Logics for Cooperation, Actions and Preferences

MSc Thesis (*Afstudeerscriptie*)

written by

Lena Kurzen
(born April 21st, 1982 in Mettingen, Germany)

under the supervision of **Dr Eric Pacuit** and **Dr Ulle Endriss**, and submitted to the Board of Examiners in partial fulfillment of the requirements for the degree of

MSc in Logic

at the *Universiteit van Amsterdam.*

Date of the public defense:
September 5th, 2007

Members of the Thesis Committee:
Prof Dr Johan van Benthem
Prof Dr Peter van Emde Boas
Dr Ulle Endriss
Dr Eric Pacuit
Olivier Roy

INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

Abstract

In this thesis, a logic for reasoning about cooperative ability, actions and preferences is developed. It is an extention of a cooperation logic with actions, developed by Sauro et al., which is a modular modal logic consisting of an environment module for reasoning about actions and their effects and an agents module for reasoning about the cooperative ability of agents to perform actions. In this thesis, that logic is combined with a preference logic with unary preference modalities. In the resulting logic, we can reason about the abilities of groups to enforce some state of affairs in an explicit way: It is explicitly represented how exactly a group can achieve some state of affairs and how this achievement relates to the preferences of single agents. It is shown that the developed cooperation logic with actions and preferences is sound and complete with respect to the class of multi-agent systems with preferences, which are set-labelled transition systems with a preference relation over the set of states and an attached model in which it is specified which actions groups of agents can perform.

The cooperative ability of agents in this framework is investigated in detail. It is shown what is the relation between the actions single agents can perform and the role the agents play within a group when the group is trying to enforce some state of affairs.

Moreover, it is shown that the semantic structures of the logic provide a formal framework for mechanism design. Apart from standard game theoretic results, we obtain results that show the relationship between the distribution of action abilities among agents and the properties of implementable choice rules.

Contents

Chapter 1

Introduction

The concept of cooperation between agents plays a major role in many fields since the formation of coalitions or groups occurs in all kinds of situations where there is more than one agent.

Therefore, the notion of cooperation has received a lot of attention within various fields such as economics, political science, social sciences and game theory. One motivation for agents to cooperate can be that acting as a coalition is more profitable for them than acting individually, i.e. as a group they can achieve something 'better' than alone. Processes of cooperative behavior consist of several different stages that are worth being investigated. First of all, it is interesting to examine how coalitions form in the first place, i.e. which agents join which coalitions and what are their motivations for doing so. On the other hand, we can also take coalitions as primitives. Then instead of looking at how and why coalitions form, we can investigate how different groups of agents interact and which results groups can collectively achieve. This has been investigated in detail in the field of cooperative game theory [Osborne and Rubinstein, 1994]. When talking about the cooperative ability of coalitions, this is usually done in terms of the results or *outcomes* that coalitions can achieve. This kind of analysis of an interactive situation in a multi-agent system reveals *what* each of the possible coalitions can achieve.

Let us look at what it means for a group to have the ability to achieve some result. Intuitively, this means that the agents in the group together have some way of ensuring that the interactive process they are in leads to this result. If we want to make this more explicit, we would say that the group has some strategy or plan that the members can follow such that it yields the desired result. Such a plan or strategy tells the group exactly how to act in order to achieve the desired outcome. These considerations show that actions and their effects also play a crucial role in the investigation of cooperative ability because when trying to describe coalitional power more explicitly this leads us directly to the actions that (groups of) agents can perform in order to achieve some outcome. It seems clear that an investigation of cooperative ability that also takes into account the actions that (groups of) agents can perform yields to deeper insights into the cooperative process since it reveals *how* coalitions can achieve certain outcomes.

Another interesting aspect that arises once we look at agents collectively acting as groups and at the interaction between groups is the following. There can be several ways how a group can cooperate and several different outcomes that they can achieve. Whenever we consider situations where single agents or groups are facing a set of alternatives (e.g. outcomes or actions they can perform) they can or must choose from, a naturally arising issue is that of the preferences of the agents over those alternatives. Looking at the preferences the members of a group have over the results the group can achieve can give us some information as to *why* a group would decide to enforce some result rather than another.

1.1 Logics for Cooperation, Actions and Preferences

From the above considerations, it seems clear that when investigating interactive situations with multiple agents it is of interest

- *what* agents can achieve when cooperating as a group,
- *how* they can achieve something,
- *why* they would want to achieve something.

Therefore, we claim that a formal investigation of cooperative ability in such systems should take these considerations into account.

Although several logical frameworks have been developed for reasoning about cooperative ability [Pauly, 2002], actions and their effects [Harel, 1984] and preferences [van Benthem et al., 2007] separately and there has been some recent work combing two of the three aforementioned concepts [Sauro et al., 2006; Borgo, 2007; Ågotnes et al., 2006, 2007a], as far as we know there has not been any formal framework developed in which all three can be investigated individually and also the connection between them. To sum up, the objective of this thesis is to examine how such a framework can be developed and also to present one possible way of doing it.

In this thesis, we do not develop a new logic for reasoning about cooperation, actions and preferences from scratch but we will use known logics and combine them. More precisely, we take the cooperation logics with actions (CLA) which has been developed by Sauro et al. [2006] and combine it with a preference logic [van Benthem et al., 2005; van Benthem et al., 2007]. The resulting logic can be used for reasoning about the cooperative ability in multi-agent systems and also allows for saying explicitly how a group can achieve some states of affairs. Moreover, it also relates cooperative ability to agents' preferences; we can reason about the ability of a group to achieve the truth of some formula and also to achieve that the next transition that the system makes is an improvement according to the preference relation of some agent. The modular approach that we follow then also allows for independent investigation of the agents cooperative abilities to perform actions, actions and their effects and the agents' preferences over states of the environment. By using the soundness and completeness of the logics of the individual submodules, soundness and completeness of the cooperation logic with actions and preferences can then also be shown.

Our main focus does not lie on the technical properties of the logic we develop. Rather, we concentrate on clarifying some conceptual issues that arise in the framework that we develop and focus on investigating which particular aspects of game-like situations in multi-agent systems we can model in our framework.

One topic from game theory that we investigate in our framework is mechanism design, which is also known as *implementation theory* [Osborne and Rubinstein, 1994]. The idea of mechanism design is the following. There is a set of players and a planner. The planner wants to implement a particular choice rule. Such a choice rule determines for every possible preference profile of the players a set of outcomes. Then it is the job of the planner to design a game in such a way that – assuming that all the players choose their actions according to some solution concept – no matter what the preferences of the players are, the possible outcomes are exactly those selected by the choice rule.

The results in the field of mechanism design that have been obtained in game theory are mainly about the relationship between the solution concept according to which the agents are assumed to play and the properties of choice functions that are implementable in such solution concepts. We show that in our framework, we can obtain similar results that are more fine grained. They do not only clarify the relationship between the solution concepts according to which the agents make their decisions and the properties of implementable choice rules but also clarify the connection to how the ability to perform certain actions is distributed among the agents. So, results that we obtain are e.g. of the form: *If the ability to perform actions is distributed among the agents in a certain way, then every $\mathcal{S}$-implementable choice rule (for $\mathcal{S}$ being a solution concept) has property P.*

Considering the cooperative ability of a group to force some state of affairs, a question that comes up is the following: How important is some member for achieving this state of affairs? Could the group also force it without him? In this thesis, we also take a look at this question and show that in a cooperation logic with actions as it is proposed by Sauro et al. [2006] this question can be investigated in an explicit way. This is done by comparing the sets of actions that a single agent can perform to the sets of actions that a group can perform that he is a member of. Such a comparison then reveals whether the agent is needed for the group to be able to achieve a certain state of affairs.

Moreover, we also investigate issues from cooperative game theory within our framework. In particular, we will look at the solution concept of the core [Osborne and Rubinstein, 1994]. The core of a cooperative game is the set of outcomes that the agents can achieve by cooperating all together in such a way that there is no group that can achieve something by itself that is strictly better for all its members.

Whereas a direct analog of the core does not seem to be characterizable in our framework, we can characterize a slightly different but closely related concept. The cooperation logic with actions and preferences allows us to characterize the property that in a state no coalition has the power of making the system move to a state that is strictly preferred over the current one by all the members of the group.

1.2 The Structure of the Thesis

The remainder of this thesis is structured as follows.

Chapter 2 gives an overview of some of the work that has been done to develop formal frameworks for reasoning about the cooperative ability of agents in multi-agent systems.

In Chapter 3, we present the cooperation logic with actions [Sauro et al., 2006] in greater detail. This logic provides a framework for reasoning about the cooperative ability of groups of agents in an explicit way. It is a modular approach based on an environment module for reasoning about the effects of actions and an agent module for reasoning about the cooperative ability of groups of agents to perform actions. Besides describing this logic, we also investigate the cooperative ability of agents in greater detail by examining how a group can minimize their uncertainty about what exactly are the effects of the actions that they choose to perform. Moreover, we show how comparing the actions that a single agent can perform to the actions a group can perform that he is a member of can give us some insights into how important the agent is for the group for achieving some state of affairs.

In Chapter 4, we give an introduction to preference logics and develop a logic for reasoning about the coalitional power of groups of agents, actions and their effects and preferences. Technically, this is achieved by combining the cooperation logic with actions described in the Chapter 3 with a preference logic [van Benthem et al., 2007]. We also sketch how completeness of this logic can be shown.

Chapter 5 then shows how problems from the field of mechanism design can be investigated in the semantic structures developed in Chapter 4. Several results are obtained that show how the way the ability to perform actions is distributed among the agents effects which kinds of choice rules can be implemented in the respective multi-agent system.

In Chapter 6, we look at the connection between the framework we develop and cooperative games. We take a look at how concepts similar to the core of a cooperative game can be characterized in our logic.

Chapter 7 concludes the thesis and gives an outline of future work.

Chapter 2

Background: Reasoning about Strategic Cooperative Ability

Cooperative ability or *coalitional power* are concepts used for describing the ability of groups of agents to bring about something by acting collectively as a group.

There are different kinds of abilities. The majority of formal systems that have been developed for modelling cooperative ability focus on *α-ability*, which is also referred to as *α-effectivity* [Abdou and Keiding, 1991]. If a coalition C is α-effective for φ, this means that there is some collective strategy for C such that if the agents in C follow this strategy then φ will be true no matter what the other agents, i.e. the ones that are not in C do. So if C is α-effective for φ then C can achieve the truth of φ irrespectively of what the other agents are doing. α-ability is also referred to as $\exists\forall$ ability, since this is the combination of quantifiers used in the definition of α-ability.

Cooperative ability in general can be viewed on different levels. On an abstract level, we can describe it by only considering the results coalitions can achieve and not taking into account how exactly they can achieve those results. In a more explicit way, cooperative abilities of groups can be described by also saying how the results can be achieved. Moreover, apart from knowing what a group can achieve and how exactly the agents can achieve some result, we might also be interested in *why* a group would want to achieve a certain result. This then leads us to the preferences the agents have over possible results they can achieve.

Let us first look at the most abstract of these levels where we describe the coalitional power by only looking at the results that groups can achieve. In general terms, a result is some state of affairs. When investigating the cooperative ability of groups to achieve some state of affairs, this can be done in two directions. Given some coalition C, it might be interesting to find out what abilities C has, i.e what results C can achieve. Conversely, for a given state of affairs φ it can also be interesting to investigate which groups can achieve it. In both cases, we have statements of the form

Coalition C can bring about φ.

Suppose that two players are playing a board game consisting of several rounds and somebody tells them that Player 1 has a winning strategy. So, the group consisting of only Player 1 has the ability of winning the game. Player 1 will surely be happy of hearing that he can win but this information is not very useful for him unless he knows a winning strategy. This example illustrates that there are situations in which it is rather of interest to find out *how* an agent can bring about some state of affairs than just to determine whether he has the ability to enforce this state of affairs. It is easy to imagine similar examples that consider the ability of a group agents instead of the ability of an individual agent. So, a more explicit way of investigating the cooperative ability of groups of agents is by considering statements of the form.

Coalition C, by doing such and such, can bring about φ.

Here, "doing such and such" can just be the description of a single action that the group can perform or it can also be in form of a complex plan the group can execute that involves sequences of actions that depend on each other. Again, there are many different aspects that are of interest here: It can for instance be interesting to find out what states of affairs a group can achieve by one particular action or plan or what are the different ways how a group can force some state of affairs etc.

There is also another aspect of coalitional power that is of interest. It is related to the agents' motivation for forcing some state of affairs. A group having the ability to enforce φ does not mean that the group would actually do it because it might be the case that its members would rather prefer doing something else. Of course, it is not clear at this point what exactly it means for a group to prefer something. However, it seems intuitively reasonable to investigate cooperative ability by also taking into account the preferences that the agents have over possible outcomes of the interactive process. Such an investigation then also gives a close connection to the way interactive situations are modelled and investigated in game theory.

In the remainder of this chapter, we will present various frameworks that have been developed for formal investigations of cooperative ability. The approaches model cooperation on different levels and focus on different aspects of cooperative behavior. In Section 2.1, we will give an outline of some of the fundamental work that has been done in order to formalize reasoning about the cooperative abilities by focussing only on the ability to bring about certain states of affairs. Then in Section 2.2 we present some approaches that directly followed and added some new aspects to their formal models in order to make the coalitional power more explicit or that model cooperation from a slightly different perspective. In Section 2.3, we will give an overview of some of the attempts that have been made to incorporate a representation of the agents' actions and plans.

2.1 Cooperation Logics

In this section, we will present some formal systems for reasoning about the cooperative ability or effectivity of groups of agents to bring about some state of affairs. The systems presented in this section are based on modal logics containing a modal operator for each set of agents. These modalities are used to express that the corresponding group of agents has the ability to enforce that the formula following the operator is satisfied. So, the basic idea is that $\langle C\rangle\varphi$ says that the coalition C can force that φ is true.

2.1.1 Alternating-time Temporal Logic

Alternating-time temporal logic (ATL), which was developed by Alur, Henzinger and Kupferman [Alur et al., 1998], plays an important role in the field of cooperation logics and many of the recently developed cooperation logics are based on ATL or some fragment of it and use similar ideas. We will not go into the technical details here but just concentrate on the general idea of the logic and the way how cooperation is modelled in the ATL framework.

Originally, ATL was developed as an extension and generalization of the computation tree logic CTL [McMillan, 1992]. CTL is a tense logic based on models of branching time. It is a modal logic containing temporal modal operators for *next* and *until*. Additionally, it has universal and existential quantifiers for quantifying over the set of paths originating from the current state. Thus, sentences of the following form can be expressed: *There is some path in which φ is true until ψ is true.*

The idea underlying ATL is that the path quantifiers in CTL that universally and existentially quantify over paths can be replaced by so called *cooperation modalities*. These modalities are of the form $\langle C\rangle$, where $\langle C\rangle\varphi$ says that the agents in C can cooperate such that they can force that φ is true of the resulting computation. The development of ATL was rather technically motivated than by the aim to develop formal system for reasoning about cooperative abilities of agents.

ATL is interpreted in concurrent game structures which are used for modelling the composition of open systems. In general terms, an open system is a system whose behavior is not only determined by its current state but also depends on how the environment behaves. In short, a concurrent game is a game that is played on a state space. This means that there are several states of the game and in each step during the play, each of the participating players chooses one of the moves that are available to him at the current stage of the game. Then the transition of the system from one state to another is determined by the choices of moves of all the players; but not every state need to be accessible from a certain state. So, in each state, depending on which moves the players choose, the system moves to one of the successor states. The choices of all the players together completely determine which of the accessible states will be the next one. So, here it becomes visible why concurrent game structures are used as models of open systems. In each state, the behavior of the system, i.e. where it will move next, is determined by the current state (which determines which states are possible successor states) and by the behavior of the agents who determine which of the accessible states will be the one the system moves into next.

A special case of concurrent game structures are turn-based game structures, where at each state the next state is determined by the move of a single player only.

As we already mentioned, ATL extends the computation tree logic which contains temporal modalities such as the unary modalities 'next' and 'always' and the binary modality 'until'. Those temporal modalities are also part of ATL.

Assume that we have a set of players N and let $C \subseteq N$ be a group of agent. Then the formula $\langle C \rangle \bigcirc \varphi$ says that there are strategies for all the agents in C such that no matter which sequences of moves the agents in $N \setminus C$ choose, the *next* state the system will move into, will satisfy φ. $\langle C \rangle \Box \varphi$ says that there are strategies for all the agents in G such that no matter what the other agents do, all states along the resulting path of the computation will satisfy φ, i.e φ will *always* be true. $\langle C \rangle \varphi \mathcal{U} \psi$ means that there is a strategy for the group C such that along any resulting path there is some state that satisfies ψ and until that state has been reached all the states satisfy φ, i.e. until ψ holds φ is always true.

Now, the relationship to CTL is easy to see. Remember that in CTL we have a universally and an existentially quantifying modality ranging over the set of paths leaving the current state. The existential one can be expressed in ATL by $\langle N \rangle$ since $\langle N \rangle \varphi$ means that there is a strategy for the grand coalition such that φ will be true of the resulting computation path. Since the strategies of all the players together completely determine the path that the computation will follow from the current state, this is then equivalent to saying that there is a path satisfying φ.

Analogously, the universally quantifying modality can be expressed by $\langle \emptyset \rangle$ since $\langle \emptyset \rangle \varphi$ means that no matter what the agents in N are doing, the resulting computation path will satisfy φ. This then also means that all the paths will satisfy φ.

Summarizing, we can say that ATL provides a framework for reasoning about the abilities of coalitions to achieve some state of affairs. Moreover, the state of affairs can have a temporal component and therefore the coalitional power of groups can be further differentiated as it can e.g. expressed that a group has the power of forcing φ to be true at the next state, always in the future, or until the moment when some ψ becomes true.

2.1.2 Coalition Logic

Coalition Logic (CL), which was developed by Marc Pauly in [Pauly, 2002], is a modal logic for reasoning about what coalitions of agents can bring about together by collectively performing actions.

The way how coalitional power is modelled in CL turns out to be closely related to ATL but as opposed to ATL, the development of CL was primarily motivated by the aim to build a logical framework for investigating coalitional power in game-like situations on a formal level.

In the simplest case of such scenarios, there is one single agent. This agent can then choose between performing different actions which have the effect of changing the state of the world. This scenario can be modelled as a set of states equipped with a binary accessibility relation R. For each state s: sRt for all the states t such that the agent can act in such a way that the resulting state will be t. In such a structure, we can reason using the language of modal logic, where the formula $\Diamond\varphi$ then means that the agent can act in such a way that the resulting state will be one where φ is true.

This framework can be extended to a system with more than one agent. Here, it is important to take into account that the actions of different agents are in general not independent from each other. Let us consider the case where we have two agents. In a given state s, every possible pair of actions has to be considered that can occur when each of the agents chooses one of the actions he can perform. Then, the performance of each of those combinations of actions will lead to some state t.

This already gives us the basic idea of the approach to modelling coalitional power as it is presented in [Pauly, 2002]. The semantic structures underlying the models for CL are so called *game frames*, which in fact are really similar to the semantic structures of ATL. Assume that in the scenario that we want to model, there are n agents. Then a game frame consists of a set of states and in each state s the next state is determined by the actions that each of the n agents takes at s. Viewing the interactive process this way, we can think of a strategic game form being associated with each state, where each outcome of the game corresponds to some accessible state.

The idea of modelling interactive processes in multi-agent systems this way is quite important as we will see in the remainder of this chapter and also in the remainder of this thesis where we will come across more approaches that are all based on very similar ideas.

Formally, game frames are defined as follows.

Definition 2.1 (Game Frame [Pauly, 2002]). *Let N be a set of players and S a nonempty set of states. Then a game frame is a pair (S, γ), where γ is a function*

$$\gamma : S \to \mathbf{\Gamma}_S^N.$$

$\mathbf{\Gamma}_S^N$ denotes the set of all strategic game forms of games among the players N over the set of states S.

Note that the definition of a game frame implies that in each state there is some game form, so everywhere in the frame some game can be played and therefore there are no final states, where there is no way to proceed.

The main aim of CL is to provide a formal framework for reasoning about coalitional ability. This is achieved by using the concept of *effectivity.*

In a state s, we say that a coalition $C \subseteq N$ is effective in achieving a set of states $X \subseteq S$ if and only if the coalition has a joint strategy such that them playing according to this strategy will lead to a state in X no matter what the other players do. Recall that this concept of effectivity has been investigated in the social choice literature under the name of α-effectivity [Abdou and Keiding, 1991].

Before giving the formal definitions that show how effectivity is formalized in CL, we define some notation.

Given a game form $\Gamma = \langle N, \{\Sigma_i | i \in N\}, o, S\rangle$, let $\sigma_C := \Pi_{i \in C}\sigma_i$ denote the strategy tuple of coalition C that arises from player i choosing strategy $\sigma_i \in \Sigma_i$. Write $\bar{C}$ to denote the set of players that are not members of the coalition C, i.e. $\bar{C} = N \setminus C$. Then, let $o(\sigma_C, \sigma_{\bar{C}})$ denote the outcome that arises from the strategy profile where the agents in C play according to σ_C and the other ones according to $\sigma_{\bar{C}}$.

Definition 2.2 (Effectivity Function [Pauly, 2002]). *Given a game G, its effectivity function $E_G : 2^N \to 2^{2^S}$ is defined as follows.*

$$X \in E_G \quad \textit{iff} \quad \exists \sigma_C \forall \sigma_{\bar{C}} o(\sigma_C, \sigma_{\bar{C}}) \in X$$

So, an effectivity function assigns to every coalition the sets of states for which it is effective.

In order to model the coalitional power of players in an appropriate way there are several properties that we might want an effectivity function to satisfy. Pauly [2002] focusses on the properties that characterize effectivity functions that belong to strategic games.

Definition 2.3 (Playability [Pauly, 2002]). *An effectivity function is called playable if it satisfies all of the following properties.*

1. $\forall C \subseteq N : \emptyset \notin E(C)$
2. $\forall C \subseteq N : S \in E(C)$
3. *E is N maximal, i.e.* $\forall X \subseteq S : (S \setminus X) \notin E(\emptyset) \Rightarrow X \in E(N)$
4. *E is outcome monotonic, which means that for all $X \subseteq X' \subseteq S$: if $X \in E(C)$ then $X' \in E(C)$*
5. *E is superadditive: For all $X_1, X_2 \subseteq S$ and $C_1, C_2 \subseteq N$ such that $C_1 \cap C_2 = \emptyset$: if $X_1 \in E(C_1)$ and $X_2 \in E(C_2)$ then $X_1 \cap X_2 \in E(C_1 \cup C_2)$*

The first property says that no coalition can be effective for the empty set. The second one says that every coalition is effective for S, the set containing all the states. N-maximality says that if the empty coalition is not effective for the complement of some set X, then the grand coalition has to be effective for X. So, if the empty coalition cannot force the outcome to *not* be in X, then all agents together should be able to force the outcome to be in X. Outcome monotonicity means that if a coalition can force the outcome to be in some set X, then it can also force it to be in any superset of X. Finally, superadditivity describes how disjoint coalitions can join forces. If one coalition can force the outcome to be in X_1 and another one can force it to be in X_2, then together they can force it to be in the intersection $X_1 \cap X_2$. The concept of superadditivity plays a major role when investigating coalitional power since it gives insight into how the cooperative ability changes when new coalitions are formed.

Pauly shows that the property of playability exactly characterizes the class of effectivity functions that belong to a strategic game.

Theorem 2.4 ([Pauly, 2002]). *An effectivity function is playable if and only if it is the effectivity function of some strategic game.*

For a proof, the reader is referred to [Pauly, 2002]. While the right to left direction is straightforward, the other one is more involved since it requires the construction of a game. The importance of this theorem is that it says which exactly are the effectivity functions that we should restrict our investigation to if we want to consider the coalitional power in scenarios that can be modelled as game frames as they have been defined above.

After presenting the considerations that underlie the semantics of CL, we now present the language that is used for reasoning about coalitional power. The language of CL is a modal language with one modality for each group of agents expressing the group's ability to achieve a certain state of affairs. We will use the following notation: For C being a group of agents, we will use $\langle C\rangle$ to refer to the modality that Pauly writes as $[C]$ since it fits better with the notation used in other cooperation logics that we will present in this thesis.

Definition 2.5 ([Pauly, 2002])**.** *Let Φ_0 be a set of propositional letters and N be a finite set of agents. Formulas of CL are generated by the following grammar.*

$$\varphi ::= \quad \top \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle C\rangle\varphi,$$

where $p \in \Phi_0$ and $C \subseteq Ag$.

$\langle C\rangle\varphi$ has then the intended meaning that the coalition C can force the outcome to be some state where φ is true. As semantic structures, so called *coalition frames* are used which are sets of states with playable effectivity functions assigned to each state.

Definition 2.6 (Coalition Frame, Coalition Model [Pauly, 2002])**.** *A coalition frame is a pair $F = \langle S, E\rangle$ where S is a nonempty set of states and*

$$E : S \to (2^N \to 2^{(2^S)})$$

maps every state s to some playable effectivity function $E(s)$. A coalition model $M = (F, V)$ is then obtained by adding a valuation function $V : \Phi_0 \to 2^S$ to a coalition frame.

So, the models in which formulas of CL are interpreted consist of a set of states and a playable effectivity function attached to each state. This effectivity function then describes the coalitional power of the agents in that state by specifying for which sets of states groups are effective.

By Theorem 2.4, coalition frames directly correspond to game frames.
The propositional formulas and Boolean combinations are interpreted in a coalition model in the standard way and for $\langle C\rangle$ we have:

$$M, s \vDash \langle C\rangle \quad \text{iff} \quad \varphi^M \in E(s)(C),$$

where $\varphi^M := \{s \in S | M, s \vDash \varphi\}$.

In [Pauly, 2002], the following axiom scheme is presented.

$$\begin{array}{ll}
(\bot) & \neg\langle C\rangle\bot \\
(\top) & \langle C\rangle\top \\
(\mathrm{N}) & \neg\langle\emptyset\rangle\neg\varphi \to \langle N\rangle\varphi \\
(\mathrm{M}) & \langle C\rangle(\varphi\wedge\psi) \to \langle C\rangle\psi \\
(\mathrm{S}) & (\langle C_1\rangle\varphi \wedge \langle C_2\rangle\psi) \to \langle C_1\cup C_2\rangle(\varphi\wedge\psi) \\
 & \text{where } C_1\cap C_2 = \emptyset
\end{array}$$

The rules of inference are modus ponens and equivalence.

$$\frac{\varphi,\ \varphi\to\psi}{\psi} \qquad\qquad \frac{\varphi\leftrightarrow\psi}{\langle C\rangle\varphi\leftrightarrow\langle C\rangle\psi}$$

Note that the above axioms directly correspond to the five conditions for playability.
For N being a set of players, let CL_{N} denote the smallest coalition logic for N which is generated by the above axioms. It is shown that CL_N is sound and complete with respect to the class of coalition models for N [Pauly, 2002]. We will not present the proof here.

In order to see the connection between CL and classical modal logics, let us take a brief look at CL_1 the smallest coalition logic for only one player. Let us call this player 1. If we write the modality $\langle\{1\}\rangle$ as $\Box$, then it is easy to see that the resulting logic is just the normal modal logic KD which is the normal modal logic of serial Kripke frames. CL for more than one player does not correspond to any normal modal logic but to non-normal modal logics.

Comparing CL to ATL as we presented it in the previous section, we observe that they are closely related. This relationship has been investigated in detail in [Goranko, 2001] where it is shown that CL corresponds to the *next time* fragment of ATL. Both ATL and CL are modal logics whose main cooperation modalities are nonstandard in the sense that they are defined using the quantifier combination $\exists\forall$ instead of just one of the quantifiers as the standard modalities $\Box$ and $\Diamond$ of the basic modal language. Consequently, also the duals of the cooperation modalities are not very intuitive. In CL, $\neg\langle C\rangle\neg\varphi$ says that for every strategy of C the other agents (i.e the group $N\setminus C$) can respond in such a way that the resulting state will satisfy φ.

Moreover, ATL and CL model coalitional power in an implicit way meaning that it is not explicitly represented where the power comes from or how a group can actually achieve some state of affairs.

2.2 More Cooperation Logics

The development of CL was a substantial and influential contribution to the field of formal reasoning about cooperative ability in multi-agent systems. There are several approaches that follow the same or very similar ideas as CL. Most of them are extensions of CL aiming at modelling some aspect of cooperation that is not accounted for in CL. Some approaches, e.g the first approach that we will present now [Ågotnes et al., 2006], also investigate cooperation from a viewpoint different from that taken in CL. In this section, we will give an outline of some of those works. For each of the logics presented in this section, we will give an outline of the motivation, its central ideas, its similarities and differences to CL or ATL and also the viewpoint from which the issue of cooperation is investigated.

2.2.1 Coalitional Game Logic (CGL)

One important feature of many cooperation logics like CL and ATL is that they are closely related to (formal) games in the sense that they provide a formal framework for reasoning about strategic cooperative ability in game-like situations. The relationship to formal games is mostly visible when considering the semantic structures of the logics: It has been shown that the models of CL can be seen as extensive games of almost perfect information [Pauly, 2001]. Moreover, when investigating cooperation logics, the property that game-theoretic solution concepts can be characterized in the cooperation logic is usually used as an argument for the logic to formalize interactive processes in multi-agent systems in an appropriate way [van der Hoek et al., 2005].

Despite the close connection between the semantic structures of CL and ATL and the structures that are used to formalize games, preferences are not represented in the logics whereas they do play a central role in game-like interactive scenarios. Of course, the aim of CL is only to provide a formal framework for reasoning about the cooperative ability to bring about certain states of affairs. So, the interactive process is investigated at a level where it is only of interest which results different groups of agents can achieve and not what are the preferences of the agents over those results. Nevertheless, since it is claimed that CL can model game-like situations, the relation between the models for CL and formal games has to be investigated in greater detail.

This can be seen as the motivation for the development of Coalitional Game Logic (CGL) [Ågotnes et al., 2006]. The aim of this approach is to develop a logic such that the connection to cooperative game theory is not left implicit but in fact the logic can be directly interpreted in coalitional games without transferable payoff.

At first glance, CGL seems to be similar to CL and ATL in the sense that it also uses cooperation modalities for expressing the cooperative ability of groups of agents. However, taking a closer look shows that CGL differs from CL in several aspects: Unlike CL it does represent the agents' preferences in an explicit way and it can be directly interpreted in coalition games without transferable payoff which is shown to not be the case for CL [Ågotnes et al., 2006].

Definition 2.7 (Coalitional Game without Transferable Payoff)**.** *A coalitional game without transferable payoff is a tuple $\langle Ag, \Omega, \mathcal{V}, \preceq \rangle$, where*

- *Ag is a finite set of players,*
- *Ω is a set of consequences (or outcomes),*
- *$\mathcal{V}$ is a function $\mathcal{V} : (2^{Ag} \setminus \emptyset) \rightarrow 2^{\Omega}$ assigning to every nonempty group of agents (a coalition) a subset of the consequences.*
- *$\preceq$ is a preference profile of the agents in Ag over the consequences, i.e. for every agent $i \in Ag$ we have a preference ordering $\preceq_i$ over the set of consequences.*

The function $\mathcal{V}$ is called the characteristic function of the game. It characterizes each coalition by a set of outcomes. In [Ågotnes et al., 2006], it is assumed that the set of outcomes is finite. So, we have a set of outcomes $\Omega = \{\omega_1, \dots, \omega_m\}$. The language that is used can be divided into two parts; an outcome language and a cooperation language. The outcome language is build from Boolean combinations of outcome symbols; there is one outcome symbol for each outcome of the game.

$$\varphi_o ::= \quad \sigma_\omega \mid \neg\varphi_o \mid \varphi_o \vee \varphi_o$$

The cooperation language is defined as follows

$$\varphi_c ::= \quad (\sigma_\omega \preceq_{\sigma_i} \sigma_{\omega'}) \mid \langle\sigma_C\rangle\varphi_o \mid \neg\varphi_c \mid \varphi_c \vee \varphi_c$$

Note that in both languages, a symbol of the form σ_x is intended to correspond to the component x of the coalitional game under consideration.

Formulas of the outcome language are interpreted in outcomes of the game.

$$\begin{array}{lll} \Gamma, \omega \models \sigma_{\omega'} & \text{iff} & \omega' = \omega \\ \Gamma, \omega \models \neg\varphi & \text{iff} & \Gamma, \omega \not\models \varphi \\ \Gamma, \omega \models \varphi \vee \psi & \text{iff} & \Gamma, \omega \models \varphi \text{ or } \Gamma, \omega \models \psi \end{array}$$

Formulas of the cooperation language are interpreted in the coalitional game.

$$\begin{array}{lll} \Gamma \models (\sigma_\omega \preceq_{\sigma_i} \sigma_{\omega'}) & \text{iff} & \omega \preceq_i \omega' \\ \Gamma \models \langle\sigma_C\rangle\varphi & \text{iff} & \exists\omega \in \mathcal{V}(C) \text{ such that } \Gamma, \omega \models \varphi \\ \Gamma \models \neg\varphi & \text{iff} & \Gamma \not\models \varphi \\ \Gamma \models \varphi \vee \psi & \text{iff} & \Gamma \models \varphi \text{ or } \Gamma \models \psi \end{array}$$

So, a coalition can achieve φ if and only if the set of outcomes that the characteristic function $\mathcal{V}$ assigns to the coalition C contains an outcome satisfying φ. In [Ågotnes et al., 2006], a complete axiomatization of the logic of coalitional games is given and it is shown that several solution concepts of cooperative games can be characterized in this logic. Moreover, the differences between the semantic structures of CGL and CL are investigated in detail. It seems tempting to think that also CL can be interpreted in coalitional games by taking the outcomes to be the state of the model. This is however not the case.

It is shown that only in the case of *limited* coalitional games, it is possible to find a coalition model of CL that is outcome equivalent to a coalitional game [Ågotnes et al., 2006]. In limited coalitional games there is one outcome such that only the grand coalition can achieve an outcome different from this outcome. Such games are not very interesting and certainly model interactive processes in multi-agent systems only in very special cases.

As concluding remarks of this section, we can say that CGL is a logic developed to provide a formal framework for reasoning about the cooperative ability of groups of agents in coalitional games without transferable payoff. Comparing the models of cooperation logics like CL and ATL to cooperative games without transferable payoff shows that even though CL seems to provide a framework for reasoning about the coalitional abilities in game-like situations at a level where it is only of interest what are the abilities of groups to achieve certain results, the relation between CL and cooperative games seems more complex as it is not possible to interpret CL in coalitional games without transferable payoff in a straightforward way.

2.2.2 Quantified Coalition Logic (QCL)

The development of Quantified Coalition Logic (QCL) [Ågotnes et al., 2007b] was motivated by the fact that in cooperative scenarios quantification over coalitions is needed in many cases. In order to model certain social procedures it can be useful to be able to express e.g. that a coalition can achieve a certain result if and only if more than half of the agents are members of the coalition. Not only the size of a coalition is of importance in many situations but it also matters who exactly are its members. In some cases, we might want to express that a coalition can achieve some state of affairs φ only if agent i is a member of this coalition.

In cooperation logics as ATL, CL and CGL, this can only be expressed by a formula of exponential length which explicitly says for each of the possible coalitions whether it can achieve the state of affairs under consideration or not. Putting it in more general terms, we would like to be able to express something like: *'There exists a coalition satisfying property P that can force φ'* and: *'Every coalition that has property P can force φ'* without getting a formula that is of exponential length.

Syntactically, QCL is obtained by adding two unary predicates *subseteq* and *supseteq* to the language that express that one coalition is a subset (superset) of another one. Adding such a restricted kind of quantification to CL does not make the logic more expressive but indeed exponentially more succinct [Ågotnes et al., 2007b].

What we get by adding the two predicates *subseteq* and *supseteq* can be thought of as $2^{|Ag|}$ many propositions of the form $subseteq(C')$ and $subseteq(C)$ which can then be interpreted in coalitions. This is done as follows.

$$C \models subseteq(C') \text{ iff } C \subseteq C'$$
$$C \models supseteq(C') \text{ iff } C \supseteq C'.$$

Boolean combinations are interpreted in the standard way.

Using the two predicates, a number of other predicates can be defined for expressing other properties such as two coalitions being disjoint, a certain agent being a member of the coalition or a coalition being nonempty.

In [Ågotnes et al., 2006], a complete axiomatization of the coalition logic with added quantification is given.

QCL provides a compact (exponentially more succinct than CL) way of formalizing e.g. majority voting and therefore appears to be quite appealing for modelling interactive situations in multi-agent systems. It seems that the idea used in QCL might also work for other cooperation logics such as CGL.

2.2.3 Explicit Representation of the Origin of Coalitional Abilities

CL and ATL are somehow based on the assumption that the origin of the coalitional power, i.e. the ability of a coalition to achieve something is implicitly represented in the semantic structures of the logic. In what follows, we will focus on approaches that aim at giving a more explicit representation of where the power of a coalition is coming from. There are several ways to think about this. One possibility is to think of the coalitional power of a group to achieve the truth of some formula φ to be the result of the abilities of the members of the group to achieve the truth of certain subformulas of φ. Another view is to say that the power of a group to achieve φ comes from its ability to collectively perform some action that leads to φ. The group's ability to perform some action can then also be seen as to originate from the single agents' abilities to perform actions individually.

In Section 2.3, we will present several approaches that follow this idea. At this point, we will now take a very short look at the logic of cooperation and propositional control (CL-PC) developed by van der Hoek and Wooldridge [van der Hoek and Wooldridge, 2005]. CL-PC is an attempt to make the coalitional power explicit by relating the coalitional power of a group to the ability of agents to determine the truth value of propositional variables true are contained in φ. In this logic, each agent has a set of propositional variables under his control. Each propositional variable is controlled by only one agent. An agent having a set of propositional variables under his control means that he alone can determine the truth value of those variables. The idea is basically the same as in what is known under the name *Boolean games* [Harrenstein et al., 2001; Bonzon et al., 2006].

2.3 Cooperative Ability and Actions

The cooperation logics that we presented so far share the feature that they capture the cooperative ability of agents to achieve some state of affairs only implicitly: There is no explicit representation of *how* a group of agents can achieve some state of affairs.

Of course, there are many situations in multi-agent systems where only the results are relevant that coalitions can achieve and not the way how exactly the coalitions can achieve them. In such cases the high level analysis as provided by the cooperation logics we presented in the last section is perfectly appropriate. On the other hand, if we look at the scenario from a planning-point of view, it is of crucial interest how exactly a coalition can bring about some state of affairs. In AI there has been done quite lot of work in order to investigate *how* some goal can be achieved or some problem can be solved in a multi-agent system [van der Hoek et al., 2005; Wooldridge and Jennings, 1994].

In fact, one of the first attempts to develop a logic for reasoning about the ability of agents to achieve some state of affairs or goal, which was done by Moore in [Moore, 1980], already tries to spell out the connection between the cooperative ability to achieve some result and the ability to perform actions. Moore investigates how knowledge effects the ability of agents to achieve a goal. The underlying idea of Moore's approach is that in order to achieve something the agent must know some plan for achieving this goal and furthermore he must be able to carry out the plan. Analogously, this holds for groups of agents. Later on, van der Hoek and Wooldridge followed the same ideas but concentrate only on epistemic aspects and not on actions [van der Hoek and Wooldridge, 2003].

Despite Moore's work in 1980, it has only been recently that formal investigations of the connection between the cooperative ability of groups of agents to achieve some state of affairs and their ability to perform certain actions and to execute certain plans have received more attention [Borgo, 2007; Sauro et al., 2006; Gerbrandy and Sauro, 2007].

These recently developed frameworks investigate the relationship between cooperative ability and actions on different levels. In [Sauro et al., 2006], a cooperation logic with actions is defined that relates the ability of groups of agents to achieve that some formula φ holds to the actions the group can perform and the effects the performance of these actions has. We will present this approach in detail in the next chapter and therefore only give a very brief summary of it at this point.

In [Sauro et al., 2006], Sauro, Gerbrandy, van der Hoek and Wooldridge investigate the cooperative ability and actions of groups of agents in a multi-agent system that is based on an environment represented as a state transition system, where the performance of actions results in transitions between the states. For reasoning about the effects of the concurrent performance of actions in the transition system, a Boolean modal logic is used that is similar to a fragment of PDL [Harel, 1984]. Then such an environment is populated by agents that each have the ability to perform certain actions and can also cooperate as groups. Additionally, there is a separate module for reasoning about the agents' cooperative ability to perform actions. Here, a CL-like cooperation logic is used that expresses which complex actions a coalition can force (as opposed to 'state of affairs' as in CL). Then the environment module and the agents module can be combined and then groups of agents can achieve certain states of affairs by forcing actions that are are guaranteed to result in such states of affairs. The basic relationship that is established here is the following.

> A group C can force φ if and only if there is some concurrent action that the group C can perform that has the effect of making φ true no matter what the other agents do that are not in C.

'Group C being able to perform some concurrent action that has the effect of making φ true' means the following: There is a set of actions that can each be performed by some member of C and the concurrent performance of this set has the effect of leading to a state that satisfies φ no matter which other actions are additionally performed (by agents not in C). So, the cooperative ability of a group C to enforce φ is made explicit in the following way.

Coalition C, by doing such and such, can enforce φ.

Whereas this approach does provide the group with an explicit description of what to do as a group, i.e. what actions to perform in order to achieve φ, it does not explicitly say what each member of the group should do individually. In some cases, the actions performed individually might be obvious from the way how the ability to perform certain actions is distributed among the members of the group; in other cases it might not be clear at all. This issue is taken care of in the Coalition Action Logic (CAL) developed by Borgo [Borgo, 2007] .

2.3.1 Coalition Action Logic (CAL)

Borgo's motivation for developing Coalition Action Logic (CAL) [Borgo, 2007] was to investigate the precise relationship between cooperation logics as CL and ATL that investigate coalitional power from such a high level perspective and cooperation logics that do represent coalitional power in a more explicit way by also saying how a coalition can achieve some result. In particular, the aim is to develop a logic that provides us with an explicit description – i.e. by explicitly saying how a group of agents can achieve some result – of exactly the kind of coalitional power as it is modelled in CL. Note that the idea differs from the one underlying CLA. In CLA, it is rather the case that it is specified how agents can perform actions and which effects the actions have. Then via this specifications, we can obtain the corresponding representation of the power of a group to enforce the truth of some formula. CAL [Borgo, 2007] is supposed to spell out exactly the kind of coalitional power that is used in CL. This is done in terms of actions for every individual player in the coalition instead of putting it in terms of group actions that a group can perform collectively.

Then it is shown that the cooperative ability to force the truth of some formula φ in the sense of CL can be spelled out using a multi-agent modal logic, which is is a dynamic logic where the modalities contain action expressions for each agent. These action expressions can be existentially or universally quantified. More precisely, the logic that is used is a fragment of the multi-agent quantificational modal logic presented in [Borgo, 2005].

Assume that we have a set of agents $Ag = \{1, \dots, n\}$. Then formulas of the action logic as it is used in [Borgo, 2007], can be of the form

$$\begin{bmatrix} Qx_1 \\ Qx_2 \\ \vdots \\ \\ Qx_n \end{bmatrix} \varphi$$

where $Q \in \{\forall, \exists\}$ ranges over actions. If a group G consists of agents $\{1, \dots, k\}$, where $k \leq n$, then instead of saying that G can force φ if and only if G can perform some action that leads to φ, like it is done in the cooperation logic with actions CLA [Sauro et al., 2006], in Borgo's framework this is expressed in the following way.

> G can force φ if and only if there are actions $A_1, \dots A_k$ such that if agents $1, \dots, k$ each perform the respective actions, then the resulting action will always lead to φ being true no matter what the agents $k+1, \dots, n$ do.

Whereas at first glance it seems that the coalitional power as represented in CL can be made explicit in exactly this way, this is however not the case. The correspondence cannot be established in this straightforward way but it is necessary to introduce an additional player. This is due to the fact that in CL, the empty coalition can bring about nontrivial results whereas this is not the case in CAL. So, in order to solve this problem, a player *nature* is added. Then, the power of a coalition C to achieve some φ can be spelled out as follows: $C = \{1, \ldots, k\}$ can achieve φ if and only if there are actions $A_0, A_1, \ldots, A_k$ such that agents $0, 1, \ldots, k$ performing those actions leads to a state where φ is true irrespectively of what the other agents do. Here, player 0 is the newly introduced player who corresponds to the empty coalition in CL. Then Borgo shows that after adding this additional player, a translation between CL and CAL can be defined [Borgo, 2007].

Whereas this approach does indeed provide us with translations that allow us to switch from CL to a corresponding logic that represents the actions of agents in an explicit way, unlike CLA [Sauro et al., 2006] it does not contain any explicit representation of the set of actions that each player is able to perform. Therefore, Borgos' framework allows us to formally investigate the connection between the performance of actions by single players and the cooperative ability of groups but not the connection between the single agents' abilities to perform actions and the cooperative ability of groups.

Both CLA [Sauro et al., 2006] and CAL [Borgo, 2007] only provide explicit representations of actions that take one step and not whole sequences of actions.

Considering the issue of planning in multi-agent systems, it is not only of interest what agents can achieve in one step but of course plans of how to achieve some goals in multiple steps are also relevant. In many cases it is more likely that achieving some goal cannot be done by just performing a single action once but rather a whole sequence of actions might be needed.

2.3.2 Plans and Cooperation

In [Gerbrandy and Sauro, 2007], Gerbrandy and Sauro basically extend the approach by Sauro, Gerbrandy, van der Hoek and Wooldridge [Sauro et al., 2006] by considering the effects of agents executing complex plans that involve sequences of actions instead of only executing single actions or sets of actions concurrently.

Like the case of CLA [Sauro et al., 2006], a modular approach is chosen. The model used for the environment is basically the same as in CLA. The environment is based on a set labelled transition system, where the transitions are labelled by sets of atomic actions. Then a propositional language is used to talk about those sets of actions and to define more complex action expressions by Boolean combinations of atomic actions. Then a planning language is build from those action expressions. Plans include the performance of complex actions, the concatenation of two plans and also expressions saying that a plan is to be executed under a certain conditions, e.g. if some formula is true. Not only the situation at the current state is relevant, but plans involve future states as well. Statements of the following form can be expressed.

$$\texttt{while } \varphi \texttt{ do } \mathbf{A}$$

where doing **A** is the execution of some plan that can involve more complex action and both quantification over paths, i.e expression of the form *in all possible evolutions of the system, the formula φ will hold*, and also temporal modalities allowing us to express *until* and *next*.

So, the environment module models the effects of the execution of complex plans. As the approach is kept modular, there is no representation of agents in the environment module. Then an agents module is defined for reasoning about the agents' capability to enforce complex plans. It is important to note at this point that this capability is independent from the states of the environment. The logic for reasoning about agents' capabilities to force plans is based on CL, the main difference being that the agents can force plans instead of states of affairs.

Moreover, it is important to to distinguish *capability* from *executability*. A group might be capable of executing some plan in general but it might be the case that in the current situation the plan is not executable because some action that is involved in the plan cannot be performed in the current state[1]. Then the agents module and the environment module are combined and a multi-agent system is obtained, in which we can also reason about which states of affairs groups of agents can achieve. In the agents module it is specified which plans they can force and the environment module then gives us the effects the execution of the plans has. The general idea in establishing that a group of agents has the ability to force φ is the following:

> Group G can force φ if there is a plan that the group can force and whose execution leads to φ.

Technically, the framework that is developed in [Gerbrandy and Sauro, 2007] can be seen as an extension of ATL that is more general than other extensions that provide ATL with action languages [Sauro et al., 2006; van der Hoek et al., 2005]. It is more general because it considers strategies that involve more than one step and the investigation is not restricted to memory-less strategies.

This overview has shown some of the different perspectives that have been taken so far when formally investigating coalitional power in game-like situations in multi-agent systems. We have seen that coalitional ability can be seen as the ability to achieve some state of affairs, the ability to achieve that some complex action is being performed or that some plan is executed. Next, we will investigate how frameworks for reasoning about agents' preferences, cooperative ability and actions and their effects can be combined in order to obtain one framework in which the different aspects of cooperation can not only be investigated separately but also their interdependence.

[1] Here, the labelled transition systems that are used in the environment differ indeed from the ones in [Sauro et al., 2006] because there it is required that every set of actions can be performed in every state.

Chapter 3

Cooperative Ability and Actions

Our aim is to develop a logic for reasoning about strategic cooperative abilities that explicitly represents how a coalition can achieve a certain state of affairs and that also takes into account the preferences the agents have over the outcomes.

We will try to build such a logic by extending an existing logic for reasoning about actions and their effects and about cooperative ability of (groups) of agents [Sauro et al., 2006]. In the next chapter, we will then add a representation of the agents' preferences to this logic.

After having given an overview of the field and the existing literature dealing with formal reasoning about cooperation, actions and preferences in multi-agent systems, we will now focus on combining cooperative ability and actions. We will present one approach to combining formal reasoning about actions with formal reasoning about cooperative ability of agents. More precisely, the approach that we present in this chapter, which has been developed in [Sauro et al., 2006], formalizes actions and cooperation by combining a model for reasoning about actions and their effects with a model for reasoning about agents' abilities to perform actions, individually and as groups. Combining these two models then allows for an explicit representations of how agents can achieve certain states of affairs, namely via the actions they can perform and the effects the performance has in the environment.

Technically, the logic that we will describe in the following section combines an action logic in the style of PDL with an cooperation logic which in a sense resembles coalition logic but describes the cooperative ability to perform actions instead of directly describing the cooperative ability to achieve certain states of affairs. After describing the cooperation logic with actions in Section 3.1, in Section 3.3 we will then investigate the effects that a group of agents performing certain sets of actions has on the uncertainty of the group concerning the result of the actions.

3.1 Cooperation Logic with Actions

The cooperation logic with actions [Sauro et al., 2006] that we will describe in what follows provides a formal framework for investigating the questions

> ***Who** can achieve some state of affairs? And **how** can they obtain it?* [Sauro et al., 2006]

in multi-agent systems.

The logic is based on two modules, an agent module for reasoning about the ability of (groups of) agents to perform certain actions and an environment module for reasoning about the effects of actions.

The modules are independent systems in the sense that the agent module describes which actions each agent and each group of agents can perform without taking into account the effects that the performance of each of the actions has. Moreover, the effects of actions as they are modelled in the environment module are independent of the agents and their ability to perform them. This means that in the environment module it is specified what happens if certain (sets) of actions are performed irrespectively of who exactly performs them.

The idea underlying the combination of both modules is that a coalition can achieve a certain state of affairs φ by performing some action together that will lead to φ no matter what the other agents do.

In the remainder of this section, we will first present the two modules of the cooperation logic with actions [Sauro et al., 2006] and then show how they are combined.

3.1.1 Environment Module for Reasoning about the Effects of Actions

An environment in which agents perform actions can be represented in several ways. The way that is chosen in [Sauro et al., 2006] is based on the idea that such an environment can be modelled by a transition system [Harel, 1984], or more precisely: a set-labelled transition system. This means that we have a set of states and a set of actions and a way of describing how the performance of actions changes the current state of the environment. Since the environment module will later be used for modelling the environment in which groups of agents are interacting by performing certain actions, we are not only interested in the effects of the performance of single actions but also in the effects of sets of actions. Moreover, we have a set of propositional letters and a propositional valuation function for describing the properties of the states of the system.

Definition 3.1 (Environment Module [Sauro et al., 2006])**.** *The environment is modelled as a set-labelled transition system (SLTS) which is a tuple*

$$\langle S, Ac, (\rightarrow)_{A \subseteq Ac}, \Phi_0, \pi \rangle,$$

where S is a set of states, Ac is a finite set of atomic action expressions, $\rightarrow_A$ is a binary relation over S for each $A \subseteq Ac$, Φ_0 is a set of propositional variables and $\pi : (S \times \Phi_0) \rightarrow \{0, 1\}$ is an interpretation function that assigns each pair consisting of a state and a propositional variable a truth value.
Furthermore, the relation $\rightarrow_A$ is required to be serial for each $A \subseteq Ac$, i.e. for every state s and every set of actions $A \subseteq Ac$ there is some state t such that $s \rightarrow_A t$.

The transition systems can be nondeterministic, i.e. if in the current state some set of action is performed there might be two or more states that the system can move into. So, we can have that $s \rightarrow_A t$ and $s \rightarrow_A t'$ for $t \neq t'$. We will first start working with this kind of transition systems and at a later point restrict our investigation to deterministic systems.
Now, we will introduce some formalism for describing sets of actions which allows us to talk about actions as more complex items than just atomic actions.

We introduce a propositional language that can be interpreted in sets of atomic actions.

Definition 3.2 (Action Language $\mathcal{L}_{ac}$[Sauro et al., 2006])**.** *Given a set Ac of atomic actions, more complex action expressions are defined by the following grammar:*

$$\alpha ::= \ a \mid \alpha \wedge \alpha \mid \neg\alpha,$$

where $a \in Ac$.

We call the action language generated by this grammar $\mathcal{L}_{ac}$.

Then action expressions can be interpreted in a set $A \subseteq Ac$ in a straightforward way.

Definition 3.3 (Action Interpretation [Sauro et al., 2006])**.** *Formulas of $\mathcal{L}_{ac}$ can be interpreted in a set $A \subseteq Ac$ as follows.*

$$A \models^{ac} \alpha \textit{ iff } A \models^{p} \alpha$$

where $\models^p$ is the interpretation relation of classical propositional logic.

Then if $A \models^{ac} \alpha$, we say 'the concurrent action A is of type α' or 'α is true of A'.

After having a way for talking about actions we can now introduce a language that allows us to talk about the environment and that is able to express the effects of actions being performed in certain states.

Definition 3.4 (Environment Language $\mathcal{L}_e$ [Sauro et al., 2006])**.** *The environment language is generated by the following grammar.*

$$\varphi ::= \ p \mid \varphi \wedge \varphi \mid \neg\varphi \mid [\alpha]\varphi$$

where $p \in \Phi_0$ and α is an expression of the action language $\mathcal{L}_{ac}$.

The important feature of this language is that we can construct new modalities $[\alpha]$ by using the action logic. The language of the Environment Logic is basically that of Boolean Modal Logic [Gargov and Passy, 1990; Blackburn et al., 2001]. We will not go into the technical details of this logic here.

Now, we can interpret the previously defined language in set-labelled transition systems.

Definition 3.5 (Interpretation of $\mathcal{L}_e$ in $SLTS$s [Sauro et al., 2006])**.** *Given an SLTS $Env = \langle S, Ac, (\rightarrow)_{A\subseteq Ac}, \Phi_0, \pi\rangle$, propositional formulas of the environment language can be interpreted in Env in the standard way and for formulas of the form $[\alpha]\varphi$, we have*

$$\begin{array}{lll} Env, s \models^e [\alpha]\varphi & \textit{iff} & \forall A \in 2^{Ac}, s' \in S : \\ & & \quad \textit{if } A \models^{ac} \alpha \textit{ and } s \rightarrow_A s' \textit{ then } Env, s' \models^e \varphi. \end{array}$$

Let us look at the expression $[\alpha]\varphi$ in more detail. If e.g. we have that $Env, s \models^e [a]\varphi$ for an atomic action a, then this means that in state s whenever a set of actions A is performed where $a \in A$ then the next state will satisfy φ. So, for any set of actions A such that $a \in A$ we have that for all t such that $Env, s \rightarrow_A t$ it holds that $Env, t \models^e \varphi$.

If we have that $Env, s \models^e [\neg a]\varphi$, then this means that whenever a is not part of the actions that are performed, then this leads to a state where φ is true. In this case, a state satisfying $\neg\varphi$ can only be reached if a set of actions containing a is performed.

For action expressions α in general, $[\alpha]\varphi$ means that being of type α is a sufficient condition for a concurrent action to result in a state satisfying φ. On the other hand, $[\neg\alpha]\neg\varphi$ means that any concurrent action that is not of type α will lead to a state not satisfying φ. In this case, being of type α is a necessary condition for a concurrent action to lead to a φ-state.

Definition 3.6 (Environment Logic Λ^E [Sauro et al., 2006]). *Consider the following set of axioms. Define the environment logic Λ^E to be the set of formulas of $\mathcal{L}_e$ that can be derived from the following set of axioms and that is closed under the rules of inference modus ponens and necessitation of each modality $[\alpha]$.*

1. *All tautologies of classical propositional logic*
2. $[\alpha](\varphi \rightarrow \psi) \rightarrow ([\alpha]\varphi \rightarrow [\alpha]\psi)$ *for each* α
3. *All axioms of the form* $[\alpha]\varphi \rightarrow [\beta]\varphi$, *if* $\vdash \beta \rightarrow \alpha$ *in propositional logic.*
4. $([\alpha]\varphi \wedge [\beta]\varphi) \rightarrow [\alpha \vee \beta]\varphi$
5. $\neg[\alpha]\neg\top$ *if* α *is consistent in propositional logic*
6. $[\bot]\bot$

It follows from Axiom 2 and necessitation for each $[\alpha]$ that each $[\alpha]$ is a normal modal operator. Axiom 3 says that if β implies α in propositional logic, then if every action of type α results in a φ-state then so does every action of type β (because every action of type β is also of type α). The forth axiom says that if every action of type α results in a φ-state and also every action of type β, then the same holds for every action of type $\alpha \vee \beta$. Axiom 5 is a seriality axiom. Together with the last axiom, it corresponds to the property that in every state every consistent set of actions can be performed.

Proposition 3.7 (Soundness and Completeness of Environment Logic [Sauro et al., 2006]). *The environment logic is sound and complete with respect to the class of environment modules.*

We will not present the proof here. The reader can find it in [Sauro et al., 2006].

The environment logic as defined above is a dynamic logic that allows us to reason about the effects of the concurrent performance of actions. As we have seen, the logic also provides a way of reasoning about the effects of *not* performing certain actions. Now, we will define an agent module that will describe the cooperative ability of agents to perform actions. Later, we will combine both modules and formalize how agents can act in an environment.

3.1.2 Agents Module for Reasoning about the Ability of (Groups of) Agents to Perform certain Actions

Now, we will give an overview of the agents module as it has been defined in [Sauro et al., 2006]. It provides a way of reasoning about the cooperative ability of agents. As opposed to coalition logic [Pauly, 2002], the cooperative ability in this framework is the ability of (groups of) agents to perform certain actions and not directly the ability to achieve certain states of affairs. However, this will be the case at a later stage after we have combined the agents module and the environment logic. This will enable us to talk about the cooperative ability of agents to achieve certain states of affairs via an intermediate step using the agents' abilities to perform actions and the specification of the effects of actions as it given in the environment module.
The essential part of the agents module is that for each agent we have a set of actions that he can perform[1].

Definition 3.8 (Agents Module [Sauro et al., 2006])**.** *Formally, the agents module is defined as* $\langle Ag, Ac, \mathsf{act}\rangle$*, where* Ag *is a set of agents,* Ac *is a set of atomic actions and* act *is a function* $\mathsf{act} : Ag \to 2^{Ac}$*, mapping each agent to the set of actions that he can perform.*

It is required that $\bigcup_{i \in Ag} \mathsf{act}(i) = Ac$*, i.e.* Ac *contains exactly those actions that can be performed by some agent.*

Note that apart from the requirement that every action can be performed by some agent, there is no assumption about how the abilities to perform actions are distributed between the agents. So, there might be actions that can be performed by several or even all agents and also actions that can only be performed by one agent. In subsequent chapters, we will take a closer look at how the interaction of the agents is influenced by the way how the abilities to perform actions are distributed among the agents.

Now we will show that we can express that a coalition has the power to enforce that an action of a certain type will be performed[2].

First, we will say how to specify the set of actions that a group can perform. Let G be a group. Then, $\mathsf{act}(G) := \bigcup_{i \in G} \mathsf{act}(i)$ represents the set of actions a group of agents G can perform.

Then we define a language for talking about the abilities of agents to perform actions.

Definition 3.9 ([Sauro et al., 2006])**.** *The agent language* $\mathcal{L}_a$ *consists of expressions of the form*

$$\langle\langle G\rangle\rangle\alpha,$$

where $G \subseteq Ag$ *is a group of agents and* α *is an action expression (Definition 3.2).*

[1] This is similar to the logic of cooperation and propositional control (CL-PC) [van der Hoek and Wooldridge, 2005], where each agent has a set of propositional variables that are under his control and also to Boolean Games [Harrenstein et al., 2001; Bonzon et al., 2006]

[2] This is similar to the cooperative ability in CL but as opposed to CL, here it is (complex) actions that can be forced, not state of affairs.

The intended meaning is that the group G is able to enforce that a set of actions of type α will be performed. Then we can say that 'G is effective for α', that 'G can enforce α' or that 'G is able to do α'.
Formally, the language is interpreted in an agent module as follows.

Definition 3.10 (Ability for Actions [Sauro et al., 2006]). *Let $\langle Ag, Ac, \mathsf{act}\rangle$ be an agents module and let $G \subseteq Ag$. Then we can interpret formulas of $\mathcal{L}_a$ in the module as follows.*

$$\begin{array}{lll}\langle Ag, Ac, \mathsf{act}\rangle \models^a \langle\langle G\rangle\rangle\alpha & \textit{iff} & \textit{there is a set of actions } A \subseteq \mathsf{act}(G) \textit{ such that} \\ & & \textit{for all sets of actions } B \subseteq \mathsf{act}(Ag \setminus G) \textit{ it holds} \\ & & \textit{that } A \cup B \models^{ac} \alpha.\end{array}$$

So, $\langle\langle G\rangle\rangle\alpha$ means that there is a set of actions that G can perform such that no matter what the other agents do, the resulting action will always be of type α. It is important to keep in mind that the cooperative ability of groups is in our case the cooperative ability to perform complex actions and not the ability to achieve some states of affairs.

Definition 3.11 (Coalition Logic for Actions [Sauro et al., 2006]). *We define the coalition logic for actions Λ^A to be the set of formulas derived from the following set of axioms and closed under modus ponens.*

1. $\langle\langle G\rangle\rangle\top$*, for all* $G \subseteq 2^{Ag}$
2. $\langle\langle G\rangle\rangle\alpha \to \neg\langle\langle Ag \setminus G\rangle\rangle\neg\alpha$
3. $\langle\langle G\rangle\rangle\alpha \to \langle\langle G\rangle\rangle\beta$ *if* $\vdash \alpha \to \beta$ *in propositional logic*
4. $\langle\langle G\rangle\rangle a \to \bigvee_{i\in G}\langle\langle\{i\}\rangle\rangle a$ *for all* $G \in 2^{Ag}$ *and atomic* $a \in Ac$
5. $(\langle\langle G_1\rangle\rangle\alpha \wedge \langle\langle G_2\rangle\rangle\beta) \to \langle\langle G_1 \cup G_2\rangle\rangle(\alpha \wedge \beta)$*, for* $G_1 \cap G_2 = \emptyset$
6. $(\langle\langle G\rangle\rangle\alpha \wedge \langle\langle G\rangle\rangle\beta) \to \langle\langle G\rangle\rangle(\alpha \wedge \beta)$ *if* α *and* β *have no common atomic actions*
7. $\langle\langle G\rangle\rangle\neg a \to \langle\langle G\rangle\rangle a$ *for atomic* $a \in Ac$
8. $\langle\langle G\rangle\rangle\alpha \to \bigvee\{\langle\langle G\rangle\rangle \bigwedge \Psi | \Psi$ *is a set of literals such that* $\bigwedge \Psi \to \alpha\}$

The following list gives some explanations for each of the axioms.

- The first axiom says that every group can enforce some trivial action.
- Axiom 2 relates the ability of a group G to the ability of the group $Ag \setminus G$. The axiom says that if G can force the next concurrent action to be of type α then it cannot be the case that $Ag \setminus G$ can force it to be of type $\neg\alpha$.
- Axiom 3 says that if a group can force an α-action and α implies β in propositional logic, then the group can also force a β-action since they can force an α-action and every α-action is also of type β.

  Together with Axiom 1, Axiom 3 implies that $\langle\langle G\rangle\rangle\alpha$ for any tautology of classical propositional logic.

- Axiom 4 says that if a group has the ability to enforce the concurrent action to be of type a where a is an atomic action then there must be a member of the group G that can perform action a. This is quite clear because a set of actions being of type a means that a is actually a member of this set. And since the actions that G can perform is just the union of all the actions its members can perform G being able to perform a then implies that there is a member who can perform a.

- Axiom 5 states that two disjoint groups can cooperate. If two disjoint groups can force an action to be of type α and type β respectively, then together they can force an action to be of type $\alpha \wedge \beta$. This is very similar to the axiom of superadditivity, which is a central axiom in coalition logic [Pauly, 2002]. As in coalition logic, also in our case the condition that the groups are disjoint is necessary. If there is an agent $i \in G_1 \cap G_2$ then it might be the case that G_1 can only force α if i does action a and that G_2 can only force β if i does *not* perform a. Then if $G_1 \cup G_2$ wants to force $\alpha \wedge \beta$, i might have to do a and to not do a at the same time.

- Axiom 6 says that if a group can enforce both α and β where α and β involve completely different atomic actions then the group can also enforce $\alpha \wedge \beta$. This can be done by performing the actions that guarantee α and at the same time also perform the ones that guarantee β. The condition that α and β contain completely different atomic actions ensures that $\alpha \wedge \beta$ is consistent.

- Remember that we require that every atomic action in the set Ac can be performed by at least one agent. As a consequence, we have Axiom 7. If $\langle\langle G \rangle\rangle \neg a$ then it cannot be the case that there is some agent in $Ag \setminus G$ who can perform a. This directly implies that there is an agent in G who can perform a and therefore G can force an action to be of type a, i.e. $\langle\langle G \rangle\rangle a$.

- Axiom 8 says that if $\langle\langle G \rangle\rangle \alpha$ then there must be a set containing only atomic formulas and negations of atomic formulas such that the conjunction of all of them together implies α. This means that if a group can enforce α then there must be an explicit description for them that says how to do it (by explicitly saying which atomic actions they should perform and which ones they should not perform).

Proposition 3.12 (Soundness and Completeness of the Coalition Logic for Actions[Sauro et al., 2006]). *The coalition logic for actions Λ^A is sound and complete with respect to the class of agents modules.*

So, we have an agents logic that we can use for reasoning about the abilities of single agents and groups of agents to perform complex actions. Now, we will combine this logic with the environment logic defined in Section 3.1.1 in order to obtain a logic for reasoning about agents that can achieve certain states of affairs by performing actions in an environment.

3.1.3 Combining the Action Module and the Agents Module – Multi-agent System

So far, we have a module that describes the environment and another module describing the abilities of individual agents and groups of agents to perform actions. In this section, we will combine both modules. This can be done by identifying the sets of actions of the modules. This combination provides us with a semantics for reasoning about agents that perform actions in an environment. Then we will also combine the coalition logic for actions with the environment logic and build a logic for reasoning about multi-agent systems.

Definition 3.13 (Multi-agent System [Sauro et al., 2006]). *Formally, a multi-agent system M is a tuple*

$$M = \langle S, Ac, (\rightarrow)_{A \subseteq Ac}, \Phi_0, \pi, Ag, \mathsf{act}\rangle,$$

where $\langle S, Ac, (\rightarrow)_{A \subseteq Ac}, \Phi_0, \pi, \rangle$ is an environment module and $\langle Ac, Ag, act\rangle$ is an agents module.

In such multi-agent system, everything that we could express in the agents module and in the environment module is still of interest. Therefore, a logic for reasoning about multi-agent systems as we just defined them should still be able to express what the environment logic and the agent logic can express. Additionally in a multi-agent system, it is also of interest whether groups of agent can achieve certain states of affairs. This question did not make much sense in the separate environment and agents modules.

We will add the expression $\langle\langle G\rangle\rangle\varphi$ to our language, where φ is not an action expression but a sentence in the language for talking about multi-agent systems that we will now define. The intended meaning of $\langle\langle G\rangle\rangle\varphi$ is that the group G has the power to achieve a state that satisfies φ.

Definition 3.14 ([Sauro et al., 2006]). *The language for multi-agent systems is generated by the following grammar:*

$$\varphi ::= \quad p \mid \varphi \wedge \varphi \mid \neg\varphi \mid [\alpha]\varphi \mid \langle\langle G\rangle\rangle\alpha \mid \langle\langle G\rangle\rangle\varphi$$

The formulas can be interpreted in a multi-agent system $M = \langle S, Ac, (\rightarrow)_{A \subseteq Ac}, \Phi_0, \pi, Ag, \mathsf{act}\rangle$.

Definition 3.15 ([Sauro et al., 2006]). *The formulas are interpreted in a multi-agent system $M = \langle S, Ac, (\rightarrow)_{A \subseteq Ac}, \Phi_0, \pi, Ag, \mathsf{act}\rangle$ as follows:*

$$\begin{array}{lll}
M, s \models^m p & \textit{iff} & \langle S, Ac, (\rightarrow)_{A \subseteq Ac}, \Phi_0, \pi\rangle, s \models^e p \\
M, s \models^m [\alpha]\varphi & \textit{iff} & \forall A \subseteq Ac, t \in S: \textit{ if } A \models^{ac} \alpha \textit{ and } s \rightarrow_A t \textit{ then } M, t \models^m \varphi \\
M, s \models^m \langle\langle G\rangle\rangle\alpha & \textit{iff} & \langle Ac, Ag, act\rangle \models^a \langle\langle G\rangle\rangle\alpha \\
M, s \models^m \langle\langle G\rangle\rangle\varphi & \textit{iff} & \exists A \subseteq \mathsf{act}(G) \textit{ such that } \forall B \subseteq \mathsf{act}(Ag \setminus G), t \in S: \\
 & & \quad \textit{if } s \rightarrow_{A \cup B} t, \textit{ then } M, s \models^m \varphi.
\end{array}$$

The interpretation of Boolean combinations of formulas is done in the standard way.

So, a group G having the ability to enforce a state of affairs φ (i.e. $\langle\langle G\rangle\rangle\varphi$) means that G can perform a concurrent action such that the resulting state will satisfy φ irrespectively of what the other agents do.

Notation 3.16. *Note that in what follows at some points we will omit the indexing of the satisfiability relations of the different logics and just write "$\models$" if it is clear from the context which satisfaction relation we mean.*

Now, we will present a proposition that gives the precise relationship between a group's ability to achieve some state of affairs φ and the group's ability to force the next concurrent action to be of a certain type that is guaranteed to lead to that state satisfying φ.

The idea is that a coalition can achieve φ if and only if it can force the concurrent action to be of a certain type which guarantees that the next state will satisfy φ. The proof is straightforward and can be found in [Sauro et al., 2006] but we will present it in more detail since it uses some ideas that will also be used in subsequent chapters.

The key step of the proof is the following:
Given that we know that a coalition has the ability to achieve that the system is moving to some state satisfying φ, we have to find out how exactly the group can enforce such a state of affairs, i.e. we have to find an action expression that the group can force that has the effect of resulting in a state that satisfies φ.

Before presenting a lemma that shows how this step is done, we will introduce some notation that is quite handy at this point and will also be used in various places throughout the remainder of this thesis. It gives us a way of talking about all the possible concurrent actions that might take place when one group of agents is performing some set of actions and we do not know what the other agents are doing.

Notation 3.17. *Let G be a group of agents and let $A \subseteq \mathsf{act}(G)$ be a set of actions that G can perform. Then we define the set $\Phi(A, G)$ as follows.*

$$\Phi(A, G) := A \cup \{\neg a | a \in (\mathsf{act}(G) \setminus A), a \notin \mathsf{act}(Ag \setminus G)\} .$$

So, the set $\Phi(A, G)$ contains exactly the actions in A and the negations of other actions that G can perform but that none of the other agents can perform. The following lemma says that if we know that a concurrent action is of type $\bigwedge \Phi(A, G)$, then this action has to contain A as a subset and besides that only contains actions that agents can perform that are not members of G. The result basically follows from the definition of $\Phi(A, G)$ but we will state the details of the proof since we will use the lemma in the proofs of several other results.

Lemma 3.18. *Let $\langle S, Ac, (\rightarrow)_{A\subseteq Ac}, \Phi_0, \pi, Ag, \mathsf{act}\rangle$ be a multi-agent system and let $G \subseteq Ag$ and $A \subseteq \mathsf{act}(G)$. Then for any $C \subseteq Ac$: if $C \models \bigwedge \Phi(A, G)$, then $C = A \cup B$, where $B \subseteq \mathsf{act}(Ag \setminus G)$.*

Proof. Since $C \models \bigwedge \Phi(A, G)$, it follows that $C \models \bigwedge A$ and therefore $A \subseteq C$.
If $A = C$, we are done.
So, let $b \in C \setminus A$. Suppose that $b \in \mathsf{act}(G)$. Then $b \in \mathsf{act}(G) \setminus A$, which implies that $\neg b \in \Phi(A, G)$ and therefore $C \models \neg b$. So, $b \notin C$, which is a contradiction.
Thus, $b \notin \mathsf{act}(G)$. Since $\bigcup_{i\in Ag} \mathsf{act}(i) = Ac, b \in \mathsf{act}(Ag \setminus G)$. Hence, $C = A \cup B$, where $B \subseteq \mathsf{act}(Ag \setminus G)$. □

So, if we know that an action of type $\bigwedge \Phi(A, G)$ has taken place, we know that A must have been performed and none of the atomic actions that are not part of A and that only agents in G can perform[3]. Conversely, if group G decides to do A and nothing else, then the resulting action will in any case be of type $\bigwedge \Phi(A, G)$.

Proposition 3.19 ([Sauro et al., 2006]). *Given a multi-agent system M and a state s of its environment, $M, s \models^m \langle\langle G\rangle\rangle\varphi$ iff there exists an action expression α such that $M, s \models^m \langle\langle G\rangle\rangle\alpha$ and $M, s \models^m [\alpha]\varphi$.*

Proof. ($\Rightarrow$) Assume that $M, s \models \langle\langle G\rangle\rangle\varphi$. This means that $\exists A \subseteq \mathsf{act}(G)$ such that for all $B \subseteq \mathsf{act}(Ag \setminus G)$ and for all states $t \in S$, if $s \rightarrow_{A\cup B} t$ then $M, t \models \varphi$.
Take such an A. Now, we show that $\bigwedge \Phi(A, G)$ is the action expression that we are looking for.

Since $A \models \bigwedge \Phi(A, G)$ and $A \subseteq \mathsf{act}(G)$, $M, s \models \langle\langle G\rangle\rangle \bigwedge \Phi(A, G)$. In order to show that $M, s \models \langle\langle G\rangle\rangle \bigwedge \Phi(A, G)$, let $s \rightarrow_C t$ for some $C \subseteq Ac$ such that $C \models \Phi(A, G)$.

Then by the previous lemma, $C = A \cup B$ for some $B \subseteq \mathsf{act}(Ag \setminus G)$.

Thus, $M, s \models \langle\langle G\rangle\rangle \bigwedge \Phi(A, G)$ and also $M, s \models [\bigwedge \Phi(A, G)]\varphi$.

($\Leftarrow$) Let α be an action expression such that $M, s \models \langle\langle G\rangle\rangle\alpha$ and $M, s \models [\alpha]\varphi$. By definition of $\langle\langle G\rangle\rangle\alpha$, there must be an $A \subseteq \mathsf{act}(G)$ such that for all $B \subseteq \mathsf{act}(Ag \setminus G)$ and for all $t \in S$ if $s \rightarrow_{A\cup B} t$ then $A \cup B \models \alpha$. Since also $M, s \models [\alpha]\varphi$, $M, t \models \varphi$. Thus, $M, s \models \langle\langle G\rangle\rangle\varphi$. □

This proposition is important in the sense that it shows how the combination of environment module and agent module allows us to express the cooperative ability of agents in an explicit way by establishing a direct relationship to how some state of affairs can be achieved.

After combining the agents module and the environment module, we will now also combine the environment logic and the coalition logic for actions in order to obtain a logic for reasoning about cooperation and actions in multi-agent systems. The axiomatic system we will present now contains all the axioms of the previously defined logics, i.e. Λ^E and Λ^A and additionally two axioms that relate the ability to perform actions to the ability to enforce the truth of a certain proposition.

Definition 3.20 (Cooperation Logic with Actions [Sauro et al., 2006]). *The cooperation logic with actions Λ^{CLA} is given by:*

1. *All axioms and rules of the environment logic Λ^E and the coalition logic for actions Λ^A,*

2. $(\langle\langle G\rangle\rangle\alpha \wedge [\alpha]\varphi) \rightarrow \langle\langle G\rangle\rangle\varphi$,

3. $\langle\langle G\rangle\rangle\varphi \rightarrow \bigvee\{\langle\langle G\rangle\rangle\alpha \wedge [\alpha]\varphi | \alpha$ *is the conjunction of a set of atomic actions or their negations*$\}$.

[3] Note the difference of our interpretation to the one given in [Sauro et al., 2006]. There it is claimed that if an action is of type $\bigwedge \Phi(A, G)$, then G must have performed A, but in fact also some other group might have done it or several groups could have done it together.

Axioms 2 and 3 relate the ability to enforce the truth of a sentence in the following way:

Axiom 2 says that if a group can force the concurrent action to be of a certain type and if every action of this type results in φ then the group can also force φ. Axiom 3 says that if a group can force φ then there is an action type that they can force that is the conjunction of literals and every action of this type leads to φ. This axioms basically says that if G can force φ then there is an action type they can force that explicitly says which atomic actions to do and which ones not to do. This is important since it provides the group with an explicit way of how to force φ.

Proposition 3.21 (Soundness an Completeness of the Cooperation Logic with Actions [Sauro et al., 2006])**.** *The cooperation logic with actions Λ^{CLA} is sound and complete with respect to the class of multi-agent systems.*

The proof can be found in [Sauro et al., 2006]. Using Propositions 3.7 and 3.12 (the soundness and completeness of the environment logic and the coalition logic with actions) it is relatively straightforward and the only case that requires some work is the newly introduced formulas of the form $\langle\langle G\rangle\rangle\varphi$. Here, Proposition 3.19 and axioms 2 and 3 are used.
As we discussed in the previous chapter, the axiom of superadditivity plays a central role in Pauly's coalition logic since it says how groups of agents can join their forces. Now, we will show, that also in our framework superadditivity is valid. It follows directly from the definitions of the ability of groups to force certain states of affairs.

Fact 3.22. *Given two groups $G_1 \subseteq Ag$, $G_2 \subseteq Ag$ such that $G_1 \cap G_2 = \emptyset$, the following holds:*
If $M, s \vDash^m \langle\langle G_1\rangle\rangle\varphi$ and $M, s \vDash^m \langle\langle G_2\rangle\rangle\psi$ then $M, s \vDash^m \langle\langle G_1 \cup G_2\rangle\rangle(\varphi \wedge \psi)$.

Proof. From $M, s \vDash^m \langle\langle G_1\rangle\rangle\varphi$ and $M, s \vDash^m \langle\langle G_2\rangle\rangle\psi$, it follows by definition that

1. $\exists A_1 \subseteq \mathsf{act}(G_1)$ such that $\forall B_1 \subseteq \mathsf{act}(Ag \setminus G_1) \forall t \in S$ it holds that if $S \rightarrow_{A_1 \cup B_1} t$, then $M, t \vDash^m \varphi$.
2. $\exists A_2 \subseteq \mathsf{act}(G_2)$ such that $\forall B_2 \subseteq \mathsf{act}(Ag \setminus G_2) \forall t \in S$ it holds that if $S \rightarrow_{A_2 \cup B_2} t$, then $M, t \vDash^m \psi$.

Take such A_1 and A_2. Next, take some $B \subseteq \mathsf{act}(Ag \setminus (G_1 \cup G_2))$ and a t such that $s \rightarrow_{(A_1 \cup A_2) \cup B} t$. Then we have that $(A_2 \cup B) \subseteq \mathsf{act}(Ag \setminus G_1)$ because of the following:

1. $A_2 \subseteq \mathsf{act}(Ag \setminus G_1)$ since $A_2 \subseteq \mathsf{act}(G_2)$ and $G_2 \subseteq Ag \setminus G_1$.
2. $B \subseteq \mathsf{act}(Ag \setminus G_1)$ since $(Ag \setminus (G_1 \cup G_2)) \subseteq (Ag \setminus G_1)$ and $B \subseteq \mathsf{act}(Ag \setminus (G_1 \cup G_2))$.

Analogously, we have that $(A_1 \cup B) \subseteq \mathsf{act}(Ag \setminus G_2)$ because of the following:

1. $A_1 \subseteq \mathsf{act}(Ag \setminus G_2)$ since $A_1 \subseteq \mathsf{act}(G_1)$ and $G_1 \subseteq Ag \setminus G_2$.
2. $B \subseteq \mathsf{act}(Ag \setminus G_2)$ since $(Ag \setminus (G_1 \cup G_2)) \subseteq (Ag \setminus G_2)$ and $B \subseteq \mathsf{act}(Ag \setminus (G_1 \cup G_2))$.

Next, since $(A_1 \cup A_2) \cup B = A_1 \cup (A_2 \cup B) = A_2 \cup (A_1 \cup B)$ and we already know that for any t such that $s \rightarrow_{A_1 \cup (A_2 \cup B)} t$ it holds that $M, t \models^m \varphi$ and for any t such that $s \rightarrow_{A_2 \cup (A_1 \cup B)} t$ it holds that $M, t \models^m \psi$, we can conclude that for any t such that $s \rightarrow_{(A_1 \cup A_2) \cup B} t$ it holds that $M, t \models^m \varphi \wedge \psi$.

Hence, $M, s \models^m \langle\langle G_1 \cup G_2 \rangle\rangle (\varphi \wedge \psi)$. □

Note that the condition that the groups are disjoint is needed because otherwise the following case can arise: $i \in G_1 \cap G_2$ and i is the only agent such that $a \in \mathsf{act}(i)$ for some action a. Then assume that we have that $M, s \models [a]\varphi$ and $M, s \models [\neg a]\neg\varphi$. Moreover, it is the case that $\langle\langle G_1 \rangle\rangle a$, because i can perform a and also $\langle\langle G_2 \rangle\rangle \neg a$ because i can force the resulting concurrent action be of type $\neg a$ by just not doing a. So, we have that $M, s \models^m \langle\langle G_1 \rangle\rangle \varphi$ and $M, s \models^m \langle\langle G_2 \rangle\rangle \neg\varphi$. Then by Fact 3.22 without the disjointness-condition this would imply that $\langle\langle G_1 \cup G_2 \rangle\rangle \varphi \wedge \neg\varphi$ which is a contradiction.

Fact 3.22 can be derived in the axiomatic system by using Axiom 4 of the coalition logic for actions Λ^A and the two newly added axioms of Λ^{CLA}.

We will now give a brief summary of the approach presented in [Sauro et al., 2006] and then discuss some aspects of it in the next section. The central ideas are the following:

The environment in which the agents act is represented by a labelled transition system. For reasoning about such a system, a PDL-like logic is developed that allows reasoning about the effects the performance of complex actions has. This logic is shown to be sound and complete with respect to the environment modules. Agents are added to this framework as actors that each have a set of actions that they can perform and can choose at each stage which set of actions they want to perform. Then a logic is developed that can be used for reasoning about the agents' ability to perform complex actions and also the cooperative ability of groups of agents to do so. This logic is sound and complete with respect to the class of agents modules. The final step is the combination of the environment module in which the effects of actions are modelled and the model of the agents and their cooperative abilities to perform complex actions. As a result, a multi-agent system is obtained, which is a model of agents performing actions in an environment. In the multi-agent systems, it can be modelled how groups of agents can achieve the truth of certain formulas; namely by together performing concurrent actions that result in states satisfying the formula. On a syntactic level, also the two corresponding logics are combined; i.e. the logic for reasoning about the agents' cooperative abilities to perform actions and the logic for reasoning about the environment. This then results in a logic for reasoning about agents acting in an environment that can also express the cooperative ability of groups of agents to achieve certain states of affairs.
That this logic is sound and complete with respect to the class of multi-agent systems basically follows from the completeness results of the sublogics.

3.1.4 Discussion

In this section, we will discuss some of the aspects of the approach presented in this chapter so far.

Preconditions

In the semantics defined in [Sauro et al., 2006], every set of actions can be performed in every state of the model. One possible objections is to say that performing some actions seems to only make sense if certain preconditions are fulfilled. As an example consider the action *switching on the light.* Somebody turning on the light does not really seem to make sense if it is already turned on. So, *turning on the light* seems to have the precondition of the light being off.

In [Sauro et al., 2006], the issue of preconditions is not explicitly mentioned. Nevertheless, it can be accounted for by e.g. saying that the state of the system does not change when a set of actions is performed whose preconditions are not fulfilled in the current state. Then performing those action does not have any effect. So, the preconditions can be seen as being implicitly represented in the structure of the model, i.e. in the way the accessibility relation is defined.

Modularity

As we already mentioned, modularity is an important feature of the approach in [Sauro et al., 2006]. It is assumed that the effects of actions can be modelled independently from the behavior of the agents that perform the actions [Sauro et al., 2006]. This does not mean that we cannot use the cooperation logic with actions to model actions whose effects do indeed depend on who performs the actions:

It is always possible to define the actions in a way such that $\mathsf{act}(i) \cap \mathsf{act}(j) = \emptyset$. Then, since the sets of atomic actions that each agent can perform are pairwise disjoint, for each atomic action, there is only one agent that can perform it. In this way, it is possible to model actions that are dependent on who performs them.

The advantage of separating the actions from the agents who perform them is that it breaks the problem of planning in a multi-agent environment into two sub-problems: one dealing with the ability of agents to perform certain actions and the other one dealing with the effects of the actions. Moreover, from a logical point of view the modularity seems quite practical since the two sub-modules are easier to be examined in isolation and already existing logics can be used for this. Moreover, the completeness of the combined logics follows from the completeness of the logics for the submodules.

Another advantage is that the different modules are based on existing approaches (the environment module on PDL and the agents module on CL). Combining them gives us also a framework in which we can investigate to what extend those existing approaches are adequate models of the aspects they are designed for.

When reasoning on an intuitive level, it seems clear that there is an interdependence of the agents‘ ability to perform actions, their ability to achieve states of affairs, and the effects that actions have. Therefore, a framework that combines reasoning about the separate aspects both on a semantic and syntactic level then also allows investigations as to how far the separate approaches are adequate models since then they should be combinable.

Empty Coalition

Is it possible that the empty coalition can achieve nontrivial things?

We have that $\mathsf{act}(\emptyset) = \emptyset$, which means that the empty coalition cannot perform any actions.

For the cooperative ability of the empty coalition, it holds that

$$M, s \vDash \langle\langle\emptyset\rangle\rangle\varphi \text{ iff } \forall B \subseteq \mathsf{act}(Ag), t \in S: \text{ if } s \rightarrow_B t, \text{ then } M, t \vDash \varphi.$$

This means that in a state s the empty coalition has the ability to achieve all those formulas φ that are true in all the states accessible from s.

This is different in [Borgo, 2007], since here the empty coalition cannot achieve anything. Often, the states of affairs that are forced by the empty coalition are seen as the influence of a player *nature*. In [Borgo, 2007], it has been shown that player nature can be introduced explicitly to this logic which then makes it equivalent to coalition logic [Pauly, 2002].

3.2 Properties of Agents and Coalitions

In this section, we will investigate different possibilities of how to express certain properties of agents in the cooperation logic with actions [Sauro et al., 2006] presented in Section 3.1.

The properties we are particularly interested in are of the following type. Given a group G and some formula φ we would like to investigate how important some member $i \in G$ is for the group for achieving φ. We will look at some conditions for a player $i \in G$ being necessary for G to achieve φ. So, let us assume that for a group G we have the following.

$$\langle\langle G\rangle\rangle\varphi \wedge \neg\langle\langle G \setminus \{i\}\rangle\rangle\varphi.$$

So, G can achieve φ but when i leaves the group, this is not possible any more. Let us look at what we can conclude about the ability of agent i in this case; by 'ability' we mean in particular the ability to perform certain actions.

The first intuition might be that if $\langle\langle G\rangle\rangle\varphi \wedge \neg\langle\langle G \setminus \{i\}\rangle\rangle\varphi$ then there must be some atomic action such that agent i is the only one in group G who can perform this action. However, this is not the case since we can e.g. have that $\mathsf{act}(i) \subseteq \mathsf{act}(G \setminus \{i\})$ and also $\mathsf{act}(i) = \mathsf{act}(j)$ for some agent $j \in G, i \neq j$.

Consider the following example. $M = \langle S, Ac, (\rightarrow)_{A\subseteq Ac}, \Phi_0, \pi, Ag, \mathsf{act}\rangle$, where $S = \{s, t\}, Ag = \{1, 2\}, Ac = \{a\}$ and $\mathsf{act}(1) = \mathsf{act}(2) = Ac$. Define the accessibility relation as follows:

$$s \rightarrow_{\{a\}} t$$

$$s \rightarrow_{\emptyset} s$$

and in t all transitions lead to t. Define the propositional valuation such that $M, s \vDash p, M, t \vDash \neg p$. Then $M, s \vDash \langle\langle\{1, 2\}\rangle\rangle p$ and $M, s \vDash \neg\langle\langle\{2\}\rangle\rangle p$.
So, $\langle\langle G\rangle\rangle\varphi \wedge \neg\langle\langle G \setminus \{i\}\rangle\rangle\varphi$ does not mean that i has different abilities (with respect to the actions he can perform) than the other agents in the group.

The above example is in fact a special case of the following fact which says that if G can force φ and in each possible way how G can force φ there is always some action of $\mathsf{act}(i)$ that is not performed, then $G \setminus \{i\}$ cannot force φ. The idea is that if achieving φ can only be done when i does not perform all the actions $\mathsf{act}(i)$ at once, then of course i can prevent $G \setminus \{i\}$ from forcing φ by simply doing all actions in $\mathsf{act}(i)$.

Fact 3.23. *Let $M = \langle S, Ac, (\rightarrow)_{A \subseteq Ac}, \Phi_0, \pi, Ag, \mathsf{act}\rangle$ be a multi-agent system, $G \subseteq Ag$, $i \in G$ and assume that $M, s \vDash \langle\langle G\rangle\rangle\varphi$. If for every $A \subseteq \mathsf{act}(G)$ such that $M, s \vDash [\bigwedge \Phi(A, G)]\varphi$ it holds that $\mathsf{act}(i) \nsubseteq A$, then $M, s \vDash \neg\langle\langle G \setminus \{i\}\rangle\rangle\varphi$.*

Proof. Suppose that $M, s \vDash \langle\langle G \setminus \{i\}\rangle\rangle\varphi$. Then there is a set $A \subseteq \mathsf{act}(G \setminus \{i\})$ such that for all $B \subseteq \mathsf{act}(Ag \setminus (G \setminus \{i\}))$ we have that for any t such that $s \rightarrow_{A\cup B} t$ it holds that $M, t \vDash \varphi$. Then also for any $B' \subseteq \mathsf{act}(Ag \setminus G)$ and $t' \in S$: If $s \rightarrow_{A\cup\mathsf{act}(i)\cup B'} t$ then $M, t \vDash \varphi$. But this then means that $M, s \vDash [\bigwedge \Phi(A\cup\mathsf{act}(i), G)]\varphi$ which is a contradiction with our initial assumptions. Thus, $M, s \vDash \neg\langle\langle G \setminus \{i\}\rangle\rangle\varphi$. □

Next we consider what we can infer about i's strategic abilities if we have that $\langle\langle G\rangle\rangle\varphi \wedge \neg\langle\langle G \setminus \{i\}\rangle\rangle\varphi$.

Once $G \setminus \{i\}$ has made a decision about which actions to perform, i could join the other agents, i.e. the coalition $Ag \setminus G$, and cooperate in a way that they can force $\neg\varphi$. This is possible because we know that for every set of actions $A \subseteq \mathsf{act}(G \setminus \{i\})$ there is a set of actions $B \subseteq \mathsf{act}(Ag \setminus (G \setminus \{i\}))$ such that for any t such that $s \rightarrow_{A\cup B} t$, it holds that $t \vDash \neg\varphi$. So, this means that for any set of actions $A \subseteq \mathsf{act}(G \setminus \{i\})$ there are sets of actions $B_i \subseteq \mathsf{act}(i)$ and $B \subseteq \mathsf{act}(Ag \setminus G)$ such that if the concurrent action $A \cup B_i \cup B$ is performed this results in a state where φ is false.

Now for the converse: Which conditions for the action-abilities of an agent are sufficient for making him necessary for a group to achieve a certain φ?

First of all, in the simplest case if $a \in \mathsf{act}(i)$ and $s \vDash [a]\neg\varphi$, then it is clear that a group cannot enforce φ if i is not a member of the group because i can always perform a and thereby make φ false.

More generally, if there is some α such that $s \vDash [\alpha]\neg\varphi$ and $s \vDash \langle\langle\{i\}\rangle\rangle\alpha$, then it is the case that $s \vDash \neg\langle\langle G \setminus \{i\}\rangle\rangle\varphi$ also if $s \vDash \langle\langle G\rangle\rangle\varphi$.

A weaker condition is the following: There is some α such that $s \vDash [\alpha]\neg\varphi$ and there is some group $H \subseteq (Ag \setminus G)$ such that $s \vDash \langle\langle H \cup \{i\}\rangle\rangle\alpha$. Then also if $s \vDash \langle\langle G\rangle\rangle\varphi$, it holds that $s \vDash \neg\langle\langle G \setminus \{i\}\rangle\rangle\varphi$ because $s \vDash \langle\langle H \cup \{i\}\rangle\rangle\neg\varphi$.

This section has shown that the cooperation logic with actions allows us to relate the action-abilities of groups and single agents to each other and by using information about the effects of the actions, we can then infer facts about how important an agent is for some group for achieving some φ.

3.3 Uncertainty about the Next State

As we have shown earlier, the cooperation logic with actions allows us to examine the connection between the agents' abilities to perform certain actions and their abilities to force certain states of affairs, i.e. forcing the truth of certain formulas. In this section, we will look at the cooperative ability of a group in more detail and from a slightly different perspective. If a group can force some formula φ, this means that they can perform some action such that no matter what the other agents do, the resulting concurrent action will lead to a state that satisfies φ.

Remark 3.24. *Let us for now restrict our investigation to deterministic multi-agent systems. This means that we only consider those systems where for any state s and any set of atomic actions A, there is at most one state t such that $s \rightarrow_A t$. Together with our assumption that the accessibility relations are serial, this then means that there is exactly one such state t.*

Still, there remains some uncertainty of group G as to which state will actually be the next state since the only thing the group knows for sure is that it will be one that satisfies φ. In principle, it could be $2^{|\mathsf{act}(Ag \setminus G)|}$ many states that are possible (a different one for each of the possible concurrent actions of $Ag \setminus G$).

As in CL, also in the cooperation logic with actions, we talk about the cooperative ability of a group in terms of their ability to achieve the truth of certain formulas. Based on this, one might argue that the only thing a group is interested in is to achieve the truth of certain formulas and it is therefore not relevant for them which state will be actually be the one the system moves into and that satisfies that formula. On the other hand, it is also reasonable that not all states that satisfy some formula are the same to an agent, being the same meaning here that he does not prefer any strictly over the other.

In the next chapter, we will extend the cooperation logic with actions by adding a representation of the agents' preferences. We chose to use the basic preference logic [van Benthem et al., 2005] where agents' preferences are represented as a reflexive and transitive relation over the set of states. Taking this approach to preferences, it is reasonable to investigate the uncertainty of a group of agents concerning which will be the next state of the system, given that they have decided which action to perform, since it can actually really matter to the agents which exactly will be the next state.

3.3.1 Decreasing the Uncertainty about the Next State

Concerning the uncertainty as to which might be the next state, it seems reasonable that a group wants to keep this uncertainty as low as possible. One motivation for this might be that being certain about where the next step in the interactive process leads to makes it also easier to already plan the future actions.

One way of decreasing the number of states that might be possible next states is to give the other agents, i.e. the ones that are not in the group, as little power as possible with respect to determining which state will result from the concurrent action performed by all the agents together. We will investigate this problem by trying to find an action for the group G that minimizes the set of states the system might move into.

Assume that we have a multi-agent system. Then let us first look at its set of actions Ac. The actions in Ac can be partitioned into the following three disjoint sets: The set of atomic actions that only the group G can perform, the set of atomic actions that both G and $Ag \setminus G$ can perform and the set of atomic actions that only $Ag \setminus G$ can perform.

$$\begin{aligned}
\mathcal{A}^{G_{only}} &:= \{a | a \in \mathsf{act}(G) \wedge a \notin \mathsf{act}(Ag \setminus G)\}, \\
\mathcal{A}^{G, Ag \setminus G} &:= \{a | a \in \mathsf{act}(G) \cap \mathsf{act}(Ag \setminus G)\}, \\
\mathcal{A}^{(Ag \setminus G)_{only}} &:= \{a | a \in \mathsf{act}(Ag \setminus G) \wedge a \notin \mathsf{act}(G)\}.
\end{aligned}$$

Suppose that group G can force φ by doing the set of actions $A \subseteq \mathsf{act}(G)$. Now, let us look at how the other agents can contribute to the concurrent action. If they do any action that is already contained in A, then this would not contribute anything to the concurrent action that will take place and $Ag \setminus G$ could as well not do it. This is due to the fact that in our framework it is only the set of actions that are being performed that counts. Therefore, it does not matter if there is only one agent that performs some atomic action or if this action is performed by several agents at the same time.

The actions that $Ag \setminus G$ could do that have an influence on where the system finally moves in the next transition are actions that are not already contained in A, i.e. actions that G is not already doing.

If G wants to give as few power as possible to $Ag \setminus G$ with respect to determining where the system moves, then one way of doing so is to make sure that there are as few actions as possible that $Ag \setminus G$ can perform that are not already contained in the actions performed by G. Now, we will specify one procedure for G how to minimize the number of actions that $Ag \setminus G$ can perform that are not already contained in the set of actions performed by G itself. The idea is that G performs – in addition to the set A, i.e. the concurrent action that enables G to force φ, – also all the actions in $\mathcal{A}^{G, Ag \setminus G}$ that are not already contained in A. Then the only ways how $Ag \setminus G$ has some influence on where the system is going is via the actions $\mathcal{A}^{(Ag \setminus G)_{only}}$, i.e. the actions that only $Ag \setminus G$ can perform.

Now, we will show that by this procedure G can indeed decrease the number of states where the system might go into from $2^{|\mathsf{act}(Ag \setminus G)|}$ to $2^{|\mathcal{A}^{(Ag \setminus G)_{only}}|}$ (clearly, since $\mathcal{A}^{(Ag \setminus G)_{only}} \subseteq \mathsf{act}(Ag \setminus G), 2^{|\mathcal{A}^{(Ag \setminus G)_{only}}|} \leq 2^{|\mathsf{act}(Ag \setminus G)|}$) and still be sure that the next state will indeed satisfy φ. The proof essentially follows the ideas that we just presented.

Proposition 3.25. *Let $M = \langle S, Ac, (\rightarrow)_{A \subseteq Ac}, \Phi_0, \pi, Ag, \mathsf{act}\rangle$ be a multi-agent system. If for a group $G \subseteq Ag$, it holds that $M, s \models \langle\langle G \rangle\rangle \varphi$, then there is a set of actions $A' \subseteq \mathsf{act}(G)$ such that $M, s \models [\bigwedge \Phi(A', G)]\varphi$ and $|\{t | s \rightarrow_{\bigwedge \Phi(A', G) \exists} t\}| \leq 2^{|\mathcal{A}^{(Ag \setminus G)_{only}}|}$.*

Proof. Assume that $M, s \models \langle\langle G \rangle\rangle \varphi$. Then there is a set of actions $A_1 \subseteq \mathsf{act}(G)$ such that for all $B \subseteq \mathsf{act}(Ag \setminus G)$ we have that for $t \in S$ such that $s \rightarrow_{A_1 \cup B} t$ it holds that $M, t \models \varphi$. Now, consider $A' = A_1 \cup (\mathcal{A}^{G, Ag \setminus G})$. A' extends A_1 by the actions in $(\mathsf{act}(G) \setminus A_1) \cap \mathsf{act}(Ag \setminus G)$, i.e. the actions that both G and $Ag \setminus G$ can perform and that are not in A_1.

Now, we claim that $M, s \models [\bigwedge \Phi(A', G)]\varphi$. So, assume that we have $s \to_{A' \cup B'} t$ for some state t and some $B' \subseteq \mathsf{act}(Ag \setminus G)$. Since $A' = A_1 \cup C$ for $C = (\mathsf{act}(Ag \setminus G) \cap \mathsf{act}(G))$[4], we can write $A' \cup B'$ as $A_1 \cup C \cup B'$, where $C \cup B' \subseteq \mathsf{act}(Ag \setminus G)$. By the properties of A_1, we know that then for t such that $s \to_{A_1 \cup (C \cup B')} t$ it holds that $M, t \models \varphi$. Since $A_1 \cup (C \cup B') = (A_1 \cup C) \cup B'$, it follows that $M, s \models [\bigwedge \Phi(A', G)]\varphi$ which concludes the first part of the proof.
So, now it remains to show that the number of states that are accessible by an action of type $\bigwedge \Phi(A', G)$ is less than or equal to the number of sets of actions that $Ag \setminus G$ can perform but G cannot, i.e. we want to show that $|\{t | s \to_{\bigwedge \Phi(A', G)_\exists} t\}| \leq 2^{|\mathcal{A}^{(Ag \setminus G)_{only}}|}$.

Here, it is sufficient to show that the number of actions of type $\bigwedge \Phi(A', G)$ is less than or equal to the number of sets of actions that $Ag \setminus G$ can perform but G cannot. So, let us look at how many concurrent actions there are of type $\bigwedge \Phi(A', G)$. By definition of $\bigwedge \Phi(A', G)$, any concurrent action of this type has to contain A' as a subset.

By Lemma 3.18, we know that if an action is of type $\bigwedge \Phi(A', G)$ then it has to be of the form $A' \cup B$ for some $B \subseteq \mathsf{act}(Ag \setminus G)$. Since A' already contains all the actions that both G and $Ag \setminus G$ can do, we then have that any action of type $\bigwedge \Phi(A', G)$ is actually of the form $A' \cup B'$ for some $B' \subseteq (\mathsf{act}(Ag \setminus G) \setminus (\mathsf{act}(Ag \setminus G) \cap \mathsf{act}(G))) = \mathcal{A}^{(Ag \setminus G)_{only}}$. Now, it follows directly that there are $\leq 2^{|\mathcal{A}^{(Ag \setminus G)_{only}}|}$ many concurrent actions of type $\bigwedge \Phi(A', G)$. Since our multi-agent systems are deterministic, this then directly implies that there are at most $2^{|\mathcal{A}^{(Ag \setminus G)_{only}}|}$ many states accessible form s by an action of type $\bigwedge \Phi(A', G)$. This concludes the proof. □

Note that the previous analysis, we only considered the worst case, which means here that we only took care about the reducing the maximum number of states the other agent can make the system move into. Of course, in general it need not be the case that once G has decided what to do, $Ag \setminus G$ has the power to make the system move into $2^{|\mathcal{A}^{(Ag \setminus G)_{only}}|}$ many different states since in some cases the effect of the concurrent action that is taking place might just be independent of some of the actions that $Ag \setminus G$ performs. This might depend on the actions that are already performed by G.

[4] Note that C and A_1 are not necessarily disjoint.

Chapter 4

Cooperation, Actions and Preferences

After concentrating on cooperation and actions in the previous chapter, in this chapter, we will focus on agents' preferences. First, we will give an introduction to how preferences can be formalized and then we will extend multi-agent systems by adding a representation of the preferences of the agents. Moreover, we will also extend the cooperation logic with actions CLA developed by Sauro et al. [2006] that we presented in the previous chapter in such a way that we can use it to reason about preferences as well.

4.1 Preference Logic

The notion of preferences has received a lot of attention within economics, game theory, social choice theory and related fields. Developing logics for formal reasoning about preferences was basically started by von Wright [von Wright, 1963], who introduced a propositional language that expresses preferences as binary relations over propositions which range over sets of states. In this language, formulas are of the form pPq and have the intended meaning that the states of affairs p are preferred over the states of affairs q.

Since then, there have been several attempts to formalize preferences in alternative ways, several of them using some modal logic. In a model, agents' preferences can be seen as ranging over the states of the model or over formulas that represent sets of states, namely the sets of states satisfying the formulas. For an overview of different preference logics, the reader is referred to Hansson [2001] and van Benthem et al. [2007].

The motivation for adding a representation of preferences to the multi-agent systems (Definition 3.13) and the respective logic is the following.

Using the cooperation logic with actions developed by Sauro et al. [2006], we cannot only say what states of affairs a group of agents can achieve but also how exactly the group can do it, i.e. which actions result in the state of affairs under consideration. After knowing *what* a coalition can achieve and *how* it can be done, the question that comes up is:

Why would the agents choose to force this state of affairs?

This is exactly the point where the preferences of the agents play a role. An investigation of the cooperative ability of groups of agents that also considers the preferences will give us better insights into the interactive process that is taking place. Moreover, a logic combining actions, cooperation and preferences also provides us with a formal framework for investigating issues of game theory and social choice theory.

Since our aim is to extend the cooperation logics with actions to a logic that also considers preferences, let us now try to get an idea of how to add preferences to the logic we have. First, consider how the semantics has been developed in [Sauro et al., 2006]. The multi-agent systems are based on an environment consisting of states, and the performance of actions has the effect of changing the current state of the environment. Then the environment is populated by agents that can each perform certain actions. By performing actions, they can then make the system change its current state. In such a setting, one intuitive way of representing the preferences is to represent them as a binary relation over the set of states of the environment, or more precisely, a binary relation for each of the agents. Of course, there are also other possibilities. We could e.g. lift the preference relation over states to one over formulas, i.e. over the sets of states that satisfy the respective formulas [van Benthem et al., 2005; van Benthem et al., 2007].
Alternatively, when looking at some state of the system and considering some group, then the preferences of the agents can also be seen as ranging over the possible actions that they themselves or the group can perform. This would then correspond to preferences over the sets of states that the system can move into after the respective actions have been performed.

We choose to consider preferences that are ranging over states of a model, mainly because it seems to be an intuitive representation and also allows us to add the preferences to our multi-agent systems in a straightforward way. Moreover, doing it this way, we can build our logic for reasoning about preferences by using well-known modal logics. Later, it is also possible to use the logic that we chose and lift the preference relation to one over formulas.

Of course, we do not take arbitrary binary relations for representing the preferences but want them to model the intuitive idea of preferences. There have been a lot of discussions about what properties should be required. One property that seems intuitively appealing is transitivity, i.e. if a is preferred over b and b over c, then a should be preferred over c. Nevertheless, there are several cases discussed in the literature where cyclic preferences of the form $a \prec b \prec a$ occur and where transitivity causes some problems [Hansson, 2001; Schumm, 1987]. However, we will ignore those problems since they seem to arise only in very special cases. Thus, we require that the preference relation is indeed transitive. Another property that is often assumed is that of completeness or totality which means that for every two alternatives a, b and every agent i, we have that i either prefers a over b or b over a, where 'prefers' is meant to be non-strict here; with strict preferences, totality then means that i either strictly prefers a over b or the other way around or is indifferent between a and b. In what follows, we will not assume totality unless otherwise stated.

Regarding our general strategy of how to add the preferences of the agents to the multi-agent systems developed by Sauro et al. [2006], it is important to say that we will keep our approach modular. For the preferences, this means that we do not add them directly to the multi-agent system but will define a separate preference module where preferences of the agents are represented as binary relations over a set of states. We will not invent a new way of doing this but will follow the ideas presented in recent work on preference logics [van Benthem et al., 2005; van Benthem et al., 2007]. We will use a standard preference language that can be interpreted in our preference modules. The next step is then to develop an axiomatic system so that we finally obtain a sound and complete preference logic. In the remainder of this section, we follow the ideas of van Benthem et al. [2007] closely.
In what follows, assume that we have a set of propositional letters Φ_0. For representing agents' preferences, we choose models consisting of a set of states, a finite set of agents, a propositional valuation function and a reflexive and transitive preference relation for each of the agents.

Definition 4.1 (Preference Model [van Benthem et al., 2005]). *A preference model is a tuple*

$$M^P = \langle S, Ag, \{\preceq_i\}_{i\in Ag}, \pi\rangle,$$

where S is a set of states, Ag is a set of agents, for each $i \in Ag$, $\preceq_i \subseteq S \times S$ is a reflexive and transitive relation and π is a propositional valuation.

We will consider a preference language similar to the basic modal preference languages as they are defined in [van Benthem et al., 2007] and [van Benthem et al., 2005][1]. Basically, everything we do in this section directly follows from the results in Section 3 of [van Benthem et al., 2007].

The language that we will use has two normal modalities for each agent: one for preferences and one for strict preferences. Whereas having a strict preference modality turns out to cause some technical work when developing a preference logic, it seems to pay off in subsequent chapters when investigating game-theoretic concepts since there at some points it is crucial whether an agent is indifferent between two alternatives or strictly prefers one over the other.

Definition 4.2 (Preference Language). *Given a set of propositional variables Φ_0 and a finite set of agents Ag, define the preference language $\mathcal{L}_p$ to be the language generated by the following syntax:*

$$\varphi := \quad p \mid \neg\varphi \mid \varphi \vee \varphi \mid \Diamond^{\preceq_i}\varphi \mid \Diamond^{\prec_i}\varphi \ ,$$

where $p \in \Phi_0$.

The Boolean formulas are interpreted in a preference model M^P in the standard way and for $\Diamond^{\preceq}\varphi$ and $\Diamond^{\prec}\varphi$ we have

$$\begin{array}{lll} M^P, s \vDash \Diamond^{\preceq_i}\varphi & \text{iff} & \exists t \in S : s \preceq_i t \text{ and } M^P, t \vDash \varphi \\ M^P, s \vDash \Diamond^{\prec_i}\varphi & \text{iff} & \exists t \in S : s \prec_i t \text{ and } M^P, t \vDash \varphi. \end{array}$$

[1] More precisely, the preference language that we will define is a fragment of the language $\mathcal{L}_{\mathcal{P}}$ defined in [van Benthem et al., 2007], page 7. The language defined there, also contains a global existential modality E.

So, $\Diamond^{\preceq_i}\varphi$ means that there is a state that agent i prefers over the current one and that satisfies φ. $\Diamond^{\prec_i}\varphi$ expresses the same but for strict preferences.

Now, we will show how a preference logic can be developed that is later shown to be sound and complete with respect to the class of preference modeles just defined.

In order to keep the notation simple and to ease readability, we will first consider the case where there is only one agent. All the results then generalize in a natural way to the multi-agent case. As usual, we define the dual of $\Diamond^{\preceq}$ as follows $\Box^{\preceq}\varphi := \neg\Diamond^{\preceq}\neg\varphi$ and analogously for $\Diamond^{\prec}$: $\Box^{\prec}\varphi := \neg\Diamond^{\prec}\neg\varphi$.

Since we want $\preceq_i$ to be transitive and reflexive, for $\Diamond^{\preceq}$, we need the axioms of S4:

$$\begin{array}{ll}
\vdash \Diamond^{\preceq}\varphi \leftrightarrow \neg\Box^{\preceq}\neg\varphi & (\mathit{Dual}_{\Diamond^{\preceq}}) \\
\vdash \Box^{\preceq}(\varphi \rightarrow \psi) \rightarrow (\Box^{\preceq}\varphi \rightarrow \Box^{\preceq}\psi) & (K_{\Box^{\preceq}}) \\
\vdash \varphi \rightarrow \Diamond^{\preceq}\varphi & (\mathit{Refl}_{\preceq}) \\
\vdash \Diamond^{\preceq}\Diamond^{\preceq}\varphi \rightarrow \Diamond^{\preceq}\varphi & (\mathit{Trans}_{\preceq})
\end{array}$$

Now, we will consider which axioms we need for $\Diamond^{\prec}$. First of all, we also want it to be normal modal operator, so we add

$$\begin{array}{ll}
\vdash \Diamond^{\prec}\varphi \leftrightarrow \neg\Box^{\prec}\neg\varphi & (\mathit{Dual}_{\Diamond^{\prec}}), \\
\vdash \Box^{\prec}(\varphi \rightarrow \psi) \rightarrow (\Box^{\prec}\varphi \rightarrow \Box^{\prec}\psi) & (K_{\Box^{\prec}}).
\end{array}$$

We also have to take care about the connection between $\Diamond^{\preceq}$ and $\Diamond^{\prec}$. Since we want $\prec$ to be a subrelation of $\preceq$, we add

$$\vdash \Diamond^{\prec}\varphi \rightarrow \Diamond^{\preceq}\varphi \quad (\mathit{Inclusion}).$$

Moreover, we add two interaction axioms that also establish a connection between $\Diamond^{\preceq}$ and $\Diamond^{\prec}$.

$$\begin{array}{ll}
\vdash \Diamond^{\preceq}\Diamond^{\prec}\varphi \rightarrow \Diamond^{\prec}\varphi & (\mathit{Interaction}_1) \\
\vdash \Diamond^{\prec}\Diamond^{\preceq}\varphi \rightarrow \Diamond^{\prec}\varphi & (\mathit{Interaction}_2)
\end{array}$$

Note that by applying *Inclusion* and *Interaction*$_1$ successively to $\Diamond^{\prec}\Diamond^{\prec}\varphi$, *Trans*$_{\prec}$: $\Diamond^{\preceq}\Diamond^{\preceq}\varphi \rightarrow \Diamond^{\preceq}\varphi$ can be derived, which corresponds to the transitivity of $\prec$.

When showing completeness with respect to the class of preference models where the strict preference relation $\prec$ is irreflexive, some technical problems arise from the fact that the property of irreflexivity is not definable in ordinary modal logic [Blackburn et al., 2001]. This difficulty can be dealt with by applying the so called *Bulldozing* technique [Blackburn et al., 2001; van Benthem et al., 2007] to the canonical model. We will not go into the technical details here but refer the interested reader to the aforementioned literature.

We do not only want $\prec$ to be transitive and irreflexive but also to be the maximal such subrelation of $\preceq$. So, the following should be equivalent:

1. $s \prec t$

2. (a) $s \preceq t$ and
   (b) $t \not\preceq s$

The implication (1) ⇒ (2a) is taken care of by the axiom *Inclusion*. In order to ensure (1) ⇒ (2b), the bulldozing technique can be modified in an appropriate way. For the details, see [van Benthem et al., 2007].

In order to ensure that the accessibility relation satisfies also the implication (2a&b) ⇒ (1), the following axiom is added [van Benthem et al., 2007].

$$\vdash \varphi \wedge \Diamond^{\preceq}\psi \rightarrow (\Diamond^{\prec}\psi \vee \Diamond^{\preceq}(\psi \wedge \Diamond^{\preceq}\varphi)) \quad (\textit{Interaction}_3)$$

It says that if some state is accessible from the current state by $\preceq$, then it is accessible by $\prec$ or it is the current state. That *Interaction*$_3$ is the axiom we are looking for follows from the following correspondence result.

Fact 4.3 ([van Benthem et al., 2007])**.**

(*i*) *If a model M is based on a frame such that for the accessibility relation it holds that* (2a&b) ⇒ (1)*, then* $M \vDash$ (*Interaction*$_3$)*.*

(*ii*) *For every frame F such that* $F \vDash$ (*Interaction*$_3$)*, it holds that F satisfies* (2a&b) ⇒ (1)*.*

The proof is straightforward and can be found in [van Benthem et al., 2007].

Since *Interaction*$_3$ is Sahlqvist formula, in the completeness proof we can infer that for the canonical frame it holds that (2a&b) ⇒ (1).
Now, it can be shown that the preference logic defined by the axioms that we mentioned so far is indeed sound and complete with respect to the class of preference models. We will just state the result but do not present the proof here since it is quite long and analogous to that of Theorem 3.9 in [van Benthem et al., 2007], the only difference being that we do not have a global existential modality.

Proposition 4.4 (Soundness and Completeness of the Preference Logic)**.** *Let Λ^P be the logic generated by the following axioms*

1.	$\vdash \Diamond^{\preceq}\varphi \leftrightarrow \neg\Box^{\preceq}\neg\varphi$	($\mathit{Dual}_{\Diamond^{\preceq}}$)
2.	$\vdash \Box^{\preceq}(\varphi \rightarrow \psi) \rightarrow (\Box^{\preceq}\varphi \rightarrow \Box^{\preceq}\psi)$	($K_{\Box^{\preceq}}$)
3.	$\vdash \varphi \rightarrow \Diamond^{\preceq}\varphi$	($\mathit{Refl}_{\preceq}$)
4.	$\vdash \Diamond^{\preceq}\Diamond^{\preceq}\varphi \rightarrow \Diamond^{\preceq}\varphi$	($\mathit{Trans}_{\preceq}$)
5.	$\vdash \Diamond^{\prec}\varphi \leftrightarrow \neg\Box^{\prec}\neg\varphi$	($\mathit{Dual}_{\Diamond^{\prec}}$)
6.	$\vdash \Box^{\prec}(\varphi \rightarrow \psi) \rightarrow (\Box^{\prec}\varphi \rightarrow \Box^{\prec}\psi)$	($K_{\Box^{\prec}}$)
7.	$\vdash \Diamond^{\prec}\varphi \rightarrow \Diamond^{\preceq}\varphi$	(*Inclusion*)
8.	$\vdash \Diamond^{\preceq}\Diamond^{\prec}\varphi \rightarrow \Diamond^{\prec}\varphi$	(*Interaction*$_1$)
9.	$\vdash \Diamond^{\prec}\Diamond^{\preceq}\varphi \rightarrow \Diamond^{\prec}\varphi$	(*Interaction*$_2$)
10.	$\vdash \varphi \wedge \Diamond^{\preceq}\psi \rightarrow (\Diamond^{\prec}\psi \vee \Diamond^{\preceq}(\psi \wedge \Diamond^{\preceq}\varphi))$	(*Interaction*$_3$)

and closed under the rules of modus ponens, necessitation and substitution of logical equivalents Then Λ^P is sound and complete with respect to the class of preference models.

Proof. Follows from Theorem 3.9 in [van Benthem et al., 2007]. □

So, now we have a way of representing agents' preferences as binary relations over a set of states. Moreover, we have a sound and complete logic for reasoning about the preferences. Next, we will see how to combine multi-agent systems with preference models and how to extend the cooperation logic with actions (CLA)[Sauro et al., 2006] by combining it with the preference logic just defined.

4.2 Cooperation Logic with Actions and Preferences

In this section, we will extend the cooperation logic with actions (CLA) [Sauro et al., 2006] that we presented in Chapter 3 by combining it with the preference logic presented in the previous section. Technically, the obvious way would be to combine a preference model and a multi-agent system by identifying the sets of agents Ag, the sets of states S, the sets of propositional variables Φ_0 and the propositional valuation functions of the two models.

Instead of doing that directly, let us first go one step back and take a look at what it actually means for the agents in our multi-agent systems (Definition 3.13) to have preferences over the states of the environment they are living in and how such kind of preferences relate to the agents' cooperative abilities and the actions they can perform. Assume that we are in state s and for the sake of simplicity also assume that the preference orderings of the agents are complete. Then each agent has the current state s at some position in his preference ordering. This position then tells us something about how the agent "values" the current state. Of course, we are taking a very simplified view here when saying that the "happiness" of an agent about the current state is completely determined by the position of the state in the preference ordering which we assume to be static and to not change during the interactive process.

Concerning the cooperative ability of a group and the preferences of individual agents, it is at this point clearly of interest to look at the ability of a group to enforce that the system moves to a state that is preferred by a certain agent (in the group) over the current state. This corresponds to the ability of a group to improve the situation for this agent. If we can express this in the formal system that we are about to develop, then we can also talk about the ability of a group to achieve e.g. that the next state is one that is better for all its members. Also, if we look at game-theoretic solution concepts, the ability to bring about some state (or rather *outcome*) that is in a certain preference relation to the current one plays an important role.

Then the next question is how to make such kind of coalitional power more explicit, i.e. we would like to say *how* a coalition can achieve that the next state is one that is preferred by some agent i. Going one step further, we would also like to express that a group has the ability of forcing the system into some state that is preferred by agent i and that also satisfies some formula φ. We will come back to that later and now focus only on the ability of a group to enforce some state that is an improvement for some agent i. Here, the idea is very similar to the one used for making the ability to achieve the truth of some formula φ more explicit [Sauro et al., 2006]. There the idea was that the ability to achieve φ corresponds to the ability to force some complex action that is guaranteed to lead to a state satisfying φ. Analogously, the ability to achieve that the system moves to some state t that is preferred by some agent i can be seen as corresponding to the ability to force an action of some type that only leads to states that i prefers over the current one. Let us look at what exactly it means for a group G being able to enforce that the system makes a transition that leads to a state that is better than the current state according to agent i. Formally, it says:

> There is a set of actions $A \subseteq \mathsf{act}(G)$ such that for all t such that $s \to_{A \cup B} t$ for some $B \subseteq \mathsf{act}(Ag \setminus G)$, we have that $s \preceq_i t$.

Then by Lemma 3.18, this is equivalent to saying that there is some set of actions $A \subseteq \mathsf{act}(G)$ such that every action of type $\bigwedge \Phi(A, G)$ leads to a state t such that $s \preceq_i t$.

4.2.1 Environment with Preferences

Since we build our approach upon the one by Sauro et al. [2006] that we presented in Chapter 3, we also take the effects of actions to be independent from who actually performs them. Therefore, it makes sense to investigate the question of how to express that every action of some type α leads to a state preferred by some agent. First we will look at this in a general manner by considering an arbitrary action expression α without relating it to any particular group. These considerations show that it might be a good idea to first only combine the environment and the preference model by unifying their sets of states, their sets of propositional variables and their propositional valuation functions. Then later we can easily establish a connection between the cooperative ability of agents to achieve some state of affairs and their preferences. This is then done via the actions that the agents can perform and the way the performance of those actions changes the state in the environment.

As a result of combining the environment with preferences we obtain an environment module with binary preference relations for each of the agents.

Definition 4.5 (Environment with Preferences)**.** *An environment model with preferences is a tuple*

$$Env^{\preceq} = \langle S, Ac, (\to)_{A \subseteq Ac}, \Phi_0, \{\preceq_i\}_{i \in Ag}, \pi \rangle,$$

where $\langle S, Ac, (\to)_{A \subseteq Ac}, \Phi_0, \{\preceq_i\}_{i \in Ag}, \pi \rangle$ *is an environment model and* $\langle S, Ag, \{\preceq_i\}_{i \in Ag}, \pi, \rangle$ *is a preference model.*

So, an environment module with preference is a framework representing the effects of actions and the agents' preferences over the states of the environment without them being able to actively enter the environment by performing actions.

In this model, we can talk about preferences and the effects of actions as we could do it in the respective separate modules. As we argued before, we would like to talk about the ability of groups to achieve that the situation is improved according to some preference ordering (i.e. to achieve that the system moves to a state preferred over the current one) in an explicit way. Therefore, we would like to reason about the effects of actions in terms of some preference relation. So, we would like to express how the performance of some action changes the position of the current state in the preference ordering of some agent. The language that we will use to talk about such environment models with preferences consists of all the formulas of the environment language and all expressions of the preference language and one additional expression with the intended meaning that all actions of a certain type will lead to a state preferred by some agent. The idea is to have a formula $[\alpha]^{\preceq_i}\varphi$ that says that all the φ-states accessible by an action of type α are preferred by agent i over the current one. Actually, as we will see later, in our case it is sufficient to just consider the case where $\varphi = \top$. Then $[\alpha]^{\preceq}\top$ says that all states accessible by an action of type α are preferred by agent i over the current one.

Definition 4.6 (Environment-Preference Language)**.** *The language of environment modules with preferences contains all expressions of the environment language (Definition 3.4) and all expressions of the preference language (Definition 4.2). Additionally, it contains expressions of the following form:*

$$\varphi := \quad [\alpha]^{\preceq_i}\top \mid [\alpha]^{\prec_i}\top,$$

where α is an action expression and i is an agent.

The formulas from the environment language and the preference language are interpreted in the separate modules. The newly introduced formulas are interpreted in environment models with preferences as follows.

$$\begin{array}{lll} Env^{\preceq}, s \models [\alpha]^{\preceq_i}\top & \text{iff} & \forall A \subseteq Ac, t \in S: \text{ if } s \rightarrow_A t \text{ and } A \models \alpha \text{ then } s \preceq_i t \\ Env^{\preceq}, s \models [\alpha]^{\prec_i}\top & \text{iff} & \forall A \subseteq Ac, t \in S: \text{ if } s \rightarrow_A t \text{ and } A \models \alpha \text{ then } s \prec_i t \end{array}$$

So, $[\alpha]^{\preceq_i}\top$ says that agent i prefers every state that is reachable from the current state by an action of type α over the current state. $[\alpha]^{\prec_i}\top$ is analogous, but for strict preference.

Now it can easily be shown that $[\alpha]^{\preceq_i}\top$ characterizes the property that every state accessible by an action of type α is also accessible by $\preceq_i$ (and analogously for $[\alpha]^{\preceq_i}\top$ and $\prec_i$)[2]. The next step is now to try to obtain a logic that is sound and complete with respect to the class of environment models with preferences. Consider the following axioms.

1. All axioms of the environment logic Λ^E(Definition 3.6)
2. All axioms of the preference logic Λ^P (see Proposition 4.4)
3. $[\alpha]^{\preceq_i}\top \rightarrow (\Box^{\preceq_i}\varphi \rightarrow [\alpha]\varphi)$
4. $[\alpha]^{\prec_i}\top \rightarrow (\Box^{\prec_i}\varphi \rightarrow [\alpha]\varphi)$

Axiom 3 says that if $[\alpha]^{\preceq_i}\top$ then if every preferred state satisfies φ then so does every state accessible via α since $[\alpha]^{\preceq_i}\top$ says that the set of states accessible

[2] These are actually frame properties since $[\alpha]^{\preceq_i}\top$ and $[\alpha]^{\prec_i}\top$ do not depend on the propositional valuations.

via α is a subset of the set of states accessible via $\preceq_i$. Axiom 4 is analogous for strict preference.
There are still axioms missing that allow us to derive $[\alpha]^{\preceq_i}\top$. The first intuition is probably to add the following.

5 a) $(\Box^{\preceq_i}\varphi \to [\alpha]\varphi) \to [\alpha]^{\preceq_i}\top$

6 a) $(\Box^{\prec_i}\varphi \to [\alpha]\varphi) \to [\alpha]^{\prec_i}\top$

However, 5 a) and 6 a) are not sound since we might have that the set of states accessible from the current state by $\preceq_i$ and the set of states accessible by an α transition are disjoint and that both sets are nonempty. Then it can be the case that every α transition leads to a state where φ is true and all the states t such that $s \preceq_i t$ also satisfy φ but the states accessible by α are actually not preferred by i over the current state and therefore $[\alpha]^{\preceq_i}\top$ is not satisfied.

We can show that $[\alpha]^{\preceq_i}\top$ is not definable using just the preference language and the environment language. The undefinability arises from the following fact: $[\alpha]^{\preceq_i}\top$ says that every state that is accessible by an action of type α is also accessible by $\preceq_i$. So, whenever a state is accessible via an α transition *this same* state also has to be accessible from the current one via the relation $\preceq_i$. The problem is that we need to refer to particular states. This is not possible in the modal languages that we are using so far. It seems as if introducing nominals would solve the problem because then we can refer directly to particular states since each state has a name. However, adding nominals does not solve the problem; at least it is not solved by adding the following axioms for k being a nominal.

5 b) $(\langle\alpha\rangle k \to \Diamond^{\preceq_i} k) \to [\alpha]^{\preceq_i}\top$,

6 b) $(\langle\alpha\rangle k \to \Diamond^{\prec_i} k) \to [\alpha]^{\prec_i}\top$,

where $\langle\alpha\rangle k$ is equivalent to $\neg[\alpha]\neg k$.

5 b and 6 b are not the appropriate axioms that we are looking for since $[\alpha]^{\preceq_i}\top$ means that for *every* state it holds that if we can access that state via an α transition this same state also has to be accessible via the relation $\preceq_i$. If we had only finitely many states then we could take an axiom of a form similar to 5 b) and 6 b) by taking the antecedent as a conjunction over all nominals.

One way of solving the problem is by introducing additional rules of inference that allow us to infer $[\alpha]^{\preceq_i}\top$ and $[\alpha]^{\prec_i}\top$.

$$\text{(PREF-ACT)}\ \frac{\Box^{\preceq_i}\varphi \to [\alpha]\varphi}{[\alpha]^{\preceq_i}\top} \qquad\qquad \text{(STRICT PREF-ACT)}\ \frac{\Box^{\prec_i}\varphi \to [\alpha]\varphi}{[\alpha]^{\prec_i}\top}$$

With these additional rules of inference, completeness can now be shown. Since our main objective here is the clarification of conceptual issues that arise when combining the environment with a preference logic, we will not go into the technical details here but just sketch how the rules are used in the proof.

Proposition 4.7. *Let Λ^{EP} be the logic generated by the following axioms and closed under the rules modus ponens, substitution of logical equivalents, PREF-ACT and STRICT PREF-ACT.*

1. *All axioms of the environment logic* Λ^E *(Definition 3.6)*
2. *All axioms of the preference logic* Λ^P
3. $[\alpha]^{\preceq_i}\top \to (\Box^{\preceq_i}\varphi \to [\alpha]\varphi)$
4. $[\alpha]^{\prec_i}\top \to (\Box^{\prec_i}\varphi \to [\alpha]\varphi)$

Then Λ^{EP} *is sound and complete with respect to the class of environment modules with preferences.*

Proof. Soundness is straightforward. For completeness, we can construct the canonical model in the usual way. Here we take MCSs to be closed also under the newly introduced rules. The accessibility relations are defined in the usual way. Then we show a truth lemma by induction on φ. The only interesting case is where φ is of the form $[\alpha]^{\preceq_i}\top$ or $[\alpha]^{\prec_i}\top$. The other cases follow from the completeness of the logics of the submodules. We will now give a sketch of the case of $[\alpha]^{\preceq_i}\top$.

Let Σ be a MCS. Showing that $[\alpha]^{\preceq_i}\top \in \Sigma \Rightarrow \Sigma \models [\alpha]^{\preceq_i}\top$ is straightforward and makes use of Axiom 3.

The other direction (i.e., $\Sigma \models [\alpha]^{\preceq_i}\top \Rightarrow [\alpha]^{\preceq_i}\top \in \Sigma$) is the one that was problematic before we introduced the new rules of inference because deriving $[\alpha]^{\preceq_i}\top$ turned out to be difficult. Assume that $\Sigma \models [\alpha]^{\preceq_i}\top$. Then for any MCS Δ and set of atomic actions A such that $A \models \alpha$ and $\Sigma \to_A \Delta$, it holds that $\Sigma \preceq_i \Delta$. Now suppose there is some ψ such that $\Box^{\preceq_i}\psi \wedge \neg[\alpha]\psi \in \Sigma$. Then by inductive hypothesis and the previous cases, $\Sigma \models \Box^{\preceq_i}\psi$ and $\Sigma \models \neg[\alpha]\psi$. From $\Sigma \models \neg[\alpha]\psi$ it follows that there is some Δ' such that $\Sigma \to_{A'} \Delta'$ for some A' such that $A' \models \alpha$ and $\Delta' \models \neg\psi$. Then by one of the previous cases, $\neg\psi \in \Delta'$. But since $\Sigma \models [\alpha]^{\preceq_i}\top$ and Δ' is accessible from Σ by a transition of type α, it follows that $\Sigma \preceq_i \Delta'$. As $\Box^{\preceq_i}\psi \in \Sigma$, we conclude that $\psi \in \Delta'$, which contradicts the fact that $\neg\psi \in \Delta'$. So, it cannot be the case that there is a ψ such that $\Box^{\preceq_i}\psi \wedge \neg[\alpha]\psi \in \Sigma$. Hence, we conclude by $\mathrm{PREF}-\mathrm{ACT}$ that $[\alpha]^{\preceq_i}\top \in \Sigma$. This concludes the sketch of the proof. The case of $[\alpha]^{\prec_i}\top$ is similar but a bit more involved since $\prec_i$ is irreflexive. □

These considerations show that adding two rules of inference enables us to obtain a logic sound and complete with respect to the class of environment modules with preferences. The important point about our logic is that it can express that an action of a certain type is guaranteed to lead to a state preferred by some agent.

We will not go into more technical investigations at this point but move on with our main objective, namely to build a multi-agent system with preferences so that we can actually reason about the cooperative ability of agents by also taking into account how they can achieve some state of affairs and how this ability relates to their preferences. When then developing a respective logic, we will make use of the expressions $[\alpha]^{\preceq_i}\top$ and $[\alpha]^{\prec_i}\top$ because they allow us to explicate the ability of group to achieve that the system moves to some preferred state satisfying some formula φ.

4.2.2 Cooperation Logic with Actions and Preferences

As we already mentioned, if we take the preferences of agents to range over states then one straightforward way of adding preferences to the cooperation logic with actions (CLA) [Sauro et al., 2006] is to let the preferences range over the set of states of the environment. So, the semantic structure that is then obtained extends a multi-agent system by a preference relation for each agent over the set of states.

Definition 4.8. *A multi-agent system with preferences $M^{\preceq}$ is a tuple*

$$\langle S, Ac, (\rightarrow)_{A \subseteq Ac}, \Phi_0, Ag, \mathsf{act}, \{\preceq_i\}_{i \in Ag}, \pi \rangle,$$

where $\langle S, Ac, (\rightarrow)_{A \subseteq Ac}, \Phi_0, \pi, Ag, \mathsf{act} \rangle$ is a multi-agent system, $\langle S, Ag, \{\preceq_i\}_{i \in Ag}, \pi \rangle$ is a preference model and $\langle S, Ac, (\rightarrow)_{A \subseteq Ac}, \Phi_0, \{\preceq_i\}_{i \in Ag}, \pi \rangle$ is an environment with preferences.

So, a multi-agent system with preferences is a multi-agent system where the preferences of each of the agents are represented by a binary preference relation over the set of states of the system. Relating a multi-agent system with preferences to an environment with preferences as previously defined, the multi-agent system with preferences can be seen as an extension of an environment with preferences that adds the agents' abilities to act in the environment (and thereby change its current state) by performing actions.

In a multi-agent system with preferences, all expressions that are interpreted in the submodules are still of interest. Additionally, we would like to talk about the agents' cooperative abilities to achieve the truth of some formula and to control how the new state is compared to the old one in terms of the agents' preferences.

The next step is to define a language that we will use for investigating the connections between cooperative ability, actions and preferences in the multi-agent systems with preferences. Clearly, we want to keep all the expressions of the language for multi-agent systems, and also the preference language. Moreover, as we argued above, we would also like to be able to say that a group has the ability to make the system move to some state satisfying φ that is preferred over the current state by some agent.

Definition 4.9. *The language for multi-agent systems with preferences extends the language of multi-agent systems (Definition 3.14) by formulas of the form*

$$\Diamond^{\preceq_i}\varphi \mid \Diamond^{\prec_i}\varphi \mid [\alpha]^{\preceq_i}\top \mid [\alpha]^{\prec_i}\top \mid \langle\langle G^{\preceq_i}\rangle\rangle\varphi \mid \langle\langle G^{\prec_i}\rangle\rangle\varphi.$$

The first four kinds of expressions are interpreted in the preference model and the environment with preferences. Formulas of the last two forms have the intended meaning of G having the ability to force the system to move into a state that is (strictly) preferred over the current one by agent i and that satisfies φ. Formally, they are interpreted in the following way.

$$\begin{array}{lll}
M^{\preceq}, s \models \langle\langle G^{\preceq_i}\rangle\rangle\varphi & \text{iff} & \exists A \subseteq \mathsf{act}(G) \text{ such that } \forall B \subseteq \mathsf{act}(Ag \setminus G), t \in S: \\
 & & \text{if } s \rightarrow_{A \cup B} t, \text{ then } M^{\preceq}, t \models \varphi \text{ and } s \preceq_i t \\
M^{\prec}, s \models \langle\langle G^{\prec_i}\rangle\rangle\varphi & \text{iff} & \exists A \subseteq \mathsf{act}(G) \text{ such that } \forall B \subseteq \mathsf{act}(Ag \setminus G), t \in S: \\
 & & \text{if } s \rightarrow_{A \cup B} t, \text{ then } M^{\preceq}, t \models \varphi \text{ and } s \prec_i t
\end{array}$$

Analogously to Proposition 3.19, we can also relate the cooperative ability of a group to achieve some state of affairs in a way that the transition will be an improvement for agent i to the group's ability to force an action that has the desired effects, i.e. it leads to a state that is preferred by agent i and that also satisfies φ. The proof is analogous to that of Proposition 3.19 and also uses Lemma 3.18.

Proposition 4.10. *Given a multi-agent system with preferences $M^{\preceq}$ and a state s of its environment,*

$$M^{\preceq}, s \vDash \langle\langle G^{\preceq_i}\rangle\rangle\varphi \quad \textit{iff} \quad \begin{array}{l}\textit{there exists an action expression } \alpha \textit{ such that}\\ M^{\preceq}, s \vDash \langle\langle G\rangle\rangle\alpha,\ M^{\preceq}, s \vDash [\alpha]\varphi \textit{ and } M^{\preceq}, s \vDash [\alpha]^{\preceq_i}\top.\end{array}$$

Proof. ($\Rightarrow$) Assume that $M^{\preceq}, s \vDash \langle\langle G^{\preceq_i}\rangle\rangle\varphi$. This means that $\exists A \subseteq \mathsf{act}(G)$ such that for all $B \subseteq \mathsf{act}(Ag \setminus G)$ and for all states $t \in S$, if $s \rightarrow_{A\cup B} t$ then $M^{\preceq}, t \vDash \varphi$ and $s \preceq_i t$.

Take such an A. Now, we show that $\bigwedge \Phi(A, G)$ is the action expression that we are looking for.

By definition of $\bigwedge \Phi(A, G)$, $M^{\preceq}, s \vDash \langle\langle G\rangle\rangle \bigwedge \Phi(A, G)$. For showing that $M^{\preceq}, s \vDash [\bigwedge \Phi(A, G)]\varphi$ and $M^{\preceq}, s \vDash [\bigwedge \Phi(A, G)]^{\preceq_i}\top$, let $s \rightarrow_C t$ for some $C \subseteq Ac$ such that $C \vDash \Phi(A, G)$. Then by Lemma 3.18, $C = A \cup B$ for some $B \subseteq \mathsf{act}(Ag \setminus G)$ and since A was a set of actions by whose performance G can force that the system moves into a state where φ is true and that is also preferred by i, it follows that $M^{\preceq}, t \vDash \varphi$ and $s \preceq_i t$.

Thus, $M^{\preceq}, s \vDash \langle\langle G\rangle\rangle \bigwedge \Phi(A, G)$, $M^{\preceq}, s \vDash [\bigwedge \Phi(A, G)]\varphi$ and also $M^{\preceq}, s \vDash [\bigwedge \Phi(A, G)]^{\preceq_i}\top$.

($\Leftarrow$) Let α be an action expression such that $M^{\preceq}, s \vDash \langle\langle G\rangle\rangle\alpha, M^{\preceq}, s \vDash [\alpha]\varphi$ and $M^{\preceq}, s \vDash [\alpha]^{\preceq_i}\top$. By definition of $\langle\langle G\rangle\rangle\alpha$, there must be an $A \subseteq \mathsf{act}(G)$ such that for all $B \subseteq \mathsf{act}(Ag \setminus G)$ $A \cup B \vDash \alpha$. Since also $M^{\preceq}, s \vDash [\alpha]\varphi$ and $M^{\preceq}, s \vDash [\alpha]^{\preceq_i}\top$, we have that for all t such that $s \rightarrow_{A\cup B}$ for some $B \subseteq \mathsf{act}(Ag \setminus G)$: $M^{\preceq}, t \vDash \varphi$ and $s \preceq_i t$. Thus, $M^{\preceq}, s \vDash \langle\langle G^{\preceq_i}\rangle\rangle\varphi$. □

Like Proposition 3.19, Proposition 4.10 enables us to reduce the ability of a group to achieve a certain state of affairs to expressions that we can interpret in the submodules. This is very useful when trying to show completeness.

An analogous proposition for strict preferences $\prec_i$ instead of $\preceq_i$ can also be shown; just replace every occurrence of $\preceq_i$ by $\prec_i$ in the statement of Proposition 4.10 and its proof.

Let us now look at how to axiomatize the logic for multi-agent systems with preferences. Besides the axioms of the submodules, i.e. the axioms and rules (also the newly introduced rules (PREF − ACT) and (STRICT PREF − ACT)) of the cooperation logic with actions (Definition 3.20), the environment logic with preferences Λ^{EP} (see Proposition 4.7) and the preference logic Λ^{P} (see Proposition 4.4), we need axioms that establish a relationship between the newly added formulas $\langle\langle G^{\preceq_i}\rangle\rangle\varphi$ and $\langle\langle G^{\prec_i}\rangle\rangle\varphi$ and expressions from the submodules.

Definition 4.11 (Cooperation Logic with Actions and Preferences)**.** *Define Λ^{CLAP} to be the smallest logic containing the following axioms*

1. *All axioms of the cooperation logic with actions (Definition 3.20)*
2. *All axioms of the environment logic with preferences (see Proposition 4.7)*

3. $(\langle\langle G\rangle\rangle\alpha \wedge [\alpha]\varphi \wedge [\alpha]^{\preceq_i}\top) \rightarrow \langle\langle G^{\preceq_i}\rangle\rangle\varphi$

4. $(\langle\langle G\rangle\rangle\alpha \wedge [\alpha]\varphi \wedge [\alpha]^{\prec_i}\top) \rightarrow \langle\langle G^{\prec_i}\rangle\rangle\varphi$

5. $\langle\langle G^{\preceq_i}\rangle\rangle\varphi \rightarrow \bigvee\{\langle\langle G\rangle\rangle\alpha \wedge [\alpha]\varphi \wedge [\alpha]^{\preceq_i}\top | \alpha$ *is a conjunction of action literals*$\}$

6. $\langle\langle G^{\prec_i}\rangle\rangle\varphi \rightarrow \bigvee\{\langle\langle G\rangle\rangle\alpha \wedge [\alpha]\varphi \wedge [\alpha]^{\prec_i}\top | \alpha$ *is a conjunction of action literals*$\}$

and being closed under modus ponens, substitution of logical equivalents, PREF − ACT *and* STRICT PREF − ACT.

The axioms 3-6 probably seem familiar to the reader. They are direct analogs of axioms of the cooperation logic with actions (Definition 3.20). We can show that Λ^{CLAP} is sound and complete with respect to the class of multi-agent systems. The result basically follows from the completeness of the logics for the submodules. Again we will not present all the technical details but just give the main idea of the proof.

Proposition 4.12. *The logic Λ^{CLAP} is sound and complete with respect to the class of multi-agent systems with preferences.*

Proof. Soundness is straightforward. For completeness, we can construct a canonical model by combining the ones of the logics of the submodules. Proving the truth lemma is done by induction on φ. Here the interesting case is when φ is of the form $\langle\langle G^{\preceq_i}\rangle\rangle\psi$ or $\langle\langle G^{\prec_i}\rangle\rangle\psi$.

We will only sketch the case of $\langle\langle G^{\preceq_i}\rangle\rangle\psi$ here. Assume that in the canonical model $\Sigma \models \langle\langle G^{\preceq_i}\rangle\rangle\psi$. Then by Proposition 4.10, there is an action expression α such that $\Sigma \models \langle\langle G\rangle\rangle\alpha$, $\Sigma \models [\alpha]\psi$ and $\Sigma \models [\alpha]^{\preceq_i}\top$. Then by inductive hypothesis and the previous cases, we can conclude that $\langle\langle G\rangle\rangle\alpha \in \Sigma$, $[\alpha]\psi \in \Sigma$ and $[\alpha]^{\preceq_i}\top \in \Sigma$. Then by maximality of Σ and Axiom 3, $\langle\langle G^{\preceq_i}\rangle\rangle\psi \in \Sigma$.

For the other direction, let $\langle\langle G^{\preceq_i}\rangle\rangle\psi \in \Sigma$. Then because Σ is maximal, by Axiom 5 there is a set of action literals $\mathcal{A}$ such that $[\bigwedge\mathcal{A}]\psi \in \Sigma, [\bigwedge\mathcal{A}]^{\preceq_i}\top \in \Sigma$ and $[\bigwedge\mathcal{A}]\psi \in \Sigma$. Then by inductive hypothesis and the previous cases $\Sigma \models \langle\langle G\rangle\rangle\bigwedge\mathcal{A}$, $\Sigma \models [\bigwedge\mathcal{A}]\psi$, $\Sigma \models [\bigwedge\mathcal{A}]^{\preceq_i}\top$. Then by Proposition 4.10, $\Sigma \models \langle\langle G^{\preceq_i}\rangle\rangle\psi$. The case for $\langle\langle G^{\prec_i}\rangle\rangle\psi$ is analogous. □

Let us now briefly summarize this section. We developed a logical framework extending the cooperation logic with actions developed by Sauro et al. [2006]. We combined this logic with a basic preference logic similar to that presented by van Benthem et al. [2007]. The combination of the semantic structures is done by letting the preferences of the agents range over the states of the environment they are living in. Then we obtain a framework for reasoning about the agents' cooperative abilities to achieve certain state of affairs, as in CL [Pauly, 2002]. Moreover, we can say explicitly how they can achieve some states, in the same manner as in CLA [Sauro et al., 2006]. Furthermore, in our logic we also have representations of the agents' preferences and can also reason about how a group can achieve that the next state the system will move into is one that satisfies some formula φ and that is preferred by a certain agent over the current state. The modularity of our approach has the advantage that we can also investigate cooperative ability, actions and preferences separately in the respective submodules. Technically, the modularity also makes it easier to show completeness.

The following diagram gives an illustration of how the cooperation logic with actions and preferences is built and provides the reader with references of where to find the respective logics in this thesis.

$$\begin{tikzcd}
& & \boxed{\Lambda^{CLAP} \text{ (p. 50)}} & & \\
& \boxed{\Lambda^{CLA} \text{ (p. 30)}} \arrow[ur] & & \boxed{\Lambda^{EP} \text{ (p. 47)}} \arrow[ul] & \\
\boxed{\Lambda^{A} \text{ (p. 26)}} \arrow[ur] & & \boxed{\Lambda^{E} \text{ (p. 24)}} \arrow[ul] \arrow[ur] & & \boxed{\Lambda^{P} \text{ (p. 43)}} \arrow[ul]
\end{tikzcd}$$

4.3 Discussion

In this section, we will take a closer look at some interesting aspects of the logic just defined.

4.3.1 Group Preferences

In the previous section, we added preferences of individual agents to the logic. Since we are looking at the cooperative abilities of groups, it is also interesting to not only consider the preferences of the single agents individually but also preferences of a whole group of agents because it is the group preference that determines what actions the group chooses to perform if the agents make this decision collectively as a group.

This issue of how to obtain group preferences from individual preferences is dealt with in social choice theory (see [Arrow, 1970] for one of the early works in this field). The main difficulty in aggregating individual preferences for forming a group preference is that properties of the individual preference relations (e.g transitivity) may not be preserved.

Instead of taking the preferences of individual agents as primitives, we could have also considered group preferences for each group of agents as primitives. One advantage of taking preferences of individuals is that we could base the preference module on an existing logic in a straightforward way. Moreover, when taking individual preferences as primitives we can also investigate the ability of a group to achieve something that e.g. the majority of its members prefers. If we take group preferences as primitives such investigation do not make so much sense any more.

4.3.2 Relation to Game Theory

As our approach is based on CL (at least the agents module for reasoning about the cooperative ability of groups to achieve the performance of certain complex actions), when adding preferences we do not obtain a direct correspondence to coalitional games meaning that our logic cannot be interpreted directly in a coalitional game by taking the outcomes to be the states. This follows from the fact that the same holds for CL [Ågotnes et al., 2006]. Nevertheless, the multi-agent systems with preferences as we developed them and the corresponding logic provide us with a logical framework that we can use to investigate game-theoretic issues. One advantage of our approach is that we can investigate the cooperative abilities on different levels, e.g. by focussing only on the single aspects of cooperation actions and preferences as they are dealt with in the sub-modules, and also the relationships between the cooperative ability to achieve some result, the individual agents' preferences and the ability to perform certain actions. As we will see in Chapter 5, we can obtain several results from the field of mechanism design in our framework. As opposed to the standard game-theoretic results, they also tell us something about how the distribution of the action abilities among the agents affects the power of individual agents or certain groups to influence the course of the interactive process.

4.3.3 Possible Extensions

Looking back at the approaches presented in the literature overview in Chapter 2, there are two immediate ways of extending our approach that we will shortly mention here. First of all, we can extend the cooperation logic with actions and preferences by adding a restricted kind of quantification following the ideas of Ågotnes et al. [2007b] such that we obtain formulas of the form $\langle\langle P^{\preceq_i}\rangle\rangle\varphi$ meaning that there is some group G that has property P and $\langle\langle G^{\preceq_i}\rangle\rangle\varphi$. The quantification could be added in a way analogous to the one described by Ågotnes et al. [2007b].

Another idea of extending our logic is to consider the agents' abilities to execute complex plans instead of just concurrent actions. Then we obtain a logic like that developed by Gerbrandy and Sauro [2007] but with the additional representation of the agents' preferences.

Chapter 5

Mechanism Design and Multi-Agent Systems with Preferences

The aim of this section is to investigate the concept of mechanism design in our framework of multi-agent systems with preferences. First, we will give an introduction to mechanism design in game theory and then try to relate it to our present framework.

5.1 Mechanism Design in Game Theory

The general idea of mechanism design, also known as *implementation theory* [Osborne and Rubinstein, 1994], is the following.

Instead of formulating a model that captures some interactive situation and investigating the set of outcomes that are consistent with some solution concept, in mechanism design one somehow goes the other way around. Besides a set of agents, there is also one distinguished individual that has a special role. This individual is called the *planner*. For each preference profile of the agents over the set of outcomes, the planner chooses an outcome that she wants to associate with this preference profile. This means that for each preference profile the planner picks an outcome (or a set of outcomes) that she wants the game to end up in given that preference profile. The task for the planner is then to design a game , i.e. to design the rules of a game, in such a way that whenever the players play the game according to the rules, it leads to the outcome that the planner associated with the preference profile of the players.

As an example, consider the situation where we have two agents and a planner who wants to assign one object to one of them [Osborne and Rubinstein, 1994]. The planner does not know how the agents value the object but she wants to assign the object to the agent that values it the most. Then in this case the planner wants to design the rules of the game such that the object is always assigned to the player who values it the most.

Next, let us see how to formalize such implementation problems. We will use the definitions presented in Osborne and Rubinstein [1994].

As explained above, for each preference profile of the agents, the planner chooses a set of outcomes that she wants the game to lead to when the players have these preferences.

Definition 5.1 (Choice Rule, Choice Function [Osborne and Rubinstein, 1994])**.** *Let N be a set of individuals, and let C be a set of feasible outcomes. A choice rule is a function $f : \mathcal{P} \to 2^C$ mapping preference profiles $\preceq \in \mathcal{P}$ to subsets of the set of feasible outcomes C. If f is singleton valued, it is called a choice function.*

Now, we will formalize what exactly it means for the planner to determine the rules of the game.

Definition 5.2 (Strategic Game [Osborne and Rubinstein, 1994])**.** *A strategic game is a tuple $\langle N, (A_i)_{i \in N}, (\preceq_i)_{i \in N} \rangle$, where N is the set of players, A_i is the set of actions available to player i, and $\preceq_i \subseteq (\times_{j \in N} A_j) \times (\times_{j \in N} A_j)$ is the preference relation for player i.*

It is important to note that in a strategic game as just defined, the preferences of the agents range over action profiles. This is reasonable because given an action profile, in a deterministic game there is a unique outcome that is determined by this action profile. Having preferences over action profiles means that the agents have preferences over the possible ways a game can evolve.

Definition 5.3 (Strategic Game Form [Osborne and Rubinstein, 1994])**.** *A strategic game form with consequences in C is a triple $\langle N, (A_i), g \rangle$ where A_i, for each $i \in N$ is the set of actions available to player i and $g : \times_{i \in N} A_i \to C$ is an outcome function that associates an outcome with every action profile.*

A strategic game form together with a preference profile $(\preceq_i)_{i \in N}$ induces a strategic game $\langle N, (A_i)_{i \in N}, (\preceq'_i)_{i \in N} \rangle$, where $\preceq'_i$ is defined as $a \preceq'_i b$ iff $g(a) \preceq_i g(b)$.

It is important to determine what information the planner has.

The environment in which the planner works consists of

- a finite set of players $N = \{1, 2, \ldots, n\}$
- a set C of outcomes
- a set $\mathcal{P}$ of preference profiles
- a set $\mathcal{G}$ of strategic game forms with consequences in C.

This means that the planner knows which players are taking part in the game, what are their possible preferences and it is also known which is the set of outcomes of interest. Moreover, the planner has a set of strategic game forms of the form $\langle N, (A_i)_{i \in N}, g \rangle$ with consequences in C. The set $\mathcal{G}$ is the set of strategic game forms, the planner can choose from.

After determining a choice rule f, the planner has to choose a $G \in \mathcal{G}$ that implements the choice rule, i.e. that leads to the same outcomes as the ones determined by the choice rule for each preference profile.

When solving an implementation problem, the planner has to take into account how the players will play any possible game. This means that given a preference profile and a strategic game form, the planner wants to predict how the players will behave in the process of playing the game. Here, she has to make some assumptions about which strategies the players adopt given their preferences. This means that the planner assumes that the players are playing according to some solution concept.

Definition 5.4 (Solution Concept)**.** *A solution concept for the environment* $\langle N, C, \mathcal{P}, \mathcal{G} \rangle$ *is a function* $\mathcal{S} : \mathcal{G} \times \mathcal{P} \to 2^{A_1 \times A_2 \times \ldots \times A_n}$

Finally, we can formalize what it means to implement a choice rule.

Definition 5.5. *Let* $\langle N, C, \mathcal{P}, \mathcal{G} \rangle$ *be an environment and let* $\mathcal{S}$ *be a solution concept. The game form* $G \in \mathcal{G}$ *with outcome function* g *is said to* $\mathcal{S}$*-implement the choice rule* $f : \mathcal{P} \to C$ *if for every preference profile* $\preceq \in \mathcal{P}$ *we have* $g(\mathcal{S}(G, \preceq)) = f(\preceq)$*. Then we say that the choice rule* f *is* $\mathcal{S}$*-implementable in* $\langle N, C, \mathcal{P}, \mathcal{G} \rangle$

If f is $\mathcal{S}$-implementable, this means that assuming that all the players play according to S, then the planner can set the rules of the game in such a way that for every preference profile of the players, their play is also in accordance with the choice rule.

5.2 Mechanism Design in the Framework of Multi-agent Systems with Preferences

Now, the next step is to investigate mechanism design in the framework of multi-agent systems with preferences. Recall that the job of the planner is to design the rules of a game in such a way that for every preference profile under consideration the possible outcomes will always be exactly those that the choice rule assigned to the preference profile. The planner has to pick from a set of game forms the one that assigns outcomes to action profiles in such a way such that if the players play in accordance to a certain solution concept, then no matter what their preferences are, the outcomes of the game will be as specified by the choice rule.

Now, the task is to transfer this to our framework of multi-agent systems. The idea is the following. After specifying her choice rule, the planner has to pick from a set of multi-agent systems that all have the same states but different accessibility relations. She has to do it in such a way that for the system she picks the following holds: Assume that the agents choose their actions according to a certain solution concept. Then for each preference profile under consideration, the agents will always act such that the possible next states the system can move into as a result of the actions of the agents are exactly those selected by the choice rule for the preference profile under consideration.

One thing that is important to note at this point is that in the multi-agent systems that we defined, one effect of the modular approach is that the effects of actions are independent of who exactly performs them. The effects of actions are determined in the environment module which is independent of the agents. If some action is performed, it has a certain effect which does not depend on who performed the action.

In the case of strategic game forms, this independence is only given if the outcome function is such that it assigns the same outcome to action profiles if taking the union of the actions results in the same sets, i.e

$$g(\langle A_1, \dots, A_n \rangle) = g(\langle B_1, \dots, B_n \rangle) \text{ if } \bigcup_{i \in Ag} A_i = \bigcup_{i \in Ag} B_i.$$

Let us now formalize our ideas of mechanism design for multi-agent systems. As we already mentioned, the planner is given a set of multi-agent systems that only differ in their accessibility relation, i.e. they have the same set of states propositional valuations, the same set of agents an agents have the same abilities with respect to which actions they can perform.

Definition 5.6 (Static Multi-agent System)**.** *A static multi-agent system is a tuple $\langle S, Ac, \Phi_0, \pi, Ag, \mathsf{act} \rangle$, where S is a set of states, Ac a set of atomic actions, Φ_0 a set of propositional variables, π a propositional valuation, Ag a set of agents and* act *a function* $\mathsf{act} : Ag \to 2^{Ac}$ *assigning sets of actions to each agent.*

So, a static multi-agent system is just a multi-agent system without accessibility relation.

We say that a multi-agent system M is based on a static multi-agent system $\bar{M}$ if it extends it by an accessibility relation.

Now, the environment the planner is working in is a set of multi-agent systems based on the same static multi-agent system, i.e. they only differ in their accessibility relations. Then the planner's task is to pick one of them that implements the desired choice rule. A choice rule assigns to each pair of preference profile and state a set of states that the planner wants the system to move into in the next transition.

Definition 5.7 (Choice Rule for Multi-agent Systems)**.** *Given a set of preference profiles $\mathcal{P}$ and a multi-agent system $M = \langle S, Ac, (\rightarrow)_{A \subseteq Ac}, \Phi_0, \pi, Ag, \mathsf{act} \rangle$, a choice rule for multi-agent systems is a function $f : \mathcal{P} \times S \to 2^S$.*

Next, we define a multi-agent analog of solution concepts as we presented them in the previous section. Given a multi-agent system, a preference profile and a state in the system, a solution concept gives us a set of concurrent actions.

Definition 5.8 (Solution Concept for Multi-agent Systems)**.** *Let $\mathbb{M}$ be a set of multi-agent systems based on a static multi-agent system $\bar{M} = \langle S, Ac, \Phi_0, \pi, Ag, \mathsf{act} \rangle$ and let $\mathcal{P}$ be a set of preference profiles of the agents in Ag over the set of states S. Then a solution concept for $\mathbb{M}$ is a function $\mathcal{S} : \mathcal{P} \times \mathbb{M} \times S \to 2^{2^{Ac}}$.*

Remark 5.9. *In what follows we will make the following two assumptions:*

- *We will assume that the multi-agent systems under consideration are deterministic, i.e. for every state s and set of actions $A \subseteq Ac$ there is at most one state t such that $s \rightarrow_A t$. Together with the property of seriality that we assumed this then means that for every state s and set of actions $A \subseteq Ac$ there is exactly one state t such that $s \rightarrow_A t$.*

- *We will also assume that the preferences of the agents are complete, i.e. for all agents i we have that for any pair of states s, t: $s \preceq_i t$ or $t \preceq_i s$.*

Now we can define what it means for a multi-agent system to implement a choice rule.

Definition 5.10 (Implementation in a Multi-agent System)**.** *Let $\mathbb{M}$ be a set of multi-agent systems based on the static multi-agent system $\bar{M} = \langle S, Ac, \Phi_0, \pi, Ag, \mathsf{act}\rangle$ and let $\mathcal{S}$ be a solution concept. The multi-agent system $M \in \mathbb{M}$ with accessibility relation $(\rightarrow)_{A \subseteq Ac}$ is said to $\mathcal{S}$-implement the choice rule f if for every preference profile $\preceq \in \mathcal{P}$ we have*

$$t \in f(\preceq, s) \text{ iff } \exists A \in \mathcal{S}(\preceq, M, s) \text{ such that } s \rightarrow_A t$$

Then we say that the choice rule f is $\mathcal{S}$-implementable in $\mathbb{M}$.

So, a choice rule is implemented by a multi-agent system if for every state and for every preference profile, the states selected by the choice rule are exactly those states that the system can move into if the all the agents choose their actions according to the solution concept under consideration.

In order to summarize this chapter we will give the main idea of mechanism design in multi-agent systems:

Implementation Problem in Multi-agent Systems – Summary

The planner is given a set of states of the environment and a set of agents that can each perform some set of atomic actions. Furthermore, we have a set of preference profiles of the agents over the set of states of the environment. The planner has a choice rule that she wants to implement. This means that for every state and preference profile, she has a certain set of states that she wants the system to move into, given the state and the preferences. Then the planner sets the accessibility relation for each of the states, i.e. the effects of actions being performed in that state, in such a way that – assuming the agents choose their actions according to a certain solution concept – the system moves exactly as the planner wants. This means that for each state and preference profile the system will move to one of the states that the choice rule selected for this pair of state and preference profile.

Next, we will take a closer look at some particular solution concepts and their implementation problems.

5.3 Implementation Problems in Multi-agent Systems

In this section, we will give an overview of several solution concepts and investigate some of the properties of choice rules that are implementable in multi-agent systems when the agents interact according to those solution concepts.

5.3.1 Maximin Implementation

The first solution concept that we will look at is the maximin solution concept. The general idea is that players play in such a way that they maximize the minimal outcome that they will get. This means that for every alternative action that they can perform they look at what will be the outcome in the worst case. Then they choose the action that has the best worst case outcome. Agents that play according to maximin strategies can be seen as pessimistic because they choose their next action by only considering the worst possible outcomes of their actions.

So, let us assume that all the agents in a multi-agent system choose their next action according to the maximin strategy. This means that every agent does the following: For every action that he can perform he checks what are the possible states the system can move into when he chooses that action (this depends of course on the actions of the other agents). Then he determines which is the worst such state. After doing that for every action that is available to him, he compares the respective worst states and finally chooses the action that has the best worst case outcome or he chooses one of them if there is more than one such action.

Now, we can define the concept of a maximin optimal action for an agent is defined as follows.

Definition 5.11 (Maximin Optimality). *Let* $M^{\preceq} = \langle S, Ac, (\rightarrow)_{A \subseteq Ac}, \Phi_0, Ag, \mathsf{act}, \{\preceq_i\}_{i \in Ag}, \pi\rangle$ *be a multi-agent system with preferences. We say that an action* $\hat{A}_i \subseteq \mathsf{act}(i)$ *is maximin optimal for agent* $i \in Ag$ *in state* s *if*

$$\min_{\preceq_i}\left(\bigcup_{B \subseteq \mathsf{act}(Ag \setminus \{i\})} \{t | s \rightarrow_{\hat{A}_i \cup B} t\}\right) \succeq_i \min_{\preceq_i}\left(\bigcup_{B \subseteq \mathsf{act}(Ag \setminus \{i\})} \{t | s \rightarrow_{A_i \cup B} t\}\right)$$

for all $A_i \subseteq \mathsf{act}(i)$.

So, an action is maximin optimal for an agent i if the worst state the system can move into after i doing this action is at least as good as the worst state the system can move into after i doing any other action. Now, we can define the maximin solution concept which gives us for every state the set of concurrent actions that can take place if every agent chooses to perform an action that is maximin optimal for him.

Definition 5.12 (Maximin Solution Concept). *The maximin solution concept is a function* $\mathcal{S}_{maximin} : \mathcal{P} \times \mathbb{M} \times S \rightarrow 2^{2^{Ac}}$, $\mathcal{S}_{maximin}(\langle \preceq, M, s\rangle) \mapsto \mathcal{A}$, *where* $\mathcal{A}$ *is defined as follows.*

$$\mathcal{A} := \{A | A = \bigcup_{i \in Ag} A_i, \text{ where } A_i \subseteq \mathsf{act}(i) \text{ is a maximin optimal action for } i \text{ in } s\}.$$

A choice rule can then be maximin implemented by a multi-agent system if, under the assumption that all the agents choose maximin optimal actions, the system moves from one state to another exactly like the planner wants, i.e. exactly like the choice rule says.

Definition 5.13 (Maximin Implementability). *Let $\mathbb{M}$ be a set of multi-agent systems based on the static multi-agent system $\bar{M} = \langle S, Ac, \Phi_0, \pi, Ag, \mathsf{act}\rangle$ and let $G \subseteq Ag$. The multi-agent system $M \in \mathbb{M}$ with accessibility relation $(\rightarrow_A)_{A \subseteq Ac}$ is said to maximin implement the choice rule f if for every preference profile $\preceq \in \mathcal{P}$ and for all states $s \in S$ it holds that*

$$t \in f(s, \preceq) \text{ iff } \exists A \in \mathcal{S}_{maximin}(\langle \preceq, M, s\rangle) \text{ such that } s \rightarrow_A t.$$

This means that $t \in f(s, \preceq)$ iff there is an set of actions $A \subseteq Ac$ such that $A = \bigcup_{i \in Ag} A_i$ where for each A_i we have that $A_i \subseteq \mathsf{act}(i)$ and A_i is maximin optimal for agent i.

So, a multi-agent system maximin implementing a choice rule means that it selects for any pair of state and preference profile exactly those states that the system might move into if all the agents choose their actions such that they get the best possible worst case outcome that might result from performing an action. The following example will illustrate the maximin implementation problem.

Example 1 (Maximin Implementation). Assume that we have a set of agents $Ag = \{1, 2\}$ and a set of actions $Ac = \{a, b\}$. Suppose the ability of the agents to perform actions is as follows: $\mathsf{act}(1) = \{a\}, \mathsf{act}(2) = \{b\}$. Let $S = \{s, t, u\}$ be a set of states and $\mathcal{P} = \{\preceq, \preceq'\}$ be a set of preference profiles where

$$\begin{array}{l} s \prec_1 t \approx_1 u, \\ s \prec_2 t \prec_2 u, \\ t \approx'_1 u \prec'_1 s, \\ t \prec'_2 u \prec'_2 s. \end{array}$$

Now, assume that there is a planner who wants to implement the following choice rule $f : \mathcal{P} \times S \rightarrow 2^S$,

$$\begin{array}{rcl} f(\preceq, s) & = & \{t\}, \\ f(\preceq', s) & = & \{s, u\}, \\ f(\preceq, x) & = & \{x\} \text{ for } x \in \{t, u\}, \end{array}$$

in some multi-agent system that is based on $\bar{M} = \langle S, Ac, \Phi_0, \pi, Ag, \mathsf{act}\rangle$. So, the planner has to find an accessibility relation that extends the static multi-agent system to a multi-agent system that implements the choice rule.

This means that in every state the accessibility relation from that state is such that – assuming that the agents interact according to the maximin solution concept – for every preference profile in $\mathcal{P}$ the system will move to one of the states that the choice rule selects for that state and preference profile.

We will show that the following accessibility relation does the job.

$$\begin{array}{lll} s & \longrightarrow_{\emptyset} & t \\ s & \longrightarrow_{\{a,b\}} & s \\ s & \longrightarrow_{\{a\}} & u \\ s & \longrightarrow_{\{b\}} & u \\ x & \longrightarrow_{A} & x \text{ for any } x \in \{t, u\} \text{ and } A \subseteq Ac. \end{array}$$

Next, we show that the multi-agent system M that results from adding the above accessibility relation to $\bar{M}$ indeed maximin implements f. In each state, agent 1 has the choice between doing nothing and doing a. Agent 2 can always choose between doing nothing and doing b.

The following table shows some facts underlying the strategic reasoning of the agents in state s.

agent i	action	possible next states	worst $(\preceq_i)$	worst $(\preceq'_i)$
1	$\emptyset$	t, u	t, u	t, u
1	a	s, u	s	u
2	$\emptyset$	t, u	t, u	t
2	b	s, u	s	u

First, assume that the agents have preferences according to $\preceq$. If 1 does $\emptyset$, i.e. he does not do any action, then the system might go into state t or state u depending on what the other agent does. If 1 does action a, then the system can go into state s or state u depending on whether agent 2 does b or $\emptyset$ respectively. Then in state s, agent 1 would do $\emptyset$ because the worst state this can lead to is t which is strictly better for him than state s which is the worst state the system can go into after agent 1 doing a.

By a similar argument, $\emptyset$ is also the only maximin action for agent 2 in state s. The worst state resulting from agent 2 doing b is s. If he does nothing instead (i.e. $\emptyset$), the worst possible state is t which is strictly better for him than s. Thus, both agents will do $\emptyset$. So, assuming that the agents play according to the maximin solution concept, the concurrent action that will take place is $\emptyset$, which will move the system from state s to state t. This is in accordance with the choice rule which says that $f(\preceq, s) = \{t\}$. In the other states (i.e. t and u), every $A \subseteq \mathsf{act}(i)$ is a maximin action for agent i since all concurrent actions lead the same state.

Next, suppose that the agents have preferences according to $\preceq'$. Then for the states u and t, again any concurrent action is a maximin action. So, let us look at state s. Again, agent 1 has the choice between doing nothing and doing a. The worst states (according to $\preceq'_1$) that the system might move into after 1 doing $\emptyset$ are both t and u. The worst one after 1 doing a is u. Since 1 is indifferent between u and t, both $\emptyset$ and a are maximin actions for agent 1 because the agent is indifferent between their respective worst case outcomes.

Now, look at agent 2. He has the choice between doing nothing and doing b. Doing nothing will lead to t or u, where t is worse for him. Doing b can result in s or u. Here, u is the worst of both and it is still strictly preferred over t, the worst case outcome of agent 2 doing $\emptyset$. Thus, agent 2 will choose to do b.

Therefore, if the agents have preferences according to $\preceq'$ and always perform maximin maximin optimal actions, then the concurrent action that will take place in s can be any of $\{b\}$ or $\{a, b\}$. So, the system will either move to u or stay in s.

Thus, assuming the agents behave according to the maximin solution concept, if they have preferences according to $\preceq$, the system will move from state s to

state t and if they have preferences according to $\preceq'$ it will stay in s or move to u. This is in accordance with the choice rule f which says $f(\preceq, s) = \{t\}$ and $f(\preceq', s) = \{s, u\}$.
So, M indeed maximin implements f.

Having formally defined the maximin implementation problem and illustrated it by an example, we can now continue our investigations by looking at properties of maximin implementable choice rules.

Properties of Maximim-implementable Choice Rules

Now, we will give an overview of some properties that choice rules can have and examine maximin implementable choice rules with respect to those properties. We will not give a complete analysis of maximin implementable choice rules but rather focus on some interesting aspects.

The first property that we will consider is monotonicity. The idea of monotonicity is the following. Assume that the preferences of the agents changed in such a way that now one of the agent ranks one of the alternatives at a higher position than before and assume that this is the only change in the preferences that has occurred. Then it should not be the case that the alternative that is now more preferred by one agents was selected by the choice rule before the preference change but is not selected any more after the change.

In the following, we will use an alternative formulation of this notion of monotonicity which says that if an alternative t was selected by the choice rule and after a preference change of the agents it is not selected any more, then it has to be the case that there is one agent who now prefers some other alternative t' strictly over t whereas originally he preferred t over t'.

Definition 5.14 (Monotonicity). *Let Ag be a set of agents, S a set of states and let $\mathcal{P}$ be a set of preference profiles of the agents in Ag over S. A choice rule $f : \mathcal{P} \times S \to 2^S$ is monotonic if whenever $t \in f(\preceq, s)$ and $t \notin f(\preceq', s)$ then there is some agent $i \in Ag$ and some state $t' \in S$ such that $t \succeq_i t'$ and $t' \succ'_i t$.*

Now, the question is whether maximin implementable choice rules are monotonic. If a state is selected by a maximin implemented choice rule or not depends on whether it is accessible by a concurrent action that results from each agent performing an action that is maximin optimal for him. Since this only depends on which states each agent considers as the worst possible states resulting from him doing some action, the intuition is that maximin implementable choice rules are in general not monotonic. The following proposition shows that this intuition is indeed correct.

Proposition 5.15. *Not every maximin implementable choice rule is monotonic.*

Proof. We can use Example 1 as a counterexample. In short, the example was the following: Assume we are given the static multi-agent system $\bar{M} = \langle \{s,t\}, \{a,b\}, \Phi_0, \pi, \{1,2\}, \mathsf{act}\rangle$, with $\mathsf{act}(1) = \{a\}, \mathsf{act}(2) = \{b\}$
and a set of preference profiles $\mathcal{P} = \{\preceq_i, \preceq'_i\}$ defined as follows:

$$\begin{aligned} &s \prec_1 t \approx_1 u,\\ &s \prec_2 t \prec_2 u,\\ &t \approx'_1 u \prec'_1 s,\\ &t \prec'_2 u \prec'_2 s. \end{aligned}$$

Then the choice rule $f : \mathcal{P} \times S \to 2^S$

$$\begin{array}{rcl} f(\preceq, s) & = & \{t\}, \\ f(\preceq', s) & = & \{s, u\}, \\ f(\preceq, x) & = & \{x\} \text{ for } x \in \{t, u\}. \end{array}$$

is maximin implemented by a multi-agent system M based on $\bar{M}$ with the accessibility relation defined as follows.

$$\begin{array}{lll} s & \longrightarrow_{\emptyset} & t \\ s & \longrightarrow_{\{a,b\}} & s \\ s & \longrightarrow_{\{a\}} & u \\ s & \longrightarrow_{\{b\}} & u \\ x & \longrightarrow_{A} & x \text{ for any } x \in \{t, u\} \text{ and } A \subseteq Ac. \end{array}$$

We have already seen that the choice rule f defined in Example 1 is maximin implemented by the multi-agent system M. So, it remains to show that f is not monotonic.

f is not monotonic because we have that $u \in f(\preceq', s)$ and $u \notin f(\preceq, s)$ but there is no agent $i \in Ag$ such that for some $v \in S$ we have that $v \preceq'_i u$ and $u \prec_i v$.

Hence, f is maximin implementable but not monotonic. □

The next property that we will look at is being a dictatorship. This is a property that is usually not desired since a dictatorial choice rule gives all the power of selecting alternatives to one agent and the preferences of the other agents are completely ignored. A choice rule is dictatorial if there is one agent such that no matter what are the preferences of the agents, the choice rule only selects alternatives that this agent prefers over all the other alternatives. So, in the framework of multi-agent system, this then means that the choice rule only selects states that are preferred over all the other states by the dictator.

Definition 5.16 (Dictatorship). *Let S be a set of states, Ag a set of agents, and let $\mathcal{P}$ be a set of preference profiles of the agents in Ag over S. A choice rule $f : \mathcal{P} \times S \to 2^S$ is said to be a d-dictatorship if there is an agent $d \in Ag$ such that for every state s and for every preference profile $\preceq$ it holds that if $t \in f(\preceq, s)$, then t is top-ranked by d, i.e. $t \succeq_d t'$ for all $t' \in S$.*

Next, we will give a condition for a multi-agent system under which it is not possible to implement any non-dictatorial choice rule. The condition basically says that there is an agent that can perform a certain set of actions that nobody else can perform. If the number of those actions is sufficiently large[1], it is possible to set the accessibility relation in such a way that the transitions of the system are completely controlled by this agent.

Proposition 5.17. *Let $\bar{M} = \langle S, Ac, \Phi_0, \pi, Ag, \mathsf{act}\rangle$ be a static multi-agent system. If there is an agent $d \in Ag$ such that there is a set of actions $A_d \subseteq \mathsf{act}(d)$ where $2^{|A_d|} \geq |S|$ and $A_d \cap \mathsf{act}(i) = \emptyset$ for all agents $i \neq d$, then there is a multi-agent system M based on $\bar{M}$ such that every choice rule that is maximin implemented by M is a d-dictatorship.*

[1] "Sufficiently large" meaning here that there are at least as many different sets of such actions as there are states in the system.

Proof. Let $\bar{M} = \langle S, Ac, \Phi_0, \pi, Ag, \mathsf{act}\rangle$ be a static multi-agent system satisfying the assumed properties. Now, construct a multi-agent system based on $\bar{M}$ as follows:

For every state $s \in S$ we do the following: For each $t \in S$ take a different set $A_d^t \subseteq A_d$. Then for every set of actions $C \subseteq Ac$ such that $C \cap A_d = A_d^t$, we set $s \rightarrow_C t$.

Finding a different set A_d^t for each t is possible because $2^{|A_d|} \geq |S|$. In order to make sure that M is a proper multi-agent system, we also have to make sure that in every state every set of actions can be performed. This can be done as follows: For every $C \subseteq Ac$ such that for every $t \in S : C \cap A_d \neq A_d^t$, we set $s \rightarrow_C t'$ for some $t' \in S$.

Then the transitions of the systems are completely determined by the actions of d. For each state t, d has the power of making the system move there by performing A_d^t. Now, suppose that a choice rule f is maximin implemented by M. This means that for all states $s, t \in S$ and preference profiles $\preceq \in \mathcal{P}$ we have that $t \in f(\preceq, s)$ iff there is a concurrent action $A \subseteq Ac$ that consists of maximin optimal actions for all agents $i \in Ag$ and $s \rightarrow_A t$. Since the transition of the system only depends on the action of d, we have $s \rightarrow_A t$ for every A such that $A \cap A_d = A_d^t$.
Then a set of actions $A \subseteq Ac$ consists of maximin optimal actions for all agents only if $A \cap A_d = A_d^{\hat{t}}$, where $\hat{t} \succeq_d t$ for all $t \in S$. Therefore, for every $t \in f(\preceq, s)$ it holds that $t \succeq_d t'$ for all $t' \in S$. Hence, f is a d-dictatorship. □

The above proposition provides an explicit connection between an agent having the power to always achieve his most preferred outcome and him having the ability to perform actions that nobody else can perform. This illustrates one of the advantages of our framework since it gives insight into one of the possible origins of the power of a dictator, namely being the only agent that has control over some of the actions. In other words, we have an explicit connection between the way how the ability to perform certain actions is distributed among the agents and the distribution of power to determine which will be the next state of the system.

The next property that we will investigate is independence of irrelevant alternatives.

There are various ways of stating this property. The general idea is that if the choice rule only chooses alternative a from the set $\{a, b\}$, then it should not be the case that after a third alternative c is introduced, a is not chosen any more but b is. Alternatively, the property can also be formulated by saying that it is not the case that the social choice with respect to two alternatives is affected by agents changing their preferences over other alternatives.

Definition 5.18 (Independence of Irrelevant Alternatives)**.** *Let $\bar{M} = \langle S, Ac, \Phi_0, \pi, Ag, \mathsf{act}\rangle$ be a static multi-agent system. A choice rule $f : \mathcal{P} \times S \rightarrow 2^S$ is said to be independent of irrelevant alternatives if the following holds:*
If for all $i \in Ag, t, t' \in S$ it holds that $t \preceq_i t'$ iff $t \preceq'_i t'$, then

1. *it is not the case that $t \in f(\preceq, s), t' \notin f(\preceq, s), t \notin f(\preceq', s)$ and $t' \in f(\preceq', s)$ and*

2. it is also not the case that $t' \in f(\preceq, s), t \notin f(\preceq, s), t' \notin f(\preceq', s)$ and $t \in f(\preceq', s)$.

Similar to the case of monotonicity, our intuition is that it should in general not be the case that maximin implementable choice rules are independent of irrelevant alternatives. The reason is again that the maximin solution primalrily depends only on what agents consider to be the worst case and if we consider the preferences over two states where none of them is the worst case state, the choice between those two states depends only on the states that are reached in the worst case and not on the states under consideration themselves. Therefore, it seems to be the case that maximin implementable choice rules can be dependent on irrelevant alternatives. The following proposition shows that this is indeed the case.

Proposition 5.19. *Not every maximin implementable choice rule is independent of irrelevant alternatives.*

Proof. Again, we can use Example 1 as a counterexample. We will not state the example again but refer the reader to page 61.

We have already seen that the choice rule f defined in Example 1 is maximin implemented by the multi-agent system M. So, it remains to show that f is not independent of irrelevant alternatives. For all agents i, we have that

$$t \preceq_i u \quad \text{iff} \quad t \preceq'_i u,$$

but from the set $\{t, u\}$ only t is selected by the choice rule for the preference profile $\preceq$ and only u is selected for $\preceq'$. Thus, f is not independent of irrelevant alternatives. □

5.3.2 Nash Implementation

In this section, we will investigate the implementation problem for the solution concept of Nash equilibrium. Whereas in the case of the maximin solution concept, the individual agents do not make any assumptions about the other agents choices of actions but choose their action such that it will be best in the worst case no matter what the others do, in the Nash equilibrium solution concept agents do make assumptions about the others agents' choices. The general idea is that an action profile or a strategy profile is a Nash equilibrium if for every agent it holds that given the fact that all the other agents play according to this profile, he is not better off by deviating from the profile, i.e. the best thing for him to do is to also play according the strategy.

If we want to define the Nash solution concept formally, it requires us to have some way of talking about the actions of the other agents. In order to make this easier, we introduce the following notation.

Notation 5.20. *Suppose we have a set of agents Ag and a set of actions $A^* \subseteq Ac$ which is of the form $A^* = \bigcup_{i \in Ag} A^*_i$. Let $i \in Ag$ be an agent. Then we write A^*_{-i} for $\bigcup_{j \in Ag \setminus \{i\}} A^*_j$.*

Now, we will define a Nash equilibrium action in the framework of multi-agent systems with preferences. Given a state of the system, a concurrent action is a

Nash equilibrium if it is the union of actions A_i^* for each of the agents such that for every agent i it holds that assuming that all the other agents perform their actions A_j^*, doing A_i^* is best for i.

Definition 5.21 (Nash Equilibrium). *Given a multi-agent system with preferences $M^{\preceq} = \langle S, Ac, (\rightarrow)_{A \subseteq Ac}, \Phi_0, Ag, \mathsf{act}, \{\preceq_i\}_{i \in Ag}, \pi \rangle$, a set of actions $A^* = \bigcup_{i \in Ag} A_i^*$ with $A_i^* \subseteq \mathsf{act}(i)$ is called a Nash Equilibrium (or Nash optimal) in s if for every agent $i \in Ag$ and for every set of actions $A_i \subseteq \mathsf{act}(i)$ that agent i can perform it holds that $t \preceq_i t^*$ for states $t, t^* \in S$ such that $s \rightarrow_{A^*} t^*$ and $s \rightarrow_{A^*_{-i} \cup A_i} t$.*

Definition 5.22 (Nash Solution Concept). *Given a set of multi agents systems $\mathbb{M}$ that are all based on the same static multi-agent system $\bar{M} = \langle S, Ac, \Phi_0, \pi, Ag, \mathsf{act} \rangle$, the Nash solution concept is defined as a function $\mathcal{S}_{Nash} : \mathcal{P} \times \mathbb{M} \times S \rightarrow 2^{2^{Ac}}$, $\mathcal{S}_{Nash}(\langle \preceq, M, s \rangle) \mapsto \mathcal{A}$, where*

$$\mathcal{A} := \{A^* \subseteq Ac | A^* \textit{ is a Nash Equilibrium action in the state } s \textit{ in } M^{\preceq}\}.$$

Now, a multi-agent system Nash implementing a choice rule means that for any state and preference profile some state is selected if and only if it is accessible by an Nash equilibrium action.

Definition 5.23 (Nash Implementability). *We say that a multi-agent system $M = \langle S, Ac, (\rightarrow)_{A \subseteq Ac}, \Phi_0, \pi, Ag, \mathsf{act} \rangle$ Nash implements a choice rule $f : \mathcal{P} \times S \rightarrow 2^S$ iff for all states $s, t \in S$ and preference profiles $\preceq \in \mathcal{P}$ it holds that*

$$t \in f(s, \preceq) \textit{ iff } \exists A \in \mathcal{S}_{Nash}(\langle \preceq, M, s \rangle) \textit{ such that } s \rightarrow_A t.$$

Properties of Nash Implementable Choice Rules

Comparing the Nash solution concept to that of maximin, the Nash solution concept seems stronger in the sense that we should be able to show that Nash implementable choice rules have some of the properties that maximin implementable choice rules do not have, as we have seen in Section 5.3.1.

First we will investigate Nash implementable choice rules by focusing on the property of having veto power. If a choice rule has veto power, this means that there is a player (a so called veto player) that has the power of stopping the group consisting of all the other agents from achieving what they want. It is quite reasonable to not want that an agent has such power. This means that if all agents but one agree on what they want the most, then this outcome is indeed achieved. In our framework, this means that if there is a state that is most preferred by $|Ag| - 1$ agents, then this state is indeed selected by the choice rule.

Definition 5.24 (No Veto Power). *If for a choice rule $f : \mathcal{P} \times S \rightarrow 2^S$ it holds that whenever $t \in S$ is ranked at the top of the preference orderings $\preceq$ of at least $|Ag| - 1$ agents, we have that $t \in f(\preceq, s)$ for all $s \in S$, then we say that f has no veto power.*

Note that this definition of no veto power is quite strong in the sense that it also implies that if a choice rule is implemented and has no veto-power then a state that is ranked a topmost position by all agents but one (or all agents) has to be accessible from everywhere in the system.

Now, we will try to give an explicit connection between a choice rule having no veto power and the effects of concurrent actions performed by groups consisting of at least $|Ag| - 1$ agents. The general idea is that if the effects of concurrent actions are completely determined by what $|Ag| - 1$ agents are doing, then the implemented choice rule has no veto power because the actions of a single agent do not have any influence on where the system is going.

Proposition 5.25. *Let f be a choice rule that is Nash implementable by a multi-agent system $M = \langle S, Ac, (\rightarrow)_{A \subseteq Ac}, \Phi_0, \pi, Ag, \mathsf{act}\rangle$ satisfying the following properties*

- *For any $G \subseteq Ag$ with $|G| \geq |Ag| - 1$ the following holds: For every set of actions $A_G \subseteq \mathsf{act}(G)$ and for all states $t, t' \in S$ we have that if $s \rightarrow_C t$ and $s \rightarrow_{C'} t'$ for $C, C' \subseteq \mathsf{act}(Ag)$ such that $C \cap \mathsf{act}(G) = C' \cap \mathsf{act}(G) = A_G$, then $t = t'$.*
- *$\bigcup_{C \subseteq Ac} \rightarrow_C = S \times S$.*
- *$\mathsf{act}(i) \cap \mathsf{act}(j) = \emptyset$ for all agents i, j such that $i \neq j$.*

Then f has no veto power.

Proof. First consider the case where $G = Ag$. Assume that $\hat{t}$ is at the topmost position of the preference orderings of all the agents. By the second assumption, for any state s there is some action $\hat{A} \subseteq Ac$ such that $s \rightarrow_{\hat{A}} \hat{t}$. Then any such $\hat{A}$ is a Nash equilibrium action in s. Thus, it is selected by the choice rule.

Next, let $G = Ag \setminus \{k\}$ for some $k \in Ag$ and let $\hat{t} \in S$ such that for every agent $i \in G$ and for every state $t \in S$ we have that $\hat{t} \succeq_i t$. Then take some $s \in S$. By the second property that we assumed, there must be a set of actions $\hat{C} \subseteq Ac$ such that $s \rightarrow_{\hat{C}} \hat{t}$.

Then $\hat{C}$ is of the form $\bigcup_{i \in Ag} \hat{A}_i$, where $\hat{A}_i \subseteq \mathsf{act}(i)$. Now define $\hat{A}_G := \bigcup_{i \in G} \hat{A}_i$.

Next, we will show that $\hat{C}$ is in fact a Nash optimal set of actions. First, consider the agent $k \notin G$. Assume that G performs $\hat{A}_G$. Then by the first property of M, the system will move to $\hat{t}$ no matter what k does. So, k does not have any influence on where the system moves next. So, doing $\hat{A}_k$ is at least as good for k as any other action. Now, consider $i \in G$. Given that all the other agents $j \in Ag \setminus \{i\}$ do actions $\hat{A}_j$, it is best for i to do $\hat{A}_i$ because that means that the system will move to state $\hat{t}$ which is ranked at topmost position by i. So, $\hat{A}$ is Nash optimal. This then implies that $\hat{t} \in f(\preceq, s)$. Hence, f has no veto power. □

As we already mentioned, the condition that $\bigcup_{C \subseteq Ac} \rightarrow_C = S \times S$ is quite strong; it says that no matter which is the current state, every state can be reached within one transition of the system. We will now define a local version of no veto power where in each state only the accessible states are considered. This means that if among the accessible states there is one that is the most preferred state of at least $|Ag| - 1$ agents, then the system will move there.

So, this local version of no veto power can also be satisfied by choice rules implemented in systems where not every state is accessible from everywhere.

We will introduce the following notation for talking about the set of states directly accessible form the current state.

Notation 5.26. *Let $M = \mathbb{M}$ be a multi agent system and let $s \in S$. Then we write $T_s := \{t | \exists A \subseteq Ac : s \to_C t\}$ for the set of states accessible from s. We use the same notation in multi-agent systems with preferences.*

Now, we can define a local version of no veto power.

Definition 5.27 (No Veto Power in a Multi-agent System)**.** *Let $M = \langle S, Ac, (\to)_{A \subseteq Ac}, \Phi_0, \pi, Ag, \mathsf{act}\rangle$ be a multi-agent system that implements a choice rule $f : \mathcal{P} \times S \to 2^S$. If for any state s and preference profile $\preceq \in \mathcal{P}$ we have the following: For any $t \in T_s$ that at least $|Ag| - 1$ agents prefer over all the other states in T_s, it holds that $t \in f(\preceq, s)$,*

then we say that f has no veto power in M.

Note that it is not the case that no veto power implies no veto power in a multi-agent system or the other way around:

- It can be the case that in a multi-agent system that implements f, there is no state that is ranked at topmost position by at least $|Ag| - 1$ agents according to any of the preference profiles under consideration. Then no veto power is of course satisfied. Now it can still be the case that there is an accessible state that is ranked at topmost position by at least $|Ag| - 1$ agents and that is not selected by the choice rule. Then we have that the choice rule has no veto power but does indeed have veto power in the multi-agent system that implements it. Then no veto power is satisfied bot no veto power in the multi-agent system that implements the choice rule is not.

- For the converse, consider the case where some state that is ranked at topmost position by at least $|Ag| - 1$ agents is not accessible from the current state. Then the implemented choice rule cannot have no veto power. But on the other hand it can still have no veto power in the multi-agent system that implements it.

Now, we can prove a statement similar to that of the previous proposition where we do not need the condition that $\bigcup_{C \subseteq Ac} \to_C = S \times S$. The proof is essentially analogous to the previous one.

Proposition 5.28. *Let f be a choice rule that is Nash implemented by a multi-agent system $M = \langle S, Ac, (\to)_{A \subseteq Ac}, \Phi_0, \pi, Ag, \mathsf{act}\rangle$ satisfying the following properties*

- *For any $G \subseteq Ag$ with $|G| \geq |Ag| - 1$ the following holds: For every set of actions $A_G \subseteq \mathsf{act}(G)$ and for all states $t, t' \in S$ we have that if $s \to_C t$ and $s \to_{C'} t'$ for $C, C' \subseteq \mathsf{act}(Ag)$ such that $C \cap \mathsf{act}(G) = C' \cap \mathsf{act}(G) = A_G$, then $t = t'$.*

- $\mathsf{act}(i) \cap \mathsf{act}(j) = \emptyset$ *for all agents i, j such that $i \neq j$.*

Then f has no veto power in M.

Proof. Let $G \subseteq Ag$ such that $|G| \geq |Ag| - 1$ and let $\hat{t} \in T_s$ such that for every agent $i \in G$ and for every state $t \in T_s$ we have that $\hat{t} \succeq_i t$. Let $\hat{C}$ be such that $s \rightarrow_{\hat{C}} \hat{t}$ (since $\hat{t}$ is accessible form s, such an action exists).

Then $\hat{C}$ is of the form $\bigcup_{i \in Ag} \hat{A}_i$, where $\hat{A}_i \subseteq \mathsf{act}(i)$. Now define $\hat{A}_G := \bigcup_{i \in G} \hat{A}_i$.

Next, we will show that $\hat{C}$ is in fact a Nash optimal set of actions. Suppose that $G = Ag$. Then $\hat{C}$ is a Nash optimal action for all the agents since for agents it leads to the most preferred of the accessible states.

Now, assume there is an agent $k \notin G$. Assume that G performs $\hat{A}_G$. Then by the first property of M, the system will move to $\hat{t}$ no matter what k does. So, k does not have any influence on where the system moves next. So, doing $\hat{A}_k$ is at least as good for k as any other action. Now, consider $i \in G$. Given that all the other agents $j \in Ag \setminus \{i\}$ do actions $\hat{A}_j$, it is best for i to do $\hat{A}_i$ because that means that the system will move to state $\hat{t}$ which is preferred over all the other accessible states. So, $\hat{A}$ is Nash optimal. This then implies that $\hat{t} \in f(\preceq, s)$. Hence, f has no veto power in M. □

Coming back to the maximin solution concept, it is quite easy to see that maximin implementable choice rules do not always satisfy no veto power. Even if there is a state that is the most preferred one of all the agents, it does not need to be selected by the choice rule since only the worst case outcomes are relevant for selecting the maximin optimal actions.

Let us now come back to the property of monotonicity, which we introduced in the previous section where we saw that maximin implementable choice rules are in general not monotonic. Recall that monotonic choice rules have the property that if a preference change of the agents leads to an alternative a not being selected any more whereas it has been before, then there must be an agent that changed his preferences by now ranking some alternative b strictly over a that was below it before.

In mechanism design in game theory, it is a straightforward result that Nash implementable choice rules are monotonic [Osborne and Rubinstein, 1994]. Assume that we have a Nash implementable choice rule and assume that the agents change their preferences and one state t that was selected by the choice rule before is not selected any more. This means that the concurrent action leading from the current state to t is not a Nash equilibrium action any more after the preference change but was one before. So, there is an agent that – assuming that all the other agents stick to their components of the action leading to t – can perform some other action such that the resulting concurrent action leads to some state t' which is strictly better for him than t. Before the preference change, t must have been at least as good as t' for the agent because otherwise already then the agent would have deviated and tried to move the system there.

Proposition 5.29. *Let $M = \langle S, Ac, (\rightarrow)_{A \subseteq Ac}, \Phi_0, \pi, Ag, \mathsf{act} \rangle$ be a multi-agent system. If M Nash implements a choice rule f, then f is monotonic.*

Proof. Let f be a choice rule that is Nash implemented by M, let $s, t^* \in S$ and $\preceq, \preceq' \in \mathcal{P}$ such that $t^* \in f(\preceq, s)$ and $t^* \notin f(\preceq', s)$.

Then since f is Nash implemented by M, there is a set of actions $A^* \subseteq Ac$ such that A^* is a Nash equilibrium action and $s \to_{A^*} t^*$. So, for all agents $i \in Ag$ we have that for all sets of actions $A_i \subseteq \mathsf{act}(i)$ it holds that for any state $t \in S$: If $s \to_{A^*_{-i} \cup A_i} t$ then $t \preceq_i t^*$.

$t^* \notin f(\preceq', s)$ implies that there is no set of actions A that is a Nash equilibrium according to $\preceq'$. Since we know that $s \to_{A^*} t^*$, there must be an agent $i \in Ag$ such that there is a set of actions $A_i \subseteq \mathsf{act}(i)$ such that $t^* \prec'_i t'$ for some $t' \in S$ such that $s \to_{A^*_{-i} \cup A_i} t'$. So, there is an agent $i \in Ag$ such that $t' \preceq_i t^*$ and $t^* \prec'_i t'$. Hence, f is monotonic. □

The converse does not hold; there are monotonic choice rules that are not Nash implementable. One example of such choice rules are choice rules that select alternatives that are the most preferred ones for some agents and the least preferred ones for other agents.

Proposition 5.30. *Not every monotonic choice rule is Nash implementable.*

Proof. Let $Ag = \{1, 2\}$ be a set of agents and $S = \{s, t, u\}$ a set of states. Let $\mathcal{P} = \{\preceq, \preceq'\}$ be a set of preference profiles over S, where $t \prec_1 u \prec_1 s, s \prec_2 t \prec_2 u$ and $t \prec'_1 s \prec'_1 u, s \prec'_2 u \prec'_2 t$.

Consider the choice rule f defined as follows. $f(\preceq, s) = \{s\}, f(\preceq', s) = \{t\}$ and for $x \in \{t, u\}$ we have that $f(\preceq, x) = f(\preceq', x) = \{x\}$. Then f is monotonic because we have for $s \in f(\preceq, s), s \notin f(\preceq', s)$: $u \prec_1 s$, $s \prec'_1 u$. Furthermore, for $t \in f(\preceq', s), t \notin f(\preceq, s)$: $u \prec'_2 t$ and $t \prec_2 u$.

Suppose that f is Nash implementable by a multi-agent system M where Ag is the set of agents and S the set of states. Then since $f(\preceq, s) = \{s\}$, there is a set of actions $A^* = A^*_1 \cup A^*_2$, where $A^*_1 \subseteq \mathsf{act}(1), A^*_2 \subseteq \mathsf{act}(2)$ such that $s \to_{A^*} s$ and given that agent 1 does A^*_1, doing A^*_2 is best for agent 2. Since according to $\preceq$, s is strictly worst for agent 2, it must be the case that for all sets of actions $A_2 \subseteq \mathsf{act}(2)$ it holds that $s \to_{A^*_1 \cup A_2} s$.

Analogously, since $f(\preceq', s) = \{t\}$, there is a set of actions $A^{*\prime} = A^{*\prime}_1 \cup A^{*\prime}_2$, where $A^{*\prime}_1 \subseteq \mathsf{act}(1), A^{*\prime}_2 \subseteq \mathsf{act}(2)$ such that $s \to_{A^{*\prime}} t$ and given that agent 2 does $A^{*\prime}_2$, doing $A^{*\prime}_1$ is best for agent 1. But we already know that $s \to_{A^{*\prime}_2 \cup A^*_1} s$ and $t \prec'_1 s$, which leads to a contradiction. Hence, f is not Nash implementable by M. □

5.3.3 Dominant Strategy Equilibrium (DSE) Implementation

The next solution concept we will investigate is a that of dominant strategies. This is a strong solution concept; a dominant strategy gives the action that is the best response to *every* action of the other agents. So, if an agent has a dominant strategy, then he would probably play according to it because he will get the best outcome no matter what the other agents do. In our framework, in a state of a multi-agent system with preferrences a DSE is given by a concurrent action that results from every agent doing a set of actions that – no matter what the other agents do – will lead to a better state for him than any other set of actions that he could perform.

Definition 5.31 (Dominant Strategy Equilibrium (DSE)). *Given a multi-agent system with preferences $M^{\preceq} = \langle S, Ac, (\rightarrow)_{A\subseteq Ac}, \Phi_0, Ag, \mathsf{act}, \{\preceq_i\}_{i\in Ag}, \pi\rangle$, a DSE action in state $s \in S$ is a set of actions $A^* = \bigcup_{i\in Ag} A_i^*$ with $A_i^* \subseteq \mathsf{act}(i)$ such that for every set of actions $A \subseteq Ac$, $A = \bigcup_{i\in Ag} A_i$ with $A_i \subseteq \mathsf{act}(i)$ and for every agent $i \in Ag$, it holds that for all states t, t^*: If $s \rightarrow_A t$ and $s \rightarrow_{A_{-i}\cup A_i^*} t^*$, then $t^* \succeq_i t$.*

Using the notion of a DSE action, we can now define the DSE Solution concept.

Definition 5.32 (DSE Solution Concept). *Given a set of multi agents systems $\mathbb{M}$ that are all based on the same static multi-agent system $\bar{M} = \langle S, Ac, \Phi_0, \pi, Ag, \mathsf{act}\rangle$, the DSE solution concept is defined as a function $\mathcal{S}_{DSE} : \mathcal{P} \times \mathbb{M} \times S \rightarrow 2^{2^{Ac}}$, $\mathcal{S}_{DSE}(\langle\preceq, M, s\rangle) \mapsto \mathcal{A}$, where*

$$\mathcal{A} := \{A^* | A^* \text{ is a DSE action in } s \text{ in } M^{\preceq}\}$$

Then a choice rule is DSE implementable if and only if for every state and every preference profile, the choice rule selects exactly those states that are accessible by some concurrent action that is a DSE action.

Definition 5.33 (DSE-implementability). *We say that a multi-agent system $M = \langle S, Ac, (\rightarrow)_{A\subseteq Ac}, \Phi_0, \pi, Ag, \mathsf{act}\rangle$ DSE-implements a choice rule $f : \mathcal{P}\times S \rightarrow 2^S$ if for all states $s, t \in S$ and for all preference profiles $\preceq \in \mathcal{P}$ it holds that*

$$t \in f(s, \preceq) \text{ iff } \exists A \in \mathcal{S}_{DSE}(\langle\preceq, M, s\rangle) \text{ such that } s \rightarrow_A t.$$

One area where many ideas of mechanism design are relevant is of voting theory. Voting theory is concerned with designing voting systems that get individual preferences (i.e. the preferences of the voters over the candidates) as input and generate a social preference, i.e. a preference order over the candidates or a social choice, i.e. one or several candidates that are selected. Of course, there are many different ways how such a system can be implemented and what is the actual procedure of choosing a candidate or a set of candidates. Of course, in most cases a voting system should not select candidates at random but the procedure should satisfy certain properties such that the actual choice of candidates respects and reflects the preferences of the voters as much as possible. Moreover, the procedure should be designed in such a way that voters do not have an incentive to misrepresent their preferences and will vote according to their true preferences over the candidates.

A central result in mechanism design and voting theory is the Gibbard-Satterthwaite Theorem. It says the following.

Assume that there are at least three alternatives and we consider all possible preference profiles of the agents over the alternatives. Then take some choice rule such that for every alternative there is a preference profile such that this alternative is the only one selected by the choice rule. Then if this choice rule is DSE implementable, it has to be dictatorial.

Proposition 5.34 (Gibbard-Satterthwaite Theorem [Osborne and Rubinstein, 1994]). *Let $\langle N, C, \mathcal{P}, \mathcal{G}\rangle$ be an environment (in the game theoretic sense as presented on page 56) in which C contains at least three members, $\mathcal{P}$ is the set of all possible preference profiles, and $\mathcal{G}$ is the set of strategic game forms with*

consequences in C. Let $f : \mathcal{P} \to 2^C$ be a choice rule that is DSE implementable and satisfies the condition that for every $c \in C$ there is a preference profile $\preceq \in \mathcal{P}$ such that $f(\preceq) = \{c\}$.
Then f is dictatorial.

The Gibbard Satterthwaite Theorem therefore shows some of our limitations for designing voting mechanisms, since the property that for every alternative there is a preference profile such that only that alternative is selected and also the property that all possible preference profiles considered and that the choice rule is non-dictatorial are indeed properties that we really might want a voting system to satisfy. The Gibbard-Satterthwaite Theorem shows that this is not possible.

We will not present the proof of the theorem in this thesis. It has been studied in great detail elsewhere, e.g. [Osborne and Rubinstein, 1994].
Instead, we will use our framework of multi-agent systems with preferences to look at issues similar to the ones dealt with in the Gibbard-Satterthwaite Theorem. In particular, we will use a certain class of multi-agent systems, namely those systems where all agents can perform the same actions. Another property of our multi-agent systems that we should keep in mind at this point is the following: Due to our modular approach, the effects of the performance of actions are independent of which agent actually performs them.

It is partly due to this property that we can show that if every agent can perform the same actions, then every DSE implementable choice rule is non-dictatorial.

The result then basically says that if every agent can perform the same actions and the effects of actions are independent of who performs them, then there cannot be a dictator. So, this result again describes the connection between the distribution of the action abilities among the agents and the agents' power of forcing the system to move to their most preferred states. If all the agents have the same ability to perform actions and an action has the same effect no matter who performs it, then every agent seems to also have the same power with respect to making the system move to certain states and therefore there cannot be a distinguished agent that has more power than the others.

Proposition 5.35. *Let $M = \langle S, Ac, (\to)_{A \subseteq Ac}, \Phi_0, \pi, Ag, \mathsf{act}\rangle$ be a multi-agent system with the properties that $|Ag| \geq 2, |S| \geq 2$ and for every agent $i \in Ag$ we have that $\mathsf{act}(i) = Ac$. Let $\mathcal{P}$ be the set of all preference profiles over S.*

Every choice rule $f : \mathcal{P} \times S \to 2^S$ that is DSE implemented by M is non-dictatorial.

Proof. Let f be DSE implemented by M and suppose it is a d-dictatorship for some $d \in Ag$.

Let $\hat{t} \in S$ and choose $\preceq \in \mathcal{P}$ such that $\hat{t} \succ_d t$ for all $t \in S, \hat{t} \neq t$ and there is an agent $k \in Ag$ such that there is a state $t' \in S$ such that $t' \succ_k \hat{t}$. $\mathcal{P}$ contains such a preference profile $\preceq$ because by assumption it contains all possible preference profiles. So according to $\preceq$, $\hat{t}$ is strictly preferred by d over all the other states in S and agent k strictly prefers t' over $\hat{t}$.

Then, since f is a d-dictatorship, it must be the case that $f(\preceq, s) = \{\hat{t}\}$. Then there is an $\hat{A} \subseteq Ac$ of the form $\hat{A} = \bigcup_{i \in Ag} \hat{A}_i$ where doing $\hat{A}_i$ is a dominant strategy for each agent i and $s \rightarrow_{\hat{A}} \hat{t}$.

Next, take $\preceq' \in \mathcal{P}$ such that $\preceq'_i = \preceq_i$ for $i \in Ag \setminus \{k, d\}$ and for agents k and d: $\preceq'_d = \preceq_k$ and $\preceq'_k = \preceq_d$. Since $\mathcal{P}$ contains all preference profiles over S, this is possible.

Then for every agent $i \in Ag \setminus \{d, k\}$, $\hat{A}_i$ is still a dominant strategy. And if $\hat{A}_d$ was a DSE action for d according to $\preceq$, then it must also be a DSE action for k according to $\preceq'$. Analogously, $\hat{A}_k$ is a DSE action for d according to $\preceq'$. This implies that $\hat{A}$ is also a DSE action according to $\preceq'$ and since $s \rightarrow_{\hat{A}} \hat{t}$, it must be the case that $\hat{t} \in f(\preceq', s)$, which is a contradiction because according to $\preceq'$, $\hat{t}$ is not top ranked by d. Hence, f is non-dictatorial. □

It also seems to be possible to generalize the above result to all other solution concepts that are invariant under permutations of preference orderings of agents that can perform the same actions, i.e. if two agents i, j with $\mathsf{act}(i) = \mathsf{act}(j)$ change their preferences such that agent i gets those of agent j and vice versa and all the other agents keep their preferences unchanged, then the solution concept still selects the same set of actions. This property is also referred to as *anonymity.*

Similar to the case of no veto power, the definition of a choice rule being dictatorial that we used so far is quite strong in the sense that if we have a dictatorial choice rule that is implemented in a multi-agent system then this also implies that from ever state of the system every state has to be accessible. This follows from the fact that every state can in principle be the most preferred one by the dictator and therefore the system would have to be able to move there in the next transition. So, according to our definition an implemented choice rule would also be called non-dictatorial even if at every state all the selected states are states that the dictator prefers over all the other accessible states, i.e. over all the alternatives that are relevant at the current state.

Motivated by these considerations, we define another version of being non-dictatorial which says that there is no agent such that in every state the choice rule only selects states that are preferred by this agent over all the other states accessible from the current one.

Definition 5.36 (Non-dictatorial in a Multi-agent System). *Let Ag be a set of agents, S a set of states and $\mathcal{P}$ a set of preference profiles of the agents. Let $f : \mathcal{P} \times S \rightarrow 2^S$ be a choice rule. If M is a multi-agent system that implements f and there is an agent $d \in Ag$ such that for all $s \in S, \preceq \in \mathcal{P}$ we have that for every $t' \in f(\preceq, s) : t' \succeq_d t$ for all $t \in T_s$, then f is said to be a d-dictatorship in M. If there is no such agent d, then we say that f is non-dictatorial in M.*

Being non-dictatorial in a multi-agent system is stronger than being non-dictatorial because it says that there is no agent such that in every state all the states selected by the choice rule are preferred by him over all the other accessible states as opposed to being preferred over all the states in S. Suppose that a choice rule f is implemented by a multi-agent system M. Then f is already non-dictatorial if for every agent i and preference profile $\preceq$ there is a state in the system such that none of the states most preferred according to $\preceq_i$ is accessible from there. But then f can still be dictatorial in M.

Now, we can prove a statement that is similar to that of Proposition 5.35. It says that when considering all possible preference profiles, every choice rule DSE implemented in a multi-agent system M in which each agent can perform the same actions and every state has at least two successors has to be non-dictatorial in M.

Proposition 5.37. *Let $M = \langle S, Ac, (\rightarrow)_{A \subseteq Ac}, \Phi_0, \pi, Ag, \mathsf{act}\rangle$ be a multi-agent system with the properties that $|Ag| \geq 2$, $|T_s| \geq 2$ for all $s \in S$ and for every agent $i \in Ag$ we have that $\mathsf{act}(i) = Ac$. Let $\mathcal{P}$ be the set of all preference profiles over S.*

Every choice rule $f : \mathcal{P} \times S \rightarrow 2^S$ that is DSE implemented by M is non-dictatorial in M.

Proof. Let f be DSE implemented by M and suppose it is a d-dictatorship in M for some $d \in Ag$.

Let $s \in S$ and $\preceq \in \mathcal{P}$ such that there is a $\hat{t} \in T_s$ that agent d strictly prefers over all the other states in T_s and there is an agent k such that $\hat{t} \prec_k t'$ for some $t' \in T_s$. This is possible since $\mathcal{P}$ contains all preference profiles over S and by assumption $|T_s| \geq 2$.
Then, since f is a d-dictatorship in M, it must be the case that $f(\preceq, s) = \{\hat{t}\}$. Then we have that $s \rightarrow_{\hat{A}} \hat{t}$ for a set of actions $\hat{A}$ such that $\hat{A} = \bigcup_{i \in Ag} \hat{A}_i$, where $A_i \subseteq \mathsf{act}(i)$ and doing A_i is a dominant strategy in s for each agent i.

Next, take $\preceq' \in \mathcal{P}$ such that $\preceq'_i = \preceq_i$ for $i \in Ag \setminus \{k, d\}$ and for agents k and d: $\preceq'_d = \preceq_k$ and $\preceq'_k = \preceq_d$. Since $\mathcal{P}$ contains all preference profiles over S, such a $\preceq'$ can be chosen.

Then for every agent $i \in Ag \setminus \{d, k\}$, $\hat{A}_i$ is still a dominant strategy. And if $\hat{A}_d$ was a DSE action for d according to $\preceq$, then it must also be a DSE action for k according to $\preceq'$. Analogously, $\hat{A}_k$ is a DSE action for d according to $\preceq'$. This implies that $\hat{A}$ is also a DSE action according to $\preceq'$ and since $s \rightarrow_{\hat{A}} \hat{t}$, it must be the case that $\hat{t} \in f(\preceq', s)$, which is a contradiction because according to $\preceq'$, $\hat{t}$ is not top ranked by d. Hence, f is non-dictatorial in M. □

Let us now come back to the Gibbart-Satterthwaite Theorem. It can be translated into the framework of implementation in multi-agent systems in at least two different ways. One version allows us to conclude that under certain conditions a DSE implementable choice rule is dictatorial and the other one that a DSE-implementable choice rule is dictatorial in the multi-agent system that implements it.

Proposition 5.38. *Let $M = \langle S, Ac, (\rightarrow)_{A \subseteq Ac}, \Phi_0, \pi, Ag, \mathsf{act}\rangle$ be a multi-agent system with $|T_s| \geq 3$ for all $s \in S$ and let $\mathcal{P}$ be the set of all preference profiles of the agents in Ag over S. If a choice rule $f : \mathcal{P} \times S \rightarrow 2^S$ is implemented by M and*

$$\forall s \in S, t \in T_s : \exists \preceq \in \mathcal{P} : f(\preceq, s) = t,$$

then f is dictatorial in M.

We also have an analogous result for a choice rule being dictatorial.

Proposition 5.39. *Let $M = \langle S, Ac, (\rightarrow)_{A \subseteq Ac}, \Phi_0, \pi, Ag, \mathsf{act}\rangle$ be a multi-agent system with $|S| \geq 3$ and let $\mathcal{P}$ be the set of all preference profiles of the agents in Ag over S. If a choice rule $f : \mathcal{P} \times S \rightarrow 2^S$ is implemented by M and*

$$\forall s, t \in S : \exists \preceq \in \mathcal{P} : f(\preceq, s) = t,$$

then f is dictatorial.

The proofs of both propositions are analogous to the one of the Gibbart-Satterthwaite Theorem and will not be presented here.
Let us now look at what are the consequences of propositions 5.35,5.37,5.38 and 5.39.

Suppose we have a choice rule f that is DSE implemented by a multi-agent system such that both the conditions of propositions 5.37 and 5.38 are satisfied. Then, we get a contradiction because we conclude that f is both dictatorial and non-dictatorial in M. So, we can conclude that there is no such f.

Analogously, when considering propositions 5.35 and 5.39 we can see that there is no choice rule that is DSE implemented in a multi-agent system such that the conditions of both propositions are satisfied. Formally, we get the following two corollaries that basically say that if every agent can perform the same actions, then no choice rule of a certain type is DSE implementable.

Corollary 5.40. *Let $M = \langle S, Ac, (\rightarrow)_{A \subseteq Ac}, \Phi_0, \pi, Ag, \mathsf{act}\rangle$ be a multi-agent system with the properties that $|S| \geq 3, |Ag| \geq 2$ and for every agent $i \in Ag$ we have that $\mathsf{act}(i) = Ac$. Let $\mathcal{P}$ be the set of all preference profiles over S and let $f : \mathcal{P} \times S \rightarrow 2^S$ be a choice rule such that for every $t \in S$ there is a preference profile $\preceq \in \mathcal{P}$ such that $f(\preceq) = \{t\}$. Then f is not DSE implemented by M.*

Corollary 5.41. *Let $M = \langle S, Ac, (\rightarrow)_{A \subseteq Ac}, \Phi_0, \pi, Ag, \mathsf{act}\rangle$ be a multi-agent system with the properties that for all $s \in S$: $|T_s| \geq 3, |Ag| \geq 2$ and for every agent $i \in Ag$ we have that $\mathsf{act}(i) = Ac$. Let $\mathcal{P}$ be the set of all preference profiles over S and let $f : \mathcal{P} \times S \rightarrow 2^S$ be a choice rule such that for every $t \in T_s$ there is a preference profile $\preceq \in \mathcal{P}$ such that $f(\preceq) = \{t\}$. Then f is not DSE implemented by M.*

Let us now briefly summarize this chapter. We showed that the semantic structures of the cooperation logic with actions and preferences can be used for investigations in the style of mechanism design. We obtained several results that make explicit how the abilities of the agents to perform certain actions affect the properties of choice rules implementable in a multi-agent system.

Chapter 6

Cooperative Games and the Cooperation Logic with Actions and Preferences

The games that are investigated by game theory can be divided into two groups: non-cooperative games, where the primitives are the actions of individual agents, and cooperative or coalitional games, where agents act together as a group. In Chapter 5, we already showed that we can use our framework of multi-agent systems with preferences to investigate issues about non-cooperative games, namely mechanism design. In this chapter, we will examine how the logic developed in Chapter 4 can be used to investigate issues related to coalitional games without transferable payoff (Definition 2.7).

Remark 6.1. *Note that as in the previous chapter, we are still assuming that the multi-agent systems (with preferences) under consideration are deterministic and that the agents‘ preference orderings are complete.*

Recall that the general idea of a coalitional game without transferable payoffs $\langle Ag, \Omega, \mathcal{V}, \preceq \rangle$ is the following: We have a set of players Ag and a set of consequences that can be seen as possible outcomes of the game. Each player i has a preference ordering $\preceq_i$ over the set of consequences. Moreover, we know the cooperative ability of groups of agents, i.e. for any coalition G we know which outcomes $\mathcal{V}(G)$ this coalition can achieve. In multi-agent systems as we defined them, the states the system can move into as a result of the interaction of the agents can be seen as outcomes of the interactive process. On a local level, when considering one state then the states that are accessible from that state can be seen as the possible outcomes of the interaction that takes place in the current state.

In a coalitional game, the outcome function $\mathcal{V}$ assigns to every coalition the set of outcomes this coalition can achieve. Transferring this idea to multi-agent systems can be done in different ways. First of all, note that in multi-agent systems as we defined them, the ability of agents to force the system to move into certain states can be seen as a local property because it depends on the current state of the system since this determines which are the accessible states.

So, in a way each state can be seen as having a corresponding game attached to it whose outcome then determines where the system moves next. This idea is similar to that of the semantic structures of CL [Pauly, 2002] which we presented in Section 2.1.2.

Consider the following two possible ways of how to define an analog of the function $\mathcal{V}$ in multi-agent systems.

- For every state s and coalition G, assign the set of formulas φ such that in state s it holds that $\langle\langle G\rangle\rangle\varphi$.
- For every state s and coalition G, assign the set of states that group G can force the system to move into.

If we continue using the cooperation logic with actions and preferences defined in Chapter 4, where the preferences of the agents are represented as a preference relation over the set of states, then the second possibility seems to be more appropriate since then it is easy to talk about the preferences of the agents in a group over the possible states that the group can force the system to move into.

If we combine the cooperation logic with actions CLA [Sauro et al., 2006] that we presented in Chapter 3 with a preference logic that represents the agents' preferences as a preference relation over formulas [van Benthem et al., 2005; van Benthem et al., 2007], then the first possibility would be suitable.

At this point it is important to recall the results of Ågotnes et al. [2006] that show that CL cannot be interpreted directly in coalitional games without transferable payoff by taking the outcomes to be states. This suggests that also in the case of the cooperation logic with actions and preferences as we defined in Chapter 4, establishing a direct correspondence to coalitional games without transferable payoff might not be possible. However, we will show that we can investigate concepts similar to that of the core of a coalitional game in the logical framework developed in this thesis.

6.1 Core

One of the most important solution concepts of coalitional games is the core. The core of a coalitional game is the set of outcomes the grand coalition can achieve such that no coalition can achieve some other outcome that is strictly better for all its members. Formally, the core is defined as follows.

Definition 6.2 (Core)**.** *The core of a coalitional game without transferrable payoff $\langle Ag, \Omega, \mathcal{V}, \preceq\rangle$ is the set of all $\omega \in \mathcal{V}(Ag)$ for which there is no coalition G such that there is an outcome $\omega' \in \mathcal{V}(G)$ for which $\omega \prec_i \omega'$ for all $i \in G$.*

In the remainder of this chapter, we will explore concepts in multi-agent systems with preferences that capture ideas similar to the core of a coalitional game. Note that we do not claim to have found direct correspondences of the core but we rather take some inspiration from the concept of the core and try to find similar concepts in the framework of multi-agent systems with preferences. Moreover, we will give an outline of how the cooperation logic with actions and preferences can deal with those concepts.

6.1.1 Global Core in a Multi-agent System as a Set of States

Let us look at a multi-agent system from the outside and not from a local perspective of a particular state. Then one way of incorporating a concept in the style of the core into multi-agent systems with preferences would be to consider the set of states that have the following property: There is no group of agents that has the ability of making the system move into a state that is strictly preferred over the current state by all the members of the group.

So, more formally, the core of a multi-agent system can be seen as the following set:

$$\begin{array}{lcl} Core(M^{\preceq}) & := & \{s \in S \mid \text{there is no group } G \subseteq Ag \text{ such that} \\ & & \text{there is a set of actions } A \subseteq \mathsf{act}(G) \text{ such that} \\ & & \text{for any } t \in S \text{ it holds that if } s \rightarrow_{A \cup B} t \text{ for some} \\ & & B \subseteq \mathsf{act}(Ag \setminus G) \text{ then } s \prec_i t \text{ for all } i \in G\}. \end{array}$$

Definition 6.3 (Global Core)**.** *Given a multi-agent system with preferences $M^{\preceq} = \langle S, Ac, (\rightarrow)_{A \subseteq Ac}, \Phi_0, Ag, \mathsf{act}, \{\preceq_i\}_{i \in Ag}, \pi\rangle$, the global core $Core(M^{\preceq})$ of it is the set of all $s \in S$ such that there is no group of agents $G \subseteq Ag$ such that the following holds: There is a set of actions $A \subseteq \mathsf{act}(G)$ such that for any $B \subseteq \mathsf{act}(Ag \setminus G)$ and $t \in S$ such that $s \rightarrow_{A \cup B} t$ it holds that for all $i \in G$ $s \prec_i t$.*

Now, we will try to characterize $Core(M^{\preceq})$ by a formula in the cooperation logic with actions and preferences that we defined earlier. We want to find a formula $\Psi_{Core(M^{\preceq})}$ such that for every state s it holds that

$$M^{\preceq}, s \models \Psi_{Core(M^{\preceq})} \quad \text{iff} \quad s \in Core(M^{\preceq}).$$

The idea is that a state is a core state if and only if for every group and action they can perform there is always at least one member that will not be better of in one of the states the system can move into after the group doing that action. Now, we claim that the following formula characterizes the states of the global core.

$$\Psi_{Core(M^{\preceq})} := \bigwedge_{G \subseteq Ag} \bigwedge_{A \subseteq \mathsf{act}(G)} \left(\bigvee_{i \in G} \neg \left[\bigwedge \Phi(A, G) \right]^{\prec_i} \top \right)$$

It says that for every group of agents it holds that for every action that they can perform there is always a member of the group who does not strictly prefer one of the possible next states over the current one. For better understanding of this formula, let us recall that by Lemma 3.18 an action of type $\bigwedge \Phi(A', G')$ is always of the form $A' \cup B$ for some $B \subseteq \mathsf{act}(Ag \setminus G')$. Also recall that $M^{\preceq}, s \models [\alpha]^{\prec_i}\top$ means that any action of type α leads to a state strictly preferred over s by i.

Next, we show that this formula indeed characterizes the states that belong to the global core.

Proposition 6.4. *Let $M^{\preceq} = \langle S, Ac, (\rightarrow)_{A \subseteq Ac}, \Phi_0, Ag, \mathsf{act}, \{\preceq_i\}_{i \in Ag}, \pi\rangle$ be a multi-agent system with preferences. Then for any state $s \in S$:*

$$s \in Core(M^{\preceq}) \textit{ iff } M^{\preceq}, s \models \Psi_{Core(M^{\preceq})}.$$

Proof.

($\Rightarrow$) Assume that $M^{\preceq}, s \nvDash \Psi_{Core(M^{\preceq})}$. Then there is a group of agents $G \subseteq Ag$ and a set of actions $A \subseteq \mathsf{act}(G)$ such that for all agents $i \in G$: $M^{\preceq}, s \vDash [\bigwedge \Phi(A,G)]^{\prec_i} \top$. Then this means that in state s any action of type $\bigwedge \Phi(A,G)$ will lead to a state that is strictly preferred by all the members of G over s. Then s cannot be in $Core(M^{\preceq})$ because by doing A group G can make the system move to a state which is strictly preferred over s by all the members of G.

($\Leftarrow$) Assume that s is not in $Core(M^{\preceq})$. Then there is a group $G' \subseteq Ag$ such that for some set of actions $A' \subseteq \mathsf{act}(G')$ that the group can perform, it holds that for all $B \subseteq \mathsf{act}(Ag \setminus G')$ and $t \in S$: if $s \rightarrow_{A' \cup B} t$ then $s \prec_i t$ for all $i \in G'$. By Lemma 3.18, which says that every action of type $\bigwedge \Phi(A', G')$ is of the form $A' \cup B$ for some $B \subseteq \mathsf{act}(Ag \setminus G')$, we can then conclude that in state s every action of type $\bigwedge \Phi(A', G')$ leads to a state that is strictly preferred over s by every member of G'. Thus, $M^{\preceq}, s \vDash \bigwedge_{i \in G'} [\bigwedge \Phi(A', G')]^{\prec_i} \top$. Thus, $M^{\preceq}, s \nvDash \Psi_{Core(M^{\preceq})}$. This concludes the proof. □

In cooperative game theory, quite a lot of work has been done investigating the solution concept of the core with respect to the conditions under which the core of a coalitional game is nonempty [Conitzer and Sandholm, 2002]. If a coalitional game has a nonempty core, this means that there is an outcome that the grand coalition can achieve such that there is no coalition that has an incentive to deviate.

Now, we will look at what a nonempty core means for our multi-agent systems with preferences.

Assume that we have a multi-agent system with preferences $M^{\preceq} = \langle S, Ac, (\rightarrow)_{A \subseteq Ac}, \Phi_0, Ag, \mathsf{act}, \{\preceq_i\}_{i \in Ag}, \pi \rangle$ such that $Core(M^{\preceq}) \neq \emptyset$. This then means that there is a state such that no coalition can make the system move into another state that is strictly preferred by all the members of the coalition. In the case of a nonempty core of a coalitional game we know that the grand coalition has the ability to achieve a core outcome. In multi-agent systems with preferences, a nonempty global core does not mean that the system will ever move into one of the core states. As we defined multi-agent systems, it is perfectly possible that there is an isolated state t that is not accessible by any set of actions from any state different from itself and also does not have access to any other state via any actions. If we add preferences to such a system, this isolated state will then always satisfy $\Psi_{Core(M^{\preceq})}$ simply because there is no action that can make the system leave this state. Suppose that the current state of our system is s where $s \neq t$. Then here is no way for the system to ever end up in t.

These considerations illustrate that in a multi-agent system with preferences, what corresponds to the coalitional game having a nonempty core is not the property that $Core(M^{\preceq}) \neq \emptyset$ alone but we would rather like to add the condition that from every state in the system there is some finite number of transitions that make the system move into a state that is in the core. A sequence of transitions corresponds to a sequence of concurrent actions that are being performed. Then this would mean that no matter in which state the interactive process starts, the agents all together can always perform a finite sequence of actions which finally leads them to a state where there is no coalition G that has the ability to make the system move to another state that is strictly preferred by all the members of G.

Formalizing this idea, we get the following.

$$\forall s \in S \exists n \in \mathbb{N} : M^{\preceq}, s \models \langle\langle Ag \rangle\rangle^n \Psi_{Core(M^{\preceq})},$$

where $\langle\langle G \rangle\rangle^n \varphi$ just means that $\underbrace{\langle\langle G \rangle\rangle \langle\langle G \rangle\rangle \ldots \langle\langle G \rangle\rangle}_{n \text{ times}} \varphi$.

Note that the language of the cooperation logic with actions and preferences allows expressions of the form $\langle\langle G \rangle\rangle^n \varphi$ but we do not have any expression of the form $\langle\langle G \rangle\rangle^* \varphi$ saying that there is some $n \in \mathbb{N}$ such that $\langle\langle G \rangle\rangle^n \varphi$.

Still, there is a problem remaining with our concept of the global core in a multi-agent system with preferences. Even if we add the condition that from everywhere in the system some state in the core has to be reachable, this still allows the core to consist of states that the system can never leave once it is in there. In such states, the formula $\Psi_{Core(M^{\preceq})}$ is trivially satisfied. Here, the stability of the state comes from the lack of possibilities to move out of it.

Even though it might not be an adequate characterization of the core, the formula $\Psi_{Core(M^{\preceq})}$ nevertheless characterizes a set of states that play a special role in the multi-agent system. Whenever the system is in one of these states, there is no group that has the incentive and ability to make the system move to another state. Within the process of interaction, whenever the system enters one of the core-states, at a local perspective, i.e. looking only at the state and not at what has happened before, the agents – if memoryless – are happy or at least cannot complain since there is no way of improving their situation anyway.

6.1.2 Local Core in a Multi-agent System as a Set of Concurrent Actions

Now, we will consider an alternative way of how to capture the concept of the core in multi-agent systems with preferences. In cooperative games, the elements of the core are outcomes that the grand coalition can achieve. We will try to find a corresponding concept in our framework on a local level. Assume that we are in a state s. Then the group consisting of all agents Ag has the ability to determine which of the accessible states the system will move into next. All together, the agents can choose any of the accessible states and make the system move there. This follows from our assumption that the system is deterministic.

Now, we will investigate the core as a solution concept that assigns to every pair of state and preference profile determines a set of concurrent actions. In cooperative games, the core contains outcomes that the grand coalition can achieve such that there is no coalition that can achieve something that is strictly better for all its members.

On a local level, looking at a state of the multi-agent system and the actions the agents can perform, the core can be seen as to correspond to a concurrent action that leads to a state such that no coalition can force the system to move to a state that is strictly better than that one for all its members. Formally, we can define a core solution concept in the following way.

Definition 6.5 (Core Actions). *Let* $M^{\preceq} = \langle S, Ac, (\rightarrow)_{A \subseteq Ac}, \Phi_0, Ag, \mathsf{act}, \{\preceq_i\}_{i \in Ag}, \pi\rangle$ *be a multi-agent system with preferences. Then the core solution concept* $\mathcal{S}_{CoreAct(M^{\preceq})}$ *is a function*

$$\mathcal{S}_{CoreAct(M^{\preceq})} : S \rightarrow 2^{2^{Ac}}.$$

It is defined as follows.

$A^* \in \mathcal{S}_{CoreAct(M^{\preceq})}$ *iff* *there is no* $G \subseteq Ag$ *and* $A \subseteq \mathsf{act}(G)$ *such that* $\forall B \subseteq \mathsf{act}(Ag \setminus G), t \in S$ *if* $s \rightarrow_{A \cup B} t$ *then for* t^* *such that* $s \rightarrow_{A^*} t^* : t^* \prec_i t$ *for all* $i \in G$.

Then, in a state of the multi-agent system with preferences a core action is a concurrent action that the grand coalition can perform such that there is no group that can perform an action that in any case will lead to a state better than the result of A^* for all its members.

This definition of a core action seems to capture the concept of a core in a cooperative game quite well. Unfortunately, in the cooperation logic with actions and preferences as we defined it, we cannot express that a certain action is a core action or that in some state there is a core action.

This follows from the following. For characterizing that in a certain state there is a core action, it is necessary to be able to compare accessible states to each other according to a preference relation $\preceq_i$. In our logic we can only compare states to the current state.

Summarizing this chapter, we can say that we are able to characterize a concept in our logic that has a similar idea as the core. It captures the states of the system in which no coalition has the power of forcing the system to move to a state that is strictly better for all its members. However, this property is trivially satisfied by states that the system can never leave. Also it can be the case that the system can never reach a state satisfying the property.

At a local level, we can define a solution concept that gives us for every state the set of actions A^* that the grand coalition can force such that there is no coalition that can force the system to move to a state strictly better than the result of A^* for all its members.

Chapter 7

Conclusion

This chapter concludes the thesis. We will give a brief summary, discuss some interesting aspects that arise from the work we presented and give an outline of future work.

7.1 Summary

In this thesis, we investigated logics for reasoning about cooperation, actions and preferences in interactive situations in multi-agent systems.

We gave an overview of several logical approaches for reasoning about one or two of the aforementioned concepts and presented one way of developing a formal framework for reasoning about all three.

The logic that we developed is an extension of the cooperation logic with actions developed by Sauro et al. [2006]. It is based on an environment represented as a transition system labelled by sets of actions. Agents are added and are provided with the ability to interact by each being able to perform a certain set of actions. Then the cooperative ability of a group of agents to perform actions is obtained in a straightforward way by saying that a group has the ability to perform exactly those sets of actions that consist of actions its members can perform. We examined the cooperative ability of groups more closely and showed how a group of agents can choose its actions in a way such that the other agents have as little power as possible to influence what exactly will be the result of the concurrent action performed by all the agents together.

Moreover, we presented some ideas of how an analysis of the distribution of action abilities among the members of a group can provide us with some insights into the role individual agents play within a group when trying to force some state of affairs φ. In particular, we can specify conditions for the action ability distribution under which some agent is needed for a group being able to force φ. For the converse, we also investigated what we can conclude about the action some agent can perform if we know that a group can only enforce some φ if he is a member of it.

Using the cooperation logic with actions as a base for our logic, we added a representation of the agents' preferences as binary preference relations over the set of states of the environment. In this process, as an intermediate step we developed a framework in which the agents do not have abilities yet to interact in the system. This environment with preferences represents how the agents see the dynamic system from the viewpoint of an external observer. They can see which effects the actions have and they have preferences over the states the dynamic system can move into but they are not yet able to actively take part in it.

The logic we developed for reasoning in this framework contains an expression saying that every action of a certain type will always lead to a state preferred over the current one according to some preference relation. The motivation for introducing such an expression was the following: When reasoning about cooperative abilities and preferences as it is e.g. done in cooperative game theory, the ability of a group to achieve some result or outcome that is better than some other one (according to a certain preference relation) comes up in many situations. Trying to make this ability more explicit in terms of *how* a group can achieve such an improvement led us to an expression saying that every action of a certain type leads to a state preferred over the current one by some agent.

We can show soundness and completeness of the environment logic with preferences by using the soundness and completeness of the environment logic and the preference logic.

Then we added the agents' abilities to perform actions to the environment with preferences and thereby gave the agents the ability to actively take part in the dynamic system.

In the resulting logic, the ability of a group to achieve some state of affairs φ and at the same time achieve that the next transition is an improvement according to some preference relation (in our case the preference relation of an individual agent) can be made explicit. It corresponds to the group's ability to perform an action of a certain type that is guaranteed to lead to a state that is better than the current one according to the preference relation under consideration and that also satisfies φ. Again, soundness and completeness can be shown by using the soundness and completeness of the sublogics, i.e. the environment logic with preferences and the cooperation logic with actions.

Chapter 5 has shown that the semantic structures that we developed, namely the multi-agent systems with preferences, can be used for investigations very much in the style of what is done in the field of mechanism design but with the additional feature that we can make explicit how the abilities of the agents to perform certain actions affect the properties of choice rules implementable in a multi-agent system.

In Chapter 6, we examined how the cooperation logics with actions and preferences that we developed can be used for reasoning that involves concepts from cooperative game theory. We can use the logic to capture that a state is stable in the sense that no coalition has the ability to make the system move to another state that is strictly better for all its members.

7.2 Discussion

In the cooperation logic with actions [Sauro et al., 2006], basically the way how cooperative ability is made explicit here is by identifying each agent with a set of actions he can perform and groups of agents with the union of the sets of actions its members can perform. This is a central point here since the whole approach that we present is based on the assumption that cooperative ability can be made explicit in this way. One possible objection is that we can think of actions that seem primitive and that can only be performed by groups that contain at least a certain number of agents. Then we would have to split such an action up into actions for the individuals representing their part of the action.

Modularity is a central property of the approach that we chose. From a modelling perspective, our modular approach has the advantage that we can use the different submodules for reasoning about individual aspects of cooperation in a multi-agent system. On a technical level, soundness and completeness can be shown by using the soundness and completeness of the logics of the submodules.

As a field of applications of our proposed framework, let us briefly consider model checking. Here the explicit representation of actions and preferences has the following advantage: For showing that the statement $\langle\langle G^{\preceq_i}\rangle\rangle\varphi$ is satisfied in some model, the model checker would then generate an explicit plan (which in this case is only of one step, i.e. one concurrent action) for the group to achieve φ in a way that the execution of the plan leads to an improvement of the situation according to agent i.

Comparing our work to other cooperation logics that combine cooperative ability and an explicit representations of actions such as CLA [Sauro et al., 2006] and CAL [Borgo, 2007], we can say that with the additional representation of the preferences that we have in our logic, we are able to make some inferences as to *why* a group would decide to force φ. Additionally this then also gives us some information about *if* they would ever do it, which is clearly more useful than the information that in general a group has the ability to force φ.

We chose to combine cooperative ability and preferences of single agents in a way that we can reason about the group's ability to force that the system will move into some state that is preferred over the current one by some agent. Alternatively, as we discussed in Section 4.3.1, instead of having expressions of the form $\langle\langle G^{\preceq_i}\rangle\rangle\varphi$, where i is an individual, we could have also considered analogous expressions with different group preference relations obtained from the individual preferences in various ways.

As we mentioned in Section 4.3.3, the cooperation logic with actions and preferences can be extended in a straightforward way to a logic in the style of quantified coalition logic [Ågotnes et al., 2007b]. Such an extension will allow for a more succinct expression of statements saying that coalitions satisfying a certain property have the ability to achieve a certain state of affairs or can force an action of a certain type.

When investigating the connection between our framework and concepts from cooperative game theory, we noticed that the connection between cooperative games and the multi-agent systems with preferences as we defined them is not straightforward. We believe that the reasons for that are related to the fact that CL cannot be interpreted directly in coalitional games without transferrable payoff as has been shown by Ågotnes et al. [2006]. The main problem in establishing the connection seems that in coalitional games the outcomes are local with respect to the groups, whereas when taking outcomes as states of the multi-agent systems with preferences this is not the case any more.

7.3 Future Work

As future work, relating the ideas presented in Section 3.2, where we investigated the connection between the abilities of an agent to perform certain actions and the agents importance for a group to achieve some state of affairs, to the preferences is particularly interesting. This will provide us with deeper insights into how the distribution of the action abilities among the agents in a group effects the power of the agent to influence the group action such that the resulting state will be one that is 'good' for him according to some solution concept.

Furthermore, the preference language that we used can be extended and the preferences of the agents over states can be lifted to preferences over formulas [van Benthem et al., 2005; van Benthem et al., 2007]. The resulting language then has a global existential modality and contains binary preference mobilities in a addition to the unary ones. Then we can express that an agent e.g. prefers all states satisfying φ over all states satisfying ψ. In such a framework, we could then reason about why a coalition would e.g. force φ rather than ψ.

The approach we presented in this thesis starts with an environment in which actions and their effects are modelled. This environment is then populated by agents and cooperative ability is obtained from the actions they can perform individually and as groups. It would be very interesting (and promising with regard to an investigation of cooperative games) to develop a logic for cooperation action and preferences by starting from another point; namely base it on a logic for cooperative ability that can indeed be directly interpreted in coalitional games like it is the case for coalitional game logic (CGL) [Ågotnes et al., 2006] and then try to make the coalitional power more explicit.

One way of doing so would be to add some representation of actions and their effects. Another way which seems easier to realize is to make the coalitional power of groups more explicit by showing how the ability of a group to achieve some outcome depends on the abilities of its members to achieve 'sub-outcomes', similar as in the case of the logic of cooperation and propositional control [van der Hoek and Wooldridge, 2005].

Acknowledgements

First of all, I would like to thank my supervisors Eric and Ulle. I am very grateful to Eric especially for introducing me to the topic and for his advice in developing the central ideas of this thesis. Thanks to Ulle especially for always providing useful feedback about the current state of my work and for his help in presenting my ideas in a clear way.

I am also grateful to Can for reading this thesis and for his comments that helped me to increase the readability.

Bibliography

J. Abdou and H. Keiding. *Effectivity Functions in Social Choice.* Springer, 1991.

T. Ågotnes, W. van der Hoek, and M. Wooldridge. On the logic of coalitional games. In *AAMAS '06: Proceedings of the fifth International Joint Conference on Autonomous Agents and Multi-agent Systems*, pages 153–160, Hakodate, Japan, 2006.

T. Ågotnes, P. E. Dunne, W. van der Hoek, and M. Wooldridge. Logics for coalitional games. In *LORI '07: Proceedings of the Workshop on Logic, Rationality and Interaction*, Beijing, China, 2007a. to appear.

T. Ågotnes, W. van der Hoek, and M. Wooldridge. Quantified coalition logic. In *IJCAI '07: Proceedings of the twentieth international joint conference on Artificial Intelligence*, pages 1181–1186, Hyderabad, India, 2007b.

R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Lecture Notes in Computer Science*, 1536:23–60, 1998.

K. J. Arrow. *Social Choice and Individual Values, Second edition.* Yale University Press, 1970.

P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic.* Number 53 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, UK, 2001.

E. Bonzon, M.-C. Lagasquie-Scheix, J. Lang, and B. Zanuttini. Boolean games revisited. In *Proc. 17th European Conference on Artificial Intelligence (ECAI'06)*, pages 265–269, 2006.

S. Borgo. Coalitions in action logic. In *IJCAI '07: Proceedings of the twentieth international joint conference on Artificial Intelligence*, pages 1822–1827, Hyderabad, India, 2007.

S. Borgo. Quantificational modal logic with sequential Kripke semantics. *Journal of Applied Non-Classical Logics*, 15(2):137–188, 2005.

V. Conitzer and T. Sandholm. Complexity of determining non-emptiness of the core. Technical report, Carnegie Mellon University, 2002.

G. Gargov and S. Passy. A note on Boolean modal logic. In P. P. Petkov, editor, *Mathematical Logic. Proceedings of the 1988 Heyting Summerschool*, pages 311–321. Plenum Press, 1990.

J. Gerbrandy and L. Sauro. Plans in cooperation logic: a modular approach. In *Proceedings of the IJCAI Workshop on Nonmontonic Reasoning, Action and Change (NRAC 2007)*, Hyderabad (India), 2007.

V. Goranko. Coalition games and alternating temporal logics. *TARK: Theoretical Aspects of Reasoning about Knowledge*, 8, 2001.

S. O. Hansson. Preference logic. In *Handbook of Philosphical Logic (Second Edition)*, pages 319–393. Kluwer, 2001.

D. Harel. Dynamic logic. In D. Gabbay and F. Guenther, editors, *Handbook of Philosophical Logic Volume II – Extensions of Classical Logic*, pages 497–604. D. Reidel Publishing Company, Dordrecht, The Netherlands, 1984.

P. Harrenstein, W. van der Hoek, J. Meyer, and C. Witteveen. Boolean games. In J. van Benthem, editor, *Proceedings of TARK 2001*, pages 287–298. Morgan Kaufmann, 2001.

K. L. McMillan. *Symbolic model checking – an approach to the state explostion problem.* PhD thesis, Carnegie Mellon University, 1992.

R. C. Moore. Reasoning about knowledge and action. Technical Note 191, Artificial Intelligence Center, SRI International, 1980.

M. J. Osborne and A. Rubinstein. *A course in game theory.* MIT Press, Cambridge, MA, 1994.

M. Pauly. *Logic for social software.* PhD thesis, University of Amsterdam, 2001. ILLC Dissertation Series DS-2001-10.

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

L. Sauro, J. Gerbrandy, W. van der Hoek, and M. Wooldridge. Reasoning about action and cooperation. In *AAMAS '06: Proceedings of the fifth International Joint Conference on Autonomous Agents and Multi-agent Systems*, pages 185–192, Hakodate, Japan, 2006.

G. F. Schumm. Transitivity, preference and indifference. *Philosophical Studies*, 52:435–437, 1987.

J. van Benthem, S. van Otterloo, and O. Roy. Preference logic, conditionals and solution concepts in games. In *Festschrift for Krister Segerberg.* University of Uppsala, 2005.

J. van Benthem, O. Roy, and P. Girard. Everything else being equal: A modal logic approach to ceteris paribus preferences, 2007.

W. van der Hoek and M. Wooldridge. Cooperation, knowledge, and time: Alternating-time temporal epistemic logic and its applications. *Studia Logica*, 75(1):125–157, 2003.

W. van der Hoek and M. Wooldridge. On the logic of cooperation and propositional control. *Artificial Intelligence*, 164(1-2):81–119, 2005.

W. van der Hoek, W. Jamroga, and M. Wooldridge. A logic for strategic reasoning. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 157–164, New York, NY, USA, 2005. ACM Press.

G. von Wright. *The logic of preference.* Edinburgh University Press, 1963.

M. Wooldridge and N. R. Jennings. Formalizing the cooperative problem solving process. In *13th International Workshop on Distributed Artificial Intelligence (IWDAI-94)*, pages 403–417, Lake Quinhalt, WA, USA, 1994.