Agreement and cooperation under degrees of homogeneity in multi-agent systems

Truls Pedersen

Dissertation for the degree philosophiae doctor (PhD) at the University of Bergen

2014

Dissertation date: 18.12.2014
I would like to thank my supervisor Thomas Ågotnes for providing both guidance and freedom during my time as a Ph. D. student. The advice has been sound and lead me to write this thesis on a topic that I really developed a relationship with over the last few years. The freedom has been as firm as the advice, even the worst possible ideas I have suggested pursuing over the last years have been met with considered suggestions of perhaps reconsidering.

Indeed I feel that the entire logic community at the University of Bergen has been supportive in letting its members, of which I am happy to be one, explore new topics in an open and supportive way. I consider myself lucky to have been in the non-hierarchical mutual support group of good people. Even though (or perhaps because) this group is small and (institutionally) scattered, the common desire to help its members through all aspects of academic life.

I am grateful to a great number of people. My coauthors, particularly deserve recognition. Besides my supervisor, Sjur Dyrkolbotn, Piotr Kązmierczak, and Erik Parmann have provided great help in developing and seeing through the development of several of the ideas that this thesis is founded on. Without them, this thesis would not have become what it has.

I hope and believe that this thesis can have some value in itself, but it is hard for me to compare this to the absolute privilege it has been for me to be able to explore this abstract niche of the universe together with such good friends. The things I have learned throughout my time as a Ph. D. student are not limited to the abstract and the formal, and I am truly grateful to you all.

Many of my friends, family members and colleagues must appear, to the reader, to be strategically homogeneous, but to me they are not. Not one of these is the same as the other and they have all acted in an impeccable way – some have talked with me when I needed it, others have not talked with me when that was what I needed. I am overwhelmed by gratitude to all of you, for being unique in each of your own ways.

If you know what an agent can do, you know a lot about this agent. Equally true however must surely be that if you know what an agent becomes able to do, you know a lot about this agent’s friends.

I am also grateful to the Institute of Information Science and Media Studies for support (not only financial), and a great and diverse faculty. I have also learned a great deal about things that I will never be able to understand fully, but I appreciate the glimpses into what I presume few computer scientists get to experience through the pursuit of their Ph. D. studies.

So, to all my friends (named and unnamed): Thank you!
Abstract

This is a thesis in the area of logic-based knowledge representation and reasoning for multi-agent systems. The state of the art in this area is based on modal logic and focus on reasoning about groups of agents, or coalitions: what can a coalition do if they cooperate? How can their behavior be coordinated? How can disagreement be resolved?

Most existing work in this area assumes that the agent population can be completely heterogeneous. In many application domains, however, there is a degree of homogeneity: several agents are similar, in the sense that they can “do the same thing”. An example is the voting domain, where all voters have the same set of actions available and where the outcome of a set of actions is typically independent of the identity of the agent who chose each of the actions (votes).

More generally, in many domains agents can naturally be modeled using roles, such as PhD student, supervisor or evaluation committee member, where agents that play the same role are similar in the same sense as above – they have the same actions available. In this thesis we take seriously the fact that many agent domains are homogeneous to a certain degree, and that modeling can be simplified by exploiting symmetries and similarities in, e.g., agents’ strategic abilities, and investigate logical and computational consequences for existing and new modal logic frameworks for reasoning about cooperation and agreement.

The fact that all voters can “do the same thing”, is the requirement that the system they are acting in (the voting process) should regard them equally. This is generally implemented by they voters’ votes being anonymous. Anonymity in formal voting theory, as well as in game theory, is defined in precisely this way. A voting procedure, or a game, is anonymous if the identity of the participant is not relevant to the outcome. In this thesis we explore the consequence of agents being able to “do the same thing” in logical models of multi-agent interaction and cooperation.

In particular, we introduce several new variants of Alternating-time Temporal Logic (ATL) with roles in the semantic models and/or in the logical object language. These new logics expose the whole spectrum from fully heterogeneous to fully homogeneous agent populations. It turns out that in both cases, the complexity of the model checking problem with respect to anonymous models is not exponential in the number of agents, which is it for non-anonymous models. This entails that we could model check properties of models consisting of a very large set of agents. Other key results include complete axiomatizations of the resulting logics under certain assumptions of the anonymity exhibited by the model.

One method of introducing heterogeneity to an otherwise homogeneous system, is by letting agents enact different norms. We show that model checking is still tractable when the number of agents are taken as a parameter. This result entails that we can
introduce an arbitrary level of heterogeneity, while curbing the complexity of model checking.

Furthermore, this framework offers optimistic outlook for cases where compliance can not be guaranteed or assumed because the underlying homogeneity ensures a form of monotonicity, not present in existing models of partial norm compliance, which enables the reduction of some forms of coalition quantification by canonical representatives. This has the immediate consequence that questions of robustness of normative systems may be answered more efficiently.

Being able to do the same, does not entail doing the same. We provide a logical formulation characterizing a theory of property-based preferences. The theory attempts to capture how agents form their preferences based on how they weigh various combinations of properties and which of these properties they are motivated by.

This theory is one among several recent theories which attempt to ground the agents’ preferences in their cognitive state. Not only does it give more structure to the agents’ basis for decision making, but it may also be useful to explain differences in behavior between similar agents.

The characterization we provide is expressed in a well-known logic which makes the formulation suitable for inclusion into other models of behavior, or analysis by existing tools and methods.

What reasons an agent bases her preferences or decision on ultimately is in many cases the result of deliberation in which various alternative reasons may be considered. We provide a system for reasoning about the possible evolutions of such a deliberation under which the agent has several conflicting possible views. The views may also be those of different agents disagreeing but seeking to find a compromise or a stable way of merging their views in a consistent way. We show that even though the agents’ views may contain an infinite number of possible reasons we may verify several properties of all possible ways the agents may arrive at under very weak assumptions.
List of papers

The technical results presented in this monograph are either published in articles listed below, or are the result of my continuation of work I have done in collaboration in the articles listed below.

1. Truls Pedersen, and Sjur Dyrkolbotn. *Computing consensus: A logic for reasoning about deliberative processes based on argumentation* [63].
   Available at: http://arxiv.org/abs/1408.1647

2. Truls Pedersen, Sjur Dyrkolbotn, Piotr Kaźmierczak, and Erik Parmann, *Concurrent Game Structures with Roles* [64].
   Available at: http://arxiv.org/abs/1303.0792

3. Truls Pedersen, Piotr Kaźmierczak, and Sjur Dyrkolbotn. *Big, but not unruly: Tractable norms for anonymous game structures* [65].
   Available at: http://arxiv.org/abs/1405.6899


5. Truls Pedersen, and Sjur Dyrkolbotn. *Agents homogeneous: A procedurally anonymous semantics characterizing the homogeneous fragment of ATL* [67].
## Contents

**Acknowledgments** ..... i

**Abstract** ..... iii

**List of papers** ..... v

### 1 Introduction

1.1 What are Multi-agent Systems? ..... 2

1.2 Voting ..... 5

1.3 Games and Game Theory ..... 7

1.3.1 Utility, Preferences and Goals ..... 8

1.4 “What” and “how” ..... 10

1.5 Social Laws and Norms ..... 13

1.6 Strategic Logic ..... 14

1.7 Homogeneity ..... 18

1.8 Expressivity and Computational Complexity ..... 19

1.9 What do we want? ..... 20

1.10 Why study Multi-agent Systems? ..... 21

### 2 Technical Preliminaries

2.1 Notation ..... 23

2.2 Propositional logic ..... 24

2.3 Modal logic ..... 25

2.3.1 Multi-modal logic ..... 27

2.4 Alternating-time Temporal Logic ..... 29

2.4.1 Model checking and size of models ..... 31

2.5 Many-valued logic ..... 32

2.5.1 Łukasiewicz (three-valued) logic – $Ł_3$ ..... 32

2.5.2 Kleene’s (strong) three-valued logic – $K^S_3$ ..... 33

2.6 Abstract Argumentation Theory ..... 34

### 3 Representation

3.1 Background ..... 39

3.2 Homogeneity ..... 39

3.3 Homogeneous Roles ..... 41

3.4 Roles ..... 42

3.5 Concurrent Game Structures with Roles ..... 45
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5.1</td>
<td>Which train is in the tunnel? – Actions are identified extensionally</td>
<td>49</td>
</tr>
<tr>
<td>3.6</td>
<td>New Semantics for ATL</td>
<td>53</td>
</tr>
<tr>
<td>3.6.1</td>
<td>Equivalence between RCGS and CGS</td>
<td>55</td>
</tr>
<tr>
<td>3.7</td>
<td>Model checking and the size of models</td>
<td>58</td>
</tr>
<tr>
<td>3.7.1</td>
<td>Model checking using roles</td>
<td>59</td>
</tr>
<tr>
<td>3.8</td>
<td>Summary</td>
<td>62</td>
</tr>
<tr>
<td>4</td>
<td><strong>Axiomatic Reasoning</strong></td>
<td>65</td>
</tr>
<tr>
<td>4.1</td>
<td>Background</td>
<td>65</td>
</tr>
<tr>
<td>4.2</td>
<td>Constellations</td>
<td>65</td>
</tr>
<tr>
<td>4.3</td>
<td>The logic HATL</td>
<td>69</td>
</tr>
<tr>
<td>4.4</td>
<td>Completeness of HATL</td>
<td>70</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Axiomatization</td>
<td>72</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Local construction</td>
<td>74</td>
</tr>
<tr>
<td>4.4.3</td>
<td>Eventuality Realization</td>
<td>78</td>
</tr>
<tr>
<td>4.4.4</td>
<td>Final model construction</td>
<td>81</td>
</tr>
<tr>
<td>4.5</td>
<td>The logic RATL</td>
<td>83</td>
</tr>
<tr>
<td>4.5.1</td>
<td>Equivalence of axiomatizations – Anonymized ATL</td>
<td>84</td>
</tr>
<tr>
<td>4.6</td>
<td>Relationship between HATL and RATL</td>
<td>85</td>
</tr>
<tr>
<td>4.7</td>
<td>Summary</td>
<td>88</td>
</tr>
<tr>
<td>5</td>
<td><strong>Norm compliance</strong></td>
<td>91</td>
</tr>
<tr>
<td>5.1</td>
<td>Background</td>
<td>91</td>
</tr>
<tr>
<td>5.2</td>
<td>Normative systems and compliance</td>
<td>92</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Norms and normative systems</td>
<td>92</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Norms and compliance defined</td>
<td>94</td>
</tr>
<tr>
<td>5.2.3</td>
<td>Reintroducing heterogeneity – the canvas society</td>
<td>95</td>
</tr>
<tr>
<td>5.3</td>
<td>The logic NCRATL – Norm compliance RATL</td>
<td>97</td>
</tr>
<tr>
<td>5.4</td>
<td>Characterizing ($\eta \upharpoonright C$)-compliant $A$-profiles</td>
<td>99</td>
</tr>
<tr>
<td>5.4.1</td>
<td>Profiles as flows</td>
<td>99</td>
</tr>
<tr>
<td>5.4.2</td>
<td>Computing $P_C^\eta(q,A)$</td>
<td>102</td>
</tr>
<tr>
<td>5.5</td>
<td>Model checking</td>
<td>104</td>
</tr>
<tr>
<td>5.6</td>
<td>Anonymizing the language</td>
<td>105</td>
</tr>
<tr>
<td>5.6.1</td>
<td>Quantifying over coalitions instantiating a constellation</td>
<td>106</td>
</tr>
<tr>
<td>5.6.2</td>
<td>The logic NCHATL</td>
<td>110</td>
</tr>
<tr>
<td>5.6.3</td>
<td>Model checking NCHATL</td>
<td>112</td>
</tr>
<tr>
<td>5.7</td>
<td>Summary</td>
<td>113</td>
</tr>
<tr>
<td>6</td>
<td><strong>Reason-based Preferences</strong></td>
<td>115</td>
</tr>
<tr>
<td>6.1</td>
<td>Background</td>
<td>115</td>
</tr>
<tr>
<td>6.2</td>
<td>Reason-based preferences</td>
<td>117</td>
</tr>
<tr>
<td>6.3</td>
<td>A modal logical account of reason-based preference</td>
<td>119</td>
</tr>
<tr>
<td>6.4</td>
<td>The characterization</td>
<td>122</td>
</tr>
<tr>
<td>6.5</td>
<td>The multi-agent case</td>
<td>128</td>
</tr>
<tr>
<td>6.6</td>
<td>Finding the closest consensus</td>
<td>130</td>
</tr>
<tr>
<td>6.7</td>
<td>Summary</td>
<td>132</td>
</tr>
</tbody>
</table>
7 Deliberative Dynamic Logic
  7.1 Background ........................................ 135
  7.2 Difference and Dialogue in Argumentation ........ 135
  7.3 Deliberative Dynamic Logic ......................... 140
  7.4 Model checking on finitary models ................. 147
  7.5 Recognizing Consensus ............................ 151
  7.6 Summary ............................................. 152

8 Conclusions ............................................ 153
<table>
<thead>
<tr>
<th>Table/Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>The prisoners’ dilemma</td>
<td>7</td>
</tr>
<tr>
<td>1.2</td>
<td>The prisoners’ dilemma as a tree (c (cooperate) and d (defect))</td>
<td>8</td>
</tr>
<tr>
<td>1.3</td>
<td>Correspondence between utility and preference relation (( u_a ) is ( a )'s utility function)</td>
<td>9</td>
</tr>
<tr>
<td>1.4</td>
<td>A snippet from a run in LTL</td>
<td>15</td>
</tr>
<tr>
<td>1.5</td>
<td>A snippet from a model of CTL</td>
<td>16</td>
</tr>
<tr>
<td>1.6</td>
<td>A snippet from a model of ATL</td>
<td>16</td>
</tr>
<tr>
<td>2.1</td>
<td>Truth tables for three connectives in Ł_3</td>
<td>33</td>
</tr>
<tr>
<td>2.2</td>
<td>Truth tables ( \rightarrow ) in Ł_3 and K_3^S</td>
<td>34</td>
</tr>
<tr>
<td>2.3</td>
<td>A sketch of an argument.</td>
<td>35</td>
</tr>
<tr>
<td>2.4</td>
<td>An abstract view of an argument. Edges in the graph indicate opposition.</td>
<td>35</td>
</tr>
<tr>
<td>2.5</td>
<td>Example argumentation framework, ((A, E))</td>
<td>38</td>
</tr>
<tr>
<td>3.1</td>
<td>Example of an “anonymous” CGS state. ( A, B ) denotes change in file ( A ) or ( B ), resp. ( C ) denotes conflict. Labels ( a, b ) and ( w ) denote the possible actions. As every label denotes an arrow, there are nine (non-dotted) arrows in this figure.</td>
<td>40</td>
</tr>
<tr>
<td>3.2</td>
<td>Example of potentially heterogeneous versus guaranteed homogeneous set of successor states.</td>
<td>41</td>
</tr>
<tr>
<td>3.3</td>
<td>Transition label in CGS models (left), and RCGS models (right).</td>
<td>47</td>
</tr>
<tr>
<td>3.4</td>
<td>Train controller model for ( n_t ) trains. ( n_t ) is the (constant) number of trains. (Example 3.5.)</td>
<td>49</td>
</tr>
<tr>
<td>3.5</td>
<td>The outgoing edges from ( q_0 ) designate a single train. (Example 3.6.)</td>
<td>51</td>
</tr>
<tr>
<td>3.6</td>
<td>Number of outgoing transitions from ( q_0 ) in CGS vs. RCGS in example model.</td>
<td>51</td>
</tr>
<tr>
<td>3.7</td>
<td>Detailed example</td>
<td>52</td>
</tr>
<tr>
<td>3.8</td>
<td>States, transitions and role assignment for the entire model of Example 3.6.</td>
<td>53</td>
</tr>
<tr>
<td>4.1</td>
<td>Axioms of HATL for every ( \sigma, \sigma' \leq \Sigma )</td>
<td>72</td>
</tr>
<tr>
<td>4.2</td>
<td>Inference rules for HATL for every ( \sigma \leq \Sigma )</td>
<td>72</td>
</tr>
<tr>
<td>4.3</td>
<td>Example of complete profile with six agents in two equally sized roles ( (\Sigma = (3,3)) )</td>
<td>75</td>
</tr>
<tr>
<td>4.4</td>
<td>Constructing and composing trees in proof of Lemma 4.19</td>
<td>80</td>
</tr>
<tr>
<td>4.5</td>
<td>Axioms of RATL for every ( A, B \subseteq A )</td>
<td>84</td>
</tr>
<tr>
<td>4.6</td>
<td>Inference rules for RATL for every ( A \subseteq A )</td>
<td>84</td>
</tr>
</tbody>
</table>
4.7 RCGS with fixed social structure, but not state independent role assignment. ........................................... 86

5.1 Illustration of coordination by compliance. (Example 5.3.) ...................................................... 96

5.2 (Compliant) profiles as flows. \( C = \{b_1\}, B = \{b_1, b_2\}, \) and \( \eta_q(b_1) = \eta_q(b_2) = \{2\} \). ................................................................. 103

5.3 Reasoning about norm compliance of constellations. ................................................................. 112

6.1 The relation \( \leq \) and the corresponding preference orderings (an arrow from \( A \) and \( B \) means that \( A \leq B \) and reflexive loops are omitted). ................................ 118

6.2 \( T_M := \{A1\} \cup A2 \cup A3 \cup A4 \cup A5 \) ................................................................. 124

6.3 Property based \( L: \rho(M_L) = L. \forall S \in S_M \exists S' \in DL: \rho(S) = \rho(S') \) and \( S := S' \) ................................................................. 128

6.4 The agents’ preferences of fruit ................................................................................................. 130

6.5 Weighing (loops omitted) ........................................................................................................ 130

7.1 \( V_a \) ........................................................................................................................................ 137

7.2 \( V_b \) ........................................................................................................................................ 137

7.3 Three consecutive states in the deliberation process. .............................................................. 138

7.4 Four potential successor states in the deliberation process. ...................................................... 138

7.5 Simple argumentation framework. ............................................................................................. 143

7.6 Simple basis, \( B = \{V_a, V_b\} \). Showing \( V_a \) on the left, and \( V_b \) on the right. 146

7.7 A fragment of the deliberative Kripke model \( \mathcal{K}(B, e) \) for \( B \) and arbitrary \( e \). 146

7.8 For \( B = \{V_a, V_b\} \), we have \( \mathcal{K}(B, e), (\emptyset, \emptyset) \models [p]q \) and \( \mathcal{K}(B, e), (\emptyset, \emptyset) \models [p][q]q \). ................................................................. 147

7.9 Is \( \neg p \) part of a consensus? .................................................................................................. 151
Chapter 1

Introduction

The study of multi-agent systems can be broadly characterized as the study of interacting computing elements. This thesis concerns mainly computational aspects of such systems. Several important problems become computationally hard to answer when the number of agents becomes large. A main theme in this thesis will be how exploiting similarities and symmetries in the set of agents can lead to more efficient problem solving for large number of agents. The exploitation of these symmetries intuitively seem to reduce the ability from models to capture differences among agent. However we may reintroduce desirable differences in various ways without relinquishing the computational benefits of our succinct representation.

In this chapter we will discuss the main ideas that has lead to the foundation of modern logical approaches to reasoning about and modeling multi-agent systems.

The development will be grounded in a historical outline and end with some reflections on a possible future. The chapter starts with describing what we mean by “multi-agent systems”. The definition is not in terms of modal logic, which is the main framework used throughout the thesis, but rather more general. Then we will exemplify multi-agent by describing roughly how voting systems are investigated mathematically in Section 1.2 on Voting Theory.

In Section 1.3 on Game Theory we will see how we can describe situations which are in a sense broader than what we considered in voting theory. Here we also introduce the notion of agents’ utilities. Towards a logical description, we will need to make some reflections on the relationship between what agents want, and how they bring it about. We will try to outline the difference between these two endeavors and show how they are often interdependent in Section 1.4. This section will also touch upon one alternative to utilities; reason-based preferences – the topic of Chapter 6.

Having described a “how” and a “what”, we need a way to regulate our system; a way of implementing our desired behavior. In Section 1.5 on Social Laws and Norms – the main topic in Chapter 5 – we elucidate, in broad strokes, how we can do exactly that. In Section 1.6 we put the components together and describe how we can use logic to reason about the systems we have constructed thus far. We will also make some preliminary observations about the usefulness of such logics and an important computational challenge. Several of the results in this thesis concern homogeneous systems. What we mean by a homogeneous system will be described in Section 1.7, and explicated in Chapters 3 and 4.

Having described strategic logic and homogeneity, we discuss briefly some com-
putational issues that arise in strategic logic and indicate how homogeneity can be a flexible and quite general way of avoiding complications anticipating the complexity results we will return to in Chapters 3 and 5.

Then in Section 1.9 we discuss what differences remain under the assumption of homogeneity and indicate one dimension of difference which can be (and often needs to be) maintained even in homogeneous systems as will be elaborated on in Chapter 7. Finally, this chapter concludes in Section 1.10 where we attempt to synthesize what we have discussed throughout the chapter and make some reflections on the relevance and future for multi-agent systems.

This entire chapter is written with as few formal definitions as can be afforded. All the technical results are confined to the remaining chapters, each of which will present some technical results on what we discuss intuitively in this chapter. This level of (or lack of) precision will undoubtedly leave some statements less precise than possible, and some notions less clear than desirable. Hopefully it will also make the intuition and motivation easier to grasp for the reader which is only passing through our field. For the reader more interested in the precise technical results of this thesis, there will be references throughout this chapter to the later chapters which describe the topic in more detail.

The title of this thesis, agreement and cooperation under degrees of homogeneity in multi-agent systems, invites us first to ask “What are multi-agent systems”?

1.1 What are Multi-agent Systems?

Multi-agent systems can be described as [82]

“[...] systems composed of multiple interacting computing elements, known as agents”.

A multi-agent system (MAS) can be a suitable characterization of models from a number of different disciplines; from concurrent programming, to game-theory, to philosophy of social interactions.

What exactly characterizes a MAS according to this description is that the agents are interacting, i.e., that they have the ability to communicate, or manipulate each other or their common environment in some way. Furthermore they are computing, i.e., they are able to perform some kind of information processing (reasoning) based on their observations of themselves, each other, and their common environment.

In the work presented in this thesis we will also use the term “agent” abstractly, but let us now try to ground this abstract description to aid our intuitions. What constitutes an “interacting computing element”, or the agent? An agent can be a computer connected to other computers in a network. Each computer has complete control over the execution of its own program, but is able to interact with the other computers by sending/receiving data to/from other computers through the network. The computer is, in principle (depending on the program it is executing) free to do what it wants with the input it receives from the other agents. If the computer is programmed to ignore all input we end up in a deterministic evolution of the system, in fact we could more eas-
1.1 What are Multi-agent Systems?

ily simply study the individual computers in isolation.\footnote{This might not be literally true, as even a computer program in isolation might rely on non-deterministic actions. However, the execution (modulo indeterminacy in the execution of actions) will be deterministic.} Important in the description is that the agents are interacting. They respond to input and produce output for the other agents to react to. Let us then consider that the agents are responding to each other for a reason.

Consider a network of computers tasked with solving a large computational problem. The computers might now be set up to try to partition the problem into smaller sub-problems and distribute the responsibility of solving these simpler problems. Later they should combine their solutions to a solution of the larger problem again. This is the key idea behind distributed computing; a network of computers working together as a much more powerful computer — an organized system. The organization of these computers however is not at all trivial and in many applications of distributed computers we find a hierarchy of so-called “slaves” and “masters”. Having certain elements act as masters simplifies the inter-agent organization greatly, and coordination becomes much easier to realize, compared to self-organizing systems.

Self-organizing systems manage to coordinate their actions without this hierarchical structure. If two autonomous computers can benefit from collaborating, they might not need a master to tell them how to organize themselves. Suppose two computers have come to realize that they both need to compute the solution to some complex problem which can be decomposed into two simpler problems. If the program(s) they are executing permits a very high level of interaction, they might agree to split the problem, solve one part each, and finally reveal the solution to the other party. This hypothesised scenario relies on a lot of interaction: each computer must be “aware” of the other’s predicament, they must agree which of them should be responsible for which sub-problem, and depending on the circumstances they must trust that the other computer will reveal the solution when the cooperation comes to an end. A general, real-world computer network tasked with solving massive problems would need to solve these intricate additional problems. The problems resemble problems we face in everyday life, such as negotiation, coordination, and commitment.

In going from the hierarchical situation to the more “autonomous” one, exactly which problems arise? How do we handle these complications? We had an implicit assumption that the two computers in question had some goal that they were trying to achieve. Presumably they had something to compute and implicitly we also assumed they wanted to compute this efficiently (otherwise they wouldn’t have needed to cooperate in this case). They had freedom to decide which actions to perform, and in this case they reasoned that it was in their best interest to cooperate. This autonomy which provides them the necessary freedom to decide to cooperate might also allow them to defect on the agreement. When is it in each agent’s best interest to cooperate? Even if both computers reason that cooperation and commitment is the best course of action, they will not necessarily agree to cooperate unless they also know that the other computer is motivated in this way.

It is tempting to transfer our findings from MAS to human affairs, and as we will see in the next sections, our models inherit many of its components from fields which do not discuss computers in a network. Notably the constructions we inherit from game theory which is continually being applied and tested by economists as a tool for
predicting human behavior as well as that of chimps and other non-human animals. Towards the end of this chapter we will return to what we can understand as an agent, and discuss what discoveries in MAS can be seen as applying to. In this thesis we will generally use the term “agent” to refer to a computer, or computer component, but we will take some freedoms particularly in this chapter. If we interpret this description broadly and apply it in retrospect we could trace the study of MAS a long way back, even if we insist on limiting our attention to formal or mathematical investigations.

Careful studies of at least particular MAS, understood quite generally, have had a long history. Theories of how a society should be organized can in many cases be categorized as a MAS problem. How do we go about, for example, answering the question “What does society want”? Even if we know the answer to this question, it is naturally followed up by the question “How can we achieve it, if at all”?

In this general sense, one could argue that particular MAS studies have been carried out as long as we have asked ourselves what is good for society and how to bring it about [9]. Questions of this form do not always lend themselves to straightforward mathematizations or simple calculations. It can be wise to avoid mathematising these questions if this leads us to have too high confidence in conclusions ultimately based on oversimplified assumptions. Experimental economists endeavor to harmonize the predictions game theory offers about human behavior in many ways.

On the one hand, using game theoretic models normatively, economists try to understand how our observed behavior differs from the predicted rational behavior and thus understand what we, the subjects, deem valuable and to our advantage and how our cognitive faculties’ estimate of various consequences differ from those of a computer or a rational agent. On the other hand, correcting game theoretic models used descriptively to try to capture observed behavior thereby attempting to provide insight into the actual workings of our decision making apparatuses. Does the model capture reality? Do the subjects perceive the essential parts of reality in a way we can analyze? Does our projection of the subjects’ value-estimates preserve their judgment?

All these empirical questions require difficult measurements themselves, and it is beyond the scope of this thesis to address these challenges. Even though the agents we will mostly refer to here are most easily thought of as computers or computer programs, we should not let this lead us into complete skepticism towards the mathematization of some of these problems. As Amartya Sen said: “Arrow’s impossibility theorem – in many ways the ‘locus classicus’ in this field – can hardly be anticipated on the basis of common sense or informal reasoning” [73]. The result he is discussing is Arrow’s seminal “Impossibility Theorem”, stating that certain combinations of desirable properties can not be realized by any social choice theory; a procedure for aggregating the agents’ preferences into a decision about the society’s preferred outcome.

In the following sections, we will describe how some social phenomena have been inspiring to the study of MAS in computer science and artificial intelligence (AI). But we feel it would be poorly grounded to argue that the inspiration is and will always proceed in one direction. In other words, the study of MAS is a sub-discipline of computer science and AI which often borrows techniques and constructs used to study human affairs and rests some of it’s intuitions and assumptions on our reflections on human behavior. This enables us for example to design software which can aim to solve high-level goals (such as solving a complex problem) without micro-managing the various computer components to the last detail of how to coordinate their actions.
1.2 Voting

The results of formal investigations of MAS would carry over again to the fields where we find our inspiration whenever the assumptions we need to make to incorporate these mechanisms into computer systems also apply there.

Attributing ancient philosophers’s work as works in MAS might be using the terms too broadly, and even though particular systems of social decision making have been studied and invented several times through the ages, the field of voting theory is arguably more easily justified as a study of MAS.\(^2\) Around the time of the French revolution, several intuitively plausible systems for social decision making received thorough scrutiny particularly by the mathematicians Marie-Jean de Condorcet and Jean-Charles de Borda [9].

1.2 Voting

Voting theory has a long history. Several questions are immediately interesting about systems of voting. Is there always a clear winner? Is a particular voting procedure prone to manipulation? The answers to such questions divides voting procedures into those that exhibit the property in question, and those that do not. There are many properties we might want to verify that these systems satisfy. Such as whether the procedure will yield (elect) the majority preference when there is one. What if there is unanimous agreement on the preferred outcome? Or on the other extreme we might ask if the process is dictatorial, i.e., whether a single agent can dictate the outcome?

Studying voting systems formally, we generally let the agents have a set of preferences over the alternatives. At this level of abstraction these are usually the only properties we need to know about the agents. The agents also have a set of possible choices; depending on the procedure we want to study the agents may be allowed to choose a single candidate alternative to vote for, a ranking of the top three alternatives, a complete ranking of all the alternatives, or whatever is appropriate for the system we are interested in. Then we specify the outcome of the vote by describing mathematically which alternative is declared the winner given any possible way the agents might choose to vote.

We can see that voting systems satisfy the description of MAS above, but the level of interaction and computation is quite limited. The interaction consists solely of casting a vote and the computation performed by the agents amounts to each agent figuring out what she is going to vote. This computation, however, need not be easy. The environment we assume the agents themselves form may have implications on what the agents know about, for example, the other agents’ preferences and intentions.

A particularly important property we often want our voting systems to satisfy is that of anonymity. Etymologically anonymity could be described as “without a name” and often in everyday language we understand this as having one’s identity hidden, that observers are unable to identify the actor of an anonymous action. Anonymity in voting theory, however, is generally understood as a property which a voting system satisfies if, and only if, it is irrelevant which agent voted for what.

To formalize this we can not let the system be dependent on the identity of the voters when computing the winner, but of course if the voters behave only slightly different,
the voting system is permitted to yield a different end result. What this formulation of anonymity results in is the statement that if any two agents swap their ballots before submitting their votes, the result will be the same.

A voting system hence, is anonymous if, and only if, it is invariant under permutation\(^3\) of its arguments (the voters’ ballots). This description is very structural. It pertains to how the result is found, rather than the agents’ identities directly. Processes which satisfy this structural property will generally prevent observers from discovering what the different agents voted under very weak assumptions except in fringe cases. The colloquial understanding of the word “anonymity”, on the other hand, is of a more epistemic nature pertaining to what we can know about the agent acting. In this other way of using the word, we are describing what the observer can discover about the acting agents’ identities regardless of other assumptions.

Regardless of whether a system is anonymous (in either understanding of the word), it may be susceptible to manipulation. By manipulation in voting theory we mean that instead of voting for our top ranked candidate or, in other words, truthfully revealing our preferences, we report false rankings in order to try to achieve an outcome which the agent prefers to the outcome which would have resulted had she reported her true preferences. Now we need to separate between the agents’ preferences and the agents’ actions. Each agent has a number of properties, like their preferences, and also some number of actions available to choose from.

Now we have to take into account that our agents have an implicit agenda: ensuring that the outcome of the voting process is one which they prefer as much as possible. To see why it could be beneficial for an agent to strategize in this way, consider that the agent has to vote between three alternatives, \(A\), \(B\) and \(C\). The agent prefers \(A\) to \(B\) and \(C\), and \(B\) to \(A\). We denote an agents preference by a special relation symbol \(\prec\) (or sometimes \(\preceq\)). If we write \(X \prec Y\) (or \(X \preceq Y\)), we mean only “\(Y\) is better than \(X\)” (or “\(Y\) is at least as good as \(X\)”). So our agents preferences could be expressed as

\[
C \prec B \text{ and } C \prec A \text{ and } B \prec A
\]

When we want to specify that we are addressing the preferences of a particular agent, we will subscript the relation symbol with her name. That is, if for some agent \(a\) prefers \(A\) to \(B\), we denote this \(B \prec_a A\).

Our agents most preferred alternative is \(A\). But suppose our agent knows that in this particular voting scenario, \(A\) is very unlikely to win and it is a pretty close race between the alternatives \(B\) and \(C\). The agent might reason that she had better vote for \(B\) rather than \(A\) in an attempt to minimize her loss. The computational part of our model becomes more obvious. The agent has some insight into the environment (such as the workings of the voting system, perhaps the other agents’ preferences, and so on), and attempts to create a strategy which will lead to the best possible outcome for her.

\(^3\)Invariant under permutation is a slightly technical term. Certain functions or procedures are said to be invariant under permutation. When the result of the function (or the output of the procedure) remains unchanged whenever we arbitrarily switch around the arguments/input. Addition is a simple example of a function which is invariant under permutation \((2 + 3 = 3 + 2)\), and division is not \((2/3 \neq 3/2)\). When the operation is binary, this property is often called commutativity.
1.3 Games and Game Theory

Game Theory gained its modern form in the middle of the last century, particularly through the work of von Neumann and Morgenstern [59]. The main object of study is a mathematical structure simply called a game. Such games consist of a number of agents, each with a number of actions available to them. Depending on the type of game we are studying the agents choose actions to perform, or play, and the game terminates in some outcome. Each agent also has preferences over the possible outcomes. An agent’s preference enables us to say, relative to the given agent, whether any two possible outcomes are equally preferred or which of the two outcomes the agent prefers to the other. Hence the outcomes, which are determined by the agents’ choice of actions, are totally ordered for each agent.

There are many forms of games and we can use game theory to analyze both cooperative and non-cooperative behavior of agents which are assumed to be rational to some degree. These technical nuances are not important to us now, but see for example [61] for further details. What is important to us is the component which game theory introduces as key to the analysis: strategies. The agents will choose their actions in a rational way with respect to their preferences over the possible outcomes.

The prisoners’ dilemma is a very popular example of a game. In this game we have two agents: the prisoners. The prisoners have been caught by the police after committing some crime and been placed in separate interrogation rooms. The police have enough evidence to get them convicted for breaking in, but not for taking the documents which were allegedly stolen. The documents were very valuable and it is important that the police uncover what happened to them. The investigators give each of the prisoners the same choice: if they betray their cohort they will receive a reduction of their prison sentence.

The break-in would give them a year in prison and the theft would yield an additional two years. Observe that the agents are not permitted to communicate before making their decision. We can illustrate the thieves’ situation in Table 1.1. The first thief (a) must choose a row, and the other (b) must choose a column. Their respective choices will identify a particular cell containing a pair of numbers. The first number is the utility for a, and the second the utility for b. We will get back to utility very shortly. Here it reflects the number of years in prison.

<table>
<thead>
<tr>
<th></th>
<th>cooperate</th>
<th>defect</th>
</tr>
</thead>
<tbody>
<tr>
<td>cooperate</td>
<td>(-1, -1)</td>
<td>(-3, 0)</td>
</tr>
<tr>
<td>defect</td>
<td>(0, -3)</td>
<td>(-2, -2)</td>
</tr>
</tbody>
</table>

Table 1.1: The prisoners’ dilemma

This is obviously a very simple sketch of the situation, but it illustrates a couple of things. First of all, the outcome is completely determined by the choices of the agents. Each agent has, for any two possible outcomes a preferred outcome. But this game permits only a limited amount of interaction. Many other games, or indeed a repeated version of this game, requires the agents to respond to other agents’ actions or the outcome of their own earlier actions. Representing the previous game as a tree as in Figure 1.2 we can illustrate the situation slightly differently.
In this alternative representation the notion of states appears. In this illustration we have five states, the initial state $q_0$ drawn in a solid box, and four terminal states $q_1, q_2, q_3, q_4$. We can interpret the labeled arrows going out from the initial state as a state transition. Looking at the labels we can also verify that the successor state (one of the terminal states) is uniquely identified by the actions the agents choose to perform. We could easily imagine that there could also be intermediate states in this tree which are neither initial or terminal, but which could constitute a restriction on the possible outcomes, i.e., prevent us from being able to reach certain terminal states. We will pursue the power of this idea when we describe strategic logics in Section 1.6.

Notice in the description of this game that if we permute the actions the agents performed as well as the utilities in the outcome, the game turns out identical to the original game. Games for which this can be done are called symmetric games and have been considered important since the inception of game theory [25, 59]. The description of symmetry is thus very similar to the definition of anonymity we saw in voting theory. It is not the same since we now also require that the agents’ utilities are permuted along with the action. To check if a game is symmetric, look at every transition e.g., $(c,d)$, and record the utilities $(-3,0)$. If we switch the actions about and look at the utilities we would then obtain we see that $(d,c)$ leads to a state with utilities $(0,-3)$. This holds for all permutations of actions/utilities. If we drop the requirement about permuting the utilities along with the actions we end up with anonymous games.

A game is said to be symmetric if, and only if, permuting actions/utilities does not change the game, and a game is said to be anonymous if, and only if, permuting the actions does not change the game (or in other words the outcome is invariant under permutation of actions) [17, 25]. So we see that anonymity in voting theory and game theory are the same concepts. After some reflections on utility, we will abandon game theory and focus on logics. What we have just discussed about anonymity will reappear in a few sections, but then as a notion of homogeneity (in Section 1.7).

1.3.1 Utility, Preferences and Goals

Often, e.g., in economic applications of game theory, we will quantify the agent’s view of the possible outcomes. We do not always need to do this. In the prisoners’ dilemma example above we could have said either
1. The agent prefers to stay one year in prison to staying two years, or
2. It is twice as bad for the agent to stay two years in prison compared to one year in prison.

In the example game above, we assigned a pair of numbers to each outcome which is a common way of presenting the example. The numbers corresponded to the number of years in prison, but this can be misleading in that we seem to reduce the utility of the agents to the mere number of years in jail.

The numbers assigned to the outcome states need not necessarily correspond to any particular value derived from the particular states of affairs. In some cases, particularly when time also needs to be considered, numbers can incorporate how various evolutions of the game compare to one another. That is, when agents’ preferences are based on how utility is accumulated, how it depreciates, and so on, the vaguer statement (1. above) might not provide us with enough information. When modeling a phenomenon in order to make a prediction on the basis of it, it can be important that the relationship between the utilities prescribed are correctly measured.

The quantification of such utilities is often very difficult to arrive at and justify. Often the precise number is not relevant, however. Often the most important information about (the agents’ perception of) the outcomes is that the numbers assigned to the states match the agents’ own preferences. A common assumption about the structure of preferences permitting us to make this simplification is that, given two possible outcomes, the agent is able to answer whether the two outcomes are equally preferred or, if not, which of them is preferred to the other. Whenever this assumption is satisfied, we permit ourselves to reduce this total relation\(^4\) by simply assigning numbers to the outcomes. We can see such a correspondence (for agent \(a\)) in Figure 1.3.

![Figure 1.3: Correspondence between utility and preference relation \((u_a a’s utility function)\)](image)

When we assign numeric utility by some map from the terminal states to numbers we also specify a total ordering of the outcomes, but not necessarily the other way around. That is, for every single preference ordering, there are several (infinitely many) utility assignments which yields exactly that preference ordering, but for every assignment of utilities, there is exactly one ordering of preferences corresponding to it. In this thesis, we will not use utility in our investigations. In Chapter 6 we look at how we could also derive preferences through another route. There we will present a logical characterization of reason-based preferences proposed by Dietrich & List [26, 27].

---

\(^4\)In mathematics, an ordering of the elements in some set, is another set: a set of statements of comparisons. When describing the preferences of a voting agent we said \(C\) is worse than both \(A\) and \(B\), and \(B\) is worse than \(A\). When every element is either equally good or either better or worse than every (other) element, we say the ordering is total. (For the pedantic: it is implicit that each element is equally good as itself, i.e., that it is reflexive. It is also generally assumed to be transitive.)
These preferences will be derived not from the utility agents enjoy in the different states, but the (qualitative) properties the states/outcomes have.

In other chapters, for example the logic presented in Chapter 4, the agents will not be endowed with preferences at all, but rather we investigate what agents could do to achieve their goals. Rather than describing the agents’ preferences or their utility in the possible outcomes and ask how they will behave, we describe a collection of the outcomes which satisfies their goal and ask if they can achieve their goals and if so, how.

### 1.4 “What” and “how”

Considering what we just discussed about reasoning about agents’ behavior with respect to a given set of goals, we can see that one problem we are attacking is similar to the question from the beginning of the chapter: *How can we achieve our goal, if at all?* Also we have tried to hint at how the reasoning applied in the problems of voting theory – which in many cases is exactly posing the question “What do we want”? – can be extended to reasoning about more involved strategic situations.

Again with broad stokes, the synthesis of these two problems into one cohesive description, yields a simple description of a certain problem [82]:

*“Practical Reasoning = Deliberation + Means-Ends Reasoning”*

This is a tabloid reduction of a massively complicated set of problems. It is fitting however, to reflect on the mutual dependence between the two issues. In order to answer the question “What do we want?”, we need a system, or protocol, which produces the answer based on the agents’ views. This, of course, would have to be a system making use of the actions (e.g., voting, or expressing opinions) of the participating agents. Such a system is therefore also a system which addresses the question “How do we reach a state complying to society’s goals, if possible?” where society’s goal in this case is to find a consensus.

Agreeing *how* we find out what is a good candidate for a common goal for a group of agents may be as important as what that goal actually is, or turns out to be. The method of forming a conclusion about what we want as a common goal might be what gives the conclusion credibility. Rules imposed on an agent through a process which does not protect the agents’ interests are more easily trespassed; *why* should an agent follow a certain set of rules? If the system through which we agree on common goals is a process which she perceives as working in her favor, the legitimacy of the rules – which are ultimately a restriction of her freedom of choice – may motivate the agent to adhere to them.

Democratic processes attempt to take the views of all the participants as a basis for their conclusions; this characteristic in itself suggests that agents generally could

---

5A single goal is a very simple form of preference – a dichotomous one. In this case, the alternatives are separated into the alternatives which satisfy the goal, and those that do not. When an agent has several goals, the combination of these may be more complicated or there may even be incomparable sets of alternatives.

6This is not an entirely correct description of what we do in the mentioned logic. More precisely we could say that the problem posed there is rather, “If some description of a set of desirable states of affairs had been some set of agents’ goals, could they achieve that goal regardless of the other agents’ behavior?”
support deliberative processes which are democratic. Obviously, there is no easy way of agreeing on which process, even among the purported democratic processes we should choose, and so on. In matters of practical considerations, we need to choose among the many candidates of how to come to agreement about our common goal(s).

These processes aim to incentivize the participants to be active; and principally here is compliance to the conclusions via the process’ legitimacy. On the other hand, a completely egalitarian democracy might very well, in some cases, make categorically worse decisions than a benevolent dictator. If two cars are approaching an intersection at the same time, it is (nearly?) universally understood that the best choice the drivers have is to suspend their freedom of choice and let the traffic light dictate their behavior. This highlights a possible conflict of interest. The driver wants to arrive at her destination as quickly as possible, but is expected to relinquish some of her freedom to facilitate coordination – coordination which of course also is beneficial for her, even though it is not necessarily a goal for her in itself.

Other situations, concerning more complicated dilemmas such as how to handle the challenges related to human induced climate change, raises questions about the efficacy of certain democratic processes. The climate strategist Jørgen Randers, for example, writes on the challenge of climate change in [71] that:

“Democracy and capitalism both share an attraction for tradition. [...] In order for the world to move briskly, there is the need to break with traditional approaches and form new partnerships that are able to agree within reasonable time limits.”

We will not even attempt to discuss issues such as democracy in this thesis, but the analogy should help ground our understanding of the often difficult balances we need to handle when we are dealing with problems which are both complex and time-critical. The problem itself is complex in nature, and when we want solutions to emerge from within a large group of autonomous agents, we get the added problem of computational complexity (we return to this issue shortly, in Section 1.8).

Many very complex social problems are very difficult particularly because of the mutual dependence between answering such “how” and “what” questions in an iterated way. As we will see in Sections 1.5 and 1.6, these problems arise when reasoning about concurrent computer systems and agent-based programming. The need for systematic ways of studying such phenomena are recognized in political science. They use models which in many ways are similar to game theoretic models, but with more emphasis of strategic interactions rather than the solution concepts conventionally employed in game theory, such as the proposal of “systematic process analysis” [41] in political science. See also [12] for an overview of various similar approaches.

As the internet is becoming an ever more important arena for our social interactions in practically any sphere of communication, we get a truly astonishing new source of data which can inform our thinking about complex social interactions. Through the internet we are systematizing many tasks by organizing large numbers of people around problems which computers alone can not solve.

Two trends are becoming increasingly popular and visible in our everyday lives, as well as core components of both communication and business; big data and social networking, where large amounts of data are processed by powerful computer systems and large numbers of people interact socially via computer systems, respectively. The
stage is set for what has been dubbed *social machines*. Berners-Lee describes a possible means to solve complex problems [13] (see [42]):

“Computers can help if we use them to create abstract social machines on the Web: processes in which the people do the creative work and the machine does the administration [...]”

Such social machines are being implemented and investigated particularly in the interesting applications where we know of no computational way of arriving at good results (see [78] for an overview). Some of these systems aim to let humans do small computational tasks which computers are simply not very good at (at the moment?), such as classifying images of galaxies (Galaxy Zoo), or creating a “market” for minor tasks for which computers are not adequate (Amazon Mechanical Turk). Other aspects of these social machines emphasize and facilitate social interaction without necessarily “solving” a specified goal. Such systems allow (practically) everybody to engage in public debates (Facebook, YouTube, Twitter, a plethora of forums, and so on).

All these systems, although they enable large-scale discussions of virtually any topic, by “lifting” the standard forms of such modes of communication to large groups of people, have been criticized for not working efficiently in large groups. There are several reasons for this, see for example Malone & Klein [58] where conventional debate formats, but with a large number of participants, is criticized for being ineffective due to the time-centric version of communication. The hypothetical case study they investigate is the debate of human induced climate change. It is, as mentioned, very complex in nature and it invokes a lot of involvement. The standard way of communicating orderly, participant by participant, becomes inefficient when the number of participants becomes too large.

They call for computers to perform an administration which is different from how public debates are usually/classically administered. In the mentioned paper, and particularly Klein [51], a particular system is suggested, namely an “issue-centric” forum based, in large part, on argumentation theory. The system they are proposing – the Deliberatorium (ibid) – changes the way the participants engage in debate.

We agree with the suggestion that models which replicate classical communication modes, do not scale well (e.g., for being too time-centric or by drowning views held by minorities). In the Deliberatorium these issues are tackled by centering the discussion around *topics* or *issues* rather than individual expressions. In Chapter 7 we present a logic we call Deliberative Dynamic Logic (DDL) which attempts to model argumentative, and subjective, points of view in a systematic way. We will return to this issue later in this chapter, in Section 1.9, but will first introduce some more technical constructs we will rely on in our analysis.

In addition to a multitude of empirical challenges there are computational challenges. We face the problems sketched here in our everyday life, and similar issues are increasingly becoming important for computers cooperating to solve computational tasks when they are asked to solve problems which are described in higher level terms, rather than as concrete imperative “execution orders”. We will not address the applicability of the formalisms discussed in this thesis to problems besides computer systems, but to the extent that they do prove applicable we will provide some optimistic results about the computational complexity of certain important problems for these models (particularly in Chapters 3 and 5).
1.5 Social Laws and Norms

Consider two firms in negotiation. They both have some benefit if they agree to cooperate, but on each firms’ part there is also some cost associated with the cooperation. Each party will have to consider whether they suspect the other party will defect on an agreement, perhaps to reap the benefits of cooperation without suffering the full cost of the cooperation. In making these considerations they can rely on the laws which apply. In analyzing the situation game theoretically, (the consequences of) laws can often be incorporated in the model as a modification of the participants’ utility in the various outcomes, but this is only possible because the law is an exogenous given. The law itself must also be formulated (and implemented) in some way.

The notion of a law has many sides, each of which offers its own very challenging problems. We will not delve into shaping and interpreting laws as studied by jurists, and even within the formal applications of laws we will focus on only one, albeit broad, definition. In game theory and social choice theory, a related concept is that of a mechanisms. We start with a model of a situation and a given set of desirable outcomes. The challenge is to amend the model by a mechanism which ensures that the outcome of rational behavior is desirable. Somewhat conversely but more in line with how we will model situations in strategic logic (introduced in Section 1.6), we are also interested in the consequences of a law. By consequences here, we must quickly add that these are strategic consequences. We will not study deontic aspects of the regulations in question, like punishment, reward, and so on.

Again, we are in a similar situation to the firms in negotiation. The law is given as something they can rely on when considering how the other party will behave. The term which best describes the construct we will study is contentious, and for good reason. The phenomenon which inspires it is a vast and important one, and is centrally important in many fields of study. However, in MAS the term “normative system”, often shortened to “norm”, seems to be settling as an acceptable term, but must always be applied with reservation and caution.

Normative systems were introduced to MAS in Shoham & Tennenholtz’ seminal paper [75] under the name “social laws” and later developed in several papers (see e.g., [1, 43, 76], see [74] for a recent introductory text). In the example of a game theoretic model we saw that in the initial state, the actions of the agents dictated the outcome of the game. In such a situation a normative system will deem certain actions illegal. The general definition of a normative system in this interpretation is to say that it is some procedure which, given a state, an agent, and one of the agent’s available actions, tells us if this action is forbidden or not (for that agent in that state). A normative system is hence a method which restricts the agents’ possible actions in some way. This understanding has been applied to MAS-based analyses of social phenomena. See for example [83] where it is applied to study a social contract.

Borrowing further from the inspiration of the construct, suppose we are in some state where for some agent a certain action is deemed to be “illegal”. The deontic statement describing the situation might be something like “the agent ought not perform that action now (i.e., in this state)”, not flat out making it illegal or impossible as our defini-

\[^7\] A similar, but distinctly different, definition of a normative system will deem certain outcomes as undesirable or forbidden. We will only investigate normative systems which exclude actions.
tion here implies. However, if we consider our agent to be lawful in the sense that she would not violate the norm, then the action will not be a viable option for her. In order to reason also about the possibility of violating the norm, the usual way of reasoning in MAS is by separating the set of agents into two disjoint subsets; the compliant and the non-compliant agents. This is incorporated into the models by expanding them to also describe norm compliance. See for example [5] for a logic of norm compliance.

Continuing this tradition, this is exactly how we will model norms in Chapter 5.

This dichotomous view of compliance separates us from the regular normative considerations we encounter in everyday life. “I ought not have this cigarette, therefore I will not” is perhaps not a statement every human can be said to live by. However, in the more computational situations this model is very reasonable. In a system of distributed computers where many things are possible (the actions available) and certain things are illegal (actions our computers should not perform), we can expect that all our computers will behave “lawfully” while any intruder will make decisions only based on what is possible.

The scenario of firms relying on laws in order to facilitate cooperation alludes to how normative systems can be conducive to coordination in an autonomous MAS, and they are proposed in MAS to aid coordination [43, 75, 76, 84]. In certain circumstances we can design a normative system “offline”. In other cases the norms can be emergent, i.e., result from the agents’ choices [82]. Implementing a normative system is a commitment to some answer to the aforementioned “how” question, and simultaneously a restriction of the future outcomes (which might rule out answers to both the “how” and the “what” question). Consider the norm stating that (all) choices of (all) agents should be anonymous. This can be achieved by a system (consisting only of compliant agents) by disallowing any action (in any situation) which would set any one agent apart from another. Anonymity, and as we shall see homogeneity, can hence be achieved by implementing an appropriate normative system. Conversely and, as we will discuss shortly and which is the main topic of Chapter 5, it is often more efficient to introduce heterogeneity by disallowing certain actions for some agents in a model with completely homogeneous agents. In this way, we can modify the behavior of agents to either increase or decrease the degree to which the agents can be said to be homogeneous.

1.6 Strategic Logic

When we discussed game theory, we saw (initial, terminal and intermediate) states, through which we progressed by means of transitions determined by the actions of the agents. In terminal states we had a summary description of the utility of the given state to each of the agents. Now we will add more structure to the states themselves and describe them by the properties they possess. Change in properties of the state of affairs introduces the idea of (logical) time. Reasoning about temporal statements (“There will be a war next year”) dates back at least to Aristotle’s initial investigations of reasoning. The initiation of modern investigations of temporal logic is often attributed in

---

8Logical time emphasizes the possibility of something “coming after” something else. It does not necessarily make any ontological commitments about (the nature of) physical time. Some properties such as “discreteness” applies directly to computers operating at some frequency (i.e., in discrete steps), but not necessarily physical time.
a large part to Arthur Prior (see [16] for a brief overview of the modern temporal logics). We will not concern ourselves with history here and rather sketch the systems of temporal logic which have been adopted by the computer science community.

If we follow the evolution of a system, from its initial state, through the intermediate states and (if the evolution ends) to some terminal state, we get a sequence of states which describes the history or the run, of a system. Suppose we have an entire such history at hand, together with some designated state among these which is the current one. In the same way as we labeled each terminal state with a utility earlier, let us now rather add a label of the properties it satisfies. If it is raining, say, in some state, we add “raining” to the description of this state. If one of these happens to precede the current state, we can say (relative to the current state) “it has rained”, and if some state which succeeds the current state satisfies this property, we can say that “it will rain”. Treating the evolution of the system at hand as a sequence of states in this way is the essential semantic construct of linear-time temporal logic (LTL).

Using LTL – one temporal logic – to reason about the correctness of computer programs was introduced by Pnueli in [68] (see [36]) several decades ago to describe the evolution of the states a computer goes through when executing a program, a run as illustrated in Figure 1.4.

![Figure 1.4: A snippet from a run in LTL](image)

A shortcoming of LTL is that the future is determined; the system is closed. When reasoning about non-deterministic operations, such as when modeling the execution of non-deterministic operations, a model of a closed system is unsatisfactory. When a computer attempts to write some data to permanent storage, in “most cases” we end up with the data stored. Occasionally, however, the data is either corrupted or the storage device is not responding correctly for some reason. As opposed to the reasoning we did about a computer executing its operations in isolation we can no longer give a full account of the state of affairs by knowing the previous state and the operation. The possible runs branch out to create two (or more) possible successor states (see Figure 1.5). This was studied around 1980 and the de facto logical framework for reasoning about such non-deterministic computer programs, Computational Tree Logic (CTL/CTL*), was introduced by Clarke, Emerson, and Halpern in the early 1980’s [23, 34].

This lead to decades of research into using formal methods to verify the correctness of computer programs. Verifying correctness of complex programs is a difficult task and the effort comes in addition to the actual development. But the extra cost of verification is worth the price in some cases. When testing is impossible, such as for the NASAs deep space missions, or difficult and costly, such as Intel’s hardware development, formal methods have become an integral part of business. See e.g., [32] for an overview of some experiences with the difficulty of applying, and cost of not applying, formal methods in deep space exploration projects, or [38] for some reflections on the

---

9Intuitively, one might expect that a storage system will actually work most of the time, but strictly speaking we only make the observation that both correct and incorrect operations are possible without qualifying which is more likely.
Branching time logic has a high level of similarity with the models of game theory. On the verge of the last century, Alur, Henzinger and Kupferman introduced Alternating-time Temporal Logic (ATL) which gave a game theoretic semantic to branching time logic\textsuperscript{10}.\textsuperscript{10} ATL and the extended version ATL\textsuperscript{*} provided a refinement of how the possible futures comes about. We add actions to each agent in each state as we did in game theory. See Figure 1.6 for an example. Suppose we are in the boxed state $q_0$. The two agents have two actions each (as in the prisoners’ dilemma example).

\textsuperscript{10}The cited article is the most recent of the three articles with the same name, by the same authors. The first published some years earlier. There is no difference between the proposed logic which is relevant to us in this chapter.
for the agent(s) which is/are not better of than any other agent”. Examining the arrows we can easily see that the two agents have a joint strategy to ensure that they end up in a state described by this label; if they both perform the action $c$ (cooperate), we transition from $q_0$ to $q_{1,1}$. Indeed, we might realize that whenever the successor state is uniquely determined by the actions of the agents, there is always a strategy for the agents to get to any particular possible successor state as long as all the agents cooperate\footnote{This does not necessarily mean that the agents are aware of what actions they need to perform in order to arrive to the designated state. Also, in some system models it is reasonable to assume that the agents do not have complete knowledge of the labels in the various states. These details are explored in epistemic extensions of ATL and are not important to at this point.}

There are three states containing the label $r$; these are $q_{1,2}$, $q_{1,3}$ and $q_{1,4}$. The first agent has a strategy (or choice of action), namely choosing the action $d$ (defect), which ensures that we end up in a state labeled $r$. We can verify this by checking which states we transition to for all (both) possible choices the second agent may make. We will often phrase this as “agent 1 can enforce that in the next step $r$ (is satisfied)”. We also saw how one agent could limit the possible outcomes other agents could reach when we discussed game theory. However, being able to enforce $r$ (in the next state), is not a special ability of only the first agent. The second agent is also able to guarantee that the system ends up in a state satisfying $r$, as we can easily verify.

The states $q_{1,3}$ and $q_{1,4}$ are the only states which satisfy $s$. If we examine the action pair labels on the transitions arrows carefully, we notice that the first agent has a strategy to ensure that the next state is one of those two, i.e., the first agent can enforce $s$ by choosing action $d$ (defect). This is not the case for the second agent. Again, careful examination of the action pair labels reveal that no matter what the second agent does, the first agent can choose her action such that we transition into a state that does not satisfy $s$. The agents have different strategic abilities. The complementary notion, that of having the same strategic abilities, is the topic of the Section 1.7 on homogeneity.

Before we examine this notion in more detail, let us observe now what we are modeling. We have a number of differently named\footnote{We mean “named” in a technical way. Our abstract entities do not have “names” in the sense that humans have names, but rather they are indexed in some way. We have a reference to each agent. We have called them “the first” and “the second agent” here, but we could just as well have called them “agent 1” and “agent 2”, or $a$ and $b$, or Frank and Hank for that matter. Each agent being uniquely identifiable by a reference is the important aspect in this discussion.} agents, each of which has a number of actions available to them (depending on the state of the system). Moreover, the next state of the system (i.e., the evolution of the system) depends only on the agents’ choice of action.

These models, just as CTL models, are suitable for describing systems of concurrent actions, but we gain the ability to discuss the components individually. That is, we can query our models whether, say, some specific set of agents are able to make certain progress regardless of the behavior of the rest of the agents. If we are concerned with computers in a network, this progress might be to maintain a certain description of the system (e.g., “the system is active”), avoid certain descriptions (e.g., “the storage system is offline”), or (eventually) reach some description (e.g., “equation X has been solved”).

This added expressivity comes at a cost. Even though checking queries on both CTL and ATL models are tractable (i.e., feasible for a computer to compute), this is not the case if we let the number of agents be a factor. Jamroga & Dix asks “Do agents
make model checking explode (computationally)?” in an article by the same name [46]. They point out that when systems for which we want to perform such queries become larger, by adding additional agents to the system, the computation becomes intractable.

As we have discussed earlier, this is exactly one phenomenon we are also seeing in the development of systems e.g., supporting the notion of social machines. We will discuss the technicalities of how we can curb this intractability by relying on the set of agents being (to some degree) homogeneous throughout several of the chapters in this thesis. Particularly, in Chapter 3, we will see how the basic case of homogeneous ATL permits tractable model checking (answering queries) even if the number of agents grow. In Chapter 5 we show how we can decrease the strict requirement of homogeneity by introducing social laws which differentiate the agents.

1.7 Homogeneity

When ATL was introduced\(^{13}\) [6] each agent was attributed with an individual set of actions, or action indexes, in each state. They were clearly a (potentially) heterogeneous bunch. Permitting heterogeneity does not rule out homogeneity, of course; it remains an interesting special case. Similar systems of concurrently acting agents have been discussed in many works. In the work which seminally introduced social laws in reasoning about concurrent computer processes, the set of agents in the system onto which a normative system would be imposed or implemented, were described explicitly as homogeneous\(^ {14}\) [75]:

“We make an assumption of homogeneity; specifically, we assume that the sets of states and the available actions are common to all agents”.

This form of homogeneity, as we can see, is certainly a constraint in a direction which seems like homogeneity, but two agents which have “the same actions” available might execute these actions differently. As there are no specifications/restrictions of the actions, what happens when one agent performs some action (say “push the blue button”) might be very different from what happens when another agent performs the same action. Perhaps in one case, the agent who pushes the button receives $10, and if the other pushes the button, it will unlock a door. Are these really the same action? Do these agents have the same actions available? If we identify actions extensionally rather than intentionally, we have another notion of “homogeneous agents”. If there is a blue button such that if some agent pushes it, she receives $10, then another agent, which has the same actions available, may also push the same button, yielding $10 to the first agent. The two agents in such a scenario certainly can be said to have the same actions available. In the work presented here (Chapters 3, 4, and 5), this extensional identification of actions will give rise to the homogeneity of main interest.

We might also distinguish between these two definitions by calling the former “structural homogeneity” and the latter “strategic homogeneity”. The type, or notion,

\(^{13}\)In this thesis we will rely on the Concurrent Game Structure (CGS) semantics introduced in [6], and not the Alternating Transition Systems (ATS) semantics used by the same authors in other articles.

\(^{14}\)Presumably, what was intended by “homogeneity” in this quote is a technical simplification which is of little significance for the technical result, except making the exposition simpler. This notion of homogeneity permits modeling any heterogeneous system, as will be indicated currently. When we shortly will introduce strategic homogeneity, it will be clear that it is a special case of the notion of homogeneity mentioned in this quote.
of homogeneity which we employ will have consequences for which normative systems
will be of interest later, and we will see how the interplay between these notions, and
their corresponding normative systems, will let us vary the “degree of homogeneity”,
from the completely homogeneous to the completely heterogeneous.

The fact that agents are even strategically homogeneous (the stronger form), does
not entail that the agents are completely indistinguishable. The model of strategic rea-
soning about homogeneous agents we present in Chapters 3 and 4, and at least to some
degree in Chapter 5, are silent about the agents’ goals and preferences. Hence their be-
havior will be different. Which action two agents of a strategically homogeneous group
choose may very well be different. Even though they both have the same actions avail-
able, with the same exact consequences, and they are completely rational and informed
about the game, their choice of actions may very well be different. That is to say, they
are not copies of each other, but have identical power over their environment and each
other.

1.8 Expressivity and Computational Complexity

After capturing the relevant details in our model, we need to ask the relevant questions.
In the previous sections we have sketched some them. What can we expect from our
voting mechanism? What can which sets of agents achieve if they cooperate? What
regulations do we need to impose on the system to guarantee that (as long as a suf-
cient number of agents comply) good cooperation is possible, or bad cooperation is
avoidable?

To answer these questions we design a formal language which is able to express
the questions we want to pose, and we define the meaning of the various parts of the
language in terms of our models. Some of the interesting language constructs will
be rather complicated, though. Consider, intuitively, if we want to check whether some
small group of agents are able to eventually ensure that some outcome will come about.
It is not unlikely that our model needs to contain a lot of states to capture the situation
we have in mind, and in each of these states\textsuperscript{15} we need to try to synthesize a common
strategy for our protagonists which will lead us towards the desired outcome. The
strategy needs to work regardless of how the remaining agents behave.

The more fine grained level of detail our models include and the more expressive
our language becomes, the more questions we are able to answer about the system. But
there is an immediate trade-off here. The added expressivity generally comes at a cost
of computational complexity. That is, the more we are able to say about our models,
the longer it will (generally) take for a computer to answer our queries. How long it
will take a computer is not measured in minutes or seconds, generally. This would not
provide a lot of insight. Rather what we are interested in is questions similar to “if I
have a computer which can solve some problem sufficiently quickly, can I also solve a
larger version of the problem”? We are interested in how much more time it takes as
the problems scale, or in other words, how the time required grows asymptotically as
the size of our input (usually the model and the question/query) grows.

\textsuperscript{15}We might not need to consider all of them, if our protagonists are able to avoid certain states, but these details
are not relevant now.
The problem of answering “yes” or “no” to the question “does this model satisfy this description” is a useful and essential question to ask. It is often called the *model checking* problem. In Chapter 3 we will see how the model checking problem for our *homogeneous* versions of ATL is never worse than the general (heterogeneous) version, and that it often *scales* better when we increase the population. That is, where model checking classical ATL models becomes very difficult with a large number of agents, we are much better off in the homogeneous case. And in Chapter 5 we expand our querying language to permit questions of (partial) norm compliance in addition. Again we will see that we can do this in a computationally efficient way. This means that we can reintroduce some level of heterogeneity among the agents and remain computationally efficient.

When we want to test our systems for a large number of agents, it is more likely that there will be some level of homogeneity. If we are analyzing a situation with only a few agents, they might very well have different abilities. But the larger our set of agents become, the more likely it is that more than one agent fits the characterization needed for our model. If the only property of interest is, say, how fast an agent can run, it is not unlikely that two agents will have different capabilities. If we have thousands however, several agents might fall into the same category up to indistinguishable differences. The homogeneous version of strategic reasoning we develop in the following chapters will exploit exactly this property to provide better scaling as the population in our model grows.

### 1.9 What do we want?

The agents in the frameworks we will investigate in this thesis are initially strategically homogeneous, but not identical. For instance, in many situations we do not anticipate the agents to behave the same way. A way of formalizing and explaining the difference in behavior is that the agents might be motivated by different desires. We have already discussed briefly in Section 1.3.1 the attribution of different preferences to different agent, and even a system of explaining the differences by means of sets of properties of the possible outcomes that the different agents might be motivated by. This can get us some way towards evaluating formal models able to characterize and understand the origin of disagreement, but we also need a method of dealing with disagreement when we want, for example, to coordinate our efforts.

Chapter 6 attempts to make sense of differing preferences when the agents are focusing on various reasons they may be motivated by. But which reasons apply when we are trying to construct a description of a common goal? Which reasons are the important reasons? We hope the analysis we present in Chapter 6 will shed some light on how differing preferences may be united in a framework wherein the agents might be fundamentally in agreement, but *seem* to disagree as explained by *non-preferential* differences (e.g., epistemic, doxastic, computational limitations).

Our analysis is based on our formalization of the work on *reason-based preferences* by Dietrich & List [26, 27]. We suggest how we can reason about *apparent*, or *superficial*, disagreement without departing from the assumption of *fundamental* agreement, by letting non-preferential properties of agents influence their reported preferences.

In Chapter 7 we present a formal framework, Dynamic Deliberative Logic (DDL)
1.10 Why study Multi-agent Systems?

Google has got smart cars driving unaided by humans in regular traffic. Amazon is delivering small packages by autonomous robots whizzing through the air. NASA has had a semi-autonomous rover exploring Mars for ten years, and recently the China National Space Association placed a similar rover on the moon. In less robotic scenarios, journalists see literally multiple perspectives of events that unfold through civilians with smart phones connected to the internet, and the go-to knowledge base, Wikipedia, is governed by “uncertified” authors working under a protocol of production which, contrary to its intuitive understanding, in some cases produces (arguably) better and more reliable data than the traditional encyclopedia written by authoritative experts. Much of our economy is being managed by computers, because human traders are simply too slow (particularly in the stock market), traffic regulations and logistic planning is becoming too complex for humans to execute efficiently. We even encounter robots in everyday life; they have already started cleaning our floors and mowing our lawns.

Because we will try to make as few assumptions as possible, many of the problems we will discuss in the following chapters apply to MAS in a fairly broad sense. However, the assumptions we do make will be more easily justified when considering for example autonomous computer components in a network, or even to some degree autonomous robots attempting to coordinate. Applying the analogy of computers evaluating a function in order to decide on which action to perform, selected from a clearly predetermined set of available action and executed in a well-behaved and deterministic system, allows us to relax some of the possible contingencies we would have to consider when hypothesizing about the consequences of human behavior. In the following chapters we find it useful to think first and foremost based on these simplifications. But humans interacting socially, politically and economically is closely related to these questions. The extension of a human is not limited to a body, but in a very important sense also by the reach of its tools.

We have a hope that this thesis conveys two overreaching statements. First, that the vocabulary of the MAS literature, which to a large extent is based on the vocabulary used to discuss human societies, is conducive to insight about how to reason about, and tackle, many kinds of complex problems. Arguably, some ways human societies are
shaped could be said to function well and be progressive. Perhaps the output of our continual process of refining our goals and protocols do not point directly at some ideal “solution”, but rather at a compromise of gradually decreasing cost. The balance between granting citizens autonomy versus granting them freedom from other citizens’ freedom of choice is perhaps best settled this way. As computer scientists, we ought perhaps seek out principles for organizing complex systems where we can find inspiration. At the same time, however, we must take care to test our formalisms carefully, and keep an open mind about the subtle nuances which have been revealed in the literature we let ourselves be inspired by. The notion of a homogeneous role presented in this thesis is a transfer of a term from a field quite distant from computer science. Hence, our treatment of it should not be taken as a clarification of the word as used in e.g., sociology. Rather, we have tried to capture a (small) part of the use and investigated the consequences of our attempt at formalizing it. We have been fortunate to find that there is indeed some merit to our understanding of the term. This brings us to the other statement we hope the thesis makes.

The second, more technical, statement, concerns the computational benefit of the role metaphor. The assumption of homogeneity is an obvious property to reflect upon when discussing MAS theoretically, and indeed it has occurred (and occurred with heterogeneity) since early in the tradition. It remains an important term and factor in many analyses. For instance, the homogeneity/heterogeneity distinction is pivotal in Stone’s overview article on MAS from an AI perspective [79]. This term, just as that of role, is not without its ambiguities and nuances, however. What precisely makes a group of agents homogeneous is that, to some degree of detail, they are similar. We are interpreting this as a group of agents being similar when they have exactly the same status in a system. This assumption sounds very strong, but, as we demonstrate in Chapter 3, we can still capture the classical logic of ATL for reasoning about (potentially) completely heterogeneous agents. In other words, the strong assumption we make do not limit our expressive ability, but only gives us computational benefits when we construct models, and when we query them for the properties generally asked about MAS systems.
Chapter 2

Technical Preliminaries

In this chapter we review the necessary of technical preliminaries.

2.1 Notation

Throughout the text we will make use of some abbreviations and notation. The most frequent conventions are given here.

We will make extensive references to the set of natural numbers from 1 to \( n \). We therefore denote this set \([n]\) for brevity. That is, we define

\[
[n] := \{1, \ldots, n\}
\]

This set is a subset even of the positive natural numbers. We define

\[
\mathbb{N} := \{0, 1, 2, \ldots\} \quad \text{and} \quad \mathbb{N}^+ := \{1, 2, 3, \ldots\}.
\]

We will be dealing with very many functions of many arguments. We therefore find it useful to add some abbreviations concerning function notation. For sets \( A \) and \( B \), a function is a set \( f \subseteq A \times B \) such that for every \( a \in A \) there is at most one \( b \in B \) such that \((a, b) \in f\). For sets \( A \) and \( B \) a total function \( f \) from \( A \) to \( B \) is a subset of \( f \subseteq A \times B \) such that for every \( a \in A \), there is exactly one \( b \in B \) such that \((a, b) \in f\). We say that \( A \) is the domain and \( B \) is the co-domain of \( f \), denoted \( \text{dom}(f) \) and \( \text{cod}(f) \), respectively. A function which is not total is called a partial function. We will generally refer to total functions simply as functions. If \( f \) is a (partial) function from \( A \) to \( B \), we denote the function’s signature as \( f : A \to B \). If \( f : A \to B \) and \( a \in A \), we generally denote the result of applying \( f \) to \( a \) as \( f(a) \) (but see below).

We denote the set of functions from \( A \) to \( B \) by \( B^A \). If \( v \in B^A \) we occasionally treat this as a function \( v : A \to B \). And if \( B \) is finite we occasionally refer to \( v \) as a \( B \)-indexed vector. For example, let \( f = \{(a, 1), (b, 2), (c, 3)\} \in \{1, 2, 3\}_{\{a,b,c\}} \) be a function. We
occasionally write $f(a) = 1$ and occasionally $f_a = 1$, or display them as either

$$
f(x) = \begin{cases} 
1 & \text{if } x = a \\
2 & \text{if } x = b \\
3 & \text{if } x = c 
\end{cases}
$$

or

$$f = \langle 1, 2, 3 \rangle$$

In the latter notation, the ordering of the indexing is usually given implicitly. We generally refer to the element of such a vector by the corresponding function argument, but also occasionally by a number. In those cases, we specify an ordering on the domain such that there should be no confusion.

If the domain of a function is a product, we regard the function $f : A \times B \to C$ as being equal to the function $f' : A \to (B \to C)$ which satisfies

$$f(a, b) = f'(a)(b) \text{ for all } a \in A \text{ and all } b \in B$$

We will generally not give these functions different names, but treat them as one. Furthermore, if a function $f$ has a product domain as above, say $f : A \times B \to C$, we often abbreviate $f(a)(b) = f(a, b)$ to $f_a(b)$. That is, we treat the first argument as an “index” of functions.

## 2.2 Propositional logic

Propositional logic is a system for reasoning about inferences from, and to, (compound) propositions. When reasoning using propositional logic we assume a set $\Pi$ of atomic propositional symbols. The language of propositional logic is given by the BNF

$$\phi ::= p \mid \neg \phi \mid \phi \land \phi$$

We generally also permit ourselves to rely on the usual abbreviations, $\phi \lor \psi ::= \neg (\neg \phi \land \neg \psi)$, and $\phi \rightarrow \psi ::= \neg \phi \lor \psi$. We say that $\lor$ is the dual of $\land$.

An interpretation, or model, in propositional logic is a map $\pi : \Pi \to \{0, 1\}$. The evaluation of a formula is given inductively as an extension $\pi$ of an interpretation of the proposition symbols as follows.

$$\pi(p) := \pi(p)$$

$$\pi(\neg \phi) := 1 - \pi(\phi)$$

$$\pi(\phi \land \psi) := \min\{\pi(\phi), \pi(\psi)\} \quad \text{(alternative definition of } \land\text{)}$$

If $\pi$ is a model, we denote the fact that $\phi$ is true in $\pi$ (i.e., $\pi(\phi) = 1$) by $\pi \models \phi$. If $\phi$ is true in all models, we write $\models \phi$. We refer to the set of such formulas as the set of propositional tautologies. If every model which satisfies every formula $\phi \in \Phi$ also satisfies $\psi$, we say that $\psi$ follows from $\Phi$ and denote this $\Phi \models \psi$.

Given a set of axioms $\Phi$ and a set of inference rules $R$, we define the set $\vdash$ as in Definition 2.6 below.
Strictly speaking, we define a proof to be a sequence of formulas such that every formula in the sequence is either an axiom or the result of applying an inference rule to an appropriate number of premises which occur earlier in the sequence. If there is a proof of \( \phi \) (with \( \phi \) as its last formula), we denote this \( \vdash \phi \). We also define a proof with assumptions. If \( \Phi = \phi_1, \ldots, \phi_n \) is a set of assumptions, a proof with these assumptions is a sequence of formulas starting with these formulas, and every formula in the remaining sequence is either an axiom or the result of applying an inference rule to premises as specified above. If there is a proof of \( \psi \) from the assumptions \( \Phi \), we write \( \Phi \vdash \psi \).

We define two constant formulas \( \top \) and \( \bot \) to represent formulas that are always true and always false, respectively. If there is a proof that \( \phi \to \bot \), we say that \( \phi \) is inconsistent. A formula is said to be consistent otherwise.

For further references on propositional logic, we refer to any introductory textbook in logic, for example [48].

### 2.3 Modal logic

Modal logic adds formulas expressing modal statements to propositional logic. In the simplest case, a modal logic over a countable set of proposition symbols \( \Pi \), call this language \( \mathcal{L} \), is given by the BNF

\[
\phi ::= p | \neg \phi | \phi \land \phi | \lozenge \phi
\]

where \( p \in \Pi \).

The boolean connectives \( \neg \) and \( \land \) are interpreted as (boolean) negation and conjunction, respectively. The diamond connective \( (\lozenge) \) is often interpreted existentially, as a statement of possibility. That is, \( \lozenge p \) is often interpreted “it is possible that \( p \)”. We interpret such statements over Kripke models, where “it is possible that \( p \)” is encoded as \( p \) being true in some possible world. Particularly when we are discussing computer systems, “some possible world” should usually be interpreted as “some accessible state”. The state is then a description of the state of affairs, providing some valuation (true or false) to every atomic proposition. This state of affair can be accessible either by being the outcome of performing some computational task, or by some (hypothetical) reasoning, or by other means.

In the most abstract, we simply refer to these as possible worlds. The semantic structures are then defined as follows.

**Definition 2.3 (Model).** A model is a tuple \( M = \langle Q, R, \pi \rangle \) where

- \( Q \) is some non-empty set of states (or “possible worlds”),
- \( R \subseteq Q \times Q \) is an accessibility relation, and
- \( \pi : Q \to 2^\Pi \) is a valuation function.

A model is a frame with a valuation. The frame is only the structure of the accessibility relation. Hence, a frame is a pair \( \langle Q, R \rangle \) where the components are defined as above. We define the interpretation of formulas inductively as follows.
Definition 2.4. Given a formula \( \phi \), a model \( M = \langle Q, R, \pi \rangle \) and a state \( q \in Q \). We say that \( \phi \) is true in \( M \) at \( q \) (denoted \( M, q \models \phi \)) if, and only if (by induction on the structure of \( \phi \))

\[
\begin{align*}
M, q \models p & \iff p \in \pi(q) \\
M, q \models \neg \psi & \iff \text{not } M, q \models \psi \\
M, q \models \psi_1 \land \psi_2 & \iff \text{both } M, q \models \psi_1 \text{ and } M, q \models \psi_2 \\
M, q \models \Diamond \psi & \iff \text{there is a } q' \in Q \text{ such that } (q, q') \in R \text{ and } M, q' \models \psi
\end{align*}
\]

When a formula \( \phi \) is true in every state of some model \( M \), we say that the formula is true in that model and denote it \( M \models \phi \). A formula is said to be valid on a frame \( \langle Q, R, \pi \rangle \) at a state \( q \in Q \) if, and only if, for every possible valuation \( \pi : Q \to 2^\Pi \), we have \( \langle Q, R, \pi \rangle, q \models \phi \). Furthermore, it is said to be valid on a frame (denoted \( \langle Q, R \rangle \models \phi \)) if, and only if, it is valid in every state of the frame.

When we study a logic, it will be relative to some class of frames or models, \( S \). A formula which is true in every state of every such semantic structure (frame or model) is said to be valid, denoted \( S \models \phi \). If our structures (models or frames) are exactly all structures of interest, we often denote this \( \models_S \phi \).

This gives rise to a particular set of formulas: the tautologies or validities.

Definition 2.5. Given a language \( \mathcal{L} \) and a class of semantic structures \( S \), the set of validities of that logic w.r.t. that class of semantic structures is the set \( \models_S \).

The set of validities can occasionally be defined and reasoned about axiomatically, as defined below.

Definition 2.6. Given a set of axioms \( \Phi \subseteq \mathcal{L} \) and a set of inference rules \( R \), the set of theorems \( \vdash \subseteq \mathcal{L} \), contain

- all axioms \( \Phi \subseteq \vdash \), and
- whenever some formulas are theorems \( \phi_1, \ldots, \phi_n \in \vdash \) and there is a rule in \( R \) taking \( n \) premises such that it can be applied to \( \phi_1, \ldots, \phi_n \) deriving \( \psi \), then \( \psi \in \vdash \).

When \( \phi \) is a theorem (i.e., an axiom or the conclusion of an inference rule applied to theorems as premises), we denote this \( \vdash \phi \).

For any set of formulas \( \Phi \cup \{ \psi \} \subseteq \mathcal{L} \), we say that \( \psi \) can be deduced from \( \Phi \) if, and only if, there are formulas \( \phi_1, \ldots, \phi_n \in \Phi \) such that \( \vdash (\phi_1 \land \cdots \land \phi_n) \rightarrow \psi \), denoted \( \Phi \vdash \psi \). A set of formulas \( \Phi \) is said to be inconsistent if there is a formula \( \psi \) such that both \( \Phi \vdash \psi \) and \( \Phi \vdash \neg \psi \).

The relationship between theorems and validities is of crucial importance to a logic. Two properties inform us about important qualities of a logic.

**Soundness** All theorems are valid, i.e., \( \vdash \phi \Rightarrow \models \phi \).

**Completeness** All valid statements are theorems, i.e., \( \models \phi \Rightarrow \vdash \phi \).
We have defined a very sparse language earlier, and we will continue doing so whenever possible. There are some omitted connectives, but these omitted connectives can be defined in terms of the other connectives.

\[ \phi \lor \psi := \neg(\neg\phi \land \neg\psi) \quad \text{(disjunction)} \]
\[ \phi \rightarrow \psi := \neg\phi \lor \psi \quad \text{(material implication)} \]
\[ \Box \phi := \neg\Diamond\neg\phi \quad \text{(necessity)} \]

The connectives \( \Box \) and \( \lor \) are said to be the dual connectives of \( \Diamond \) and \( \land \), respectively. Dual connectives are inter-definable, the dual \( \nabla \) connective of some connective \( \Delta \) have the same arity, say \( n \), and satisfies \( \neg\nabla(x_1, \ldots, x_n) = \Delta(\neg x_1, \ldots, \neg x_n) \). See [16] for more details on modal logic.

### 2.3.1 Multi-modal logic

The logics discussed in this work are all multi-modal. In one chapter, we have a logic with exactly two modal connectives. The remaining logics will generally be extensions or variations of ATL (which we will discuss shortly).

The modal logic \( KT + U \) consists of two modal connectives, \( \Diamond \) and \( E \). The former corresponds to the the “possibility operator” introduced earlier, and the latter is a universally quantifying connective. \( \Diamond p \) states that “it is possible that \( p \)” (that is, \( p \) is true in some accessible world) and \( E p \) “it is possible somewhere that \( p \)”. When interpreting the \( E \), we consider the accessibility relation which reaches every state from every state. This is called “universal possibility”.

The language of \( KT + U \) is given by the BNF

\[ \phi :::= p \mid \neg\phi \mid \phi \land \phi \mid \Diamond\phi \mid E\phi \]

The dual connective of \( E \), is \( A \) (interpreted “in every possible world”). It is defined as an abbreviation \( A \phi := \neg E \neg\phi \). The models we are describing satisfy a structural condition on the accessibility relation, namely that it is reflexive, i.e., every world is accessible from itself.

The satisfaction of a formula \( \phi \) over a model \( M = \langle Q, R, \pi \rangle \) is defined by structural induction on \( \phi \):

\[
\begin{align*}
M, q \models p & \iff p \in \pi(q) \\
M, q \models \neg\psi & \iff \text{not } M, q \models \psi \\
M, q \models \psi_1 \land \psi_2 & \iff \text{both } M, q \models \psi_1 \text{ and } M, q \models \psi_2 \\
M, q \models \Diamond\psi & \iff \text{there is a } q' \in Q \text{ such that } (q, q') \in R \text{ and } M, q' \models \psi \\
M, q \models E\psi & \iff \text{there is a } q' \in Q \text{ such that } M, q' \models \psi
\end{align*}
\]

Without particular justification (see [16] for the full argument), we will state an
axiomatization of this system. The axioms are

\( (\text{Prop}) \) Enough propositional tautologies.

\( (K\Diamond) \) \( \Box(p \rightarrow q) \rightarrow (\Box p \rightarrow \Box q) \)

\( (T\Diamond) \) \( p \rightarrow \Diamond p \)

\( (Dual\Diamond) \) \( \Diamond p \leftrightarrow \neg \Box \neg p \)

\( (K_E) \) \( A(p \rightarrow q) \rightarrow (A p \rightarrow A q) \)

\( (T_E) \) \( p \rightarrow E p \)

\( (4_E) \) \( E E p \rightarrow E p \)

\( (5_E) \) \( E p \rightarrow A E p \)

\( (Dual_E) \) \( E p \leftrightarrow \neg A \neg p \)

\( (\Box \subseteq E) \) \( \Diamond p \rightarrow E p \)

The inference rules of the system are

\[
\frac{\vdash \phi}{\vdash \phi \rightarrow \psi} \quad (MP)
\]

Modus ponens

\[
\frac{\vdash \phi}{\vdash \Box \phi} \quad (Nec\Diamond)
\]

\( \Diamond - \) Necessitation

\[
\frac{\vdash \phi}{\vdash A \phi} \quad (NecE)
\]

\( E - \) Necessitation

\[
\frac{\vdash \phi}{\vdash \phi_{p/\psi}} \quad (US)
\]

Uniform Substitution

In the application of uniform substitution (US), \( \phi_{p/\psi} \) denotes the formula \( \phi \) except that every occurrence of the propositional symbol \( p \) has been replaced by the formula \( \psi \). \( p \) and \( \psi \) may be chosen arbitrarily.

The set of axioms given above, together with the inference rules, define a set of theorems \( \vdash \) which is exactly the set of validities \( \models \). The argument for such claims, are usually separated into two parts: a proof of soundness and a proof of completeness of the axiomatization.

To show soundness of an axiomatic system it is enough to verify that every axiom is valid, and that every rule preserves validity from premises to conclusion (i.e., when we apply any rule of inference to premises which are valid, then the conclusion is also valid). From this we would conclude that \( \vdash \phi \Rightarrow \models \phi \).

To show completeness is generally more complicated. In Chapter 4 we give a completeness proof based on an simple reformulation of the completeness property, and in Chapter 6 we rely on this together with the proof of completeness for the axiomatization just stated. The reformulation says that every consistent formula is satisfiable.

\[\text{1We generally just say "enough propositional tautologies" to avoid explicitly referring to actual axioms of propositional logic. To be perfectly proper, in this case, we could have added the three axioms } (A_1) \ p \rightarrow (q \rightarrow p), \ (A_2) \ (p \rightarrow (q \rightarrow r)) \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r)), \text{ and } (A_3) \ (\neg p \rightarrow \neg q) \rightarrow (q \rightarrow p). \text{ The set of theorems would be the same.}\]
That is, we prove that from the assumption that a formula is consistent, it is satisfiable (true somewhere in some model). Formally, we show that, for every formula \( \phi \), if not \( \vdash \neg \phi \) (i.e., \( \phi \) is consistent), then \( M, q \models \phi \) for some model \( M \) and some state \( q \) in \( M \). This is equivalent to showing that for every formula \( \phi \), if for every model \( M \) and every state \( q \) in \( M \), it is not the case that \( M, q \models \phi \), then it is the case that \( \vdash \neg \phi \) (i.e., \( \phi \) is inconsistent).

Checking the definition of satisfaction we can see that the antecedent in this conditional is the same as saying that for every model \( M \) and every state \( q \) in \( M \), it is the case that \( M, q \models \neg \phi \). We can also see that every formula is semantically equivalent to itself negated twice,\(^2\) and provably equivalent with itself negated twice.\(^3\)

In Chapter 4 we show completeness by showing that every consistent formula is satisfied by some model. We show this by actually specifying how to construct a model that satisfies the given formula.

### 2.4 Alternating-time Temporal Logic

One particular multi-modal logic, is that of \textit{Alternating-time Temporal logic} ATL, which we discussed in the previous chapter. The language is not only parameterised by a set of propositional symbols, but also a non-empty set of agents \( \mathcal{A} \).

As discussed in the previous chapter, the interpretation of ATL formulas are closely related to what sets of collaborating agents can achieve regardless of the remaining agents’ choice of action. A set of agents \( A \subseteq \mathcal{A} \) is called a \textit{coalition}.

The language of ATL, as presented in [6], specified in BNF is defined as follows.

**Definition 2.7.** The language of ATL is the following:

\[
\phi ::= p \mid \neg \phi \mid \phi \land \phi \mid \langle A \rangle \bigcirc \phi \mid \langle A \rangle \Box \phi \mid \langle A \rangle \phi \cup \phi
\]

where \( p \) is propositional letter, and \( A \) is a coalition of agents. We follow standard abbreviations discussed earlier, and also define \( \langle A \rangle \hat{\bigcirc} \phi := \langle A \rangle \top \cup \phi \).

The language of ATL is evaluated over \textit{concurrent game structures}.

**Definition 2.8 (Concurrent Game Structure).** A CGS is a tuple \( M = \langle \mathcal{A}, Q, \Pi, \pi, A, \delta \rangle \) where

- \( Q \) is a non-empty set of states,
- \( A \) is a finite, non-empty set of agents,
- \( \Pi \) is a countable set of proposition symbols, and
- \( \pi : Q \rightarrow 2^\Pi \) is a valuation function.
- \( A : Q \times \mathcal{A} \rightarrow \mathbb{N}^+ \) denotes the number of actions, in a given state, is available to a given agent. For each state \( q \), the set of complete profiles \( P(q) \) is a vector of numbers given by

\[
P(q) := \prod_{a \in \mathcal{A}} A_{\mathcal{A}}(q)
\]

\(^2\)Every model \( M \) and state \( q \) in \( M \) satisfies \( \phi \) if, and only if, it satisfies \( \neg \neg \phi \).

\(^3\)Each of \( \phi \) and \( \neg \neg \phi \) is derivable from the other.
• \( \delta : Q \times P \rightarrow Q \) is the transition map. For every state \( q \in Q \), \( \delta_q : P(q) \rightarrow Q \), assigns to each complete profile in \( q \) a unique successor state.

To formalize the satisfiability of ATL formulas, we need to introduce some more terminology for describing agents’ choices.

The definition of a complete profile is given in Definition 2.8. For a coalition \( A \subseteq \mathcal{A} \), an \( A \)-profile in a state \( q \) is a vector of numbers. The set of \( A \)-profiles at \( q \) is gathered in the set \( P(q,A) \) defined by

\[
P(q,A) := \prod_{a \in A} A_a(q)
\]

For every state \( q \) and two coalitions \( A, B \) such that \( A \subseteq B \), if \( v_A \in P(q,A) \) and \( v_B \in P(q,B) \), then we say \( v_B \) extends \( v_A \) if, and only if, for every \( i \in A \), \( v_A(q)(i) = v_B(q)(i) \). We denote this \( v_A \leq v_B \). The elements of such a profile are simply numbers (action indexes). A strategy for a coalition \( A \subseteq \mathcal{A} \) is a map \( s_A : Q \times A \rightarrow \mathbb{N}^+ \) specifying for every state \( q \in Q \) and agent \( a \in A \), some \( A \)-profile for that state (i.e., number \( s_A(q,a) \in A(q,a) \) for every agent \( a \in A \) corresponding to an action that agent performs in that state). We denote the set of \( A \)-strategies by \( \text{strat}(A) \). If there are coalitions \( A \) and \( B \), such that \( A \subseteq B \), we say that a \( B \)-strategy \( s_B \) extends an \( A \)-strategy \( s_A \) if, and only if, for every state \( q \), \( s_A(q) \leq s_B(q) \). We denote this \( s_B \geq s_A \).

In every state \( q \), a strategy \( s \in \text{strat}(\mathcal{A}) \) for all agents, yields a unique successor state \( \delta(q,s(q)) \). Starting in any state \( q \), any \( \mathcal{A} \)-strategy defines an infinite sequence of states called a computation \( \lambda = q_0, q_1, q_2, \ldots \) where \( q = q_0 \) and \( q_{i+1} = \delta(q_i,s(q_i)) \). The computation resulting from the \( \mathcal{A} \)-strategy \( s \) starting in state \( q \), is denoted \( \lambda_{s,q} \). Given a computation \( \lambda \), we denote the \( i \)-th state in the computation by \( \lambda[i] \), the prefix \( q_0, q_1, q_2, \ldots, q_i \) by \( \lambda[0,i] \), and the infinite suffix \( q_i, q_{i+1}, q_{i+2}, \ldots \) by \( \lambda[i, \infty] \).

The set of outcomes of an \( A \)-strategy \( s_A \) is defined as a set of such computations starting at some state \( q \).

\[
\text{out}(q,s_A) = \{ \lambda_{s,q} | s \in \text{strat}({\mathcal{A}}) \text{ and } s \geq s_A \}
\]

We are now ready to define the satisfaction relation for ATL formulas with respect to CGS models.

**Definition 2.9 (Satisfaction (ATL)).** Given a formula \( \phi \in \mathcal{L}_{ATL} \) and a CGS \( S = \)
\langle Q, \mathcal{A}, \Pi, \pi, \delta \rangle$, the satisfaction of the formula is defined inductively for points $q \in Q$.

\[
\begin{align*}
S, q \models p & \iff p \in \pi(q) \\
S, q \models \neg \phi & \iff \text{not } S, q \models \phi \\
S, q \models \phi \land \phi' & \iff S, q \models \phi \text{ and } S, q \models \phi' \\
S, q \models \langle\langle \mathcal{A} \rangle\rangle \phi & \iff \text{there is } s_A \in \text{strat}(A) \text{ such that} \\
& \quad \text{for all } \lambda \in \text{out}(s_A, q), \text{ we have } S, \lambda[1] \models \phi \\
S, q \models \langle\langle \mathcal{A} \rangle\rangle \Box \phi & \iff \text{there is } s_A \in \text{strat}(A) \text{ such that} \\
& \quad \text{for all } \lambda \in \text{out}(s_A, q) \text{ and all } i \geq 0 \\
& \quad \text{we have } S, \lambda[i] \models \phi \\
S, q \models \langle\langle \mathcal{A} \rangle\rangle \phi U \phi' & \iff \text{there is } s_A \in \text{strat}(A) \text{ such that} \\
& \quad \text{for all } \lambda \in \text{out}(s_A, q), \text{ for some } i \geq 0 \text{ and all } 0 \leq j < i \\
& \quad \text{we have } S, \lambda[i] \models \phi' \text{ and } S, \lambda[j] \models \phi
\end{align*}
\]

### 2.4.1 Model checking and size of models

Model checking is the problem of determining whether some model and state in that model satisfies a given formula. Both the model and the input formulas are given as input. In [6] the authors gave a model checking algorithm which is linear in the size of the input (model and formula), i.e., $O(ml)$ where $m$ is the size of the model and $l$ is the length of the formula. Generally, the largest component of the model (a metric for the size of the model) is the set of transition labels, and the size of this set can be very large when the number of agents increase. A more fine grained analysis of the parameters will not necessarily yield an optimistic complexity result. It was already observed in the original presentation of the algorithm that the size of the model might be very large depending on other natural parameters, such as the number of agents, states, and so on.

In [46] the complexity is analyzed in terms of the following parameters:

- $n$ the number of states,
- $k$ the number of agents,
- $d$ maximal number of actions available to some agent in any state, and
- $l$ the length of the formula.

The size of the model can be seen to be bounded above by an expression of these parameters, $m = O(nd^k)$, which results in an exponential complexity of the original algorithm, i.e., $O(ml) = O(nd^kl)$. Hence [46] shows that model checking ATL over CGS is intractable.\footnote{\cite[The mentioned article stated that model checking ATL over CGS was $\Sigma^p_2$ complete. This was revised by Laroussinie et. al. in [54] where the complexity was shown to be $\Delta^p_2$-complete (c.f.,[47]).}}
2.5 Many-valued logic

In classical logic, a central assumption is the principle of the excluded middle. The assumption that the veracity of a proposition is dichotomous. This assumption is not purported by all logicians, however, and has been questioned since even ancient times. One source of inspiration for the development of modal logic is shared by the development of many-valued logic. Indeed, the first logic we will discuss briefly in this section, the three-valued Łukasiewicz logic, was advocated as a logic of possibility.

A many-valued logic has the classical truth values true and false, but adds at least one. We will confine ourselves to the three-valued logics and denote the values 1 (true), 1/2 (the third truth value), and 0 (false). What the third truth value describes is a matter of interpretation. We will sketch very briefly two logics: Łukasiewicz’s Ł3 and Kleene’s K3. The S indicates strong.

A three valued logic specifies a set of three truth values including truth and falsum. Additionally it needs to specify the set of designated values. In the logics we will cover, the set of designated values is the singleton \{1\}, meaning that a formula is defined to be valid if, and only if, for every interpretation of the atoms occurring in it, the formula is evaluated to a truth value of 1.

Common for three-valued logics is that they are truth functional. This means that the truth value of a compound formula is the result of applying a function interpreting the outermost/active connective to the truth values of its constituent formulas. This is different from the modal logics we defined in the previous section. Strictly speaking, a three-valued logic may be viewed as a modal logic (as Łukasiewicz did at some point), but we feel justified in distinguishing the two as has become de facto jargon in the logic community.

A logic is truth-functional when the truth value of a sentence is the result of a function application to the truth values of its constituents. If \(\odot\) is some connective applied to some appropriate number of formulas \(x_1, x_2, \ldots\) to form a compound formula \(\odot(x_1, x_2, \ldots)\) the truth value of the formula \(T(\odot(x_1, x_2, \ldots))\) is determined by a function interpreting \(\odot\) applied to the truth values of the constituting parts. That is, there is a function \(f_\odot\) such that \(T(\odot(x_1, x_2, \ldots)) = f_\odot(T(x_1), T(x_2), \ldots)\).

2.5.1 Łukasiewicz (three-valued) logic – Ł3

Initially, Łukasiewicz introduced a many-valued logic in an attempt to capture (alethic) modalities in statements, i.e., statements of the form “it is possible that \(\phi\)” (where \(\phi\) is some state of affairs). The language supports the definition of additional connectives \(L\) and \(M\) (to be defined shortly) where the intuitive interpretations of formulas constructed with these connectives, say \(Lp\) and \(Mp\), are “it is necessary that \(p\)” and “it is possible that \(p\)”, respectively [77]. If it so happens that \(p\) is possible, but not true, he argues, then the (truth) value of \(p\) is neither “true” nor “false”. The view suggests a third possible value of the proposition \(p\).

Ł3 is truth functional; the interpretation of the simple statement “\(p\) is possible”, \(Mp\), that \(Mp\) is true just in case \(p\) is possible.

This suggests the function which interprets negation over this extended domain of truth values. We give the truth tables for three connectives in Figure 2.1. The logic containing only one extra truth value (beyond the classical values), is denoted Ł3.
These expressions might be more succinctly defined by equations as shown in the following:

\[
T(\neg \phi) = 1 - T(\Phi) \tag{2.10}
\]
\[
T(\phi \rightarrow \psi) = \min\{1, 1 - (T(\phi) - T(\psi))\} \tag{2.11}
\]
\[
T(\phi \land \psi) = \max\{T(\phi), T(\psi)\} \tag{2.12}
\]

The reader can easily verify that the disjunction connective, as just explicated, may be derived from the implication connective.

\[
\neg p := p \rightarrow \bot \tag{2.13}
\]
\[
p \lor q := (p \rightarrow q) \rightarrow q \tag{2.14}
\]
\[
p \land q := \neg(p \lor \neg q) \tag{2.15}
\]

In addition to these “traditional” connectives, we will also introduce some connectives to better deal with the third value. The connective we discussed earlier, and which Łukasiewicz referred to as “possibility”, \(M\), is definable in terms of implication (and the derivable negation connective). We read \(M\) as “non-falsum” as it yields a value of 1 if, and only if, the truth value of the predicate it is applied to is not falsum (0), and 0 otherwise. The reader may easily verify that the following abbreviations express what they claim to express.

\[
M(\phi) := \neg \phi \rightarrow \phi \quad \text{non-falsum (possibility) \; (2.16)}
\]
\[
U(\phi) := \phi \rightarrow \neg \phi \quad \text{non-truth \; (2.17)}
\]
\[
L(\phi) := \neg M(\neg \phi) \quad \text{truth \; (2.18)}
\]

From now on we refer to the third value as “undetermined”. We are departing from reading this logic as a modal logic (as a logic concerning the possible and necessary), and attribute the third value rather a reading of “undetermined” or “indeterminable”.

### 2.5.2 Kleene’s (strong) three-valued logic – \(K^S_3\)

In discussing partial functions, Kleene defined a three-valued logic [50]. In introducing the third truth value, he refers to it as meaning “undefined” and states that this is not
“on a par” with the other truth values (1 and 0). It is, perhaps, not a degree of truth, but rather something different.

This indicates a justification for the main technical difference between $K_3^S$ and $Ł_3$, the interpretation of the $→$ connective. The truth tables show that they differ by assigning different truth value to statements of the form $φ → ψ$ when neither $φ$ nor $ψ$ have a classical truth value, as shown in Figure 2.2.

\[
\begin{array}{c|cc|c|cc|c}
→ & 1 & \frac{1}{2} & 0 & \rightarrow & 1 & \frac{1}{2} & 0 \\
1 & 1 & \frac{1}{2} & 0 & 1 & 1 & \frac{1}{2} & 0 \\
\frac{1}{2} & 1 & 1 & \frac{1}{2} & \frac{1}{2} & 1 & \frac{1}{2} & 1 \\
0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\
\end{array}
\]

Figure 2.2: Truth tables $→$ in $Ł_3$ and $K_3^S$.

The distinction, discussed by Kleene when introducing his logic, has several technical and conceptual consequences. As an example of the use of a strong interpretation of a conditional statement, Kleene offers an example similar to the statement “if $f(x)$ is defined, then $φ$”. If $φ$ is such that it is meaningless when $f(x)$ is not defined, what sense should we make of the claim as a whole? In the antecedent we have an undefined term, and equally so in the consequent and it would be erroneous to accept the inference.

As mentioned, the third value is referred to in [50] as “undefined”, but other possible meanings are also considered in the introductory discussion there, namely “unknown” or “value immaterial”.

We will encounter three-value logics briefly in the next section, on Abstract Argumentation Theory, and then again in Chapter 7.

2.6 Abstract Argumentation Theory

Argumentation theory studies the content of arguments used for drawing inferences. This is done by analyzing the structures of arguments, generally systematically in terms of argumentation frameworks. An example of an argumentation framework can be based on a debate between two agents discussing whether it will rain tomorrow. The example is taken from [11].

Suppose we have the following exchange of claims between agents $a$ and $b$:

\begin{align*}
a &: \text{“Tomorrow will rain because the national weather forecast says so.”} \\
b &: \text{“Tomorrow will not rain because the regional weather forecast says so.”}
\end{align*}

From this scenario of conflicting (expressed) opinions, the argumentation theorist will analyze the proposals and suggest (if possible) an acceptable conclusion. In this example, we might conclude that it will not rain tomorrow (or, at least, accept this as the more likely prediction) if we attribute the regional forecast higher accuracy than we do the national one. (We refer to [11] for further justification.)

One tool in the argumentation theorist toolbox, is the argumentation scheme. These are regularly occurring usages of arguments, accompanied with several properties such as critical questions. The theorist may seek a pattern in an argument which fits (closely,
not necessarily absolutely) to a scheme and identify either critical questions to challenge an argument, enthymemes (implicit premises or conclusion), committed fallacies, and so on. The use of argumentation schemes require us to “look inside” an arguing agent’s stated position. It is seldom a purely deductive procedure.

This stands in contrast to the classical logical approach of identifying and focusing on the deductively valid where we premises and conclusion is clearly stated and a judgment about the deductive validity. This approach is sometimes called monological, where the argumentation approach is called dialogical, or dialectical [81].

Another tool is the argumentation framework. Let us visualize the previous dispute by drawing a graph illustrating the situation. Let $p_a$ be $a$’s claim that it will rain tomorrow, and $p_b$ be $b$’s statement. The conclusion of the two claims are in direct contradiction. Notice that identifying the conclusion is a part of the “craftsmanship” performed by the argumentation theorist when reading or analyzing the argument. We have two arguments in mutual opposition. We sketch exactly this aspect in Figure 2.3.

“Tomorrow will rain because the national weather forecast says so.”

(Inferred) opposition

“Inferred) opposition

“Tomorrow will not rain because the regional weather forecast says so.”

Figure 2.3: A sketch of an argument.

By the content of these arguments, we have concluded that they are in mutual opposition. This in itself informs us, without any further inspection of the content of the arguments, that a rational (consistent) agent can not possibly accept both of them simultaneously. However, it does seem (without further appeal to their content) that she might accept either. (Recall that when we declared $p_b$ the better argument, we gave $b$’s source of information relatively superior credibility.)

The abstract structure of the argument, hence, may tell us something about the discourse without appeal to more than (derived) relations of opposition between arguments. This is the underlying premise of abstract argumentation theory. We redraw the scenario from above as a purely abstract model in Figure 2.4.

Figure 2.4: An abstract view of an argument. Edges in the graph indicate opposition.

An (abstract) argumentation framework is a (directed) graph over a set of arguments.

An argument is an atomic construct in our framework in a similar sense to how proposition symbols are in propositional logic. To be meaningful, they refer to actual

---

6From this point on, throughout the book, we will only consider abstract argumentation, and abstract argumentation frameworks. For simplicity, and without confusion, we will generally refer to these without explicitly prepending the words with “abstract”.

---
arguments, but we are here interested in them for their relation to the rest of the arguments. The formal definition of these structures originate in [30] where Dung defines an argumentation framework to be a set of arguments together with an attack relation.

**Definition 2.19 (Argumentation Framework).** An argumentation framework \((AF)\) is a pair \(AF = \langle A, R \rangle\) where

- \(A\) is a non-empty set of arguments, and
- \(R \subseteq A \times A\) is an attack relation.

If \((x, y) \in E\) we say that the argument \(x\) attacks argument \(y\).

An argumentation framework (AF) essentially describes an argumentation setting as a directed graph. A crucial question when faced with such an argumentation setting is what is the sensible position to take with respect to the arguments which have been “presented”, or are otherwise being assessed.

There may be several viable positions to take, or occasionally no sensible position. What is sensible is not a straightforward conclusion to reach either. The/a sensible position, either way, will be some set of arguments among the arguments in question. Such a subset of the arguments is called an extension. Several semantics have been proposed. A semantic always have the same signature, as defined in the following definition.

**Definition 2.20 (Extension-based Semantic).** Let \(\\mathcal{A}\) be some collection of argumentation frameworks over arguments \(A\). An (extension-based) semantic is a map \(\varepsilon : \mathcal{A} \to 2^2^A\). That is, for an argumentation framework \(\langle A, R \rangle \in \mathcal{A}\), \(\varepsilon \langle A, R \rangle \subseteq 2^A\) yields a set of extensions.

There are several semantics for argumentation theory. Each with properties which speak for and against it, depending on how we perceive argumentation to function, or rather how we think argumentation ought to function.

Some common, and simple, candidates are listed in the following definition.

**Definition 2.21 (Semantics).** Some semantics for argumentation theory are:

- **Admissible** An extension \(E \subseteq A\) is admissible if, and only if, \(E\) is conflict free (i.e., no \(x, y \in E\) is \((x, y) \in R\)), and defends all its elements (i.e., if there is an \(x \in E\) such that \((y, x) \in R\), then there is a \(z \in E\), such that \((z, y) \in R\)).

- **Preferred** An extension \(E \subseteq A\) is a preferred extension if, and only if, \(E\) is a maximal elements with respect to set inclusion among the admissible sets.

- **Stable** An extension \(E \subseteq A\) is stable if, and only if, \(E\) is conflict free and every element \(y \in A \setminus E\) outside \(E\) are attacked by an element in \(E\) (i.e., there is an \(x \in E\) such that \((x, y) \in R\)).

The notion of attacking, and being attacked is central in argumentation theory. We introduce some notation for denoting the set of attackers or attacked arguments of a given argument. Let \(a \in A\) be some argument. The set of arguments attacking \(x\) is denoted \(R^-(x)\) and is simply the set consisting of arguments \(y\) for which \((y, x) \in R\).
Analogously, the set of arguments \( x \) attacks is denoted \( R^+(x) \). We can readily generalize this notation to sets. For a set of arguments \( X \subseteq A \), we denote by \( R^-(X) \) the set \( \bigcup_{x \in X} R^-(x) \), and analogously define \( R^-(X) := \bigcup_{x \in X} R^+(x) \). In each case, say we are concerned with the admissible semantics currently, the semantics specifies a set of sets of arguments. The set of admissible extensions can thus be defined as

\[
\varepsilon(A,R) = \{ X \subseteq A \mid R^-(X) \subseteq R^+(X) \text{ and } R^+(X) \subseteq A \setminus X \}
\]

For the purpose of this text we will not need more knowledge about particular extensions and their properties. We will be, so to speak, agnostic about argument semantics.

Given an AF, the set of its extensions may be empty, unique (singleton), or contain several extensions. An argument which is contained in at least one of these extensions is said to be _credulously accepted_, and an argument is said to be _skeptically accepted_ if it is contained in all of them.

**Definition 2.22** (Credulous and Sceptical Acceptance). Given an argumentation framework \((A, E)\) with an argument \( x \in A \) and a semantic \( \varepsilon \), we say that

- \( x \) is credulously accepted if, and only if, \( x \in \bigcup \varepsilon(A,E) \), and
- \( x \) is skeptically accepted if, and only if, \( x \in \bigcap \varepsilon(A,E) \).

What we may infer from an AF’s extensions, may also be inferred from so-called labelings.

**Definition 2.23** (Argument Labelling). If \((A, E)\) is an AF, an argument labeling is a map \( \text{lab} : A \rightarrow \Theta \), where \( \Theta \) is the set of labels, typically either \( \{ \text{in}, \text{out} \} \), or \( \{ \text{in}, \text{undec}, \text{out} \} \).

There is a close relation between extensions and labelings. One translation of extensions to labeling is to label the arguments in the extension with \( \text{in} \), the arguments attacked by the arguments in the extension with \( \text{out} \), and the remaining arguments with \( \text{undec} \). And conversely, from a labeling, we may consider a corresponding extension which is simply the arguments labeled \( \text{in} \). For a more elaborate exposition of this relationship we refer to [8, 21].

The reader may already suspect the continuation of this line of reasoning. Three labels, \( \text{in} \), \( \text{undec} \), and \( \text{out} \) seem very close to some interpretations of many-valued logic. Indeed, a view of a labeling as an interpretation of a three-valued logical model has been suggested in the literature, see [8, 33].

For the purpose of this text it is sufficient to see the existence of such interpretations, and their connection to more classical semantics for argumentation theory. Corresponding to the previous definition of argument labeling, we give give a definition of an _argument valuation_ in a three-valued logic.

Ultimately, the choice of semantic and, if applicable, the logic used to interpret the state of an argument, is not settled and may even be a question of pragmatics. The particulars will not be essential in our discussions.

**Definition 2.24** (Argument Valuation). If \((A, E)\) is an AF, an argument valuation is a 3-partitioning \( \pi \in 3^A \).
This valuation may be extended by means of logical connectives in the usual sense. We define this extension in terms of Łukasiewicz three-value logic because this is the basis for our framework presented in Chapter 7. However, other extensions may be defined equivalently, say for Kleene logic. We define a language over the arguments
\[ \alpha ::= p \mid \neg \alpha \mid \alpha \rightarrow \alpha \]
where \( p \in A \) is an argument.

**Definition 2.25.** If \((A, E)\) is an AF and \(\pi\) is an argument valuation, we define the extension of \(\pi\), \(\overline{\pi}\) inductively

\[
\begin{align*}
\overline{\pi}(p) &:= \pi(p) \\
\overline{\pi}(\neg \alpha) &:= 1 - \overline{\pi}(\alpha) \\
\overline{\pi}(\alpha_1 \rightarrow \alpha_2) &:= \min\{1, 1 - (\overline{\pi}(\alpha_1) - \overline{\pi}(\alpha_2))\}
\end{align*}
\]

For more details we refer to [8].

To summarize this section, let us consider an example of a simple argumentation framework together with an extension, a corresponding argument labeling, and an argument valuation.

**Example 2.26.** Let \((A, E)\) be the argumentation framework consisting of three arguments \(A = \{a, b, c\}\) such that \(a\) and \(b\) are in mutual opposition, so are \(b\) and \(c\). Additionally, \(c\) even opposes itself. The argumentation framework is illustrated in Figure 2.5.

![Figure 2.5: Example argumentation framework, \((A, E)\).](image)

Assuming we are employing admissible semantics, the only possible extension (except the empty set which is always admissible) is the set \(\{a\}\) meaning that accepting \(a\), and only \(a\), is an acceptable stance with respect to the argument. Formally, we denote this \(\{a\} \in \varepsilon(A, E)\). Equivalently, we may say that labeling \(a\) with \textit{in}, \(b\) with \textit{out}, and \(c\) with \textit{undec}, we have characterized/labeled the arguments in an admissible way. Or, appealing to an argument valuation approach we may let \(\pi(a) = 1\), \(\pi(b) = 0\), and \(\pi(c) = \frac{1}{2}\) be a valuation corresponding to the just mentioned labeling.

For more information about argumentation theory (abstract and concrete), we refer to [70].
Chapter 3

Representation

3.1 Background

Logics for reasoning about the strategic capabilities of agents and coalitions of agents have been an influential thread of research in the field of MAS for several decades. The starting point for our analysis is the branching-time temporal logic ATL [6]. In this chapter we introduce the notion of roles in the semantics by defining concurrent game structures with roles (RCGS). We show how this can lead to an improvement in the complexity of the model checking problem. This chapter is largely based on [64].

3.2 Homogeneity

Recall from Chapter 1 that a group of agents is homogeneous if the agents have the same strategic ability, i.e., the same ability to bring things about. The outcomes must be invariant under permutations of actions. In games there is as we mentioned a distinction between symmetric and anonymous games. The difference between the two was that in the former, we permuted the outcome, or utility, along with the actions, while in the latter we permuted only the actions.

When we move from games with ascribed utilities as used in game theory to more qualitative models, as the CGS models of ATL, we lose the notion of utility. However, as utilities are based on, or part of, the state of affairs, we can still recognize when the utilities of two states are the same. That is, in every case, whenever the states of affairs are the same, the utilities must also be the same. Conversely, in some scenarios any change in the state of affair might constitute a difference of utility.

What exactly the relationship between state of affairs and utilities may be is difficult or impossible to give a general answer to.\(^1\) In this chapter we will not appeal to utility judgments, and without the mechanism of utility it is difficult or impossible to transfer verbatim the notion of symmetric games into qualitative strategic settings. The condition which defined anonymous games however, carry over since we do not need to “permute the utilities” (whatever they may be). The outcome must be invariant under permutation of action, leaving the description of the state of affairs unchanged.

That is, in any state, for any two agents – say agents 1 and 2, for any pair of actions they might have chosen – say 1 chose \(\alpha\) and 2 chose \(\beta\), the state of affairs we end up

\(^1\)In Chapter 6 we will investigate one possible approach this problem.
in, is the same as if they had switched actions among themselves – i.e., if agent 1 chose $\beta$ and 2 chose $\alpha$.

Two conditions become immediately clear:

1. In every state, all agents must have the same actions available, and
2. the actions must be extensionally equal (i.e., must have the same consequences).

Let us consider a state and its outgoing transitions in a CGS which satisfies these two conditions.

**Example 3.1 (Strategical homogeneity in CGS).** Consider two processes (our agents), running in a computer system with a shared file system. There are two files of interest which both agents can write to; file A and file B. If they both attempt to write to the same file at the same time, we get a collision. Consider that, in a particular state ($q_0$), both agents only have three actions available; write to file A, write to file B, and wait.

In this particular state, each agent might do either of the three actions. If both agents wait we will loop and return to the same state. The current state $q_0$ and its immediate successors might look like the snippet in Figure 3.1.

![Figure 3.1: Example of an “anonymous” CGS state. A, B denotes change in file A or B, resp. C denotes conflict. Labels a, b and w denote the possible actions. As every label denotes an arrow, there are nine (non-dotted) arrows in this figure.](image)

We can easily verify that every possible combination of choice of actions is represented by some arrow, and that the outcome is not changed when we permute the actions of the various agents. Interpreting this model as a game (regardless of how we translate “state of affair” into utility) we end up with an anonymous game. That is, since any two action profiles which are permutations of each other yields the same state, the agents receive the same utility regardless of how the state of affairs are translated into utility.

The property of being invariant under permutation of actions means that we do not need to keep track of which particular agent performed which action among the performed actions, but rather only the actions that were performed.
3.3 Homogeneous Roles

Schematically, the possible action combinations which could occur in the example can be seen as the Cartesian product of the available actions to each of the agents, but since we require the various permutations to have the same outcome we end up with a much simpler set of possibilities. In Figure 3.2 we can see an illustration of this difference. Whereas on the left-hand side, the set of successor states correspond to the product of all actions available to all agents, on the right-hand side we have only the number of agents which performed each of the available actions.

![Figure 3.2: Example of potentially heterogeneous versus guaranteed homogeneous set of successor states.](image)

Even though the illustration on the right in Figure 3.2 appears to have higher dimension (as there are three axes, rather than two on the left-hand side), this is only for ease of representation. The geometrical shape we get is a simplex of dimension equal to the number of actions available to the agents.

Many CGS models may very well satisfy these properties. Every homogeneous model is of course just a special case among the potentially heterogeneous models. The conditions above stipulate that two states reached from a single state by, respectively, any complete action profile and a permutation of this profile must be bisimilar, and not necessarily identical. The fact that our language can not distinguish between bisimilar states in part justifies the collapse of these two (hypothetically) different but bisimilar states into one. The inability to distinguish between these states is captured by the models resulting in the model also having fewer edges (labels) than would a corresponding CGS.

Furthermore, the significant decrease in the number of edges in the model that results from this collapse will have significant consequence in computational complexity which we will spell out in detail in Section 3.7. To obtain such a result we do not need to require that every single agent in our model has similar strategic abilities as every other agent. It suffices to introduce a mechanism for distinguishing between the strategic abilities of different agents which enact different roles in the model.

3.3 Homogeneous Roles

The information we have removed from the models now, which allows agents to actually differ in terms of strategic capabilities, is of course important for many models in
Representation

42

ATL. We can let the set of agents retain their differences by putting them into different (homogeneous) roles.\(^2\)

This is then a part of the modified semantic structure. We assign to each agent (in each state) a specific role, and ascribe to each role a certain set of actions which are available to the agents in that role. Thus, we add components specifying the role of each agent in each state and which actions each agent (through its role) have available to her. The set of agents belonging to any role is homogeneous. We modify the definition of CGS to include the assignment of agents to roles.

3.4 Roles

The term role has entered the scientific vocabulary through sociology, and has entered MAS gradually and in several guises. An early, and seminal, use by Linton of the term can be found in [55]. Linton describes roles as a property interdependent on that of status.

“A status, in the abstract, is a position in a particular pattern. It is thus quite correct to speak of each individual as having many statuses, since each individual participates in the expression of a number of patterns. However, unless the term is qualified in some way, the status of any individual means the sum total of all statuses which he occupies. It represents his position with relation to the total society. [...] A status, as distinct from the individual who may occupy it, is simply a collection of rights and duties”.

In terms of MAS, the status of a participating agent, the interacting computing element, is similarly compound. Per status one may recognize it by verifying some condition; is the agent ready to receive/transmit, is the agent able to write to a persistent storage unit, has the agent’s identity been verified? Being a member of a homogeneous role can be interpreted as possessing a certain status. As we will see in this chapter, the system (and the agents therein) are unable to distinguish between any two agents in the same role except for these agents’ possible futures.\(^3\)

In a relatively sparsely populated MAS with many possible statuses modeled in our framework, one might expect that the system needs to accommodate a great number of roles, and each role would contain few agents. The sum status of agents would perhaps seldom overlap completely. However, as the number of agents grow, regardless of the number of statuses and possible values of these statuses, invariably, the roles will be more populous.

In terms of such a characterization, we may see membership of a role as being in line with that of occupying a status. Concerning roles, Linton says:

\(^2\)The choice of the term “role” might be contentious. Daskalakis & Papadimitriou apply a similar notion for anonymous games in [25] but use the term “types”. We have opted for the term “role” in part because we will let the role of an agent be state-dependent, and we feel that it is more natural that the role, rather than the type, of an agent changes over time, and it is closely related to the use of the word in social science as we argue in the next section.

\(^3\)In the next chapter we will investigate the consequence of agents belonging to the same role being entirely indistinguishable.
“A rôle represents the dynamic aspect of a status. The individual is socially assigned to a status and occupies it with relation to other statuses. When he puts the rights and duties which constitute the status into effect, he is performing a rôle. Rôle and status are quite inseparable, and the distinction between them is of only academic interest. There are no rôles without statues or statues without rôles”.

The role of an agent, analogously to the status of an agent, when used without qualification, Linton defines to be the sum compound of rôles.

As the use and scrutiny of the terms evolved in the sociological literature, there has been some refinement in the terminology. In more recent times, the foundational concepts of (particularly of structural and functionalist) role theory have been settling near some central notions. They vary somewhat from author to author. Some of the central terms are sketched below. (See [15] for an overview.)

Task/Expectation The task is a description of an accomplished goal, and a system (or coalition or agent) having a task can equivalently be said to have the goal of bringing about the result of the task being completed.

In [15], the near synonymous term used is “expectation”. As Biddle points out there, various authors use different words and have different understanding of the mechanisms in play. Some see the different expectations as an expression of a prescriptive phenomena, or a norm. Others as an expression of the agents’ belief about the system, and others still as the result of individual preferences or attitudes.

The differences are of critical importance to the empiricist, but in the logical framework, we will allow a plurality of interpretations of the difference in behavior. When we approach this from a logical perspective and discuss first what are the possible outcomes, we unify the notions.

Specifying the expected outcome among the possible is not part of our investigation. On the level of abstraction within which we are operating, the notion of an expectation as just elaborated is very closely related to that of a task, or goal. To the extent we discuss particular tasks or expectations, we will not discuss their origins and assume they are exogenous. However we return to issues related to such questions in Chapters 6 and 7.

Person/(Social) Position/Role The notion of a person as employed by Oeser & Harary in [60], does not refer to the acting agent. They distinguish explicitly between people (the concrete agents in the system), versus persons which enact roles. Persons, they state, are taken as a set of attributes. The person, then, is an abstraction of an agent which enact a role. The person can be seen as the arch-typical role enacting agent. The two terms, “person” and “role”, can be seen somewhat dually.

A person, in [60] can be seen as the occupant of a position. A person, being an abstraction of an agent, is hence further abstracted to give rise to the notion of a position. The precise nature of a (social) position depends on the application and the level of abstraction we are interested in. Informally, we equate a role with
a position and a number specifying the number of agents enacting that role, or occupying such a position. That is, a position, similar to Linton’s “status” defines the occupying agents rights and duties (which we interpret as the set of possible actions). We can recognize a position, Oeser & Harary say by them always being referred to by generic names (“teacher”, “carpenters”, and so on), and the people assigned to a position being exchangeable.

We will not address the former point directly, a position will be a member of some generic role \( r \). If \( r \) is the role of “teacher”, it would be correct to refer to a position in this role as “a teacher”. The assumption that the person which is assigned to a position being exchangeable is quite important. For a fixed social structure (collection of roles), we will see that the consequence of this assumption can be significant. In the next chapter we investigate the consequences of dropping this assumption, by requiring that the distribution of persons over positions remains fixed. There we will also discuss these abstract notions of the agent further.

Also the term “role”, Biddle explains in [15], is used in various ways by different authors including role as a collection of norms that is organized about a function, or a pattern of behaviors and attitudes. Or the behavior which results from normative expectations which are associated with a position. The former, in particular, unites tasks and positions. While the treatment we give in this text can be seen to focus on the “position” part of such an interpretation, the term “role” has been used in MAS to focus on the task element (as we will discuss shortly). In either case we abstract away from the actual expected behavior if this is understood to be a particular reaction to a situation, but rather model the set of possible reactions.

In each case, the behavior of the agents is central. When we discuss the behavior of agents in our formal model, we discuss the behavior of agents very broadly: the consequences of every possible strategy for any set of agents. Thus, the formal models we describe permits us to reason about the consequences of permitting agents to act according to their position, where their position define, precisely, their available actions and the precise consequences of these actions.

The term “role” has been used in a formal system of MAS particularly by Ryan & Schobbens in [72] exploring the notion of a role based on the concept of a refinement relation [7] by Alur, Henzinger, Kupferman & Vardi. In their approach, a role is tightly connected to a task or expectation. The refinement relation details how a single compound action agent can be verified to be equivalent to a set of “sub-agent”; agents which can be seen to constitute the compound agent. This compound agent is modeled to be responsible for a certain task, e.g., the sender is responsible for sending data.

The sender which is a role in the example provided in [72], is then shown to correspond, in that particular model, to three concrete agents trans (sends packets of data), ack-rcvr (tracks which messages that have been acknowledged) and re-trans (re-transmits packets which have not been acknowledged) performing the actual atomic actions. This refinement relation permits us to limit the number of agents in the model by abstracting away a number of agents and replacing them by one role. The members (or concrete constituting agents) of a role in the sense of [72] will not typically be
homogeneous as in our formalism i.e., they are occupying different (sum) roles as per Linton. They will generally be different, performing different sub-tasks.

This is certainly a legitimate use of the term “role” which is in line with the above description, but it does not offer a complete picture. When we now use the term to be quite different, we use it to denote the social position of the agents occupying a certain role. The concrete agent \textit{trans} which is only a part of the role \textit{sender} (in the sense of [72]) has certain “rights and duties” (c.f., Linton [55]), which are clearly distinct from the other concrete agents that in sum constitute the role \textit{sender}.

In our terminology, the fact that concrete agents have different rights and duties would place them in different roles. This, however, can be seen as being in line with how the term is used in sociology. While the term “role” is also often connected to a task, as is the basis of Ryan & Schobbens’ example, it is not necessarily the only correct use of the notion. The term as applied in sociology is quite broad and encompasses both our “homogeneous roles” and, in some schools, Ryan & Schobbens’ task-based role.

### 3.5 Concurrent Game Structures with Roles

Before we state the formal definition of the semantic construct Concurrent Game Structures with Roles (RCGS), we need to introduce some more terminology. Anticipating Definition 3.4 of RCGS which we will state shortly, we clarify some elements of the definition which might seem somewhat convoluted. For every state \( q \), a role \( r \) can be thought of as a set of agents. As the role assignment may change from state to state, we will denote this \( A_{q,r} \subseteq A \) where \( A \) is the set of all agents. For every state \( q \), and every role \( r \), we will have a number of actions available in that state to the agents belonging to that role, \( A_{q,r} \). This specifies only the number of actions, and functions as an index which will let us identify the actions.

We will need some book keeping of the various set and subsets of agents in our models. Therefore we will define a structure pre-RCGS with the very basic components we need before we introduce the strategic elements. A pre-RCGS consists of the carrying sets; agents, roles, propositions and states; for each state an assigned role for each agent; for each state an assigned number of actions available to the agents in any given role; and a proposition labeling map.

**Definition 3.2** (pre-RCGS). A pre-RCGS structure is a tuple \( \langle A, R, \rho, Q, \Pi, \pi, A \rangle \), where:

- \( A \) is a finite, non-empty set of players.
- \( Q \) is the non-empty set of states.
- \( R \) is a finite, non-empty set of roles.
- \( \rho : Q \times A \rightarrow R \) assigning a role to an agent depending on the state.
- \( \Pi \) is a non-empty set of propositional symbols.
- \( \pi : Q \rightarrow 2^\Pi \) maps states to the propositions true at that state.
• \( A : \mathcal{Q} \times R \rightarrow \mathbb{N}^+ \) is the number of available actions for the agents in a given role at a given state.

A subset \( \mathcal{A} \subseteq \mathcal{A} \) of agents is called a coalition. Given a pre-RCGS, we will need to be able to discuss certain specific coalitions. We introduce the following terminology.

\[
\mathcal{A} := \mathcal{A} \subseteq \mathcal{A} \quad \text{coALITION}
\]

\[
\mathcal{A}_{q,r} := \{ a \in \mathcal{A} \mid \rho(q,a) = r \} \quad \text{agents from } \mathcal{A} \text{ in role } r \text{ at } q
\]

Notice particularly, that \( \mathcal{A}_{q,r} \) is the set of all agents in role \( r \) at state \( q \).

Given a coalition \( \mathcal{A} \) and a set of alternatives the agents may choose from \( \mathcal{A} \), the representation of the anonymous choices will be called (partial) profiles, or occasionally votes. The consequences of these choices will be a consequence of the strategic circumstances and defined in detail in Definition 3.4.

**Definition 3.3 (A-profile/complete profile for role).** Given a pre-RCGS \( \langle \mathcal{A}, R, \rho, \mathcal{Q}, \Pi, \pi, \mathcal{A} \rangle \), a coalition \( \mathcal{A} \subseteq \mathcal{A} \), a role \( r \) and a state \( q \), the A-profiles for \( r \) at \( q \) is defined as

\[
P_r(q,\mathcal{A}) = \left\{ v \in \{0, \ldots, |\mathcal{A}_{q,r}|\}^{|\mathcal{A}_{q,r}|} \mid \sum_{1 \leq i \leq |\mathcal{A}_{q,r}|} v(i) = |\mathcal{A}_{q,r}| \right\}
\]

The set \( P_r(q,\mathcal{A}) \) is the set of \( \mathcal{A}_{q,r} \)-tuples with only non-negative elements, the sum of which equals the size of the coalition restricted to members of role \( r \) at \( q \).

It might be easier to think of the elements of this set as functions

\[
v : [\mathcal{A}_{q,r}] \rightarrow \{0, \ldots, |\mathcal{A}_{q,r}|\}
\]

Particularly, we pick out the \( i \)-th element of the tuple by applying it to \( i \).

In the special case that \( \mathcal{A} = \mathcal{A} \), we refer to this set as the set of complete \( r \)-profiles and denote it \( P_r(q) \)

\[
P_r(q) = P_r(q,\mathcal{A}).
\]

If an A-profile is in \( P_r(q) \), we say that it is complete.

In Figure 3.1 we saw some examples of labels in a CGS structure. Recall that in that model, the two agents had three actions to choose from in state \( q_0 \): \( a, b \) and \( c \). The profile structure is based on an indexing of the available actions\(^4\) so let us order the actions from that example as \( a \mapsto 1, b \mapsto 2 \) and \( w \mapsto 3 \). The two edges from \( q_0 \) to \( q_A \), for example, \( (a,w) \) and \( (w,a) \) would correspond to the same element in the set of complete profiles: the complete profile which represents one agent choosing the first action \( a \) and one agent choosing the third action \( w \). Say the first agent chooses \( a \) and the second chooses \( w \). Then the transition label for a CGS model is as illustrated by the upper left arrow in Figure 3.3. The upper right arrow illustrates the corresponding label in a RCGS model.

If the model consists of more than one role, we would amalgamate complete profiles for each role into a complete profile (Definition 3.4 below). To illustrate this consider

\(^4\)The actions do not need to be indexed by a number, of course. We could as easily have indexed the elements by the actions they correspond to, but this would make the notation and the illustrations more involved and, we feel, unclear.
two cases. One which is identical to the situation described in Figure 3.1 with two agents choosing among three actions, and a more elaborate example. Consider several instances of three different programs running in a common environment. Let’s say there are three instances of each program, and each of these programs have two options (raise an exception or continue). Suppose all programs choose to perform action 2 (continuing execution) except the second instance of the third program. We would have a transition label corresponding to the lower left arrow in Figure 3.3. If the system needs to react identically to exceptions of any instance of the same program, we may suitably model the scenario with RCGSs. The corresponding arrow, or label, is depicted by the lower right arrow in the same figure.

We could have chosen another way to represent these complete profiles. For example multi-sets of actions, or the lists from CGS but with restrictions. The representation we use here might take some time to familiarize oneself with. The key idea is to order the roles, then order the actions (in each role), and specify for each such position how many agents perform the indexed action.

We choose to use two types of parentheses when denoting complete profiles. The outermost parentheses \( \langle \ldots \rangle \) is a \( r \)-tuple, where \( r \) is the number of roles. Each element corresponds to a complete profile for a role. The innermost parentheses \( \ldots \) is an \( n \) tuple where \( n \) is the number of actions available to the agents in that role. We omit the innermost parenthesis when there is only one role, or the agents in that role only has a single action to choose from.

Putting this together we get a definition of concurrent game structures with roles (RCGS) as follows.

**Definition 3.4 (RCGS).** An RCGS is a tuple \( H = \langle A, R, \rho, Q, \Pi, \pi, A, \delta \rangle \) where:

- \( \langle A, R, \rho, Q, \Pi, \pi, A \rangle \) is a pre-RCGS (Definition 3.2).

**Auxiliary** For a state \( q \) and role \( r \), the set of complete profiles for a role \( P_r(q) \) is as stated in Definition 3.3 and the set of complete profiles \( P(q) = \prod_{r \in R} P_r(q) \).

- For each \( q \in Q \) there is a function \( \delta_q : P(q) \to Q \).

\[ \delta : Q \times \bigcup_{q \in Q} P(q) \to Q \] such that for all \( q \in Q, v \in P(q) \), \( \delta_q(v) \in Q \).

To see how RCGS models differ from CGS models, let us consider an example based on the well-known (in the ATL literature) train-controller-example originally from [6]. In our scenario there are \( n_t \) trains rather than a single train.
Example 3.5 (Homogeneous trains). We will have two roles in the example: train and controller. Throughout this example, the role assignment remains fixed. The trains are assigned to the train role in every state, and likewise the controller is assigned to the controller role in every state. We give a graphical presentation of the model in Figure 3.4. In the figure, in order to not draw too many edges, we add several labels to the arrows. Technically, each label corresponds to an edge. There are four states:

- In $q_0$ our initial state satisfies only the propositional symbol $out$. There are two possible successor states; if a request (to enter the tunnel) is made, we transition to state $q_1$, otherwise we loop back to $q_0$. Making a request is one of two possible actions the trains (any train) can make. The controller has only one possible action (waiting for a request).

- In $q_1$ the trains have only one action (i.e., they have no choice). The controller has three choices: delay (looping back to $q_1$), deny (returning to $q_0$), or accept (transitioning to $q_2$). Two propositional symbols hold in this state: $out$ and $req$, meaning that the trains are out of the tunnel and a request has been made.

- If the controller accepted the request (leading us to state $q_2$), the controller again has no choice (only one possible action). The trains now have two choices. They can either (try to) enter the tunnel, or they can relinquish the permission (by no trains entering the tunnel). Two propositional symbols are satisfied in this model: $out$ and $grant$ signifying that all trains are out of the tunnel and the trains have permission to enter the tunnel.

- If some train, we will discuss the identity of this train shortly, enters the tunnel, we transition into state $q_3$ which satisfies a single propositional symbol in (i.e., there is a train in the tunnel). There, trains have two choices and the controller has a single possible action (and hence no choice). The two actions the trains can choose among are leave or do nothing. If no train chooses to leave, we stay in $q_3$ and otherwise we transition back into $q_0$.

Notice that in the single-train case ($n_t = 1$), the train can not wait before entering the tunnel after being granted permission (and retain the permission). This could of course easily be avoided by adding another action. More importantly, in the case of several trains, the controller can not distinguish between the different trains, so permission must be granted to all or none. This is a consequence of the strict homogeneity in the model: not only are the agents homogeneous in terms of the actions available to them, we can not reasonably distinguish between them as long as they remain in the same role. Notice that this feature allow us to add any number of trains to the scenario without incurring more than a polynomial increase in the size of the model (total number of profiles; we discuss this in detail in Section 3.7). This would not be possible if we did not have roles. If the model above was to be rendered as a CGS, the number of possible ways in which trains could act would be exponential in all states where trains have to make a choice of what action to perform. This would be the case even if, as in the scenario above, almost all possible combinations of choices should be treated in the same way by the system, i.e., even if there are only a few possible successor states.

Now we have all the strategic components and a complete definition for RCGS in place. We need to introduce some more terminology in order to reason about the
3.5 Concurrent Game Structures with Roles

Figure 3.4: Train controller model for \( n_t \) trains. \( n_t \) is the (constant) number of trains. (Example 3.5.)

strategic capabilities of the agents. The next subsection introduces the necessary terminology. The terminology will be easily recognizable as straightforward modifications of the corresponding terminology about CGS.

3.5.1 Which train is in the tunnel? – Actions are identified extensionally

When the trains were granted permission to enter the tunnel in the previous example, a slight oddity occurred. We had, with the current model, no way of telling which train was in the tunnel. If a particular train was said to enter the tunnel and that train alone had the power to ensure that the tunnel became empty in the next step, that train could not possibly be a member of a strategically homogeneous role (or group of agents). This would perhaps be an intuitive way of modeling the scenario, but this would distinguish that agent’s power from all the other trains.

Generally, actions shared among the agents in a role must be extensionally similar. If, say, Train 1 has the ability to “leave the tunnel”, then any other train in the same role must have the ability to “make Train 1 leave the tunnel”. In the next example, we will give the trains more autonomy, and provide the particular trains exactly this ability uniquely.

The coincidence of power is not the only collapse that strategic homogeneity incurs on our models. There can not be propositional symbols which grant power to some, but not other, agents in the same role. In other words, there is no way of letting a single agent perform some action which will make her distinguished from the remaining agents, unless any other agent in the same role can also make that agent distinguished in the same sense. If one agent is permitted to enter the tunnel, all trains (in the same
role) are permitted to make that train enter the tunnel.

Unlike what we did in the previous example, we can distinguish the train which enters the tunnel if we use the role assignment component more actively. Any agent in the homogeneous group might be moved out from the homogeneous group and be assigned a distinguished role. This role may in some cases be a role extending the previous role, or in other cases completely different, at the discretion of the modeler. In the next example, we will show one way of doing exactly this. The example will further highlight the similarity between RCGS models and anonymous voting.

In the next two examples we will develop a model where a specific, named, train is permitted to enter the tunnel. Which train, in the homogeneous role of trains, should be permitted could be chosen in a number of ways. In this example we will use majority voting with lexicographic tie breaking.

**Example 3.6** (Cost of autonomy). *In the previous example all trains were equal before the controller; the controller could not distinguish between trains. We could grant the agents much more autonomy and individual identity by simply adding one more role, and in this example we sketch the result of doing so. We will now have three roles: train, privileged train and controller. We still have a single controller and \( n_t \) trains, a total of \( n_t + 1 \) agents. In this example, let us say we have three trains: \( a, b \) and \( c \).

We start in a state \( q_0 \) and enter one of three successor states \( q_{1,a}, q_{1,b} \) or \( q_{1,c} \). For the sake of this example, we assume that the trains need to elect a train to designate among themselves, i.e., without the influence of the controller. All trains should have the opportunity to enter the designated role, and hence we assign in \( q_0 \) three available actions to the agents in the train-role. A complete profile for the trains is a triple (three actions) of numbers, the elements of which sum up to three (the number of agents).

In Figure 3.5 we can see that \( q_0 \) has three possible successor states: \( q_{1,a}, q_{1,b} \) and \( q_{1,c} \). In the illustration these states are decorated with two sets below the state names. The leftmost set \( \{a, b, c\} \) in state \( q_0 \) shows the members of the role train, and the rightmost set \( \emptyset \) in \( q_0 \) shows the members of the role designated train. In each of the remaining states, one of these trains have been moved to a new role. In this example, we assume for simplicity that the controller (not shown as a role in the figure) has no influence on which train is elected. Each of the three trains have three actions available to them, and is allowed to vote for any train (including itself). If there are more votes for train \( a \) (corresponding to the first action available to the trains) than votes for any other train (or there is a tie we break to a’s favor), we transition to state \( q_{1,a} \).

Compared to Example 3.5, the model has grown, so the trains gain autonomy at a cost. Still, this cost is much less than the cost of modeling this scenario in a CGS. There, if each train is to have the option to “vote” for any train in \( q_0 \), each train must have \( n_t \) actions available. We would get \( n_t^3 \) edges leading out from \( q_0 \)! In the RCGS model we get a substantially smaller degree, Table 3.6 summarizes the difference (formulas for counting the degree are explained and discussed further in Section 3.7).

RCGS models are very suitable for modeling scenarios where a number of agents belong to a homogeneous group. This does not mean that these are the only interactions we are able to model, however. The influence agents outside the role exert on the outcome need not conform to any constraint with respect to the choice of the agents in the role, other than the fact that every such agent in the other role can exert exactly the same influence.
Any anonymous social choice function can be applied here, but we do not need to satisfy any other properties often of interest in voting theory such as majority voting. The designation of a train need not be an anonymous process at all, however. This is only the case if the controller has no possible choices, only the contribution from the members of a homogeneous group needs to behave as an anonymous voting among the agents in that group.

After having designated a single train, the model proceeds more or less as the original example from Alur et al. [6].

Example 3.7 (Increasing autonomy). Continuing the previous example, we have now decided how to place one train from the homogeneous train role into a designated role with the intention of granting this agent more autonomy. In this example, we show how to model the strategic differentiation we need to model in order to allow the designated train to utilize this separation. Towards constructing such a model, we expand the model for each of the successor states described in Example 3.6. In this example we elaborate on the successors of state $q_{1,a}$ in which Train a has just been designated as the privileged train.

In Figure 3.7 we show the initial state $q_0$ and the “loop” which describes Train a being designated and the possible outcomes of this situation. Recall that we assume majority voting, we resolve ties lexicographical. This is reflected in the complete profiles which label the arrows from $q_0$ to $q_{1,a}$.

Upon reaching $q_{1,a}$, the controller can grant or reject the request. Contrary to the previous example the controller now knows which train is being proposed/requests permission. If the controller grants the request, Train a is granted the sole choice of what to do with the permission.

Let us look at the edges from some of these states in detail.

\[ \begin{array}{c|cccccc}
 n_i: & 3 & 4 & 5 & 6 & \ldots & n \\
 CGS: & 27 & 256 & 3125 & 46656 & \ldots & n^a \\
 RCGS: & 10 & 35 & 127 & 462 & \ldots & \frac{(2n-1)!}{n!(n-1)!} \\
\end{array} \]

Table 3.6: Number of outgoing transitions from $q_0$ in CGS vs. RCGS in example model.
The trains have three options: voting, respectively, for a, b, or c. The votes in which a (the first action) is the majority choice or is tied to both of the other trains (tie-breaking lexicographically), leads to state $q_{1,a}$. When we transition to $q_{1,a}$, the role assignment changes and a is now placed, alone, in the role of designated train.

$q_{1,a}$ There are now two agents in the role train and they have only one action (hence no choice). There is one agent in the role designated train and it has one action (and hence no choice). Thus, the three outgoing edges from $q_{1,a}$ all start with 2 (two trains choosing their only action) and 1 (one train choosing its only action). The controller (the single agent in the controller role) has three actions and her choice corresponds to: deny the request for a (1,0,0), postpone decision (0,1,0), or grant permission for a (0,0,1).

$q_{2,a}$ Finally, in $q_{2,a}$, agent a has been designated and has been granted permission. Agent a now has three possible actions to choose from: waiting (1,0,0), entering the tunnel (0,1,0), or relinquishing the permission (0,0,1). The other agents have no choice of actions, and they all choose the one action available to them.

Regardless of how the designated train is picked out, we end up with a model consisting of nodes and edges as shown in Figure 3.8. In this figure we omit multiple arrows, as well as all actions and proposition symbols. The number of states is identical to the (smallest) corresponding CGS model, but the number of edges has been decreased significantly, and the as we illustrated in Example 3.6, even if we let the model scale to accommodate an increasing number of trains (but keep the roles fixed),
the number of edges increases in a way which is bounded above by a polynomial expression (we show this for the general case in Section 3.7).

With the intuition of how agents are assigned to homogeneous roles, and how we may increase heterogeneity or autonomy by permitting agents to leave and enter different roles, we now show how RCGS models function as a semantic for ATL and show that RCGS and CGS are equivalent model classes.

### 3.6 New Semantics for ATL

Recall from Chapter 2 that the language of ATL is defined inductively with respect to a countable set of proposition symbols $\Pi$ and a finite, non-empty set of agents $\mathcal{A}$, by the following BNF

$$
\phi ::= p \mid \neg \phi \mid \phi \land \phi \mid \langle \langle A \rangle \rangle \bigcirc \phi \mid \langle \langle A \rangle \rangle \Box \phi \mid \langle \langle A \rangle \rangle \phi \mathcal{U} \phi
$$

where $p \in \Pi$ and $A \subseteq \mathcal{A}$. In Chapter 2 we also gave the definition of satisfaction of ATL formulas with respect to states in CGS models.

We now give the definition of satisfaction of ATL formulas with respect to RCGS models. Before we do this we need to recall, and modify appropriately, the notions of strategies for a coalition and the possible outcomes of such strategies. The definition
may seem identical to the definition of satisfaction with respect to CGS models, but as the notion of actions and strategies have changed, we state it in full.

The \(A\)-profiles for \(r\) at \(q\) gives the possible ways agents in \(A\) that are in role \(r\) at \(q\) can act. Given a state \(q\) and a coalition \(A\), we define the set of \(A\)-profiles at \(q\):

\[
P(q, A) = \{ \langle v_1, \ldots, v_{|R|} \rangle \mid 1 \leq i \leq |R| : v_i \in P_r(q, A) \}
\]

For any \(v \in P_r(q, A)\) and \(w \in P_r(q, B)\) we write \(v \leq w\) iff for all \(i \in [\mathcal{A}(q, r)]\) we have \(v(i) \leq w(i)\). If \(v \leq w\), we say that \(w\) extends \(v\). If \(F = \langle v_1, \ldots, v_R \rangle \in P(q, A)\) and \(F' = \langle v'_1, \ldots, v'_R \rangle \in P(q, B)\) with \(v_i \leq v'_i\) for every \(1 \leq i \leq |R|\), we say that \(F \leq F'\) and that \(F'\) extends \(F\).

An \(A\)-profile \(F \in P(q, A)\) is a complete profile if, and only if, the sum of its components equals \(|\mathcal{A}|\). For a given state \(q\) denote this set of complete profiles in that state \(P(q)\). Given a (partial) profile \(F'\) at a state \(q\) we write \(\text{ext}(q, F)\) for the set of all complete profiles that extend \(F'\).

Given two states \(q, q' \in Q\), we say that \(q'\) is a successor of \(q\) if there is some \(F \in P(q)\) such that \(\delta(q, F) = q'\).

A computation is an infinite sequence \(\lambda = q_0q_1\ldots\) of states such that for all positions \(i \geq 0\), \(q_{i+1}\) is a successor of \(q_i\). We follow the standard abbreviations, hence \(q\)-computation denotes a computation starting at \(q\), and \(\lambda[i], \lambda[0,i]\) and \(\lambda[i,\infty]\) denote the \(i\)-th state, the finite prefix \(q_0q_1\ldots q_i\) and the infinite suffix \(q_iq_{i+1}\ldots\) of \(\lambda\) for any computation \(\lambda\) and position \(i \geq 0\).

An \(A\)-strategy for coalition \(A \subseteq \mathcal{A}\) is a function \(s_A : Q \to \bigcup_{q \in Q} P(q, A)\) such that \(s_A(q) \in P(q, A)\) for all \(q \in Q\). That is, \(s_A\) maps states to \(A\)-profiles at that state. The set of all \(A\)-strategies is denoted by \(\text{strat}(A)\). If \(s\) is an \(A\)-strategy and we apply \(\delta_q\) to \(s(q)\), we obtain a unique new state \(q' = \delta_q(s(q))\). Iterating, we get the induced computation \(\lambda_{s,q} = q_0q_1\ldots\) such that \(q = q_0\) and \(\forall i \geq 0 : \delta_{q_i}(s(q_i)) = q_{i+1}\).

Given two strategies \(s\) and \(s'\), we say that \(s \leq s'\) iff \(\forall q \in Q : s(q) \leq s'(q)\). Given an \(A\)-strategy \(s_A\) and a state \(q\) we get an associated set of computations \(\text{out}(q,s_A)\). This is the set of all computations that can result when, at any state, the players in \(A\) are acting in the way specified by \(s_A\):

\[
\text{out}(q,s_A) = \{ \lambda_{s,q} \mid s \in \text{strat}(A)\text{ and }s \geq s_A \}
\]

It will also be useful to have access to the set of states that can result in the next step when \(A \subseteq \mathcal{A}\) follows strategy \(s_A\) at state \(q\),

\[
\text{succ}(q,s_A) = \{ q' \in Q \mid \text{for some } F \in \text{ext}(q,s_A)\text{ s.t. } \delta(q,F) = q' \} = \{ \lambda[1] \mid \lambda \in \text{out}(q,s_A) \}
\]

**Definition 3.8.** Given a RCGS \(S\) and a state \(q\) in \(S\), we define the satisfaction relation

\[
\text{def}(q,A) = \{ v \in P(q,A) \mid \forall i \in [\mathcal{A}(q, r)] : v(i) \in R \}
\]
\( \models \) inductively:

\[
S, q \models p \iff p \in \pi(q)
\]

\[
S, q \models \neg \phi \iff \text{not } S, q \models \phi
\]

\[
S, q \models \phi \land \phi' \iff S, q \models \phi \text{ and } S, q \models \phi'
\]

\[
S, q \models \langle\langle A \rangle\rangle \diamond \phi \iff \text{there is } s_A \in \text{strat}(A) \text{ such that for all } \lambda \in \text{out}(s_A, q) \text{, we have } S, \lambda[1] \models \phi
\]

\[
S, q \models \langle\langle A \rangle\rangle \square \phi \iff \text{there is } s_A \in \text{strat}(A) \text{ such that for all } \lambda \in \text{out}(s_A, q) \text{ and all } i \geq 0 \text{ we have } S, \lambda[i] \models \phi
\]

\[
S, q \models \langle\langle A \rangle\rangle \phi \lor \phi' \iff \text{there is } s_A \in \text{strat}(A) \text{ such that for all } \lambda \in \text{out}(s_A, q) \text{, for some } i \geq 0 \text{ and all } 0 \leq j < i \text{ we have } S, \lambda[i] \models \phi' \text{ and } S, \lambda[j] \models \phi
\]

### 3.6.1 Equivalence between RCGS and CGS

The only impact of using RCGS as opposed to CGS is increased efficiency of model checking, and better scaling as the number of agents increase. We argue for the claim about efficiency in the next section. In this section we show that this is the only change, i.e., that the the formulas valid on RCGS models are exactly the same as the formulas valid on CGS models.

One way is obvious. For every CGS model, there is a corresponding RCGS model satisfying exactly the same formulas. To see this, we construct an RCGS with identical state-space in which every agent is assigned to her own role. In every state the agents in that role (there is always exactly one) has the same actions available as the corresponding agent in the original model.

The other direction requires a more technical argument. We do this by first giving a surjective function \( f \) that takes an RCGS and returns a CGS. Then we show that \( S \) and \( f(S) \) satisfy the same ATL formulas.

**Definition 3.9.** The translation function \( f \) from RCGS to CGS is defined as follows:

\[
f\langle\langle A, R, \rho, Q, \Pi, \Delta, \delta \rangle\rangle = \langle\langle A, Q, \Pi, \Delta, d, \delta' \rangle\rangle
\]

where:

- \( d_a(q) = \Delta(q, r) \) whenever \( a \in \rho(q, r) \)
- \( \delta'(q, \alpha_1, \ldots, \alpha_n) = \delta(q, v_1, \ldots, v_{|R|}) \) where for each role \( r \)
  \[
v_r = \langle \{|i \in \rho(q, r) | \alpha_i = 1\}, \ldots, \{|i \in \rho(q, r) | \alpha_i = \Delta(q, r)\} \rangle
\]

By the above argument that every CGS could be turned into an equivalent RCGS, we can easily see that \( f \) is surjective. It is easy to verify that any CGS \( S \) translated to a RCGS \( S_R \) as described above would satisfy \( S = f(S_R) \). A profile for a role \( r, v_r \), at
Given either a CGS or an RCGS $S$, we define the set of sets of states that a coalition $A$ can enforce in the next state of the game:

$$\text{enforce}(S, q, A) = \{\text{succ}(q, s_A) \mid s_A \text{ is a strategy for } A \text{ in } S\}.$$ 

The first thing we do towards showing equivalence is to describe a surjective function $m : \text{strat}(f(S)) \to \text{strat}(S)$ mapping action tuples and strategies of $f(S)$ to profiles and strategies of $S$ respectively. For all $A \subseteq \mathcal{A}$ and any action tuple for $A$ at $q$, $t_q = \langle \alpha_{a_1}, \alpha_{a_2}, \ldots, \alpha_{a_{|A|}} \rangle$ with $1 \leq \alpha_{a_i} \leq d_{a_i}(q)$ for all $1 \leq i \leq |A|$, the $A$-profile $m(t_q)$ is defined in the following way:

$$m(t_q) = \langle v(t_q, 1), \ldots, v(t_q, |R|) \rangle$$  where for all $1 \leq r \leq |R|$ we have

$$v(t_q, r) = \langle |\{a \in A_{q,r} \mid \alpha_a = 1\}|, \ldots, |\{a \in A_{q,r} \mid \alpha_a = |A|\}| \rangle$$

Thus the $i$-th component of $v(t_q, r)$ will be the number of agents from $A$ in role $r$ at $q$ that perform action $i$.

Given a strategy $s_A$ in $f(S)$ we define the strategy $m(s_A)$ for $S$ by taking $m(s_A)(q) = m(s_A(q))$ for all $q \in \mathcal{Q}$.

Surjectivity of $m$ is helpful since it means that for every possible strategy that exists in the RCGS $S$, there is a corresponding one in $f(S)$. This in turn means that when we quantify over strategies in one of $S$ and $f(S)$ we are implicitly also quantifying over strategies in the other. Showing equivalence, then, can be done by showing that these corresponding strategies have the same strength. Before we proceed, we give a proof of surjectivity of $m$.

### Lemma 3.10

For any RCGS $S$ and any $A \subseteq \mathcal{A}$, the function $m : \text{strat}(f(S), A) \to \text{strat}(S, A)$ is surjective.
Proof. Let $p_A$ be some strategy for $A$ in $S$. We must show there is a strategy $s_A$ in $f(S)$ such that $m(s_A) = p_A$. For all $q \in Q$, we must define $s_A(q)$ appropriately. Consider the profile $p_A(q) = \langle v_1, \ldots, v_{|R|} \rangle$ and note that by definition of a profile, all $v_r$ for $1 \leq r \leq |R|$ are $A$-profiles for $r$ and that by definition of an $A$-profile, we have $\sum_{1 \leq i \leq \delta(q,r)} v_r(i) = |A_{q,r}|$. Also, for all agents $a, a' \in A_{q,r}$ we know, by definition of $f$, that $d_a(q) = d_{a'}(q) = \mathbb{A}(q,r)$.

From this it follows that there are functions $\alpha : A \to \mathbb{N}^+$ such that for all $a \in A$, $\alpha(a) \in \alpha(v_r)$ and $\{a \in A_{q,r} \mid \alpha(a) = i\} = v_r(i)$ for all $1 \leq i \leq \mathbb{A}(q,r)$, i.e.

$$v_r = \langle \{a \in A_{q,r} \mid \alpha(a) = 1\}, \ldots, \{a \in A_{q,r} \mid \alpha(a) = \mathbb{A}(q,r)\} \rangle$$

We choose some such $\alpha$ and $s_A = \langle \alpha(a_1), \ldots, \alpha(a_{|A|}) \rangle$. Having defined $s_A$ in this way, it is clear that $m(s_A) = p_A$. \hfill \Box

Using the surjective function $m$ we can prove the following lemma, showing that the "next time" strength of any coalition $A$ is the same in $S$ as it is in $f(S)$.

**Lemma 3.11.** For any RCGS $S$, any state $q \in Q$ and any coalition $A \subseteq A$, we have $\text{enforce}(S,A,q) = \text{enforce}(f(S),A,q)$.

**Proof.** By definition of $\text{enforce}$ and Lemma 3.10 it is sufficient to show that for all $s_A \in \text{strat}(f(S),A)$, we have $\text{succ}(S,m(s_A),q) = \text{succ}(f(S),s_A,q)$.

(\subseteq) Assume that $q' \in \text{force}(S,m(s_A),q)$. Then there is some complete profile $P = \langle v_1, \ldots, v_{|R|} \rangle$, extending $m(s_A)(q)$, such that $\delta(q,P) = q'$. Let $m(s_A)(q) = \langle w_1, \ldots, w_{|R|} \rangle$ and form $P' = \langle v'_1, \ldots, v'_{|R|} \rangle$ defined by $v'_i = v_i - w_i$ for all $1 \leq i \leq |R|$. Then each $v'_i$ is an $(A \setminus A)$-profile for role $i$, meaning that the sum of entries in the tuple $v'_i$ is $|\{A \setminus A\}_{q,r}|$. This means that we can define a function $\alpha : A \to \mathbb{N}^+$ such that for all $a \in A$, $\alpha(a) \in d(a)(q)$ and for all $a \in A$, $\alpha(a) = s_a(q)$ and for every $r \in R$ and every $a \in (A \setminus A)$, and every $1 \leq j \leq \mathbb{A}(q,r)$, $|\{a \in (A \setminus A)_{q,r} \mid \alpha(a) = j\}| = v'_r(j)$.

Having defined $\alpha$ like this it follows by definition of $m$ that for all $1 \leq j \leq \mathbb{A}(q,r)$, $|\{a \in A_{q,r} \mid \alpha(a) = j\}| = w_r(j)$. Then for all $r \in R$ and all $1 \leq j \leq \mathbb{A}(q,r)$ we have $|\{a \in \rho(q,r) \mid \alpha(a) = j\}| = v_r(j)$. By definition of $f(S)$ it follows that $q' = \delta(q,P) = \delta'(q,\alpha)$ so that $q' \in \text{force}(f(S),s_A,q)$. We conclude that $\text{force}(S,f(s_A),q) \subseteq \text{force}(f(S),s_A,q)$.

(\supseteq) The direction $\supseteq$ follows easily from the definitions of $m$ and $f$.

\hfill \Box

Given a structure $S$ (with or without roles), and a formula $\phi$, we define $\text{true}(S,\phi) = \{q \in Q \mid S,q \models \phi\}$. Equivalence of models $S$ and $f(S)$ is now demonstrated by showing that the equivalence in next time strength established in Lemma 3.11 suffices to conclude that $\text{true}(S,\phi) = \text{true}(f(S),\phi)$ for all $\phi$.

**Theorem 3.12.** For any RCGS $S$, any $\phi$ and any $q \in Q$, we have $S,q \models \phi$ iff $f(S),q \models_{CGS} \phi$. 

We prove the theorem by showing that for all $\phi$, we have $true(S, \phi) = true(f(S), \phi)$. We use induction on complexity of $\phi$. The base case for atomic formulas and the inductive steps for Boolean connectives are trivial, while the case of $\langle A \rangle \bigcirc \phi$ is a straightforward application of Lemma 3.11. For the cases of $\langle A \rangle \Box \phi$ and $\langle A \rangle \phi \ll \psi$ we rely on the following fixed point characterizations, which are well-known to hold for ATL, see for instance [45], and are also easily verified against Definition 3.8:

\[
\langle A \rangle \bigcirc \phi \leftrightarrow \phi \land \langle A \rangle \bigcirc \langle A \rangle \bigcirc \phi \\
\langle A \rangle \phi \ll \psi \leftrightarrow \psi \lor (\phi \land \langle A \rangle \bigcirc \langle A \rangle \phi \ll \psi)
\] (3.13)

We show the induction step for $\langle A \rangle \Box \phi$, taking as induction hypothesis $true(S, \phi) = true(f(S), \phi)$. The first equivalence above identifies $Q' = true(S, \langle A \rangle \Box \phi)$ as the maximal subset of $Q$ such that $\phi$ is true at every state in $Q'$ and such that $A$ can enforce a state in $Q'$ from every state in $Q'$, i.e. such that $\forall q \in Q' : \exists Q'' \in force(q, A) : Q'' \subseteq Q'$. Notice that a unique such set always exists. This is clear since the union of two sets satisfying the two requirements will itself satisfy them (possibly the empty set). The first requirement, namely that $\phi$ is true at all states in $Q'$, holds for $S$ iff it holds for $f(S)$ by induction hypothesis. Lemma 3.11 states $force(S, q, A) = force(f(S), q, A)$, and this implies that also the second requirement holds in $S$ iff it holds in $f(S)$. From this we conclude $true(S, \langle A \rangle \Box \phi) = true(f(S), \langle A \rangle \Box \phi)$ as desired. The case for $\langle A \rangle \phi \ll \psi$ is similar, using the second equivalence.

This establishes the observation that CGS and RCGS are equivalent model classes for ATL. In the next section we show how model checking becomes more efficient for RCGS and argue that, particularly when specifying models where the number of agents in a fixed number of predefined roles, RCGS models are preferable to CGS models.

### 3.7 Model checking and the size of models

We have already seen that using roles can lead to a dramatic decrease in the size of ATL-models. In this section we give a more formal account, first by investigating the size of models in terms of the number of roles, players and actions, then by an analysis of model checking ATL over concurrent game structures with roles.

Given a set of numbers $[a]$ and a number $n$, it is a well-known combinatorial fact that the number of ways in which to choose $n$ elements from $[a]$, allowing repetitions, is $\frac{(n+(a-1))!}{n!(a-1)!}$. Furthermore, this number satisfies the following two inequalities:6

\[
\frac{(n+(a-1))!}{n!(a-1)!} \leq a^n, \quad \frac{(n+(a-1))!}{n!(a-1)!} \leq n^a
\] (3.14)

These two inequalities provide us with an upper bound on the size of RCGS models that makes it easy to compare their sizes to that of CGS models. Typically, the size of concurrent game structures is dominated by the size of the domain of the transition function. For an RCGS and a given state $q \in Q$ this is the number of complete profiles

---

6If this is not clear, remember that $n^a$ and $a^n$ are the number of functions $[n]^{[a]}$ and $[a]^{[n]}$ respectively. It should not be hard to see that all ways in which to choose $n$ elements from $a$ induce non-intersecting sets of functions of both types.
at \( q \). To measure it, remember that every complete profile is an \(|R|\)-tuple of profiles \( v_r \), one for each role \( r \in R \). It follows that \(|P(q)|\) is the set of all possible combinations of profiles for each role. Also remember that a profile \( v_r \) for \( r \in R \) is an \( \mathbb{A}(q,r) \)-tuple such that the sum of entries is the number of agents in that role, \(|\rho(q,r)|\). Equivalently, the profile \( v_r \) can be seen as the number of ways in which we can make \(|\rho(q,r)|\) choices, allowing repetitions, from a set of \( \mathbb{A}(q,r) \) alternatives. Looking at it this way, we obtain:

\[
|P(q)| = \prod_{r \in R} \frac{(|\rho(q,r)| + (\mathbb{A}(q,r) - 1))!}{|\rho(q,r)|!(\mathbb{A}(q,r) - 1))!}
\]

We sum over all \( q \in Q \) to obtain what we consider to be the size of an RCGS \( S \). In light of the inequalities in Equation 3.14, it follows that the size of \( S \) is upper bounded by both of the following expressions.

\[
O(\sum_{q \in Q} \prod_{r \in R} |\rho(q,r)|^{\mathbb{A}(q,r)}), \quad O(\sum_{q \in Q} \prod_{r \in R} \mathbb{A}(q,r)^{\rho(q,r)})
\]  

(3.15)

We observe that growth in the size of models is polynomial in \( a = \max_{q \in Q, r \in R} \mathbb{A}(q,r) \) if \( n = \mathcal{A} \) and \(|R|\) is fixed, and polynomial in \( p = \max_{q \in Q, r \in R} |\rho(q,r)| \) if \( a \) and \(|R|\) are fixed. This identifies a significant potential advantage arising from introducing roles to the semantics of ATL. The size of a CGS \( M \), when measured in the same way, replacing complete profiles at \( q \) by complete action tuples at \( q \), grows exponentially in the players whenever \( d_a(q) > 1 \) for each player \( a \).

We show that it is possible to use them to give the semantics of ATL, but this does not mean that there is not more to be said about them. Particularly crucial is the question of model checking over RCGS models.

### 3.7.1 Model checking using roles

For strategic logics, checking satisfiability is usually non-tractable, and the question of model checking is often crucial in assessing the usefulness of different logics. For ATL there is a well known “standard” algorithm, see e.g. [6]. It does model checking in time linear in the length of the formula and the size of the model. The algorithm is based on the fixed point Equation 3.13 from the proof of Theorem 3.12, so it will work also when model checking RCGS models. It is not clear, however, how the high level description should be implemented and, crucially, what the complexity will be in terms of the new parameters that arise.

Given a structure with roles, \( S \), and a formula \( \phi \), the standard model checking algorithm returns the set \( \text{true}(S, \phi) \), proceeding as detailed in Algorithms 1 and 2.

Given a structure \( S \), a coalition \( A \), a state \( q \in Q \) and a set of states \( Q' \), the method \texttt{enforce} answers true or false depending on whether or not \( A \) can enforce \( Q' \) from \( q \). That is, it tells us if at \( q \) there is \( Q'' \in \text{enforce}(q, A) \) such that \( Q'' \subseteq Q' \). Given a fixed length formula and a fixed number of states, this step dominates the running time of \texttt{mcheck} (Algorithm 1). It is also the only part of the standard algorithm that behaves in a different way after addition of roles to the structures. It involves the following steps:\footnote{In implementations one would seek to take advantage of information collected by repeating calls to \texttt{enforce} and not just do a Boolean check for every new instance in the way we do it here. This aspect is not crucial for our analysis, so we do not address it further.}
Algorithm 1 Model checking algorithm for RCGS (relies on enforce)

if $\phi = p \in \Pi$ then
  return $\pi(p)$
endif

if $\phi = \neg \psi$ then
  return $Q \setminus \text{mcheck}(S, \psi)$
endif

if $\phi = \psi \land \psi'$ then
  return $\text{mcheck}(S, \psi) \cap \text{mcheck}(S, \psi')$
endif

if $\phi = \langle A \rangle \bigcirc \psi$ then
  $Q_1 := Q$, $Q_2 := \text{mcheck}(S, \psi)$
  while $Q_1 \nsubseteq Q_2$ do
    $Q_1 := Q_2$, $Q_2 := \{q \in Q \mid \text{enforce}(S, A, q, Q_2)\} \cap Q_2$
  endwhile
  return $Q_1$
endif

if $\phi = \langle A \rangle \psi \lor \psi'$ then
  $Q_1 := \emptyset$, $Q_2 = \text{mcheck}(S, \psi)$, $Q_3 = \text{mcheck}(S, \psi')$
  while $Q_3 \nsubseteq Q_1$ do
    $Q_1 := Q_1 \cup Q_3$, $Q_3 := \{q \in Q \mid \text{enforce}(S, A, q, Q_1)\} \cap Q_2$
  endwhile
  return $Q_3$
endif

For all profiles $F \in P(q, A)$ the algorithm runs through all complete profiles $F' \in P(q)$ that extend $F$. Over CGSs, given a coalition $A$ and two action tuples $t = (\alpha_{a_1}, \alpha_{a_2}, \ldots, \alpha_{a_{|A|}}), t' = (\alpha'_{a_1}, \alpha'_{a_2}, \ldots, \alpha'_{a_{|A|}})$ for $A$ at $q$, the sets of complete action tuples that extend $t$ and $t'$ respectively do not intersect. It follows that running through all such extensions for all possible action tuples for $A$ at $q$ is at most linear in the total number of complete action tuples at $q$. This is no longer the case for RCGS models. Given two profiles $P, P'$ for $A$ at $q$, there can be many shared extensions. In fact, $P$ and $P'$ can share exponentially many in terms of the number of players and actions available.\footnote{To see this, consider $P = \langle v_1, v_2, \ldots, v_{|R|} \rangle$ and $P' = \langle v_1', v_2', \ldots, v_{|R|}' \rangle$. Each $v_r, v_r' \in P_A(q, r)$ sums to $\Sigma_{1 \leq j \leq A(q, r)} v_r(j) = |A_q|r$. Then form a complete profile $P'' = \langle v_1'', v_2'', \ldots, v_{|R|}''' \rangle$ at $q$ such that for all $1 \leq r \leq |R|$ and all $1 \leq j \leq A(q, r)$ we have $v_r''(j) = \max(v_r(j), v_r'(j))$. Then, if it exists, choose a coalition $A'$ such that $|A_q r'| = \Sigma_{1 \leq j \leq A(q, r)} v''_r(j)$. It is clear that the number of complete profiles that extends both $v$ and $v'$ is equal to the number of all $A \setminus A'$-profiles at $q$.} So, in general, running enforce requires us to make several passes through the set of all complete profiles, and the complexity is no longer linear. Still, it is polynomial in the number of complete profiles, since for any coalition $A$ and state $q$ we have $|P(q, A)| \leq |P(q)|$, meaning that the complexity of enforce is upper bounded by $|P(q)|^2$. It follows that model checking of ATL over RCGS is polynomial in the size of the model. We summarize this result.
Algorithm 2 $\text{enforce}(S,q,A,Q')$

\begin{algorithm}
\begin{algorithmic}
\For{$F \in P(q,A)$}
\State $p = true$
\For{$F' \in \text{ext}(q,F)$}
\If{$\delta(q,F') \notin Q'$}
\State $p = false$
\EndIf
\EndFor
\If{$p = true$}
\State \Return $true$
\EndIf
\EndFor
\State \Return $false$
\end{algorithmic}
\end{algorithm}

**Proposition 3.18.** Given a CGS $S$ and a formula $\phi$, $mcheck(S,\phi)$ takes time $\Theta(le^2)$ where $l$ is the length of $\phi$ and $e = \sum_{q \in Q} P(q)$ is the total number of transitions in $S$.

Since model checking ATL over CGSs takes only linear time, $\Theta(le)$, adding roles apparently makes model checking harder. On the other hand, the size of CGS models can be bigger by an exponential factor, making model checking much easier after adding roles. In light of the bounds we have on the size of models, c.f. Equation 3.15, we find that as long as the roles and the actions remain fixed, complexity of model checking is only polynomial in the number of agents. This is a potentially significant argument in favor of roles.

In practice, however, finding an optimal RCGS for a given CGS model $M$ might be at least as difficult as model checking on $M$ directly. It involves identifying the structure from $f^-(M)$ that has the minimum number of roles. In general, one cannot expect this task to have sub-linear complexity in the size of $M$. Roles should be used at the modeling stage, as they give the modeler an opportunity for exploiting homogeneity in the system under consideration. We think that it is reasonable to hypothesize that in practice, most large scale systems that lends themselves well to modeling by ATL do so precisely because they exhibit significant homogeneity. If not, identifying an accurate ATL model of the system, and model checking it, seems unlikely to be tractable at all.

The question arises as to whether or not using an RCGS is always the best choice, or if there are situations when the losses incurred in the complexity of model checking outweigh the gains we make in terms of the size of models. A general investigation of this in terms of how fixing or bounding the number of roles affect membership in complexity classes is left for future work. Here, we conclude with the following proposition which states that as long we use the standard algorithm, model checking any CGS $M$ can be done at least as quickly by model checking an arbitrary $S \in f^-(M)$.

**Proposition 3.19.** Given any CGS-model $M$ and any formula $\phi$, let $c(mcheck(M,\phi))$ denote the complexity of running $mcheck(M,\phi)$. We have, for all $S \in f^-(M)$, that complexity of running $mcheck(S,\phi)$ is $\Theta(c(mcheck(M,\phi)))$.

\(^9\)Although in many practical cases, when models are given in some compressed form, the situation might be such that it is possible. The question of how to efficiently find small RCGS-models will be investigated in future work.
Proof. It is clear that for any \( S \in f^{-1}(M) \), a difference in overall complexity of running \( mcheck(S, \phi) \) and \( mcheck(M, \phi) \) can arise only from a difference in the complexity of \( \text{enforce} \). So we compare the complexity of \( \text{enforce}(S, A, q, Q') \) for a RCGS \( S \) and \( \text{enforce}(M, A, q, Q') \) for a CGS \( M \) for some arbitrary \( q \in Q \). The complexity in both cases involves passing through all complete extensions of all strategies for \( A \) at \( q \). The sizes of these sets are compared as follows, the first inequality is an instance of Equation 3.14 and the equalities follow from definition of \( f \) and the fact that \( M = f(S) \).

\[
\prod_{r \in R} \left( \frac{|A_{q,r}| + (A(q,r) - 1)!}{|A_{q,r}|!(A(q,r) - 1)!} \right) \times \prod_{r \in R} \left( \frac{(|\rho(q,r)| - |A_{q,r}|) + (A(q,r) - 1)!}{(|\rho(q,r)| - |A_{q,r}|)!(A(q,r) - 1)!} \right) \\
\leq \left( \prod_{r \in R} A(q,r)^{|A_{q,r}|} \times \prod_{r \in R} A(q,r)^{|\rho(q,r)| - |A_{q,r}|} \right) \\
= \prod_{r \in R} \left( \prod_{a \in A_{q,r}} A(q,r) \right) \times \prod_{r \in R} \left( \prod_{a \in \rho(a,r) \setminus A_{q,r}} A(q,r) \right) \\
= \left( \prod_{a \in A} d_a(q) \times \prod_{a \in A \setminus A} d_a(q) \right) = \prod_{a \in A} d_a(q)
\]

We started with the number of profiles (transitions) we need to inspect when running \( \text{enforce} \) on \( S \) at \( q \), and ended with the number of action tuples (transitions) we need to inspect when running \( \text{enforce} \) on \( M = f(S) \). Since we showed the first to be smaller or equal to the latter and the execution of all other elements of \( mcheck \) are identical between \( S \) and \( M \), the claim follows.

\[\square\]

3.8 Summary

In this chapter we have defined \textit{concurrent game structures with roles} (RCGS) which add the construct of \textit{homogeneous roles} to the (traditional) \textit{concurrent game structures} (CGS) introduced in [6].

Agents belonging to the same homogeneous role are \textit{anonymous} as the term is used in voting theory, social choice theory and game theory (see Chapter 1). That is to say, that they are strategically homogeneous; able to bring about exactly the same effects through their available actions. As long as the role assigned to an agent is allowed to depend on the state, as it allowed in our Definition 3.4, it follows from Lemma 3.10 and Theorem 3.12 that RCGS and CGS are equivalent model classes for ATL.

The notion of anonymity, or strategic homogeneity, has import in many MAS settings which occur naturally when studying social procedures formally. It leads to structural constraints which lend themselves to important theories, such as Arrow’s impossibility theorem, as well as computational benefits from simplicity of representation. In [25], they give an efficient algorithm for computing equilibria in anonymous games. There, they make the observation that the case where the entire agent set is anonymous
3.8 Summary

easily generalizes to cases where the agents can be partitioned into types, where the agents are anonymous modulo their type. The type of an agent is hence similar from our notion of role here. Agents may leave or enter into roles as the system evolves, but their type does not change. Hence, types correspond to roles with static membership, as explored in the next two chapters.

The roles in our construct occur in many natural settings. When modeling large agent sets for particular situation in which we do not know details about the agents, we have can use roles to simplify our models. For instance, when modeling the behavior of “consumers” in a market, we ascribe the same strategic abilities to all agents (similar to type). If what the agent is permitted or able to do depends on her individual properties (say cash balance), we can have two types of consumers; increasing heterogeneity while benefiting from the simple symmetric representation of each type. Such a static labeling of consumers entails that consumers can not change their properties sufficiently to enter into a new role. If a consumer with a low balance is unable to perform an expensive action, but still chooses a strategy which allows her to accumulate more money, how would this affect the model? To model such dynamic role assignments, we need, as we have argued in this chapter, the entire framework of ATL. The notion of roles permits us to retain the benefit of strategic homogeneity, which can be substantial for large number of agents, to also such dynamic models.

Consider alternatively an object oriented software system consisting of a large number of processes which instantiate the same class. Consider further, in one extreme case, that all variables that these processes have access to/are “global” or “static”.10 (We ignore problems of synchronization and interference from objects which are not one of the discussed processes). In this case, all the processes have the same strategic abilities, they are all of the same type. Alter the object so that there is also a boolean local variable and that this variable effects the process it belongs to. There are now two types of processes; those with the variable set to true, and those with the variable set to false. If this variable can be altered during the execution of the system, we are not dealing with two types of processes, but two roles which the processes enact. Generally, when the effect of local variables on strategic capabilities can be partitioned and is susceptible to alteration, we can describe the homogeneous roles our processes/instances may occupy and model the overall evolution of the system with a RCGS taking advantage of the symmetries which may occur, whenever they do occur.

In this chapter we have presented the semantic structure RCGS and shown that as a model class for ATL, they are equivalent to the traditional CGS. We have also shown that the model checking problem, although it now seems to belong to a higher class, is never worse than model checking the corresponding CGS model. The fact that RCGS is equivalent to CGS is in one way trivial (we can always let every agent have a designated role), but also somewhat odd. What exactly is the difference between homogeneous CGS models as presented here and arbitrary CGS models? The difference will be made clear in the next chapter where we show that strategic homogeneity is the same as requiring that the role association function \( \rho \) is fixed. This leads also to a new logic that we will axiomatize.

---
10 By “global” or “static” here we mean the same thing; that the content of the variable is equally accessible, for reading or writing, to all the processes. In, say Java, such variables are called “static”.
Chapter 4

Axiomatic Reasoning

4.1 Background

As we demonstrated in the previous chapter, RCGS models yield the same validities as those of CGS models. Recall that we relied on the agents being assigned to roles per state. In this chapter, we are interested in the RCGS models where we have a fixed number of roles and where the role assignment is invariant over states. Towards the end of the chapter, we will weaken this assumption somewhat and illustrate the effect of having a fixed number of roles and for each role a fixed number of participants in each role. That is, we will permit the agents to change roles, but we require that each role is occupied by a constant number of agents throughout the model.

In this chapter, we introduce two languages, $\mathcal{L}_{\text{HATL}}$ (for Homogeneous ATL) and a corresponding reasoning system which we show to be complete, and $\mathcal{L}_{\text{RATL}}$ (for ATL with Roles) and a corresponding reasoning system which we then show to be equivalent to the former. In $\mathcal{L}_{\text{HATL}}$ we will not have direct references to any agents.

Recall that agents which are in the same role are indistinguishable, but we were able to introduce some heterogeneity to our set of agents by letting some agents change their role, as we did in Example 3.7. With state invariant role assignments, this is no longer possible. This entails that whether we refer to “train a” or “train b” when making a query about the strategic abilities of the a train, is irrelevant. As it is no longer relevant which train we are referring to we will change how agents are referred to from discussing “train a” to simply discussing “a train”.

4.2 Constellations

Recall from the discussion of the term “role” in the previous chapter, that we take roles to be a homogeneous collection of agents; agents which are all in the same position. That is, an agent is in some role if, and only if, that agent is occupying a position which is constitutive for that role. For example, we can imagine having a system consisting of, among other roles, the role “teacher” enacted by six agents. We associate the role “teacher” with six “teacher” positions.

\[\text{In the vocabulary of (algorithmic) game theory (see e.g., Daskalakis & Papadimitriou [25]), we assign the agents types rather than roles.}\]
Less abstractly than is usual for ATL, we may consider the systems we are describing in terms of first-order modal logic (see e.g., [37]). We will not give an extensive analysis at this level, but ground our use of some terms that will be useful to us. CGS models are a particular kind of Kripke models, and when giving an analysis of the states in these models, we may say that the object domain of every possible world contains at least all the agents.

In traditional CGS models, these agents are referred to by name. We formulate statements such as

\[(\text{agent}) a \text{ can bring about } \phi\]

where \(a\) is taken to be the proper name of an agent.

In an RCGS model, we take the role an agent is enacting to be a predicate the enacting agent satisfies. We then refer to this agent by this description, i.e., we are departing from referring by proper names to referring by descriptions. We have assumed that the role of an agent completely determines all her properties, particularly her strategic abilities in a given state. The predicates we need to employ to refer to the acting objects in our domain then, are precisely the positions and roles.

The teacher role is populated by agents in teacher positions, or, as we interpret it here, satisfy the “teacher” predicate. The statements we formulate now, no longer referring to the atomic acting object by name, are of the form

\[\text{a teacher can bring about } \phi\]

Our references in the singular (the atomic acting object) have thus changed from a proper name to that of a description formulated as a position.

When we are discussing the strategic abilities of these atomic acting objects, we get a simple enough interpretation of the “next-time” fragment of our language. When we say

\[\text{a teacher can ensure that there will be a test tomorrow}\]

we may elect some agent (which satisfies the predicate “teacher”), and verify that indeed that teacher may ensure this.

The interpretation of the generalized modal connectives \(\Box\), \(\Diamond\), and \(\mathcal{U}\), however, are different. We may say that

\[\text{a teacher can make sure Paul gets through his education while having all his questions answered when they are presented}\]

It might be that the agent enacting the role teacher (of Paul) remains fixed throughout the run, but, as discussed in the previous chapter, the actual agent enacting a role (or occupying a position) is not relevant. Rather, we should interpret the statement as:

\[\text{in every state in which Paul is still a pupil, the/an agent enacting the role teacher (of Paul) is able to answer the questions Paul may present}\]

Which agent should we assign this task to, or verify is able to satisfy this goal? None. We assign the task to the position, and verify that the position is able to satisfy this goal.
This is how we often think about the abilities of such unnamed agents. Indeed it is useful to refer to agents in this manner when we are describing some property we want our system to satisfy when we suppose the identity of the role enactors may change. Suppose that we are trying to establish a social structure where, at all times, *some* agent is pivotal in the progress of the system. This might be a system similar to the hierarchical systems of distributed computing discussed in Chapter 1 (in which some component is the “master”), or a more intuitive scenario where *the president* is responsible for approving a bill.

We may say

“the president can implement a fair taxation system”

(assuming such a taxation system exists). We are not discussing the abilities of any one particular agent, rather we are describing *the position* itself.

Both the previous examples (about “a teacher” and “the president”) are examples of references to agents by description, rather than names. This is a deep problem in philosophical ((first-order) modal) logic, and we must hasten to state that we are not addressing the question of descriptions as references. Indeed not even the nuance which has received attention by philosophers between definite and indefinite descriptions (see [57] for a recent overview). Our investigation is only a very simple special care, because the predicates which we permit are very limited.

If \( r \) is some role (e.g., “teacher” or “president”) the predicate \( r \) refers to some agent in \( r \) (i.e., “a teacher” or “the president”). Whether our description is definite or indefinite is hence given precisely by the size of the role we are describing, and this size is a constant given as a parameter to our language. In other words, our descriptions are indefinite unless the role contains exactly one agent at all times (in which case it is definite, but still not proper since no unique individual is picked out).

Furthermore, since we only have predicates for these positions, and all agents in the same position have the same properties, which teacher we “pick out” when discussing “a teacher” is irrelevant. Hence, the only definite descriptions we are permitted to use are those which describe roles enacted by a single agent, and every description of an agent enacting a role consisting of more than one agent will be indefinite.

The atomic acting object in a CGS model is the agent. In RCGS models, agents are still the acting objects, but in this chapter we will only refer to them using positions. Hence, we will abstract away from agents altogether and talk as though positions themselves “act” on their environment. Similarly, we will introduce a notion corresponding to the coalition. In a CGS model, we interpret statements (about coalitions) like

“(agents) \( a \), \( b \) and \( c \) can bring about \( \phi \)”

when we say that they have a joint strategy which ensures \( \phi \). As we no longer have/need access to the names, a corresponding statement interpreted in a RCGS could then be (assuming \( a \) and \( b \) are be teachers, and \( c \) is a gymnast):

“two teachers and a gymnast can bring about \( \phi \)”.

We refer to such predicates as *constellations*. A constellation is a predicate on the form “(exactly) \( n \) (distinct) \( r \)”, or a conjunction of such predicates with at most one occurrence of any given role \( r \).
The term constellation, similarly to the term role, is used quite frequently when
discussing human affairs. Constellations of power, like “the cabinet” in politics, or
the “management” in organizations. Also as certain regular patterns; the “father-son
constellation”, “a constellation of symptoms” and so on. We attempt here to give this
term a formal characterization which does not depart too far from the colloquial or other
non-formal uses. Unlike “role”, I have not come across definitions or precise usage of
“constellation” in the literature.

A predicate in the appropriate form which is not instantiated by a coalition is not un-
derstood as a constellation in this chapter in order to avoid non-designating terms. For
some set of agents $\mathcal{A}$ and a role assignment $\rho : \mathcal{A} \rightarrow R$, the constellation corresponding
to $\mathcal{A}$ under $\rho$ is an $R$ vector of natural numbers

$$
\Sigma = (|A_1|, \ldots, |A_{|R|}|)
$$

where

$$
A_r = \{ a \in \mathcal{A} \mid \rho(a) = r \}
$$

We will refer to this constellation $\Sigma$ as the grand constellation (just as we refer to
the set of agents $\mathcal{A}$ as the grand coalition) or as the social structure (in cases ana-
logous to when it would be natural to refer to the set of agents $\mathcal{A}$ as the society). The
smallest constellation, of course, is a vector consisting of $R$ copies of the number zero.
This constellation is satisfied by exactly one coalition, the empty coalition. The small-
est non-zero constellation, i.e., constellations which sum of elements equals one, we
occasionally call positions.

Coalitions may be formed by different coalitions by the standard set operations, but
collections are constructed by arithmetic operations. Hence, the composition of two
collections is not closed. We will call two constellations are composable if their sum
is instantiated by a coalition. That is, a vector $\sigma$ of numbers is a constellation if

$$
0 = (0, \ldots, 0) \leq \sigma = (\sigma_1, \ldots, \sigma_{|R|}) \leq \Sigma = (\Sigma_1, \ldots, \Sigma_{|R|}) = (|A_1|, \ldots, |A_{|R|}|)
$$

For two constellations $0 \leq \sigma, \sigma' \leq \Sigma$, we say that $\sigma$ and $\sigma'$ are composable if, and
only if, $\sigma + \sigma' \leq \Sigma$.

**Definition 4.1 (Constellation).** Given the (non-empty) set of agents $\mathcal{A}$, a non-empty set
of roles $R$ and a state independent role assignment $\rho$, the grand constellation of $\mathcal{A}$ is
the vector $\Sigma \subseteq \mathbb{N}^R$ where $\Sigma_r = |A_r|$ for all $r$. Unless the parameters are clear from the
context, we denote the grand constellation for $\mathcal{A}$ with roles $\rho$, $\Sigma(\mathcal{A}, \rho)$.

Given a coalition $A \subseteq \mathcal{A}$, the constellation of a coalition $A$ is $\sigma_A \leq \Sigma(\mathcal{A}, \rho)$ where
for every role $r$, $(\sigma_A)_r = |A_r|$.

As mentioned, we require that constellations never fail to designate. We ensure
this in two different ways. When specifying the language HATL we assume that the
role assignment is state independent. We show completeness for a logic based on this
assumption. Towards the end of the chapter, in Example 4.30, to illustrate the difference
between coalitions and constellations (hence between agents and positions), we relax
this assumption and simply require that the grand constellation is invariant in the model,
i.e., that the social structure is constant.
4.3 The logic HATL

We define the language, $L_{\text{HATL}}$, where we have constellations $\sigma$ rather than coalitions.

**Definition 4.2.** The language of $L_{\text{HATL}}$ for agents $A$ distributed into $R \geq 1$ roles by a state independent role assignment $\rho$ over propositional symbols $\Pi$, is given by the following BFN

$$\phi ::= p \mid \neg \phi \mid \phi \land \phi \mid \langle \langle \sigma \rangle \rangle \bigcirc \phi \mid \langle \langle \sigma \rangle \rangle \Box \phi \mid \langle \langle \sigma \rangle \rangle \phi \cup \phi$$

where $p \in \Pi$ and $0 \leq \sigma \leq \Sigma(A, \rho)$. We refer to formulas on the form $\langle \langle \sigma \rangle \rangle \bigcirc \phi$ or $\neg \langle \langle \sigma \rangle \rangle \bigcirc \phi$ as $\bigcirc$-formulas. That is, the $\bigcirc$-formulas, are the formulas which has as its outermost connective either a connective $\langle \langle \sigma \rangle \rangle \bigcirc$, or is the negation of one such connective. A formula is in normal form if, and only if, every negation symbol occurs directly in front of a propositional symbol or a strategic modal connective.

Similarly to the terminology for coalitions, we have the notion of a profile and a strategy for a constellation. In any given state $q$, the profiles for $\sigma$ for a role $r$ is

$$P_r(q, \sigma) = \left\{ v \in \{0, \ldots, \sigma_r\}^{[A_{q,r}]} \mid \sum_{1 \leq i \leq A_{q,r}} v(i) = \sigma_r \right\} \quad (4.3)$$

Notice that this definition is slightly different than the corresponding definition for coalitions. We are not counting agents, but rather we have a number for every role $\sigma_r$ which specifies the sum of the elements in the profile. A $\sigma$-profile is then

$$P(q, \sigma) = \left\{ \langle v_1, \ldots, v_{|R|} \rangle \mid 1 \leq i \leq |R| : v_i \in P_r(q, \sigma) \right\} \quad (4.4)$$

The $\sigma$-profiles for a role $r$ in a state $q$ is $P_r(q, \sigma)$ (if $\sigma = \Sigma$ these are the complete profiles for $r$ at $q$), and the $\sigma$-profiles (respectively, complete profiles) at $q$ is $P(q, \sigma) = \prod_{r \in R} P_r(q, \sigma)$ (respectively $P(q) = P(q, \Sigma)$). This naturally also extends to the notion of a strategy. For a constellation $\sigma$, an $\sigma$-strategy is a function $s_\sigma : Q \rightarrow \bigcup_{q \in Q} P(q, \sigma)$ such that for every state $q \in Q$, $s_\sigma(q) \in P(q, \sigma)$. We define $\text{strat}(\sigma)$ to be the collection of these $\sigma$-strategies.

Even though we have defined terms, such as a strategy, which are agent-centric artifacts, we have defined them without reference to agents at all. Constellations are always instantiated by a coalition, but which agent performs any given action being performed is irrelevant. This permits us to define satisfaction of formulas in the new language over RCGS models. We will offer an alternative definition of the language in Section 4.5.

**Definition 4.5.** Given a RCGS $S$ with state independent role assignment and a state $q$, ...
we define the satisfaction relation \( \models_{\text{HATL}} \) (denoted \( \models \) for brevity) inductively

\[
\begin{align*}
S, q \models p & \iff p \in \pi(q) \\
S, q \models \neg \phi & \iff \text{not } S, q \models \phi \\
S, q \models \phi_1 \land \phi_2 & \iff S, q \models \phi_1 \text{ and } S, q \models \phi_2 \\
S, q \models \langle \langle \sigma \rangle \rangle \circ \phi & \iff \text{there is } s_{\sigma} \in \text{strat}(\sigma) \text{ such that } \\
& \quad \text{for all } \lambda \in \text{out}(s_{\sigma}, q), \text{ we have } S, \lambda[1] \models \phi \\
S, q \models \langle \langle \sigma \rangle \rangle \Box \phi & \iff \text{there is } s_{\sigma} \in \text{strat}(\sigma) \text{ such that } \\
& \quad \text{for all } \lambda \in \text{out}(s_{\sigma}, q) \text{ and all } i \geq 0 \\
& \quad \text{we have } S, \lambda[i] \models \phi \\
S, q \models \langle \langle \sigma \rangle \rangle \phi_1 \cup \phi_2 & \iff \text{there is } s_{\sigma} \in \text{strat}(\sigma) \text{ such that } \\
& \quad \text{for all } \lambda \in \text{out}(s_{\sigma}, q), \text{ for some } i \geq 0 \text{ and all } 0 \leq j < i \\
& \quad \text{we have } S, \lambda[i] \models \phi_2 \text{ and } S, \lambda[j] \models \phi_1
\end{align*}
\]

### 4.4 Completeness of HATL

In this section, we show completeness of HATL. The proof is analogous to the completeness proof of ATL in [40] and we will omit certain parts which are identical for the two proofs.

Fix a consistent formula \( \Psi \). We construct a RCGS \( S_\Psi \) with state independent role assignment which satisfies \( \Psi \) (Definition 4.23). We will construct this tree in several steps. As in [40] we construct an auxiliary structure, a tree, which will be the basis for this RCGS.

**Definition 4.6 ((Extended) Closure).** The closure of \( \Psi \), denoted \( cl(\Psi) \), is the smallest set satisfying the following conditions

- \( \Psi \) and every subformula of \( \Psi \) is in \( cl(\Psi) \).
- If \( \langle \langle \sigma \rangle \rangle \Box \phi \in cl(\Psi) \), then \( \langle \langle \sigma \rangle \rangle \circ \langle \langle \sigma \rangle \rangle \Box \phi \in cl(\Psi) \).
- If \( \neg \langle \langle \sigma \rangle \rangle \Box \phi \in cl(\Psi) \), then \( \neg \langle \langle \sigma \rangle \rangle \circ \langle \langle \sigma \rangle \rangle \Box \phi \in cl(\Psi) \).
- If \( \langle \langle \sigma \rangle \rangle \phi_1 \cup \phi_2 \in cl(\Psi) \), then \( \langle \langle \sigma \rangle \rangle \circ \langle \langle \sigma \rangle \rangle \phi_1 \cup \phi_2 \in cl(\Psi) \).
- If \( \neg \langle \langle \sigma \rangle \rangle \phi_1 \cup \phi_2 \), then \( \neg \langle \langle \sigma \rangle \rangle \circ \langle \langle \sigma \rangle \rangle \phi_1 \cup \phi_2 \in cl(\Psi) \).
- If \( \phi \in cl(\Psi) \), then \( \sim \phi \in cl(\Psi) \), where \( \sim \phi \) is the normal form of \( \neg \phi \).

Define the auxiliary set \( cl^+(\Psi) \) as \( cl(\Psi) \cup \{ \langle \langle \sigma \rangle \rangle \circ \phi \mid \langle \langle \sigma \rangle \rangle \circ \phi \in cl(\Psi) \text{ and } 0 \leq \sigma' \leq \Sigma \} \).

The extended closure of \( \Psi \), denoted \( ecl(\Psi) \), is the set \( cl^+(\Psi) \) closed under all \( \land \) and \( \lor \) up to logical equivalence. Notice that \( ecl(\Psi) \) is a finite set.

In [40], the tree constructed is a tree of fixed branching degree. The nodes in the tree will correspond closely to the states in the final model. Every non-leaf tree has branching degree \( k^n \), from the fact that all the agents, in every state of the constructed model, have \( k \) actions available to them. When we in this chapter construct a RCGS in
a similar fashion as the CGS constructed in [40], we will also fix the number of actions available to every agent to a constant $A$, and the tree we construct has fixed branching degree.

Every agent, in every role, will have a number of actions available to her which is dependent on the set of $\circ$-formulas in some set of $\Phi$. Let $\Phi_\oplus$ and $\Phi_\ominus$ be the subsets consisting of the positive, respectively negative, $\circ$-formulas in the set of formulas $\Phi$.

$$\Phi_\oplus := \{ \theta \in \Phi \mid \theta = \langle\langle \sigma \rangle\rangle \circ \phi \}$$

$$\Phi_\ominus := \{ \theta \in \Phi \mid \theta = \neg\langle\langle \sigma \rangle\rangle \circ \phi \}$$

We also introduce some constants for two particular sets of $\circ$-formulas:

$$\Psi_\oplus := \{ \theta \in \text{ecl}(\Psi) \mid \theta = \langle\langle \sigma \rangle\rangle \circ \phi \}$$

$$\Psi_\ominus := \{ \theta \in \text{ecl}(\Psi) \mid \theta = \neg\langle\langle \sigma \rangle\rangle \circ \phi \}$$

We will make some more assumptions on the content of these sets when we specify our local construction. The local construction will depend on a set $\Phi$ which we will show satisfies the inequality $|\Phi_\oplus| + |\Phi_\ominus|^2 \leq |\Psi_\oplus| + (|\Psi_\ominus| + 1)^2$ which will be a condition in the lemma where it is relevant (e.g., Lemma 4.16). Every agent, regardless of state and role, will have the same actions available $A(q, r) = \Phi_\oplus + \Phi_\ominus + x$, where $x$ is $(|\Psi_\oplus| + (|\Psi_\ominus| + 1)^2) - (|\Phi_\oplus| + |\Phi_\ominus|^2)$. As this is a constant, we will just write $A$. That is, there are three types of actions, there first type are votes for positive $\circ$-formulas, the second type are votes for negative $\circ$-formulas, and finally some actions are “buffer actions” of no specific consequence which we keep to ensure that every agent has a fixed number of actions available. As the number of roles, and the assignment of roles to the agents, is fixed, the number of complete action profiles will be fixed in every state. The number of actions do not depend on the size of the set $\Phi$ or the subsets just specified. As discussed in Chapter 3, the number of complete profiles (in any state) will be

$$P = \prod_{r \in R} \left( \frac{|A_r| + |A| - 1)!}{|A_r|!(|A| - 1)!} \right)$$

In [40], a tree is defined as a prefix closed set of sequences of natural numbers, the empty sequence $\varepsilon$ is the root of the tree. An intuitive reading of these trees is that if $t$ is a node in the tree, and $0 \leq c < k^n$ is a complete action profile, then $t \cdot c$ is the successor of $t$ if $c$ is the complete profile, where $\cdot$ is the concatenation operator. A node in the tree, is hence represented by a sequence of complete action profiles which progresses from the root to that state.

We will define trees in the same way, but give some more structure to the labels compared to the natural numbers encoding the trees in [40]. The formulations, of course, are equivalent, but the representation as explicit action profiles will simplify some of the argumentation later.

**Definition 4.7.** A labeled tree over $\Theta$ is a pair $\langle T, V \rangle$ where

- $T$ is a prefix-closed set of sequences of elements in $P$ containing at least the empty sequence $\varepsilon$, and
• \( V : T \rightarrow \Theta \) is a labeling.

A simple labeled tree, is a labeled tree \( \langle T, V \rangle \) which consists of a root and all, and only, the immediate successors of it \( (T = \{ \epsilon \} \cup P) \).

The intuition from [40] extends to the trees defined in this manner. If \( v \in P \) and \( t \in T \) is a non-leaf node in some tree, then when we later consider \( t \) a state, the successor state which obtains when the agents perform the complete profile \( v \) will be represented by the child \( t \cdot v \in T \) of \( t \).

We can now assign a number to each element of \( P \), and use the construction of trees as prefixed-closed sets of natural numbers. As we mentioned it will be easier to consider the elements of \( P \) as complete profiles.

We extend the terminology we have for describing particular subsets of successor states in RCGS models to trees. As the number of actions available to the agents in every role and in every state is constant, every partial profile will define a set of complete profiles extending it. This set will even be independent of the state. For any constellation \( 0 \leq \sigma \leq \Sigma \) and state \( q \), the set of complete profiles extending the \( \sigma \)-profile \( v_\sigma \in P(q, \sigma) \), is the set

\[
\text{ext}(q, v_\sigma) := \{ v \in P(q) \mid \text{for every } 1 \leq i \leq |R| : v_\sigma(i) \leq v(i) \}
\]

### 4.4.1 Axiomatization

The axiomatization we present is based on that of [40], but we have constellations rather than coalitions.

**Table 4.1:** Axioms of HATL for every \( \sigma, \sigma' \leq \Sigma \).

<table>
<thead>
<tr>
<th>Axiom</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(TAUT)</td>
<td>Propositional tautologies</td>
</tr>
<tr>
<td>((\perp))</td>
<td>( \langle \langle \sigma \rangle \rangle \circ \perp )</td>
</tr>
<tr>
<td>((\top))</td>
<td>( \langle \langle \sigma \rangle \rangle \circ \top )</td>
</tr>
<tr>
<td>((\Sigma))</td>
<td>( \neg \langle \langle 0 \rangle \rangle \circ \neg \phi \rightarrow \langle \langle \Sigma \rangle \rangle \phi )</td>
</tr>
<tr>
<td>((S))</td>
<td>( \langle \langle \sigma \rangle \rangle \circ (\phi_1 \land \langle \langle \sigma' \rangle \rangle \circ \phi_2 \rightarrow \langle \langle \sigma + \sigma' \rangle \rangle \circ (\phi_1 \land \phi_2) ) whenever ( \sigma + \sigma' \leq \Sigma )</td>
</tr>
<tr>
<td>((\mathbf{FP}_\square))</td>
<td>( \langle \langle \sigma \rangle \rangle \square \phi \leftrightarrow \phi \land \langle \langle \sigma \rangle \rangle \circ \langle \langle \sigma \rangle \rangle \square \phi )</td>
</tr>
<tr>
<td>((\mathbf{GFP}_\square))</td>
<td>( \langle \langle \theta \rangle \rangle \square (\theta \rightarrow (\phi \land \langle \langle \sigma \rangle \rangle \circ \theta)) \rightarrow \langle \langle \theta \rangle \rangle \square (\theta \rightarrow \langle \langle \sigma \rangle \rangle \square \phi) )</td>
</tr>
<tr>
<td>((\mathbf{FP}_U))</td>
<td>( \langle \langle \sigma \rangle \rangle \phi_1 \sqcup \phi_2 \leftrightarrow \phi_2 \lor (\phi_1 \land \langle \langle \sigma \rangle \rangle \circ \langle \langle \sigma \rangle \rangle \phi_1 \sqcup \phi_2) )</td>
</tr>
<tr>
<td>((\mathbf{LFP}_U))</td>
<td>( \langle \langle \theta \rangle \rangle (\phi_2 \lor (\phi_1 \land \langle \langle \sigma \rangle \rangle \circ \theta)) \rightarrow \theta \rightarrow \langle \langle \theta \rangle \rangle \square (\langle \langle \sigma \rangle \rangle \phi_1 \sqcup \phi_2 \rightarrow \theta) )</td>
</tr>
</tbody>
</table>

**Table 4.2:** Inference rules for HATL for every \( \sigma \leq \Sigma \).

\[
\begin{align*}
\phi_1, \phi_2 & \quad \Rightarrow \quad \phi_2 & (\text{MP}) \\
\phi_1 \rightarrow \phi_2 & \quad \Rightarrow \quad \langle \langle \sigma \rangle \rangle \circ \phi_1 \rightarrow \langle \langle \sigma \rangle \rangle \circ \phi_2 & (\langle \langle \sigma \rangle \rangle \circ\text{-monotonicity}) \\
\phi & \quad \Rightarrow \quad \langle \langle 0 \rangle \rangle \square \phi & (\langle \langle 0 \rangle \rangle \square\text{-necessitation})
\end{align*}
\]
As is usual, we define \( \vdash \subseteq L_{\text{HATL}} \) to be the set containing all formulas \( \phi \) (and denote membership \( \vdash \phi \)), which are instances of the axioms, or for which there are formulas \( \psi, \psi' \) such that \( \vdash \psi, \vdash \psi' \) and there is a rule for which one of

\[
\begin{align*}
\vdash \psi \quad \text{or} \quad \vdash \psi' ; \vdash \phi
\end{align*}
\]

is an instance of a rule.

We define a constant set of sets of formulas based on the fixed formula \( \Psi \), the notion of an extended closure and satisfaction of formulas.

**Definition 4.8.** Given a formula \( \Psi \), the set of maximal consistent subsets of \( \text{ecl}(\Psi) \) is denoted \( \Gamma \).

The soundness of these rules are argued for exactly as for the case of ATL ([40]), except the (S) axiom which is slightly different from the super-additivity axiom we know from ATL. We show the soundness of this axiom.

**Proposition 4.9.** Let \( S \) be an arbitrary RCGS, and \( q \) an arbitrary state in \( S \). Whenever two constellations \( \sigma, \sigma' \leq \Sigma \) satisfy \( \sigma + \sigma' \leq \Sigma \), we have \( \vdash \langle \langle \sigma \rangle \rangle \cup \langle \langle \sigma' \rangle \rangle \cup \phi_2 \rightarrow \langle \langle \sigma + \sigma' \rangle \rangle \circ (\phi_1 \land \phi_2) \).

**Proof.** Suppose \( \sigma + \sigma' \leq \Sigma \) and for an arbitrary RCGS and state \( S, q \), we have \( \vdash \langle \langle \sigma \rangle \rangle \cup \langle \langle \sigma' \rangle \rangle \cup \phi_2 \). Each conjunct is satisfied, so let \( S_\sigma \) be a strategy for constellation \( \sigma \) such that, for each complete profile extending this profile \( v \in \text{ext}(q, S_\sigma) \), we have \( S, \delta(q, v) \models \phi_1 \).

Analogously, we have that there is a strategy \( S_{\sigma'} \) such that, for every \( v' \in \text{ext}(q, S_{\sigma'}) \), we have \( S, \delta(q, v') \models \phi_2 \). By our assumption \( \sigma + \sigma' \leq \Sigma \), so let \( S_{\sigma + \sigma'} \) be the strategy which in any given state \( q \), assigns the sum of actions assigned by \( S_\sigma \) and \( S_{\sigma'} \), i.e., \( S_{\sigma + \sigma'}(q) = S_\sigma(q) + S_{\sigma'}(q) \). \( \sigma + \sigma' \) is a constellation, and \( S_{\sigma + \sigma'} \) is a strategy. Notice that, for every state \( q \), \( \text{ext}(q, S_{\sigma + \sigma'}) \subseteq \text{ext}(q, S_\sigma) \) and \( \text{ext}(q, S_{\sigma + \sigma'}) \subseteq \text{ext}(q, S_{\sigma'}) \), hence every \( v \in \text{ext}(q, S_{\sigma + \sigma'}) \) satisfies \( S, \delta(q, v) \models \phi_1 \land \phi_2 \).

The results from ATL largely carry over to when we are speaking about HATL. We will make explicit reference to one derived rule and two validities.

**Proposition 4.10.** The following formulas are valid, and the rule is sound.

- **Constellation Monotonicity** If \( 0 \leq \sigma \leq \sigma' \leq \Sigma \) and \( \vdash \langle \langle \sigma \rangle \rangle \cup \phi \), then \( \vdash \langle \langle \sigma' \rangle \rangle \cup \phi \).

  This validity corresponds to the validity commonly referred to as “coalition monotonicity” for ATL.

- **Regularity** \( \langle \langle \sigma \rangle \rangle \cup \phi \rightarrow \neg \langle \langle \Sigma - \sigma \rangle \rangle \cup \neg \phi \).

  \( \langle \langle \sigma \rangle \rangle \cup \phi \rightarrow \neg \langle \langle \Sigma - \sigma \rangle \rangle \cup \neg \phi \).

  From \( (\phi_2 \lor (\phi_1 \land \langle \langle \sigma \rangle \rangle \cup \theta)) \rightarrow \theta \), infer \( \langle \langle \sigma \rangle \rangle \cup \phi \rightarrow \theta \).
4.4.2 Local construction

In this section we show how we can construct a simple labeled tree which will satisfy a set of \( \bigcirc\)-formulas. This construction will be relied upon when we induce a large tree which will be the basis of a RCGS satisfying our fixed formula \( \Psi \).

A tree labeled with sets of formulas, is said to be locally consistent if the labeling satisfies the positive and negative \( \bigcirc\)-formulas. In this definition, we will rely on the fact that we build trees as sequences of complete profiles.

**Definition 4.11** (Local Consistency). A labeled tree \( \langle T, V \rangle \) is said to be locally consistent if, for each non-leaf node \( t \in T \),

- if \( \langle \sigma \rangle \bigcirc \phi \in V(t) \), there is an \( \sigma \)-vote \( v_\sigma \), such for every \( v \in \text{ext}(q,v_\sigma) \), we have \( \phi \in V(t \cdot v) \), and

- if \( \neg \langle \langle \sigma \rangle \rangle \bigcirc \eta \in V(t) \), then for every \( \sigma \)-vote \( v_\sigma \), there is a \( v \in \text{ext}(q,v_\sigma) \), for which \( \neg \eta \in V(t \cdot v) \).

The construction of the locally consistent simple labeled tree for an (appropriate) set of formulas, will depend on the following lemma. The lemma is similar to [40, Lemma 31 (Disjoint Coalition Consistency)], but applies to composable constellations rather than disjoint coalitions.

**Lemma 4.12** (Composable Constellation Consistency). Let \( \{ \langle \langle \sigma_1 \rangle \rangle \bigcirc \phi_1, \ldots, \langle \langle \sigma_k \rangle \rangle \bigcirc \phi_k, \neg \langle \langle \sigma \rangle \rangle \bigcirc \eta \} \) be a consistent set of formulas. If \( \sigma_1 + \cdots + \sigma_k \leq \sigma \leq \Sigma \), then \( \{ \phi_1, \ldots, \phi_k, \neg \eta \} \) is consistent.

**Proof.** Let \( \{ \langle \langle \sigma_1 \rangle \rangle \bigcirc \phi_1, \ldots, \langle \langle \sigma_k \rangle \rangle \bigcirc \phi_k, \neg \langle \langle \sigma \rangle \rangle \bigcirc \eta \} \) be a consistent set. Since \( (\sigma_1 + \cdots + \sigma_k) \leq \sigma \leq \Sigma \), we have, from repeated applications of (S) that \( \langle \langle \sigma_1 \rangle \rangle \bigcirc \phi_1 \wedge \cdots \wedge \langle \langle \sigma_k \rangle \rangle \bigcirc \phi_k \rightarrow \langle \langle \sigma_1 + \cdots + \sigma_k \rangle \rangle \bigcirc (\phi_1 \wedge \cdots \wedge \phi_k) \). Since \( \sigma_1 + \cdots + \sigma_k \leq \sigma \), by constellation monotonicity, \( \langle \langle \sigma \rangle \rangle \bigcirc (\phi_1 \wedge \cdots \wedge \phi_k) \) is consistent.

Assume towards a contradiction that \( \{ \phi_1, \ldots, \phi_k, \neg \eta \} \) is not consistent. Then \( (\phi_1 \wedge \cdots \wedge \phi_k) \rightarrow \eta \) is a tautology. It would follow that \( \langle \langle \sigma \rangle \rangle \bigcirc (\phi_1 \wedge \cdots \wedge \phi_k) \rightarrow \langle \langle \sigma \rangle \rangle \bigcirc \eta \) by \( \langle \langle \sigma \rangle \rangle \bigcirc \)-monotonicity. Hence \( \langle \langle \sigma_1 \rangle \rangle \bigcirc \phi_1 \wedge \cdots \wedge \langle \langle \sigma_k \rangle \rangle \bigcirc \phi_k \rightarrow \langle \langle \sigma \rangle \rangle \bigcirc \eta \). This contradicts the assumption. \( \Box \)

Before we state and prove the claim that we can construct locally consistent trees for appropriate consistent sets of formulas, \( \Phi \), we define some auxiliary notation. Let \( \Phi \) be a finite consistent set of formulas, and \( \Phi^+ \) and \( \Phi^- \) be the sets of positive and negative \( \bigcirc\)-formulas as discussed earlier, such that \( |\Phi^+| + |\Phi^-|^2 \leq |\Psi^+| + (|\Psi^-| + 1)^2 \).

All agents, regardless of role assignment, will have \( |\Psi^+| + (|\Psi^-| + 1)^2 \) actions available, so the vote of a role will have a profile in the form (depending on the argument set of formulas \( \Phi \) satisfying the above mentioned inequality):

\[
v_r = \langle 1, \ldots, |\Phi^+|, (|\Phi^+| + 1) \times (1, \ldots, |\Phi^-|), b_1, \ldots, b_{(|\Psi^+| + (|\Psi^-| + 1)^2) - (|\Phi^+| + |\Phi^-|^2)} \rangle
\]

For each \( r \)-profile, we denote the number of agents which selected a positive \( \bigcirc\)-formula, say \( \langle \langle \sigma_i \rangle \rangle \bigcirc \phi_i \), i.e., the \( i \)-th number in the vector, as \( v^+_i(i) \). The complete profiles \( P \) are amalgamations of such vectors for the roles.
For a positive $\odot$-formula $\langle\langle \sigma_i \rangle\rangle \odot \phi_i$, we denote the constellation which selected for that formula by $v^+(i) = \prod_{r \in R} v^+_r(i)$. Notice that $v^+(i)$ is an $R$-vector of natural numbers, and indeed, $0 \leq v^+(i) \leq \Sigma$.

We define some selection functions for such complete action profiles.

**Definition 4.13 (Positive $\odot$-formula selection).** Let $\Phi$ be a finite set of formulas, for any $v \in P$, let

$$\alpha(v) = \{ \langle\langle \sigma_i \rangle\rangle \odot \phi_i \in \Phi_\odot \mid \sigma_i \leq v^+_i \}$$

and, we also define a particular constellation by summing the constellations which appear in the outermost connectives of the formulas in $\alpha(c)$. The total required constellation for $\alpha(v)$, denoted $\alpha^\Sigma(v)$ and defined as

$$\alpha^\Sigma(v) = \sum \{ \sigma_i \mid \langle\langle \sigma_i \rangle\rangle \odot \phi_i \in \alpha(v) \}$$

We also define a function projecting the consequences of these actions, i.e., the end product of the coordination.

$$\alpha^\rightarrow(v) = \{ \phi \mid \langle\langle \sigma_i \rangle\rangle \odot \phi \in \alpha(v) \}$$

We will also define corresponding selection functions for the negative formulas, but we will consider a small example first to aid the intuition.

**Example 4.14.** Let there be two roles, and three agents in each role $\Sigma = (3, 3)$. Consider a set of formulas $\Phi$ containing at least the two formulas $\langle\langle (1, 2) \rangle\rangle \odot \phi_1$ and $\langle\langle (1, 1) \rangle\rangle \odot \phi_2$.

Consider a complete profile $v$ in which we let one position from role 1 and a constellation consisting of two positions from role 2 vote for the first positive $\odot$-formula, and the rest vote for the second positive $\odot$-formula. We have a complete profile as illustrated in Figure 4.3.

![Figure 4.3: Example of complete profile with six agents in two equally sized roles ($\Sigma = (3, 3)$).](image)

The formulas which will be selected for $v$ are $\alpha(v)$. We can see easily that the necessary constellation for selecting $\langle\langle (1, 2) \rangle\rangle \odot \phi_1$ is satisfied since $(1, 2) \leq v^+_1 = (1, 2)$. Also, $(1, 1) \leq v^+_2$, so $\langle\langle (1, 1) \rangle\rangle \odot \phi_2 \in \alpha(v)$. The total required agents, $\alpha^\Sigma(v)$, is the sum of the constellations occurring in $\alpha(v)$, that is $\alpha^\Sigma(v) = (1, 2) + (1, 1) = (2, 3)$. We are not counting the number of actual contributions, but rather the number required, i.e., the number occurring in the formula of the selected positive $\odot$-formulas.

The consequence $\alpha^\rightarrow(v) = \{ \phi_1, \phi_2 \}$. 


Similarly to the notation introduced for the votes for positive $\odot$-formulas, we will introduce some notation for the negative ones. However, the vote for a negative formula is a pair $(1,1) \leq (i,j) \leq (|\Phi_\odot|,|\Phi_\odot|)$. The number of agents from role $r$ that selected the pair of number $(i,j)$, will be denoted $v_r^-(i,j)$. We will be interested in counting these elements separately, so we define some shorthand.

$$v_r^-(*,j) := \sum_{1 \leq i \leq |\Phi_\odot|} v_r^-(i,j) \quad \text{number}$$

$$v_r^-(i,*) := \sum_{1 \leq j \leq |\Phi_\odot|} v_r^-(i,j) \quad \text{number}$$

$$v^-(i,*) := \prod_{r \in R} v_r^-(i,*) \quad \text{constellation}$$

We now define an auxiliary function $\gamma$ for complete profiles, which is a simple weighted sum over the second component of votes for negative formulas.

$$\gamma(v) := \sum_{r \in R} \sum_{1 \leq j \leq |\Phi_\odot|} jv_r^-(*,j)$$

Notice that $\gamma(v)$ is a number $0 \leq \gamma(v) \leq n|\Phi_\odot|$, where $n$ is the number of agents ($n = \sum_{r \in R} \Sigma_r$).

Now, corresponding to the selection of positive $\odot$-formulas, we define functions $\beta$ and $\beta^{-}$ which selects up to one formula in $\Phi_\odot$ for each complete profile $v$.

$$\beta(v) := \{ -\langle \sigma_i \rangle \odot \eta_i \mid \alpha^\Sigma \leq \sigma_i, (\Sigma - \sigma_i) \leq v^-(i,*), \text{ and } \gamma(v) \mod |\Phi_\odot| = i - 1 \}$$

$$\beta^{-} := \begin{cases} \{ -\eta \} & \text{if } \beta(v) = \{ -\langle \sigma \rangle \odot \eta \} \\ \emptyset & \text{otherwise} \end{cases}$$

**Example 4.15.** Let there be two roles each with three agents. Suppose the set of negative $\odot$-formulas $\Phi$, ordered arbitrarily, consists of exactly three formulas. The 2-nd formula being $\neg \langle \langle \sigma_2 \rangle \rangle \odot \eta_2$, where $\sigma_2 = (1,2)$. The following is a possible complete vote in which no agent selects for a positive $\odot$-formula, nor any for any of the “buffer” actions (if there are any).

$$v = \left< \begin{pmatrix} 0, \ldots, 0, 1, 0, 0, \\ 0, 0, 0, \\ 0, 0, 0, \\ 0, 1, 1, \\ 0, 0, 0, \\ 0, 0, 0, \\ 0, 0, 0 \end{pmatrix}, \right>$$

There is one agent in the first role which selected an action with first coordinate 1, hence $v^-_1(1,*) = 1$. Also one in the second role, $v^-_2(1,*) = 1$. The introduced shorthand then gives us the constellation which selects a contrubution towards the first negative $\odot$-formula in $\Phi_\odot$, $v^-_1(1,*) = (v^-_1(1,*), v^-_1(1,*)) = (1,1)$. Similarly $v^-_2(2,*) = (2,1)$ and $v^-_3(3,*) = (0,1)$. 
The number $\gamma(v)$ is a weighted sum of the contributions in the second coordinate. For role 1, the sum is $1v_1^\ast(\ast,1)+2v_1^\ast(\ast,2)+3v_1^\ast(\ast,3) = 1+4+0 = 5$. And the second role, $1v_2^\ast(\ast,1)+2v_2^\ast(\ast,2)+3v_2^\ast(\ast,3) = 0+2+6$. So $\gamma(v) = 13$.

How about the set $\beta(v)$? It contains at most one element, and the only candidate for inclusion is the $i$-th element of $\Phi_\circ$, where $i = (\gamma(v) \mod |\Phi_\circ|)+1$. We have $(\gamma(v) \mod |\Phi_\circ|) = (13 \mod 3) = 1$, so the candidate formula is the formula indexed 2, i.e., $\neg\langle\langle \sigma_2 \rangle\rangle \bigcirc \eta_2$. For this formula to be included, two more conditions need to be satisfied.

- $(0,0) = \alpha^\Sigma \leq \sigma_2 = (1,2)$ which in this example where no agent selected a positive formula is trivial to verify, and
- a sufficient number of agents have selected for this formula, $(2,1) = v^-(2,\ast) \geq (\Sigma - \sigma_2) = (2,1)$.

So, indeed, the set $\beta(v)$ contains a single formula $\beta(v) = \{-\langle\langle \sigma_2 \rangle\rangle \bigcirc \eta_2\}$, and the consequence of this $\beta^{-}(v) = \{-\eta_2\}$.

**Lemma 4.16.** For every consistent set of formulas $\Phi$ with $|\Phi_\bullet| + |\Phi_\circ|^2 \leq |\Psi_\circ| + (|\Psi_\circ| + 1)^2$, there is a locally consistent simple labeled tree $\langle T, V \rangle$ with the root labeled $\Phi$ and branching degree $|P|$, where $P$ is as described earlier.

**Proof.** We assume that $\Phi$ contains only formulas in normal form. We also assume it contains no formulas of the form $\neg\langle\langle \Sigma \rangle\rangle \bigcirc \eta$. If it does, replace it with the equivalent formula $\langle\langle \emptyset \rangle\rangle \bigcirc \neg\eta$. Also, we will assume that there is at least one positive $\bigcirc$-formula. If it does not we add an instance of the $(\top)$ axiom $\langle\langle \emptyset \rangle\rangle \bigcirc \top$.

Let $\langle T, V \rangle$ be a simple labeled tree $\langle T = \{\varepsilon\} \cup P, V \rangle$ and let $V(\varepsilon) = \Phi$ and $V(v) = \alpha^{-}(v) \cup \beta^{-}(v)$ for every $v \in P$.

**Consistency** The label of the root node $\Phi$ is consistent by assumption. For every $v \in P$, $v$ is labeled $\alpha^{-}(v) \cup \beta^{-}(v)$. To see that $\alpha^{-}(v)$ is consistent, it is sufficient to show that $\alpha^\Sigma \leq \Sigma$. Every formula $\langle\langle \sigma_i \rangle\rangle \bigcirc \phi_i \in \alpha(v)$ was included since at least $\sigma_i$ agents used one action to fulfill the condition (an appropriate number of agents for each role). Every agent can only choose one action, and only in its role. Hence the total number of agents needed to select the selected formulas is less than all agents.

If $\beta^{-}(v) \neq \emptyset$ (otherwise we are done), then $\beta(v) = \{-\langle\langle \sigma_i \rangle\rangle \bigcirc \eta_i\}$ and $\alpha^\Sigma \leq \sigma_i$. By our assumption that $\Phi$ is consistent, we have that $\alpha(v) \cup \beta(v) \subseteq \Phi$ is consistent. Since $\alpha^\Sigma(v) \leq \sigma$, $\alpha^{-}(v) \cup \beta^{-}(v)$ is consistent by Lemma 4.12.

**Positive $\bigcirc$ satisfaction** Let $\langle\langle \sigma_i \rangle\rangle \bigcirc \phi \in \Phi_\bullet$ be an arbitrary positive $\bigcirc$-formula. The strategy $S_\sigma$ where every agent votes for $\phi$ will ensure that $\phi \in \alpha^{-}(v)$ for every $v \in \text{ext}(\varepsilon, S_\sigma)$.

**Negative $\bigcirc$ satisfaction** Let $\neg\langle\langle \sigma_i \rangle\rangle \bigcirc \eta_i \in \Phi_\circ$ be an arbitrary formula and $v_{\sigma_i}$ be an arbitrary partial profile for $\sigma_i$. We will show that there is a $v \in \text{ext}(\varepsilon, v_{\sigma_i})$ such that $\neg\eta_i \in V(v)$.

Consider $v_{\sigma_i}$, and let $x = \gamma(v_{\sigma_i}) \mod |\Phi_\circ|$. We know that $\Sigma - \sigma_i \neq 0$. Let a position $\sigma^*$ in $\Sigma - \sigma_i$ select $(i, i - x + 1)$. Let all the remaining positions in $\Sigma -$
σ_i - σ*, if there are any, select (i, |Φ_0|). This guarantees that \( \Sigma - \sigma_i \leq v^-(i,*) \)
and that \( \gamma(v) = i - 1 \). Since at most the positions in \( \sigma_i \) participated towards \( v_{\sigma_i}, \)
\( \alpha^\Sigma(v_{\sigma_i}) \leq \sigma_i \). This completion, which is in \( ext(\epsilon, v_{\sigma_i}) \), has \( \beta^\gamma(v) = \{ -\eta \} \).

We conclude that \( \langle T, V \rangle \) is a locally consistent tree with the root labeled \( \Phi \) and with the correct branching degree.

\[ \square \]

### 4.4.3 Eventuality Realization

In the eventuality realization section of the proof in [40, Section 4.4], the authors show that the temporal operators other than the \( \bigcirc \)-formulas, i.e., formulas containing \( \langle\langle \sigma \rangle\rangle \Box \phi \) or \( \langle\langle \sigma \rangle\rangle \phi_1 \cup \phi_2 \), can also be satisfied by constructing trees, as detailed in the previous section, by amalgamating such locally consistent simple labeled trees.

We show one case (corresponding to [40, Lemma 36]), that of the existence of a finite locally consistent tree which realizes an eventuality of the form \( \langle\langle \sigma \rangle\rangle \phi_1 \cup \phi_2 \). The proof closely resembles that of [40] being different only in the size of the locally consistent trees constructed as specified in Lemma 4.16 (corresponding to [40, Lemma 33]).

**Definition 4.17** (Relization). An eventuality of the form \( \langle\langle \sigma \rangle\rangle \phi_1 \cup \phi_2 \) is said to be realized from a node \( t \) of the labeled tree \( \langle T, V \rangle \) with maximal consistent subsets of \( \text{ecl}(\Psi) \) (elements of \( \Gamma \)) when there exists an \( \sigma \)-strategy \( s_{\sigma} \) such that for all \( \lambda \in \text{out}(t, s_{\sigma}) \), there is some \( i \) such that \( \phi_2 \in V(\lambda[i]) \) and for all \( 0 \leq j < i, \phi_1 \in V(\lambda[j]) \) (c.f., [40, Definition 34]).

We omit the other cases (formulas of the form \( \neg\langle\langle \sigma \rangle\rangle \Box \phi \), \( \langle\langle \sigma \rangle\rangle \Box \phi \) and \( \neg\langle\langle \sigma \rangle\rangle \phi_1 \cup \phi_2 \)) and refer to [40] for details. As \( \text{ecl}(\Psi) \) is closed under the boolean connectives, there is a characteristic formula \( \chi_Z \) (up to propositional equivalence) for every subset \( Z \subseteq \Gamma \), such that for every \( z \in \Gamma, \chi_Z \in z \) if and only if \( z \in Z \). (See [40, Lemma 35] for details.)

We now state and show the eventuality realization argument for eventualities of the form \( \langle\langle \sigma \rangle\rangle \phi_1 \cup \phi_2 \) in Lemma 4.20, but unwrap it in Definition 4.18 and Proposition 4.19.

**Definition 4.18** (Satisfying conditions for \( \langle\langle \sigma \rangle\rangle \phi_1 \cup \phi_2 \)). Fix a formula \( \tau = \langle\langle \sigma \rangle\rangle \phi_1 \cup \phi_2 \in \text{ecl}(\Psi) \). For any maximal consistent set \( x \in \Gamma \), we say that the conditions are satisfied for \( \tau \) in \( x \in \Gamma \) by \( \langle T, V \rangle \) if, and only if,

- \( \langle T, V \rangle \) is a finite labeled tree over \( \Gamma \).
- \( \langle T, V \rangle \) is of fixed branching degree \( |P| \), (as described earlier).
- \( \langle T, V \rangle \) is locally consistent,
- \( V(\epsilon) = x, \) and
- \( \text{if } \tau \in x, \text{ then } \tau \text{ is realized from } \epsilon. \)

For a given eventuality \( \tau \) on the form \( \langle\langle \sigma \rangle\rangle \phi_1 \cup \phi_2 \), for any \( x \in \Gamma \), if \( \langle T, V \rangle \) is a tree satisfying the above conditions, we write \( R(\tau, x, \langle T, V \rangle) \).

The set of maximal consistent sets in \( \Gamma \) which satisfies these conditions are gathered in the set \( Z_\tau \).

\[
Z_\tau := \{ x \in \Gamma \mid \text{there is a } \langle T, V \rangle \text{ s.t. } R(\tau, x, \langle T, V \rangle) \}
\]
We will use this definition to show that certain formulas are tautologies. We will rely on these tautologies in the eventuality realization argument.

**Proposition 4.19.** Given a formula of the form $\tau = \langle\langle \sigma \rangle \rangle \phi_1 \cup \phi_2 \in \text{ecl}(\Psi)$, let $\chi_{Z_\tau}$ be the characteristic formula of the set of maximal consistent sets of formulas from $\text{ecl}(\Psi)$ satisfying the above definition. The following propositions are tautologies

1. $\phi_2 \rightarrow \chi_{Z_\tau}$,
2. $(\phi_1 \land \langle\langle \sigma \rangle \rangle \phi) \rightarrow \chi_{Z_\tau}$, and
3. $\langle\langle \sigma \rangle \rangle \phi_1 \cup \phi_2 \rightarrow \chi_{Z_\tau}$.

**Proof.** In 1. and 2. cases, we show that every maximal consistent set (not subsets of $\text{ecl}(\Psi)$) contain the formula.

1. We show that $\phi_2 \rightarrow \chi_{Z_\tau}$ is a member of every maximal consistent set. Let $r$ be an arbitrary maximal consistent set. If $\phi_2 \notin r$, then $\phi_2 \rightarrow \chi_{Z_\tau} \in r$. Otherwise, if $\phi_2 \in r$, construct a tree $\langle T, V \rangle$ as specified in Lemma 4.16 with the root labeled $V(\varepsilon) = r \cap \text{ecl}(\Psi)$. Since $\phi_2 \in \text{ecl}(\Psi)$, we have that $\tau$ is realized immediately in the root of the constructed tree. The tree is finite, has fixed branching degree $|P|$, is locally consistent, the root is labeled $r \cap \text{ecl}(\Psi) \in \Gamma$ and realizes $\tau$ from $\varepsilon$ (hence, if $\langle\langle \sigma \rangle \rangle \phi_1 \cup \phi_2 \in r \cap \text{ecl}(\Psi)$, then the last condition in Definition 4.18 is also satisfied). The constructed tree satisfies Definition 4.18, and hence $R(\tau, r \cap \text{ecl}(\Psi), \langle T, V \rangle)$, or equivalently $\chi_{Z_\tau} \in r \cap \text{ecl}(\Psi)$, and particularly $\chi_{Z_\tau} \in r$. As $r$ is a maximal consistent set containing $\chi_{Z_\tau}$, it also contains $\phi_2 \rightarrow \chi_{Z_\tau}$.

2. Let $r$ be an arbitrary maximal consistent set. If $\phi_1 \land \langle\langle \sigma \rangle \rangle \phi \notin r$, we are done. So suppose that $\phi_1 \land \langle\langle \sigma \rangle \rangle \phi \in r$. Let $\Psi' = \{ \theta \in r \cap \text{ecl}(\Psi) \mid \theta = \langle\langle \sigma \rangle \rangle \phi \text{ or } \theta = -\langle\langle \sigma \rangle \rangle \eta \}$ or $\{ \langle\langle \sigma \rangle \rangle \phi \}$. Even though $\chi_{Z_\tau} \in \text{ecl}(\Psi)$, we do not necessarily have $\langle\langle \sigma \rangle \rangle \phi \in \text{ecl}(\Psi)$. Notice also that $|\Psi'_\phi| + |\Psi'_\eta| \leq |\Psi_\phi| + (|\Psi_\eta| + 1)^2$. Let $\langle T', V' \rangle$ be the tree constructed according to Lemma 4.16 where the root is labeled $V'(\varepsilon) = \Psi'$.

Construct a new tree $\langle T, V \rangle$ where the root is labeled $V(\varepsilon) = r \cap \text{ecl}(\Psi)$. Notice that $r \cap \text{ecl}(\Psi) \in \Gamma$. In this tree, we let the agents have the same number of actions available as we did in the local construction, hence the set of complete votes can be seen to be $P$. For each complete profile $v \in P$, label the successor arbitrarily such that $V(v) \in \{ x \in \Gamma \mid V'(v) \cap \text{ecl}(\Psi) \subseteq x \}$. That is, the successor state yielded by the complete profile $v$ is labeled by a maximal consistent set $x \in \Gamma$ such that it contains the labels of the corresponding successor state in the tree $\langle T', V' \rangle$.

Since $\chi_{Z_\tau} \in \text{ecl}(\Psi)$, all the successors $v$ in $\langle T', V' \rangle$ which contain $\chi_{Z_\tau}$ also contain $\chi_{Z_\tau}$ in $\langle T, V \rangle$, and therefore the constellation $\sigma$ has a strategy to ensure that the successor state is labeled $\chi_{Z_\tau}$ in both trees; i.e., in $\langle T', V' \rangle$ by construction, and in $\langle T, V \rangle$ as it is isomorphic under $\text{ecl}(\Psi)$ to $\langle T', V' \rangle$.

In $\langle T, V \rangle$, consider every complete profile $v$ which is labeled $\chi_{Z_\tau} \in V(v)$. The label $V(v) \in \Gamma$ contains $\chi_{Z_\tau}$, so there is a tree $\langle T_v, V_v \rangle$ such that $R(\tau, V(v), \langle T_v, V_v \rangle)$ by Definition 4.18 and [40, Lemma 35 ("Characteristic Formula"). Hence, $\langle T_v, V_v \rangle$ is a locally consistent finite labeled tree over $\Gamma$ with fixed branching
Axiomatic Reasoning

degree \(|P|\) labeled \(V_\nu(\varepsilon) = V(v)\) which, if \(\tau \in V(v)\), realizes \(\tau\) from \(\varepsilon\). Replace that node \(v\) (an arbitrary node labeled \(\chi_{Z_\tau} \in V(v)\)) with the tree \(\langle T_v, V_v \rangle\).

To see that the tree \(\langle T, V \rangle\) constructed this way realizes \(\tau\), observe that the tree \(\langle T', V' \rangle\) constructed according to Lemma 4.16 is locally consistent and the root of this tree was labeled such that it contained \(\langle \langle \sigma \rangle \rangle \cup \chi_{Z_\tau} \in V'(\varepsilon)\). Hence there is a partial profile \(v_\sigma\) for the constellation \(\sigma\), such that every extension yields a successor state labeled with \(\chi_{Z_\tau}\). Every successor state in \(\langle T, V \rangle\) in \(\text{ext}(v_\sigma)\) is also labeled \(\chi_{Z_\tau}\). And we know that these states realize \(\tau\).

As we have shown that there is a locally consistent labeled tree, \(\langle T, V \rangle\), over \(\Gamma\) which is labeled \(r \cap \text{ecl}(\Psi)\) and realizes \(\tau\) at \(\varepsilon\), we conclude that \(r \cap \text{ecl}(\Psi) \in Z_\tau\), and hence \(\chi_{Z_\tau} \in r\). Since \(r\) is a maximal consistent set containing \(\chi_{Z_\tau}\), it must also contain \((\phi_1 \cup \langle \langle \sigma \rangle \rangle \cup \chi_{Z_\tau}) \rightarrow \chi_{Z_\tau}\).

We conclude that \((\phi_1 \cup \langle \langle \sigma \rangle \rangle \cup \chi_{Z_\tau}) \rightarrow \chi_{Z_\tau}\) is contained in every maximal consistent set (not restricted to \(\text{ecl}(\Psi)\)), and that it is tautological.

![Figure 4.4: Constructing and composing trees in proof of Lemma 4.19](image)

3. From points 1. and 2. and \(\langle \langle \sigma \rangle \rangle \cup\)-induction.

\[
\begin{align*}
\vdash \phi_2 & \rightarrow \chi_{Z_\tau} & \vdash \phi_1 \cup \langle \langle \sigma \rangle \rangle \cup \chi_{Z_\tau} & \rightarrow \chi_{Z_\tau}\\
\vdash \phi_2 \lor (\phi_1 \cup \langle \langle \sigma \rangle \rangle \cup \chi_{Z_\tau}) & \rightarrow \chi_{Z_\tau} & \vdash \langle \langle \sigma \rangle \rangle \cup \phi_2 & \rightarrow \chi_{Z_\tau} \langle \langle \sigma \rangle \rangle \cup\)-ind
\end{align*}
\]

With these tautologies in place, we are ready to show the main claim of this section.

**Lemma 4.20** (Eventuality Realization 1). *(C.f., [40, Lemma 36]) For every eventuality \(\tau = \langle \langle \sigma \rangle \rangle \cup \phi_2 \in \text{ecl}(\Psi)\), and every maximal consistent subset of \(\text{ecl}(\Psi)\), \(x \in \Gamma\), there is a tree \(\langle T, V \rangle\) such that \(R(\tau, x, \langle T, V \rangle)\).*
4.4 Completeness of HATL

Proof. Let $\tau = \langle\langle \sigma \rangle\rangle \phi_1 \cup \phi_2 \in \text{ecl}(\Psi)$ be an arbitrary eventualty of this form, and $x \in \Gamma$ be an arbitrary maximal consistent subset of $\text{ecl}(\Psi)$. We show that there is a tree $\langle T, V \rangle$ such that $R(\tau, x, \langle T, V \rangle)$. We proceed by two cases:

$\tau \notin x$ : Let $\langle T, V \rangle$ be the tree constructed according to Proposition 4.16. This tree satisfies the conditions of Definition 4.18, the last condition is satisfied trivially since $\tau \notin x$. We conclude that for this tree, $R(\tau, x, \langle T, V \rangle)$.

$\tau \in x$ : From Proposition 4.19 we know that $\langle\langle \sigma \rangle\rangle \phi_1 \cup \phi_2 \rightarrow \chi_Z\tau$ is a tautology. Since both formulas $\tau$ and $\chi_Z\tau$ are in $\text{ecl}(\Psi)$, it follows from $\langle\langle \sigma \rangle\rangle \phi_1 \cup \phi_2 = \tau \in x$, that $\chi_Z\tau \in x$. In other words $x \in Z_{\tau}$. Hence there exists a tree $\langle T, V \rangle$ such that $R(\tau, x, \langle T, V \rangle)$.

$\Box$

In [40], there are corresponding proofs for the existence of finite trees ([40, Lemma 37]) realizing ([40, Definition 34]) eventualties of the form $\neg \langle\langle \sigma \rangle\rangle \phi$. A change in the branching degree corresponding to the change we did in the previous lemma needs to be implemented also when translating the result from ATL to HATL. Otherwise, the proof is the same. In that proof, we would add a negative $\bigcirc$-formula which is not (necessarily) in $\text{ecl}(\Psi)$. Since the number of actions corresponding to a vote for a negative $\bigcirc$-formula is squared, the inequality permits us to add an additional negative formula not (necessarily) in $\Psi^\bigcirc$ or $\Psi^\bigcirc$.

4.4.4 Final model construction

In this section, we present the argument from Section 4.5 of [40] where they construct a final model from the previous definitions and lemmas. This will yield a model which is shown to satisfy the formula $\Psi$ which we assumed to be a consistent formula in the beginning of this chapter. The model $S^\Psi$ will be presented later, in Definition 4.23.

As presented in the previous section, there is for every pair of eventuality $\tau$ and maximal consistent subset of $\text{ecl}(\Psi)$, $x \in \Gamma$, a finite locally consistent labeled tree $\langle T_{x, \tau}, V_{x, \tau} \rangle$ which realizes the eventuality. These trees are referred to as final model tree components. Such trees all have a fixed branching degree of $|P|$, they are locally consistent and finite (c.f., [40, Definition 38 and Lemma 39]).

We now show how to construct what Goranko & van Drimmelen refer to as final labeled trees.

Definition 4.21 (Final labeled tree). For the consistent formula $\Psi$, enumerate all eventualties in $\text{ecl}(\Psi)$ as $\tau_0, \tau_1, \ldots, \tau_{m-1}$. We define an infinite tree $\langle T_i, V_i \rangle$ inductively. The final labeled tree $\langle T_{\Psi}, V_{\Psi} \rangle$ is the limit of this definition.

Base case. Let $\langle T_0, V_0 \rangle = \langle T_{x, \tau_0}, V_{x, \tau_0} \rangle$ where $x \in \Gamma$ is an arbitrary maximal consistent subset of $\text{ecl}(\Psi)$ which contains $\Psi$ (i.e., $\Psi \in x$). That is, the tree in the base case of the definition is a final model tree component $\langle T_{x, \tau_0}, V_{x, \tau_0} \rangle$ where the label of the root contains $\Psi \in V_{x, \tau_0}(\epsilon)$ and which realizes eventuality $\tau_0$.

Induction step. For every leaf node $v$ in $\langle T_i, V_i \rangle$, associate that leaf with the final model tree component $\langle T_{V_i(v), \tau_{(i+1) \mod m}}, V_{V_i(v), \tau_{(i+1) \mod m}} \rangle$. 
Before we state and prove the Truth Lemma, we state some lemmas that will be useful in the proof of the Truth Lemma.

**Proposition 4.22.** The following claims are shown to be correct in [40]. The definition for a formula being realized, except the one stated in Definition 4.17 are analogous to the stated one and can be found in [40, Definition 34].

- If an eventuality of the form $\langle\langle\sigma\rangle\rangle\phi_1\cup\phi_2$ or $\langle\langle\sigma\rangle\rangle\Box\phi$ is in the label of a node $t$ in the final labeled tree $\langle T_\Psi, V_\Psi \rangle$, then that eventuality is realized from $t$. (See [40, Lemma 41] for proof.)

- If a formula of the form $\langle\langle\sigma\rangle\rangle\Box\phi$ or $\neg\langle\langle\phi\rangle\rangle_1\cup\phi_2$ is in the label of a node $t$ in the final labeled tree $\langle T_\Psi, V_\Psi \rangle$, then that formula is realized from $t$. (See [40, Lemma 42] for proof.)

The final model for a consistent formula $\Psi$ is given in terms of the final labeled tree.

**Definition 4.23 (RCGS for $\Psi$).** Given a consistent formula $\Psi$ and the final labeled tree $\langle T_\Psi, V_\Psi \rangle$, define $\text{RCGS} S_\Psi = \langle A, R, \rho, T_\Psi, \Pi, \pi, A, \delta \rangle$ where

- $T_\Psi$ is the set of states,
- $A, R, \rho,$ and $\Pi$ are given as parameters,
- $\pi(t) = V_\Psi(t) \cap \Pi$,
- $A(t,r) = |\Psi_\oplus| + (|\Psi_\ominus| + 1)^2$, and
- $\delta(t,v) = t \cdot v$.

The set of states in $S_\Psi$ contains a root, $\varepsilon$. Every state offer exactly the same actions to the agents. So, the set of complete action profiles $P$ is not dependent on the state. The successor of any state is uniquely determined (through $\delta$) by a state and a complete action profile $v \in P$. States are simply sequences of complete profiles (including the empty sequence).

**Lemma 4.24 (Truth).** Let $S_\Psi = \langle A, R, \rho, T_\Psi, \Pi, \pi, A, \delta \rangle$ be the final model for the consistent formula $\Psi$. For every state $t \in T_\Psi$, and every formula $\phi \in \text{ecl}(\Psi)$, if $\phi \in V(t)$ then $S_\Psi \models \phi$.

**Proof.** By induction on the complexity of $\phi$. (The proof is analogous to [40, Lemma 43].)

- If $\phi = p$ or any propositional symbol in $\Pi$, as $p \in V_\Psi(t)$, we have $p \in \pi(t)$ and $S_\Psi, t \models p$.
- If $\phi = \neg p$ or the negation of any propositional symbol in $\Pi$, then since $V_\Psi(t)$ is a maximal consistent subset of $\text{ecl}(\Psi)$, we have $p \notin V_\Psi(t)$, and hence $p \notin \pi(t)$, so $S_\Psi, t \models \neg p$.
- If $\phi = \phi_1 \lor \phi_2$, respectively $\phi = \phi_1 \land \phi_2$, then since $V_\Psi(t)$ is a maximal consistent subset either $\phi_1$ or $\phi_2$ is in $V_\Psi(t)$, or respectively both $\phi_1$ and $\phi_2$ are in $V_\Psi(t)$. The claim follows by the induction hypothesis.
4.5 The logic RATL

- If $\phi = \langle\langle \sigma \rangle\rangle \circ \phi'$. By the construction of $\langle T_\Psi, V_\Psi \rangle$, is locally consistent so there is a partial profile $v_\sigma$ for $\sigma$, such that all complete profiles extending it $v \in \text{ext}(v_\sigma, t)$, is labeled $\phi' \in V_\Psi(t \cdot v)$. Hence, by the induction hypothesis for every complete profile $v$ in $S_\Psi$ in $\text{ext}(v_\sigma, t)$, $S_\Psi(t \cdot v) \models \phi'$. The partial profile $s_\sigma$ entails that, by the definition of $\delta$, at $t$ the successor is a state $t \cdot v$ where $v \in \text{ext}(v_\sigma, t)$, so $S_\Psi, t \models \langle\langle \sigma \rangle\rangle \circ \phi'$.

- Suppose $\phi = \neg\langle\langle \sigma \rangle\rangle \circ \eta$. The tree is locally consistent, so when $\neg\langle\langle \sigma \rangle\rangle \circ \eta$ then for every $\sigma$-profile $v_\sigma \in P(t, \sigma)$, there is a complete profile extending it $v \in \text{ext}(t, v_\sigma)$ such that $\neg \eta \in V_\Psi(t \cdot v)$. By the induction hypothesis, every successor state corresponding to such a child (in the tree) satisfies $S_\Psi, t \cdot v \models \neg \eta$. Since the (partial) profiles in the model correspond to those in the tree, we have $S_\Psi, t \models \neg\langle\langle \sigma \rangle\rangle \circ \eta$.

- If the formula is on the form $\langle\langle \sigma \rangle\rangle \phi_1 U \phi_2$, $\neg\langle\langle \sigma \rangle\rangle \Box \phi'$, $\neg\langle\langle \sigma \rangle\rangle \phi_1 U \phi_2$, or $\langle\langle \sigma \rangle\rangle \Box \phi'$, the formula is realized in $\langle T_\Psi, V_\Psi \rangle$, and by application of the induction hypothesis are satisfied in the model.

Since $\Psi \in V_\Psi(\varepsilon)$, we have $S_\Psi, \varepsilon \models \Psi$. This entails the final statement in the completeness argument.

**Theorem 4.25.** For any consistent HATL formula $\Psi$, there is a pointed model satisfying it, $S_\Psi, \varepsilon \models \Psi$.

### 4.5 The logic RATL

With this logic in place, we define a new logic, RATL (ATL with roles), which we show is equivalent to HATL. We keep the traditional ATL language, hence reinstating the agents in the language (and semantics), but add an axiom to express our homogeneity constraint.

The language $\mathcal{L}_{\text{RATL}}$ is similar to that of traditional ATL, but we have the additional parameters $R$ and $\rho$ (which are not visible in the syntax).

**Definition 4.26.** The language of $\mathcal{L}_{\text{RATL}}$, for agents $A$ distributed independently of state into $R \geq 1$ roles by $\rho$ over propositional symbols $\Pi$, is given by the following BFN

$$
\phi ::= p \mid \neg \phi \mid \phi \land \phi \mid \langle\langle A \rangle\rangle \circ \phi \mid \langle\langle A \rangle\rangle \Box \phi \mid \langle\langle A \rangle\rangle \phi \cup \phi
$$

where $\rho \in \Pi$ and $A \subseteq A$.

The role assignment $\rho$ is state independent. This gives rise to a new validity.

**Proposition 4.27** (Validity of the (Hom) axiom). Given an arbitrary RCGS $M = \langle A, R, \rho, Q, \Pi, \pi, A, \delta \rangle$, and any state $q \in Q$, and any two coalitions $A$ and $B$ such that $\sigma_A = \sigma_B$, for any formula $\phi$, the following formula is true in that state

$$
M, q \models \langle\langle A \rangle\rangle \circ \phi \rightarrow \langle\langle B \rangle\rangle \circ \phi.$$

Proof. Recall that $\sigma_A$ is the constellation coalition $A$ instantiates. So coalitions $A$ and $B$ instantiate the same constellation. Since $\rho$ is state independent, this is true in every state of every model. The claim follows immediately from the observation that $A$ and $B$ have exactly the same (partial) profiles ($P(q, A) = P(q, B)$), for any such (partial) profile, their extensions are the same. \hfill $\square$

Adding this formula, which we refer to as (Hom), the homogeneity axiom, to the axiomatization of ATL [40], we get the following axiomatization. The validity of the (Hom) axiom is Proposition 4.27, and the validity of the remaining axioms, as well as the soundness of the inference rules, are argued for analogously as for ATL.

The axiomatization of RATL is given in Tables 4.5 and 4.6.

\begin{itemize}
\item[(TAUT)] Propositional tautologies
\item[\perp] $\neg\langle\langle A \rangle\rangle \circ \perp$
\item[\top] $\langle\langle A \rangle\rangle \circ \top$
\item[(A)] $\neg\langle\langle \emptyset \rangle\rangle \circ \neg \phi \rightarrow \langle\langle A \rangle\rangle \phi$
\item[(S)] $\langle\langle A \rangle\rangle \circ \phi_1 \land \langle\langle B \rangle\rangle \circ \phi_2 \rightarrow \langle\langle A \cup B \rangle\rangle \circ (\phi_1 \land \phi_2)$ \hspace{1cm} whenever $A \cap B = \emptyset$
\item[(Hom)] $\langle\langle A \rangle\rangle \circ \phi \rightarrow \langle\langle B \rangle\rangle \circ \phi$ \hspace{1cm} whenever $\sigma_A = \sigma_B$
\item[(FP)\perp] $\langle\langle A \rangle\rangle \Box \phi \leftrightarrow \phi \land \langle\langle A \rangle\rangle \Box \phi$
\item[(GFP)\perp] $\langle\langle \emptyset \rangle\rangle \Box (\theta \rightarrow (\phi \land \langle\langle A \rangle\rangle \Box \theta)) \rightarrow \langle\langle \emptyset \rangle\rangle \Box (\theta \rightarrow \langle\langle A \rangle\rangle \Box \phi)$
\item[(FP)U] $\langle\langle A \rangle\rangle \phi_1 \cup \phi_2 \leftrightarrow \phi_2 \lor (\phi_1 \land \langle\langle A \rangle\rangle \circ \langle\langle A \rangle\rangle \phi_1 \cup \phi_2)$
\item[(LFP)U] $\langle\langle \emptyset \rangle\rangle (\phi_2 \lor (\phi_1 \land \langle\langle A \rangle\rangle \Box \theta) \rightarrow \theta) \rightarrow \langle\langle \emptyset \rangle\rangle \Box (\langle\langle A \rangle\rangle \phi_1 \cup \phi_2 \rightarrow \theta)$
\end{itemize}

Table 4.5: Axioms of RATL for every $A, B \subseteq A$.

\begin{itemize}
\item[(MP)] $\phi_1 ; \phi_1 \rightarrow \phi_2 \rightarrow \phi_2$
\item[(\perp)-monotonicity] $\langle\langle A \rangle\rangle \circ \phi_1 \rightarrow \langle\langle A \rangle\rangle \circ \phi_1 \rightarrow \langle\langle A \rangle\rangle \circ \phi_2$
\item[(\perp)-necessitation] $\phi \rightarrow \langle\langle A \rangle\rangle \Box \phi$
\end{itemize}

Table 4.6: Inference rules for RATL for every $A \subseteq A$.

### 4.5.1 Equivalence of axiomatizations – Anonymized ATL

In order to see that RATL is equivalent to HATL, we construct a simple function from the language with coalitions (RATL) to the language with constellations (HATL). The required assumption is that the role assignment is state independent. When it is not, there is no single constellation which describes a coalition, as some of the members may change role through the development of the system. The function, which is clearly surjective, simply replaces a coalition with the constellation it instantiates.
Definition 4.28. We define an anonymizing function $\text{AN} : \mathcal{L}_{\text{RATL}} \rightarrow \mathcal{L}_{\text{HATL}}$ inductively

\[
\begin{align*}
\text{AN}(p) & := p \\
\text{AN}(\neg \phi) & := \neg \text{AN}(\phi) \\
\text{AN}(\phi_1 \land \phi_2) & := \text{AN}(\phi_1) \land \text{AN}(\phi_2) \\
\text{AN}(\langle A \rangle \bigcirc \phi) & := \langle \sigma_A \rangle \bigcirc \text{AN}(\phi) \\
\text{AN}(\langle A \rangle \Box \phi) & := \langle \sigma_A \rangle \Box \text{AN}(\phi) \\
\text{AN}(\langle A \rangle \phi_1 \bigcup \phi_2) & := \langle \sigma_A \rangle \text{AN}(\phi_1) \bigcup \text{AN}(\phi_2)
\end{align*}
\]

Proposition 4.29. For every $\phi \in \mathcal{L}_{\text{RATL}}$, we have

\[
\vdash_{\text{RATL}} \phi \iff \vdash_{\text{HATL}} \text{AN}(\phi)
\]

Proof. We prove this by showing that the validity of the axioms and the soundness of the rules are preserved in both directions by this translation. Most of the axioms and rules are obvious. We show the cases of (S) and (Hom).

$\Rightarrow$ We show that $\vdash_{\text{HATL}} \text{AN}(\text{(Hom)})$ and $\vdash_{\text{HATL}} \text{AN}(\text{(S)})$.

The (Hom) axiom is trivial. Assume $\langle A \rangle \bigcirc \phi \rightarrow \langle B \rangle \bigcirc \phi$ and $\sigma_A = \sigma_B$, then $\text{AN}(\langle A \rangle \bigcirc \phi \rightarrow \langle B \rangle \bigcirc \phi) = \langle \sigma_A \rangle \bigcirc \text{AN}(\phi) \rightarrow \langle \sigma_B \rangle \bigcirc \text{AN}(\phi)$. Since $\sigma_A = \sigma_B$ this is just a propositional tautology.

For the (S) axiom, let $A, B \subseteq A$ with $A \cap B = \emptyset$. It follows that $\sigma_A, \sigma_B \leq \Sigma$ and that $\sigma_A + \sigma_B \leq \Sigma$. Hence, the translation of the (S) axiom in RATL is an instance of the (S) axiom in HATL.

$\Leftarrow$ We show a stronger claim. Since $\text{AN}(\underline{\text{-}})$ is surjective, we can show that for every formula $\phi' \in \mathcal{L}_{\text{HATL}}$ there is a formula $\phi \in \mathcal{L}_{\text{RATL}}$ with $\text{AN}(\phi) = \phi'$. We show that if $\vdash_{\text{HATL}} \phi'$, then $\vdash_{\text{RATL}} \phi$. We need to show that if something is derived from the HATL version of (S), then it could be derived directly in RATL. Let $\langle A \rangle \bigcirc \phi_1$ and $\langle B \rangle \bigcirc \phi_2$ such that $\sigma_A + \sigma_B \leq \Sigma$. Let $C$ be any coalition such that $\sigma_C = \sigma_A + \sigma_B$. Since $\sigma_A + \sigma_B \leq \Sigma$ there is a coalition $A' \subseteq A \setminus B$ such that $|A| = |A'|$. Then, in RATL

\[
\begin{align*}
\vdash \langle A \rangle \bigcirc \phi_1 \quad \text{(Hom)} + \text{MP} & \quad \vdash \langle A' \bigcup B \rangle \bigcirc (\phi_1 \land \phi_2) \quad \text{(S)} + \text{MP} \\
\vdash \langle A' \bigcup B \rangle \bigcirc (\phi_1 \land \phi_2) \quad \text{(Hom)} + \text{MP}
\end{align*}
\]

4.6 Relationship between HATL and RATL

We have shown that we have a complete axiomatization of both HATL and RATL. In HATL the atomic acting objects were, in a sense, the positions rather than, as in ATL and RATL, the agents. When we spoke of agents, they were in each case arbitrary
representatives of the roles in the topical state. The satisfaction of the generalized temporal connectives (□, ◊ and U), were all argued for in terms of the fixed-point representation of these, reducing the arguments to statements about the ○-fragment.

Keeping this in mind, we may observe that the identity of the agents enacting the topical roles may vary throughout a run, as a new representative will be “picked out” anew in every state by the ○-formulas we reduce it to. This observation entails that the role assignment ρ does not really have to be state independent in the strictest sense. It suffices that, at every state, the number of agents assigned to each role is constant, i.e., that it keeps a fixed social structure.

The agents are still in the semantics, however, and we can reintroduce them into the logic (RATL) and establish completeness for this logic as well. We can discuss the difference between (coalitions of) agents and (constellations of) positions in a meaningful way. When the role assignment is independent of the state, the notions overlap. However, if we do not require that the role assignment is constant, but rather that the grand constellation in every state is fixed, then coalitions (generally) will have different strategic capabilities than constellations.

If a happens to be a president (role 1), and b happens to be a voter (role 2), then, with state independent role assignment, the coalition \{a, b\} will have the same strategic abilities as the constellation (1, 1). If the role assignment is not state independent, but we require that every role, in every state, has a fixed number of agents occupying it, then the mentioned coalition and constellation will have the same strategic abilities only as long as a remains a president and b remains a voter. A fixed social structure is exactly this case of having a constant grand constellation.

**Example 4.30** (Constellations and coalitions in fixed social structures). Consider the following scenario of a system consisting of four agents \(A = \{a, b, c, d\}\). We will consider agent d a dummy agent. We will have a fixed social structure \(\Sigma = (1, 3)\) for roles \(R = \{\text{pres}, \text{pop}\}\), i.e., for every state \(q\) in the system \(|A_{\text{pres}}| = 1\) and \(|A_{\text{pop}}| = 3\).

![Figure 4.7: RCGS with fixed social structure, but not state independent role assignment.](image)

The outgoing edges from the initial state \(q_i\), indicated with a * label, represent the voting mechanism we described in the Example 3.6 where the agents vote for the agent
to take a privileged position. By majority voting and arbitrary tie-breaking, some agent (other than agent d) is selected. We transition into either $q_a$, $q_b$ or $q_c$ depending on the outcome of the vote.

From $q_a$ (respectively $q_b$ or $q_c$), the president a (respectively b or c) can dictate either R or Q. The outgoing edges are labeled $\langle(1,0),3\rangle$ and $\langle(0,1),3\rangle$ signifying that the agent in role 1 (pres) has two actions available, while the agents in role 2 (pop) has only one action and hence no choice. The outgoing edges from $q_b$ and $q_c$ are dotted to avoid cluttering. They would be labeled identically to the outgoing edges from $q_a$.

Any coalition consisting of more than half of the voters can dictate that the presidential decision. And the constellation consisting of “a president”, but no constellation consisting of only the general population, can also enforce this decision.

Some formulas satisfied in $q_0$:

- A majority of the voters are able to end up with any candidate as president: $\langle\langle(0,2)\rangle\rangle \odot (p_a), \langle\langle(0,2)\rangle\rangle \odot (p_b)$ and $\langle\langle(0,2)\rangle\rangle \odot (p_c)$.

- The president (this is not a concrete agent, but the object referred to by the non-rigid designator “president”) can ensure that R (which is true only in state $q_R$): $\langle\langle(1,0)\rangle\rangle \top \cup R = \langle\langle(1,0)\rangle\rangle \Diamond R$.

- A coalition consisting of two agents (other than d) may elect one of its members as president, and ensure R: $\langle\langle\{a,b\}\rangle\rangle \Diamond R, \langle\langle\{a,c\}\rangle\rangle \Diamond R$, and $\langle\langle\{b,c\}\rangle\rangle \Diamond R$.

- No constellation not including the president can ensure this: $\neg\langle\langle(0,3)\rangle\rangle \Diamond R$.

Finally, observe that the assumption of homogeneous roles in a society gives a kind of “egalitarian guarantee”. Any constellation which is “small enough” to be composed with itself, is not a dictatorial constellation.

**Proposition 4.31.** The following statements are true.

- If $R = 1$ and $\{\langle\sigma\rangle \odot \phi, \langle\sigma\rangle \odot \neg \phi\}$ is consistent, then $\sigma > \Sigma/2$. Or, more generally,

- If $\{\langle\sigma\rangle \odot \phi, \langle\sigma\rangle \odot \neg \phi\}$ is consistent, then $\sigma + \sigma > \Sigma$, i.e., $\sigma$ is not composable with itself.

**Proof.** We show the second point, as the first point is a special case of it. Assume $\langle\sigma\rangle \odot \phi$ and that $\sigma + \sigma \leq \Sigma$. It follows that $\sigma \leq \Sigma - \sigma$, so we may apply the constellation monotonicity rule, and obtain a claim such that an application of regularity and modus ponens yields the desired conclusion.

\[
\frac{\langle\sigma\rangle \odot \phi}{\langle\Sigma - \sigma\rangle \odot \phi} \quad \text{Constellation monotonicity}
\]

\[
\frac{\langle\Sigma - \sigma\rangle \odot \phi}{\neg\langle\sigma\rangle \odot \neg \phi} \quad \text{Regularity + MP}
\]

\[\square\]
4.7 Summary

In this chapter we have shown completeness for homogeneous ATL; the logic which results from removing the names of agents from ATL in such a way that only role membership determines the strategic ability of the agents. As the particular agents are not relevant, there are no names in the logic, nor reference to them in the semantic valuations of formulas.

We have argued that one, we feel, attractive way of reasoning about groups of anonymous agents is by referring to their strategic abilities exclusively in terms of the role they enact in the model. By replacing coalitions with their anonymous counterpart, the constellations, we have shown that HATL is complete with respect to RCGS models. The difference between ATL and HATL being only the lack of names, we feel this justifies the claim that HATL is indeed a logic corresponding to an “anonymous version” of ATL.

It was an easy exercise to see that HATL corresponded to RATL under the assumption of a state independent role assignment. That is, as long as agents did not change roles, the anonymous version of ATL corresponds also to the version of ATL where every agent (in the same role) had the same strategic abilities. That is, we feel justified in claiming that the logic of ATL with strategic homogeneity (as expressed by the \((\text{Hom})\) axiom), RATL, corresponds to anonymous ATL, HATL.

The smallest non-zero constellation, a position, can be seen to be a unique (but arbitrary) agent as long as the role assignment is state independent. When this is no longer the case, but we retain a fixed social structure in order to avoid non-designating references, a difference emerges.

We have not presented any formal results concerning the interplay between agents and positions. We have however presented an example of how these two concepts can be substantially different when we permit the agents to change roles.

In our discussion of the notion of a role in the previous chapter, we mentioned a different interpretation of the term which has been introduced into the MAS literature. A “role” in Ryan & Schobbens’ [72] was closely related to a task (or expectation). This relied on a refinement relation (introduced by Alur et. al. in [7]) which linked an abstract agent responsible for a task (the “sender” in the example from [72]) to a set of concrete agents, each responsible for its concrete sub-task.

We envision that a combination of the approach presented here in combination with the abstraction tool presented in [7, 72] can be used together to synthesize a social structure (set of roles) which satisfy a certain set of constraints.

One could create a social structure with one role for each constraint and refine the abstract agents thus created until the actions agents must perform are atomic. A role, at the most abstract level, can be interpreted to correspond to the responsibility of maintaining/achieving some constraint/goal. It will also correspond to a constellation in the refined, more detailed, model. Each position may be occupied with any agent fitting the bill. If there are no concrete agent, we may continue refining our model yielding (possibly) additional roles each with responsibilities derived from the responsibilities of the role or position just refined, according to the refinement.

We posit that when specifying constraints for a complex system in this manner, it is natural to start with specifications given abstractly and formulated in terms of positions rather than agents. However, care must be taken when relying on such a synthesis as
the strategic abilities of (concrete) coalitions are not necessarily the same as those of constellations.

It remains an interesting challenge to investigate the possibility of designing a social structure (a system by specifications in terms of positions, roles and tasks or goals), and test whether a given heterogeneous system could be made to conform to these specification for example by means of implementing (normative) constraints on the heterogeneous system.
Chapter 5

Norm compliance

5.1 Background

In the two previous chapters, we have introduced a semantic structure RCGS which we have shown to be equivalent to CGS models as a semantic class for ATL. We saw that there are benefits for modeling scenarios by RCGS rather than CGS when model checking. We also saw that RCGS naturally gave rise to models where the agents are anonymous within the role they occupy.

In this chapter we will introduce the notion of a normative system which places behavioral constraints on the agents, and develop a logic for reasoning about the strategic abilities of the agents in the system when some of them comply to these restrictions. As argued in the previous chapter, we do not need to keep the names of the agents when they all belong to the same homogeneous role. However, when only a certain set of agents comply to the normative system, or the norm is not such that it preserves anonymity, the role is no longer homogeneous. We will define normative systems so that they permit differentiating among the agents based on their identity. So even if every agent complies to the norm, the collection of agents will not necessarily be homogeneous. Therefore, we need to define this logic with agent names.

In Section 5.3 we present a logic Norm compliance RATL (NCRATL), which extends the logic of RATL, with one new connective $\langle A \rangle$ stating intuitively that “the agents in $A$ comply to the normative system”. This connective is very similar to the connective in Norm compliance CTL (NCCTL) [4].

Then in Section 5.6 we discuss how a logic corresponding to NCRATL but applying to constellations rather than coalitions may be defined. We call this logic Norm compliance HATL (NCHATL). This shift introduces some additional challenges not encountered in the logic of NCRATL. Particularly, we may no longer easily reason about non-anonymous (individual) norms, and we need to introduce a weak form of quantification to account for the uncertainty about the coalitions which instantiate the topical constellations.

This chapter is a continuation of the work presented in [65].
5.2 Normative systems and compliance

The main formal framework we present in this chapter, NCRATL, carries with it two terms with many possible interpretations. First of all, the term “norm”. We will devote some space to elaborate on the origin in the scientific literature of the interpretation our systems rely on, ending up with a strict formal definition. Second, we discuss compliance. Given the formal definition we will present of a norm, what compliance means will be straightforward. However, the focus on compliance sets up a contrast between our logical investigations and the related and well-studied problem of mechanism design in game theory.

5.2.1 Norms and normative systems

We borrow the term “norm” from discussions of human affairs. We will use the term metaphorically as a systematic way of studying the effects of constraining the behavior of agents, and as we will see when discussing compliance, the effects of the agents abiding to or violating these constraints.

The notion is used very broadly when applied to studies of human behavior. An early article promoting a unified definition of norms can be found in [39]. There, Gibbs provides a survey of several uses in the sociological literature of the term, and a “typology of norms” consisting of criteria which enables him to classify norms into twenty classes (five of which, he suggests might be null classes, i.e., logically impossible or empirically untestable). In summary of the definitional attributes (as opposed to the contingent), Gibbs concludes that [39] (original emphasis):

“A norm in the generic sense [...] involves: (1) a collective evaluation of behavior in terms of what it ought to be; (2) a collective expectation as to what behavior will be; and/or (3) particular reactions to behavior, including attempts to apply sanctions or otherwise induce a particular kind of conduct.”

We will offer a definition which can be seen to focus on an essential part of the first two. However, our analysis will be very instrumental. Regarding the first point, our formalism will not address what ought or ought not be, but it will provide a tool which permits us to evaluate the behavior under the assumption of some degree of compliance. That is, our framework can be seen as to aid in the evaluation of a particular normative system.

The way we do this, and in relation to the second point, is that we will provide a framework which will let us deduce what behavior will be from some assumption of compliance. We depart somewhat from the formulation as we will not explicitly relate our norms to the collective expectations. Indeed, our norms will be taken as exogenous givens, and our framework will discuss the consequences of the/some set of agents complying to the given norms.

More modern expositions of norms depart from the early “typology” but still remain similar to what we saw in Gibbs’ statement. Both in terms of what a norm, or particularly a social norm as we observe in human societies, is or can be said to be, and also in matters of compliance. What a norm is together with explanations of emer-
gence, persistence, and compliance are interconnected in a complex interplay and there is no final agreement about the theory which best explains them.

A characteristic of norms, as recognized throughout the sociological study of norms, is the role norms may play in cooperation. However, it would be incorrect to equate norms with behavior, nor even regular cooperative behavior. This, it can be argued would include certain patterns of behavior which are not considered norms or the result of norms [14]. Something can yet be said about the induced behavior, norms encourage conformity. As will be evident in our application of what we define norms to be, we will permit potentially grave violations of this particular characteristic.

A main problem in studies of social norms is the question of compliance, a question we will touch upon. The questions we ask (and, in the abstract, answer) are “what are the consequences of complying to this norm”? Traditionally the more pressing problem has been “why should an agent follow this norm”? The analysis of compliance to norms by humans has been observed to be some combination of the agent’s empirical and normative expectations, possibly coupled with some expectations concerning (positive and/or negative) sanctions (ibid).

We believe that in order to ascribe to society a certain belief about behavior (e.g., for a constituting agent to conclude that the other agents may expect compliance), the agent in this society needs to be able to consider the consequences of society complying. This, in part, is one problem we hope to be able to assist the agent in solving.

As discussed in Chapter 1, the term was initially introduced into the MAS literature by Shoham & Tennenholtz in [75, 76] then called “social laws”, and further developed under the name “norms” or “normative systems” for CTL (e.g., [1, 2, 5]) and for ATL [43] also with the notion of compliance. The reading is then that in some given situation q, the action α which agent a could possibly perform might be deemed illegal. If the agent never performs an illegal action, we say the agent is compliant. Hence we separate actions (relative to an agent and a state) into the “legal” actions, and the “illegal” or “blacklisted” actions.

Why should an agent comply to this arbitrary constraint? We do not study the agent’s motivation to comply to a norm, but rather the strategic consequence of such compliance, or the lack there of, as it may be. This differentiates the study of norms in MAS from both norms as investigated in e.g., sociology or other social sciences (as we have just seen), and that of mechanism design in game theory as we will now explain.

The problem addressed in mechanism design is to construct a game such that, no matter the agents’ true preferences, we obtain some desired behavior in a system we construct. That is, similar to our assumptions, the modeler is agnostic with respect to the agents’ preferences, but has some description of what behavior the constructed system induces (particularly under the assumption of rational behavior). See for example [61, 74] for an introduction to mechanism design.

In the logical framework we are proposing, we do not consider the agents’ motivations and preferences, but rather only what strategic abilities the agents posses under assumptions of compliance or non-compliance. We are agnostic about the agents’ preferences because they do not enter into the questions we pose.

1This is not strictly speaking a correct characterization. Rather, we equate the assumption that an agent a complies, with the fact that the actions deemed illegal are not options for this agents.

2We emphasize “strategic” here to make it clear that we are not investigating possible reactions or sanctions often associated with norms, nor are we addressing the deontic implications.
Clearly, in order to claim that the constructions we study model norms in the broadest sense, we would have to discuss other aspects such as how sanctions are imposed when the norm is violated, how well grounded is the norm in the society, and so on. Some such work is being carried out for these norms. Our approach here, although we deviate from the mechanisms employed to study norms in human societies, are motivated in part by the following.

1. We are continuing the tradition which is gaining grounds in the MAS community,

2. We believe that mechanisms for reasoning about the consequences of norms and compliance to these are essential to the evaluation of the value of a norm and the value of compliance to such a norm, and finally

3. We see the mechanism as an effective tool for sculpting, or introducing small changes to, an established model, and particularly reintroducing some level of heterogeneity in an otherwise completely homogeneous system.

The function norms fulfills in MAS is generally that of modifying a system on the micro-level (the actions of the agents), in order to produce some macro-level effect (a property of the system). That is, they are a way of coordinating the behavior of the agents to ensure that the system as a whole satisfies certain properties. For example, computer components that need to cooperate with regard to access to common resources, need to make sure that there is no deadlock situation, or that the system generally avoids entering unsafe or otherwise undesirable states.

Clearly the mechanism referred to as a “norm” in the MAS literature does not do full justice to the term as it is used colloquially, in other branches of science or indeed in other branches of formal logic (e.g., deontic logic). It may therefore seem reasonable to call it by a name referring to phenomena which to a higher degree resembles what is modeled.

It may be more suitable, perhaps, to refer to the constructs as “conventions”, “agreements” or “commitments”, but with these disclaimers in place, we will continue using the term as sketched, and as it is being used in the MAS literature.

### 5.2.2 Norms and compliance defined

We will define a normative system to be a set of actions for every agent in every state, but such that every agent always have at least one action available. For an introductory text to this use of norms in MAS (there also referred to as “social laws”), see [82].

**Definition 5.1 (Normative system).** Given a RCGS $M = (A, R, \rho, Q, \Pi, \pi, A, \delta)$, a norm $\eta$ is a map in every state $q \in Q$

$$\eta_q(a) \subseteq [A_{q, \rho(q,a)}]$$

In any state $q$, the number of actions available for an agent $a \in A$ is $A_{q, \rho(q,a)}$. However, some of these actions might be deemed illegal by the normative system, so if $a$ complies with the norm, the remaining actions (i.e., the legal actions) are
Notice that since $\eta_q(a)$ is a proper subset of $A_q, \rho(q, a)$, there is always at least one permitted action.

When a possible action $i \in A_q, \rho(q, a)$ is in $\eta_q(a)$, we say it is “blacklisted” or “illegal” for $a$ at $q$, otherwise the possible action which is not in $\eta_q(a)$ is referred to as a “legal” action.

We model compliance to a normative system by examining the strategic consequences of the complete profiles where the agents have selected only legal action. We also discuss partial compliance in the same sense investigated for example in NCCTL [5]. That is, we are reasoning about the strategic consequences when some set of agents $C \subseteq A$ comply and the remaining agents $A \setminus C$ select actions from the entire set of available actions. Technically, we reason about this by restricting the full normative system to contain only prescriptions for the complying agents. We use the $\uparrow$ symbol to denote such restrictions, and formally define it below:

**Definition 5.2.** Given a normative system $\eta$ and a set of agents $C \subseteq A$, we define $\eta$ restricted to $C$ (denoted $\eta \uparrow C$) as follows

$$(\eta \uparrow C)_q(a) = \begin{cases} 
\eta_q(a) & \text{if } a \in C \\
\emptyset & \text{otherwise}.
\end{cases}$$

The norm construction provides the opportunity to restrict the agents’ actions on a state and identity basis. If our initial system is heterogeneous in that different agents have different abilities, we can use these norms to increase homogeneity by blacklisting actions which differentiate between agents.

On the other hand, if we start out with a homogeneous system, we may introduce heterogeneity by blacklisting certain actions for some, but not all, agents.

### 5.2.3 Reintroducing heterogeneity – the canvas society

As just discussed, the mechanisms we study in this chapter, those of “norms” and “compliance”, do not directly correspond to the entire phenomena often called by such names.

Starting with a *completely homogeneous society*, which we may refer to as a “canvas society”, we can design a “norm”, the implementation of which will produce *any* arbitrary “sub-system”. When we have discussed model checking in Section 5.5, we will see that this may even be an efficient way of modeling certain heterogeneous systems.

It should be clear, at least intuitively, how we can now use the notion of norm compliance to reintroduce heterogeneity into our models. We return to the trains we encountered in the examples from Chapter 3. In the following example we show how we can grant a single train in an otherwise homogeneous group autonomy, and the unique ability to enter the tunnel. This time we will not have a controller granting permission, but let the trains manage the coordination on their own.

**Example 5.3.** Consider a scenario similar to the three trains in Example 3.6 where we have three trains which need to coordinate to get through a tunnel. We will have an

\[ [A_{q, \rho(q, a)}] \setminus \eta_q(a). \]  

Recall that we treat the number $A_{q, \rho(q, a)}$ in some state $q$ as the number of actions available to the agents belonging to the role enacted by agent $a$. The set of actions available is then the set of natural numbers from 1 to $A_{q, \rho(q, a)}$, i.e., $[A_{q, \rho(q, a)}]$. 
initial state $q_0$ from which we may transition into one of either $q_{1,a}$, $q_{1,b}$ and $q_{1,c}$ by, say, majority voting with lexicographic tie breaking (similar to Example 3.6).

We omit the description of states correspond to trains $b$ or $c$ being elected. They are analogous. Consider the RCGS consisting of three trains, where in the initial state, one train is “elected”. The transitions for the case where $a$ has been elected is as illustrated in Figure 5.1.

![Figure 5.1: Illustration of coordination by compliance. (Example 5.3.)](image)

We also need to describe the normative system we are considering. We describe the actions available and the blacklisted actions every state. We describe these for the states in which $a$ is elected ($q_{1,a}$ and $q_{\text{tunnel},a}$). The corresponding states for trains $b$ and $c$ would be described similarly, only exchanging the name $a$ with either $b$ or $c$ depending on which train has been elected.

$q_0$: There are three actions available, corresponding to voting for train $a$, train $b$, and train $c$, respectively. The norm, for every agent $i \in \{a, b, c\}$ is empty $\eta_{q_0}(i) = \emptyset$.

$q_{1,a}$: There are two actions available. The first action corresponds to “enter the tunnel”, and the second corresponds to “yield” or “wait”. The normative system blacklists the action for entering the tunnel for agents $b$ and $c$ since they were not elected. In this example we leave the normative system for $a$ in this state empty. (We could blacklist the action of “yielding” for agent $a$ to ensure that agent $a$ enters the tunnel.)

$$\eta_{q_{1,a}}(a) = \emptyset$$
$$\eta_{q_{1,a}}(b) = \eta_{q_{1,a}}(c) = \{1\}$$

If more than one agent simultaneously selects “enter the tunnel”, we transition to the bad state $q_{\text{crash}}$. If no agent attempt to enter the tunnel ($(0,3)$), we remain in this state, and if exactly one agent attempts to “enter the tunnel”, we transition into $q_{\text{tunnel},a}$.

$q_{\text{tunnel},a}$: There are two actions available “exit tunnel” and “wait”, respectively. The normative system in this state is again empty $\eta_{q_{\text{tunnel},a}}(i) = \emptyset$ for every $i \in \{a, b, c\}$.
5.3 The logic NCRATL – Norm compliance RATL

(Again, we could ensure some behavior, such as b and c not being able to force train a out of the tunnel or train a not being permitted to linger, by a different normative system.)

$q_{\text{crash}}$: There is only one possible action in this state, and hence it is impossible to define a normative system other than the empty one, $\eta_{q_{\text{crash}}}(i) = \emptyset$ for every $i \in \{a,b,c\}$.

Under the assumption that every train complies with the norm, intuitively, it seems that we can avoid the collision state. We seem to have regained some heterogeneity under the assumption that the agents comply to the norm.

The example given now needs to include states which were not necessary in the example where we gave the trains autonomy by moving them into a designated role (Example 3.6). We now need to model all possible outcomes, and then reduce the model to a model with satisfactory heterogeneity by designing a normative system. Notice, however, that we do not know which train chose the action “enter tunnel” in the transition from $q_{1,a}$ to $q_{\text{tunnel},a}$. We can only know this under the assumption that at least agents b and c comply to the norm.

The fact that the heterogeneity is not “complete”, i.e., it is not the same level of heterogeneity as we can (and generally do) model in a CGS model, will be significant when we consider model checking for this logic. It will turn out that we can reintroduce heterogeneity, as shown in the previous example, while retaining the benefit for computational complexity as we demonstrated in Chapter 3. Particularly, we avoid the model checking being exponential in the number of agents.

5.3 The logic NCRATL – Norm compliance RATL

In this section we define the logic of norm compliance ATL with roles (NCRATL). The logic is an extension of the logic RATL defined in the previous chapter. The logic enables us to reason about RCGS models in which we also have constraints on the behavior of the agents, as modeled by norms, under the assumption that a subset of the agents actually comply to these constraints.

The ability to reason about partial compliance, or equivalently about deviation from the norm or violation of the constraints, is useful in cases where we do not know the agents’ motivation or preferences. As discussed in [5], the reasons to comply or not comply might be one of, or a combination of, several possible considerations. The agent might be malicious and attempt to make the system enter a state the modeler wishes to avoid. In these cases, we may want to verify that the system is sufficiently robust in the sense that as long as some constellation of agents comply, we guarantee that malicious agents will be unsuccessful. A rational agent might be motivated to violate the constraints because they do not seem (to the agent) to be in her best interest.

Otherwise, the agents opt to not comply to the norm for completely arbitrary reasons. The agent might be deliberately irrational, or not sufficiently informed about the system to always make a choice of action which is deemed legal. In these cases, one would want to find the thresholds, so to speak, of the types and numbers of agents that must comply in order for the system to behave in a desired manner.
We define a language which extends RATL with a single connective $\langle A \rangle \phi$ which states, informally, that “if the agents in $A$ comply to the norm, then $\phi$ (holds)”. The norm is taken as a part of the model (as will be defined shortly in Definition 5.6).

**Definition 5.4.** Given a countable set of proposition symbols $\Pi$ and a finite, non-empty set of agents $\mathcal{A}$, the language $L_{NCRA\text{T}L}$ is defined by the following BNF

$$
\phi := p | \neg \phi | \phi \land \phi | \langle \langle A \rangle \rangle \bigcirc \phi | \langle \langle A \rangle \rangle \Box \phi | \langle \langle A \rangle \rangle \phi \cup \phi_2 | \langle A \rangle \phi
$$

where $p \in \Pi$, and $A \subseteq \mathcal{A}$.

Unlike earlier approaches, we will not *implement* normative systems into our models, but rather keep, as a part of the description of the state of the system, which agents comply to which normative system in the context. Hence, the satisfaction relation now has a slightly new signature. It also entails that we need to take care of the compliance in the notion of a strategy. We will shortly define the satisfaction of, say formulas on the form $\langle \langle A \rangle \rangle \bigcirc \phi$ with respect to some state of some RCGS, a normative system $\eta$ and a set of agents $C$ which are assumed to comply to this normative system. Intuitively, this will be defined as there being an $A$-profile not violating the assumption that $C$ complies to $\eta$, such that for every extension of the $A$-profile which does not violate this assumption, the successor state necessarily satisfies $\phi$.

This entails that we need to define a certain set of strategies, namely those which specify an $A$-profile in every state which does not violate the assumption that $C$ is complying to $\eta$ in that state. In the definition, we need to reintroduce the identity of the acting agents temporarily. We will see in the next section that this is not necessary in order to compute this set.

**Definition 5.5 ($\eta \upharpoonright C$)-compliant $A$-profiles.** Given a RCGS $S = \langle A, R, \rho, Q, \Pi, \pi, \Delta, \delta \rangle$, a normative system $\eta$ and two coalitions $A, C \subseteq A$, an $\eta \upharpoonright C$-compliant $A$-profile in a state $q$ is any $A$-profile $v \in P(q, A)$ such that there is a collection of maps (one for each role $r$), $(\tau_r)_{1 \leq r \leq |R|}$ with

- $\tau_r : A_{q,r} \rightarrow \mathbb{A}_{q,r}$,
- $\tau_r(a) \notin (\eta \upharpoonright C)_q(a)$, and
- $|\tau_r^{-1}(i)| = v_r(i)$ where $\tau_r^{-1}(i) : \mathbb{A}_{q,r} \rightarrow 2^{A_{q,r}}$ such that $\tau_r^{-1}(i) = \{ a \in A_{q,r} | \tau_r(a) = i \}$.

That is, the $\eta \upharpoonright C$-compliant $A$-profiles are anonymous partial profiles as defined for RCGS, such that there exists a corresponding non-anonymous partial profile as defined for CGS in which each agent has chosen an action which is both available and legal to her when also the normative system is taken into consideration.

Not only profiles need to be restricted to the appropriate subset of their original form, but also the notion of a strategy and the set of outcomes. We refer to the $\eta \upharpoonright C$-compliant $A$-strategies as $\text{strat}_{\mathcal{C}}^\eta(A)$, and the outcome of a strategy (where the strategies we consider are all $\eta \upharpoonright C$-compliant) as $\text{out}_{\mathcal{C}}^\eta(s, q)$ for any $\eta \upharpoonright C$-compliant strategy $s$ and state $q$. 
5.4 Characterizing $\eta \upharpoonright C$-compliant $A$-profiles

**Definition 5.6.** Given a RCGS $S = \langle A, R, \rho, Q, \Pi, \pi, A, \delta \rangle$ and a state $q \in Q$, a normative system $\eta$, and a coalition $C \subseteq A$, the satisfaction relation $|=_{\text{NCRA TL}}$ (denoted $|=_{\text{NCRA TL}}$ where there is no confusion) is defined inductively as follows:

\[
S, \eta, C, q \models p \iff p \in \pi(q)
\]

\[
S, \eta, C, q \models \neg \phi \iff \text{not } S, \eta, C, q \models \phi
\]

\[
S, \eta, C, q \models \phi_1 \land \phi_2 \iff S, \eta, C, q \models \phi_1 \text{ and } S, \eta, C, q \models \phi_2
\]

\[
S, \eta, C, q \models \langle \langle A \rangle \rangle \Box \phi \iff \text{there is } s_A \in \text{strat}_c^{\eta}(A) \text{ such that}
\]

\[
\text{for all } \lambda \in \text{out}_c^{\eta}(s_A, q), \text{ we have } S, \eta, C, \lambda[1] \models \phi
\]

\[
S, \eta, C, q \models \langle \langle A \rangle \rangle \Diamond \phi \iff \text{there is } s_A \in \text{strat}_c^{\eta}(A) \text{ such that}
\]

\[
\text{for all } \lambda \in \text{out}_c^{\eta}(s_A, q) \text{ and all } i \geq 0,
\]

\[
\text{we have } S, \eta, C, \lambda[i] \models \phi
\]

\[
S, \eta, C, q \models \langle \langle A \rangle \rangle \phi_1 \cup \phi_2 \iff \text{there is } s_A \in \text{strat}_c^{\eta}(A) \text{ such that}
\]

\[
\text{for all } \lambda \in \text{out}_c^{\eta}(s_A, q) \text{ for some } i \geq 0 \text{ and all } 0 \leq j < i,
\]

\[
\text{we have } S, \eta, C, \lambda[j] \models \phi_2 \text{ and } S, \eta, C, \lambda[i] \models \phi_1
\]

\[
S, \eta, C, q \models \langle A \rangle \phi \iff S, \eta, A, q \models \phi
\]

The language RATL defined in the previous chapter is clearly a fragment of this language. As long as the set of agents complying to the normative system is empty $C = \emptyset$, or the norm is an empty norm ($\eta_q(a) = \emptyset$ for each state $q$ and agent $a$), the satisfaction relation is the same for this fragment, i.e., we have not changed the semantics with respect to the strategic aspects, but only added the connective for altering the state of compliance.

5.4 Characterizing $(\eta \upharpoonright C)$-compliant $A$-profiles

The crux of our argument for an efficient model checking algorithm, will be that we can compute the set of $(\eta \upharpoonright C)$-compliant $A$-profiles efficiently. Notice that if we use the definition of $(\eta \upharpoonright C)$-compliant $A$-profiles to compute this set directly, we may need to consider every element several times. This is because the definition invites us to construct a regular, non-anonymous, action profile (à la CGS) to ensure compliance, and then transform these into anonymous (RCGS) profiles.

We do not need to construct the non-anonymous profile, however. We simply need to verify that for a given anonymous labeling, there exists a corresponding non-anonymous profile such that the norm is not violated. The proof is constructive, so whenever there exists such a profile, we will construct it.

5.4.1 Profiles as flows

We will construct a simple network with a source $s$ and a sink $t$. Between these two special nodes, we will have two layers of nodes, corresponding to the topical agents, and their possible actions. Verifying that a certain profile $v \in P(q, A)$ is indeed also in
$P^\eta_n(q,A)$ will be solved by a call to a max-flow algorithm on this network. In general, we see that there is a tight connection between profiles and flows in the network.\footnote{A s-t-flow in a network $G = (N,E,l)$ where $N$ is a set of nodes containing $s$ and $t$, $E \subseteq N^2$, and $l : E \rightarrow \mathbb{N}$ is a capacity map, is a triple $(R,f,F)$ where $R \subseteq E$ and $f : R \rightarrow \mathbb{N}$ such that $f(x) \leq l(x)$. And such that for every $x \in N \setminus \{s,t\}$, we have $\sum_{y \in N} f(y,x) = \sum_{y \in N} f(x,y)$ and $F = \sum_{(x,t) \in R} f(x,t)$. We call $F$ the flow or load in the network[52]. Alternatively, we may define a set $F \subseteq \mathbb{N} \times ([s] \cdot (N \setminus \{s,t\})^r \cdot \{t\})$ containing pairs of natural numbers and s-t-paths in $G$ such that for every $(x,y) \in E$ the x-y-load in $F$ does not exceed $l(x,y)$, where, whenever there are $m$ occurrences of $(xy)$ in $p$, the x-y-load in $(n,p) \in F$, $c(n,p) = mm$, and the x-y-load in $F$ is $\sum_{(n,p) \in F} c(n,p)$. We call $(n,p) \in F$ a flow/an n-flow in $G$.\footnote{In game theory, a pure is a concrete action. The alternative of mixed strategy is a probabilistic division among several possible actions.}}

The edges in these networks are labeled by the capacity they have, and the well-known problem of maximum flow through such a network is to find a distribution of numbers to these edges, called loads, which satisfy certain properties. For these numbers to constitute a flow, the sum of such assigned numbers on edges going into a node, must equal the sum of the assigned numbers on the edges going out from the node. A maximal flow is hence a distribution of load over the edges in the network.

The network is often thought of as representing an actual network with capacities such as a network of oil pipes which can relay some number of liters of oil per minute, or an electronic network where the various channels has a certain bandwidth. Such limitations are the capacities we will label the edges with. The load we assign to any edge must of course not exceed its capacity.

A profile on the other hand, be it anonymous as in RCGS models or non-anonymous as in CGS models, is an assignment of tasks to agents. Colloquially one often speaks of a workload, and in our algorithm we make use of the connection between assigning tasks to agents and loads to edges with capacities. In the network we construct, a flow will run through an agent (represented by a node), and also through some task (also represented by a node).

We will only use integer flows. This corresponds to how actions are handled in ATL in that we have pure action profiles. We will construct a network such that the total capacity of all edges into a node is exactly one, hence the flow through any given agent is either zero or one. If it is zero, the agent does not act (has no workload), otherwise the agent performs the action which the flow out from the agent node passes through.

This tight connection between flows in a network and action profiles for agents, seems like a fruitful analogy, and it is particularly fruitful, as we will argue in the remaining parts of this section, when the actions available to the agents coincide. The fact that the agents belonging to the same role have the same actions (and the consequences of these actions are identical regardless of which agent performed them), is what yields a running time bounded above by an expression polynomial in the number of agents.

For every agent $b \in B$, we will add a node corresponding to each of her legal actions. As we are representing actions by numerical indices, we tag the actions with the agent’s role. Several agents in the same role may very well have some coinciding legal actions, but we only add one instance of each action per role.

We define this set of action nodes $X$ as the actions available to some agent $b$, tagged with the role that this agent enacts (defined formally in Definition 5.7), and we add an edge from $b$ to $(i, \rho(q,b))$, if, and only if, $i \not\in \eta_q(b)$. 
5.4 Characterizing \((\eta \mid C)\)-compliant \(A\)-profiles

Consider some \(B\)-profile \(v \in P(q, B)\) (i.e., a regular \(B\)-profile where we have no assumptions about compliance). For each role \(r\) which contains an agent \(b \in B\) (i.e., \(1 \leq r \leq |R|\) such that \(B_{q, r} \neq \emptyset\)), for each action \(i\) available to the agents in this role (i.e., \(i \in A(q, r)\)) which is performed by some agent (i.e., \(v_r(i) \neq 0\)), do either:

- If \((i, r)\) is not a node, conclude that \(v \notin P^n_C(q, B)\).
- Otherwise, if \((i, r)\) is a node, add an edge from \((i, r)\) to \(t\) labeled with capacity \(v_r(i)\) to the network.

Clearly, if \(v\) prescribes some (number of) agent(s) to perform an action which is not legal to any agent in \(B\), we can safely conclude that \(v \notin P^n_C(q, B)\) and we need not continue constructing our network. (These nodes will become “islands” in the algorithm we present, having no incoming edges and hence will be mute in the construction.)

Otherwise, we will now argue, we have that \(v \in P^n_C(q, B)\) if, and only if, there is a flow from \(s\) to \(t\) with capacity \(|B|\).

We give this definition formally, then prove the correspondence between profiles and flows.

**Definition 5.7.** Given a RCGS \(S\), a state \(q\) in \(S\), a normative system \(\eta\) for \(S\) and coalitions \(C\) and \(B\) such that \(C\) complies to \(\eta\), the profile network is a graph \(G = (B \cup X \cup \{s, t\}, E, l)\) where

- \(X\) is the set of role-tagged actions legal to some agent in \(B\)

\[
X = \{ (i, j) \mid \exists b \in B, j = \rho(q, b), i \in [A(q, j)] \setminus (\eta \mid C)_q(b) \}
\]

- for every agent \(b \in B\), there is an edge \((s, b) \in E\), and for every \((i, \rho(q, b)) \in X\), there is an edge \((b, (i, \rho(q, b)))\) if, and only if, \(i \notin (\eta \mid C)_q(b)\). And finally, for every \((i, j) \in X\), there is an edge \(((i, j), t) \in E\).

- Consider an edge \((x, y) \in E\). We label these edges

\[
l(x, y) = \begin{cases} 1, & \text{if } \{x, y\} \cap B \neq \emptyset \\ v_r(i) & \text{if } x = (i, r) \end{cases}
\]

The condition \(\{x, y\} \cap B \neq \emptyset\) is true for exactly the edges which are adjacent to a \(b\)-node. The remaining edges are edges from a role-tagged action to the sink \(t\).

**Lemma 5.8.** Given a RCGS \(S\), a state \(q\) in \(S\), a norm \(\eta\) for \(S\), two coalitions \(B, C \subseteq A\), a \(B\)-profile \(v \in P(q, B)\) and the graph as defined in Definition 5.7, we have \(v \in P^n_C(q, B)\) if, and only if, there exists a flow from \(s\) to \(t\) with capacity \(|B|\).

**Proof.** We show the claim in two parts.

\(\Rightarrow\) Suppose \(v \in P^n_C(q, B)\). By Definition 5.5 we know that there exists a collection of maps \((\tau_r)_{1 \leq r \leq |R|}\), such that for every agent \(b \in B\), \(\tau_r(b) \in [A(q, \rho(q, b))] \setminus (\eta \mid C)_q(b)\).

Since both in Definition 5.5 (first and second point) of compliant profiles and in Definition 5.7 (second point defining the outgoing edges), we only associate an
Let us consider a very simple example. Let B consist of two agents b and b, belonging to the single role \( \{r\} \) with 2 available actions. Let \( \eta \), \( \eta \) comply. Say \( \eta \), \( \eta \) can, and must, be a non-zero load. Say this (unique) edge is the edge \((b, (i, \rho(q,b)))\). Let \( \tau(b) = i \).

From a \( b \)-node to \( t \), all paths pass through action nodes corresponding to the legal actions for \( b \). Exactly one edge exiting \( b \) can, and must, be a non-zero load. Say the (unique) edge is the edge \((b, (i, \rho(q,b)))\). Let \( \tau(b) = i \).

The sum of the weights on the incoming edges into \( t \) equals \( |B| \) by construction. Since we assumed that the flow had capacity \( |B| \), we have that for every role \( r \), and every action \( i \in \{\mathbb{A}(q,r)\} \), we have \( |\tau^{-1}(i)| = v_r(i) \). We conclude that \( \tau \) establishes that indeed, \( v \in P_C^n(q,B) \).

\( \square \)

**Example 5.9.** Let us consider a very simple example. Let B consist of two agents \( b_1 \) and \( b_2 \) belonging to the single role \((r = |R| = 1)\) with \( \mathbb{A}(q,r) = 2 \) available actions. Let \( \eta(b_1) = \eta(b_2) = \{2\} \). We suppose that only agent \( b_1 \) complies.

Now, consider the two \( B \)-profiles \( v = (2,0) \in P(q,B) \) and \( v' = (0,2) \in P(q,B) \). To determine whether these are also \( (\eta \triangleright \mathbb{A}(q,b_1),r_1) \)-compliant profiles, i.e., whether \( v, v' \in P_C^n(b_1)(q,B) \), we construct the network as detailed earlier. The networks will be identical except the labels on the edges into \( t \) will need to be changed to correspond to the profiles.

In the network on the left in Figure 5.2, we see the network constructed to test whether \( v \in P_C^n(b_1)(q,B) \). Indeed, we see that there is a flow with capacity \( |B| = 2 \). The network on the right in the same figure, shows the labels corresponding to the profile \( v' = (0,2) \). We see that there is no flow from \( s \) to \( t \) with size \( |B| = 2 \).

We conclude that \( v \in P_C^n(b_1)(q,B) \) and \( v' \notin P_C^n(b_1)(q,B) \). That is, that \( v \), but not \( v' \), is a \( (\eta \triangleright \mathbb{A}(q,b_1)) \)-compliant \( B \)-profile.

### 5.4.2 Computing \( P_C^n(q,A) \)

Computing the set of profiles compatible with the assumption of some coalition \( C \) complying to \( \eta \), \( P_C^n(q,A) \), directly as indicated by Definition 5.5 would be exponential. That is, computing all possible non-anonymous profiles and then computing the corresponding anonymous profile is not feasible. It has been observed in the literature of algorithmic game theory [18, 20], however, that we need not compute all these profiles. In this section we show that we can compute the elements \( P_C^n(q,A) \) by performing a polynomial time algorithm on the elements of \( P_C^n(q,A) \) which will decide whether the arbitrary, hence not necessarily complying profile, corresponds to a complying profile.

This observation relies on a correspondence to Hall’s marriage theorem (see e.g., [52]). In our approach we rely on a closely related problem, that of matching
by means of a correspondence to the maximum flow through a bipartite graph augmented by a source node and a sink node. The algorithm, which we call profiles, is shown in Algorithm 3.

We disregard the representation of sets in our analysis. A more fine grained analysis could take this into consideration. In the main model checking algorithm, mcheck, we make use of the set theoretic operations of union, intersection and complement, but in the profiles algorithm we only add elements to sets and test for membership. The set $E$ (which we add to, and test membership in) is bounded above by $|A| a$ where $a$ is the number of actions available to the agents in the role which has the most number of actions available to its members.

$$a = \max \bigcup_{q \in Q} \{|A_q, 1|, \ldots, |A_q, R|\}$$

These simple set operations do not contribute to a dramatic increase in complexity and we neglect it in the remaining analysis.

Observe that the profiles algorithm, for each $v \in P(q, B)$, sequentially performs two blocks of instructions. The first block constructs the graph from Definition 5.7. The second block computes the maximum flow in the network. Let $a$ be as just described.

Constructing the graph is bounded above by $O(|A| a)$. The second, computing the maximum flow, can be performed by e.g., the Ford-Fulkerson algorithm (see e.g., [52]). On networks of the form our graph has (i.e., a bipartite graph where all arrows are in the same direction), the algorithm is bounded above by $O(|A||E|)$ where $|E|$ is the number of edges. Since $|E|$ is $O(|A| a)$, we conclude that this block of instructions is bounded above by $O(|A|^2 a)$.
Algorithm 3 profiles($S, q, C, \eta, B$) (Pairs corresponding to edges delineated by ⟨⟩-brackets.)

\[
P^\eta_C(q, B) = \emptyset
\]
for $v \in P(q, B)$ do
  \[
  E = \emptyset
  \]
  for $b \in B$ do
    Add $(\langle s, b \rangle, 1)$ to $E$
    for $i \in \mathcal{A}(q, \rho(q, b))$ do
      if $i \notin (\mathcal{C}_q(b) \cap \eta)$ then
        Add $(\langle b, (i, \rho(q, b)) \rangle, 1)$ to $E$
      end if
    end for
    Add $(\langle i, t \rangle, v_{\rho(q, b)}(i))$ to $E$
  end for
  if maxflow$_{s,t}(E) < |B|$ then
    Add $v$ to $P^\eta_C(q, B)$
  end if
end for
return $P^\eta_C(q, B)$

5.5 Model checking

The main structure of the model checking algorithm, $\text{mcheck}$ (Algorithm 4), for NCRATL remains largely unchanged from the version given for model checking ATL over RCGS models in Chapter 3. There will be two additional parameters (corresponding to the norm and the set of complying agents), but these are not accessed at all, but altered when we evaluate the new compliance connective, and otherwise simply passed on to the $\text{enforce}$ algorithm (Algorithm 5). We show the two illuminating clauses of this algorithm, corresponding to evaluating $\langle \langle A \rangle \rangle$ and $\langle A \rangle$-formulas.

Algorithm 4 Model checking NCRATL. $\text{mcheck}(S, \eta, C, \phi)$ (relies on $\text{enforce}$)

// most clauses omitted
if $\phi = \langle \langle A \rangle \rangle \mathbin{\lor} \psi$ then
  return \{ $q$ | $\text{enforce}(S, q, A, \text{mcheck}(S, \eta, C, \psi))$ \}
end if
if $\phi = \langle C' \rangle \psi$ then
  return $\text{mcheck}(S, \eta, C', \psi)$
end if

The remaining clauses are identical to the model checking algorithm for ATL over RCGS models (Algorithm 1, $\text{mcheck}$, from Chapter 3) except the two additional parameters as mentioned above.

The algorithm $\text{enforce}$, Algorithm 5, is also similar to the one presented in Algorithm 2, $\text{enforce}$, from Chapter 3, but we need to modify it to make use of the profiles algorithm presented in Algorithm 3.

We argued in Chapter 3 that the number of profiles in a given state is not exponential in the number of agents. The $\text{mcheck}$ algorithm will make at most $|Q|$ calls to the $\text{enforce}$ algorithm for each subformula of $\phi$ (this holds for model checking ATL over
5.6 Anonymizing the language

Before we conclude the chapter, we will consider the possibility of reasoning about norm compliance with respect to constellations. The constellations (as discussed in the previous chapter) can be seen to be a particular kind of coalition predicate. With the assumptions we made about constellations (they always denote, and any two coalitions which instantiate the same constellation, are equivalent in our models), we did not need to worry about the mechanics of quantification.

Transferring the norm compliance system to the homogeneous case offers some new challenges in this regard. We need to change our definition of a norm to be “anonym-ous” relative to a role. Throughout the rest of the chapter, we will understand a norm $\eta$ to be a function from states and roles to a proper subset of the actions available to the agents in that role at that state. That is, a norm is a set of maps (one for each state)

$$\eta_q(r) \subset A(q, r).$$

Suppose informally we assume that some constellation $\sigma^\eta$ complies with the (non-empty) norm $\eta$, and we want to query our model whether some other constellation $\sigma$...
can achieve some outcome. If these constellations are such that they can be instantiated by non-disjoint coalitions $A$ and $B$, but also, for example, by coalitions $A'$ and $B'$ that are disjoint, which of the two pairs of coalitions do we choose to instantiate the constellations $\sigma^0$ and $\sigma$?

For example, suppose we have two roles “pres” ($r = 1$) and “pop” ($r = 2$) consisting of at least two voters, $a$ and $b$, in role 2. We interpret the constellations $\sigma = \langle 0, 1 \rangle$ as “a voter”. Consider then the question

“Assuming a voter ($\langle 0, 1 \rangle$) complies with the norm ($\eta$). Can a voter ($\langle 0, 1 \rangle$) achieve $\phi$?”

We may, in one interpretation, take “a voter” to correspond to the same agent (say $a$) in both occurrences of the constellation “a voter”. Otherwise, we might let them be distinct, say we assume $a$ complies and we query whether $b$ can achieve $\phi$.

Clearly, which of these interpretations we choose might influence the answer. If the agent whose strategic ability we are evaluating complies to the norm, she has fewer actions to choose from. On the other hand, if she does not comply, then the agents we consider “antagonists” when evaluating the strategic ability, will have fewer actions to choose from when responding to the first agent’s action.

Quantifying over coalitions which satisfy some coalition predicate is not novel. See, for example Quantified Coalition Logic (QCL) [3]. We can adopt this idea to quantifying over coalitions which may instantiate constellations. We deviate from the notation, however, and write $A$ and $B$ to indicate whether we mean “all coalitions instantiating the constellation $\sigma$” or “some coalition instantiating the constellation $\sigma$”.

5.6.1 Quantifying over coalitions instantiating a constellation

As alluded to, we need to quantify over the coalitions instantiating any given constellation. We will only encounter situations as the one described above where we have one constellation assumed to comply to the norm, and another for which we are evaluating strategic ability. We can fix any coalition we assume complies to the norm, and quantify over the other. We summarize this in the following proposition, rendered in NCRATL.

**Proposition 5.13.** Given two coalition $C, C'$ instantiating the same constellation ($\sigma_C = \sigma_{C'}$), for any constellation $\sigma$, if every coalition $A$ instantiating $\sigma$ (i.e., $\sigma_A = \sigma$) can achieve $\phi$ assuming $C$, and only $C$, complies to the norm, then, and only then, can every coalition $A$ instantiating $\sigma$ achieve $\phi$ assuming $C'$, and only $C'$, complies to the norm. In other words, let $X_\sigma = \{ A \subseteq A \mid \sigma_A = \sigma \}$, then

$$S, \eta, C, q \models \bigwedge_{A \in X_\sigma} \langle \langle A \rangle \rangle \circ \phi \iff S, \eta, C', q \models \bigwedge_{A \in X_\sigma} \langle \langle A \rangle \rangle \circ \phi$$

The previous proposition is not surprising, but permits us to quantify only when evaluating the strategic connectives. We also show that there is a kind of monotonicity in the logic, which will enable a very simple formulation of quantification.

**Proposition 5.14.** For any RCGS $S$ and state $q$ in $S$, a norm $\eta$ for $S$, coalitions $C$ and $A$ such that $S, \eta, C, q \models \langle \langle A \rangle \rangle \circ \phi$, if there are agents $a, a'$ with
• \( \rho(q,a) = \rho(q,a') \),
• \( a \in A \cap C \), and
• \( a' \notin A \cup C \)

then

\[
S, \eta, C, q \models \langle (A \setminus \{a\}) \cup \{a'\} \rangle \circ \phi.
\]

Proof. Assume there is a \( v_A \in P_C^\eta(q,A) \) such that for every \( \overline{v}_A \in P_C^\eta(q,A \setminus A) \) we have

\[
S, \eta, C, \delta(q,v_A + \overline{v}_A) \models \phi \quad \text{(i.e., that } S, \eta, C, q \models \langle A \rangle \circ \phi).\]

Let \( A' = (A \setminus \{a\}) \cup \{a'\} \). We need to show that there is a \( v_{A'} \in P_C^\eta(q,A') \) such that for every \( \overline{v}_{A'} \in P_C^\eta(q,A \setminus A') \), we have \( S, \eta, C, \delta(q,v_{A'} + \overline{v}_{A'}) \models \phi \).

To see this, notice that for the protagonists we have exchanged the complying agent \( a \) with the non-complying agent \( a' \), and for the antagonists we have exchanged the non-complying agent \( a' \) with the complying agent \( a \). Formally, the claim follows from two points:

• Since \( a' \) does not comply, \( a' \) can choose at least the same actions as \( a \), hence \( v_A \in P_C^\eta(q,A') \).
• Since \( a \) complies while \( a' \) does not, we have \( P_C^\eta(q,A \setminus A') \subseteq P_C^\eta(q,A \setminus A) \).

Let \( v_{A'} = v_A \). Because we had \( S, \eta, C, \delta(q,v_A + \overline{v}_A) \models \phi \) for every \( \overline{v}_A \in P_C^\eta(q,A \setminus A) \), we also have \( S, \eta, C, \delta(q,v_A + \overline{v}_{A'}) \models \phi \) for every \( \overline{v}_{A'} \in P_C^\eta(q,A \setminus A') \).

Observe in the previous lemma, that \( \sigma_A = \sigma_{A'} \). From the lemma it follows that for coalitions instantiating the same constellations, a coalition with fewer complying agents can achieve at least the same outcomes as a coalition with more complying agents.

We can use this observation to simplify quantification of the coalitions in question. To do this we define some particular constellations which will be useful in handling this quantification.

**Definition 5.15.** Let \( \sigma^\eta \) be a constellation which complies with the norm \( \eta \). Given a constellation \( \sigma \), the minimal complying sub-constellation of \( \sigma \), \( \sigma^\text{min} \), and the maximal complying sub-constellation of \( \sigma \), \( \sigma^\text{max} \), are defined, per role \( 1 \leq r \leq |R| \), respectively as

\[
\sigma^\text{min}_r = \max \{ 0, (\sigma_r + \sigma^\eta_r) - \Sigma_r \}
\]

\[
\sigma^\text{max}_r = \min \{ \sigma_r, \sigma^\eta_r \}
\]

The constellations \( \sigma^\text{min} \) and \( \sigma^\text{max} \) are always defined in terms of two constellations, the complying constellation \( \sigma^\eta \) and the strategizing constellation \( \sigma \). It will usually be clear from the context which these are. There will be occasions where it might not be clear which strategizing constellation we are considering. In these cases we write \( \sigma^\text{min}_A \) or \( \sigma^\text{max}_A \) to clarify. The constellations may also be defined for coalitions \( C \) (complying) and \( A \) (strategizing). In these cases, we define \( \sigma^\text{min} \) and \( \sigma^\text{max} \) in terms of the constellations the strategizing coalition instantiates, i.e., \( \sigma^\text{min}_A \) and \( \sigma^\text{max}_A \) respectively.

These particular constellations satisfies \( 0 \leq \sigma^\text{min} \leq \sigma^\text{max} \leq \sigma \), and are such that when evaluating the strategic ability of some, or all, coalitions which instantiate \( \sigma \),
they are respectively the smallest sub-constellation of $\sigma$ that must comply to $\eta$, or the largest sub-constellation of $\sigma$ that may comply to $\eta$ assuming some constellation $\sigma^c$ complies.

We formalize this in the following proposition.

**Proposition 5.16.** Given a RCGS $S$, a state $q$ in $S$, a norm $\eta$ for $S$, two coalitions $A, C \subseteq A$ where we assume $C$, and only $C$, complies to $\eta$, the following propositions hold.\(^6\) Let $X_\sigma$ denote the set of coalitions instantiating $\sigma$, i.e., $X_\sigma = \{ A \subseteq A \mid \sigma_A = \sigma \}$.

1. There is a coalition $A_{\text{min}} \in X_\sigma$ such that $\sigma_{A \cap C} = \sigma_{\text{min}}$.
2. There is a coalition $A_{\text{max}} \in X_\sigma$ such that $\sigma_{A \cap C} = \sigma_{\text{max}}$.
3. For every coalition $A \in X_\sigma$, $\sigma_{\text{min}} \leq \sigma_{A \cap C} \leq \sigma_{\text{max}}$.
4. If $S, \eta, C, q \models \langle \langle A \rangle \rangle \circ \phi$, then for every coalition $A' \in X_\sigma$ with $\sigma_{A' \cap C} \leq \sigma_{A \cap C}$ we have $S, \eta, C, q \models \langle \langle A' \rangle \rangle \circ \phi$.

**Proof.**

1. Consider an arbitrary role $r$. Order the agents enacting the role $\Sigma_r$ as $a_1, a_2, \ldots, a_{\Sigma_r}$ such that if $a_i \notin C$ and $a_j \in C$ then $i < j$. Let $A_{\text{min}}^r = \{ a_1, a_2, \ldots, a_{\sigma_r} \}$. Then $(\sigma_{A_{\text{min}}}^r)_r = \sigma_{\text{min}}^r$.

![Diagram showing coalition $A_{\text{min}}$]

Clearly, if $(\sigma_{A_{\text{min}}}^r)_r + (\sigma_C)_r \leq \Sigma_r$, then we need not include any agents which comply (and $\sigma_{\text{min}}^r = 0$). Otherwise, we need to include the overlapping agents, $a_{i+1}, \ldots, a_j$. This set contains $((\sigma_{A_{\text{min}}}^r)_r + (\sigma_C)_r) - \Sigma_r = \sigma_{\text{min}}^r$ agents.

Thus, for every role $r$ there is a coalition $A_{\text{min}} \in X_\sigma$ such that it overlaps minimally, i.e., $(\sigma_{A \cap C})_r = \sigma_{\text{min}}^r$.

2. Similarly to the above argument, we can show that the maximal number of complying agents we may choose in any role $r$ is $\sigma_{\text{max}}^r$.

![Diagram showing coalition $A_{\text{max}}$]

We can include at most $\max\{ (\sigma_A)_r, \sigma_{\text{max}}^r \}$ complying agents in our coalition. Since this holds for every role, we conclude that there is a coalition $A_{\text{max}} \in X_\sigma$ such that $\sigma_{A_{\text{max}} \cap C} = \sigma_{\text{max}}$.

3. It follows from the arguments in the previous points that no coalition in $X_\sigma$ can contain, in role $r$, fewer than $\sigma_{\text{min}}^r$ complying agents, nor more than $\sigma_{\text{max}}^r$ complying agents.

---

\(^6\)Recall that the grand constellation is denoted $\Sigma$, and the constellation a coalition $A$ instantiates is denoted $\sigma_A$, c.f., Definition 4.1.
4. We show the claim by induction on \( \Delta = (\sigma_{A \cap C} - \sigma_{A' \cap C}) \).

**Base case** (\( \Delta = 0 \)) This is basically the \((Hom)\) axiom from the previous chapter.

Suppose For any two coalitions instantiating the same coalition, and where, for each role, there is an equal number of complying agents, they two coalitions have the same strategic abilities.

**Induction step** (\( \Delta > 0 \)) Let \( \sigma = \sigma_{A \cap C} \) and \( \sigma' = \sigma_{A' \cap C} \). Since \( \sigma' < \sigma \) there must be at least some role \( r \) for which \( \sigma'_r < \sigma_r \). Then there must be some agent \( a \in A_{q,r} \cap C \) and some agent \( a' \in A'_{q,r} \) such that \( a' \notin A \cup C \).

Let \( A'' = (A' \setminus \{a'\}) \cup \{a\} \). Observe that
- \( \sigma_{A''} = \sigma_{A' \setminus \{a'\}} = \sigma_A \), and
- \( (\sigma_{A \cap C} - \sigma_{A' \cap C}) > (\sigma_{A \cap C} - \sigma_{A'' \cap C}) \).

By induction hypothesis we get that \( S, \eta, C, q \models \langle A'' \rangle \circ \phi \).

Since \( A' = (A'' \setminus \{a\}) \cup \{a'\} \), \( \rho(q,a) = \rho(q,a') \), \( a \in A'' \cap C \) and \( a' \notin A'' \cup C \), we have by Proposition 5.14 that \( S, \eta, C, q \models \langle A' \rangle \circ \phi \).

\[ \square \]

Now we can use the previous observations to reduce quantification to explicit witness coalitions \( A_{\min} \) and \( A_{\max} \) as described in the following proposition.

**Proposition 5.17.** For an arbitrary RCGS \( S \) and state \( q \) in \( S \), a norm \( \eta \) for \( S \), a constellation \( \sigma^n \) which complies with \( \eta \) instantiated by any coalition \( C \) (i.e., \( \sigma_C = \sigma^n \)), and any constellation \( \sigma \), let \( X_\sigma = \{ A \subseteq A \mid \sigma_A = \sigma \} \). Let \( A_{\min} \in X_\sigma \) be a coalition such that \( \sigma_{(A \cap C)} = \sigma_{\min} \), and \( A_{\max} \in X_\sigma \) be a coalition such that \( \sigma_{(A \cap C)} = \sigma_{\max} \). The following are true

\[
S, \eta, C, q \models \bigvee_{A \in X_\sigma} \langle A \rangle \circ \phi \iff S, \eta, C, q \models \langle A_{\min} \rangle \circ \phi \tag{5.18}
\]

\[
S, \eta, C, q \models \bigwedge_{A \in X_\sigma} \langle A \rangle \circ \phi \iff S, \eta, C, q \models \langle A_{\max} \rangle \circ \phi \tag{5.19}
\]

**Proof.** We show two cases in the proof of each proposition.

**Existential reduction (5.18)** We show both directions. One is trivial.

\[ \Rightarrow \) Suppose \( A \in X_\sigma \) witnesses the satisfaction of the disjunction (i.e., that \( S, \eta, C, q \models \langle A \rangle \circ \phi \)). From Proposition 5.16 (points 1., 3., and 4.) it follows, respectively, that
- \( A_{\min} \in X_\sigma \),
- \( \sigma_{A_{\min} \cap C} = \sigma_{\min} \leq \sigma_{A \cap C} \), and hence
- \( S, \eta, C, q \models \langle A_{\min} \rangle \circ \phi \).

\[ \Leftarrow \) Trivial since \( A_{\min} \in X_\sigma \).

We conclude that \( S, \eta, C, q \models \bigvee_{A \in X_\sigma} \langle A \rangle \circ \phi \iff S, \eta, C, q \models \langle A_{\min} \rangle \circ \phi \).

**Universal reduction (5.19)** We show both directions. One is trivial.
110 Norm compliance

⇒ ) Since $A_{\max} \in X_\sigma$, the claim follows trivially.

⇐ ) Simple from Proposition 5.16 (points 2., 3., and 4.) it follows, respectively, that

- $A_{\max} \in X_\sigma$.
- for every $A \in X_\sigma$, $\sigma_{A \cap C} \leq \sigma_{\max} = \sigma_{A_{\max} \cap C}$, and hence
- for every $A \in X_\sigma$, $S, \eta, C, q \models \langle\langle A \rangle\rangle \bigcirc \phi$.

In other words $S, \eta, C, q \models \bigwedge_{A \in X_\sigma} \langle\langle A \rangle\rangle \bigcirc \phi$.

Notice that $\sigma_A^{\min} + \sigma_{(A \setminus A)}^{\max} = \sigma_\eta$. To see this, consider the two cases (in each case, consider the cases of each $r$ independently) of when $\sigma_\eta < \sigma_{A \setminus A}$ and when it is not. In the first case $\sigma_{\max}^A$ for $\sigma_{A \setminus A}$ is $\min\{\sigma_{(A \setminus A)}, \sigma_\eta\} = \sigma_\eta$. Since it follows that $\sigma_A + \sigma_\eta < \Sigma$ we have $\sigma_{\min}^A$ for $A$ is 0. In the other case, we have $\sigma_{\max_{A \setminus A}}$ for $A$ is $\Sigma - \sigma_A$ which is less than $\sigma_\eta$. Indeed there is $\sigma_\eta - (\Sigma - \sigma_A) = \sigma_{\min}^A$. The sum of these terms are $(\Sigma - \sigma) + ((\sigma + \sigma_\eta) - \Sigma) = \sigma_\eta$.

The minimal sub-constellation of $A$ which complies to $\eta$ complements the maximal sub-constellation of $A \setminus A$ which complies with $\eta$ such that the complying sub-coalitions, in sum, constitute the entire complying constellation.

With these equivalences in place, we see that when we need to quantify over the coalitions instantiating a certain constellation, we need only select some arbitrary coalition for which the intersection with the (arbitrary) complying coalition $C$, is either $\sigma_{\min}$ (for existential quantification), or $\sigma_{\max}$ (for universal quantification).

When we now introduce a logic for reasoning about the strategic abilities of constellations under the assumption that some constellation complies to the norm, we can emulate the necessary quantification by performing some simple arithmetic comparisons on the coalitions in question. Indeed, when we now define the notions of ($\langle\langle \sigma_\eta \rangle\rangle$-compliant) $\sigma$-profiles, we do not need to identify these coalitions at all.

### 5.6.2 The logic NCHATL

The language $L_{\text{NCHATL}}$ is defined for a finite non-empty set of agents $\mathcal{A}$, a finite, non-empty set of roles $\mathcal{R}$, a state independent role assignment $\rho$, and a countable set of proposition symbols $\Pi$, is defined by the following BNF\footnote{Recall that this yields the set of constellations, $0 \leq \sigma \leq \Sigma$ as defined in Definition 4.1 from the previous chapter.}

\[
\phi ::= \quad p \mid \neg \phi \mid \phi \land \phi \mid \langle\langle A \sigma \rangle\rangle \bigcirc \phi \mid \langle\langle E \sigma \rangle\rangle \bigcirc \phi \mid \langle\langle A \sigma \rangle\rangle \Box \phi \mid \langle\langle E \sigma \rangle\rangle \Box \phi
\]

\[
\langle\langle A \sigma \rangle\rangle \phi \land \phi \mid \langle\langle E \sigma \rangle\rangle \phi \land \phi \mid \langle\sigma \rangle \phi
\]

The formulas in this language will be interpreted pointed RCGS models, together with a pair $\eta$ and $\sigma$ of a norm $\eta$ and constellation we assume complies to it. Because of the quantification issue, we need to nuance the auxiliary terminology.
The minimally complying \((\eta \uparrow \sigma^\eta)\)-compliant \(\sigma\)-profiles for role \(r\), are denoted \(P^\eta_{r,\sigma^\eta \downarrow} (q, \sigma)\) and defined as

\[
P^\eta_{r,\sigma^\eta \downarrow} (q, \sigma) = \left\{ v_r + v'_r \mid v_r \in \left\{ \mathbb{N}[^A(q,r)] \left| \left( \sum_{i \in A(q,r)} v_r(i) \right) = \sigma_r^\text{min}, \left( \sum_{i \in \eta(q,r)} v_r(i) \right) = 0 \right\} \right\}
\]

\[
v'_r \in \left\{ \mathbb{N}[^A(q,r)] \left| \left( \sum_{i \in \eta(q,r)} v'_r(i) \right) = (\sigma_r - \sigma_r^\text{min}) \right\} \right\}
\]

The maximally complying \((\eta \uparrow \sigma^\eta)\)-compliant \(\sigma\)-profile for role \(r\) is denoted \(P^\eta_{r,\sigma^\eta \uparrow} (q, \sigma)\) and is defined analogously replaces the occurrences of \(\sigma^\text{min}\) by \(\sigma^\text{max}\). The set containing all minimally complying \((\eta \uparrow \sigma^\eta)\)-compliant \(\sigma\)-profiles, and all maximally complying \((\eta \uparrow \sigma^\eta)\)-compliant \(\sigma\)-profiles are, respectively

\[
P^\eta_{\sigma^\eta \downarrow} (q, \sigma) = \{ (v_1, \ldots, v_{|R|}) \mid 1 \leq r \leq |R| : v_i \in P^\eta_{r,\sigma^\eta \downarrow} (q, \sigma) \}
\]

\[
P^\eta_{\sigma^\eta \uparrow} (q, \sigma) = \{ (v_1, \ldots, v_{|R|}) \mid 1 \leq r \leq |R| : v_i \in P^\eta_{r,\sigma^\eta \uparrow} (q, \sigma) \}
\]

These are the usual profiles for constellations (as defined in Equations 4.4 and 4.3, but such that it corresponds to a coalition \(A\) instantiating \(\sigma\) in such a way that whichever coalition \(C\) instantiates \(\sigma^\eta\), \(A\) intersects minimally, or respectively maximally, with \(C\).

Notice that for any coalition \(C\) and \(A^\text{min}\) (respectively \(A^\text{max}\)) instantiating a constellation \(\sigma\), we have \(P^\eta_{\sigma^\eta \downarrow} (q, \sigma) = P_C^\eta (q, A^\text{min})\) (respectively, \(P^\eta_{\sigma^\eta \uparrow} (q, \sigma) = P_C^\eta (q, A^\text{max})\)). From this, it is easy to verify that for every \(v_{\sigma} \in P^\eta_{\sigma^\eta \downarrow} (q, \sigma)\) and every \(v \in P^\eta_{\sigma^\eta \downarrow} (q, \Sigma)\), \((v - v_{\sigma}) \in P^\eta_{\sigma^\eta \uparrow} (q, \Sigma - \sigma)\).

That is, the minimally and maximally complying profiles are complementary. The profiles extend to strategies. The strategies which assign for every state \(q\) a profile \(v \in P^\eta_{\sigma^\eta \downarrow} (q, \sigma)\) to the constellation \(\sigma\), are denoted \(str^\eta_{\sigma^\eta \downarrow} (\sigma)\). And analogously, a strategy which assigns a profile \(v \in P^\eta_{\sigma^\eta \uparrow} (q, \sigma)\) in every state \(q\) to the constellation \(\sigma\) is denoted \(str^\eta_{\sigma^\eta \uparrow} (\sigma)\).

We need to take care that the antagonists are given the appropriate set of profiles when defining the set of outcomes. Let \(s_{\sigma}\) be a strategy. The set of computations which this strategy may yield is either

\[
\text{out} \downarrow (q, s_{\sigma}) = \{ \lambda_{s,q} \mid \text{there is a } s_{\text{ant}} \in strat \downarrow (\Sigma - \sigma), s = s_{\text{ant}} + s_{\sigma} \}
\]

\[
\text{out} \uparrow (q, s_{\sigma}) = \{ \lambda_{s,q} \mid \text{there is a } s_{\text{ant}} \in strat \uparrow (\Sigma - \sigma), s = s_{\text{ant}} + s_{\sigma} \}
\]

**Definition 5.20.** Given a RCGS \(S = (A, \mathcal{R}, \rho, Q, \Pi, \pi, A, \Delta, \delta)\) and a state \(q \in Q\), a normative system \(\eta\), and a constellation \(0 \leq \sigma^\eta \leq \Sigma\), the satisfaction relation \(\models_{\text{NCHATL}}\) (denoted \(\models\) where there is no confusion) is defined inductively as indicated below. We only give the clauses for the next-time fragment. (The remaining cases being analogous...
modifications of the satisfaction given in Definition 5.6.)

\[ S, \eta, \sigma, q \models \langle A \sigma \rangle \circ \phi \iff \text{there is } s_\sigma \in \text{strat}^{\eta}_{\sigma} \uparrow (\sigma) \text{ such that}
\]

\[ \text{for all } \lambda \in \text{out}^{\eta}_{\sigma} \downarrow (q, s_\sigma), \text{ we have } S, \eta, \sigma, \lambda[1] \models \phi \]

\[ S, \eta, \sigma, q \models \langle E \sigma \rangle \circ \phi \iff \text{there is } s_A \in \text{strat}^{\eta}_{\sigma} \downarrow (\sigma) \text{ such that}
\]

\[ \text{for all } \lambda \in \text{out}^{\eta}_{\sigma} \uparrow (q, s_\sigma), \text{ we have } S, \eta, \sigma, \lambda[1] \models \phi \]

**Example 5.21.** Consider a very simple example of a committee consisting of a hundred members in a single role \(|A| = 100\) and \(|R| = 1\) which have a decision to make: b or \(\neg b\). Say that there is some supposition that advocating \(\neg b\) is somehow immoral, or otherwise bad. Not all agents adhere to the norm of not enacting \(\neg b\), however. We illustrate the scenario in Figure 5.3. (We have omitted the outgoing edges from \(q_b\) and \(q_{\neg b}\).) Under the assumption that forty-five agents comply to the norm specifying that one ought not promote \(\neg b\) (\(\eta_{q_0} = \langle 2 \rangle\)), can sixty voters still enact \(\neg b\)? It is immediately clear that we need to quantify over the coalitions we might be referring to by “sixty voters”. There certainly is such a coalition. There is no coalition which instantiates the constellation “sixty voters” in which no agent complies to the norm. However, the minimally complying coalition consists of only five complying agents and fifty-five non-complying agents. If all these non-complying agents promote \(\neg b\) by selecting the second action we transition to \(q_{\neg b}\). That is, this model satisfies the formula \(\langle 45 \rangle \langle A \rangle \langle 60 \rangle \models \neg \phi\).

However, not every coalition which instantiates this constellation can ensure this. There are several coalitions instantiating the constellation “sixty voters” which are all unable to bring about \(\neg b\). For example, the maximally complying coalition consisting of forty complying agents, and only twenty-five non-complying agents. Hence, not every suitable coalition can achieve \(\neg b\), or in terms of a NCHATL formula, \(\langle 45 \rangle \langle A \rangle \langle 60 \rangle \models \neg \phi\).

### 5.6.3 Model checking NCHATL

We can reduce the model checking problem of NCHATL to NCRATL by a simple function. As a part of the description of the state we have a constellation \(\sigma^\eta\) assumed to be complying to the norms. The translation is relative to a corresponding initial
coalition $C$ assumed to be complying to the norm. This coalition may be any coalition which instantiates $\sigma^\eta$.

The translation is straightforward given the results that have just been presented. Every anonymous norm corresponds to a non-anonymous norm (by simply disregarding the agent argument), so a normative system for NCHATL is also a normative system for NCRATL modulo a slight change of signature.

When translating the norm compliance connective $\langle \sigma \rangle$ we simply replace the constellation by an arbitrary coalition instantiating it. The strategic connectives are quantified, so we distinguish between $\langle \langle E \sigma \rangle \rangle$ and $\langle \langle A \sigma \rangle \rangle$. In the former case, we translate this connective to $\langle \langle A_{\sigma}^{\max} \rangle \rangle$ where $A_{\sigma}^{\max} \in X_\sigma$ is arbitrary such that $\sigma_{A_{\sigma}^{\max} \cap C} = \sigma_{\max}$. Analogously in the latter case, we translate it to $\langle \langle A_{\sigma}^{\min} \sigma \rangle \rangle$ where $A_{\sigma}^{\min} \in X_\sigma$ is arbitrary such that $\sigma_{A_{\sigma}^{\min} \cap C} = \sigma_{\min}$.

**Definition 5.22.** We define a translation function $\theta : \mathcal{A} \times \mathcal{L}_\text{NCHATL} \rightarrow \mathcal{L}_\text{NCRATL}$ inductively. The coalition which is assumed to comply initially is chosen arbitrarily among those which instantiate the constellation assumed to comply.

\[
\begin{align*}
\theta_C(p) & := p \\
\theta_C(-\phi) & := -\theta_C(\phi) \\
\theta_C(\phi_1 \land \phi_2) & := \theta_C(\phi_1) \land \theta_C(\phi_2) \\
\theta_C(\langle \langle A_{\sigma} \rangle \rangle \circ \phi) & := \langle \langle A_{\sigma}^{\max} \rangle \rangle \circ \theta_C(\phi) \quad \text{where } A_{\sigma}^{\max} \in X_\sigma \text{ is arbitrary such that } \sigma_{A_{\sigma}^{\max} \cap C} = \sigma_{\max} \\
\theta_C(\langle \langle E_{\sigma} \rangle \rangle \circ \phi) & := \langle \langle A_{\sigma}^{\min} \rangle \rangle \circ \theta_C(\phi) \quad \text{where } A_{\sigma}^{\min} \in X_\sigma \text{ is arbitrary such that } \sigma_{A_{\sigma}^{\min} \cap C} = \sigma_{\min} \\
\vdots \\
\theta_C(\langle \sigma \rangle \phi) & := \theta_{C'}(\phi) \quad \text{where } C' \text{ is arbitrary such that } \sigma_{C'} = \sigma
\end{align*}
\]

**Proposition 5.23.** Let $S$ be an RCGS with a state independent role assignment, $q$ a state in $S$, $\eta$ a norm for $S$, and $\sigma^\eta$ be the constellation assumed to comply to $\eta$. Let $\phi \in \mathcal{L}_\text{NCHTL}$, and $C$ an arbitrary coalition such that $\sigma_C = \sigma^\eta$, then

\[
S, \eta, \sigma^\eta, q \models \phi \iff S, \eta, C, q \models \theta_C(\phi)
\]

The proof of this proposition is inductive on the structure of $\phi$ and the claim follows immediately for the boolean connectives, and trivially by Proposition 5.17 for the strategic connectives.

### 5.7 Summary

In this chapter we have shown that the model checking problem for *Norm compliance* RATL is tractable, despite the fact that we can reintroduce an arbitrary level of heterogeneity. That is, we can obtain an arbitrary level of heterogeneity in some initial description of our system under the assumption that all agents comply to the norms
which differentiates them. This violates the agreement found in particularly the sociological literature on (social) norms, but serves as an efficient technical tool to model heterogeneous systems.

The flavor of heterogeneity introduced in this manner, say imposing different norms to different agents in a canvas society (consisting of a single homogeneous role), is different from the kind of heterogeneity embedded in a CGS model in an important sense. By continually adding agents to the model, even though we get no exponential explosion in the model checking complexity, we get an increasing level of homogeneity as the roles get steadily more saturated.

That is, as long as we keep both the number of roles and the number of actions available to the agents in those roles fixed, the model checking problem does not explode as we add agents. However, as we add more agents, the number of agents in (at least) some role (possibly all roles) increases. These agents will be indistinguishable in the semantics and the level of heterogeneity our models capture “flattens out”. This is what permits efficient model checking also when we let the number of agents be a parameter in the complexity analysis.

As we feel we have argued in the previous chapter that strategic structures based on homogeneous roles enables a model for discussing the strategic abilities of groups of anonymous agents. Hence we feel that we have proposed a system suitable for reasoning about the (temporal) development of system consisting of a large number of agents. As the number of agents increases, we assume it likely that concrete specification of individual agents’ unique abilities becomes at least one of either unfeasible or unnecessary.

We expect that coincidence of power will be a more frequent phenomenon as the number of agents increases. Under this assumption we may very well remove the agents’ names altogether, as we argued in the previous chapter. Thus, we feel that when modeling massively populated scenarios, we may benefit from discussing constellations rather than coalitions.

We have shown that when reasoning about norm compliance by constellations, we need to take care of the quantification over coalitions satisfying these particular coalition predicates. However, the quantification can be dealt with easily by observing that there is a kind of monotonicity in the logic, and that we can compute a representative coalition for both the “for all” and “for some” form of quantification.
Chapter 6

Reason-based Preferences

6.1 Background

Reasoning about strategically homogeneous sets of agents does not entail that we are assuming that the agents are identical even though the similarity assumption is very strong. The previous chapter shows this. In the first half of Chapter 5 we discussed an initially completely homogeneous set of agents into which we then introduced differences by assuming the agents complied to different norms. However, as we assumed that some number of agents might not comply while the others did, differences appeared along the complying/non-complying distinction.

In this chapter, we will investigate how we can reason about yet another source of heterogeneity for a strategically homogeneous set of agents. They might vary in their preferences.

We will provide a logical characterization of a framework of reason-based preferences introduced by Dietrich & List in [26, 27]. The framework adds structure to the preference relations which we introduced in Chapter 1. There we defined a preference relation as an arbitrary total, transitive and reflexive relation over a set of alternatives. Rather than being completely arbitrary, Dietrich & List suggest that it may be fruitful to assume they are reason-based, a property we will define formally later.

When we postulate that an agent’s preferences are reason-based, intuitively, an agent’s preference between two objects are in some way dependent on which properties the objects have. If two objects have the exact same properties, the agent would be preferentially indifferent about the two objects. Assuming an agent’s preferences are reason-based is quite close to the assumption that if the agent is not indifferent between two objects, then she is able to provide a reason for the distinction in the way of pointing to a property (or a combination of properties) which distinguishes them.

Why such-and-such bundles of properties are better than some alternative combination does not require justification in this framework. Dietrich & List refer to this relation over property combinations as a weighing relation. This becomes the new (exogenous) primitive relation.

How an agent weighs one bundle of properties against another might have its origin in a number of different attitudes. Suppose we place two coins (say, of equal monetary value) on a table in front of two agents. The agents may very well prefer one to the other. The first agent might prefer the first coin, perhaps because it is more shiny, having (for the agent) higher aesthetic appeal, or it might be closer to her on the table
and she might prefer to expend less energy if she is permitted to take it.

The other agent might prefer the second coin. Perhaps because of its history, or because there are some intriguing scratchings on its face.

In all these cases, the agent identifies a property (or a combination of properties) which distinguishes the objects and applies some deeper weighing of these properties to form a statement of preference. This weighing is the primitive relation which, in the framework of reason-based preferences, yields the agent’s preference relation. However, generally in logic, objects are indefinitely distinguishable and hence the structures might yield any arbitrary preference relation and the added structure might not provide us with much to reason about. The second component of reason-based preferences, which addresses this issue directly, is the notion of a motivationally salient set of properties.

In the scenario of the two coins and two agents, we might say that the first agent’s motivational state is such that she emphasizes the aesthetic properties of objects, and that “being shiny” is (aesthetically) preferable (to the agent), than not being shiny.

The various sets of properties which may be motivationally salient are gathered in a set functioning as a domain of preference relations. We will discuss the construction formally in the next section.

To summarize, an agent’s weighing relation, in combination with some set of motivationally salient properties, will yield a preference relation.

Clearly, both the weighing relation and the set of motivationally salient properties might differ among two agents without this having any bearing on whether they are strategically homogeneous or not.

In recent years logic-based frameworks for reasoning about preferences have been an active line of research [56]. Most logic-based frameworks for representing and reasoning about preferences take preference relations as a fundamental concept, assuming that preferences are arbitrarily given, typically as a ranking of a set of alternatives, or represented using utilities, with little concern about how preferences are formed or where they come from.

Many logical investigations of preference require that the agents’ ranking of the alternatives is a total relation. However, this is not always the case. In the seminal article discussing dynamic preference upgrade [80], preference relations are permitted not to be complete. In the article, an agent’s preferences are altered by “suggestions” in a similar way that an agent’s knowledge might be altered by “public announcements” in dynamic epistemic logic (see e.g., [29]). If the agent takes two contradictory suggestions to heart, for example, two alternatives might become incompatible because of the “conflict” which has been instilled in the agent.

In contrast to the dynamic logic of preference upgrade [80] our preference relations are total. However, our weighing relation need not be. As will become evident, the weighing relation is very similar to a preference relation.

Recent work in rational choice theory has devoted attention to giving more “internal structure” to the notion of rationality, focusing more on the faculties of individual agents, such as their mood, mindset, and motivating reasons, see e.g., [28].

While there is some recent work on logical representations of the structure behind

---

1 At least if you assume objects are extensionally identified, as we do in this chapter. We will return to this point when we introduce the framework formally.
preferences [19, 49, 53, 62], there is no previous work on reasoning about Dietrich and List’s [27] framework in logic in general nor in modal logic in particular. We show how we can use modal logic to represent and reason about reason-based preferences. We discuss related work in more detail in Section 6.7.

This chapter is largely based on [66].

6.2 Reason-based preferences

Following [27] (with some minor differences as noted), our starting point is a countable set of properties \( P \) and a finite set \( M \subseteq 2^P \) of finite\(^2\) subsets of \( P \), which we refer to as the set of motivationally salient reasons. We define the set \( M \top := \bigcup M \), of possibly relevant properties, but we do not require that it is itself motivationally salient. Finally, we assume given a set of alternatives \( X \subseteq 2^{M \top} \).\(^3\) In [27], expressions are of the form “\( x \succsim_M y \)” for \( x, y \in X \), saying, intuitively, that \( x \) is no worse than \( y \) when \( M \) is the motivational state of the agent, the properties he chooses to focus on.

We assume that \( M \) remains fixed. Then, a motivational structure is an \( M \)-indexed collection of preference orderings \( L = (\succsim_M)_{M \in M} \) such that for each \( M \in M \), \( \succsim_M \subseteq X \times X \) is a total order on \( X \). We write \( x \prec_M y \) for \( x \succsim_M y \) and not \( y \succsim_M x \), and \( x \sim_M y \) for \( x \succsim_M y \) and \( y \succsim_M x \). The set of all motivational structures over \( X \) and \( M \) is denoted \( M \).

Given the alternatives \( X \) and possibly salient reasons \( M \), we can now define the set of views,

\[
C_{X,M} := \{x \cap M \mid x \in X, M \in M\}.
\]

This set collects all the different perspectives that an agent might have on the alternatives. In particular, given an alternative \( x \) and a motivational set \( M \), the set \( x \cap M \) is the subjective impression that the agent has of the alternative \( x \) when he is motivated by \( M \). This is not to say that the agent is unable to recognize that the object \( x \) may have some property not in \( M \), but that these properties will not influence the agent’s preference. Hence \( C_{X,M} \) is the relevant collection of objects to consider when analyzing the agent’s preferences. He does not care about the alternatives in themselves, but forms preferences based on how he views them, given a motivational state.

Following Dietrich and List, we now define the class of motivational structures that are property based. These are structures \( L = (\succsim_M)_{M \in M} \) for which there exists some relation \( \leq \subseteq C_{X,M} \times C_{X,M} \) such that for all \( x, y \in X, M \in M \), we have

\[
x \succsim_M y \iff (x \cap M) \leq (y \cap M) \quad (6.1)
\]

The relation \( \leq \) is called a weighing relation.

This picks out a set of relations on \( C_{X,M} \), namely those that encode motivational structures. Dietrich and List consider the question of what conditions we must place on such structures \( M \) in order to ensure that they are property-based. It turns out that

\(^2\)The finiteness assumption is not made in [27].

\(^3\)This definition amounts to an extensional definition of alternatives, identifying alternatives with the properties they have, and relies on the assumption that alternatives can be completely characterized by using only such properties that could potentially be considered relevant. In [27], an intentional approach is used instead. The intentional approach is more general, but for our purposes the two approaches are equivalent, and we have chosen the extensional approach for ease of presentation.
some very weak and natural conditions are all we need, suggesting that most interesting motivational structures can indeed be encoded by a single relation on the set of views.

Example 6.2 (Comparing apples and oranges). Contrary to popular belief, apples and oranges can be compared [10]. However, the result of such a comparison will depend crucially on the properties one wishes to emphasize. Using motivational structures, we can model this, showing how different properties may lead to different results. The properties we focus on in our example are $p$ ("Red") and $q$ ("Has seeds"), and we assume that both are true for apples, and false for oranges. Moreover, we assume in the structure that red is the preferred color, and that no seeds are better than seeds. Then, intuitively, if an agent is motivated by color, he will pick an apple, while if an agent is motivated by seed-presence, he will pick an orange. If he is motivated by both properties, he is indifferent.\footnote{In fact, since we will encode the structure using a weighing relation, it must be property-based, from which indifference follows automatically for this case.}

We formalize the example as follows. The set of properties is $P = \{p, q\}$, and we have the set of alternatives $X = \{A, O\}$ where $A = \{p, q\}$ and $O = \emptyset$.

Let the set of possible motivationally salient reasons $M$ be

$$M = \{\emptyset, \{p\}, \{q\}, \{p,q\}\}.$$ 

We consider the property-based motivational structure $L_{ao} = (\leq_{M})_{M \in M}$ in terms of $\leq$, a binary relation over $C_{X,M}$, defined by the graph in Figure 6.1, with the corresponding preference orderings shown on the right.

$$M = \emptyset : A \sim_{M} O$$

$$M = \{p\} : A \succ_{M} O$$

$$M = \{q\} : A \prec_{M} O$$

$$M = \{p, q\} : A \sim_{M} O$$

Figure 6.1: The relation $\leq$ and the corresponding preference orderings (an arrow from $A$ and $B$ means that $A \leq B$ and reflexive loops are omitted).

Our intuitive judgment is validated by the model; apples are preferred to oranges when $\{p\}$ is the motivational state, and oranges are preferred to apples when $\{q\}$ is motivating. Also, notice that the relation $\leq$ on views is not itself a preference ordering. It is not transitive, in particular, since $\{p\}$ and $\{q\}$ are not related. Intuitively, since there is no motivationally salient set of properties $M$ such that both $\{p\}$ and $\{q\}$ are views of objects, we do not need a direct relation between these two views. However, if we are motivated by both $p$ and $q$, then these two views of an apple are no longer in play, and we arrive at the conclusion that the fruits cannot be distinguished. So even though comparison of apples and oranges, based on real scientific properties, can lead to a distinguishing conclusion, certain other properties (under certain motivating reasons) might well validate the colloquial intuition that the two fruits are indistinguishable.
6.3 A modal logical account of reason-based preference

We now go on to define a modal logic [16] for reason-based preferences. The formulas of the logic are interpreted as statements about property-based motivational structures. This is done by viewing the weighing relation as the relation of a Kripke model with \( C_{X,M} \) as states. Formally, let the set of properties \( P \), the set of motivationally salient reasons \( M \) and the set of alternatives \( X \) be given as discussed above. The notion of a motivational model, a special kind of Kripke model, is defined as follows. Let \( L \in M \) be a motivational structure, and let \( \leq \) be a weighing relation for \( L \) (i.e., let \( \leq \) be such that Equation (6.1) holds for all \( x, y \in X, M \in M \)). The motivational model corresponding to \( L \) is a tuple \( M_L = (W,R,V) \) where \( W \) is a set of states (or worlds), \( V : W \rightarrow C_{X,M} \) is a valuation function, which we take to be a bijective correspondence between worlds and views, and the relation \( R \subseteq W \times W \) is defined such that \( wRv \) iff \( V(w) \leq V(v) \). We let \( DL \) be the class of all motivational models, i.e., all models that correspond to some motivational structure and weighing relation.

Note that the set of states in a motivational model corresponds exactly to the set \( C_{X,M} \) of views; each state represents one and only one view\(^5\). We will henceforth sometimes abuse notation and treat members of \( C_{X,M} \) as states, without confusion.

To reason about motivational models, we can now use a standard modal language \( L \), as defined by the following grammar:

\[
\phi ::= p | \neg \phi | \phi \land \psi | \Box \phi
\]

where \( p \in M_\top \). We use the usual propositional abbreviations, such as \( \phi \lor \psi \) for \( \neg (\neg \phi \land \neg \psi) \), as well as \( \Box \phi \) for \( \neg \Diamond \neg \phi \).

We also introduce the following useful abbreviations for all \( x \in X, M \subseteq M_\top \):

\[
(x \cap M)^\phi = \bigwedge_{p \in x \cap M} p \land \bigwedge_{p \in M_\top \setminus (x \cap M)} \neg p
\]

The notation reflects the intended meaning: we want to view the set of properties \( x \cap M \) as a formula \((x \cap M)^\phi \) in order to talk about it using modal logic.

The semantics of \( L \)-formulas interpreted in motivational models is now defined as usual in modal logic, giving us a notion of motivational truth and validity and a logic of reason-based preferences. Let \( \phi \in L \) and \( S \in DL \), and let \( x \in C_{X,M} \). We interpret \( \phi \) as a statement about the combination of \( S \) and \( x \). The fact that \( \phi \) is true in the context of \( S \) and \( x \), denoted \( S,x \models_{DL} \phi \), is defined formally as follows.

\[
\begin{align*}
S,x \models_{DL} p & \iff p \in V(x) \\
S,x \models_{DL} \neg \phi & \iff S,x \not\models_{DL} \phi \\
S,x \models_{DL} \phi \land \psi & \iff S,x \models_{DL} \phi \text{ and } S,x \models_{DL} \psi \\
S,x \models_{DL} \Diamond \phi & \iff \exists y \in W : xRy \text{ and } S,y \models_{DL} \phi
\end{align*}
\]

We write \( S \models_{DL} \phi \) for the fact that \( S,x \models_{DL} \phi \) for all \( x \in C_{X,M} \). Thus, \( S \models_{DL} \phi \) means that the motivational model \( S \) has the property \( \phi \). We also write \( \models_{DL} \phi \) for the fact that \( S \models_{DL} \phi \) for all \( S \in DL \). The following examples illustrate the meaning of formulas.

\(^5\)A single view might correspond to various combinations of alternatives and sets of motivating reasons, but at least one. In the example the view \( \phi \) represents the apple under \( M = \emptyset \) and the orange under any \( M \).
• “The agent (weakly) prefers $y$ to $x$ when motivated by $M$ ($x \preceq_M y$)”: 
\[
S, x \cap M \models_{DL} \Diamond (y \cap M) \phi, \text{ or}
\]

• “The agent (weakly) prefers $x$ to all other alternatives when motivated by $M$”: 
\[
S, x \cap M \models_{DL} \Box \left( \bigvee_{y \in X} (y \cap M) \phi \rightarrow \Diamond (x \cap M) \phi \right).
\]

• “The agent (weakly) prefers $x$ under the view of $M$ to all other comparable views”: 
\[
S, x \cap M \models_{DL} \Box \Diamond (x \cap M) \phi.
\]

• “The agent (weakly) prefers the view of $x$ under $M$ to all other views”: 
\[
S \models_{DL} \Box \Diamond (x \cap M) \phi.
\]

Notice that in the above, the second point requires explicitly quantifying over alternatives. The reason is that a view of $x$ under $M$ can well coincide with a view of some distinct alternative $y$ under some other motivating set $M'$. This is an interesting feature of motivational logic, and it will play a major role in the technical work that is to follow. On the conceptual side, it demonstrates that property-based motivational structures are both subtle and expressive, and that a treatment by way of formal logic is appropriate. We notice, in particular, that the third formula expresses something stronger than the second, namely that as far as the agent is concerned, he has found a view of an alternative such that even if we assume that he has local knowledge of the weighing relation that induces all the different preference relations, he prefers his current view. That is, he prefers both the alternative and his choice of perspective, compared to all other views with which it is comparable. The final formula is stronger still, since global truth expresses something stronger about the view of $x$ under motivating reasons $M$, namely that the alternative $x$ under motivation $M$ is (weakly) preferred to any alternative $y$ under any motivating reason $M'$, necessitating, in particular, that it can be compared to all these views. As illustrated by the example of apples and oranges, this can by no means be taken for granted.

Example 6.3 (The truth about apples and oranges). We continue the example of apples and oranges introduced in Section 6.2. Let $M_{ao} = M_{lao}$ be the motivational model corresponding to $L_{ao}$ and let’s consider the three motivating reasons $R = \{p\}$ (cares about color), $S = \{q\}$ (cares about seed-presence), $RS = \{p, q\}$ (cares about everything). The following are some formulas which are true in $M_{ao}$.

• “When color, and only color, is motivating and red is preferred (as in this example), apples are strictly preferred to oranges”: 
\[
M_{ao} \models_{DL} ((O \cap R) \phi \rightarrow \Diamond (A \cap R) \phi) \land \neg ((A \cap R) \phi \rightarrow \Diamond (O \cap R) \phi)
\]

• “If the agent is only motivated by color, then she strictly prefer apples in her current motivational state over any other fruit in any other motivational state”: 
\[
M_{ao} \models_{DL} (A \cap R) \phi \rightarrow \Box (A \cap R) \phi
\]
6.3 A modal logical account of reason-based preference

- “If you are motivated by seed-presence, you prefer your view of an orange to your view of an apple”:
  \[ M_{ao} \models_{DL} (A \cap S)^\phi \rightarrow \Diamond (O \cap S)^\phi \]

- “The agent weighs the view of an apple under the motivation of color (weakly) greater than the view of an orange under the motivation of color”:
  \[ M_{ao} \models_{DL} (O \cap S)^\phi \rightarrow \Diamond (A \cap R)^\phi \]

- “An orange is an orange, regardless of your point of view”:
  \[ M_{ao} \models_{DL} (O \cap S)^\phi \leftrightarrow (O \cap R)^\phi \]

- “Being motivated by seed-presence the agent (weakly) prefers the orange to the apple, while at the same time the orange (indeed, in any motivational state) is strictly less preferred than the apple when motivated by color”:
  \[ M_{ao} \models_{DL} (A \cap S)^\phi \rightarrow \Diamond ((O \cap S)^\phi \land \Box\Diamond (A \cap R)^\phi) \]

- “Regardless of which view the agent considers, permitting the agent to alter motivational state we can obtain the view that apples, when motivated by color, are strictly preferred to all other views (the same does not hold for the orange)”:
  \[ M_{ao} \models_{DL} \Diamond \Box\Diamond (A \cap R)^\phi \]

The last point encodes the fact that a color-based view of an apple is alone in being considered the best option, according to the weighing relation. Apples are such that we prefer to (strictly) prefer them, irrespectively of what motivates us. What to make of this? In some sense, it seems to suggest that apples are better than oranges, but this appears counter-intuitive. Luckily, we can analyze the situation more fully using the universal modality, which we introduce in the next section. It will allow us to express more properties of the weighing relation and its relationship to the preference orderings it induces, culminating in the promised results that property-based motivational structures themselves can be characterized logically, in terms of the modal truths admitted by the weighing relations characterizing them.

We think that the example of apples and oranges shows clearly that motivational logic can express interesting properties of property-based preferences, and, in particular, properties that have to do with the weighing relation as an aggregated relation, emerging from the given property-based collection of preference relations. Reasoning in this logic involves solving the validity problem (or, equivalently, the satisfiability problem): given a formula \( \phi \in \mathcal{L} \), does \( \models_{DL} \phi \) hold? In the next section we provide one solution to this problem, obtained by translating the problem to a validity problem of a standard, well-known, modal logic.
6.4 The characterization

In this section we provide a translation of the validity problem for the logic developed in the previous section, to a validity problem of the modal logic KT with universal modality. In more detail, we provide a translation of formulas $\phi \in \mathcal{L}$ of our logic into KT with universal modality formulas $\phi'$ such that $\models \phi'$ iff $\models_{DL} \phi$, where $\models$ denotes validity in KT with universal modality.

Let $\mathcal{L}^G$ be the standard modal language enhanced with universal modality, with the set of properties $\mathcal{P}$ as primitive propositions, as defined by the following grammar:

$$
\phi \ ::= \ p \mid \neg \phi \mid \phi \land \phi \mid \Diamond \phi \mid A\phi,
$$

where $p \in \mathcal{P}$. The universal modality is $A$, and its dual is $E$, i.e., $E\phi = \neg A\neg \phi$.

We will use standard Kripke models with reflexive relations to interpret this language. Formally, a model is a tuple $S = (W, R, V)$ where $W$ is a set of states (or worlds), $R \subseteq W \times W$ is a reflexive relation and $V : W \rightarrow 2^\mathcal{P}$ is a valuation function assigning each state to a set of properties. We let $S$ denote all such models.

We use $S, w \models \phi$ to denote the fact that a formula $\phi \in \mathcal{L}^G$ is true in state $w$ of $S \in S$, formally:

$$
S, w \models p \iff p \in V(w)
$$

$$
S, w \models \neg \phi \iff S, w \not\models \phi
$$

$$
S, w \models \phi \land \psi \iff S, w \models \phi \text{ and } S, w \models \psi
$$

$$
S, w \models \Diamond \phi \iff \exists v \in W : wRv \text{ and } S, v \models \phi
$$

$$
S, w \models A\phi \iff \forall v \in W : S, v \models \phi
$$

Global ($\models \phi$) and model-level ($S \models \phi$) validity is defined as for $\models_{DL}$. We also write $\Gamma \models \phi$ where $\Gamma \subseteq \mathcal{L}^G$, to denote the fact that $S, w \models \phi$ whenever $S, w \models \Gamma$ for all $S$ and $w$ (where $S, w \models \Gamma$ iff $S, w \models \psi$ for all $\psi \in \Gamma$). The fact that we require the relations to be reflexive, means that this is the standard modal logic KT, with universal modality. Note that we use $\models$ for this logic, and $\models_{DL}$ for our motivational logic introduced in the previous section. We define the following modality, for all $M \in \mathcal{M}$

$$
[M]\phi \iff \phi \land \bigwedge_{M' \subseteq M} \left( \left( \bigwedge_{p \in M'} p \land \bigwedge_{p \in M \setminus M'} \neg p \right) \rightarrow A \left( \left( \bigwedge_{p \in M'} p \land \bigwedge_{p \in M \setminus M'} \neg p \right) \rightarrow \phi \right) \right)
$$

We also define the dual $\langle M \rangle \phi = \neg [M] \neg \phi$.\footnote{Let us also remark that the universal modality can be defined by $A\phi = [\emptyset] \phi$.}

Unwinding the definition above, we get, for all $S \in S$, $\phi \in \mathcal{L}$, $M \in \mathcal{M}$ and all $w \in W$:

$$
S, w \models [M] \phi \iff \text{for all } u \in W : \text{if } V(w) \cap M = V(u) \cap M, \text{ then } M, u \models \phi
$$

The universal (or global) modality is a standard tool for internalizing global satisfiability, and it gives us increased expressivity. In particular, we can quantify over alternatives that are in agreement under $M$, for instance by stating that an agent (weakly) prefers $y$ under the view $M$, to all other objects under any view ($M'$) which have at least the properties as the $x/M$-view has. Or, more simply: $S \models (x \cap M) \phi \rightarrow [M] \Diamond (y \cap M) \phi$.\footnote{Let us also remark that the universal modality can be defined by $A\phi = [\emptyset] \phi$.}
Example 6.4 (Apples and oranges, equally good once again). Using the new modality, we are able to state a further truth about apples and oranges. It shows, in particular, that our judgment to the effect that we prefer to (strictly) prefer apples, while correct, is only able to provide an incomplete account of the weighing relation. For while the last formula of Section 6.3 seemed to suggest that an apple is the better fruit, and that caring only about color is the best way for us to evaluate the matter, the following formula expresses the sense in which this conclusion can appear rather too optimistic about both apples and color.

- “Regardless of which view you consider, even if you are motivated by color, there is a view, indistinguishable from your current view (w.r.t. color), such that you (weakly) prefer an orange”:

\[ M_{ao} \models \langle R \rangle \Diamond (O \cap R)^{\phi} \]

In light of this, it seems we can really only ask again: which is better, an apple or an orange? The question, moreover, appears to be a non-trivial one, and we will not attempt to provide definite answers here, beyond noting that at this point, we have the machinery to express a more specific sense in which they are – after all – equally good, even in the presence of motivating sets that distinguish them. We hope to return to apples and oranges in future work, however, and to apply logical tools to clarify in more depth how a weighing relation encoding a collection of individual preference relations is to be understood in the context of rational choice.

We now have the machinery to state our goal in more detail: we want a finite collection of formulas \( T_M \subseteq \mathcal{L}^G \) such that for all \( \phi \in \mathcal{L} \), we have

\[ \models \bigwedge T_M \rightarrow \phi \iff \models_{DL} \phi. \]

In the following, we demonstrate that the theory consisting of the union of all the sets of formulas shown in Figure 6.2 will do this job. We will refer to this theory as \( T_M \). Notice that as all these formulas appear inside the scope of a universal modality; local and global satisfaction of these formulas coincide. We have, in particular, for all models \( S \in \mathcal{S} \) and all \( w \in W \), \( S, w \models \bigwedge T_M \iff S \models \bigwedge T_M \). It also follows that \( T_M \models \phi \iff \models \bigwedge T_M \rightarrow \phi \), which is an observation we will use in the proof of our main result.

Intuitively, the formula \( A_1 \) ensures that every world instantiates a view. Since we are working with arbitrary reflexive frames, a view can be instantiated by many different worlds. To ensure that all these behave in the same way, we use the set of formulas \( A_2 \) saying that any two worlds that agree on all variables from \( M^\top \) – such that they represent the same view – must also agree about successors, i.e., they must agree on what the preferred views are. To ensure that our models encode a preference relation for each motivating set, we use the formulas \( A_3 - A_4 \). \( A_3 \) ensures totality, while \( A_4 \) encodes transitivity, but parameterised by motivating sets, so that if you restrict attention to one \( M \) and collapse alternatives under agreement on \( M \) (possible using the modality \( [M] \)), our models become total orders.\(^8\)

\(^7\)Note that \( M \) is a part of the name and not a parameter.

\(^8\)This does not mean that the frame itself is required to be a collection of total orders, a phenomenon that is discussed further in Dietrich and List [26], where conditions are also stipulated ensuring that encodings of motivational systems must themselves be total orders on views. It is shown, in particular, that this happens just in case the views are completely instantiated, i.e., for every view there is an alternative that is identical to it. Further exploration of this notion in logical terms is a topic we intend to investigate in future work.
The most interesting formulas in the theory are collected in A5. Intuitively, they ensure that worlds are only connected under R when they represent compatible views, that is, when they represent views with respect to the same M. By A1, every world is required to represent a view, but it is quite possible that the same world can represent several different views simultaneously. So in order to make a judgment regarding compatibility, it is not enough to say that a view under M can only point to other views under M. It is quite possible that w represents a view under M, but is related to a world v which represents a view under M’ that does not correspond to any view under M. Then, however, we must ensure that w and v are compatible on some M”, i.e., that w also represents a view under an M” for which there is some view represented by v. This, in words, is what A5 says, in modal language.9

Now, in light of the informal presentation just given, it should be clear that the following proposition holds.

**Proposition 6.5.** For every motivational model S ∈ DL, we have \( S \models \bigwedge T_M \).

**Proof.** Let \( L = (\leq_M)_{M \in M} \) be a motivational structure for which \( \leq \) witnesses that it is property based, and \( M_L = \langle W, R, V \rangle \) be the motivational model in DL corresponding to L. Then \( W = C_{X,M}, V = id \), and \( wRw' \) iff \( w \leq w' \).

A1: \( M_L \) satisfies A1 iff every world \( (y \cap M') \in W \) satisfies \( \bigvee_{x \in X} \bigvee_{M' \in M} (x \cap M')^\phi \). In the disjunctions when \( x = y \) and \( M = M' \), the formula \( (x \cap M')^\phi \) is satisfied.

A2: Let \( (y \cap M') \) be an arbitrary world. We need to show that \( M_L, (y \cap M') \models \bigwedge (x \cap M)^\phi \rightarrow [M_+] \bigwedge (x \cap M)^\phi \). Assume the antecedent. \([M_+] \phi \) in a given world iff \( \phi \) in

---

9To further clarify the meaning of A5, it might help to notice that the boxed sub-formulas \( \bigwedge (x \cap M)^\phi \rightarrow \bigvee_{x \in X} (z \cap M')^\phi \) can be everywhere replaced by formulas of the form \( \bigwedge (x \cap M)^\phi \land \bigvee_{x \in X} (z \cap M')^\phi \). The effect would be the same; we would still require \( (x \cap M) = (z \cap M') \) for some \( z \in X, M' \in M \), which is exactly what we need.
this world and every world which agrees with the properties in $M_\top$. There is only
one world in $W = C_{X,M}$ which agrees on all properties, i.e., $y \cap M'$. Hence the
claim follows.

A3: Let $x, y \in X$ and $M \in M$ be arbitrary. We have either $x \not\preceq_M y$ or $y \not\preceq_M x$. Assume
WLOG that $x \not\preceq_M y$. By Equation (6.1) $(x \cap M) \leq (y \cap M)$ and hence $(x \cap M)R(y \cap M)$. The outermost connective
is the existential modality $E$. Let this pick the world $(x \cap M)$ and it is easy to see that $M_L, (x \cap M) \models ((x \cap M) \phi \land \diamond (y \cap M) \phi) \lor ((y \cap M) \phi \land \diamond (x \cap M) \phi)$.

A4: The proof of A4 (transitivity of the preference relations) is similar to that of A3 (totality of the preference relations). Let $x, y, z \in X$ and $M \in M$ be arbitrary
and assume that an arbitrary world satisfies the antecedent. We can identify that
world by $(x \cap M)$ because it satisfies (the first conjunct in the antecedent). We
need to show that it follows that $M_L, (x \cap M) \models \diamond (z \cap M) \phi$, but this follows since
$(x \cap M) \leq (z \cap M)$ by Equation (6.1).

A5: Let $w$ be an arbitrary world satisfying the antecedent. Then it is some view $w \in C_{X,M}$ s.t. $wR(x \cap M)$, that is $(x \cap M)$ is some view which $w$ stands
in relation with. From Equation (6.1) there are $y$ and $z$ in $X$ and some $M' \in M$ such that $w = y \cap M'$
and $x \cap M = z \cap M'$. The first conjunct in the consequent is clearly satisfied in $w$.
Further, all successor worlds which satisfy exactly the properties in $x \cap M$ (which
is only $x \cap M$, i.e., $z \cap M'$), also satisfy $z \cap M'$.

This, essentially, establishes one direction of our desired result. To show the other
direction, we will partition the models satisfying $T_M$ into classes, each containing
a member of DL, and show that all members of such a class are truth-equivalent on $L$.
To do this we will use the translation $\rho : S \rightarrow M$, defined by $\rho(\langle W,R,V \rangle) = (\preceq_M)_{M \in M}$
where

- for each $M \in M$, for each $x, y \in X$, $x \not\preceq_M y \Leftrightarrow S \models (x \cap M) \phi \rightarrow \diamond (y \cap M) \phi$.

Let $S_{T_M}$ denote the class of all models satisfying $\land T_M$. In light of Proposition 6.5,
we have that for all property-based $L \in M$, the corresponding encoding $M_L \in DL$ is in
$S_{T_M}$, i.e., we have $DL \subseteq S_{T_M}$. Moreover, it is not hard to see that all property-based
structures are included in $\rho(S_{T_M})$; applying $\rho$ to a member of DL obviously gives back
the corresponding motivational structure, i.e., for all $L \in DL$, we have $\rho(M_L) = L$.

The first step towards our main result is to establish that $\rho(S_{T_M})$ characterizes
the property-based motivational structures, and to do this, it remains to show that $\rho(S_{T_M})$
only contains such structures. This, however, is ensured by the theory $T_M$, as detailed
in the following proposition.

**Proposition 6.6.** For all models $S \in S_{T_M}$, we have that $L = \rho(S)$ is a property-based
motivational structure.

**Proof.** Let $L = \rho(S)$ for some arbitrary $S \in S_{T_M}$ and consider arbitrary $M \in M$. We
show that $\preceq_M \subseteq X \times X$ defined by $\rho$ is a total order on $X$ and that it is property-based.
Reflexivity is immediate since $S$ is a model on a reflexive frame.
Totality: Consider arbitrary \( x, y \in X, M \in \mathcal{M} \). We know that \( S \models A3 \). This allows us to assume, wlog, that there is \( w \in W \) such that \( S, w \models (x \wedge M)^\varphi \wedge (y \wedge M)^\varphi \). It follows from A2 that we have \( S, u \models (\wedge \wedge M)^\varphi \) for all \( u \in W \) such that \( V(u) \wedge M = V(w) \wedge M \), meaning, in particular, that we have \( S, u \models (\wedge \wedge M)^\varphi \) for all \( u \in W \) such that \( S, u \models (x \wedge M)^\varphi \). So \( S \models (x \wedge M)^\varphi \to (\wedge \wedge M)^\varphi \) and by definition of \( \rho \) we get \( x \preceq_M y \in L \), and we are done.

Transitivity: Consider arbitrary \( x, y, z \in X, M \in \mathcal{M} \) such that \( x \preceq_M y \preceq_M z \) in \( L \). This means, by definition of \( \rho \), that we have \( S \models (x \wedge M)^\varphi \to (\wedge \wedge M)^\varphi \) and \( S \models (y \wedge M)^\varphi \to (\wedge \wedge M)^\varphi \). By A3, we get that there must be \( w \in W \) such that \( S, w \models (x \wedge M)^\varphi \). Then, by definition of \( \rho \), we get \( S, w \models (\wedge \wedge M)^\varphi \). Let \( v \in W \) be such that \( wRv \) and \( S, v \models (y \wedge M)^\varphi \). Then, again by definition of \( \rho \), we get \( S, v \models (\wedge \wedge M)^\varphi \), meaning we have \( S, w \models (y \wedge M)^\varphi \). By A4 we get \( S, w \models (\wedge \wedge M)^\varphi \), and by A2 it follows that for all \( u \in W \) such that \( S, u \models (x \wedge M)^\varphi \), we have \( S, u \models (\wedge \wedge M)^\varphi \), meaning \( x \preceq_M z \) by definition of \( \rho \).

Property-basedness: What remains is to show that \( L \) is property-based. To this end, simply define \( \subseteq \subseteq C_{X, M} \times C_{X, M} \) by \( x \wedge M \leq y \wedge M \) iff \( S \models (x \wedge M)^\varphi \to (y \wedge M)^\varphi \), i.e., completely mirroring the definition of \( \rho \). If this is well-defined, then we are done, so assume towards contradiction that it is not. Then there must be \( x, y, x', y' \in X, M, M' \in \mathcal{M} \) with \( x \wedge M = x' \wedge M' \) and \( y \wedge M = y' \wedge M' \) such that \( S \models (x \wedge M)^\varphi \to (\wedge \wedge M)^\varphi \) and \( S \not\models (x' \wedge M')^\varphi \to (\wedge \wedge M')^\varphi \). But from \( x \wedge M = x' \wedge M' \) and \( y \wedge M = y' \wedge M' \) we get \( (x \wedge M)^\varphi = (x' \wedge M')^\varphi \) and \( (y \wedge M)^\varphi = (y' \wedge M')^\varphi \), and a contradiction.

For each \( S \in S_{M_\top} \), we let \( S_\rho \in \mathcal{DL} \) denote the motivational model corresponding to the motivational structure \( \rho(S) \). This partitions \( S_{M_\top} \) into classes \( S_{M_\top} = \bigcup_{M \in \mathcal{M}} \{ S \in S_{M_\top} \mid S_\rho = M \} \). We will show that all members of all \( S \in S_{M_\top} \) are truth-equivalent on \( L \), i.e., that they satisfy the same formulas. We do this using a bisimulation modulo \( M_\top \), defined as a relation \( \beta \subseteq W \times W' \) such that

**Preservation:** for all \( w \in W \), if \( w \beta w' \), then \( V(w) \cap M_\top = V'(w') \cap M_\top \)

**Forth:** for all \( w \in W \) and all \( w' \in W' \) s.t. \( w \beta w' \), if \( wRv \), then there is \( v' \in W' \) s.t. \( v \beta v' \) and \( w'R'v' \)

**Back:** for all \( w' \in W' \) and all \( w \in W \) s.t. \( w \beta w' \), if \( w'R'v' \), then there is \( v \in W \) s.t. \( v \beta v' \) and \( wRv \)

We need the following proposition, the proof of which we omit as it is a simple variation of the standard proof.\(^{10}\)

\(^{10}\)It is just a slight variation of the standard truth-preservation result for bisimulations, adapted to account for the fact that preservation is only required to hold for symbols in \( M_\top \). Formally, a simple induction on the complexity of formulas will do, taking note also of the fact that formulas from \( L \) only involve variables from \( M_\top \).
Proposition 6.7. If $S, S'$ are bisimilar modulo $M_\top$, then for all $\phi \in \mathcal{L}$ we have $S \models \phi$ iff $S' \models \phi$.

Next, to show that models in the same class from $\mathcal{S}_{TM}$ are truth-equivalent, we only need to show that they are bisimilar modulo $M_\top$. This is the content of the following lemma.

Lemma 6.8. For all $S \in \mathcal{S}_{TM}$ and all $S, S' \in S$, we have that $S$ and $S'$ are bisimilar modulo $M_\top$.

Proof. Consider the relation $\beta \subseteq W \times W'$ defined by taking $w \beta w'$ iff $V(w) \cap M_\top = V'(w') \cap M_\top$. Obviously, $\beta$ satisfies preservation modulo $M_\top$. To complete the proof we show that it also satisfies “forth”, the proof for “back” being symmetric. Assume that $wRv$ and $w \beta w'$. Since $S$ satisfies $A1$ we know that there is $x \in X, M \in M$ such that $V(v) \cap M_\top = x \cap M$. We have, in particular, $S, w \models \Diamond (x \cap M)^\theta$. This means, by $A5$ and modus ponens, that there are $y, z \in X, M' \in M$ such that $S, w \models (y \cap M')^\theta$ and $S, v \models (z \cap M')^\theta$, and, in particular, that we have $V(w) \cap M_\top = y \cap M'$ and $V(v) \cap M_\top = z \cap M'$. Since $S, w \models \Diamond (z \cap M')^\theta$, it follows by $A2$ that we have $S, u \models \Diamond (z \cap M')^\theta$ for all $u$ such that $V(u) \cap M_\top = V(w) \cap M_\top = y \cap M'$, i.e., for all $u$ such that $S, u \models (y \cap M')^\theta$. So $S \models (y \cap M')^\theta \rightarrow \Diamond (z \cap M')^\theta$, meaning that $y \not\sim_{M'} z$ in the motivational structure $L = \rho(S)$. Since $w \beta w'$, we know that $V'(w') = y \cap M'$. On the other hand, since $S, S' \in S$, we have $L = \rho(S) = \rho(S')$, and then, since $y \not\sim_{M'} z$ in $L$, we must have $S' \models (y \cap M')^\theta \rightarrow \Diamond (z \cap M')^\theta$. In particular, we have $S', w' \models (y \cap M')^\theta \rightarrow \Diamond (z \cap M')^\theta$, and since $S', w' \models (y \cap M')^\theta$, this means that $S', w' \models \Diamond (z \cap M')^\theta$. From this we conclude that there is some $v' \in W'$ with $w'R'v'$ such that $S', v' \models (z \cap M')^\theta$, which means that $V'(v') \cap M_\top = z \cap M' = V(v) \cap M_\top$. By definition of $\beta$, we then have $v \beta v'$ as desired.

In light of this lemma, the main theorem follows easily.

Theorem 6.9. For every formula $\phi \in \mathcal{L}$, we have $\models_{DL} \phi$ iff $T_M \models \phi$

Proof. We have two cases synthesizing what we have shown earlier. Part of the setup can be illustrated as in Figure 6.3.

$\Leftarrow$) Assume that $T_M \models \phi$. By Proposition 6.5, we have $S \models \land T_M$ for all $S \in DL$. So we get $S \models \phi$ for all $S \in DL$ and then $\models_{DL} \phi$ as desired.

$\Rightarrow$) By Lemma 6.8, we know that for all $S \in \mathcal{S}_{TM}$, all $S, S' \in S$ are bisimilar modulo $M_\top$, meaning, by Proposition 6.7, that $S \models \phi$ iff $S' \models \phi$ for all $\phi \in \mathcal{L}$. By Proposition 6.6, every $S \in \mathcal{S}_{TM}$ contains a unique $M \in DL$, and it follows that if we have $M \models \phi$ for all $M \in DL$, we also have $S \models \phi$ for all $S \in \mathcal{S}_{TM}$. By definition of $\mathcal{S}_{TM}$ it follows that for all models $S$, if $S \models \land T_M$, then $S \models \phi$, and we are done.

$\Box$
6.5 The multi-agent case

In this section we look at how motivational logic can be applied in a multi-agent setting. First, let us point out that there is an interesting notion of agency that can be modeled using motivational sets. It seems, in particular, that agents occasionally disagree only because they have a different perspective on the matter at hand. They share the same weighing relation, yet are motivated by different properties. In such cases, a notion of inter-agent relations emerges in motivational logic as soon as we map agents to motivating sets. With respect to such an interpretation of agents’ disagreeing, the weighing relation, being a global relation that represents all individual preference, becomes even more important. Also, focus naturally shifts from the case of talking about alternatives, and what agents prefer individually, to talking about views, agents’ perspectives on objects, and how these are related. We have already seen that motivational logic allows us to express non-trivial properties in this regard, and we think it deserves further investigation.

We can also arrive at a model of agency more explicitly, however, by generalizing the notion of motivational structures. To sketch this development, let us consider an agent-indexed collections of such structures, $\succeq^a_M$ for $a \in \mathcal{A}$, $M \in \mathcal{M}$, where $\mathcal{A}$ is some finite collection of agents. Extending the definition of a motivational model to obtain multi-agent motivational models is straightforward as long as $\succeq^a_M$ is property based for each $a \in \mathcal{A}$ and as long as all agents have the same motivational sets available. Then we can simply collect, for each $a \in \mathcal{A}$, the relations $R_a$ used to characterize $L_a = (\succeq^a_M)_{M \in \mathcal{M}}$ in the motivational model $M_{L_a}$. We obtain a multi-agent model $\langle W, (R_a)_{a \in \mathcal{A}}, V \rangle$ where $W$ and $V$ are defined as in $M_{L_a}$.

As long as the number of agents is finite, the characterization in terms of theories $A1$-$A5$ generalizes similarly, as soon as we alter the language by replacing the modal connective $\Diamond$ with an $\mathcal{A}$-indexed collection of connectives $\Diamond_a$ for each $a \in \mathcal{A}$ (and adapt the definition of the satisfaction accordingly). Then, a characterization of multi-agent motivational models in terms of finite multi-agent theories on reflexive frames can be provided by simply requiring the corresponding instance of $A1$-$A5$ to hold for...

---

11One might interpret this in other ways as well; the agents might contribute only one preference-relation each, with a corresponding motivational set which details their perception of what matters, or, even, their different beliefs about what properties the alternatives really have. From this collection, provided it is property-based, we can aggregate the weighing relation and reason about relationships between the views of different agents.
all modalities $\Diamond_a$, with $a \in A$.

What interesting properties can we express and study using this extended framework for motivational reasoning? Let us note that we get a notion of unanimous consensus which applies relative to a vector of motivational sets, $\vec{M} = \langle M_1, \ldots, M_{|A|} \rangle$, and holds just in case all agents agree on some alternative $x$ being a best alternative under their view in $\vec{M}$. Semantically, we collect all alternatives that are preferred in this sense, under $\vec{M}$, in the following set.

$$ p(\vec{M}) = \bigcap_{M_i \in \vec{M}} t(\nimp_{M_i}), \text{ where } t(\nimp_{M_i}) = \{ x \in X \mid \forall y \in X : y \nimp_{M_i} x \} $$

A logical formulation of $x \in t(\nimp_{M_i})$ is obtained as a simple agent-indexation of which the characterization is for the single-agent case, as follows.

$$ S \models (x \cap M_i)^{\phi} \to \Box_i \left( \bigvee_{y \in X} (y \cap M_i)^{\phi} \to \Diamond_i (x \cap M_i)^{\phi} \right) $$

That is, $x \cap M_i$ is regarded as a best alternative by agent $i$ if for every other view the agent might prefer, we have that if it is a view that $i$ has on some alternative when motivated by $M_i$, then $x \cap M_i$ is preferred to this view as well (that is, they are equally good). We will call this a position consensus, since it depends on the motivational vector, the motivational position the agents find themselves in when attempting to decide on a best alternative.

Logically, a position consensus for $x$ at $\vec{M}$ can be characterized as follows.

$$ S \models \bigwedge_{M_i \in \vec{M}} \left( (x \cap M_i)^{\phi} \to \Box_i \left( \bigvee_{y \in X} (y \cap M_i)^{\phi} \to \Diamond_i (x \cap M_i)^{\phi} \right) \right) $$

A position consensus represents a (combined) motivational state where agents may unanimously decide on an alternative that they all consider to be a best choice.

The other notion which seems natural to characterize is stronger, and encodes the state of fundamental agreement (on the level of the weighing relation). Fundamental agreement does not depend on the position, and two agents, $a$ and $b$, are in fundamental agreement iff

$$ S \models \bigwedge_{M_i \in M} \bigwedge_{x \in X} (\Diamond_a (x \cap M)^{\phi} \leftrightarrow \Diamond_b (x \cap M)^{\phi}) $$

All agents are in fundamental agreement if each pair of agents are. When some agents are not in fundamental agreement, we say they are in fundamental disagreement.

**Example 6.10** (Consensus concerning fruit among agents who fundamentally disagree). We return to oranges and apples, but now also introduce two agents. We consider the scenario as shown in Figure 6.4 and in the two graphs of Figure 6.5, giving the weighing relation for agent $a$ on the left and agent $b$ on the right. Assume that $M_a = \{ p, q \}$ and $M_b = \{ p \}$, that is, agent $a$ takes both properties into account when ordering the alternatives, but agent $b$ only considers whether the objects are red or not. Clearly we do not have fundamental agreement in this case. There is a state, e.g., $\emptyset$,
from which there is a set of motivational reasons $M = \{p\}$ for which the two agents fail to have the same successors:

$$S, \emptyset \nvdash \Box_a (A \cap \{p\})^\phi \quad \text{and} \quad S, \emptyset \models \Diamond_b (A \cap \{p\})^\phi$$

It can also be easily verified that the orange (and only the orange) is a position consensus in this case among the two agents. In every world which is a view of an object under the view $M_a$ (only $\emptyset$ and $\{p\}$), there is an $R_a$-edge to a world representing the view of the orange under $M_a$ (that is $\emptyset$). Furthermore, every successor state from these views which themselves are $M_a$-views have an $R_a$-edge to the $M_a$-view of the orange ($\emptyset$). Similarly for agent $b$. Formally,

$$S \models (O \cap M_a)^\phi \rightarrow \Box_a \left( \bigvee_{y \in X} (y \cap M_a)^\phi \rightarrow \Box_a (O \cap M_a)^\phi \right)$$

$$\land (O \cap M_b)^\phi \rightarrow \Box_b \left( \bigvee_{y \in X} (y \cap M_b)^\phi \rightarrow \Box_b (O \cap M_b)^\phi \right)$$

6.6 Finding the closest consensus

In this section we analyze the situation where agents disagree only superficially, i.e., they have a common weighing relation and the fact that they have different motivating sets is the only reason why there is no consensus. What is the least change in their motivational states that leads to agreement? Let us define a consensus position as a vector which supports agreement as defined earlier, denoted $C(M)$.

**Definition 6.11.** For a multi-agent motivational model $S$, a position vector $M$ supports a consensus, denoted $C(M)$ if, and only if, $p(M) \neq \emptyset$.

A reasonable definition of distance between a motivational vector and a consensus position can be based on the idea that the agents should be required to adopt or abandon as few new reasons as possible. Viewing their motivating sets, say $M_a$ and $M_b$, as bit strings of length $|M_a|$ we can see that such a formalization of distance can be arrived at by considering the Hamming distance between these sets represented as bit strings. The smallest change which leads to consensus is then defined formally as follows.
6.6 Finding the closest consensus

Definition 6.12. For a set of agents with common weighing relation and views given by the vector \( M = (M_a)_{a \in A} \), an individual closest consensus position is a vector \( M' = (M'_a)_{a \in A} \) such that \( C(M') \) and for all motivating \( M'' = (M''_a)_{a \in A} \) with \( C(M'') \), we have

\[
\max \{ |M_a \setminus M'_a| + |M'_a \setminus M_a| \mid a \in A \} \leq \max \{ |M_a \setminus M''_a| + |M''_a \setminus M_a| \mid a \in A \}
\]

The definition asks us to consider how many changes we need to make to the motivating set of the agent that needs to change his perspective the most. In order to find the closest consensus under this conception of distance, observe that we are upper-bounded by the “radius” of the set of motivating sets. That is, the \( M \in \mathcal{M} \) which has the shortest distance to all \( M_i \) in \( \mathcal{M} \), solves the question.

Rather than offering a concrete algorithm for finding the closest consensus – which should in any case be adapted to reflect the relative difference in size between the motivating sets, the number of different alternatives, and the number of properties of alternatives currently considered relevant – we make some more general observations. First let us observe that all the sets we are working with are subsets of the (finite) set \( M_\uparrow \). Hence if we order this set arbitrarily, we can consider both the sets \( M \in \mathcal{M} \) and sets of the form \( (x \cap M) \in C_{X, M} \) as bit strings of length \( |M_\uparrow| \). We can then do intersection tests and unions on these sets in linear time (with respect to \( M_\uparrow \)). Then an algorithm can be given which consists of three main phases.

1. For each \( M \in \mathcal{M} \), calculate \( t(\preceq_M) \). Under the assumption that the agents have a common weighing relation \( \leq \), there are up to \( |M| \) preference relations we need to calculate. These do not depend on the agents, but only on the motivating set \( M \in \mathcal{M} \). If we have an arbitrary weighing relation \( \leq \) over an arbitrary set of sets of motivating reasons \( \mathcal{M} \), the algorithm may proceed by traversing the ordering induced by \( M \) directly. But we clearly only need to consider views under \( M \), and hence may avoid time dependence on the set of alternatives (which might be exponential in the number of views). Moreover, if the conditions for Axiom 3 in [27] are satisfied we can calculate the preferred elements of all \( \preceq_M \) relations more quickly by exploiting conditions for when an alternative enters or leaves the set of preferred alternatives. Axiom 3 in [27] states that for any two alternatives \( x, y \in X \) and any two motivational states \( M, M' \in \mathcal{M} \) with \( M \subseteq M' \),

\[
(x \cap M' \setminus M) = (y \cap M' \setminus M) \Rightarrow x \preceq_M y \iff x \preceq_{M'} y.
\]

2. For each agent \( a \), order \( \mathcal{M} \) by increasing Hamming distance from \( M_a \). Each Hamming distance can be considered a “window” on possible sets of motivational properties. For each of these sets with the same distance \( i \) from the current position, with \( 0 \leq i \leq \max (\text{max is the maximal distance as described above}) \) from \( M_a \), we gather the union of all preferred elements under these sets in \( t(a, i) \). That is

\[
t(a, 0) := t(\preceq_{M_a})
\]

\[
t(a, n + 1) := t(a, n) \cup \bigcup_{M_a \rightharpoonup M} t(\preceq_M)
\]
where by $M \xrightarrow{H_n} M'$ we mean that the Hamming distance between $M$ and $M'$ is $n$. Formally, $M \xrightarrow{H_n} M'$ if, and only if, there exists sets $D \subseteq M \cup M'$ and $E \subseteq M$ such that $|D \cup E| = n$ and $(M \cup D) \setminus E = M'$.

We do not need to calculate all these sets. If the solution to our problem is $k \leq \text{max}$, we can stop after calculating $t(a,k)$, showing how the complexity of the algorithm will increase the number of views we might have to consider by at most $2^k$ for solutions of distance at most $k$ from the current position (for each additional property we will at most double the number of views). Moreover, from the next step, it is easy to see that discovering when to stop is easy.

3. For each $0 \leq i \leq (\text{max distance})$, check if $(\bigcap_{x \in A} t(x,i)) \neq \emptyset$. When yes, we may return $i$, the minimum distance from the current position to a consensus position. The complexity of computing this $i$, moreover, is polynomial in the original sets $M$ up to multiplication with $2^i$, serving to upper bound the number of new views that may have to be taken into account.

### 6.7 Summary

We have provided a formalization of reason-based preference in modal logic. Starting from a straightforward Kripke construction based on the framework in [26, 27], we showed how the class of models thus defined could be characterized in modal logic with universal modality. This means that basic modal reasoning about these structures can be carried out using established techniques. It means, in particular, that the problem of deciding validity over motivational structures is in PSPACE. Moreover, we showed how generalizing the framework to the multi-agent case is straightforward, and we analyzed the problem of computing a minimal change consensus when there was only superficial disagreement between agents.

An overreaching goal is to generalize the framework to account for multi-agent arbitration concerning alternatives. It seems that this is where the real potential of reason-based preference lies. Instead of simply assuming a given state of disagreement, one can begin to ask, within a logical framework, why agents disagree, the hope being that we can eventually also arrive at formal notions of agreement and argument that incorporates, and takes into account, the motivational states of agents.

The closest related work is probably Osherson and Weinstein [62], who develop a logic for reasoning about reason-based preferences, based on a model that is more general than the one of Dietrich and List [27]. It is probably possible to use the logic in [62] to reason about Dietrich and List’s motivational structures, and to obtain a similar result to the one we present in this chapter, but this seems cumbersome. Axioms “picking out” the corresponding models would have to be developed and proven sufficient – like we do in this chapter. Moreover, here we have shown how a standard, normal modal logic can be used to reason about reason-based preferences, while [62] uses a set-up with more non-standard elements.

De Jongh and Liu [49] also introduce and study a notion of “property-based” preferences, which can be seen as a special and more limited case of the model we consider in this chapter (a single, fixed, weighing relation and a single motivational state), see [27].
Finally, formalisms for compact representation of preferences, including logic-based ones, have been developed and studied in the AI and Knowledge Representation and Reasoning communities for particular domains or applications, such as for preferences over combinatorial domains [53] or preferences in Boolean Games [19].

Strategically homogeneous agents might very well disagree about the utility of various alternatives. In this chapter we have characterized a logical system for reasoning about reason-based preferences. The agents might even be, fundamentally, in agreement in terms of their weighing relation. Simply being in different motivational states (e.g., having different desires or goals, being aware of conscious of different properties, and so on), might well accommodate immediate, or superficial, disagreement.

In the next chapter we will pick up this thread when we suggest a dynamic logic of deliberation in which the agents put forth arguments which alters the agents’ common deliberative state. Again, we will disregard the agents’ preferences (as we have done in the preceding chapters), but it will be central that agents’ views on alternatives might differ in some non-logical way. Whether the difference of views is due to a fundamental disagreement about weighing alternatives differently, or simply because the agents are motivated by different properties will not be crucial to that analysis, but the reader may well keep in mind the possibilities of disagreement discussed in this chapter when reading the next.
Chapter 7

Deliberative Dynamic Logic

7.1 Background

In this chapter we investigate a framework for argumentative deliberation. We refer to the process as “argumentative” not to emphasize that the participating agents are arguing in order to achieve their own individual interest, but because we model a process where agents advance a (common) state by means of putting forth arguments. Indeed, we will not address the agents’ motivation beyond some simple faithfulness constraints on their behavior.

We will present a logic for reasoning about agents with possibly conflicting views forming a common partial view. Partial because we assume that the agents’ views are represented as graphs over the entire set of arguments, which we consider to be infinitely large. The resulting view that they cooperatively construct will necessarily be finite.

Under these assumptions it is not clear that we may verify any properties at all about the possible developments of this process. However, we are able to show that as long as the agents’ views are not such that the topical arguments are attacked by an infinite number of counter-arguments, we are able to perform model checking over the set of traces such an exchange of arguments produces.

Our logic will have some similarities to propositional dynamic logic, but we will have arguments in place of atomic programs, and we will not cover iteration and composition. We will introduce a strong notion of choice, however, selecting from the entire (infinite) set of arguments. The model checking problem can also be solved when we permit this general modal operator.

We will discuss how an interpretation of consensus may be formalized in this logic.

7.2 Difference and Dialogue in Argumentation

Argumentation is an interactive process, and the dynamic aspect of this process is fundamental. An argument develops. We will base our framework on the argumentation framework presented by Dung in [30]. The introductory example in the article shows clearly the interactive and dynamic aspect of argumentation theory. The mock argumentation given as an example, sadly still a timely one, is an exchange of arguments between an Israeli and an Arab discussing the Middle East conflict.
Besides being highly interactive, the scenario also points to the difference of views held by the participants. Indeed, if there was no difference of opinion, arguing would be moot.

A central problem tackled in argumentation theory, as discussed in Chapter 2, asks us to “pause” the process and investigate the argumentative situation. Given the proposed set of arguments (and their relation to each other) what is a reasonable set of arguments to accept. Solution concepts to this problem are defined as semantics (as defined in Chapter 2) which provides us with a set of extensions or labelings, or, as we will assume in this chapter, as a set of argument valuations. (These notions where defined in Chapter 2.)

Such semantics seek to address the static part of the argumentation. In this chapter we will not make a choice of semantics beyond some very weak assumptions of reasonableness. We will investigate the logically possible dynamic developments of the argumentation. We will reason about these dynamic developments in terms of what changes (in the “static” states) we may observe in two different states in a sequence of such states, obtained by the participants putting forth arguments.

This, in itself, is not novel. Several investigations of various dialogue games, for example, have been proposed and investigated, such as persuasion games, negotiation games, and so on. See e.g., [69] for a formal treatment of persuasion and brief overview of the taxonomy.

One element of our approach to modeling the deliberative process, however, is our representation of differences. We will encode each agent’s view as an argumentation framework over the entire set of arguments. Generally, the number of arguments may be considered to be an infinite (countable) set of atoms. We will be working with respect to the set of arguments as a fixed parameter. A special kind of argumentation framework is an argumentation framework over the entire set of arguments. We refer to these argumentation frameworks as views.

**Definition 7.1.** Given a countable, infinite set of arguments \( \Pi \), a view is a graph \( V \subseteq \Pi \times \Pi \).

That is, a view is a (directed) graph \((\Pi, V)\) where \( V \subseteq \Pi \times \Pi \), but as \( \Pi \) is a fixed set of arguments, we omit it and refer to it only by \( V \).

The view \( V_a \) of some agent \( a \) will be a complete representation of that agent’s evaluation of the various possible arguments. That is, if an agent disagrees with any argument \( p \in \Pi \), then there will be some counter argument \( q \) such that \((q, p) \in V_a\). Or in other words, if \( a \) does not accept some argument \((p)\), then she has a reason for not accepting it \((q)\).

In order for deliberation to take place, we need to have several different views. We therefore define the basis of deliberation as an agent indexed collection of views.

We assume given a finite non-empty set \( A \) of agents and a countably infinite set \( \Pi \) of arguments, or possibly “statements” or “positions”, depending on the context of application. We will sometimes also think of these as arguments to perform a certain action. This might well be a communicative act such as decreeing, agreeing, or assenting. The basic building block of dynamic deliberative logic is provided in the following definition.

**Definition 7.2.** A basis for deliberation is an \( A \)-indexed collection of digraphs \( \mathcal{B} = (V_a)_{(a \in A)} \), such that for each \( a \in A \), \( V_a \subseteq \Pi \times \Pi \).
Given a basis which encodes agents’ view of the arguments, we are interested in the possible ways in which agents can deliberate to reach agreement on how arguments are related. That is, we are interested in the set of all AFs that can plausibly be seen as resulting from a consensus regarding the status of the arguments in $\Pi$.

The argumentation frameworks which are candidates for being a consensus and which we are generating from a basis, are produced by arguments being added to an initially empty argumentation one at a time. For every argument added, we branch out along every possible way that the argument may be received by the set of agents. As the agents’ views are argumentation frameworks over the entire set of arguments, the agents all have an understanding of how they perceive the argument, although it may not all be encoded in what has been put forth by any agent (yet).

**Example 7.3.** Let us consider a scenario consisting of two agents which need to agree on which of $p$ or $q$. They both recognize that $p$ and $q$ are mutually exclusive. However, this has not been established yet. Suppose we have the views illustrated in Figures 7.1 and 7.2.

![Figure 7.1: $V_a$](image1)

![Figure 7.2: $V_b$](image2)

Let us say that $p$ is the statement “we should enter the tunnel”, and let $q$ be the statement “we should stay outside the tunnel”. Naturally, these are short, unreasoned “arguments” which would be more elaborate in a realistic setting. Furthermore, let $r$ be “we should not stray from the light” (e.g., by entering a dark tunnel), and $s$ be “we need to progress”.

As we can see from the views of the agents, the agents agree about most of the setting. They both agree that we can not both endorse entering the tunnel and staying outside. They both agree also that entering the tunnel would be straying from the light. The agent that wishes to not stray from the light, hence, will not support entering the tunnel. Staying outside is a hindrance to progress.

In this case we assume that what we need to decide is which subset of arguments from \{p, q\} we should endorse. The way we envision deliberation, it is not crucial who proposes a given argument. What is important is how the argument is received. In this, we assume each participant will have the possibility to voice her opinion and no voice automatically trumps any other.

---

1The observant reader may have noticed that we have only four arguments in our example, but that the definition of a view presupposes that there are infinitely many arguments. For the sake of this example we only need four, but we may assume that there are infinitely many arguments in the model we are discussing but such that none of the arguments mentioned in the example explicitly are connected to any argument which is not mentioned explicitly. This entails, in our framework, that we can disregard the other arguments (most of the time). See Definition 7.6 why we may make this assumption, and the definition of satisfaction of $\Box$-formulas in Definition 7.13 for why this is possible only most of the time.
As a first step in the deliberation process, one may suggest $p$.\textsuperscript{2} Then this argument stands undefeated in its isolation. Then, some agent could propose $q$. Regardless of whether $a$ or $b$ proposes $q$, it is clear that it is in mutual opposition to $p$. This is made explicit when the argument is put forth and no agent has any possibility to truthfully disagree so far. We transition from a state (1) to state (2) in Figure 7.3.

This sets the stage for the dilemma the agents need to resolve. The agents, at their discretion, might of course decide to stop deliberating at this point and simply declare the agreement that “we can not do both”. However, this is unsatisfactory and further reasons need to be explored. Suppose some agent proposes that we ought to not enter the tunnel because it is dark ($r$).\textsuperscript{3} Both agents agree about how $r$ relates to other already presented arguments, so we transition into (3).

\begin{figure}[h]
\centering
\begin{tikzpicture}
\node (p1) at (0,0) {$p$};
\node (p2) at (2,0) {$p$};
\node (q) at (2,1) {$q$};
\draw (p1) -- (p2);
\node (r) at (2,2) {$r$};
\node (s) at (2,3) {$s$};
\draw (p2) -- (q);
\draw (p1) -- (p2);
\end{tikzpicture}
\caption{Three consecutive states in the deliberation process.}
\end{figure}

At this point, we may consider what is a reasonable set of arguments to accept. We may agree that $r$, being uncontested, should be accepted. This leaves $p$ rejected. We may freely choose to accept $q$ at this stage as no acceptable arguments attack it. We will inherit this conclusion from argumentation theory semantics and make few technical appeals in this direction. However, as we and the reader have insight into the agents’ views on the arguments, one would perhaps be surprised if the deliberation stopped at this point. We know that some agent has further arguments that could advance the state of affairs. Indeed, both agents have such views concerning the arguments.

Consider how the deliberation might proceed at this point. The argument $s$ could be proposed, as an argument against $q$ (solely). There is no disagreement about the relationship between $s$ and $q$ and hence, by assuming the agents are deliberating truthfully, this relation must be incorporated in the common argumentation framework. This case is illustrated in (4\textsubscript{1}) in Figure 7.4.

\begin{figure}[h]
\centering
\begin{tikzpicture}
\node (p1) at (0,0) {$p$};
\node (p2) at (2,0) {$p$};
\node (q) at (2,1) {$q$};
\node (r) at (0,1) {$r$};
\node (s) at (0,2) {$s$};
\node (r1) at (4,0) {$r$};
\node (s1) at (4,2) {$s$};
\node (r2) at (4,1) {$r$};
\node (s2) at (6,2) {$s$};
\node (r3) at (6,1) {$r$};
\node (s3) at (6,3) {$s$};
\node (r4) at (8,1) {$r$};
\node (s4) at (8,3) {$s$};
\draw (p1) -- (p2);
\draw (p2) -- (q);
\draw (p1) -- (p2);
\draw (r) -- (s);
\draw (r1) -- (s1);
\draw (r2) -- (s2);
\draw (r3) -- (s3);
\draw (r4) -- (s4);
\end{tikzpicture}
\caption{Four potential successor states in the deliberation process.}
\end{figure}

\textsuperscript{2}The example would develop equivalently if we start with $q$.

\textsuperscript{3}It might be tempting to think that if $r$ is argued, then this argument is probably put forth by $a$ (who seems to support $q$). However, we are not focusing on the strategic dynamics, and either agent could put forth $r$. 
How should we interpret the outcome depicted in \((4_1)\) of Figure 7.4? We could interpret this as the state in which \(s\) has been put forth but, as the relationship between \(s\) and \(r\) is contested, the agents have accepted a compromise. The agents have agreed that, given the state of affairs, it is impossible to accept either \(p\) or \(q\). There seems to be, however, agreement about the principles, we should not stray from the light, but also pursue progress.

Similarly, the agents could reach a compromise state in which they consider \(r\) and \(s\) to be in mutual opposition. Such an outcome is depicted in \((4_4)\). Each agent has conceded that the other party’s view on the relationship between \(s\) and \(r\) is legitimate. In this case it seems that the agents agree that the “principles” (\(r\) and \(s\)) are not consistent in the sense that we could accept one, but not both. Which is accepted is arbitrary (given the information in the argumentation framework), and which of these the agents accept decides the outcome of the deliberation in the sense that accepting \(r\) permits the acceptance of \(q\) and rules out the acceptance of \(p\). Similarly, accepting \(s\) supports the acceptance of \(p\), but not of \(q\).

The particular details of which exact subsets of arguments we consider acceptable in an argumentation framework is, as discussed earlier, not central in this exposition. We rely on the work on semantics in argumentation theory to determine this.

In \((4_2)\) (and respectively \((4_3)\)) of Figure 7.4 agent \(a\) (respectively, agent \(b\)) has “won through” in the sense that the other agent seems to have conceded to her view. This is also a possible outcome, particularly if we grant that different agents have different powers of persuasion. These outcomes are certainly also plausible outcomes in a deliberation.

Which of these outcomes is the correct outcome? Normatively, we should perhaps disregard \((4_2)\) and \((4_3)\) from the example, since they are not proper compromises, one agent just wins outright. But we refrain from making any claims about the best, or about the most likely, outcome. As we are trying to model the process of deliberation from a logical perspective, we will retain all these outcomes as possible.

Even though we disregard the problems of stipulating which of the outcomes is the rational outcome, and which is the most likely outcome, we do not dismiss this problem. Recently, the problem of merging such agent-relative views have been devoted some attention. We mention that one approach to this has been investigated by Coste-Marquis et al. in [24]. In this article, the authors show why a simplistic approach to merging views by simply taking the union of the agents’ attack relations, is unreasonable in terms of argumentation theory. They also investigate how one may find more reasonable solutions to the merging problem for example in terms of voting, and in terms of distance between the solution and the agents’ views.

Another related article by Endriss & Grandi is [35] in which the merging of graphs is approached more conceptually. The tools and techniques provided in this article addresses graph aggregation not only in the case of merging agents’ (subjective) views, but certainly has interesting applications also to this problem.

In either case, we propose that the outcomes, stipulated by any approach to find the rational or the expected outcome in a particular aggregation or merging problem, should be included in the set we refer to as the possible outcomes. Hence, we consider the problem of merging or aggregating the agents’ views as the problem of isolating an appropriate member (or subset) of the possible outcomes. Importantly, we introduce a
modal logic for doing this.

As in the example above, we will place some minimal constraints on the argumentation frameworks produced in a deliberative process. It seems that while many restrictions might arise from pragmatic considerations, or from specific protocols for “good” deliberation in specific contexts, there are few restrictions that can be regarded as completely general.

For instance, while there is often good reason to think that the position held by the majority will be part of a consensus, it is hardly possible to stipulate an axiomatic restriction on the notion of consensus amounting to the principle of majority rule. Indeed, sometimes deliberation takes place and leads to a single dissenting voice convincing all the others, and often, these deliberative processes are far more interesting than those that transpire along more conventional lines. (However, see [31] for one way of reasoning about axioms for argumentation. Our framework satisfies two of these axioms as we will mention later.)

However, it seems reasonable to assume that whenever all agents agree on how an argument $p$ is related to an argument $q$, then this relationship is part of any consensus. This, indeed, is the only restriction we will place on the notion of a consensus; that when the AF $(A,E)$ containing both $p$ and $q$ is a consensus for a basis, it must satisfy the following faithfulness requirement.

- For all $p,q \in \Pi$, if there is no disagreement about $p$’s relationship to $q$ (attack/not attack), then this relationship is part of $(A,E)$

This will be the only restriction we make on the argumentation frameworks to be considered as possible consensus positions. Further restrictions may be considered reasonable, but we will not pursue this line of inquiry in this chapter.

We now turn to how we may reason about the dynamic development of the consensus process.

### 7.3 Deliberative Dynamic Logic

Given a finite, non-empty set of agents $A$ and a countably infinite set of arguments $\Pi$. We say that a basis is an agent indexed collection of views as defined in Definitions 7.1 and 7.2. Furthermore, we assume that there is a semantic $\varepsilon$ with the signature

$$\varepsilon(A,E) \subseteq 3^A.$$  

Many proposals exists in the literature, we point to [11] for a survey and formal comparison of different semantics. The framework we are suggesting assumes that we have a semantic as a parameter. We will only use admissible and preferred semantics in our examples.

Given an argumentation framework $(S,E)$ we denote the outgoing edges from a set of arguments $A \subseteq S$ as

$$E^+(A) := \{ y \in S \mid x \in A, (x,y) \in E \}$$

and the incoming edges to a set of arguments $A \subseteq S$ as

$$E^-(A) := \{ y \in S \mid x \in A, (y,x) \in E \}$$
Given a argumentation framework \((S, E)\), an admissible extension is a subset of arguments \(A \subseteq S\) such that

- every argument that attacks an argument in \(A\) is attacked by an argument in \(A\), and
- no argument in \(A\) attacks an argument in \(A\).

These two conditions have been called acceptability of \(A\) with respect to \(S\) and conflictfreeness respectively at least since Dung’s seminal article [30]. An admissible extension which is subset-maximal (not contained in any other admissible extension) is said to be a preferred extension. Succinctly, we may define these semantics as in the following definition.

**Definition 7.4.** Given an argumentation framework \((S, E)\), we define the set of admissible and preferred extensions, or the admissible and preferred (extension-based) semantics \(\varepsilon_a^E\) and \(\varepsilon_p^E\), respectively, as follows

**Admissible semantics** The set of admissible extensions is

\[
\varepsilon_a^E(S,E) := \{ A \subseteq S \mid E^-(A) \subseteq E^+(A) \subseteq S \setminus A \}
\]

**Preferred semantics** The set of preferred extensions is the set

\[
\varepsilon_p^E(S,E) := \{ A \in \varepsilon_a^E(S,E) \mid \text{for no } A' \in \varepsilon_a^E(S,E) \text{ it holds that } A \subset A' \}
\]

We will assume that our semantics are valuation-based (as discussed in Chapter 2). An intuitive way of transferring the notion of an extension-based semantics to valuation-based semantics is to attribute the value of 1 (accepted) to the members of an extension, a value of 0 (rejected) to arguments attacked by an accepted argument, and \(1/2\) to the remaining arguments.

For an argumentation valuation \(\pi : S \rightarrow \mathbb{L}\) we will subscript the value to indicate the preimage of that value. That is, for an argument valuation for an argumentation framework \((S, E)\)

\[
\pi_x := \{ p \in S \mid \pi(p) = x \}
\]

That is, given an argument labeling \(\pi\) we denote the set of accepted arguments, i.e., the arguments \(p\) for which \(\pi(p) = 1\), by \(\pi_1\). Similarly for the set of rejected and undecided/undecidable arguments.

The admissible and preferred valuation-based semantics can hence be defined as follows:

**Definition 7.5.** Given an argumentation framework \((S, E)\) the valuation-based semantic \(\varepsilon\) corresponding to some extension-based semantic \(\varepsilon^E\), are defined as sets of maps from arguments to the set \(\mathbb{L} = \{1, 1/2, 0\}\) as follows.

\[
\varepsilon(S,E) := \{ \pi \in \mathbb{L}^S \mid A \in \varepsilon^E, \ x \in A \iff \pi(x) = 1, \ y \in E^+(A) \iff \pi(y) = 0, \text{ and } \ z \notin A \cup E^+(A) \iff \pi(z) = 1/2 \}
\]
In particular, we will refer to the valuation-based admissible and preferred semantics, \( \varepsilon_a \) and \( \varepsilon_p \) respectively, defined according to the above. Alternatively, we can define the valuation-based admissible semantics as

\[
\varepsilon(S, E) := \{ \pi \in \mathbb{L}^S | \pi_0 = E^+(\pi_1), E^-(\pi_1) \subseteq \pi_0 \subseteq S \setminus \pi_1 \}
\]

For a discussion of the relation between the extension-based and valuation-based semantics, we refer to [8].

Since argumentation semantics typically only restrict the choice of successful arguments, without determining it completely, a modal notion of acceptance arises, usually referred to as skeptical acceptance in argumentation parlance, whereby an argument is said to be skeptically accepted by \( (A, E) \) under \( \varepsilon \) if \( \forall S \in \varepsilon((A, E)) \) we have \( p \in S \). The dual notion is called credulous acceptance, and obtains just in case \( \exists S \in \varepsilon((A, E)) \) such that \( p \in S \).

The framework does not rely on any particular semantics, but we do require that the semantic satisfy a particular normality property. We require that the status of an argument depends only on its relation to other arguments through the attack relation.

**Definition 7.6 (Normal Semantics).** Let \( C(S, E) \) be the set of all connected components in \( (S, E) \). A semantic \( \varepsilon \) is normal if, and only if, for every argumentation framework \( (S, E) \),

- for every labeling \( A \in \varepsilon(S, E) \), there is a map \( B : C(S, E) \rightarrow 2^{(3^S)} \) such that for every \( X \in C(S, E) \) we have \( B(X) \in \varepsilon(X) \) and
  \[
  A = \bigcup_{X \in C(S, E)} B(X)
  \]

- for every map \( B : C(S, E) \rightarrow 2^{(3^S)} \) which satisfies \( B(X) \in \varepsilon(X) \) for every \( X \in C(S, E) \), there is an \( A \in \varepsilon(S, E) \) such that
  \[
  A = \bigcup_{X \in C(S, E)} B(X)
  \]

That is, a semantics is normal if the status of an argument depends only on those arguments to which it has some (indirect) relationship through a sequence of attacks. We remark that all argumentation semantics of which we are aware satisfies this requirement, hence we feel justified in dubbing it normality.

Finally, we also require that the semantics satisfies a further property, that it is compatible with an extension-based view as per Definition 7.5. We dub this property extension compatible.

**Definition 7.7 (Extension Compatible Semantics).** A semantic \( \varepsilon \) is extension compatible if, and only if, for every argumentation framework \( (S, E) \) and every \( \pi \in \varepsilon(S, E) \), we have \( \pi_0 = E^+(\pi_1) \).

We will interpret the elements in \( \varepsilon(S, E) \) as argument valuations which we then extend by Łukasiewicz three-value logic \( \mathbb{L}_3 \).
We define a language $\mathcal{L}_L$ by BNF as
\[
\alpha ::= p \mid \neg \alpha \mid \alpha \rightarrow \alpha
\]
where $p \in \Pi$.

We rely on the usual abbreviations for Łukasiewicz logic.

**Definition 7.8** ($\alpha$-satisfaction). For any argument valuation $\pi = (\pi_1, \pi_0, \pi_{1/2})$ of $\Pi$, we define
\[
\begin{align*}
\pi(p) &= x \text{ such that } p \in \pi_x \\
\pi(\neg \alpha) &= 1 - \pi(\alpha) \\
\pi(\alpha_1 \rightarrow \alpha_2) &= \min \{1, 1 - (\pi(\alpha_1) - \pi(\alpha_2))\}
\end{align*}
\]

We can now discuss the acceptability of various arguments given an argument valuation.

**Example 7.9.** Suppose we have three arguments $\{a, b, c\}$ in the argumentation framework illustrated in Figure 7.5.

\[a \leftrightarrow b \leftrightarrow c \Rightarrow\]

Figure 7.5: Simple argumentation framework.

One preferred extension in this case is $\{a\}$. So let us consider the corresponding argument valuation, i.e., $\pi(a) = 1$, $\pi(b) = 0$, and $\pi(c) = 1/2$.

We may state that “$a$ is accepted and the status of $c$ is undecided” by the formula $a \land (c \leftrightarrow \neg c)$.

These considerations prompts us to ask what the agents might hope to achieve in a deliberation. We define the set $\mathcal{T}(\mathcal{B})$, which we will call the set of complete assents for $\mathcal{B}$, collecting all AFs that are faithful to $\mathcal{B} = (V_a)_{a \in A}$.

\[
\mathcal{T}(\mathcal{B}) = \left\{ (A, E) \subseteq \Pi \times \Pi \mid \bigcap_{a \in A} V_a \subseteq (A, E) \subseteq \bigcup_{a \in A} V_a \right\} \tag{7.10}
\]

An element of $\mathcal{T}(\mathcal{B})$ represents a possible consensus among agents in $A$, but it is an idealization of the notion of assent, since it disregards the fact that in practice, assent tends to be partial, since it results from a dynamic process, emerging through deliberation. Indeed, as long as the number of arguments is not bounded we can never hope to arrive at complete assent via deliberation. We can, however, initiate a process by which we reach agreement on more and more arguments, in the hope that this will approximate some complete assent, or maybe even be robust, in the sense that there is no deliberative future where the results of current partial agreement end up being undermined. Complete assent, however, arises only in the limit.

When and how deliberation might successfully lead to an approximation of complete assent is a question well suited to investigation with the help of dynamic logic. The dynamic element will be encoded using a notion of a deliberative event – centered
on an argument – such that the set of ways in which to relate this argument to arguments previously considered gives rise to a space of possible deliberative time-lines, each encoding the continued step-wise construction of a joint point of view. This, in turn, will be encoded as a monotonically growing argumentation framework \((S, E)\) where \(S \subseteq \Pi, E \subseteq S \times S\) and such that faithfulness is observed by all deliberative events. That is, an event consists of adding to \((S, E)\) the agents’ combined view of \(p\) with respect to the set \(S \cup \{ p \}\).

We need to be able to refer to particular subgraphs, so let us denote the subgraph of \((S, E)\) induced by the nodes \(A \subseteq S\) as \((S, E)|_A\). The induced subgraph is the subgraph 

\[ (A, E') = (S, E)|_A \] 

where \(A \subseteq S\) and 

\[ E' = E \cap (A \times A) \]

This leads to the following collection of possible events, given a basis \(\mathcal{B}\), a partial consensus\(^{4}\) \((S, E)\) and an argument \(p \in \Pi\):

\[
\mathcal{U}_\mathcal{B}((S, E), p) = \left\{ X \subseteq \Pi \times \Pi \mid \bigcap_{a \in A} V_a|_{S \cup \{ p \}} \subseteq X \subseteq \bigcup_{a \in A} V_a|_{S \cup \{ p \}} \right\} \tag{7.11}
\]

To provide a semantics for a logical approach to deliberation based on such events, we will use Kripke models.

**Definition 7.12 (Deliberative Kripke model).** Given a normal and extension consistent argumentation semantics \(\varepsilon\) and a set of views \(\mathcal{B}\), the deliberative Kripke models induced by \(\mathcal{B}\) and \(\varepsilon\) is the triple \(K_{(\mathcal{B}, \varepsilon)} = (Q_\mathcal{B}, R_\mathcal{B}, \varepsilon)\) such that

- \(Q_\mathcal{B}\), the set of points, is the set of all pairs of the form \(q = (q_S, q_E)\) where \(q_S \subseteq \Pi\) and 

\[
\bigcap_{a \in A} V_a|_{q_S} \subseteq q_E \subseteq \bigcup_{a \in A} V_a|_{q_S}
\]

The basis \(\mathcal{B}\) together with our definition of an event, given in Equation 7.11, induces the following function, mapping states to their possible deliberative successors, defined for all \(p \in \Pi, q \in Q_\mathcal{B}\) as follows

\[
succ(p, q) := \{ (q_S \cup \{ p \}, q_E \cup X) \mid X \in \mathcal{U}_\mathcal{B}(q, p) \}
\]

We also define a lifting, for all states \(q \in Q_\mathcal{B}\):

\[
succ(q) := \{ q' \mid \exists p \in \Pi : q' \in succ(q, p) \}
\]

- \(R_\mathcal{B} : \Pi \cup \{ \exists \} \rightarrow 2^{Q_\mathcal{B} \times Q_\mathcal{B}}\) is a map from symbols to relations on \(Q_\mathcal{B}\) such that

  - \(R_\mathcal{B}(p) = \{ (q, q') \mid q' \in succ(p, q) \}\) for all \(p \in \Pi\) and
  - \(R_\mathcal{B}(\exists) = \{ (q, q') \mid q' \in succ(q) \}\).

\(^{4}\)These “partial consensuses” are sometimes referred to as “contexts” when they are used to describe graphs inductively, as we will do later.
• $\varepsilon : Q_B \rightarrow 2^{(3^\Pi)}$ maps states to sets of labelings.

Notice that the default status, attributed to all arguments not in $q_S$, is $\frac{1}{2}$ by extension consistency. Every state in every model in our framework satisfies two of the axioms considered in [31]:

**Unanimous Attack (UA)** corresponding to the left set-inclusion of the definition of the set of states. That is, for arguments $x$ and $y$ in some state, if all agents agree that $x$ attacks $y$, then this edge must be included in the state, and

**Attack Closure (AC)** corresponding to the right set-inclusion of the definition of the set of states. For arguments $x$ and $y$, if no agent is of the view that $x$ attacks $y$, then there may not be any edge from $x$ to $y$ in the state.

These axioms are also satisfied in every complete assent.

The logical language we will use consists in two levels. For the lower level, used to talk about static argumentation, we follow [8, 33] in using Łukasiewicz three-valued logic. Then, for the next level, we use a dynamic modal language which allows us to express consequences of updating with a given argument, and also provides us with existential quantification over arguments, allowing us to express claims like “there is an update such that $\phi$”. This leads to the language $L_{DDL}$ defined by the following BNF’s:

$$\phi ::= \Diamond \alpha | \neg \phi | \phi \land \phi | \langle p \rangle \phi | \Diamond \phi$$

where $p \in \Pi$ and $\alpha \in L_L$.

We also use standard abbreviations such that $\Box \phi = \neg \Diamond \neg \phi$, $[p] \phi = \neg \langle p \rangle \neg \phi$ and $\lozenge \alpha = \neg \Diamond \neg \alpha$. We also consider that standard boolean connectives abbreviated as usual for connectives not occurring inside a $\Diamond$-connective and abbreviations for connectives of Łukasiewicz logic in the scope of $\Diamond$-connectives.

Now we can give a semantic interpretation of the full language as follows.

**Definition 7.13 ($L_{DDL}$-satisfaction).** Given an argumentation semantics $\varepsilon$ and a basis $B$, truth on $K_{(B,\varepsilon)}$ is defined inductively as follows, in all points $q \in Q_B$.

1. $K_{(B,\varepsilon)}, q \models \Diamond \alpha \iff$ there is $\pi \in e_\varepsilon(q)$ such that $\overline{\pi}(\phi) = 1$
2. $K_{(B,\varepsilon)}, q \models \neg \phi \iff$ not $K_{(B,\varepsilon)}, q \models \phi$
3. $K_{(B,\varepsilon)}, q \models \phi \land \psi \iff$ both $K_{(B,\varepsilon)}, q \models \phi$ and $K_{(B,\varepsilon)}, q \models \psi$
4. $K_{(B,\varepsilon)}, q \models \langle p \rangle \phi \iff$ $\exists (q, q') \in R_B(p) : K_{(B,\varepsilon)}, q' \models \phi$
5. $K_{(B,\varepsilon)}, q \models \Diamond \phi \iff$ $\exists (q, q') \in R_B(\exists) : K_{(B,\varepsilon)}, q' \models \phi$

**Example 7.14.** Consider a simple example consisting of two agents over a domain consisting of only two arguments, $p$ and $q$. They agree on their relationship, they are mutually opposed to each other. However, agent $a$ views $p$ as a contradiction, and agent $b$ views $q$ as a contradiction. We illustrate their views in Figure 7.6. In Figure 7.7, we depict a fragment of the corresponding Kripke model, in particular a fragment consisting of various successors of $(\emptyset, \emptyset)$. 
Let us assume that $\varepsilon$ is the preferred semantics. Then the following list gives some formulas that are true on $\mathcal{K}_{(B, \varepsilon)}$ at the point $(0, 0)$, and the reader should easily be able to verify them by consulting the above fragment of $\mathcal{K}_{(B, \varepsilon)}$.

\[
\begin{align*}
&\langle p \rangle [p] \phi \leftrightarrow [p] \langle p \rangle \phi \\
&\langle p \rangle [q] \phi \rightarrow [q] \langle p \rangle \phi \\
&\diamond \square \phi \rightarrow \square \diamond \phi \\
&\langle p \rangle \langle p \rangle \phi \rightarrow \langle p \rangle \phi
\end{align*}
\]

We can also record some validities that are easy to verify against Definition 7.12.

**Proposition 7.15.** The following formulas are all validities of $\mathcal{L}_{\text{DDL}}$, for any $p, q \in \Pi$, $\phi \in \mathcal{L}_{\text{DDL}}$.

1. $\langle p \rangle \langle q \rangle \phi \leftrightarrow \langle q \rangle \langle p \rangle \phi$
2. $\langle p \rangle [q] \phi \rightarrow [q] \langle p \rangle \phi$
3. $\diamond \square \phi \rightarrow \square \diamond \phi$
4. $\langle p \rangle \langle p \rangle \phi \rightarrow \langle p \rangle \phi$

We remark that $[q] \langle p \rangle \phi \rightarrow \langle p \rangle [q] \phi$ is not valid, as witnessed for instance by basis $B$ shown in Figure 7.8, for which we have $\mathcal{K}_{(B, p)}(0, 0) \models [q] \langle p \rangle \Box p$ but also $\mathcal{K}_{(B, p)}(0, 0) \not\models [q] \langle p \rangle \Box q$ (as the reader may easily verify by considering the corresponding Kripke model).
Finally, let us notice that as $\Pi$ is generally infinite, we must expect to encounter infinite bases. This means, in particular, that our Kripke models are often infinite. However, in the next section we show that as long as $B$ is finitary, meaning that no agent $a \in A$ has a view where an argument is attacked by infinitely many other arguments, we can solve the model-checking problem also on infinite models.

### 7.4 Model checking on finitary models

Towards this result, we now introduce some notation and a few abstractions to simplify our further arguments. We will work with labeled trees, in particular, where we take a tree over labels $X$ to be some non-empty, prefix-closed subset of $X^*$ (finite sequences of elements of $X$). Notice that trees thus defined contain no infinite sequences. This is intentional, since we will “shrink” our models (which may contain infinite sequences of related points), by mapping them to trees. To this end we will use the following structures. We use $e$ to denote the empty string.

**Definition 7.16.** Given a basis $B$, we define $J(B)$, a set of sequences over $\Pi \times 2^\Pi$ labeled by argumentation frameworks, defined inductively as follows

**Base case:** $e \in J(B)$ and is labeled by the argumentation framework $(A, E)(e) = (S(e), E(e))$ where $S(e) = \emptyset = E(e)$.

**Induction step:** If $x \in J(B)$, then for any $p \in \Pi$ and any partial assent $X = \mathcal{U}_B(x, p)$, we have $x; (p, X) \in J(B)$ labeled by the argumentation framework $(A, E)(x; (p, X))$ where $S(x; (p, X)) = S(x) \cup \{p\}$ and $E(x; (p, X)) = E(x) \cup X$.

It should not be confused with the argumentation semantics $\varepsilon$. This will also be clear from the context. We next define tree-representations of our Kripke models.

**Definition 7.17.** Let $\mathcal{K}_{(B, \varepsilon)}$ be some model. The tree representation of $\mathcal{K}_{(B, \varepsilon)}$ is the set $T$, together with the representation map $\gamma: Q_B \rightarrow 2^T$, defined inductively as follows

**Base case** $e \in T$ is the root with $\gamma((\emptyset, \emptyset)) = \{ e \}$.

**Induction step** For any $x \in T$, $q \in \mathcal{K}_{(B, \varepsilon)}$ with $x \in \gamma(q)$ and $q' \in \text{succ}(q)$ witnessed by $p \in \Pi$ and $X = \mathcal{U}_B(x, p)$, we have $x; (p, X) \in T$ with $q' \in \gamma(x; (p, X))$.

Notice that the tree-representation is a tree where each node is an element of $J(B)$. Some single states in $\mathcal{K}_{(B, \varepsilon)}$ will have several representations in a tree. That is, $\gamma(q)$ may not be a singleton. On the other hand, it is easy to see that for every state $q \in \mathcal{K}_{(B, \varepsilon)}$, and every path from $(\emptyset, \emptyset)$ to $q$, there will be a node $x \in T$ such that $q \in \gamma(x)$.

The main result of our paper is that model checking $L_{DDL}$-truth at $(\emptyset, \emptyset)$ is tractable as long as all views are finitely branching, i.e., such that for all $a \in A$, $p \in \Pi$, $p$ has only finitely many attackers in $V_a$. Clearly this requires shrinking the models since the modality $\Diamond$ quantifies over an infinite domain whenever $\Pi$ is infinite. We show,
however, that attention can be restricted to arguments from $\Pi$ that are relevant to the formula we are considering. To make the notion of relevance formal, we will need the following measure of complexity of formulas.

**Definition 7.18.** The white modal depth of $\phi \in \mathcal{L}_{DDL}$ is $|\phi|^\diamond \in \mathbb{N}$, which is defined inductively as follows:

\[
\begin{align*}
|\alpha|^\diamond & := 0 & \text{no white connectives in these formulas} \\
|\diamond \alpha|^\diamond & := 0 \\
|\neg \phi|^\diamond & := |\phi|^\diamond & \text{depth is deepest nesting of} \\
|\phi \land \psi|^\diamond & := \max\{|\phi|^\diamond, |\psi|^\diamond\} \\
|\diamond \phi|^\diamond & := 1 + |\phi|^\diamond \\
|\langle p \rangle \phi|^\diamond & := 1 + |\phi|^\diamond
\end{align*}
\]

We let $\Pi(\phi)$ denote the set of arguments occurring in $\phi$ in sub-formulas from $\mathcal{L}^\diamond$. Notice that given a state $q \in Q_B$, the satisfaction of a formula of the form $\phi = \diamond \alpha$ at the argumentation framework encoded by $q$ is not dependent on the entire digraph $q = (q_S, q_E)$. Indeed, this is what motivated our definition of normality for an argumentation semantics, leading to the following simple lemma, which is the first step towards shrinking Kripke structures for the purpose of model checking. Given a model $K_{(B, \epsilon)}$ and a state $q \in Q_B$, we let $C(q, \Phi)$ denote the digraph consisting of all connected components from $q$ which contains a symbol from $\Phi$. Then we obtain the following.

**Lemma 7.19.** Given a semantics $\epsilon$ and two bases $B$ and $B'$, we have, for any two states $q \in K_{(B, \epsilon)}$ and $q' \in K_{(B', \epsilon)}$ and for any formula $\phi \in \mathcal{L}_{DDL}$ with $|\phi|^\diamond = 0$:

\[ (C(q, \Pi(\phi)) = C(q', \Pi(\phi))) \Rightarrow (K_{(B, \epsilon)}, q \models \phi \leftrightarrow K_{(B', \epsilon)}, q' \models \phi) \]

In order to complete our argument in this section, we will make use of $n$-bisimulations modulo a set of symbols.

**Definition 7.20.** Given two models (with possibly different bases, but with common set of symbols $\Pi$ and semantic $\epsilon$) $K_B = (Q_B, R, \epsilon)$ and $K_B' = (Q_B', R', \epsilon)$, states $q \in Q_B$ and $q' \in Q_B'$, a natural number $n$ and a set $\Phi \subseteq \Pi$, then we say that $q$ and $q'$ are $n$-bisimilar modulo $\Phi$ (denoted $(K_B, q) \sim_{n, \Phi} (K_{B'}, q')$), if, and only if, there are $n + 1$ relations relation $Z_n \subseteq Z_{n-1} \subseteq \cdots \subseteq Z_0 \subseteq Q_B \times Q_B'$ such that

1. $qZ_n q'$,
2. whenever $(v, v') \in Z_0$, then $C(v, \Phi) = C(v', \Phi)$,
3. whenever $(v, v') \in Z_{i+1}$ and $vRu$, then there is a $u'$ such that $v'R'u'$ and $uZ_i u'$,
4. whenever $(v, v') \in Z_{i+1}$ and $v'R'u'$, then there is a $u$ such that $vRu$ and $uZ_i u'$.

Let us now also define a particular subset of arguments, the arguments which have at most distance $n$ from some given set of arguments:
Definition 7.21. Given a basis $\mathcal{B} = (V_a)_{a \in A}$, a subset $\Phi \subseteq \Pi$ and a number $n$, the $n$-vicinity of $\Phi$ is $D(\mathcal{B}, \Phi, i) \subseteq \Pi$, defined inductively as follows

\[
D(\mathcal{B}, \Phi, 0) = \Phi \\
D(\mathcal{B}, \Phi, n + 1) = D(\mathcal{B}, \Phi, n) \\
\cup \left\{ p \in \Pi \mid \exists q \in D(\mathcal{B}, \Phi, n) \text{ with } \{(p, q), (q, p)\} \cap \bigcup_{a \in A} V_a \neq \emptyset \right\}
\]

Notice that as long as $\Phi$ is finite and all agents’ views have finite branching, then the set $D$ is also finite. Also notice that an equivalent characterization of the set $D((V_a)_{a \in A}, \Phi, i)$ can be given in terms of paths as follows: an argument $p \in \Pi$ is in $D(\mathcal{B}, \Phi, i)$ if, and only if, there is a path $p = x_1x_2\ldots x_n$ in $\bigcup_{a \in A} V_a$ such that $x_n \in \Phi$ and $n \leq i$ (we consider an argument $p$ equivalently as an empty path at $p$).

Definition 7.22. Given a formula $\phi \in \mathcal{L}_{\mathcal{DDL}}$. Let $(V_a)_{a \in A}$ be a possibly infinite basis, we define $\rho_{\phi}((V_a)_{a \in A})$ such that

- for every $a \in A$, $\rho_{\phi}(V_a) := V_a \cap D(V_a, \Pi(\phi), |\phi|^\circ)$

Notice that the Kripke model for $\rho(\mathcal{B})$ will have finite branching as long as the argument symbols in the $|\phi|^\circ$-vicinity of the argument symbols in $\phi$ have finite branching in all agents’ views. In the following, we will show that for any finitely branching $\mathcal{B}$ and normal $\varepsilon$, we have $\mathcal{K}_{(\mathcal{B}, \varepsilon)}((\emptyset, \emptyset)) \models \phi$ if, and only if, $\mathcal{K}_{(\rho_{\phi}(\mathcal{B}), \varepsilon)}((\emptyset, \emptyset)) \models \phi$.

Theorem 7.23. Let $\mathcal{B}$ be an arbitrary basis, and $\phi \in \mathcal{L}_{\mathcal{DDL}}$.

\[
\left(\mathcal{K}_{(\mathcal{B}, \varepsilon)}((\emptyset, \emptyset))\right) \models^{\Pi(\phi)} \left(\mathcal{K}_{(\rho_{\phi}(\mathcal{B}), \varepsilon)}((\emptyset, \emptyset))\right)
\]

Proof. Let $\mathcal{K}_{(\mathcal{B}, \varepsilon)}$ be an arbitrary model and let $T$ denote its tree representation, while $T'$ denotes the tree representation of $\mathcal{K}_{(\rho_{\phi}(\mathcal{B}), \varepsilon)}$.

We take $n = |\phi|^\circ$ and let $\Phi$ be the atoms occurring in $\phi$ inside the scope of some $\diamond$-operator. Moreover, for brevity, we denote $D = D(\mathcal{B}, \Phi, n)$.

Definition of $(Z_i)_{0 \leq i \leq |\phi|^\circ}$: We define all the relations $Z_i$ inductively using the tree-representations as follows.

Base case: $(i = 0)$ For all $0 \leq i \leq n$, we let $eZ_i e$.

Induction step: $(0 < i \leq n)$ For all $y = x; (v, X) \in T$ and $y' = x'; (v', X') \in T'$, both of length $i$, with $x(Z_{i+1})x'$. We let, for every $k \leq i$, $y(Z_k)y'$ if, and only if, $v = v'$, and $X \cap (D \times D) = X'$.

Notice that if $x(Z_i)x'$, then $S(x) = S(x')$ and $|S(x)| \leq (n - i)$. Moreover, by consulting Definition 7.17 it is not hard to see that for all $q \in Q_{\mathcal{B}}, q' \in Q_{\rho(\mathcal{B})}$ we have, for all $0 \leq i \leq n$ and all $q \in Q_{\mathcal{B}}, q' \in Q_{\rho(\mathcal{B})}$:

\[
\forall x_1, x_2 \in \gamma(q) : \forall x_1', x_2' \in \gamma(q') : x_1(Z_i)x_2 \Leftrightarrow x_1'(Z_i)x_2'
\]
This means, in particular, that the following lifting of \((Z_i)_{0 \leq i \leq n}\) to models is well-defined, for all \(q \in Q_B, q' \in Q_{\rho(B)}\) and all \(0 \leq i \leq n\):

\[
q(Z_i)q' \Leftrightarrow x(Z_i)x'
\]

for some \(x \in \gamma(x), x' \in \gamma(q')\).

Next we show that \((Z_i)_{0 \leq i \leq n}\) so defined is an n-bisimulation between \(\mathcal{K}_{(B,\varepsilon)}\) and \(\mathcal{K}_{(\rho(B),\varepsilon)}\).

\((Z_i)_{0 \leq i \leq n}\) witnesses n-bisimulation: We address all the points of the definition of n-bisimulation modulo \(\Phi\) in order.

1. Clearly, \((\emptyset, \emptyset)Z_n(\emptyset, \emptyset)\). Hence the first condition of the definition is satisfied.

2. Consider any arbitrary states \(q, q'\) and let \(x = x_1; x_2; \ldots ; x_m\) and \(x' = x'_1; x'_2; \ldots ; x'_m\) be the corresponding nodes from \(T, T'\) that witnesses to \(q(Z_0)q'\). By definition of \(Z_0\) we have \(S(x) = S(x')\), but it is possible that we have \(E(x) \neq E(x')\). However, we must have \(C((A, E)(x), \Phi) = C((A, E)(x'), \Phi)\), and to see this, it is enough to observe that as \(m \leq n\), each of \(x\) and \(x'\) contains at most \(n\) nodes. Then, since \((A, E)(x) = q\) and \((A, E)(x') = q'\) are the same on \(D\), and the distance from \(\Pi\) to \(\Phi\) is greater than \(n\). That is, any path from an argument in \(\Pi \setminus D\) to an argument in \(\Phi = \Pi(\phi)\) would be a path consisting of at least \(n + 1\) nodes. It follows that no element from \(\Phi\) can be in a connected components containing elements outside of \(D\).

3. Consider now \(q, q'\) corresponding to \(x\) and \(x'\) such that \(x(Z_{i+1})x'\). Notice that \((q, r) \in R_B(\Xi)\) if, and only if, there is a \((p, X)\) such that \(xR(x; (p, X))\). So all we need to show is that \(X \cap (D \times D)\) is in \(\cup_{\rho(B)}(x', p)\). Then it will follow that there is a successor to \(x'\), namely \((p, X \cap (D \times D))\), with \((x')R(x'; (p, X \cap (D \times D)))\). This is a straightforward consequence of the Definition 7.22 of \(\rho\). The argument for the particular sub relations \(R_B(p)\) is analogous.

4. Finally consider \(q, q'\) corresponding to \(x\) and \(x'\) such that \(x(Z_{i+1})x'\) for \((p, X)\) such that \(x'R(x'; (p, X'))\). Again we need to ensure that there is an \(X \in \cup_B(x, p)\) such that \(X' = X \cap (D \times D)\), and again this follows from the Definition 7.22 of \(\rho\). The argument for the particular sub relations \(R_B(p)\) is analogous.

\[\Box\]

**Proposition 7.24.** Let \(\phi \in \mathcal{L}_{DDL}\) and \(B, B'\) arbitrary bases. If states \(q \in \mathcal{K}_{(B,\varepsilon)}\) and \(q' \in \mathcal{K}_{(B',\varepsilon)}\) are \(|\phi|^\diamond\)-bisimilar modulo \(\Pi(\phi)\), then \(\mathcal{K}_{(B,\varepsilon)}, q \models \phi \Leftrightarrow \mathcal{K}_{(B',\varepsilon)}, q' \models \phi\).

Or, succinctly

\[
\left( (\mathcal{K}_{(B,\varepsilon)}, q) \overset{\Pi(\phi)}{\leftrightarrow} (\mathcal{K}_{(B',\varepsilon)}, q') \right) \Rightarrow (\mathcal{K}_{(B,\varepsilon)}, q \models \phi \Leftrightarrow \mathcal{K}_{(B',\varepsilon)}, q' \models \phi) .
\]

**Proof.** The proof is by induction on \(|\phi|^\diamond\).
7.5 Recognizing Consensus

**Base case:** \(|\phi\rangle^\diamond = 0\) There are no white connectives, and our states, \(q\) and \(q'\), are clearly 0-bisimilar modulo \(\Phi\). It is also easy to see, consulting Definition 7.12, that the truth of a formula of modal depth 0 is only dependent on the argumentation frameworks \(q\). Then it follows from the fact that \(\varepsilon\) is assumed to be normal that the truth of \(\phi\) is in fact only dependent on \(C(q, \Phi)\). From \(q(Z_0)q'\), we obtain \(C(q, \Phi) = C(q', \Phi)\) and the claim follows.

**Induction step:** \(|\phi\rangle^\diamond > 0\) We skip the boolean cases as these are trivial, so let \(\phi := \Box \psi\) (the case of white connectives with an explicit argument is similar). Suppose \(|\phi\rangle^\diamond = i + 1\) and \(q(Z_{i+1})q'\). Suppose further that \(\mathcal{K}_{(B, \varepsilon), q} = \Diamond \psi\). Then there is a successor of \(q\), \(v \in \text{succ}(q)\) such that \(\mathcal{K}_{(B, \varepsilon), v} = \psi\). All successors of \(q\) will be \(i\)-bisimilar to a successor of \(q'\) (point 3. of Definition 7.20). So we have \((\mathcal{K}_{(B, \varepsilon), v}) \leftrightarrow^{\Phi}_{\Diamond} (\mathcal{K}_{(B', \varepsilon), v'})\). As \(|\psi\rangle^\diamond < |\Diamond \psi\rangle^\diamond\) we can apply our induction hypothesis to obtain \(\mathcal{K}_{(B', \varepsilon), v'} = \psi\), and \(\mathcal{K}_{(B, \varepsilon), q'} = \Diamond \psi\) as desired.

\(\Box\)

7.5 Recognizing Consensus

With these logical tools in place, it is tempting to ask how we could recognize a consensus. Surely, if some argument is accepted beyond refutation in the sense that no further arguments would successfully refute it, then this argument is a candidate for a consensus extension. Suppose we are working with preferred semantics and we have arrived at the state illustrated in (1) of Figure 7.9.

```
(1) (2)
    r1    r2
  p  p1  p  p1
 r1    r2
  p  p2  p  p2
 r2    r3
  p  p3  p  p3
 r3  ...
  p  pn-1  p  pn-1
```

**Figure 7.9:** Is \(\neg p\) part of a consensus?

In (1) of Figure 7.9 our state is the argumentation framework consisting of arguments \(\{p, p_1\}\) containing no edges. Clearly, \(\Box p\) since \(\varepsilon(\{p, p_1\}, \emptyset) = \{\{p, p_1\}\}\). Let us further place this in a context of deliberation between two agents with identical views. There is no disagreement about the relationship between any argument, but only \(p\) and \(p_1\) have been put forward. We could now argue erroneously that a consensus that \(p\) has been established since we have that \(\Box \Box p\) ("no matter which argument is put forth, \(p\) is skeptically accepted"). However, in just two steps, \(p\) may become irrevocably rejected!

\[
\Diamond \Diamond (\Box \neg p \land \Box \Box \neg p \land \Box \Box \Box \neg p \land \Box \Box \Box \Box \neg p)
\]

The number of \(\Box\)-boxes in this formula could be extended indefinitely, without affecting the status of \(p\) as skeptically rejected. In every complete assent, \(p\) is skeptically rejected, however there might be states in which \(p\) is skeptically accepted and no matter what the next \(n\) arguments are \(p\) will remain accepted. That is, there is no single argument which may make \(p\) anything except skeptically accepted. However, there are two arguments \((r_1 \text{ and } r_2)\) which after they have been put forth, \(p\) is skeptically rejected and there is no way to reinstate \(p\) even credulously.
Indeed, we can easily construct views which satisfy this particular property. Suppose the current state of deliberation is based on coinciding views corresponding to (2) in Figure 7.9 contains the arguments \( p, p_1, p_2, \ldots, p_{n-1} \), and none of the \( r_i \) arguments. Then no matter what the next \( n-1 \) arguments are, \( p \) will be skeptically accepted, however, there are \( n \) arguments (i.e., \( r_1, \ldots, r_n \)) such that \( p \) is skeptically rejected and no further argument can reinstate it.

If we are not limited in the views we are considering, we could, for an arbitrary number \( n \) construct a view in which there is a state such that

\[
\Box^n p \land \Diamond^{n+1} \neg p
\]

where by \( \Box^n \) we mean \( n \) successive occurrences of \( \Box \), and similarly for \( \Diamond^{n+1} \).

### 7.6 Summary

We have argued for a logical analysis of deliberative processes by way of modal logic, where we avoid making restrictions that may not be generally applicable, and instead focus on logical analysis of the space of possible outcomes. The deliberative dynamic logic (DDL) was put forth as a concrete proposal, and we showed some results on model checking.

We notice that DDL only allows us to study deliberative processes where every step in the process is explicitly mentioned in the formula. That is, while we quantify over the arguments involved and the way in which updates take place, we do not quantify over the depth of the update. For instance, a formula like \( \Diamond \Box p \) reads that there is a deliberative update such that no matter what update we perform next, we get \( \phi \). A natural next step is to consider instead a formula \( \Diamond \Box^* \phi \), with the intended reading that there is an update which not only makes \( \phi \) true, but ensures that it remains true for all possible future sequences of updates. Introducing such formulas to the logic, allowing the deliberative modalities to be iterated, is an important challenge for future work. Moreover, we would also like to consider even more complex temporal operators, such as those of computational tree logic, or even \( \mu \)-calculus.

Finding finite representations for the deliberative truths that can be expressed in such languages appears to be much more challenging, but we would like to explore the possibility of doing so.

Also, we would like to explore the question of validity for the resulting logics, and the possibility of obtaining some compactness results. Indeed, it seems that if we introduce temporal operators we will be able to express truths on arbitrary points \( q \in Q_B \) by corresponding formulas that are true at \((\emptyset, \emptyset)\), thus capturing the way in which complete assent can be faithfully captured by a finite (albeit unbounded) notion of iterated deliberation.

If the history of the human race is anything to go by, it seems clear that we never run out of arguments or controversy. But it might also be that some patterns or structures are decisive enough that they warrant us to conclude that the truth has been settled, even if deliberation may go on indefinitely. A further logical inquiry into this and related questions will be investigated in future work.
Chapter 8

Conclusions

In Chapter 3 we defined RCGS models, a semantic construction which is central in several of the chapters in this thesis. We argued that model checking ATL formulas over RCGS models is never worse than over CGS models. The idea of utilizing symmetries in CGS models is not novel, and our approach is in some ways similar to refinement relations [7]. Using this notion as a basis for implementing the metaphor of roles is also used in MAS literature [72].

Our treatment of roles based on these symmetries are rather complementary to that found in [72] in the sense that agents belonging to some role in [72] are united by having some common task, whereas our use of the term unite agents in their capacities.

That is, rather than using “role” to describe a group of agents which should express some global behavior but behave individually very differently, our use of “role” describes agents which behave similarly.

We understand a role to be a component of a MAS in which the agent enacting a role is given her abilities to act from her enactment. The police officer can perform an arrest in lieu of her enacting the role of police officer, and the server may relay a message in lieu of enacting the role of server. We posit that this way of reasoning is familiar to humans and is hence a plausible framework for expressing constraints on system principles.

Finding the smallest, or most compact, model is challenging. This problem is addressed in the literature discussing the refinement relations. Our treatment is complementary in the sense that we encourage the system designer to incorporate the notion of homogeneous roles from the first stage of modeling. Specifying the abilities of roles and the number of agents needed in different roles is not a simple task, however.

Our axiomatization of HATL and RATL in Chapter 4 goes some way in aiding the construction of such specifications. The completeness proof is constructive and any consistent specification will produce a model satisfying it.

The relation between HATL and RATL is interesting as well. The fact that reasoning about strategically homogeneous agents is the same as reasoning about agents without a name strengthens the claim that we have also defined an anonymous version of ATL. This form of anonymity is inherited from voting theory, social choice/welfare theory, and game theory. We are not familiar with earlier adaptations in MAS logics. In MAS many questions are expressed in terms of coalitions. The notion of a constellation which we introduced in Chapter 4 is simply removing explicit references to agents, and replacing our characterization of action entities by certain predicates describing the
role membership and multiplicity of the constituting parts (the actual agents).

We feel that the notion of constellations is particularly well suited for expressing properties of systems without explicit reference to the constituting agents. Weakening the assumption we needed to make about a fixed number of agents enacting every role is a challenge we want to pursue in future work.

In Chapter 5 we showed how we may increase heterogeneity in our models by introducing norms (as they are generally defined in MAS literature). This also seems to suggest an interesting solution to the technical challenge left open in the axiomatization; if we embed all agents in one single homogeneous role and introduce differences by letting different agents comply to different norms, we may emulate roles by an associated norm.

This seems to us like a prudent continuation of the work presented in this thesis. For one, it is close to observations made regularly in sociological literature about the related nature of roles in society and norms participants are expected to comply to. Also, this could aid the specification of roles through the synthesis of norms. This is already a well-established problem in game theory and gaining attention in logical investigations of MAS.

Whenever applicable, models of homogeneous roles might prove useful also in designing systems in which compliance can not be guaranteed. We have shown that such systems satisfy a form of monotonicity which simplifies certain problems involving quantification over coalitions. In the logic NCHATL presented in Chapter 5, we showed how we can avoid quantifying over coalition altogether. It seems likely that some problems regarding partial norm compliance which we have not addressed in this thesis, such as robustness, could also benefit from such properties of our representation.

In Chapter 6 we shifted our focus from the space of possible outcomes, towards a basis for reasoning about expected outcomes. The preferences of agents in the framework we presented there were reason-based. This property, as formulated in this thesis, was proposed by Dietrich & List and is one of several theories of property-based preferences. In that chapter, we formulated a theory in a well-known modal logic which let us discuss both the preferences of agents (given a set of motivating properties) as well as how these preferences change as the agent becomes motivated by different properties.

In Chapter 7 we presented DDL, a logic to reason about a particular form of argumentation. We depart from a common assumption of agents having individual agenda when arguing, and rather try to capture the entire set of possible outcomes when agents truthfully put forth arguments based on their subjective view of the matter.

We argued that if a consensus situation is to be recognized as a temporally stable joint understanding/position, then looking only a bounded number of arguments into the development may never be enough unless we place structural constraints on the agents’ views. We do not suggest any such constraints, and are unable with our technique to provide finite methods of determining the outcome of arbitrary numbers of consequitive arguments. Both of these problems, as well as the relationship between DDL and existing suggestions for (concrete) procedures for aggregating agent views, seem like interesting questions to be tackles in the future.

As the reader will soon recall, in Chapter 8 we concluded.
Bibliography


[3] Ågotnes, T., van der Hoek, W., and Wooldridge, M. Quantified coalition logic. Synthese (Special Section on Knowledge, Rationality and Action) 165, 2 (2008), 269–294. 5.6


[26] Dietrich, F., and List, C. A reason-based theory of rational choice. Mpra paper, University Library of Munich, Germany, 2011. 1.3.1, 1.9, 6.1, 8, 6.7

[27] Dietrich, F., and List, C. Where do preferences come from? Mpra paper, University Library of Munich, Germany, 2011. 1.3.1, 1.9, 6.1, 6.2, 2, 3, 1, 6.7


[46] Jamroga, W., and Dix, J. Do agents make model checking explode (computationally)? In *Multi-Agent Systems and Applications IV (LNAI Volume 3690)* (2005), M. Pechoucek, P. Petta, and L. Z. Varga, Eds. 1.6, 2.4.1


[50] Kleene, S. *Introduction to Metamathematics*. Van Nostrand, New York, 1952. 2.5.2, 2.5.2


[60] Oeser, O., and Harary, F. *A Mathematical Model for Structural Role Theory*, I. Tavistock Institute of Human Relations, 1962. 3.4


[63] Pedersen, T., and Dyrkolbotn, S. Computing consensus: A logic for reasoning about deliberative processes based on argumentation (*Presented at 1st International Workshop on Argument for Agreement and Assurance (AAA 2013)*), 2013. 1


[65] Pedersen, T., Dyrkolbotn, S., and Kaźmierczak, P. Big, but not unruly: Tractable norms for anonymous game structures, 2014. 3, 5.1


