work to be done in using ATL and CGSs to describe properties of SCFs, we propose future work in Section 5.3. The author finds the questions of which properties we can describe without an SCF-describing logic, and how the descriptive power of the SCF-describing logic interacts with ATL and the CGS constructed from one or several social choice functions, to be the most interesting ones.

6.2 Conclusion

The main aim of this thesis was the construction of a logic capable of expressing interesting properties of social choice functions. We chose the Gibbard-Satterthwaite theorem as a metric, as it expresses the interaction between fundamental properties of SCFs.

We constructed the language $\mathcal{L}^{a,b}_{n,U}$, evaluated on models extracted from SCFs, resulting in the logic $\mathbf{VL}_{n,U}$. $\mathcal{L}^{a,b}_{n,U}$ has a "two-world" semantics, where each point is the comparison of two worlds, the actual world and a possible lie. The agents can compare the outcome in the actual world against the outcome in other worlds accessible to them by lying about their preferences. This gives a natural way to express strategyproofness, and by adding a modality enabling the transition between different real-world pairs, dictatorship is also quite naturally expressible.

Unfortunately, we are not able to construct a decidable logic capable of expressing the Gibbard-Satterthwaite theorem. In exploring the decidability of $\mathbf{VL}_{n,U}$ we investigated the logic $\mathbf{VL}_n$ for the language $\mathcal{L}_n^{a,b}$. This language is a sub-language of $\mathcal{L}_{n,U}^{a,b}$, evaluated on the same models extracted from SCFs, but the language lacks the ability to move between different real-world pairs. The result is that it can only express the interaction between different lies in one actual world. On the other hand, this turns out to be a quite powerful feature. We show equisatisfiability with $\mathbf{S5}^m$, the modal logic of products of m equivalence classes. As $\mathbf{S5}^m$ is undecidable, so is $\mathcal{L}_n^{a,b}$ on SCFs. We investigated the interaction between $\mathbf{S5}^m$ and $\mathbf{VL}_n$ thoroughly, resulting in a completeness proof of $\mathbf{VL}_2$ using $\mathbf{S5}^2$.

In mapping the boundaries of the undecidability of $\mathbf{VL}_n$, we looked into a modified language. We temporarily let go of the requirement of arbitrary stacking of the boxes, and looked into a restricted variant, disallowing arbitrary stacking. This reflects the fact that strategyproofness is a property regarding the lies of single agents, not arbitrary coalitions of agents. We discovered that in $\mathbf{S5}^m$, all formulae using arbitrary stacking of boxes can be translated to an equisatisfiable formula using only the grand coalition (all boxes) and single boxes, essentially showing that disallowing arbitrary stacking does not help in achieving decidability.

This result does not immediately translate to $\mathbf{VL}_n$, due to its restricted use of propositional letters. It is possible that $\mathcal{L}_n^{a,b}$ without arbitrary stacking is decidable, and that this can be extended to $\mathcal{L}_{n,U}^{a,b}$. But it hints at the danger of introducing unrestricted propositional letters into the language, as this is the distinguishing feature of $\mathcal{L}_n$ used to show the result for $\mathbf{S5}^m$. We argued that there are features of the Gibbard-Satterthwaite theorem which naturally give rise to undecidable modal logics, as it expresses properties of the interaction of several equivalence classes. It might be the case that the only way to achieve decidability is to make a highly restricted language.

Leaving $\mathbf{S5}^m$ and $\mathbf{VL}_n$ behind, we proposed a framework using ATL and CGS, combined with an SCF-describing logic. This framework is capable of expressing properties of certain classes of families of SCFs, namely the self-describing ones. We looked at several modifications, allowing the framework to work with a larger range of families, and with different SCF-describing logics. One of these approaches allows for using a single SCF, and allows for the expression of dictatorship in pure ATL. This shows that this line of work can be useful, even if we are only interested in single SCFs.

Nomenclature

(A, n) frame Frames build from a set of alternatives A and n agents, page 33

$\Box^U$ Semi-global modality, page 36

$\circ$ $\circ \colon \Delta^n \to \mathbf{M}^p(\mathbf{SCF}(n))$, page 52

C^m The class of m-products of frames in C, page 28

$\dagger$ $\dagger \colon \mathcal{L}_m \to \mathcal{L}_n^{a,b}$, page 45

Δ^n All the pointed U^n models satisfying δ^n, page 52

δ^n Conjunction of COMP,EQ and REACH for n agents, page 52

EQ Class of τ_1-frames where the relation is an equivalence relation, page 26

$L(A)$ Set of linear orders over A, page 20

$\mathcal{L}_n$ Same as $\mathcal{L}_n^{\mathsf{P}}$, page 24

$\mathcal{L}_n^{\Phi}$ The modal language with n boxes and a countable set of propositional letters Φ, page 24

$\mathcal{L}_n^{a,b}$ The modal language $ML(\tau_n, \Phi_n^{a,b})$, page 24

$\mathcal{L}_{n,U}^{a,b}$ The modal language $ML(\tau_{n,U}, \Phi_n^{a,b})$, page 36

$\mathbf{M}(F)$ The (A, n) model induced by F, page 34

$ML(\tau, \Phi)$ The modal language over similarity type τ and propositional letters Φ, page 23

$\mathbf{Mod}(C)$ All models built on frames in C, page 25

$\mathbf{Mod}^p(C)$ All pointed models built on frames in C, page 25

$\mathbf{M}^p(F)$ All pointed (A, n) models induced by F, page 34

$\mathbf{M}(\Phi)$ $\mathbf{M}(\Phi) = \{\mathbf{M}(F) \mid F \in \Phi\}$, page 34

P Countable set of propositional letters, page 23

$\mathcal{P}_{fin}(\Phi)$ All finite subsets of Φ, page 45

$\Phi_n^{a,b}$ Set of propositional letters a_i and b_i for each $i \leq n$, page 24

$\Phi(\varphi)$ The propositional letters occurring φ, page 44

$\sim_i$ Relating two profiles accessible for agent i, page 20

$\mathbf{SCF}(A, n)$ The set of all SCFs on A and n, page 20

$\mathbf{S5}^m$ The logic of products of m equivalence relations, page 27

$sLog^2$ All formula in $\mathcal{L}_2^{a,b}$ s.t. $\vdash_{\mathbf{SCF}} \varphi$, page 58

$\star$ $\mathcal{L}_n^{a,b} \to \mathcal{L}_n$, page 51

τ_n Similarity type with n unary boxes, page 23

θ $\mathbf{Mod}^p(U^m) \times \mathcal{P}_{fin}(\mathsf{P}) \to \mathbf{M}^p(\mathbf{SCF}(n))$, page 45

U Class of τ_1-frames where the relation is the universal relation, page 26

$\mathbf{VL}_{n,U}$ The logic $\{\varphi \in \mathcal{L}_{n,U}^{a,b} \mid \quad \mathbf{M}(\mathbf{SCF}(n)) \models_n \varphi\}$, page 38

$\mathbf{VL}_n$ The logic $\{\varphi \in \mathcal{L}_n^{a,b} \mid \quad \mathbf{M}(\mathbf{SCF}(n)) \models_n \varphi\}$, page 44

Bibliography

[1] M. Agrawal, N. Kayal, and N. Saxena. PRIMES is in p. *Annals of Mathematics*, page 781–793, 2004.

[2] Rajeev Alur, Thomas A Henzinger, and Orna Kupferman. Alternating-time temporal logic. *J. ACM*, 49(5):672–713, September 2002. ACM ID: 585270.

[3] Kenneth Joseph Arrow. *Social choice and individual values*. Number 12. Yale Univ Pr, 1963.

[4] Salvador Barberà. An introduction to strategy-proof social choice functions. *Social Choice and Welfare*, 18(4):619–653, 2001.

[5] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal logic*. Cambridge University Press, New York, NY, USA, 2001.

[6] Pål Grønås Drange. Epistemic effectivity: Neighbourhood semantics for coalitional ability and group knowledge. Master's thesis, University of Bergen, Norway, November 2011. Preprint.

[7] Dov M. Gabbay, A Kurucz, F Wolter, and M Zakharyaschev. *Many-Dimensional Modal Logics: Theory and Applications*, volume 148 of *Studies in Logic and the Foundations of Mathematics*. Elsevier North Holland, 2003.

[8] Allan Gibbard. Manipulation of voting schemes: A general result. *Econometrica*, 41(4):587–601, July 1973.

[9] Umberto Grandi and Ulle Endriss. First-order logic formalisation of arrow's theorem. In *Proceedings of the 2nd international conference on Logic, rationality and interaction*, LORI'09, page 133–146, Berlin, Heidelberg, 2009. Springer-Verlag.

[10] Maarten Marx and Y. Venema. *Multi-dimensional modal logic*. Applied logic series. Kluwer Academic Publishers, 1997.

[11] Tobias Nipkow. Social choice theory in HOL. *Journal of Automated Reasoning*, 43(3):289–304, 2009. 10.1007/s10817-009-9147-4.

[12] Marc Pauly. On the role of language in social choice theory ,. *Synthese*, 163(2), 2008.

[13] Philip J. Reny. Arrow's theorem and the Gibbard-Satterthwaite theorem: a unified approach. *Economics Letters*, 70(1):99–105, January 2001.

[14] Mark Allen Satterthwaite. Strategy-proofness and arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10(2):187–217, April 1975.

[15] Lars-Gunnar Svensson. The proof of the Gibbard-Satterthwaite theorem revisited. Working Papers 1999:1, Lund University, Department of Economics, 1999.

[16] P. Tang and F. Lin. Computer-aided proofs of arrow's and other impossibility theorems. *Artificial Intelligence*, 173(11):1041–1053, 2009.

[17] Thomas Ågotnes, Wiebe van der Hoek, and Michael Wooldridge. Towards a logic of social welfare. In *In Proceedings of LOFT-2006*, 2006.

[18] Thomas Ågotnes, Wiebe van der Hoek, and Michael Wooldridge. On the logic of preference and judgment aggregation. *Autonomous Agents and Multi-Agent Systems*, 22:4–30, 2011. 10.1007/s10458-009-9115-8.