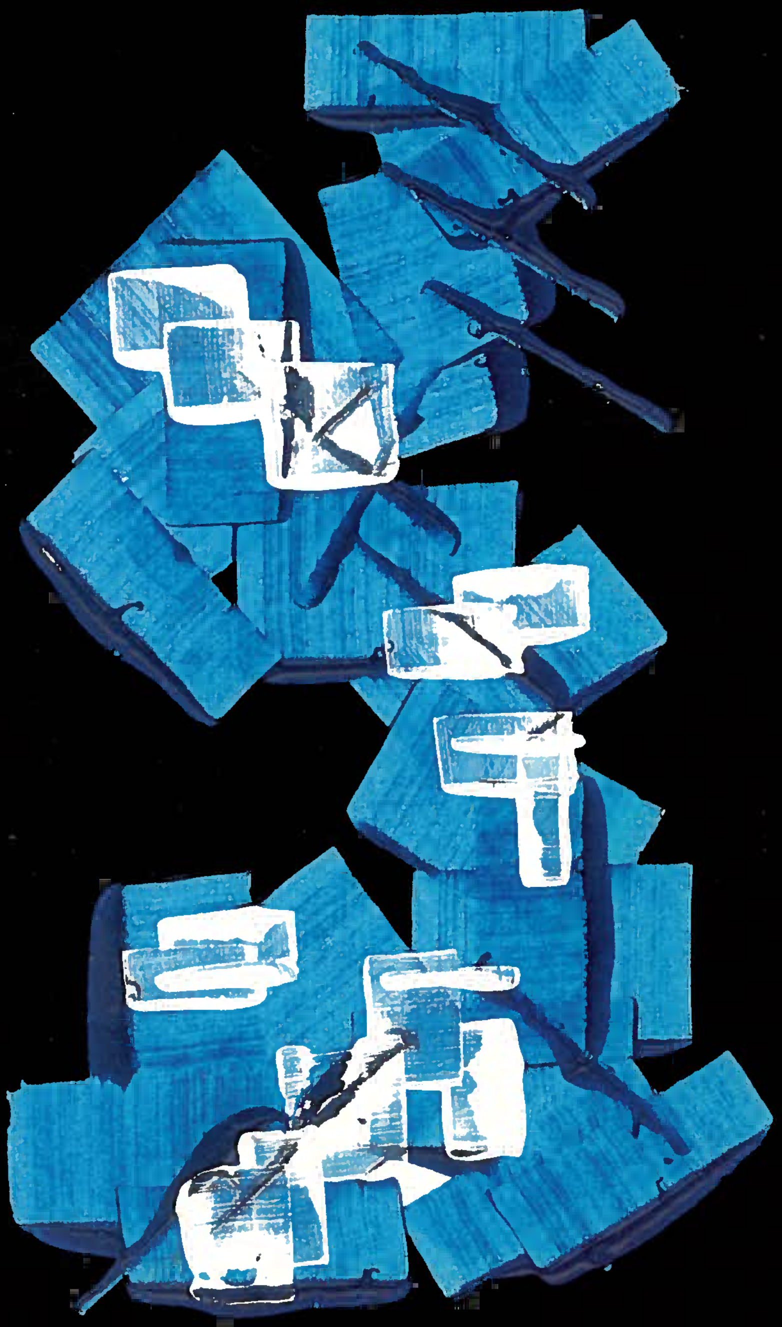
Logic for Social Software



Marc Pauly

Logic for Social Software

ILLC Dissertation Series 2001-10

INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

For further information about ILLC-publications, please contact

Institute for Logic, Language and Computation
Universiteit van Amsterdam
Plantage Muidergracht 24
1018 TV Amsterdam
phone: +31-20-525 6051
fax: +31-20-525 5206
e-mail: illc@wins.uva.nl
homepage: http://www.illc.uva.nl/

Logic for Social Software

Academisch Proefschrift

ter verkrijging van de graad van doctor aan de
Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof.dr. J.J.M. Franse
ten overstaan van een door het college voor
promoties ingestelde commissie, in het openbaar
te verdedigen in de Aula der Universiteit
op donderdag 13 december 2001, te 10.00 uur

door

Marc Pauly

geboren te Mönchengladbach, Duitsland.

Promotores: Prof.dr. J. van Benthem
Prof.dr. J. van Eijck

Faculteit der Natuurwetenschappen, Wiskunde en Informatica
Universiteit van Amsterdam
Plantage Muidergracht 24
1018 TV Amsterdam

The research for this thesis was carried out at the Center for Mathematics and Computer Science (CWI) in Amsterdam as part of the Spinoza project *Logic in Action*.

Cover and bookmark design by Barbara Düsselberg
based on an art work by Agnes Pauly.

ISBN: 90–6196–510–1

Contents

Acknowledgments

My mother would like to thank Johan van Benthem for bringing me to the Netherlands, which reduced the travel distance to my parents' house from 10 hours by plane to 2 hours by train. My father would like to thank Jan van Eijck for organizing OzsL school weeks, which introduced me to Italian culture. I would like to thank both Johan and Jan for their advice regarding this thesis and their general support during the past four years.

For comments on earlier drafts of this thesis I am especially grateful to Dietmar Berwanger, Hans van Ditmarsch, Peter van Emde Boas, Valentin Goranko, Rohit Parikh, Ben Rodenhäuser and Yde Venema. Erich Grädel, Martin van Hees and Ronald de Wolf have provided much appreciated further suggestions regarding, respectively, the μ-calculus, social choice theory and Sperner's theorem. Also, I thank all members of the logic and games gang in and around Amsterdam for the many discussions we have had.

Eva Hoogland has done her best trying to shorten the length of my sentences, but I'm afraid that since she only read the Dutch abstract, the rest of this thesis may still be viewed by some as an example of German philosophical prose at its worst, with sentences which go on and on until finally at the very end they reveal what one would have liked to know at the beginning, the verb.

The Centrum voor Wiskunde en Informatica (CWI) and the Institute for Logic, Language and Computation (ILLC) have provided intellectually stimulating environments to work in. Furthermore, short research visits to places outside the Netherlands have convinced me that universities are like travel agencies and airline companies, they allow their employees to travel the world at almost no cost. Besides, already behind their desks at home, scientists are just tourists of knowledge after all, as a colleague once remarked. (In fact, he made this remark before going on vacation, in order to console those who had to stay at home learning for exams.) But before starting to give an account of the 4 years of this purely intellectual tourism, I will briefly recapitulate some of the cases where inner and outer travel coincided, also wishing to thank those who made these

travels possible.

In February of 1999, Gabriel Sandu invited me to Helsinki together with Theo Janssen, resulting in a visit which furthered not only my interest in Independence-Friendly logic but also my knowledge of extreme climate experiences with a few days of $-20°$ Celsius. We only survived these due to the sauna of the university guest house. Part of spring 2000 I spent in New York, visiting Rohit Parikh. There, my study of Coalition Logic got an unintended practical bend: The first night at my apartment I was welcomed by an overflowing toilet. While this was probably not the best way of introducing myself to the neighbors, we ended up with a coalitional strategy for taking care of the mess. Later in that year, I visited Erich Grädel and his group in Aachen for a week in October. Together with Dietmar Berwanger, we had a rather intense week trying to figure out the exact relationship between Game Logic and the μ-calculus. In particular sections 6.4 and 7.2.2 have benefitted from these discussions. January 2001 gave me another opportunity to dip into the German university culture when Hugo Volger invited me to Passau. This trip conclusively established that I was turning into a real Dutchman: German formalities triggered some amusement when I was asked for my academic title, to be printed in the lecture announcement. My answer "preferably none" was unable to prevent the announcement of *Herr M.Sc. Marc Pauly, Amsterdam.* The talk (held in German) went well, and I was particularly happy to be congratulated with my German which, after all, contained only a minor Dutch accent. This experience yielded the final impetus for finishing this thesis before it was too late …

Abstract

The term *social software* refers to the project of analyzing social procedures and processes using the formal methods of computer science. Examples of the social procedures we have in mind are cake-cutting algorithms and voting procedures. What distinguishes these procedures is that there are a number of agents who interact strategically in a well-defined manner. We are interested in developing logical tools for proving the correctness and efficiency of such social software. For a cake-cutting algorithm, this may mean showing that the algorithm can guarantee a fair piece to everyone with only a few cuts. For a voting procedure, we would want a fair distribution of political power while reducing the number of votes which need to be taken to a minimum.

Two logics will be developed and studied for that purpose. Coalition Logic, introduced in chapter 3, allows one to reason about the power of coalitions in various kinds of extensive games. The formula $[C]\varphi$ expresses that at the present state of the game/process, the group of agents C has a joint strategy for bringing about a next state where φ holds. We provide complete axiomatizations of this logic for extensive games with and without simultaneous moves, as well as complexity results for model checking and the satisfiability problem. Chapter 4 extends Coalition Logic with an extra modality, adding formula $[C^*]\varphi$ which expresses that the group of agents C has a joint strategy for bringing about φ at some time in the future.

The metatheoretic results obtained for Coalition Logic also allow us to compare reasoning about individuals to reasoning about groups: For certain classes of social processes, reasoning about individuals is less complex than reasoning about groups, assuming that NP $\neq$ PSPACE. Similarly, we can compare different classes of processes in terms of their complexity, asking, e.g., whether reasoning about situations where agents can act simultaneously is more or less complex than reasoning about situations where agents can only act one after the other.

Chapter 5 provides some examples of how Coalition Logic can be applied in the analysis and synthesis of social processes. Most of the examples are essen-

tially voting procedures, including also the Bonn vs. Berlin debate of the German parliament. The verification of properties of a social process can be done via model checking in Coalition Logic. Generating a process which satisfies a certain specification on the other hand can be formulated as a satisfiability problem.

Game Logic is the second logic for social software we study (chapters 6 and 7). In contrast to Coalition Logic, here the social process is an explicit part of the logical language, where the formula $\langle\gamma\rangle\varphi$ expresses that player 1 has a strategy in process/game γ for achieving an outcome which satisfies φ. The language of Game Logic allows one to reason about determined 2-player games. It also contains game operations such as sequential composition, choice, and role interchange for constructing complex games which have internal structure.

We compare Game Logic to a number of well-known logics which have been proposed for reasoning about programs (i.e., 1-player games). By looking at the complexity and expressive power of the different logics, we are able to compare how reasoning about programs differs from reasoning about games. As may be expected, games can be more complex than programs, and more generally, verifying properties of a game becomes more complex the more players alternate in taking turns.

This thesis tries to build a bridge between computer science on the one hand and game theory and social choice theory on the other hand. The logics discussed in this thesis are extensions of modal logics used in computer science for reasoning about computational processes. More precisely, Coalition Logic is closely related to Alternating Temporal Logic whereas Game Logic is a cousin of Propositional Dynamic Logic and the modal μ-calculus. On the other hand, we make use of notions and results from game theory and the theory of social choice, in particular in chapter 2 which develops a general semantic model underlying both Coalition Logic and Game Logic. At the core of this model lies the notion of an effectivity function which has been studied extensively in social choice theory.

Samenvatting

De term *sociale software* verwijst naar het projekt, dat probeert om met behulp van formele methoden uit de informatica, sociale procedures en processen te analyseren. Voorbeelden van zulke processen zijn algoritmen voor het verdelen van een koek en verkiezingsprocedures. Kenmerkend voor dit soort procedures is dat een aantal agenten betrokken zijn in een wel-gedefinieerde strategische interactie. Wij zijn geïnteresseerd in het ontwikkelen van logische hulpmiddelen om de correctheid en efficiëntie van zulke sociale software te kunnen bewijzen. Voor het koek-verdeel algoritme betekent dit aan te tonen dat iedere persoon met zo min mogelijk sneden een voldoende groot stuk krijgt. Voor een verkiezingsprocedure zouden wij willen aantonen dat de politieke macht op een juiste manier is verdeeld en dat het aantal stemmingen tot een minimum wordt beperkt.

Wij behandelen twee logica's voor deze taak. Coalition Logic, geïntroduceerd in hoofdstuk 3, is geschikt voor het redeneren over de macht van coalities in verschillende soorten extensieve spelen. De formule $[C]\varphi$ beweert dat op dit moment van het spel de groep C een gemeenschappelijke strategie heeft om in één stap een situatie te bereiken die aan φ voldoet. Wij geven volledige axiomatiseringen van deze logica voor extensieve spelen met en zonder simultane zetten, en geven verder resultaten over de complexiteit van model checking en het vervulbaarheidsprobleem. Hoofdstuk 4 breidt Coalition Logic uit met een extra modaliteit. De nieuwe formule $[C^*]\varphi$ beweert dat groep C een gemeenschappelijke strategie heeft om φ in een willekeurig aantal stappen te bereiken.

De metatheoretische resultaten voor Coalition Logic staan ons ook toe een vergelijking te maken tussen het redeneren over individuen en het redeneren over groepen: Voor sommige klassen van sociale processen is redeneren over individuen makkelijker dan redeneren over groepen, verondersteld dat NP $\neq$ PSPACE. Op dezelfde manier vergelijken wij de complexiteit van verschillende soorten processen, bijv. door te vragen of het redeneren over situaties waar agenten simultaan handelingen verrichten minder of meer complex is dan het redeneren over situaties waar agenten alleen achter elkaar acties kunnen uitvoeren.

Hoofdstuk 5 geeft voorbeelden die laten zien hoe Coalition Logic kan worden gebruikt in de analyse en synthese van sociale processen. De meeste voorbeelden zijn in essentie verkiezingsprocedures zoals het debat over het Bonn-Berlijn vraagstuk in de Duitse Bondsdag. De verificatie van eigenschappen van een sociaal proces gebeurt via model checking in Coalition Logic. De generatie van een sociaal proces dat voldoet aan een bepaalde specificatie aan de andere kant wordt beschreven in termen van een vervulbaarheidsprobleem.

Game Logic is de tweede logica voor sociale software die wij bestuderen (hoofdstukken 6 en 7). In tegenstelling tot Coalition Logic is het sociale proces in Game Logic een expliciet onderdeel van de logische taal, waarbij de formule $\langle\gamma\rangle\varphi$ weergeeft dat speler 1 in spel γ een strategie heeft voor het bereiken van een situatie die aan φ voldoet. De taal van Game Logic is geschikt om te redeneren over gedetermineerde spelen voor 2 spelers. In deze taal kunnen ook speloperaties uitgedrukt worden zoals sequentiele compositie, keuze en rolverwisseling.

Wij vergelijken Game Logic met een aantal bekende logica's voor het redeneren over programma's (i.e., spelen voor 1 speler) door te kijken naar de complexiteit en de uitdrukkingskracht van de verschillende logica's. Op die manier kunnen wij redeneren over programma's vergelijken met redeneren over spelen. Zoals te verwachten is, zijn spelen vaak complexer dan programma's, en algemener geldt dat het verifiëren van eigenschappen in spelen complexer wordt naarmate de beurten van de spelers vaker alterneren.

Dit proefschrift probeert een brug te slaan tussen informatica aan de ene kant en speltheorie en sociale keuze theorie aan de andere kant. De logica's die in dit proefschrift worden besproken zijn uitbreidingen van de modale logica's die in de informatica worden gebruikt om over computationele processen te redeneren. Zo is Coalition Logic nauw gerelateerd aan Alternating Temporal Logic en Game Logic heeft veel verbanden met Propositional Dynamic Logic en de modale μ-calculus. Aan de andere kant maken wij nogal wat gebruik van noties en resultaten uit de speltheorie en de sociale keuze theorie, met name in hoofdstuk 2 waar wij een algemeen semantisch model ontwikkelen voor zowel Coalition Logic als ook Game Logic. Het model is gebaseerd op de notie van een effectiviteitsfunctie, een notie waar veel onderzoek naar is gedaan in de sociale keuze theorie.

Chapter 1

Let the Games Begin

> *It is an old idea that thinking to the bottom of our knowledge, all human activity seems merely play. Those who are willing to content themselves with a metaphysical conclusion of this kind should not read this book.*
>
> Johan Huizinga

1.1 Logic and Games

Logicians like to play games. In fact, not only logicians like to play games, for homo sapiens seems to be quite a close relative of homo ludens [69]. For the logician, however, playing games can be very useful besides being entertaining. As a consequence, the professional logician comes close to fulfilling one of his childhood dreams: getting paid to play games. In this respect then, he can be compared to professional tennis players, which explains the occasional complaint of a logician about being underpaid. One may even argue that such a complaint gains additional weight by the fact that the logician often manages to come up with winning strategies (i.e., strategies which will be successful against any opponent), something that even the best tennis players can only dream of.

Games have been useful in logic in a variety of ways, some of the most prominent ones being the following (see [12] for a more extensive survey):

Game-Theoretic Semantics: For First-Order Logic, truth in a model can be defined using game-theoretic semantics (see, e.g., [66]). A game-theoretically natural extension of this semantics has led to Independence-Friendly Logic [65, 109]. Likewise, game-theoretic semantics have been proposed for Linear Logic [21, 2].

Dialogue Games: The relation of logical consequence has been viewed as a dialogue game in [83], where the precise rules of the game determine whether one obtains intuitionistic or classical logical consequence. Extensions of this work have yielded dialogue games for various modal logics and Linear Logic [110, 111].

Model-Comparison Games: In recent expositions of model theory (see, e.g., [41, 67]), Ehrenfeucht-Fraïssé games "operationalize" elementary equivalence between models. The number of pebbles used in these games corresponds to the number of variables needed to express a particular property.

Like logic, theoretical computer science also had its share of playfulness, on the most fundamental level through models of computation which are essentially 2-player games [31]. We will have more to say on this issue in section 1.2.

Instead of using games for the purposes of logic, a second line of investigation tries to use logic for the purposes of game theory. And although it seems natural that the study of games and how players should behave in games takes into account players' reasoning and rationality (the domain of logic, it would seem), formal logical approaches to games are of a rather recent origin. The work in this area can be categorized as follows:

Epistemic Logic: Solution concepts such as the Nash equilibrium or the subgame-perfect equilibrium can be investigated concerning their epistemic presuppositions, asking, e.g., which assumptions about rationality and players' belief/knowledge of rationality are necessary to guarantee a Nash-equilibrium outcome. Modal logics have been used to obtain formal epistemic axiomatizations of various solution concepts, both on the propositional [123, 34, 30] and on the first-order level [74].

Dynamic Epistemic Logic: Epistemic logics have also been extended with action modalities in order to express knowledge and belief change. These dynamic epistemic logics can not only express statements like "player 2 knows that player 1 holds the queen of hearts" but also statements like "player 1 knows that after showing his card to player 3, player 2 will know that player 1 holds the queen of hearts" [9, 22]. This approach has been used to formally model the board game *Clue* or *Cluedo* in [40].

Others: Besides epistemic logic, temporal logic has been extended with a prediction relation which captures the backward induction solution of a game [23], and similarly an extension of dynamic logic can be used to axiomatize backward induction [62]. Maybe surprisingly, even simple propositional logic can be used as a description language for extensive games of perfect information which also allows one to formulate axioms of rationality [24]. The role of language in economic decision making has been investigated in [112] using formal logical languages. The work in this thesis also falls into the non-epistemic category.

Note that the above classification does take a rather narrow view of logic. On a less restricted interpretation, all of the discussion in game theory which relates to the epistemic aspects of games falls into the scope of logic, and indeed the modal/epistemic logician recognizes quite a number of the technical notions discussed in the game theory literature [10]. On the other hand, we choose to define the use of logic in game theory more narrowly as the use of formal logical languages. Among other things, using formal languages allows one to ask a number of interesting meta-theoretic questions, examples of which shall be given in this thesis.

1.2 Linking Two Fields of Research

Below we discuss more concretely how logic and game theory come together in this thesis. Additionally, we comment on the general virtues of interdisciplinary research.

1.2.1 Logic in Computer Science

Computers do not always work the way they should (some people would use the more radical scoping: computers always do not work the way they should). Since there may be applications, however, where it is extremely important that computers do work as intended (think, e.g., of air traffic control and other safety-critical tasks), it would be useful to be able to prove that a program works as intended by the programmer, or at least that the program satisfies some crucial properties. Formal logics are used in various branches of computer science as tools for reasoning about programs and software systems more generally. We shall briefly discuss two of these branches below.

The theory of program correctness as it has been understood traditionally [82, 73, 91] is concerned with verifying the correctness of software written in imperative programming languages such as PASCAL, C, and so on. As an example, consider the following program γ_1 which is known as Euclid's algorithm:

$$
\begin{array}{l}
x, y \;\mathtt{:Nat} \\
\mathtt{while}\; x \neq y \;\mathtt{do} \\
\qquad \mathtt{if}\; x > y \;\mathtt{then}\; x := x - y \\
\qquad\qquad\qquad \mathtt{else}\; y := y - x \\
\mathtt{end}
\end{array}
$$

The program is written in a commonly used pseudo-code which is almost self-explanatory: Two variables x and y with the value of natural numbers are compared. If they are equal, the program ends. Otherwise, if x is greater than y, y is subtracted from x, and if y is greater than x, x is subtracted from y. This test and action are repeated until $x = y$.

It is by no means obvious that this program γ_1 in fact calculates the greatest common divisor (gcd) of two natural numbers a and b. If initially $x = a$ and $y = b$, then the program will terminate with $x = y = gcd(a, b)$. Proving the correctness of this program thus would entail a proof of the following statement:

> If γ_1 is started in a state where $x = a$ and $y = b$, then γ_1 will terminate and upon termination $x = y = gcd(a, b)$.

One natural way to represent programs semantically is by way of a *state transformer*, i.e., a relation $R_\gamma \subseteq S \times S$ over the set of states S. We interpret $sR_\gamma t$ as "starting at state s, there is an execution of program γ which terminates in state t". Our program γ_1 has the further property of being *deterministic*, for every state s there is at most one state t such that $sR_{\gamma_1}t$ holds. In general, we allow for programs being nondeterministic. In the case at hand, we can think of S as the set of variable assignments for x and y, that is $S = \{s | s : \{x, y\} \to \mathbb{N}\}$. Proving the above claim then would amount to showing that $sR_{\gamma_1}t$ holds if and only if $t(x) = t(y) = gcd(s(x), s(y))$.

A second way to think about programs is in terms of a *predicate transformer* $F_\gamma : \mathcal{P}(S) \to \mathcal{P}(S)$ which maps a set of states X to the set of states $F_\gamma(X)$ at which some execution of program γ will terminate in a state in X. (Usually one wants *all* executions to terminate in a state in X, but for our present purposes the existential reading is more natural, and for γ_1 the two coincide anyhow.) This approach to program semantics advocated by Dijkstra in [38, 39] is very well-suited for the kind of backward-reasoning one often does when studying the correctness of a program: If the program should bring about a particular state of affairs after termination, which initial condition has to be satisfied for the program to succeed in doing that? To give an example, assume that we want our program γ_1 to end in a state where the value of both x and y equals 3, then under what condition will the program succeed in doing that? Let t be the state where $t(x) = t(y) = 3$. Then $F_{\gamma_1}(\{t\}) = \{s \in S | gcd(s(x), s(y)) = 3\}$, i.e., initially the greatest common divisor of x and y must be 3. A general proof of correctness would then amount to showing that if we start in a state where $x = a$ and $y = b$, the program is guaranteed to end in a state where both variables are set to the greatest common divisor of a and b. Formally, for all states t with $t(x) = t(y) = gcd(a, b)$ we have

$$\{s \in S | s(x) = a \text{ and } s(y) = b\} \subseteq F_{\gamma_1}(\{t\}).$$

While our exposition of predicate transformers was semantic, reasoning about the correctness of programs will usually be done syntactically using axioms and inference rules. Propositional Dynamic Logic [59, 77] can serve as a very simple formalism which links the syntactic and the semantic perspective on predicate transformers on the propositional level. The formula $\langle\gamma\rangle\varphi$ expresses that there is an execution of program γ which terminates in a state satisfying φ. Thus, the modality $\langle\gamma\rangle$ is the syntactic analogue of the predicate transformer F_γ. Logical axioms and rules then syntactically describe the behavior of the predicate transformers, for instance the inference rule

$$\frac{\varphi \to \langle\gamma_1\rangle\psi \qquad \neg\varphi \to \langle\gamma_2\rangle\psi}{\langle\ \texttt{if}\ \varphi\ \texttt{then}\ \gamma_1\ \texttt{else}\ \gamma_2\rangle\psi}$$

says that if the truth of φ implies a terminating γ_1-execution satisfying ψ and the falsity of φ implies a terminating γ_2-execution satisfying ψ, then there is a terminating execution of $\texttt{if}\ \varphi\ \texttt{then}\ \gamma_1\ \texttt{else}\ \gamma_2$ which satisfies ψ.

The two main disadvantages of the traditional approach to program verification are its restricted domain of application and its infeasibility in practice. On the one hand, the kinds of systems which can be analyzed using Dijkstra's approach are programs which can be specified compositionally by means of a fixed number of programming constructs. Furthermore, when started in some initial state, these programs are supposed to terminate in some final state whose properties are then examined. This means that software systems which are intended to run without ever terminating (such as operating systems) as well as systems which have not arisen from a compositional specification cannot be analyzed and verified using a Dijkstra-style approach. On the other hand, even for systems which are amenable to this approach, real-life systems usually turn out to be too complicated to make automatic verification feasible, since such a verification entails theorem proving in a very complex logical formalism.

Because of these drawbacks, another approach to software verification has been investigated more recently which makes use of temporal logics [43, 70]. Here, programs are not part of the logical language but rather the semantic models over which expressions of temporal logic are evaluated. More specifically, process algebras [8, 49] and other tools can be used to describe and generate a process graph which serves as the semantic model of the software system. Temporal logics such as CTL can then express safety and liveness properties of the system which are verified via model checking. This approach allows for the verification of non-terminating systems, and verification is generally more feasible, since it is based on model checking rather than theorem proving. We shall have more to say about the relative merits of the two approaches in section 8.1.

1.2.2 Game Theory and Social Choice

Computer programs of the kind we discussed represent processes of a very special kind: They are formally specified and they do not contain any interaction. Social processes on the other hand are usually much harder to analyze because they are interactive by nature and because it is often extremely difficult to model them mathematically. The latter problem can be simplified by abstraction and by studying those social processes which are more regulated to begin with, examples being elections, auctions, and the process of obtaining a doctorate degree. Social choice theory [75, 86] studies collective decision making: given the preferences of the individuals of some society, how should the society as a whole choose between the different options? To these considerations game theory [19, 93] adds a strategic component, asking, e.g., how such methods of collective decision making can be manipulated by individuals or groups of individuals.

To model the ability of a group of individuals N, social choice theory has developed the notion of an effectivity function [86, 1]. Given a set of alternatives S between which the individuals must choose, an *effectivity function* $E : \mathcal{P}(N) \to \mathcal{P}(\mathcal{P}(S))$ maps groups of individuals to a collection of subsets of alternatives. We interpret $X \in E(C)$ as "coalition C is effective for X", or as "coalition C is able to achieve an outcome in X". This interpretation is deliberately noncommittal about how exactly "effectivity" is to be interpreted. In this thesis, we will interpret it as having a strategy which is successful no matter what. Hence, $X \in E(C)$ holds iff coalition C has a joint strategy to achieve an outcome in X no matter what the other players do.

Consider the following example from [52]: Angelina has to decide whether she wants to marry Edwin, the (male) judge, or stay single. Edwin and the judge each can similarly decide whether they want to stay single or marry Angelina. If we assume that the three individuals live in a society where nobody can be forced to marry against his/her will, this situation can be modeled using effectivity functions as follows: The set of players is $N = \{a, e, j\}$ and the set of alternatives is $S = \{s_s, s_e, s_j\}$, where s_s denotes the situation where Angelina remains single, s_e where she marries Edwin, and s_j where she marries the judge. Angelina (a) has the right to remain single, so $\{s_s\} \in E(\{a\})$, whereas Edwin can only guarantee that he does not marry Angelina; whether she marries the judge or remains single is not up to him. Consequently, we have $\{s_s, s_j\} \in E(\{e\})$ and there is no proper subset X of $\{s_s, s_j\}$ such that $X \in E(\{e\})$. Analogously for the judge, we have $\{s_s, s_e\} \in E(\{j\})$. Angelina and Edwin together can achieve any situation except the one where Angelina marries the judge (since this alternative would require the judge's consent), and hence $\{s_s\}, \{s_e\} \in E(\{a, e\})$. Again, the situation is similar for the judge: $\{s_s\}, \{s_j\} \in E(\{a, j\})$.

Another way to think about this marriage constellation is in terms of games. We already suggested that Angelina has three strategies, she can decide to remain single, or go for either the judge or Edwin. (Note that we ignore the temporal/dynamic character of more realistic strategies, where Angelina might propose to the judge and, if he does not accept within 3 days, go for Edwin.) The two men on the other hand only have two options each, either to decide to remain single or to go for marriage. The strategic game below pictures this situation: Angelina chooses one of the three tables, Edwin chooses the upper or lower row, and the judge chooses the left or right column. For each such strategy triple we have given the resulting alternative. As an example, consider the soap opera situation where Angelina wants to marry Edwin (middle table), Edwin wants to remain single (upper row), but the judge wants to marry Angelina (right column). The result is that everyone remains single (alternative s_s).

	s	m
s	s_s	s_s
m	s_s	s_s

	s	m
s	s_s	s_s
m	s_e	s_e

	s	m
s	s_s	s_j
m	s_s	s_j

Depending on the kind of social process under consideration, the effectivity function will satisfy certain properties. Note, e.g., that the effectivity function of the example is monotonic, i.e., if a group of individuals is effective for a set X then it is also effective for any superset of X. Another important property which is satisfied is the property of *superadditivity*: If $X_1 \in E(C_1)$ and $X_2 \in E(C_2)$ then $X_1 \cap X_2 \in E(C_1 \cup C_2)$, provided $C_1 \cap C_2 = \emptyset$. Intuitively, if one group of players is able to achieve an outcome in X_1 and another disjoint group is able to achieve an outcome in X_2, then they can join their strategies to achieve an outcome in $X_1 \cap X_2$. This and other properties of effectivity functions have been studied in the literature to characterize certain classes of social processes, and we shall see examples of this in chapter 2.

In this thesis, we shall have nothing to say about the preferences which players might have over the set of alternatives. In other words, we only concern ourselves with what players *can* do, not with what they *should* or *will* do. If one does consider players' preferences, various solution concepts can be investigated. For effectivity functions, a prominent solution concept is the *core*, the set of all undominated alternatives. An alternative s is dominated by a set of alternatives X if there is some coalition C which is effective for X and every member of C prefers every alternative in X to s. It is reasonable to assume that such an alternative s will never be realized, since coalition C will see to it that X is realized instead. An effectivity function is called *stable* in case its core is nonempty no matter what preferences the players have. Conditions under which an effectivity function is stable have been investigated extensively, and it has been shown in [85] that the problem of checking stability of an effectivity function is NP-complete.

1.2.3 Dijkstra meets von Neumann

Programs can be viewed as 1-player games. If the program is deterministic like the *gcd*-algorithm above, the game is of a particularly boring sort since it does not involve any choice points where the player can choose between two or more different possible actions. Nondeterministic programs, however, do involve choices, as the program γ_2 below illustrates:

$$
\begin{array}{l}
x, y\ \texttt{:Nat} \\
\texttt{case}\ x \geq y \longrightarrow y := y + 1 \\
\qquad\ \ y \geq x \longrightarrow x := x + 1 \\
\texttt{end}
\end{array}
$$

In case $x > y$, 1 is added to y, and in case $y > x$, 1 is added to x, but in case $x = y$, a nondeterministic choice is made between the two increments. In case we have a state s with $s(x) = s(y) = 0$, we thus have $sR_{\gamma_2}t_1$ and $sR_{\gamma_2}t_2$, where $t_1(x) = t_2(y) = 0$ and $t_1(y) = t_2(x) = 1$.

$$\begin{tikzcd} & x = 0, y = 0 \arrow[dl] \arrow[dr] & \\ x = 1, y = 0 & & x = 0, y = 1 \end{tikzcd}$$

In words, there is an execution of γ_2 which starts in a state with $x = y = 0$ and ends in a state with $x = 0$, but not all executions have that property. This program thus presents a more interesting 1-player game, in the sense that the player (which we usually assume to be Nature) can choose which transition to make in case $x = y$.

Predicate transformers can thus be given a game-theoretic interpretation: $s \in F_{\gamma_2}(X)$ holds in case at state s, Nature has a strategy in program/game γ_2 for achieving a state in X. Note that this interpretation is in no way dependent on the fact that γ_2 was a 1-player game, for the notion of "strategy" applies to games generally, independent of the number of players. This move from programs to games has also been carried out in the area of temporal logic [3], and we shall have more to say about this in section 4.4.

The two traditions outlined in the previous section apply a similar mathematical model to capture similar ideas. For an effectivity function E, $X \in E(C)$ provided that coalition C has a joint strategy to achieve an outcome in X. For a predicate transformer F and a state s, $s \in F(X)$ provided that Nature has a strategy to achieve an outcome in X. Given this similarity, it may not be surprising that one can discover analogies in the issues investigated in both traditions: The characterization result of social choice theory which isolates the effectivity functions which correspond to strategic games (theorem 2.27) is the multi-agent coalitional analogue of the computer science result which characterizes the predicate transformers which correspond to state transformers (theorem 2.16). Similarly, questions of mechanism design can be related to questions of program synthesis (see chapter 5).

The first difference between the notions of an effectivity function and a predicate transformer is that predicate transformers only capture the abilities of a single player (Nature) whereas effectivity functions model the abilities of many agents and even of groups of agents. The second difference is that effectivity functions are static while predicate transformers are dynamic. An effectivity function describes the social process as a simple one-shot event, the players' abilities do not depend on the current state. Predicate transformers on the other hand link the player's powers to the current state, and these powers may change as the state itself changes. We shall try to take the best from both traditions, the multi-agent perspective of effectivity functions and the dynamic approach of predicate transformers. The result will be a new dynamic multi-agent model of power for which we will develop logical formalisms which can express properties of this type of model.

1.2.4 Praise of FoLLI

The question of why the two traditions mentioned should be brought together can also be viewed as an instance of the more general question regarding the motivation of interdisciplinary research. In the list below, we discuss some general qualities of interdisciplinary study, also referring to some specific results in this thesis which can serve as examples for the case of logic and game theory at hand.

(1.) Answering Old Questions: Probably the most immediate hope one might have is that an open question in one field can be solved using techniques or results from the other field. This hope is also verbalized when the game theorist asks the logician how the logician's research on logic and games will contribute to solving game-theoretic problems. The logician in turn will be tempted to quote John F. Kennedy's "ask not what your country can do for you, ask what you can do for your country."

A concrete example of a result from social choice theory which has been used to solve an open problem in logic will be discussed in section 4.4: Using a characterization result (theorem 2.27) which extends a similar result from the social choice theory literature, we are able to obtain an axiomatization (theorem 3.14) which has been used in [56] to provide a complete axiomatization of Alternating Temporal Logic [3].

(2.) Raising New Questions: More than answering old questions, interdisciplinary research will generate new questions (and answers) in both fields involved. One of the two main topics of this thesis, comparing reasoning about individual ability to reasoning about coalitional ability in various kinds of social processes, seems to exemplify a new type of investigation which could be relevant to social choice theory. Logic has a number of tools and techniques which allow one to carry out such a comparison in terms of expressiveness, complexity, and so on.

An example of a new question for logic will be discussed in section 7.2.2: It turns out that Game Logic can be translated into the propositional modal μ-calculus [76]. This translation has a particular syntactic property, for it uses at most two set variables. This raises the question whether there is a strict finite-variable hierarchy for the μ-calculus, i.e., whether one can show that formulas with n variables can express less than formulas with $n+1$ variables. For the alternation hierarchy, it has been shown that the hierarchy does not collapse [25, 81], whereas the finite-variable hierarchy apparently has not been investigated so far.

(3.) Unification: Obtaining a general framework which describes a wide variety of situations arising in different disciplines is desirable even if no old questions can be answered and no new questions arise. Unification creates links between phenomena and notions which had not been connected before, reducing the number of basic notions or axioms needed to understand (at least a small part of) the world, increasing order in the mind and creating a pleasant feeling in the stomach.

The coalition models introduced in chapter 2 are general enough to capture ability in extensive games with and without simultaneous moves as well as Kripke models. Interpreting the modality $\Diamond\varphi$ as the existence of a strategy in a game leads to a general coalition logic of which both normal and non-normal modal logics are particular instances (section 3.5).

(4.) Diversification: As the dual of unification, where, e.g., two notions are unified into one, diversification refers to the opposite process, providing new perspectives on old friends. While physics seems to strive for *the* theory of everything [57], finding such a theory may only be the beginning of discovering different "equivalent" theories of everything or different interpretations of the theory. In computer science, the various equivalent formalizations of the most central notion, computation, are a paradigm example (see, e.g., [72]).

In the field of logic, modal logic provides an example of a formal system which is open to a wealth of interpretations, and this diversity also testifies to the importance of modal logic. In section 3.5 we shall add yet another view to these multiple perspectives, characterizing normal modal logic as the logic of 1-player games (theorem 3.22) and monotonic non-normal modal logic as the logic of determined 2-player games (theorem 3.24).

1.3 Objectives

Linking the two fields of research discussed above is a natural project to consider, given that the conceptual and mathematical notions involved are very similar. More importantly, however, it will allow us to construct logics which can be used to reason about social software, and to investigate how reasoning about social software differs from reasoning about standard computer software.

1.3.1 Logics for Social Software

In [95], Parikh introduced the term "social software" to refer to the project of analyzing social procedures and processes using the formal methods of computer science. One of his example concerns the well-known problem of dividing a cake fairly among a number of people. For two people, the algorithm "I cut, you choose" is a well-known method to ensure that both people can guarantee themselves at least half of the cake according to their own perception. For more than 2 people, there are extensions of this cut-and-choose algorithm which can yield *fair* solutions for all the participants [27]. An algorithm which does offer everyone a strategy to guarantee himself a fair share of the cake can be considered *correct*, and Parikh calls for the development of logical tools to be able to prove the correctness and efficiency of such an algorithm.

A further example of a social process amenable to formal analysis are voting procedures as studied in social choice theory [84]. The number of alternatives, the order of voting and the different electoral bodies involved can all influence the final outcome. Proving the correctness of a voting procedure would mean to show that it satisfies certain desirable properties (e.g., that there is no dictator, i.e., no individual has the power to determine the outcome by himself), and establishing that it is efficient would entail a proof that there is no simpler procedure with the same properties.

The first thesis of this study is that the logics developed and studied here, Coalition Logic and Game Logic, provide the means to analyze social processes like the ones given. Chapter 5 will provide arguments for this claim by applying Coalition Logic to a number of examples. Since these examples lie well in the domain of game theory (and social choice theory), the question arises how the perspective of social software differs from the perspective of game theory. On the one hand, the social software approach strives for a formal axiomatic theory of social processes which is very explicit. As a consequence, arguments can be formulated in a logical language which lends itself to automated verification, the advantage being a greater degree of confidence in as well as the possible automation of game-theoretic argumentation. On the other hand, the social software perspective provides new theoretical questions and insights about social processes which broaden the scope of game theory. These questions can be grouped into two comparisons which we shall discuss subsequently.

1.3.2 Programs vs. Games

The first comparison we shall engage in relates programs to games. As we have seen, programs can be viewed as 1-player games, so the question arises in what respect reasoning about 1-player games differs from reasoning about 2-player games. Does the addition of a second player make an essential difference, and if so, in what respect? To use a marginally related example from a different domain, it appears that the addition of a second character complicates the automatic generation of humorous film sequences significantly [88].

One way in which we will investigate this question is using Game Logic. This logic uses the formula $\langle\gamma\rangle\varphi$ to express that Player 1 has a strategy to achieve φ in game γ. The game γ here is an expression denoting a complex game which is constructed by means of a number of operations, like the programming operation `if...then...else...` we have seen earlier. As an example, consider $\gamma_1 = (a \cup b); (c \cap d)$, denoting the game where first player 1 chooses between doing a or b and then player 2 chooses between doing c or d. To compare games to programs, we compare Game Logic with its program fragment, the formulas containing only those modalities $\langle\gamma\rangle\varphi$ where γ is a 1-player game, i.e., where player 1 makes all the choices. The game γ_1 is not in the program fragment, but, e.g., $\gamma_2 = (a \cup b); (c \cup d)$ is, where the second choice is also made by player 1. The program fragment can then be compared to full Game Logic in terms of expressive power (are there properties which cannot be expressed using only programs?), axiomatization (what are the basic principles of reasoning about games and programs?) and complexity (is reasoning about games more complex than reasoning about programs?).

Besides this syntactic component of the comparison between programs and games, there is also a semantic component having to do with the basic building blocks from which complex games like γ_1 are constructed. Two cases can be investigated: First, one can assume that all interaction is introduced through the game operations, and that the atomic games are simple 1-player games where the same player makes all the choices (chapter 7). Second, one can allow interaction in the atomic games as well, yielding a more general system which is studied first in chapter 6.

1.3.3 Individuals vs. Coalitions

Once we have moved from programs to games with 2 or more players, we can reason about what the different players can achieve in the game. But there is still another issue here, namely how reasoning about these individual agents may differ from reasoning about groups of agents. It may seem for instance that reasoning about what single agents can bring about in a complex multi-agent process should be simpler than reasoning about what groups of agents can bring about.

In chapters 3 and 4 we will look at this issue using the semantic models developed in chapter 2. The coalition models developed there will allow us to capture various kinds of multi-agent processes or games (including 1-player games), and for each of these games we will investigate the differences between reasoning about individuals and reasoning about coalitions. We will do this first using Coalition Logic (chapter 3), a logical language which contains expressions of the form $[C]\varphi$ where C is a group of agents. The formula expresses that coalition C is able to achieve φ in one move, i.e., C has a strategy for φ. To compare individuals to coalitions, we compare the full language of Coalition Logic with its individual fragment, the restricted language which can only talk about single-agent coalitions, i.e., which only contains formulas $[i]\varphi$ expressing that agent i has a φ-strategy.

While basic Coalition Logic only contains modalities to talk about what groups of agents can achieve in one move of the game, chapter 4 introduces Extended Coalition Logic which can also express long-term ability. The formula $[C^*]\varphi$ expresses that coalition C has a strategy to bring about φ at some point in the future. As with basic Coalition Logic, one can compare individuals to coalitions in this richer language.

1.4 Overview

In section 1.2.1, we briefly discussed 2 approaches to program verification, an internal approach based on temporal logic and an external approach based on programming logics like PDL. Both of these approaches shall be lifted from programs to games, and hence this thesis naturally falls into two parts, chapters 3 to 5 which focus on the internal approach using the logical framework of Coalition Logic, and chapters 6 and 7 which deal with the external approach using the framework of Game Logic. Chapter 2 will introduce a semantic framework which unifies both the internal and the external approach. We shall now proceed to describe the content of the different chapters in some more detail.

Chapter 2 develops a very general dynamic ability model based on the notion of an effectivity function. This model serves to capture group ability in programs and extensive games with and without simultaneous moves among 2 or more players. Quite some time will be spent on characterizing the precise class of ability models which corresponds to each of these processes. We illustrate how these ability models are open to two different interpretations, either as internal descriptions of a single game or as external descriptions of a collection of games. We also consider models of individual ability, asking again which conditions on individual ability characterize ability in various classes of games, and in which situations individual ability completely determines group ability. Finally, we extend the notion of bisimulation to serve as an equivalence notion for the ability models defined.

Chapter 3 introduces Coalition Logic, a modal logic which contains formulas $[C]\varphi$ expressing that the group of agents C is able to achieve in one move an outcome where φ is true. The formulas are evaluated over the ability models defined in chapter 2 and interpreted as an internal description of a social process. To compare reasoning about individuals to reasoning about coalitions, we isolate the individual fragment of Coalition Logic and compare some of its properties (complexity and expressiveness) to those of full Coalition Logic. Finally, we show how normal and non-normal modal logics can be viewed as instances of this more general Coalition Logic.

Chapter 4 extends basic coalition logic with an additional modality $[C^*]\varphi$ asserting that coalition C can achieve φ at some point in the future. This additional modality opens the door to a variety of different applications of Coalition Logic in the analysis of social processes. We also consider two meta-theoretic questions concerning the complexity of Extended Coalition Logic and its expressive power. Related formalisms such as Alternating Temporal Logic and the multi-agent logics developed in distributed artificial intelligence are compared to (extended) Coalition Logic.

Chapter 5 applies Coalition Logic and Extended Coalition Logic to problems mainly from the area of social choice theory. Put succinctly, we show how Coalition Logic can be used to guarantee personal liberties, avoid excessive or obscure legislative procedures and reduce telephone costs.

Chapter 6 changes from the internal to the external view of games. Ability models are viewed as a collection of interlinked games and the formal language of Game Logic is used to describe what players can bring about in complex games. Here, the modal expression $\langle\gamma\rangle\varphi$ states the existence of a strategy for player 1 in game γ which achieves φ. We define a generalized version of the μ-calculus of which Game Logic is a fragment. This perspective allows us to locate an interesting difference between programs and games which will lead to different complexity results for model checking. Finally, we also consider the complexity of the satisfiability problem.

Chapter 7 continues the study of Game Logic, now focusing on Game Logic when interpreted over the restricted class of Kripke models. Over these models, Game Logic can be compared to Propositional Dynamic Logic (PDL) on the one hand and the standard modal μ-calculus on the other hand in terms of expressiveness and complexity. In particular the comparison with PDL will yield some interesting differences between programs and games, one of them being that games allow for two different kinds of iteration. Using bisimulation, we also obtain a characterization of the set of game operations needed to construct all first-order definable games. This result can then be compared to an analogous result obtained for programs.

Chapter 8 summarizes the main results of this thesis, addressing again the relationship between Game Logic and Coalition Logic, and also discussing the algebraic counterpart of Game Logic and its application in the development of social software.

Since chapters 4 and 6 make use of fixpoint constructions to define long-term ability and iteration, we have summarized some (mostly well-known) results concerning fixpoints in appendix A which will be appealed to in these chapters.

Chapter 2

Multi-Agent Models of Power

We introduce the notion of an individual effectivity function to model what individual agents can bring about in a situation of interaction. Such an individual effectivity function maps each agent to the sets of outcomes which he is able to achieve. We show that this model is general enough to cover ability in various kinds of games as well as programs: Results are proved which characterize the classes of effectivity functions which correspond to ability in strategic games and 1-player games.

After emphasizing individual ability in section 2.2, groups and their ability are the focus of the third section, where individual effectivity functions are generalized to coalitional effectivity functions. Analogous to the previous section, we look for conditions on group effectivity which can guarantee that it represents group effectivity in various kinds of games. Furthermore, we ask under what circumstances knowledge of a group's ability is more informative than knowledge of the ability of its individual members. In other words, for the different classes of effectivity functions introduced, we investigate whether individual ability completely determines group ability.

In the fourth section, we switch from the static model of effectivity functions to the dynamic model of effectivity frames. These contain a set of possible worlds each of which is associated with a coalitional or individual effectivity function where the outcomes are states of the world again. Suitable restrictions of these effectivity frames will form the basis of both Coalition Logic and Game Logic, to be discussed in chapters 3 and 6, respectively.

In section 2.5 finally, we introduce the notion of bisimulation for dynamic effectivity models. Bisimulation is an equivalence notion for models which will play an important role in later chapters of this thesis.

2.1 Types of Interaction

The dynamic models of interaction which we shall define in section 2.4 will associate particular types of interaction to each state. One of the most general types of interaction is a strategic game where all players simultaneously decide on a strategy, and the strategies chosen together determine the new state of the system. The system as a whole then corresponds to an extensive game with simultaneous moves [93], or an extensive game of almost perfect information [113].

Alternatively, each state knows a particular player, the local dictator, who decides the new state. If the local dictator is the same at every state, we are dealing with a 1-player extensive game, otherwise with a multi-player extensive game of perfect information.

Finally, we will discuss empty games where no interaction happens but instead the players are labeled as winners and losers. In the dynamic models to be defined in section 2.4, these empty games will be used to model terminal states of an extensive game.

2.1.1 Strategic Games

Depending on the situation in which different agents interact, their abilities will be related in a particular way. One of the most general models for situations of strategic interaction is that of a strategic game. Because of its generality, strategic games form the standard model in game theory. In a strategic game, the different players choose one of their available alternative actions/strategies, and taken together, these actions determine the outcome of the game.

▶ Definition 2.1 (Strategic Game). A *strategic game* $G = (N, \{\Sigma_i | i \in N\}, o, S)$ consists of a nonempty finite set of agents N, a nonempty set of strategies or actions Σ_i for every player $i \in N$, the nonempty set of states S and an outcome function $o : \Pi_{i \in N}\Sigma_i \to S$ which associates with every tuple of strategies of the players (strategy profile) an outcome state in S.

In game theory, strategic games also come equipped with a preference relation $\succeq_i \subseteq S \times S$ for every player $i \in N$ which indicates which outcomes a player prefers. Strictly speaking, our strategic games are only *game forms* which can be turned into a game by adding these preference relations.

For notational convenience, let $\sigma_C := (\sigma_i)_{i \in C}$ denote the strategy tuple for coalition $C \subseteq N$ which consists of player i choosing strategy $\sigma_i \in \Sigma_i$. Then given two strategy tuples σ_C and $\sigma_{\overline{C}}$ (where $\overline{C} := N \setminus C$), $o(\sigma_C, \sigma_{\overline{C}})$ denotes the outcome state associated with the strategy profile induced by σ_C and $\sigma_{\overline{C}}$. We shall also write $-i$ for $N \setminus \{i\}$.

Figure 2.1 below provides an example of strategic game among three players in the usual matrix depiction. Unless noted otherwise, we will assume that player

1 chooses the row, player 2 the column, and the third player chooses between the left and the right table.

	l	m	r
l	s_1	s_2	s_1
r	s_2	s_1	s_3

	l	m	r
l	s_3	s_2	s_1
r	s_2	s_3	s_3

Figure 2.1: A strategic game for three players.

In this game, for every joint strategy of players 1 and 3, player 2 has a strategy which yields outcome s_2. Note, however, that this strategy depends on the strategies chosen by players 1 and 3, i.e., player 2 has no strategy which will guarantee outcome s_2 independent of the strategies of players 1 and 3. The coalition consisting of players 1 and 2 on the other hand does have a joint strategy (r, l) which guarantees s_2 independent of player 3's strategy. In section 2.2, we shall introduce terminology to distinguish these different kinds of effectivity.

2.1.2 Dictatorships

On the one hand, it seems that not much needs to be said about 1-player games: If the set of players N is a singleton, the strategic games will only contain choices for this one player who can freely choose an outcome. On the other hand, there may also be multi-player games which are essentially 1-player games. Consider, e.g., the following 2-player game:

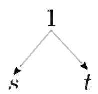
	l	r
l	s	s
m	s	t
r	t	t

Clearly, it is player 1 (the row player) who has complete freedom in choosing the outcome of the game, and hence, although formally a 2-player game, this game is in essence a 1-player game. The notion of dictatorship makes precise what we mean by "in essence".

▶ Definition 2.2 (Dictatorship). A strategic game $G = (N, \{\Sigma_i | i \in N\}, o, S)$ is a *d-dictatorship* iff $\forall s \in ran(o) \exists \sigma_d \forall \sigma_{-d} \ o(\sigma) = s$.

In such a dictatorship, there is an individual d (the dictator) whose choices completely determine the outcome, independent of what the others do. Note that in case there is more than one dictator, the outcome function is constant (i.e., $\exists s \forall \sigma \ o(\sigma) = s$) and hence every player is a dictator. Furthermore, every 1-player strategic game is trivially a dictatorship. More generally, note that any strategic game can also be viewed as a dictatorship of the masses: for every outcome state, the set of all players has a strategy to achieve it.

For our purposes, we lose nothing by modeling a dictatorship by a 1-player 1-move game where the dictator $d \in N$ can choose the successor state independent of what actions the other players choose to take. The example above would thus be modeled as follows:



In such a game, 1 is effective for any possible outcome, whereas all other players cannot guarantee anything more than the set of all outcomes among which 1 chooses. Note, however, that it is not the case that the dictator can necessarily achieve any outcome from S, for it may be the case that $ran(o) \neq S$. Hence there may be states of the world which even a dictator cannot achieve.

2.1.3 Empty Games

Later in this chapter we shall combine static models of ability in order to obtain a dynamic model. For this we need the following notion.

▶ DEFINITION 2.3 (EMPTY GAME). An *empty strategic game* is any $G \subsetneq N$, i.e., any subset of players other than N itself.

Depending on the context, we shall also refer to an empty strategic game as an *empty dictatorship*. Note that, formally speaking, an empty strategic game is not a strategic game, but the terminology chosen will simplify the formulation of some results later.

Game-theoretically, an empty strategic game simply models a payoff vector where we interpret G as the set of players who win the game and, e.g., obtain payoff 1 whereas the others obtain payoff -1. In the extensive games we shall discuss in section 2.4, empty games shall model endpoints of the game, where, for example, an endpoint

$$\{1, 4, 5\}$$

in a 5-player extensive game will denote the payoff vector $(1, -1, -1, 1, 1)$.

To use a different metaphor, an empty strategic game can be viewed as an interaction failure. Depending on the situation, there may be many reasons for such a failure: Two agents may attempt to communicate by simultaneously performing `send(a)` and `read(b)` resulting in deadlock and a failure of communication (process algebra), not yielding any new state.

A further interpretation can be given in terms of contracts (as is done in [7]): The rules of a game formalize the contractual obligations of the players. One or more agents may violate these rules, thereby resulting in an immediate loss for them and a win for the other players. For example, an agent may commit himself to the truth of a certain formula which fails to be true after all, in which case he has violated his contractual obligation resulting in an immediate loss for that player (see the test game in Game Logic, chapter 6).

Note that we assume that someone has to lose, i.e., there are no empty games where all players win. This rather pessimistic restriction carries some intuitive appeal: If all players play according to the rules, the result will be a particular outcome state rather than a win for all of them. As shown in [98], for the purposes of basic Coalition Logic we can easily do without empty games, and except for section 3.5, the results of chapter 3 do not depend on having empty games. But while the presence of empty games makes the proofs of some results a bit more complicated, they are needed in order to provide a unified semantic model for both the internal (Coalition Logic) and the external (Game Logic) approach to games.

2.2 A Static Model of Individual Ability

Throughout this thesis, we assume that a nonempty finite set N of agents or players is given, as well as a nonempty set of states S.

▶ DEFINITION 2.4 (INDIVIDUAL EFFECTIVITY FUNCTION). An *individual effectivity function* is any function $E : N \to \mathcal{P}(\mathcal{P}(S))$ which is (outcome-)monotonic: For every individual $i \in N$, $X \in E(i)$ implies $Y \in E(i)$ whenever $X \subseteq Y \subseteq S$.

The function E associates to every player the sets of outcomes for which he is effective. In most situations we will consider, "effectivity" can be interpreted as follows: For a player $i \in N$, $X \in E(i)$ will hold iff player i has a strategy for bringing about an outcome in X. Given this interpretation, outcome-monotonicity is a natural requirement: if a player has a strategy to bring about an outcome in X, that strategy will also bring about an outcome in any superset of X.

In most circumstances, we will want to impose certain additional restrictions on individual effectivity functions besides monotonicity. For instance, if $N = \{1, 2\}$ and $X \in E(1)$ we would want $\overline{X} \notin E(2)$ to be the case, for otherwise player 1 would have a strategy to achieve an outcome in X and player 2 would have a strategy to achieve an outcome in $\overline{X}$, so if both of them utilize their strategies, the resulting outcome would have to be both in X and not in X. We will consider various conditions on effectivity functions below. The precise conditions which one wants to assume will depend on the nature of the interaction one wants to study.

2.2.1 Individual Ability in Empty Games

Recall that we interpreted an empty strategic game $G \subsetneq N$ as the set of players who win the game. In this situation, we will stipulate that a player who loses will not be effective for anything, not even for the set of all states. If he is a winner on the other hand, we stipulate that he can bring about any set of states, even the empty set. Recall that in a regular strategic game, every player can bring about something (e.g., the set of all states) but not everything (e.g., the empty set). Thus, it makes sense to use the two extreme cases to identify empty games, one for winners and the other for losers.

▶ DEFINITION 2.5 (α-EFFECTIVITY IN EMPTY GAMES). Given an empty game $G \subsetneq N$, its individual *α-effectivity function* $E^{\alpha}_G : N \to \mathcal{P}(\mathcal{P}(S))$ is defined as follows: $X \in E^{\alpha}_G(i)$ iff $i \in G$.

In words, a player is effective for everything if he is a winner; otherwise, he is effective for nothing. Hence, in empty games we can simply talk about effective and ineffective players.

We say that an individual effectivity function $E : N \to \mathcal{P}(\mathcal{P}(S))$ *α-corresponds* to an empty strategic game G iff $E = E^{\alpha}_G$. If we are given some effectivity function E, we may want to know under what conditions E is the α-effectivity function of some empty game. The following result characterizes the properties of individual effectivity in empty strategic games.

▶ THEOREM 2.6. An individual effectivity function E α-corresponds to an empty strategic game iff (1) $\forall i\, \forall X, Y : X{\in}E(i) \Rightarrow Y{\in}E(i)$, and (2) $\exists i\, \forall X : X{\notin}E(i)$.

PROOF. For any empty strategic game G, E^{α}_G satisfies the two conditions; conversely, if E satisfies the two conditions then for $G = \{i \in N | \exists X : X \in E(i)\}$ we have $E = E^{\alpha}_G$. ■

2.2.2 Individual Ability in Strategic Games

Given a game G, a player $i \in N$ will be *α-effective* for a set $X \subseteq S$ iff he has a strategy which will result in an outcome in X no matter what strategies the other players choose.

▶ DEFINITION 2.7 (α-EFFECTIVITY IN STRATEGIC GAMES). For a nonempty strategic game G, its *individual α-effectivity function* $E^{\alpha}_G : N \to \mathcal{P}(\mathcal{P}(S))$ is defined as follows: $X \in E^{\alpha}_G(i)$ iff $\exists \sigma_i \forall \sigma_{-i}\;\; o(\sigma_i, \sigma_{-i}) \in X$.

A related weaker notion of effectivity is β-effectivity. For a given game G, the *β-effectivity function* is defined as $X \in E^{\beta}_G(i)$ iff $\forall \sigma_{-i} \exists \sigma_i\;\; o(\sigma_i, \sigma_{-i}) \in X$. Clearly, β-effectivity is weaker than α-effectivity, as the example of figure 2.1

has illustrated, for in that game player 2 is β-effective for $\{s_2\}$ without being α-effective for it.

As for empty strategic games, we say that an individual effectivity function $E : N \to \mathcal{P}(\mathcal{P}(S))$ *α-corresponds* to a nonempty strategic game G iff $E = E^{\alpha}_{G}$. The question to be examined now is which effectivity functions α-correspond to some strategic game. An answer to this question will provide a complete characterization of the properties of individual ability in strategic games. Note that the following result is limited to situations where there are at least 2 players; we shall consider the case where $|N| = 1$ later.

▶ DEFINITION 2.8 (STRONG AMUSEMENT). An individual effectivity function $E : N \to \mathcal{P}(\mathcal{P}(S))$ is *strongly amusing* iff (1) $\bigcap_{i \in N} X_i \neq \emptyset$ whenever $\forall i \in N : X_i \in E(i)$, and (2) $\forall i \in N : E(i) \neq \emptyset$.

▶ THEOREM 2.9. If $|N| > 1$, an individual effectivity function E α-corresponds to a nonempty strategic game iff E is strongly amusing.

PROOF. One can easily check that the α-effectivity function of any strategic game is strongly amusing. As for the other direction, let E be a strongly amusing effectivity function. We shall construct a game G such that $E = E^{\alpha}_{G}$. To simplify our definitions, assume that $N = \{1, \ldots, n\}$.

Let $H = \{h : \mathcal{P}(S) \backslash \{\emptyset\} \to S \mid h(X) \in X\}$ contain all functions which pick an element from a nonempty set. We then define the strategic game $G = (N, \{\Sigma_i | i \in N\}, o, S)$ as follows: Let

$$\Sigma_i = E(i) \times N \times H \text{ and } o(\sigma_N) = h_{i_0}(\bigcap_{i \in N} e_i),$$

where $\sigma_N = (e_i, t_i, h_i)_{i \in N}$ is a strategy profile and $i_0 = ((t_1 + \cdots + t_n) \ mod \ n) + 1$. The idea here is that a strategy specifies the set a player chooses to force (e_i) and a function which chooses an outcome from every possible nonempty set (h_i). Given a strategy profile, the outcome of the game will be in the intersection of the sets forced, and the precise outcome which is forced is then determined by i_0 which indicates the player who has the power to determine the outcome. Note that the condition we put on E guarantees that $o(\sigma_N)$ is well-defined.

To check that $E^{\alpha}_{G} = E$, suppose that $X \in E(i)$. Then player i can play a strategy (e_i, t_i, h_i) with $e_i = X$ which will guarantee the outcome to lie in X, and hence $X \in E^{\alpha}_{G}(i)$. Conversely, suppose that $X \notin E(i)$ and consider any strategy (e_i, t_i, h_i) of player i, where there is some $c \in e_i \cap \overline{X}$. Since $|N| > 1$, for all players $j \neq i$ we can define (e_j, t_j, h_j) so that $e_j = S$, $h_j(e_i) = c$ and the t_j are chosen so that $((t_1 + \cdots + t_n) \ mod \ n) + 1 \neq i$. Then $o(\sigma_N) = c$, and hence $X \notin E^{\alpha}_{G}(i)$. ■

If we also want to allow for empty games, we get a corollary to the preceding result using the notion of weak amusement.

▶ DEFINITION 2.10 (WEAK AMUSEMENT). An individual effectivity function $E : N \to \mathcal{P}(\mathcal{P}(S))$ is *weakly amusing* iff (1) $\bigcap_{i \in N} X_i \neq \emptyset$ whenever $\forall i \in N : X_i \in E(i)$, and (2) if there is some $j \in N$ such that $E(j) = \emptyset$, then for all $i \in N$, if $X \in E(i)$ and $Y \subseteq X$ then $Y \in E(i)$.

Note that if E α-corresponds to an empty game, condition (1) of the definition is vacuous since by theorem 2.6 it cannot be the case that $\forall i \in N : X_i \in E(i)$.

▶ COROLLARY 2.11. If $|N| > 1$, an individual effectivity function E α-corresponds to a possibly empty strategic game iff it is weakly amusing.

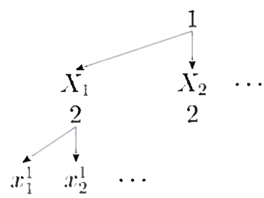
PROOF. In case there is some $j \in N$ such that $E(j) = \emptyset$, let G be the empty strategic game defined as $G = \{i \in N | S \in E(i)\}$. One can easily check that for all $i \in N$, $E(i) = E^{\alpha}_{G}(i)$. ■

A special case which shall become important later are determined 2-player games. The notion can be defined uniformly for empty as well as nonempty strategic games. We call a possibly empty 2-player game G *determined* iff for all X, $X \notin E^{\alpha}_{G}(1)$ implies $\overline{X} \in E^{\alpha}_{G}(2)$, where we assume $N = \{1, 2\}$. Note that in case G is empty, determinacy ensures that there is exactly one winner. As it turns out, possibly empty determined 2-player games allow for a rather elegant characterization result:

▶ THEOREM 2.12. For $N = \{1, 2\}$, an individual effectivity function E α-corresponds to a possibly empty determined strategic game iff for all X, $X \in E(1)$ iff $\overline{X} \notin E(2)$.

PROOF. In case $E(1) = \emptyset$, G is the empty game where player 1 loses, i.e., $G = \{2\}$. In case $\emptyset \in E(1)$, player 1 wins, i.e., $G = \{1\}$. Otherwise, we define $G = (\{1, 2\}, \{\Sigma_1, \Sigma_2\}, o, S)$ as follows: $\Sigma_1 = \{X \subseteq S | X \in E(1)\}$, $\Sigma_2 = \{f : \Sigma_1 \to S | f(X) \in X\}$ and $o(X, f) = f(X)$. In words, player 1 chooses any set he is able to force whereas player 2 chooses a function selecting an element from every nonempty subset of S. it is easily checked that $E^{\alpha}_{G} = E$.

Note that G is in fact the strategic normal form of the following extensive game form: First, player 1 chooses a set $X \in E(1)$, and after that, player 2 chooses an element $x \in X$. The x chosen will be the outcome state of the game.



■

Note that the proof of the theorem demonstrates that for $N = \{1, 2\}$, any individual effectivity function satisfying $X \in E(1)$ iff $\overline{X} \notin E(2)$ α-corresponds to a determined strategic game of special kind, namely to the strategic normal form of a a 2-move extensive game of perfect information.

As a consequence of this theorem, individual effectivity in a determined 2-player game can simply be modeled by a superset-closed set $E \subseteq \mathcal{P}(S)$ which we interpret as the ability of player 1. As the theorem states, every such E can be linked to a possibly empty determined strategic game.

▶ COROLLARY 2.13. Every individual effectivity function $E_1 : \{1\} \to \mathcal{P}(\mathcal{P}(S))$ can be extended to an individual effectivity function $E_2 : \{1, 2\} \to \mathcal{P}(\mathcal{P}(S))$ which α-corresponds to a possibly empty determined strategic game.

2.2.3 Individual Ability in Dictatorships

As with strategic games, we would like to characterize the class of individual effectivity functions which α-correspond to a dictatorship. The crucial property needed here is a distribution property:

▶ DEFINITION 2.14 (DISJUNCTIVITY). A set $E \subseteq \mathcal{P}(S)$ is *disjunctive* iff for all $V \subseteq \mathcal{P}(S)$ we have $\bigcup_{X \in V} X \in E$ iff there is some $X \in V$ such that $X \in E$. (In case $V = \emptyset$, we define $\bigcup_{X \in V} X$ to be $\emptyset$.)

Note that disjunctivity implies monotonicity, and that if E is disjunctive then $\emptyset \notin E$.

▶ THEOREM 2.15. An individual effectivity function E α-corresponds to a possibly empty d-dictatorship iff $E(d)$ is disjunctive and for all $i \neq d$, $X \in E(i)$ iff $\overline{X} \notin E(d)$.

PROOF. In case $E(d) = \emptyset$, we define G to be the empty strategic game where player d loses and everyone else wins. Otherwise, define $G = (N, \{\Sigma_i | i \in N\}, o, S)$ where $\Sigma_d = \{s \in S | \{s\} \in E(d)\}$, and for all $i \neq d$, Σ_i is a singleton. The outcome of the game is defined by setting $o(\sigma) = \sigma_d$. Then for all $i \in N$, $E(i) = E^\alpha_G(i)$.

Consider first the case where $i = d$. Clearly, $E^\alpha_G(d) \subseteq E(d)$ by monotonicity. For the opposite inclusion, we make use of disjunctivity: If $X \in E(d)$ we know that $X \neq \emptyset$ and that $X = \bigcup_{s \in X}\{s\}$. Then for some $s \in X$ we have $\{s\} \in E(d)$ and hence the player can choose outcome s in G and hence also bring about X. In case $i \neq d$,

$$X \in E(i) \text{ iff } \overline{X} \notin E(d) \text{ iff } \overline{X} \notin E^\alpha_G(d) \text{ iff } X \in E^\alpha_G(i).$$

■

Note that in case $|N| = 1$, every strategic game is automatically a dictatorship, and hence as a corollary to the previous theorem, we obtain the missing case of theorem 2.11. The following result has been well-known in the literature on program semantics, stating that the condition of disjunctivity is sufficient for a predicate transformer to correspond to a state transformer [5, 99].

▶ COROLLARY 2.16. If $N = \{1\}$, an individual effectivity function E α-corresponds to a possibly empty strategic game (= dictatorship) iff $E(1)$ is disjunctive.

2.3 A Static Model of Group Ability

▶ DEFINITION 2.17 (COALITIONAL EFFECTIVITY FUNCTION). A *coalitional effectivity function* is any function $E : \mathcal{P}(N) \to \mathcal{P}(\mathcal{P}(S))$ which is monotonic: For every coalition $C \subseteq N$, $X \in E(C)$ implies $Y \in E(C)$ whenever $X \subseteq Y \subseteq S$.

The function E associates to every group of players the sets of outcomes for which the group is effective; as before, we usually interpret $X \in E(C)$ as "the players in C have a joint strategy for bringing about an outcome in X". Whenever we shall speak of effectivity functions in the future, it shall be clear from the context whether we refer to individual or coalitional effectivity functions.

As with individual effectivity functions, in most situations we will want coalitional effectivity functions to satisfy some additional properties besides monotonicity. The following ones will play a central role later on:

Coalition-Monotonicity: In many circumstances one will want to assume that a group which becomes larger has possibly more power but certainly not less. In that case, E is *coalition-monotonic*, i.e., for $C \subseteq C' \subseteq N$, $E(C) \subseteq E(C')$.

Regularity: As a basic consistency requirement, we usually want to exclude cases where complementary coalitions are effective for complementary things, for in that case, both coalitions could use their power and end up in an inconsistent situation. The notion of regularity captures this concern: E is *C-regular* if for all X, if $X \in E(C)$ then $\overline{X} \notin E(\overline{C})$. E is *regular* iff for all coalitions C it is C-regular.

Maximality: As a converse to regularity, call E *C-maximal* if for all X, if $\overline{X} \notin E(\overline{C})$ then $X \in E(C)$. E is *maximal* iff for all coalitions C it is C-maximal.

Superadditivity: The most interesting principle governs the formation of coalitions. It states that coalitions can combine their strategies to (possibly) achieve more: E is *superadditive* if for all X_1, X_2, C_1, C_2 such that $C_1 \cap C_2 = \emptyset$, $X_1 \in E(C_1)$ and $X_2 \in E(C_2)$ imply that $X_1 \cap X_2 \in E(C_1 \cup C_2)$.

As natural as these conditions may seem, one can imagine situations in which they are violated. For an example violating coalition-monotonicity, think of RoboCup, where a team of robots playing soccer may lose its ability to win if the team is joined by a completely malfunctioning robot which always blocks the goal of the opposing team. We have seen earlier that in the case of 2-player games, maximality expresses some kind of determinacy condition which is violated by many games, for example the well-known matching pennies game where player 1 wins in case both pennies come up with the same face, and player 2 wins in case the faces of the two pennies differ.

	H	T
H	win_1	win_2
T	win_2	win_1

Neither of the two players has a winning strategy, for player 1 is not effective for win_1 and player 2 is not effective for win_2, and hence maximality is violated. As a counterpart to maximality, regularity expresses that a game is strictly competitive (zero-sum): If player 1 has a winning strategy (she is effective for win_1) then player 2 cannot have a winning strategy as well (2 is not effective for win_2). For an example violating both regularity and superadditivity, we refer the reader to section 5.1. Note, however, that while the examples given are prima facie violations of the respective conditions, a different modeling of the example may very well avoid the violation.

▶ **Definition 2.18** (Dual Effectivity Function). Given a coalitional effectivity function $E : \mathcal{P}(N) \to \mathcal{P}(\mathcal{P}(S))$, its *dual* effectivity function $\tilde{E}$ is defined as follows: $X \in \tilde{E}(C)$ iff $\overline{X} \notin E(\overline{C})$.

Using the notion of duality, we can elegantly rephrase regularity and maximality as:

$$\begin{array}{ll} E(C) \subseteq \tilde{E}(C) & \text{(regularity)} \\ \tilde{E}(C) \subseteq E(C) & \text{(maximality)} \end{array}$$

In some of the definitions used in the completeness and complexity arguments of the following chapters, we will define effectivity functions E with two separate clauses, one for all coalitions $C \neq N$, and another one for $C = N$. Verifying that an effectivity function defined in such a way is superadditive is facilitated by the following lemma which is extremely simple yet often useful.

▶ **Lemma 2.19.** An effectivity function which is regular, N-maximal and superadditive for $C_1 \cup C_2 \neq N$ is superadditive.

Proof. Assume $X_1 \in E(C_1)$ and $X_2 \in E(C_2)$ where $C_1 \cap C_2 = \emptyset$, $C_1 \cup C_2 = N$ and hence $C_1 = \overline{C_2}$. We assume w.l.o.g. that $C_2 \neq N$. Assume by reductio that $X_1 \cap X_2 \notin E(C_1 \cup C_2)$. By N-maximality, $\overline{X_1 \cap X_2} \in E(\emptyset)$ and by superadditivity (since $\emptyset \cup C_2 \neq N$), $\overline{X_1} \cap X_2 \in E(C_2)$. By monotonicity, $\overline{X_1} \in E(C_2)$ and by regularity, $X_1 \notin E(C_1)$, a contradiction. ■

2.3.1 Group Ability in Empty Games

In extending α-effectivity from individuals to coalitions, we will call a coalition α-effective for some set if all of its members are winners.

► DEFINITION 2.20 (α-EFFECTIVITY IN EMPTY GAMES). For an empty game $G \subsetneq N$, its coalitional *α-effectivity function* $E^{\alpha}_{G} : N \to \mathcal{P}(\mathcal{P}(S))$ is defined as follows: $X \in E^{\alpha}_{G}(C)$ iff $C \subseteq G$.

The notion of α-correspondence can easily be extended to the coalitional case: We say that a coalitional effectivity function $E : \mathcal{P}(N) \to \mathcal{P}(\mathcal{P}(S))$ *α-corresponds* to an empty strategic game G iff $E = E^{\alpha}_{G}$.

► DEFINITION 2.21 (TERMINAL EFFECTIVITY FUNCTION). An effectivity function $E : \mathcal{P}(N) \to \mathcal{P}(\mathcal{P}(S))$ is *terminal* iff (1) it is superadditive, (2) $C \subseteq D$ implies $E(D) \subseteq E(C)$, (3) $\forall X, Y : X \in E(C)$ implies $Y \in E(C)$, (4) $\emptyset \notin E(N)$, and (5) $S \in E(\emptyset)$.

► THEOREM 2.22. A coalitional effectivity function E α-corresponds to an empty strategic game iff E is terminal.

PROOF. The 5 terminality conditions are easily verified for α-effectivity functions of empty games. Conversely, if E is a terminal effectivity function, let $G = \{i \in N | \emptyset \in E(\{i\})\}$. Note that $G \neq N$, for otherwise we would have $\emptyset \in E(N)$ by superadditivity. To show that $E = E^{\alpha}_{G}$, one shows that $X \in E(C)$ iff $C \subseteq G$. From left to right, if $X \in E(C)$ then for all $i \in C$ we have $X \in E(\{i\})$ and hence $\emptyset \in E(\{i\})$, so $C \subseteq G$. Conversely, if $C \subseteq G$, $\forall i \in C$ we have $\emptyset \in E(\{i\})$. In case $C = \emptyset$, the conclusion easily follows. Otherwise, superadditivity gives us $\emptyset \in E(C)$ and hence by monotonicity $X \in E(C)$. ■

Note that terminal effectivity functions are downward monotonic when it comes to coalitions, i.e., any subset of an effective coalition is also effective. An alternative would have been to call a coalition effective in case at least one of its members is a winner, yielding upwards coalition-monotonicity. The problem with this alternative definition is on the one hand that it does not match our proposed interpretation as well: a coalition can only be effective in case none of its members has violated the rules of the game, the fact that some of its members have played according to the rules is not sufficient. On the other hand, the alternative definition would not allow us to establish the link with modal logic so easily (see section 3.5): in order to establish this link, the coalition of all players must be ineffective in empty games and the empty coalition must be effective.

2.3.2 Group Ability in Strategic Games

The notion of α-effectivity which we defined for individual effectivity functions can easily be generalized to coalitional effectivity functions. In fact, the study of effectivity functions has concentrated on this coalitional setting. Given a game G, a coalition $C \subseteq N$ will be *α-effective* for a set $X \subseteq S$ iff the coalition has a joint strategy which will result in an outcome in X no matter what strategies the other players choose.

▶ Definition 2.23 (α-Effectivity in Strategic Games). For a nonempty strategic game G, its coalitional *α-effectivity function* $E^{\alpha}_{G} : \mathcal{P}(N) \to \mathcal{P}(\mathcal{P}(S))$ is defined as follows: $X \in E^{\alpha}_{G}(C)$ iff $\exists \sigma_C \forall \sigma_{\overline{C}}\ o(\sigma_C, \sigma_{\overline{C}}) \in X$.

We say that a coalitional effectivity function $E : \mathcal{P}(N) \to \mathcal{P}(\mathcal{P}(S))$ *α-corresponds* to a nonempty strategic game G iff $E = E^{\alpha}_{G}$.

Also, in the coalitional case, we can define the *β-effectivity function* as

$$X \in E^{\beta}_{G}(C) \text{ iff } \forall \sigma_{\overline{C}} \exists \sigma_C\ o(\sigma_C, \sigma_{\overline{C}}) \in X.$$

Recall that in figure 2.1, player 2 was β-effective for $\{s_2\}$ without being α-effective for it. The coalition consisting of players 1 and 2 on the other hand is α-effective for $\{s_2\}$.

For 2-player games, we have already defined the notion of determinacy in section 2.2.2. Using coalitional α-effectivity, we can now generalize this definition to n-player games: Note that an empty strategic game is determined iff $|G| = |N| - 1$, i.e., iff there is exactly one loser.

▶ Definition 2.24 (Determinacy). We call a possibly empty game G *determined* iff E^{α}_{G} is maximal.

As we characterized the class of individual effectivity functions which α-correspond to a strategic game in theorem 2.9, below we obtain an analogous characterization result for coalitional effectivity functions.

▶ Definition 2.25 (Strong Playability). A coalitional effectivity function $E : \mathcal{P}(N) \to \mathcal{P}(\mathcal{P}(S))$ is *strongly playable* if it satisfies the following four conditions: (1) $\forall C \subseteq N : \emptyset \notin E(C)$, (2) $\forall C \subseteq N : S \in E(C)$, (3) E is N-maximal, and (4) E is superadditive.

▶ Lemma 2.26. Every strongly playable effectivity function is regular and also coalition-monotonic.

Proof. For regularity, let $X \in E(C)$ and assume by reductio that $\overline{X} \in E(\overline{C})$. By superadditivity, $\emptyset \in E(N)$, contradicting condition (1) of strong playability.

For coalition monotonicity, let $X \in E(C)$ and $C \subseteq C'$. For $C''' := C' \setminus C$, condition (2) of strong playability give us $S \in E(C''')$ and so by superadditivity, $X \in E(C \cup C''') = E(C')$. ■

The proof of the next result follows the same general pattern as the proof of theorem 2.9.

▶ THEOREM 2.27. A coalitional effectivity function E α-corresponds to a non-empty strategic game iff E is strongly playable.

PROOF. One can easily check that the α-effectivity function of any strategic game satisfies the four properties of strong playability. As for the other direction, let E be an effectivity function satisfying the four properties. We shall construct a game G such that $E = E_G^\alpha$. To simplify our definitions, assume that $N = \{1, \ldots, n\}$.

To guide the reader through the following technical proof, we first provide a more informal sketch of the main argument: Given the playable effectivity function E, we construct a strategic game $G = (N, \{\Sigma_i | i \in N\}, o, S)$. A strategy σ_i for player i will be a triple (f_i, t_i, h_i): For every coalition C_i of which i is a member, the function f_i picks a set which C_i can force, and for every nonempty set X, the function h_i picks an element of X. Thus, if player i ends up as a member of coalition C_i, he will force $f_i(C_i)$, and if the choice is up to her, she will pick the outcome using h_i. Since all players will force certain sets as part of their strategy σ_i, we use the t_is to determine which player will get the power to determine the outcome state. The outcome of the game will be determined by the outcome function o roughly as follows: Given $(f_1, \ldots, f_n)$, N is partitioned into coalitions (as big as possible) such that all members of a coalition choose to force the same set. The outcome set will then be the result of each coalition forcing its set, i.e., the intersection of all the sets forced. The player who chooses which state in this set will be realized is then determined by adding up (modulo n) all the indices chosen as t_i. The effectivity function of this game is just E.

Formally, for $i \in N$, let $\mathcal{C}_i = \{C \subseteq N | i \in C\}$ be the set of coalitions of which i is a member. Let

$$F_i = \{f_i : \mathcal{C}_i \to \mathcal{P}(S) \mid \forall C : f_i(C) \in E(C)\},$$

so F_i consist of all functions f_i which associate to every coalition C in which i participates a set of outcomes for which C is effective. Note that since for all coalitions C, $S \in E(C)$, F_i will be nonempty for every player i.

Given $f \in \Pi_{i \in N} F_i = F_N$ and a coalition C, let $P(f, C)$ be the coarsest partition $\langle C_1, \ldots, C_m \rangle$ of C such that

$$\forall l \leq m \forall i, j \in C_l : f_i(C) = f_j(C).$$

Then given f, let

$$\begin{array}{rcl}
P_0(f) & = & \langle N \rangle \\
P_1(f) & = & P(f, N) = \langle C_1^1, \ldots, C_{k_1}^1 \rangle \\
P_2(f) & = & \langle P(f, C_1^1), \ldots, P(f, C_{k_1}^1) \rangle = \langle C_1^2, \ldots, C_{k_2}^2 \rangle \\
& \vdots & \\
P_r(f) & = & \langle C_1^r, \ldots, C_{k_r}^r \rangle.
\end{array}$$

Since there are only finitely many players, this partitioning process will eventually stop at some state r where $P_r(f) = P_{r+1}(f)$, and we let $P_\infty(f) = P_r(f) = \langle C_1, \ldots, C_k \rangle$. Since for all $l \leq k$ and $i, j \in C_l$ we have $f_i(C_l) = f_j(C_l)$ we will simply write $f(C_l)$ for it. Now let

$$G(f) = \bigcap_{l=1}^{k} f(C_l).$$

Claim: $G(f) \neq \emptyset$. Proof: Since C_l is effective for $f(C_l)$, i.e., $f(C_l) \in E(C_l)$ for all $l \leq k$, $\bigcap_{l=1}^{k} f(C_l) = G(f) \in E(N)$ by superadditivity, and hence since $\emptyset \notin E(N)$, $G(f)$ cannot be empty.

Now we can define the strategic game $G = (N, \{\Sigma_i | i \in N\}, o, S)$ as follows: Let $H = \{h : \mathcal{P}(S) \setminus \{\emptyset\} \to S \mid h(X) \in X\}$. Then we define

$$\Sigma_i = F_i \times N \times H \text{ and } o(\sigma_N) = h_{i_0}(G(f)),$$

where $\sigma_N = (f_i, t_i, h_i)_{i \in N}$ is a strategy profile and $i_0 = ((t_1 + \cdots + t_n) \ mod \ n) + 1$ indicates the player who has the power to determine the outcome. It remains to show that for all $C \subseteq N$, $E(C) = E^{\alpha}_G(C)$.

For the inclusion from left to right, assume that $X \in E(C)$. Choose any C-strategy $\sigma_C = (f_i, t_i, h_i)_{i \in C}$ such that for all $i \in C$ and for all $C'' \supseteq C$ we have $f_i(C'') = X$. By coalition-monotonicity, such f_i exist. Take any $\overline{C}$-strategy $\sigma_{\overline{C}} = (f_i, t_i, h_i)_{i \in \overline{C}}$. We need to show that $o(\sigma_C, \sigma_{\overline{C}}) \in X$. To see this, note that C must be a subset of one of the partitions C_l in $P_\infty(f)$. Hence,

$$o(\sigma_N) = o(\sigma_C, \sigma_{\overline{C}}) = h_{i_0}(G(f)) \in G(f) \subseteq f(C_l) = X.$$

For the inclusion from right to left, assume that $X \notin E(C)$. Suppose first that $C = N$. Then by N-maximality, $\overline{X} \in E(\emptyset)$, and by the previous part of the proof, $\overline{X} \in E^{\alpha}_G(\emptyset)$. Since E^{α}_G is strongly playable, it is regular (by the previous lemma) and so $X \notin E^{\alpha}_G(N)$, and we established the result.

So assume from now on that $C \neq N$, and let $j_0 \in N \setminus C$. Let σ_C be any strategy for coalition C. We must show that there is a strategy $\sigma_{\overline{C}}$ such that $o(\sigma_C, \sigma_{\overline{C}}) \notin X$. Define $\sigma_{\overline{C}} = (f_i, t_i, h_i)_{i \in \overline{C}}$ such that for all $C'' \supseteq \overline{C}$ and for all $i \in \overline{C}$ we have $f_i(C'') = S$. Then choose a t_{j_0} such that $((t_1 + \cdots + t_n) \ mod \ n) + 1 = j_0$.

Note that $\overline{C}$ must be a subset of one of the partitions C_l in $P_\infty(f)$. For the other partitions, superadditivity implies that there is some $C_0 \subseteq C$ such that $G(f) \in E(C_0)$, and hence by coalition-monotonicity, $G(f) \in E(C)$. Since $X \notin E(C)$, $G(f) \not\subseteq X$ by outcome-monotonicity, so there is some $s_0 \in G(f) \cap \overline{X}$. Now we define $h_{j_0}(G(f)) = s_0$. Then

$$o(\sigma_C, \sigma_{\overline{C}}) = h_{j_0}(G(f)) = s_0 \notin X.$$

■

The following theorem extends the previous result to possibly empty strategic games. The collection of properties needed to characterize possibly empty strategic games is called weak playability:

▶ DEFINITION 2.28 (WEAK PLAYABILITY). A coalitional effectivity function $E : \mathcal{P}(N) \to \mathcal{P}(\mathcal{P}(S))$ is *weakly playable* if it satisfies the following five conditions: (1) $\emptyset \notin E(N)$, (2) if $\emptyset \in E(C)$ and $C' \subseteq C$ then $\emptyset \in E(C')$, (3) if $\emptyset \notin E(\emptyset)$ then $S \in E(C)$ for all $C \subseteq N$, (4) E is N-maximal, and (5) E is superadditive.

It is easy to check that the terminology is justified in the sense that strong playability implies weak playability but not vice versa. The following result shows that terminality distinguishes weak from strong playability.

▶ THEOREM 2.29. A coalitional effectivity function is weakly playable iff it is strongly playable or terminal.

PROOF. Strongly playable as well as terminal effectivity functions are easily seen to be weakly playable. For the converse, assume that E is weakly playable and not strongly playable. Then there are two possible cases:

(i) There is some C such that $S \notin E(C)$. Then $\emptyset \in E(\emptyset)$, and whenever $X \in E(D)$, using superadditivity we have $\emptyset \in E(D)$ and so using monotonicity we also have $Y \in E(D)$ for every $Y \subseteq S$ and $X \in E(D')$ for every $D' \subseteq D$.

(ii) There is some C such that $\emptyset \in E(C)$. Then again we have $\emptyset \in E(\emptyset)$ and we can proceed as before. ■

▶ COROLLARY 2.30. A coalitional effectivity function E α-corresponds to a possibly empty strategic game iff E is weakly playable.

PROOF. An easy consequence of theorems 2.22, 2.27 and 2.29. ■

2.3.3 Group Ability in Dictatorships

▶ DEFINITION 2.31 (INDIVIDUALISM). A coalitional effectivity function E is *strongly (weakly) individualistic* iff it is strongly (weakly) playable and $E(N) \subseteq \bigcup_{i \in N} E(\{i\})$.

The condition ensures that everything which can be forced at all can be forced already by some individual. Since every strongly playable effectivity function is coalition-monotonic by lemma 2.26, the converse inclusion $\bigcup_{i \in N} E(\{i\}) \subseteq E(N)$ holds for all strongly playable effectivity functions.

The following result shows that individualism is an extremely strong assumption: While it seems to say only that the whole is equal to the sum of its parts, due to superadditivity, it actually says that the whole is equal to one particular part.

▶ THEOREM 2.32. A coalitional effectivity function E α-corresponds to a non-empty dictatorship iff E is strongly individualistic.

PROOF. First, if E is the effectivity function of a dictatorship with dictator d, E is easily seen to be strongly individualistic since $E(N) \subseteq E(\{d\})$.

Second, assume E is strongly individualistic, and so there is a strategic game G such that $E = E_G^{\alpha}$. We can assume that G has at least two distinct outcomes t_1 and t_2, for otherwise G is trivially a dictatorship. For every outcome t_i we know that $\{t_i\} \in E_G^{\alpha}(N) = E(N)$, and hence we know that some individual must be able to guarantee that outcome. Suppose by reductio that there are two individuals $i \neq j \in N$ such that $\{t_1\} \in E(\{i\})$ and $\{t_2\} \in E(\{j\})$. Then by superadditivity, $\emptyset \in E(\{i, j\})$, a contradiction. Hence, there is a player who can force any outcome, so G is a dictatorship. ■

Put positively, unless we have a dictatorship, coalitions of agents can sometimes achieve more than their members individually, cooperation is thus advantageous.

▶ COROLLARY 2.33. A coalitional effectivity function E α-corresponds to a possibly empty dictatorship iff E is weakly individualistic.

PROOF. The result follows from theorems 2.22, 2.29 and 2.32. ■

2.3.4 Individual vs. Group Ability

Clearly, the ability of coalitions is to some extent determined by the ability of its members. To take the simplest example, we know that in any coalition-monotonic effectivity function, a 2-player coalition can achieve at least as much as its two members individually. However, the 2-player coalition may be able to achieve more than we can determine from the individual powers of its members.

We say that a coalitional effectivity function $E : \mathcal{P}(N) \to \mathcal{P}(\mathcal{P}(S))$ *extends* an individual effectivity function $E' : N \to \mathcal{P}(\mathcal{P}(S))$ iff for all $i \in N$, $E(\{i\}) = E'(i)$. In the most general case, an individual effectivity function can be extended in an arbitrary way to a coalitional effectivity function as long as monotonicity is satisfied. In the case where we want to extend an individual effectivity function to a coalitional α-effectivity function of a strategic game, it seems that there will be less flexibility, however, and one might even wonder whether in that case individual effectivity completely determines group effectivity.

▶ DEFINITION 2.34 (INDIVIDUALLY DETERMINED). A class of coalitional effectivity functions K is *individually determined* if for any two effectivity functions $E_1, E_2 \in K$, if $E_1(\{i\}) = E_2(\{i\})$ for all agents $i \in N$, then $E_1 = E_2$.

The class of coalitional effectivity functions which α-correspond to an empty strategic game are easily seen to be individually determined: If $E_{G_1}^{\alpha}$ and $E_{G_2}^{\alpha}$ agree on singleton coalitions, $G_1 = G_2$. Since empty games only formalize the individual payoffs, they contain no further coalitional information.

▶ THEOREM 2.35. For $|N| > 1$, individualistic effectivity functions are individually determined, playable effectivity functions are not.

PROOF. For $N = \{1, 2\}$ and $S = \{s, t\}$, consider the coalitional α-effectivity functions associated with the two strategic games of figure 2.2, where E_l corresponds to the left game and E_r to the right one.

$$\begin{array}{c|c|c|} & l & r \\ \hline l & s & s \\ \hline r & s & t \\ \hline \end{array} \qquad \qquad \begin{array}{c|c|c|} & l & r \\ \hline l & s & s \\ \hline r & s & s \\ \hline \end{array}$$

Figure 2.2: Two strategic games which differ only in coalitional ability.

One can easily verify that $E_l(\{1\}) = E_r(\{1\})$ and $E_l(\{2\}) = E_r(\{2\})$, while $E_l(\{1, 2\}) \neq E_r(\{1, 2\})$. As a consequence, both strongly and weakly playable effectivity functions are not individually determined.

As for strongly individualistic effectivity functions and dictatorships, coalitions do not add anything to individual agents. If E is strongly individualistic, $E(C) = \bigcup_{i \in C} E(\{i\})$ for every nonempty coalition C. Since $X \in E(\emptyset)$ iff $\overline{X} \notin E(N)$, E is individually determined. Due to the fact that empty strategic games are individually determined as well, both strongly and weakly individualistic effectivity functions are individually determined. ■

It follows from the theorem that no class of effectivity functions which includes the strongly playable ones is individually determined. Note also that the strategic games which were given as examples in the proof are in fact determined, thereby demonstrating the stronger claim that even the class of maximal strongly playable effectivity functions is not individually determined.

While these results are neither surprising nor difficult to prove, they are nonetheless important to keep in mind for two reasons: First, when we consider dynamic models of ability with rich languages for describing them, it will turn out that even in models based on individualistic effectivity functions, *long-term* ability of groups need *not* be individually determined (see section 4.3.3). Second, the fact that even determined 2-player games are not individually determined shows that ignoring coalitional ability even in such simple games does mean a loss of information.

2.4 Dynamic Models of Ability

2.4.1 Dynamic Effectivity Frames

▶ DEFINITION 2.36 (DYNAMIC EFFECTIVITY FRAME). Given the set of agents N and a set of atomic games Γ_0, a *dynamic effectivity frame* $\mathcal{F} = (S, \{E_{C,g} | C \subseteq$

N and $g \in \Gamma_0\}$) consists of a nonempty set of states S and monotonic functions $E_{C,g} : S \to \mathcal{P}(\mathcal{P}(S))$, i.e., $X \in E_{C,g}(s)$ and $X \subseteq Y$ imply $Y \in E_{C,g}(s)$.

Intuitively, $E_{C,g}$ associates to every state $s \in S$ the sets of states for which coalition C is effective in game g. We assume that at every state there are a number of possible interactions or games Γ_0 which can occur. In each such interaction, different coalitions may be effective for different sets of states at which new interactions may happen. We assume that for every $g \in \Gamma_0$ and state $s \in S$, the function $E_{g,s} : \mathcal{P}(N) \to \mathcal{P}(\mathcal{P}(S))$ defined by $E_{g,s}(C) = E_{C,g}(s)$ is an effectivity function (i.e., it is monotonic). For easier readability, we shall often write $sE_{C,g}X$ instead of $X \in E_{C,g}(s)$, and we let $E_{C,g}(X) = \{s \in S | sE_{C,g}X\}$.

Conceptually, dynamic effectivity frames can be interpreted in two different ways. In case Γ_0 is a singleton, every state is associated with only one game (at most), and we can view the whole model itself as a complex game where the states of the world are associated to game positions. The interaction which may take place at each state is then a move in the game (where possibly players move simultaneously). We call this view the *internal view*, since the model describes the internal structure of a certain type of game. According to the *external view*, a dynamic effectivity frame describes interactions from the outside. The model itself is not viewed as a game, rather, the interactions which can take place at each state are complex games themselves, and we are interested in how one can move through this state space by combining different interactions by means of certain game operations. We will have more to say about these two views in sections 2.4.2 and 2.4.3. For now it is sufficient to realize that these two views are two conceptual sides of the same (technical) coin, for both views can be captured using the dynamic effectivity frames just defined. The situation is thus parallel to the situation with processes, where Kripke frames can also be used both as a model of the internal structure of a single process and as a model of the external structure of multiple processes.

Call a dynamic effectivity frame *uniformly finitary* iff there are a finite number of finite sets $X_1, \ldots, X_k \subseteq S$ such that if $sE_{C,g}X$ then there exists some $i \leq k$ such that $sE_{C,g}X_i$ and $X_i \subseteq X$. Clearly, all frames where S is finite will be uniformly finitary.

All of the properties of effectivity functions which we introduced previously can be lifted to dynamic effectivity frames: A dynamic effectivity frame $\mathcal{F} = (S, \{E_{C,g} | C \subseteq N$ and $g \in \Gamma_0\})$ has a given property (e.g. maximality) iff for every state $s \in S$ and every game $g \in \Gamma_0$, $E_{g,s}$ has the property. Similarly, the notion of α-correspondence can be lifted to frames in this way.

▶ Definition 2.37 (Kripke Frame). Given the set of atomic games Γ_0, a *Kripke frame* $\mathcal{K} = (S, \{R_g | g \in \Gamma_0\})$ consists of a nonempty set of states S and relations $R_g \subseteq S \times S$.

In many cases $|\Gamma_0| = 1$, so that a Kripke frame can simply be written as $\mathcal{K} = (S, R)$.

While frames provide information about how different states are linked among each other by way of coalitional effectivity, they do not contain any information about the properties of those states themselves. In order to add this type of information, we assume a set Φ_0 of atomic propositions which can describe properties of states. Elements of Φ_0 are typically denoted by $p, q, r, \ldots$, and given the set of states S, a *valuation function* $V : \Phi_0 \to \mathcal{P}(S)$ assigns to every proposition $p \in \Phi_0$ the set of states where p is true. Adding a valuation function to a dynamic effectivity frame $\mathcal{F}$, we obtain a *dynamic effectivity model* $\mathcal{M} = (\mathcal{F}, V)$.

The subsequent chapters will deal with restricted kinds of dynamic effectivity frames, coalition frames and game frames. These restricted frames are simpler and more suitable to investigate the differences between individual and coalitional effectivity on the one hand, and programs and games on the other hand. Nonetheless, the generality of dynamic effectivity frames is sometimes useful, for it allows us to formulate notions such as bisimulation (see section 2.5) in a fully general setting.

2.4.2 The Internal View: Coalition Frames

Chapters 3 and 4 will use coalition frames which ignore the possibility of multiple interactions and assume that at every state there is only one possible interaction which can occur.

▶ Definition 2.38 (Coalition Frame). A *coalition frame* is a pair $\mathcal{F} = (S, \{E_C | C \subseteq N\})$ where $E_C : S \to \mathcal{P}(\mathcal{P}(S))$ is monotonic, i.e., $X \in E_C(s)$ and $X \subseteq Y$ imply $Y \in E_C(s)$.

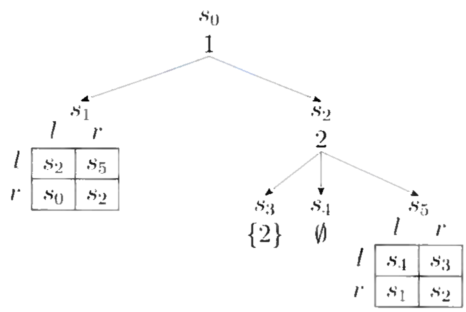
Call a state $s \in S$ *terminal* in case $E(s)$ is a terminal effectivity function. Terminal states thus mark the end of an interaction and assign payoffs to the players.

Certain coalition frames represent the ability of agents in well-known forms of strategic interaction:

Playable Frames: Extensive Games of Almost Perfect Information

Weakly playable coalition frames α-correspond to extensive games with simultaneous moves, i.e., extensive games of almost perfect information. Every state of the frame is either linked to a strategic game (i.e., a simultaneous move by all players) or to a payoff vector represented by an empty strategic game. The only difference with the standard notion of an extensive game is that the payoffs we associate to terminal states are of a restricted form: Players either win or lose (i.e., the payoffs are either 1 or -1) and at least one player must lose. This restriction, however, is not intrinsic to our framework but is mainly caused by the link we want to establish with modal logic in section 3.5. Note also that weakly playable coalition frames allow for infinite runs/plays as well as cycles.

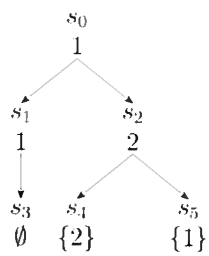
The following example of an extensive 2-player game with simultaneous moves contains two states s_0 and s_2 where the associated strategic games are dictatorships with different local dictators. At states s_1 and s_5, non-dictatorial strategic games are played. Finally, states s_3 and s_4 are terminal states which are linked to empty strategic games. Both players lose at state s_4 whereas only player 1 loses at state s_3.



Because of the correspondence between weakly playable coalition frames and extensive games of almost perfect information we shall usually not be very careful to distinguish them. Note also that by disregarding the effectivity information at terminal states of weakly playable coalition frames, these frames can be used to model effectivity in extensive game *forms* which do not contain any information about the players' payoffs.

Individualistic Frames: Extensive Games of Perfect Information

Weakly individualistic coalition frames α-correspond to extensive games without simultaneous moves, i.e., extensive games of perfect information. Every state is either a dictatorship, the dictator being the player whose turn it is at that state, or it is an empty strategic game and hence a payoff vector. The following game tree gives a visual example.

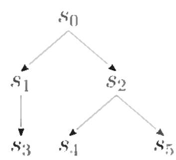


At state s_0, player 1 is the local dictator who has the local power to decide between states s_1 and s_2. Similarly at state s_2, player 2 has the power to decide between states s_4 and s_5, whereas at state s_1, player 1 can only choose state s_3. The states s_3, s_4 and s_5 are states linked to an empty strategic game which only consists of the sets of players who win, e.g., player 1 at state s_5.

Individualistic 1-Player Frames: Extensive 1-Player Games

Weakly individualistic/playable 1-player coalition frames α-correspond to extensive 1-player games, i.e., programs. Either player 1 can choose a successor state or she loses. We will see later how Coalition Logic over this class of frames is nothing but normal modal logic.

As the following example demonstrates, weakly playable 1-player frames are similar to weakly individualistic frames, except that the dictator is the same at every state and that payoffs are only given to one player. Since the same player chooses at every state, we do not need to label states with players anymore.



Note also that an empty game associated to a state must be a loss for the player which is why we leave the terminal states unmarked; they are all losses for the player.

In case $|N| = 1$, every weakly playable coalition frame $\mathcal{F} = (S, \{E_{\{1\}}, E_{\emptyset}\})$ α-corresponds to a Kripke frame $\mathcal{K} = (S, R)$, where

$$sE_{\{1\}}X \text{ iff } \exists t \in X : sRt. \tag{2.1}$$

Note that this essentially instantiates the definition of α-effectivity for 1-player strategic games. As a consequence, sRt holds iff $sE_{\{1\}}\{t\}$. R denotes the strategic game as an accessibility relation: At state $s \in S$, the strategic game associated with s lets player 1 choose among all states t such that sRt. Consequently, we associate the empty strategic game with s if there is no $t \in S$ such that sRt.

Coalition Models

A *coalitional model* $\mathcal{M} = ((S, E), V)$ consists of a coalition frame (S, E) and a valuation function $V : \Phi_0 \to \mathcal{P}(S)$. The restricted classes of frames we have considered can then easily be lifted to models, so that we can talk, e.g., about the class of weakly playable coalition models, referring to coalition models with a weakly playable effectivity frame. We will investigate a number of specific classes of coalition models in the following chapter, where the following notation will be used:

Mon	all coalition models
Play	weakly playable coalition models
MaxPlay	maximal weakly playable coalition models
Ind	weakly individualistic coalition models
n$-X$	coalition models of class X for $\|N\| = n$

Thanks to the characterization results of section 2.3, the classes Play and Ind capture precisely the ability structures which α-correspond to extensive games with and without simultaneous moves. 1$-$Play captures the ability associated with processes, and besides being interesting as a limiting case, it will be used together with 2$-$MaxPlay when making the connection with standard modal logic. Note that since every 1-player strategic game is automatically a dictatorship, 1$-$Play $=$ 1$-$Ind.

2.4.3 The External View: Game Frames

Where coalition frames are dynamic effectivity frames which allow for only one interaction per state, game frames allow for multiple interactions but only model the ability of a single player.

▶ DEFINITION 2.39 (GAME FRAME). A *game frame* is a pair $\mathcal{F} = (S, \{E_g | g \in \Gamma_0\})$ where $E_g : S \to \mathcal{P}(\mathcal{P}(S))$ is monotonic, i.e., $X \in E_g(s)$ and $X \subseteq Y$ imply $Y \in E_g(s)$.

Intuitively, $E_g : S \to \mathcal{P}(\mathcal{P}(S))$ models the ability of player 1 in game g. As has been shown in corollary 2.13, such an $E_g(s)$ always corresponds to the ability of a player in a determined 2-player strategic game. Note that game frames do not contain any information about coalitional ability such as the effectivity of both players together. It follows from theorem 2.35 that this coalitional effectivity information cannot be recovered from the individual effectivity information. A *game model* $\mathcal{M} = ((S, E), V)$ consists of a game frame (S, E) and a valuation function $V : \Phi_0 \to \mathcal{P}(S)$.

Using theorem 2.16, we can isolate the class of game frames which α-correspond to Kripke frames. Call an operation $F : \mathcal{P}(S) \to \mathcal{P}(S)$ *disjunctive* iff for all $V \subseteq \mathcal{P}(S)$ we have

$$F(\bigcup_{X \in V} X) = \bigcup_{X \in V} F(X).$$

Note that disjunctivity implies monotonicity and that $F(\emptyset) = \emptyset$. Recall that in section 2.2.3 we already defined what it means for a set $E \subseteq \mathcal{P}(S)$ to be disjunctive. Making use of this earlier definition, we can also say that an operation

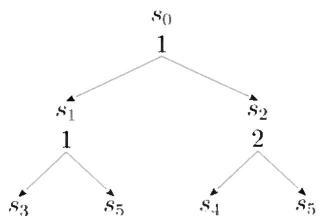
F is disjunctive iff for all $s \in S$, the set $\{X \subseteq S | s \in F(X)\}$ is disjunctive. By theorem 2.16, every disjunctive game frame $\mathcal{F} = (S, \{E_g | g \in \Gamma_0\})$ α-corresponds to a Kripke frame $\mathcal{K} = (S, \{R_g | g \in \Gamma_0\})$, where

$$sE_gX \text{ iff } \exists t \in X : sR_gt. \tag{2.2}$$

When we considered coalition frames, we assumed the internal view of frames, interpreting, e.g., a playable coalition frame as an extensive game with simultaneous moves, each state of the frame being associated with a strategic game. Thus, the frame as a whole represented a game of some sort and the local effectivity function modeled the kind of move which could be taken at that particular game position by one or many players.

Game frames on the other hand are most naturally interpreted as external models of many games. Each state is associated to a number of complex determined 2-player games. By corollary 2.13, the ability of the two players in such a game can be modeled by a collection of subsets of S, all the sets of states which player 1 can bring about. Hence every state allows for a number of different possible interactions in which the players can engage in. The result of such an interaction will again be a new state where further interactions may be possible. The interactions are not modeled in any detail, but only in terms of what the players can bring about at the end of the interaction.

As an example of how the internal and external view can be related, consider the following extensive game form of perfect information G_{ext} on the left:



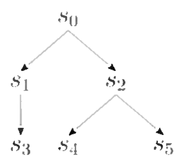
	l	r
ll	s_3	s_3
lr	s_5	s_5
rl	s_4	s_5
rr	s_4	s_5

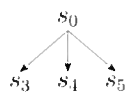
As discussed earlier, this extensive game form can be modeled by a weakly individualistic coalition model. Furthermore, we can consider the *strategic normal form* G_{snf} of this game which is depicted to the right of the extensive game G_{ext}. Every complex strategy of player 1 in the extensive game (such as ll where player 1 chooses the left move at both choice points) is a strategy in the normal form, and similarly for player 2. Note that the strategic game G_{snf} is determined.

G_{ext} can be modeled by a coalition frame where at state s_0 player 1 is effective for $\{s_1\}$, i.e., $s_0E_{\{1\}}\{s_1\}$ holds. When G_{ext} is condensed to its strategic normal form G_{snf}, it can be associated to a state in a game frame, in which case player 1 would be effective for $\{s_3\}$, i.e., $s_0E_{G_{snf}}\{s_3\}$ since he has a global strategy in G_{ext} which brings about state s_3.

In case G_{ext} is a game rather than a game form, the terminal states s_3, s_4 and s_5 will be marked either as a win for player 1 or as a win for player 2 (they cannot both lose since we are dealing with determined games only). Consequently, one of the two players i will have a winning strategy in G_{ext} by Zermelo's theorem [127, 19], and the strategic game G_{snf} which captures this information will simply be the empty strategic game $G = \{i\}$.

In terms of programs, the relationship between an extensive game and its strategic normal form is analogous to the relationship between the tree of execution sequences of a terminating nondeterministic program and its input-output relation. As the example below demonstrates, a Kripke frame can be used to model both of these perspectives: if the left Kripke frame models the execution sequences of a nondeterministic program, the right Kripke frame shows its input-output relation for input state s_0.





2.5 Identity Politics

When are two dynamic effectivity models the same? We consider isomorphism too strong, since we are primarily interested in the observable properties (as, e.g., specified by the valuation function) which a given state has. The notion of bisimulation expresses that two states s and s' are equivalent if they do not differ in the atomic properties which they have, and if whenever some agent or group is able to achieve X from s, then there should be some set Y such that that agent/group can achieve Y from s', and for every outcome in Y there is an equivalent outcome in X. Loosely speaking, if a group of agents can achieve something from one state, they can achieve at least as much from the other state, and vice versa. We first present the concept of bisimulation in its most general form, namely in its formulation for dynamic effectivity models.

► Definition 2.40 (Bisimulation). Let $\mathcal{M} = ((S, \{E_{C,g} | C \subseteq N \text{ and } g \in \Gamma_0\}), V)$ and $\mathcal{M}' = ((S', \{E'_{C,g} | C \subseteq N \text{ and } g \in \Gamma_0\}), V')$ be two dynamic effectivity models . Then $\leftrightarrows \subseteq S \times S'$ is a *bisimulation* between $\mathcal{M}$ and $\mathcal{M}'$ iff for any $s \leftrightarrows s'$ we have

1. $s \in V(p)$ iff $s' \in V'(p)$ for all $p \in \Phi_0$.

2. For all $C \subseteq N$, $g \in \Gamma_0$ and $X \subseteq S$: If $sE_{C,g}X$ then $\exists X' \subseteq S'$ such that $s'E'_{C,g}X'$ and $\forall x' \in X' \, \exists x \in X : x \leftrightarrows x'$.

3. For all $C \subseteq N$, $g \in \Gamma_0$ and $X' \subseteq S'$: If $s'E'_{C,g}X'$ then $\exists X \subseteq S$ such that $sE_{C,g}X$ and $\forall x \in X\ \exists x' \in X' : x \leftrightarrows x'$.

Two states are *bisimilar* iff there is a bisimulation $\leftrightarrows$ such that $s \leftrightarrows s'$. If we want to make the underlying models explicit, we also write $\mathcal{M}, s \leftrightarrows \mathcal{M}', s'$ instead of $s \leftrightarrows s'$.

It is easy to see how bisimulation can be instantiated for coalition models and game models, by leaving out the game component in the first case and the coalition component in the second case. Note that depending on whether one assumes the internal or the external view of a model, the interpretation of a bisimulation will be different. On the internal view of, e.g., an individualistic coalition model, a bisimulation establishes local similarity: If two positions of an extensive game without simultaneous moves are bisimilar, it means that at the present position and every future position of the game, the players have the same powers to force the game into a new position. On the external view of, e.g., a game model, bisimilarity establishes global similarity: A state may be linked to multiple complex games represented by their effectivity function, and whatever a player can achieve through playing a particular game at one state, she can achieve through playing the same game at the other state. Thus, what bisimilarity means will depend on the concrete interpretation of the model one has in mind. Note also that for terminal states, bisimilarity guarantees that the set of winning players is the same at both states.

One special instance of bisimulation is worth a few more remarks. Consider dynamic effectivity models which are weakly playable and where N contains only one player, say $N = \{1\}$. As pointed out before, such models α-correspond to Kripke models, and it turns out that for Kripke models, bisimulation reduces to the standard notion of bisimulation:

▶ Definition 2.41 (Kripke-Bisimulation). Let $\mathcal{M}_K = ((S, \{R_g | g \in \Gamma_0\}), V)$ and $\mathcal{M}'_K = ((S', \{R'_g | g \in \Gamma_0\}), V')$ be two Kripke models. Then $\leftrightarrows \subseteq S \times S'$ is a *Kripke-bisimulation* between $\mathcal{M}_K$ and $\mathcal{M}'_K$ iff for any $s \leftrightarrows s'$ we have

1. $s \in V(p)$ iff $s' \in V'(p)$ for all $p \in \Phi_0$.

2. For all $g \in \Gamma_0$: If $sR_g t$, then there is a $t' \in S'$ such that $s'R'_g t'$ and $t \leftrightarrows t'$.

3. For all $g \in \Gamma_0$: If $s'R'_g t'$, then there is a $t \in S$ such that $sR_g t$ and $t \leftrightarrows t'$.

▶ Theorem 2.42. For Kripke models, every bisimulation is a Kripke-bisimulation and vice versa. More precisely: Kripke models are Kripke-bisimilar iff their α-corresponding game/coalition models are bisimilar.

Proof. The result follows rather directly from equivalences (2.1) and (2.2) which link Kripke models to their corresponding coalition/game models. The following argument is formulated for coalition models.

Suppose a relation $\leftrightarroweq$ is a Kripke-bisimulation, and assume that $s \leftrightarroweq s'$ and $sE_{\{1\}}X$, i.e., for some state $t \in X$ we have sRt. Hence there is some t' such that $s'R't'$ and $t \leftrightarroweq t'$. But then $s'E'_{\{1\}}\{t'\}$, so $\leftrightarroweq$ is also a bisimulation. As for the other coalition, if $sE_{\emptyset}X$, then $X \supseteq \{t|sRt\}$. We know that $s'E'_{\emptyset}\{t'|s'R't'\}$, and hence it is sufficient to show that for all $t' \in S'$, if $s'R't'$ then there is some $t \in S$ such that sRt and $t \leftrightarroweq t'$, which follows from Kripke-bisimulation.

Conversely, suppose a relation $\leftrightarroweq$ is a bisimulation, and assume that $s \leftrightarroweq s'$ and sRt. Then $sE_{\{1\}}\{t\}$ and so there is some X' such that $s'E'_{\{1\}}X'$ and $\forall x' \in X' : t \leftrightarroweq x'$. Hence, there is some $t' \in X'$ such that $s'R't'$ and thus $t \leftrightarroweq t'$. ■

Like Kripke models, extensive games without simultaneous moves allow for a simple reformulation of bisimulation. Suppose we are given two extensive games without simultaneous moves G and G'. In order for two states s and s' to be bisimilar, they have to agree on the propositional atoms. Similarly, if one state is terminal then the other one is as well, and the states have the same set of winning players. If both states are non-terminal, we distinguish two situations: (1) In case the same player has to move at both states, we apply the standard bisimulation clauses: For every successor of s, s' has to have a bisimilar successor, and vice versa. (2) In case different players move at the two states, every successor of s has to be bisimilar to every successor of s'. It can be checked that the conditions given are indeed necessary and sufficient for bisimulation in the case of extensive games of perfect information.

2.6 Summary

The main concern of this chapter has been to characterize individual and group ability in strategic and extensive games. To give one example, on the static level, we characterized the coalitional effectivity functions which α-correspond to strategic games as the strongly playable ones. Similar results have been obtained for individual effectivity functions as well as for dictatorships. Figure 2.3 summarizes the results. Note that (with one exception) we have shown these results in two versions, depending on whether we allow for empty games or not.

These results thus provide a complete characterization of individual and group ability in strategic games and dictatorships. Dynamically, they yield a full characterization of local ability in extensive games with and without simultaneous moves.

In one particular case (theorem 2.12), the result obtained yielded a particularly interesting characterization, showing not only that an effectivity function of a certain kind α-corresponds to a determined strategic game, but also that this

static/dynamic game	individual ability	coalitional ability
strategic game/extensive game with simultaneous moves	amusing	playable
dictatorship/extensive game without simultaneous moves	disjunctive	individualistic

Figure 2.3: A summary of the characterization results.

strategic game is in fact the strategic normal form of an extensive game of perfect information where every player moves exactly once. This suggests a direction in which the results obtained may be extended: characterizing the effectivity functions which α-correspond to the normal form of an extensive game of a particular form, or in other words, characterizing the global effectivity functions of various classes of extensive games.

As theorem 2.35 has shown, if we want a model of group ability in extensive games with simultaneous moves, we need to employ coalitional effectivity functions. In extensive games without simultaneous moves on the other hand, all information about (local) group ability can be recovered from the individual effectivity function, the whole (ability of a group) being simply the sum of its parts (abilities of its members).

Finally, the generalized version of bisimulation introduced in section 2.5 is an extremely general notion of game equivalences. Besides generalizing Kripke-bisimulation, it also has a particularly simple instantiation for extensive games of perfect information. Furthermore, we shall see in chapters 3 and 6 that it is an appropriate equivalence notion for Coalition Logic and Game Logic.

2.7 Bibliographic Notes

Some of the characterization results of section 2.3 have originally been published in [98], more specifically theorems 2.27 and 2.32. Theorem 2.27 extends characterization results obtained in [86, 105, 94]. The result differs from the literature in that the outcome function of a strategic game is not assumed to be surjective, a central property for dynamic effectivity frames.

For textbooks on game theory, see [93, 19, 113]. Coalitional effectivity functions and the notion of α-effectivity (for nonempty strategic games) have been studied in [87, 86, 1, 105].

While we consider any outcome-monotonic function $E : \mathcal{P}(N) \to \mathcal{P}(\mathcal{P}(S))$ to be an effectivity function, most of the literature has taken a more restrictive view, requiring E to satisfy various basic properties (e.g., coalition-monotonicity).

The choice of these basic properties, however, is somewhat arbitrary, and this opinion is supported by the fact that authors differ in which basic properties they require. Here, we decided to be as general as possible regarding the notion of an effectivity function, applying restrictions only when these are actually needed to characterize a certain kind of ability or when technically unavoidable (as in the case of outcome-monotonicity).

For processes, a large variety of different equivalence notions has been studied, see, e.g., [53, 16]. For games this question has received relatively little attention [29, 13, 93]. In [17], the notion of bisimulation proposed in section 2.5 has been considered as an equivalence notion for concurrent processes in CPDL.

Chapter 3

Coalition Logic

In this chapter we introduce *Coalition Logic* (CL), a very general modal logic to reason about the coalition frames studied in the previous chapter. After defining the syntax and semantics of Coalition Logic, we first show that bisimilar coalition models cannot be distinguished by Coalition Logic formulas, and that over finite models, the converse holds as well. The rest of this chapter then consists of addressing a number of technical questions, namely the complexity of the model-checking problem, axiomatization and the complexity of the satisfiability problem.

As there are many different modal logics depending on the frame conditions which are imposed, there are different Coalition Logics depending on which class of frames is considered. We shall consider essentially three different classes of coalition frames, all of which have been introduced in the previous chapter: the class of all coalition frames and the classes of frames corresponding to extensive games with and without simultaneous moves. We investigate the technical questions just mentioned for all these frame classes, yielding a comparative analysis of how, e.g., extensive games with simultaneous moves differ from extensive games without simultaneous moves.

Besides comparing different classes of coalition frames, we also compare the language of full Coalition Logic to its fragment which can only express the ability of individuals. We examine for which classes of frames coalitional formulas add expressive power, and whether there is any difference in complexity between reasoning about coalitions and reasoning about individuals.

Finally, as one upshot of all this work, we present a coalitional game-theoretic view of modal logic. Normal modal logic, the logic of Kripke frames, is identical to Coalition Logic for one player when interpreted over extensive games. Non-normal modal logic on the other hand corresponds to the individual fragment of Coalition Logic over determined 2-player games. Hence, normal and non-normal modal logics are fragments of Coalition Logic, and viewing them as such we can gain new insights into some properties of these logics such as their complexity.

3.1 Syntax and Semantics

▶ Definition 3.1 (Coalition Logic Syntax). Given a finite nonempty set of agents/players N and a set of atomic propositions Φ_0, formulas φ of *Coalition Logic* can have the following syntactic form:

$$\varphi := \ \bot \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid [C]\varphi$$

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

As usual, we define $\top := \neg\bot$, $\varphi \wedge \psi := \neg(\neg\varphi \vee \neg\psi)$, $\varphi \rightarrow \psi := \neg\varphi \vee \psi$ and $\varphi \leftrightarrow \psi := (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$ as abbreviations. In case $C = \{i\}$, we write $[i]\varphi$ instead of $[\{i\}]\varphi$, and we use $[C]^k$ to denote a sequence of k $[C]$-modalities: inductively, $[C]^0\varphi = \varphi$ and $[C]^{k+1}\varphi = [C][C]^k\varphi$.

A further important shorthand notation concerns $[\bot]$ which shall abbreviate $\bigvee_{i\in N} \neg[i]\top$ and which will be true in playable coalition models at terminal states.

Recall from the previous chapter that a coalition model is a pair $\mathcal{M} = (\mathcal{F}, V)$ where $\mathcal{F}$ is a coalition frame $(S, \{E_C | C \subseteq N\})$ and $V : \Phi_0 \rightarrow \mathcal{P}(S)$ is the usual valuation function for the propositional letters.

▶ Definition 3.2 (Coalition Logic Semantics). Given a coalition model $\mathcal{M} = ((S, \{E_C | C \subseteq N\}), V)$, the truth of a formula φ in a model $\mathcal{M}$ at a state s, denoted as $\mathcal{M}, s \models \varphi$, is defined as follows:

$$\begin{array}{ll}
\mathcal{M}, s \not\models \bot & \\
\mathcal{M}, s \models p & \text{iff } p \in \Phi_0 \text{ and } s \in V(p) \\
\mathcal{M}, s \models \neg\varphi & \text{iff } \mathcal{M}, s \not\models \varphi \\
\mathcal{M}, s \models \varphi \vee \psi & \text{iff } \mathcal{M}, s \models \varphi \text{ or } \mathcal{M}, s \models \psi \\
\mathcal{M}, s \models [C]\varphi & \text{iff } sE_C\varphi^{\mathcal{M}}
\end{array}$$

where $\varphi^{\mathcal{M}} = \{s \in S | \mathcal{M}, s \models \varphi\}$.

Hence, a formula $[C]\varphi$ holds at a state s iff coalition C is effective for $\varphi^{\mathcal{M}}$ at s. The notions of validity, satisfiability and logical consequence can be defined in the standard way: A formula φ is *valid* in a model $\mathcal{M}$ with universe S, denoted as $\mathcal{M} \models \varphi$, iff $\varphi^{\mathcal{M}} = S$, and φ is valid in a class of models K (denoted as $\models_{\mathsf{K}} \varphi$, or $\models \varphi$ if K is the class of all models Mon) iff for all models $\mathcal{M} \in \mathsf{K}$ we have $\mathcal{M} \models \varphi$. Formula φ is *satisfiable* in a model $\mathcal{M}$ iff $\varphi^{\mathcal{M}} \neq \emptyset$, and φ is satisfiable in a class of models K iff there is some model $\mathcal{M} \in \mathsf{K}$ in which φ is satisfiable.

We write $\Sigma \models_{\mathsf{K}} \varphi$ (or again $\Sigma \models \varphi$ in case K includes all models) to denote that φ is a *(local) logical consequence* of Σ: For all models $\mathcal{M} \in \mathsf{K}$ and every state s of the universe of $\mathcal{M}$, if $\mathcal{M}, s \models \Sigma$ (i.e., all formulas of Σ are true at s) then $\mathcal{M}, s \models \varphi$. We will also on a few occasions make use of the global consequence relation: $\Sigma \models_g \varphi$ denotes that φ is a *global logical consequence* of Σ: For all models $\mathcal{M}$ (of a given class K of models), if $\mathcal{M} \models \Sigma$ then $\mathcal{M} \models \varphi$. For global consequence, the relevant class K is not part of the mathematical notation but will be clear from the context.

While we shall usually assume that the set of agents N is a *fixed* parameter, it is worth considering what role this parameter plays for the satisfiability of a formula. Considering a formula φ of Coalition Logic, let $N(\varphi)$ be the set of agents which occur in φ in some coalition, e.g. $N([2]p \vee [\{3,5\}]q) = \{2,3,5\}$. Any model satisfying φ will have to be for a set of agents N which at least includes $N(\varphi)$. But is it possible that φ is satisfiable for $N = N(\varphi)$ but not satisfiable for some strict superset of $N(\varphi)$? Similarly, can a formula φ which is not satisfiable for $N(\varphi)$ become satisfiable by adding a new agent which does not occur in φ? At least the latter question has a (possibly) surprising answer: Over playable coalition models, the formula $\neg[1]p \wedge \neg[1]q \wedge [1](p \vee q)$ is not satisfiable for $N = \{1\}$ while it is satisfiable, e.g., for $N = \{1,2\}$. The following result captures how adding agents can influence satisfiability.

▶ THEOREM 3.3. (1) For every class of models $\mathsf{K} \in \{\mathsf{Mon}, \mathsf{Play}, \mathsf{MaxPlay}, \mathsf{Ind}\}$ and any formula φ: If φ is K-satisfiable for $N = N(\varphi)$, then it is K-satisfiable for every $N \supseteq N(\varphi)$. (2) For any formula φ: If φ is not Mon-satisfiable for $N = N(\varphi)$, then it is not Mon-satisfiable for any $N \supseteq N(\varphi)$. (3) For every class of models $\mathsf{K} \in \{\mathsf{Play}, \mathsf{MaxPlay}, \mathsf{Ind}\}$ and any formula φ: If φ is not K-satisfiable for $N = N(\varphi)$ and for some $N \supsetneq N(\varphi)$, then it is not K-satisfiable for any $N \supseteq N(\varphi)$.

PROOF. (1) For Mon, it is easy to extend a model satisfying φ by one agent so that monotonicity holds. Note also that for empty games, adding an additional player will not cause a problem, no matter whether we add the additional player to the set of winners or not. For Play, consider a model for agents $N(\varphi)$ satisfying φ. We already saw that terminal states can easily be extended by an additional player. For non-terminal states, a nonempty $|N(\varphi)|$-player strategic game G can be extended by an additional player who only has a single strategy yielding a game G', thus leaving the effectivity of the other players unchanged. In case we are dealing with a determined game G, G' will also be determined and the same holds for empty games if we add the new player to the set of winners. Finally for Ind, adding a new player causes no problem at all, the individualistic coalition frame can even formally remain unchanged.

(2) For Mon, if φ is satisfiable for $N \supsetneq N(\varphi)$, then the restriction of this satisfying model to agents $N(\varphi)$ will be monotonic and also satisfy φ.

(3) Assume that φ is neither K-satisfiable for $N(\varphi)$ nor for $M \supsetneq N(\varphi)$. Assume by reductio that φ is K-satisfiable for some $N \supsetneq N(\varphi)$. Note that since the names of the new players added to $N(\varphi)$ do not matter (they do not occur in φ), we can assume that $M \subsetneq N$, for otherwise $N \subsetneq M$ which would contradict (1). Hence, we have $N(\varphi) \subsetneq M \subsetneq N$.

Consider first the case of Ind. Assume that $i \in M \cap \overline{N(\varphi)}$. Then given a model $\mathcal{M}_N$ for N which satisfies φ, at every non-terminal state with local dictator $d \notin N(\varphi)$, we can replace d by i. At a terminal state s with set of winners $G(s) \subsetneq N$, we define the new empty strategic game $G'(s) = G(s) \cap M$ in case $G(s) \neq M$, otherwise we let $G'(s) = M \setminus \{i\}$. This definition guarantees that $G'(s) \neq M$. As a consequence, we obtain an extensive game for players M which leaves the effectivity of the players in $N(\varphi)$ in tact and hence satisfies φ.

Consider next the case of Play, where we assume w.l.o.g. that $N(\varphi) = \{1\}$, $M = \{1, 2\}$ and $N = \{1, 2, 3, 4\}$. Let $\mathcal{M}_N$ be the model satisfying φ, and consider an arbitrary state s and its associated strategic game $G(s) = (N, \{\Sigma_i | i \in N\}, o, S)$. We construct a model $\mathcal{M}_M$ for agents M by associating to state s game $G'(s) = (M, \{\Sigma'_i | i \in M\}, o', S)$ as follows: $\Sigma'_1 = \Sigma_1$ and $\Sigma'_2 = \Sigma_2 \times \Sigma_3 \times \Sigma_4$ and $o'(\sigma_1, (\sigma_2, \sigma_3, \sigma_4)) = o(\sigma_1, \sigma_2, \sigma_3, \sigma_4)$. Note that the α-effectivity of player 1 is the same in both games, since we only collapsed three of his opponents into one with the same abilities. We deal with terminal states as in the case of Ind. The model $\mathcal{M}_M$ defined in this way is playable and will satisfy φ since the effectivity of players $N(\varphi)$ has not been changed, a contradiction. Furthermore, if $\mathcal{M}_N$ is maximal, $\mathcal{M}_M$ will be maximal as well, which takes care of the claim for MaxPlay. ■

For Mon, adding agents does not influence satisfiability. As a consequence, if we want to know whether φ is satisfiable for some set of agents, it is sufficient to check satisfiability for $N = N(\varphi)$. For Play, MaxPlay and Ind on the other hand, if we want to know whether φ is satisfiable for some set of agents, we need to check for satisfiability twice, once for $N = N(\varphi)$ and once for some $N \supsetneq N(\varphi)$.

3.2 Bisimulation Invariance

Once we have defined a logical language for describing certain mathematical structures, we automatically obtain an equivalence notion based upon that language. In the case of Coalition Logic, we say that two states s and s' are *CL-equivalent* iff they satisfy the same Coalition Logic formulas. Since we have come across another equivalence notion before, it is worth-while considering how these two equivalence notions relate. A formula whose truth-value does not differ at bisimilar states is called *bisimulation invariant*. The following result shows that all CL-formulas are bisimulation invariant.

▶ THEOREM 3.4. Bisimilarity implies CL-equivalence.

PROOF. The proof is by induction on CL-formulas. Assume that $\mathcal{M}, s \leftrightarroweq \mathcal{M}', s'$. For the only non-trivial case, assume that $\mathcal{M}, s \models [C]\varphi$, i.e., $sE_C\varphi^{\mathcal{M}}$. By bisimilarity, there is some X' such that $s'E'_CX'$ and for all $x' \in X'$ there is some t such that $\mathcal{M}, t \models \varphi$ and $t \leftrightarroweq x'$. By induction hypothesis, $X' \subseteq \varphi^{\mathcal{M}'}$, and hence by monotonicity, $s'E'_C\varphi^{\mathcal{M}'}$. ■

The converse of this result does not hold in all cases: Since normal modal logic forms a fragment of Coalition Logic, the Kripke models below which are modally equivalent but not bisimilar at the root (see, e.g., [20]) also form a counterexample to the converse of theorem 3.4. Both models contain branches of every finite length, but the right Kripke model also contains an infinite branch.

But as in the case of modal logic where image-finiteness is sufficient to establish the converse, for Coalition Logic, the converse holds for uniformly finitary models.

▶ THEOREM 3.5. For uniformly finitary coalition models, CL-equivalence implies bisimilarity.

PROOF. Consider two coalition models $\mathcal{M}$ and $\mathcal{M}'$ with respective universes S and S'. Let $\equiv \subseteq S \times S'$ be defined as the set of pairs of states which are CL-equivalent. We show that $\equiv$ is a bisimulation.

Assume that $sE_C X$ and since $\mathcal{M}$ is uniformly finitary, there is some finite $X_0 \subseteq X$ such that $sE_C X_0$. Suppose by reductio that for all X' such that $s'E'_C X'$ there is some $x' \in X'$ such that for all $x \in X_0$: $x \not\equiv x'$. Since the models are uniformly finitary, there are $X'_1, \ldots, X'_k \subseteq S'$ such that whenever $s'E'_C X'$, there is some $X'_i \subseteq X'$ such that $s'E'_C X'_i$. Hence whenever $s'E'_C X'_i$ there is some $x' \in X'_i$ such that for all $x \in X_0$: $x \not\equiv x'$. Let the finite set of these x' be denoted by Δ.

So given some X'_i, there is some state $x' \in X'_i$ and some formula $\varphi_{x'} = \bigvee_{x \in X_0} \psi_x$ such that ψ_x is true at x but false at x'. Consequently, $\varphi_{x'}$ is false at x' and true at all $x \in X_0$.

Now note that $\mathcal{M}, s \models [C] \bigwedge_{x' \in \Delta} \varphi_{x'}$ since any state in X_0 makes $\bigwedge_{x' \in \Delta} \varphi_{x'}$ true. On the other hand, $\mathcal{M}', s' \not\models [C] \bigwedge_{x' \in \Delta} \varphi_{x'}$, a contradiction. ■

3.3 Complexity I: Model Checking

The complexity of a logic can be measured in terms of two main decision problems, satisfiability and model checking. The complexity of the satisfiability problem will be discussed in section 3.6. The global *model-checking problem* is to find the set of states at which a given formula is true in a given finite model.

Model checking can be used to answer various questions of a game-theoretic nature using Coalition Logic. To give one example (further examples will be given in chapter 5), for any extensive game with simultaneous moves, finding out whether a player or a group of players has a winning strategy in the game (i.e., a strategy leading to an empty game where the player(s) win) can be formulated as a model checking problem in Coalition Logic. We simply represent the extensive game as a weakly playable coalition model where the terminal states are linked to payoffs via empty games. If the extensive game has length n (i.e., at most n moves can be made before reaching a terminal state), coalition C has a winning strategy iff the formula $[C]^{n+1}\bot$ is true at the initial state of the coalition model. The formula is true in case coalition C has a strategy to reach an empty game where C wins after at most n moves. Consequently, a model-checking algorithm also provides us with an algorithm to determine who has a winning strategy in an extensive game. Furthermore, an analysis of the complexity of the model-checking algorithm will yield an upper bound for the complexity of determining whether a player has a winning strategy. In order to analyze model checking, however, we first need to agree on how to measure the size of a formula and the representation and size of a model.

3.3.1 Representation of Coalition Models

Since we assume our coalition models to satisfy outcome-monotonicity, a lot of the information they contain is in fact redundant since it follows automatically by monotonicity. A consequence for the representation of a model is that coalition models can be represented more succinctly than by giving the full effectivity functions: We simply encode only the effectivity information which is not implied by monotonicity. As it turns out, however, the gain in succinctness may not be very big. To see this, consider a coalition model $\mathcal{M} = ((S, E), V)$. For every state $s \in S$ and coalition $C \subseteq N$, we know that $|\{X \subseteq S | sE_C X\}| \leq 2^{|S|}$. Suppose now we represent E by its *non-monotonic core* E^c, where $sE^c_C X$ holds iff $sE_C X$ and there is no $X_0 \subseteq X$ such that $sE_C X_0$. By Sperner's theorem [114], $|\{X \subseteq S | sE^c_C X\}| \leq \binom{|S|}{\lceil |S|/2 \rceil}$, since this binomial coefficient gives the size of the greatest antichain of subsets of a set S. This bound is the best possible since one can define $sE_C X$ iff $|X| \geq \lceil |S|/2 \rceil$ for which $sE^c_C X$ iff $|X| = \lceil |S|/2 \rceil$. Unfortunately, $\binom{|S|}{\lceil |S|/2 \rceil}$ is still exponential, so in general, representing a coalition model by its non-monotonic core does not essentially reduce the representation size.

As it turns out, however, there is at least one important class of coalition models where E^c is exponentially more succinct than E. As a consequence, we shall define the size of a coalition model in terms of its non-monotonic core.

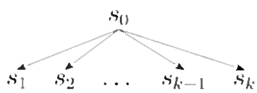
▶ DEFINITION 3.6 (MODEL SIZE). The size $|\mathcal{M}|$ of a finite coalition model $\mathcal{M} =$

$((S, E), V)$ is defined as

$$|\mathcal{M}| = |S| + \sum_{\{s|s\in S\}} \sum_{\{C|C\subseteq N\}} \sum_{\{X|sE^c_C X\}} |X|.$$

In words, we add to the number of states the cardinality of X for every state s and every coalition C such that $sE^c_C X$ holds. Note that the size of the model is independent of its valuation, hence we can equally well speak of the size of a frame.

As can easily be seen, $|\mathcal{M}|$ can be very big, exponential in the number of states. For some classes of coalition models, however, the additional structure put upon the effectivity functions can yield a substantially smaller representation. The most important case is the class of coalition models which correspond to Kripke models. To see this, we compare the size of a coalition model to the size of its corresponding Kripke model. Consider a particular state s_0 of a Kripke model with successors $s_1, \ldots, s_k$.



In general, the size of a Kripke model $|\mathcal{M}_K|$ is defined to be the number of states plus the size of its accessibility relation, i.e., $|\mathcal{M}_K| = |S| + |\{(s,t)|sRt\}|$. Since the set of states is the same in a Kripke model and its corresponding coalition model, it is sufficient to compare for the given state s_0 the number of successors k with

$$\sum_{s_0 E^c_{\{1\}} X} |X| \; + \sum_{s_0 E^c_{\emptyset} X} |X| = k + k = 2k.$$

This illustrates that the representation of a Kripke model is about half the size of the corresponding coalition model core due to the complete redundancy of $E_\emptyset$ when given $E_{\{1\}}$.

For games with more than one player, the redundancy in representation will usually be somewhat greater for coalition models, since, for example, even the non-monotonic core of a coalition model does allow for redundancy regarding coalition-monotonicity. Still, in many cases the simple structure of the effectivity function will allow for a representation which is much more succinct than the worst-case exponential upper bound suggested by $|\mathcal{M}|$. Furthermore, the generality of coalition models is a great conceptual asset, for it allows for a uniform treatment of effectivity in very different kinds of models. The algorithm provided in the next subsection will provide a uniform method for checking the truth of a formula in all these different models, and while there might be algorithms tailor-made for a particular model class which work with somewhat more efficient representations, the arguments presented suggest that such a gain in space efficiency will often not be dramatic enough to justify switching to a more restrictive model of ability.

3.3.2 Time Complexity of Model Checking

Let the size or length of a formula φ be the number of its subformulas, i.e., we define $|\varphi| = |sf(\varphi)|$. As in normal modal logic, verifying the truth of a formula at a given state of a model can be done very efficiently, essentially in time linear in the size of the model times the size of the formula. Furthermore, as in normal modal logic, the algorithm is uniform for all models, i.e., the complexity is independent of the model class.

▶ THEOREM 3.7. Given a Coalition Logic formula φ and a coalition model $\mathcal{M}$, there is an algorithm for calculating $\varphi^{\mathcal{M}}$ which runs in time $O(|\mathcal{M}| \times |\varphi|)$.

PROOF. Let $\mathcal{M} = ((S, E), V)$ and let $\varphi_1, \ldots, \varphi_n$ be the subformulas of φ ordered according to length where $\varphi_n = \varphi$. So if φ_i is a subformula of φ_j we have $i < j$. We show by induction on $k \leq n$ that we can determine all $\varphi_i^{\mathcal{M}}$ for $i \leq k$ in time $O(k \times |\mathcal{M}|)$. For $k = 1$, φ must be an atom and $\varphi^{\mathcal{M}}$ is already part of the description of the model. For $k + 1$, we proceed by cases based on the structure of φ_{k+1}. The case for $\varphi_{k+1} = \neg\psi$ is immediate, so consider $\varphi_{k+1} = \alpha \vee \beta$ where we assume w.l.o.g. that $\alpha = \varphi_{k-1}$ and $\beta = \varphi_k$. Then the truth values of both α and β can be determined globally in time $O(k \times |\mathcal{M}|)$ and hence the truth value of $\alpha \vee \beta$ can be determined in time $O(|S| + k \times |\mathcal{M}|)$. In case $\varphi_{k+1} = [C]\varphi_k$, after determining all the $\varphi_i^{\mathcal{M}}$ for $i \leq k$ in time $O(k \times |\mathcal{M}|)$, we check for every $sE_C^c X$ whether $X \subseteq \varphi_k^{\mathcal{M}}$ which can be done in $O(|\mathcal{M}|)$. ■

Note that contrary to what is suggested by the definition of $|\mathcal{M}|$, the time-complexity of model checking does not depend on the number of players in N. While adding an additional player will usually greatly increase $|\mathcal{M}|$ given that all subsets of players are considered, for model-checking we only need to investigate one subset per modality, not all of them. In other words, rather then considering all coalitions, it is sufficient to consider the worst-case coalition. This is analogous to Kripke models and standard normal modal logic: In case there are multiple accessibility relations for different agents, the time-complexity of model checking depends only on the worst-case accessibility relation.

3.4 Axiomatization

In this section, we shall axiomatize the validities of all the model classes introduced previously. Axiomatizations can be obtained by translating the appropriate effectivity function conditions into the language of Coalition Logic.

▶ DEFINITION 3.8 (COALITION LOGIC AXIOMATICS). Given the set of players N, a *coalition logic for N* is a set of formulas Λ which contains all propositional tautologies and which is closed under the rules of Modus Ponens and Monotonicity shown in figure 3.1 below.

$$\frac{\varphi \quad \varphi \to \psi}{\psi} \qquad\qquad \frac{\varphi \to \psi}{[C]\varphi \to [C]\psi}$$

Figure 3.1: The two inference rules of Coalition Logic: Modus Ponens and Monotonicity

Which additional axioms need to be adopted will depend on the class of models under consideration. Given a coalition logic Λ, we write $\vdash_\Lambda \varphi$ for $\varphi \in \Lambda$ and $\Sigma \vdash_\Lambda \varphi$ if there exist $\sigma_1, \ldots, \sigma_n \in \Sigma$ such that $(\sigma_1 \wedge \ldots \wedge \sigma_n) \to \varphi \in \Lambda$. When the subscript Λ is omitted and we talk about $\vdash \varphi$ and $\Sigma \vdash \varphi$, we take Λ to be the smallest coalition logic. Finally, a set of formulas Σ is Λ-inconsistent iff $\Sigma \vdash_\Lambda \bot$.

Call a coalition logic Λ *sound* with respect to a class of coalition models K if $\Sigma \vdash_\Lambda \varphi$ implies $\Sigma \models_{\mathsf{K}} \varphi$, and *complete* if the converse holds.

Let Λ be any coalition logic. Via the standard argument of Lindenbaum's lemma (see e.g. [20]), every Λ-consistent set of formulas Σ can be extended to a maximally Λ-consistent set $\Sigma' \supseteq \Sigma$ with the usual properties: (1) for every formula φ, $\varphi \in \Sigma'$ or $\neg\varphi \in \Sigma'$, (2) $\varphi \vee \psi \in \Sigma'$ iff $\varphi \in \Sigma'$ or $\psi \in \Sigma'$, and (3) if $\Sigma' \vdash_\Lambda \varphi$ then $\varphi \in \Sigma'$.

3.4.1 General Coalition Frames

As might be expected, ability in general coalition frames needs only the axioms of propositional logic. Let Λ be any coalition logic, let S^Λ be the set of all maximally Λ-consistent sets of formulas, and let $\widehat{\varphi} := \{s \in S^\Lambda | \varphi \in s\}$. Define the canonical Λ-model $\mathcal{C}^\Lambda = ((S^\Lambda, E^\Lambda), V^\Lambda)$ as follows:

$$\begin{aligned} s \in V^\Lambda(p) \quad & \text{iff } p \in s \\ sE^\Lambda_C X \quad & \text{iff } \exists \widehat{\varphi} \subseteq X : [C]\varphi \in s \end{aligned}$$

To see that E^Λ is well-defined, note first that E^Λ is outcome-monotonic. Furthermore, if $\widehat{\varphi_1} = \widehat{\varphi_2}$, $\vdash_\Lambda \varphi_1 \leftrightarrow \varphi_2$ and so $\vdash_\Lambda [C]\varphi_1 \leftrightarrow [C]\varphi_2$ which implies that for all $s \in S^\Lambda$, $[C]\varphi_1 \in s$ iff $[C]\varphi_2 \in s$. From our definition, one can easily prove the following truth lemma by induction.

▶ LEMMA 3.9. For any maximally Λ-consistent set $s \in S^\Lambda$ and any formula φ: $\mathcal{C}^\Lambda, s \models \varphi$ iff $\varphi \in s$. Equivalently, $\varphi^{\mathcal{C}^\Lambda} = \widehat{\varphi}$.

PROOF. For atomic formulas and for the boolean inductive steps, the argument is standard. For $[C]\varphi$, suppose $s \in ([C]\varphi)^{\mathcal{C}^\Lambda}$, i.e., there is some $\widehat{\varphi_0} \subseteq \varphi^{\mathcal{C}^\Lambda}$ such that $[C]\varphi_0 \in s$. Since by induction hypothesis $\varphi^{\mathcal{C}^\Lambda} = \widehat{\varphi}$, $\vdash_\Lambda \varphi_o \to \varphi$ and so using the monotonicity rule, $[C]\varphi \in s$ as well.

Conversely, if $[C]\varphi \in s$, given that $\varphi^{\mathcal{C}^\Lambda} = \widehat{\varphi}$ by induction hypothesis, the result follows immediately. ■

Then using the following canonical model theorem, we obtain axiomatic completeness as a corollary.

▶ THEOREM 3.10. Every coalition logic Λ is sound and complete with respect to its canonical model $\mathcal{C}^\Lambda$.

PROOF. Let Λ be any coalition logic. If $\Sigma \not\models_{\{\mathcal{C}^\Lambda\}} \varphi$, there is some maximally Λ-consistent set $\Sigma' \in S^\Lambda$ such that $\mathcal{C}^\Lambda, \Sigma' \models \bigwedge \Sigma \wedge \neg\varphi$. By the truth lemma, $\Sigma \subseteq \Sigma'$ and $\varphi \notin \Sigma'$. Consequently, $\Sigma' \not\vdash_\Lambda \varphi$ and hence also $\Sigma \not\vdash_\Lambda \varphi$. For the converse, suppose $\Sigma \not\vdash_\Lambda \varphi$, so $\Sigma \cup \{\neg\varphi\}$ is Λ-consistent, and so there is a maximally Λ-consistent set $\Sigma' \in S^\Lambda$ with $\Sigma \cup \{\neg\varphi\} \subseteq \Sigma'$ such that $\mathcal{C}^\Lambda, \Sigma' \models \bigwedge \Sigma$ while $\mathcal{C}^\Lambda, \Sigma' \not\models \varphi$, showing that $\Sigma \not\models_{\{\mathcal{C}^\Lambda\}} \varphi$. ■

▶ COROLLARY 3.11. $\Sigma \models \varphi$ iff $\Sigma \vdash \varphi$.

3.4.2 Extensive Games with Simultaneous Moves

▶ DEFINITION 3.12 (PLAY). `Play` is the smallest coalition logic which contains all instances of the axioms shown in figure 3.2 below.

($\mathtt{N}\bot$)	$\neg[N]\bot$
($\top$)	$\neg[\emptyset]\bot \to [C]\top$
($\bot$)	$[C]\bot \to [C']\bot$ where $C' \subseteq C$
(N)	$\neg[\emptyset]\neg\varphi \to [N]\varphi$
(S)	$([C_1]\varphi_1 \wedge [C_2]\varphi_2) \to [C_1 \cup C_2](\varphi_1 \wedge \varphi_2)$, where $C_1 \cap C_2 = \emptyset$

Figure 3.2: The axiom schemas for weak playability.

Notice that the axioms are direct translations of the weak playability conditions into the modal language.

Now let $\Lambda \supseteq$ `Play` and let S^Λ be the set of all maximally Λ-consistent sets of formulas. Define the canonical Λ-model $\mathcal{C}^\Lambda = ((S^\Lambda, E^\Lambda), V^\Lambda)$ as follows:

$$\begin{array}{rl} s \in V^\Lambda(p) & \text{iff } p \in s \\ sE^\Lambda_C X & \text{iff } \left\{ \begin{array}{ll} \exists\widehat{\varphi} \subseteq X : [C]\varphi \in s & \text{for } C \neq N \\ \forall\widehat{\varphi} \subseteq \overline{X} : [\emptyset]\varphi \notin s & \text{for } C = N \end{array} \right. \end{array}$$

Note that again E^Λ is outcome-monotonic by definition.

▶ LEMMA 3.13. (S^Λ, E^Λ) is weakly playable.

PROOF. Straightforward. Consider, e.g., superadditivity. By lemma 2.19, we can assume that $C_1 \cup C_2 \neq N$. For $C_1 \cap C_2 = \emptyset$, assume that $[C_1]\varphi_1, [C_2]\varphi_2 \in s$ for $\widehat{\varphi_1} \subseteq X_1$ and $\widehat{\varphi_2} \subseteq X_2$. By the superadditivity axiom, $[C_1 \cup C_2](\varphi_1 \wedge \varphi_2) \in s$, and since $\widehat{\varphi_1 \wedge \varphi_2} \subseteq X_1 \cap X_2$, we have $sE^{\Lambda}_{C_1 \cup C_2}(X_1 \cap X_2)$. ■

▶ THEOREM 3.14. **Play** is sound and complete with respect to the class of all weakly playable coalition models: $\Sigma \models_{\mathtt{Play}} \varphi$ iff $\Sigma \vdash_{\mathtt{Play}} \varphi$.

PROOF. As before, we can prove the following truth lemma: For any maximally Λ-consistent set $s \in S^{\Lambda}$ and any formula φ: $\mathcal{C}^{\Lambda}, s \models \varphi$ iff $\varphi \in s$. Equivalently, $\varphi^{\mathcal{C}^{\Lambda}} = \widehat{\varphi}$. For the modality, it is sufficient to consider the case where $C \neq N$, since $sE^{\Lambda}_N X$ iff not $sE^{\Lambda}_{\emptyset}\overline{X}$ and $\vdash_{\mathtt{Play}} [N]\varphi \leftrightarrow \neg[\emptyset]\neg\varphi$. For $C \neq N$, the proof is as in lemma 3.9. Next, one can show that every coalition logic $\Lambda \supseteq$ **Play** is sound and complete with respect to its canonical model $\mathcal{C}^{\Lambda}$. Since by lemma 3.13 the canonical model $\mathcal{C}^{\Lambda}$ is weakly playable, we obtain completeness as a corollary. ■

The previous line of argumentation can also be modified to yield a completeness result for **MaxPlay**, the smallest coalition logic containing **Play** and the following maximality axiom **Max**:

$$\neg[C]\neg\varphi \rightarrow [\overline{C}]\varphi$$

▶ THEOREM 3.15. **MaxPlay** is sound and complete with respect to the class of all maximal weakly playable coalition models: $\Sigma \models_{\mathtt{MaxPlay}} \varphi$ iff $\Sigma \vdash_{\mathtt{MaxPlay}} \varphi$.

PROOF. Let $P \subseteq \mathcal{P}(N)$ be such that it includes all $C \subseteq N$ such that $|C| < \frac{1}{2}|N|$ and in case $|N|$ is even also randomly picked C with $|C| = \frac{1}{2}|N|$ such that $C \in P$ iff $\overline{C} \notin P$. Then P is closed under subsets and for all $C \subseteq N$, $C \in P$ iff $\overline{C} \notin P$. Define the canonical model $\mathcal{C}^{\Lambda}$ as before, except that

$$sE^{\Lambda}_C X \text{ iff } \begin{cases} \exists \widehat{\varphi} \subseteq X : [C]\varphi \in s & \text{for } C \in P \\ \forall \widehat{\varphi} \subseteq \overline{X} : [\overline{C}]\varphi \notin s & \text{for } C \notin P \end{cases}$$

Then $\mathcal{C}^{\Lambda}$ is maximal and weakly playable. We show the proof of the most difficult playability conditions:

(1) Not $sE^{\Lambda}_{\emptyset}\emptyset$ implies $sE^{\Lambda}_C S^{\Lambda}$.

If not $sE^{\Lambda}_{\emptyset}\emptyset$, then $[\emptyset]\bot \notin s$ and by the axiom $(\top)$, $[C]\top \in s$. In case $C \in P$, we are done. Otherwise, assume by reductio that $[\overline{C}]\bot \in s$. Then by superadditivity, $[N]\bot \in s$, a contradiction.

(2) $sE^{\Lambda}_C\emptyset$ and $D \subseteq C$ imply $sE^{\Lambda}_D\emptyset$.

Four cases can be distinguished: (i) $C, D \in P$, the simple case.

(ii) $C, D \notin P$. Suppose by reductio that $\exists \widehat{\varphi} \subseteq S^{\Lambda}$ such that $[\overline{D}]\varphi \in s$ and furthermore $[\overline{C}]\top \notin s$. By the maximality axiom, $[C]\perp \in s$ and so by axiom ($\perp$), $[D]\perp \in s$. By superadditivity then, $[N]\perp \in s$, a contradiction.

(iii) $C \in P$, $D \notin P$. Then $[C]\perp \in s$ and so by axiom ($\perp$) also $[D]\perp \in s$. Now suppose by reductio that there is some $\widehat{\varphi}$ such that $[\overline{D}]\varphi \in s$. Then by superadditivity, $[N]\perp \in s$, a contradiction.

(iv) $D \in P$, $C \notin P$. Then for all $\widehat{\varphi}$ we have $[\overline{C}]\varphi \notin s$, and in particular, $[\overline{C}]\top \notin s$. By maximality, $[C]\perp \in s$ and hence also $[D]\perp \in s$.

> (3) Superadditivity: $sE^{\Lambda}_{C_1}X_1$ and $sE^{\Lambda}_{C_2}X_2$ imply $sE^{\Lambda}_{C_1 \cup C_2}(X_1 \cap X_2)$, provided $C_1 \cap C_2 = \emptyset$.

As the following table shows, in principle there are 8 possible cases, but only cases (1) and (4) need to be proved.

	C_1	C_2	$C_1 \cup C_2$	comments
(1)	P	P	P	
(2)	P	P	$\overline{P}$	implied by (1)
(3)	P	$\overline{P}$	P	impossible
(4)	P	$\overline{P}$	$\overline{P}$	
(5)	$\overline{P}$	P	P	impossible
(6)	$\overline{P}$	P	$\overline{P}$	same as (4)
(7)	$\overline{P}$	$\overline{P}$	P	impossible
(8)	$\overline{P}$	$\overline{P}$	$\overline{P}$	impossible

Due to the construction of P, three cases are impossible, for if one coalition is not in P, the union cannot be in P either. Similarly, it is impossible that both coalitions fail to be in P, for in that case they cannot be disjoint. Case (2) on the other hand is implied by (1) and the fact that $\exists \widehat{\varphi} \subseteq X : [C]\varphi \in s$ implies $\forall \widehat{\varphi} \subseteq \overline{X} : [\overline{C}]\varphi \notin s$ (using axioms `S` and `N⊥`).

Since case (1) is simple to take care of, consider case (4): $\exists \widehat{\varphi_1} \subseteq X_1 : [C_1]\varphi_1 \in s$ and using maximality, $\forall \widehat{\varphi_2} \subseteq \overline{X_2} : [C_2]\neg\varphi_2 \in s$. If we assume by reductio that $\exists \widehat{\delta} \subseteq \overline{X_1 \cap X_2}$ such that $[\overline{C_1 \cup C_2}]\delta \in s$, then since $\widehat{\delta \wedge \varphi_1} \subseteq \overline{X_2}$, $[C_2]\neg(\delta \wedge \varphi_1) \in s$. By two applications of the superadditivity axiom, $[C_1 \cup C_2](\varphi_1 \wedge \neg\delta) \in s$ and so $[N]\perp \in s$ so that s would be inconsistent. ∎

3.4.3 Extensive Games without Simultaneous Moves

▶ DEFINITION 3.16 (IND). **Ind** is the smallest coalition logic which includes `Play` and the following axiom (`D`)

$$[N]\varphi \to \bigvee_{i \in N} [i]\varphi.$$

Let $\Lambda \supseteq$ **Ind**. We again consider the set of maximally Λ-consistent sets of formulas S^Λ. The axiom states that whenever something is possible at all, there must be some individual who can bring it about. Note that in case $|N| = 1$, the axiom is a tautology, and hence also axiomatically, **Ind** = **Play**. As it turns out, a more general version of axiom (D) holds:

▶ LEMMA 3.17. For all $C \neq \emptyset$, $\vdash_{\texttt{Ind}} [C]\varphi \to \bigvee_{i \in C}[i]\varphi$.

PROOF. We show that every $s \in S^\Lambda$ contains the formula, call it δ. If $[\emptyset]\bot \in s$, $\delta \in s$ by axiom ($\bot$), so assume that $[\emptyset]\bot \notin s$.

Claim 1: For all $D \subseteq N$, $[D]\varphi \to \neg[\overline{D}]\neg\varphi \in s$. This follows from the superadditivity axiom (S).

Claim 2: For all $D \subseteq N$, $\neg[\overline{D}]\neg\varphi \to [D]\varphi \in s$. Assume by reductio that $\neg[\overline{D}]\neg\varphi, \neg[D]\varphi \in s$. Using coalition monotonicity and N-maximality, we have $[N]\varphi, [N]\neg\varphi \in s$. By axiom (D), $[i]\varphi, [i]\neg\varphi \in s$. Note that the same player here has to be able to force both φ and $\neg\varphi$, for two distinct players would be able to force $\bot$ using superadditivity. Now in case $i \in D$, coalition monotonicity gives us $[D]\varphi \in s$, otherwise we get $[\overline{D}]\neg\varphi \in s$, hence both cases lead to a contradiction.

To see that $\delta \in s$, assume by reductio that $[C]\varphi \in s$ and there is some $i \notin C$ such that $[i]\varphi \in s$ whereas for all $j \in C$ we have $\neg[j]\varphi \in s$. By claim 2, for all $j \in C$ we have $[N - \{j\}]\neg\varphi \in s$ and hence also $[N]\neg\varphi \in s$. Using axiom (D), there must be some $j \in N$ such that $[j]\neg\varphi \in s$. Since $[i]\varphi \in s$, we must have $i = j$, for otherwise the superadditivity axiom would allow us to derive $[\{i, j\}]\bot \in s$. Now since $i \notin C$, by coalition-monotonicity we have $[\overline{C}]\neg\varphi \in s$, and finally by claim 1, $[C]\varphi \notin s$, a contradiction. ■

The crucial lemma needed for the completeness proof is that every nonterminal $s \in S^\Lambda$ has a local dictator.

▶ LEMMA 3.18. For any $s \in S^\Lambda$ such that $\neg[\emptyset]\bot \in s$, there is some $d_s \in N$ such that for all formulas φ, if $[N]\varphi \in s$ then $[d_s]\varphi \in s$.

PROOF. Let $\neg[\emptyset]\bot \in s \in S^\Lambda$. Assume by reductio that for every $i \in N = \{1, \dots, n\}$ there is some formula φ_i such that $[N]\varphi_i \in s$ and $[i]\varphi_i \notin s$. By axiom (D), for every $i \in N$ we have $[f(i)]\varphi_i \in s$, where $f : N \to N$ such that $f(i) \neq i$. Let $f^k(m)$ denote k applications of f to m. When considering the sequence $1, f(1), f^2(1), \dots$, there must be a smallest cycle of length k such that for some m, $m = f^k(m)$ and no element in the sequence $m, f(m), \dots, f^{k-1}(m)$ occurs more than once. So if $C = \{m, f(m), \dots, f^{k-1}(m)\}$, we have $[f(m)]\varphi_m, \dots, [f^k(m)]\varphi_{f^{k-1}(m)} \in s$ and by superadditivity $[C](\varphi_m \wedge \dots \wedge \varphi_{f^{k-1}(m)}) \in s$. By the previous lemma, there must be some $i \in C$ such that $[i](\varphi_m \wedge \dots \wedge \varphi_{f^{k-1}(m)}) \in s$. However we cannot have $i = m$, since $[m]\varphi_m \notin s$, and similarly i cannot be $f^r(m)$ for some $r < k$, since $[f^r(m)]\varphi_{f^r(m)} \notin s$. Thus we obtained our contradiction. ■

The presence of a (local) dictator essentially turns coalition models into Kripke models. The completeness proof below should be seen as a translation of the completeness proof for normal modal logic into the coalitional setup. The same holds for the following existence lemma:

▶ LEMMA 3.19. For all $s \in S^\Lambda$ with $\neg[\emptyset]\bot$, $[C]\varphi \in s$ and $d_s \in C$, there is some $x \in \widehat{\varphi}$ such that for all $\delta \in x$, $[C]\delta \in s$.

PROOF. Let $x_0 = \{\varphi\} \cup \{\delta | [\overline{C}]\delta \in s\}$. Assume by reductio that x_0 is inconsistent, i.e., $\vdash_\Lambda \bigwedge\Delta \to \bot$ where w.l.o.g. $\Delta = \{\varphi, \delta_1, \ldots, \delta_n\}$. This would mean that $[C]\varphi, [\overline{C}]\delta_1, \ldots, [\overline{C}]\delta_n \in s$, and hence using the previous lemma, we have $[\emptyset]\delta_i \in s$ for $1 \leq i \leq n$. Using the superadditivity axiom, we then obtain $[C](\varphi \wedge \delta_1 \wedge \ldots \wedge \delta_n) \in s$. Since $\bigwedge\Delta$ implies $\bot$ this means that $[C]\bot \in s$ which contradicts our assumption that $[\emptyset]\bot \notin s$. This shows that x_0 is indeed consistent.

Consequently, x_0 can be extended to a maximally Λ-consistent set $x \supseteq x_0$ which satisfies the condition: take any $\delta \in x$ and assume by reductio that $[C]\delta \notin s$. Then $[\overline{C}]\neg\delta \in s$ and hence $\neg\delta \in x_0 \subseteq x$, contradicting the consistency of x. ■

▶ THEOREM 3.20. **Ind** is sound and complete with respect to the class of all weakly individualistic coalition models: $\Sigma \vdash_{\mathbf{Ind}} \varphi$ iff $\Sigma \models_{\mathbf{Ind}} \varphi$.

PROOF. For any coalition logic $\Lambda \supseteq$ **Ind**, we again consider the set of maximally Λ-consistent sets of formulas S^Λ, and we link to every state $s \in S^\Lambda$ a strategic game $G(s)$. If $[\emptyset]\bot \in s$, $G(s)$ will be an empty game with $G(s) = \{i \in N | [i]\bot \in s\}$. If $[\emptyset]\bot \notin s$, $G(s)$ will be a dictatorship with dictator d_s (provided by the previous lemma). The dictatorship will be captured by the accessibility relation $R^\Lambda \subseteq S \times S$ where d_s can choose from all the states t such that $sR^\Lambda t$. We define $sR^\Lambda t$ iff for all φ, $\varphi \in t$ implies $[N]\varphi \in s$. In order for this dictatorship to be well-defined, we need to check that there is some t such that $sR^\Lambda t$ which follows from the existence lemma.

Now we can construct the canonical model $\mathcal{C}^\Lambda = ((S^\Lambda, \{E^\Lambda_C | C \subseteq N\}), V^\Lambda)$ as usual, with $sE^\Lambda_C X$ iff coalition C is α-effective for X in $G(s)$. Using the existence lemma, the truth lemma can then be established, and the completeness proof continues as before. ■

3.5 Modal Logic as Coalition Logic

3.5.1 One-Player Games: Normal Modal Logic

In case the set of players N is a singleton, coalition models α-correspond to Kripke models: The strategic game associated with each state will trivially be a (possibly empty) dictatorship and hence the extensive game without simultaneous moves to which the coalition frame α-corresponds is a Kripke model, the local dictator being the same at every state. In terms of modalities, $\Box\varphi$ in modal logic corresponds to $[\emptyset]\varphi$ in coalition logic, and $\Diamond\varphi$ corresponds to $[N]\varphi$. Using this identification, the validities of Kripke models are precisely the validities of $1-$Play. This is also the point where it should become clear why in our definition of an empty strategic game we have insisted that some player must lose, for if we had allowed all players to win, there would have been terminal states where $\Diamond\bot$ is true, contrary to the semantics of modal logic. This correspondence for which we argued semantically can also be established axiomatically.

▶ Definition 3.21 (Modal Logic K). K is the smallest set of formulas which includes all propositional tautologies and the axioms of figure 3.3, and which is closed under the rules of Modus Ponens and Monotonicity (for $\Box$ only).

(1) $\Diamond\varphi \leftrightarrow \neg\Box\neg\varphi$	(2) $(\Box\varphi \wedge \Box\psi) \rightarrow \Box(\varphi \wedge \psi)$	(3) $\Box\top$

Figure 3.3: Axioms of K

The logic K is sound and complete with respect to the class of all Kripke models [32]. The complexity of the satisfiability problem over the class of all Kripke models is PSPACE-complete [20].

▶ Theorem 3.22. $\mathtt{K} = \mathtt{1-Play}$

Proof. By induction on the length of a derivation, it can easily be shown that $\mathtt{K} = \mathtt{1-Play}$ whose axioms we have given below, writing $[\emptyset]\varphi$ as $\Box\varphi$ and $[N]\varphi$ as $\Diamond\varphi$, and omitting those axioms which are consequences of the ones presented.

$\neg\Diamond\bot$	$\neg\Box\neg\varphi \rightarrow \Diamond\varphi$
$\Box\varphi \wedge \Box\psi \rightarrow \Box(\varphi \wedge \psi)$	$\Diamond\varphi \wedge \Box\psi \rightarrow \Diamond(\varphi \wedge \psi)$

∎

3.5.2 Two-Player Games: Non-Normal Modal Logic

Moving from 1-player games to 2-player games corresponds to moving from normal to non-normal modal logic. Non-normal modal logics describe *neighborhood models* $\mathcal{M} = ((S, N), V)$ which are almost like coalition models except that they contain only a single effectivity function, i.e., $N : S \rightarrow \mathcal{P}(\mathcal{P}(S))$. We assume here that for all $s \in S$, $N(s)$ is monotonic: $X \in N(s)$ implies $X' \in N(s)$ provided that $X \subseteq X'$. $\Box\varphi$ will be true at s if there is a neighborhood of s such that every state in that neighborhood makes φ true:

$$\mathcal{M}, s \models \Box\varphi \text{ iff } \{t \in S | \mathcal{M}, t \models \varphi\} \in N(s) \tag{3.1}$$

It can be shown that for these monotonic neighborhood models, the set of valid formulas is axiomatized by the following logic M:

▶ Definition 3.23 (Modal Logic M). $\mathtt{M}$ is the smallest set of formulas which contains all propositional tautologies and the Box-Diamond duality axiom $\Diamond\varphi \leftrightarrow \neg\Box\neg\varphi$, and which is closed under the rules of Modus Ponens and Monotonicity (for $\Box$ only).

The complexity of the satisfiability problem over the class of all monotonic neighborhood models is NP-complete [120].

Consider now the coalition logic $2\text{−}\mathtt{MaxPlay}$. The following theorem shows that the logic $\mathtt{M}$ is nothing but the individual fragment of $2 - \mathsf{MaxPlay}$. We will have more to say about individual fragments in general in section 3.7.

▶ Theorem 3.24. Identifying $[1]\varphi$ with $\Diamond\varphi$ and $[2]\varphi$ with $\Box\varphi$, we have $\mathtt{M} = 2\text{−}\mathtt{MaxPlay} \cap \{\varphi | \text{if } C \text{ occurs in } \varphi \text{ then } |C| = 1\}$.

Proof. $\mathtt{M} \subseteq 2\text{−}\mathtt{MaxPlay}$: The only non-obvious case is showing that $[1]\varphi \to \neg[2]\neg\varphi \in 2\text{−}\mathtt{MaxPlay}$ which follows from superadditivity. Conversely, for every formula φ containing only the two singleton coalitions, $\varphi \in 2\text{−}\mathtt{MaxPlay}$ implies $\varphi \in \mathtt{M}$. Given a neighborhood model $\mathcal{M} = ((S, N), V)$ satisfying $\neg\varphi$, construct a maximal weakly playable coalition model $\mathcal{M}' = ((S, E), V)$ such that $sE_{\{2\}}X$ iff $X \in N(s)$ and $sE_{\{1\}}X$ iff $\overline{X} \notin N(s)$. By corollary 2.13 of the previous chapter, this can be done, and hence $\mathcal{M}'$ satisfies $\neg\varphi$, showing that $\varphi \notin 2\text{−}\mathtt{MaxPlay}$. ■

3.6 Complexity II: Satisfiability

Probably the most important decision problem associated with modal logics is the satisfiability problem: Given a set of agents N and a modal formula φ, how much time or space does it take to find out whether or not the formula is satisfiable? We will look at the complexity of the satisfiability problem for the various coalition logics discussed.

The game-theoretic question associated with the satisfiability problem is the following: Given a certain specification of requirements as to what various groups of players can achieve after some moves, is there a game of a particular kind which satisfies this specification? Thus, whereas the model-checking problem captures game analysis, the satisfiability problem captures game synthesis. Questions of game synthesis play a role in implementation theory, where, e.g., one wants to know whether a particular set of rights can be decentralized into some game form. We will have more to say about this issue in chapter 5.

A few remarks concerning the size of the set of agents N and the role it plays in the following results. We are interested in satisfiability of a formula φ in a given model class K for a fixed set of agents N, which we shall abbreviate as $\mathit{Sat}(\varphi, \mathsf{K}, N)$. The results below are ordered according to the various model classes under investigation, and the results are uniform for all N, i.e., the results hold for all N. This does not mean, however, that the complexity of checking satisfiability is independent of the number of players. It only means that different sizes of N do not yield different complexity classes. Still, the degree of some polynomial which describes the running time will usually depend on $|N|$, as an inspection of the proofs of the results shows.

A different satisfiability question is the following: Given a class of models K and a formula φ, is there some $N \supseteq N(\varphi)$ for which φ is K-satisfiable ($Sat(\varphi, \mathsf{K})$)? In contrast to the earlier problem, N is not a fixed input parameter in this case. Using theorem 3.3, however, $Sat(\varphi, \mathsf{K})$ can easily be reduced to $Sat(\varphi, \mathsf{K}, N)$. For Mon, we have $Sat(\varphi, \mathsf{Mon})$ iff $Sat(\varphi, \mathsf{Mon}, N(\varphi))$, so we only need to check satisfiability for one particular N, the set of players which contains precisely the agents mentioned in φ. For $\mathsf{K} \in \{\mathsf{Play}, \mathsf{MaxPlay}, \mathsf{Ind}\}$, we have $Sat(\varphi, \mathsf{K})$ iff $Sat(\varphi, \mathsf{K}, N(\varphi))$ or $Sat(\varphi, \mathsf{K}, N(\varphi) \cup \{i\})$ for some $i \notin N(\varphi)$. Hence, we need to check at most two different N to find out whether there is any N such that $Sat(\varphi, \mathsf{K}, N)$. Consequently, the two decision problems $Sat(\varphi, \mathsf{K}, N)$ and $Sat(\varphi, \mathsf{K})$ are of the same order of complexity.

3.6.1 General Ability

General coalition models are multi-modal generalizations of neighborhood models. Since there is no coalitional interaction at all, none of the typical coalitional principles such as superadditivity is valid, and the complexity of the satisfiability problem is the same as in the case of non-normal modal logics. The result that classical modal systems have an NP-complete satisfiability problem can easily be lifted to its multi-modal generalization of Coalition Logic.

The heart of the algorithm relies on lemma 3.25 which reduces the satisfiability of φ to the satisfiability of certain combinations of subformulas of φ which have smaller modal depth.

Let $sf(\varphi)$ be the set of subformulas of φ, and let $Cl(\varphi) = sf(\varphi) \cup \{\neg\delta | \delta \in sf(\varphi)\}$. Note that $Cl(\varphi)$ is finite and that it is still closed under subformulas and their negations. A *semi-valuation* for φ is a function $v : Cl(\varphi) \to \{0, 1\}$ such that (1) $v(\psi) = 1$ iff $v(\neg\psi) = 0$, (2) $v(\psi_1 \vee \psi_2) = 1$ iff $v(\psi_1) = 1$ or $v(\psi_2) = 1$, (3) $v(\bot) = 0$, and (4) $v(\varphi) = 1$.

The following lemma provides the crucial link between satisfiability of a formula and satisfiability of its subformulas. The condition of the lemma captures outcome-monotonicity in terms of semi-valuations.

▶ LEMMA 3.25. A formula φ is satisfiable iff there exists a semi-valuation v for φ such that if $[C]\psi_1, [C]\psi_2 \in Cl(\varphi)$, $v([C]\psi_1) = 1$ and $v([C]\psi_2) = 0$ then $\psi_1 \wedge \neg\psi_2$ is satisfiable.

PROOF. From left to right, suppose φ is satisfiable in a coalition model $\mathcal{M} = ((S, E), V)$ at state $s \in S$. Then v defined by $v(\psi) = 1$ iff $\mathcal{M}, s \models \psi$ is a semi-valuation for φ, and it will satisfy the condition in virtue of $E(s)$ being outcome-monotonic.

From right to left, suppose we have a semi-valuation v satisfying the condition. This means that for every $[C]\psi_1, [C]\psi_2 \in Cl(\varphi)$ which meet the condition, there is a model $\mathcal{M}$ and a state s such that $\mathcal{M}, s \models \psi_1 \wedge \neg\psi_2$. Thus, we have a sequence of models $\mathcal{M}_1, \ldots, \mathcal{M}_n$ and a sequence of states $s_1, \ldots, s_n$ which serve as witnesses to the condition. We can assume w.l.o.g. that the universes of these models are pairwise disjoint, i.e., for all $\mathcal{M}_i = ((S_i, E_i), V_i)$ and $\mathcal{M}_j = ((S_j, E_j), V_j)$ with $i \neq j$ we have $S_i \cap S_j = \emptyset$. To simplify notation, we shall also use $V_i(\psi)$ for $\psi^{\mathcal{M}_i}$ when ψ is not atomic.

We will now construct a model $\mathcal{M} = ((S, E), V)$ satisfying φ which is roughly the union of the $\mathcal{M}_i$ models. For a new state s_0 which shall correspond to v, let $S = \{s_0\} \cup \bigcup_{i>0} S_i$. Let $V_0 : Cl(\varphi) \to \mathcal{P}(\{s_0\})$ be defined as $V_0(\psi) = \{s_0\}$ if $v(\psi) = 1$ and $\emptyset$ otherwise. Let $J : Cl(\varphi) \to \mathcal{P}(S)$ be defined by $J(\psi) = \bigcup_{i\geq 0} V_i(\psi)$. By construction, $J(\neg\psi) = S \setminus J(\psi)$ and $J(\psi_1 \vee \psi_2) = J(\psi_1) \cup J(\psi_2)$.

To complete the definition of our newly constructed model, let $V(p) = J(p)$ for $p \in \Phi_0$, and for $s_a \neq s_0$, let $E(s_a)$ be defined as follows:

$$s_a E_C X \text{ iff } \exists [C]\psi \in Cl(\varphi) : J(\psi) \subseteq X \text{ and } \mathcal{M}_a, s_a \models [C]\psi$$

For s_0, we use an analogous definition in terms of the semi-valuation v: $E(s_0)$ is defined as above, except that we replace the last conjunct by $v([C]\psi) = 1$. Clearly, $\mathcal{M}$ is outcome-monotonic.

Claim 1: If $[C]\psi \in Cl(\varphi)$, and $s_a E_C J(\psi)$ for $a > 0$, then $\mathcal{M}_a, s_a \models [C]\psi$. The proof uses the monotonicity of $\mathcal{M}_a$.

Claim 2: If $[C]\psi \in Cl(\varphi)$ and $s_0 E_C J(\psi)$, then $v([C]\psi) = 1$. The proof uses the main condition of the lemma.

To show that $\mathcal{M}, s_0 \models \varphi$, we show that for all $\psi \in Cl(\varphi)$, $V(\psi) = J(\psi)$. The proof is by induction on ψ, and base case and boolean cases are immediate. Let $s_a \in J([C]\psi)$. Depending on a, this either means that $\mathcal{M}_a, s_a \models [C]\psi$ or that $v([C]\psi) = 1$. In both cases, $s_a E_C J(\psi)$ holds and by induction hypothesis, $s_a E_C V(\psi)$ and hence $s_a \in V([C]\psi)$. Similarly for the converse direction, using claim 1 and 2. ∎

▶ THEOREM 3.26. The satisfiability problem for Mon is NP-complete.

PROOF. As for the lower bound, since Coalition Logic includes propositional logic, it must be NP-hard. As for the upper bound, we can write a nondeterministic algorithm which guesses a semi-valuation and checks for satisfiability recursively. If $|sf(\varphi)| = n$, the size of $Cl(\varphi)$ is $2n$ and hence at most $4n^2$ formulas of the form $\psi_1 \wedge \neg\psi_2$ need to be checked recursively. Note that it is sufficient to construct the closure $CL(\varphi)$ once for φ only, since $Cl(\psi_1 \wedge \neg\psi_2)$ does not contain any coalitional formulas $[C]\delta$ beyond those already present in $Cl(\varphi)$. The satisfiability of all these formulas $\psi_1 \wedge \neg\psi_2$ can be checked according to increasing modal depth using dynamic programming techniques [35]. Hence the algorithm runs in nondeterministic polynomial time. ∎

3.6.2 Extensive Games with Simultaneous Moves

As in the previous case, the lower bound is the easy direction.

▶ THEOREM 3.27. The satisfiability problem for Play is PSPACE-hard.

PROOF. First, recall that we have shown in section 3.5 that the normal modal logic $\mathtt{K} = \mathtt{1{-}Play}$. Second, if N is any nonempty set of players, for every formula φ of $\mathtt{1{-}Play}$, $\varphi \in \mathtt{1{-}Play}$ iff $\varphi^\circ \in \mathtt{Play}$, where φ° is the same as φ except that coalition N is substituted for coalition $\{1\}$. Inspecting the axioms, one sees that $\varphi \in \mathtt{1{-}Play}$ implies that $\varphi^\circ \in \mathtt{Play}$. For the other direction, if $\varphi \notin \mathtt{1{-}Play}$, there is a coalition model $\mathcal{M}_1$ satisfying $\neg\varphi$. $\mathcal{M}_1$ α-corresponds to a Kripke model $\mathcal{M}_K$, where player 1 makes the choices at every state. $\mathcal{M}_K$ can also be viewed as an N-player game in which the ability of coalition N coincides with the ability of player 1. As a consequence, the coalition model for N which α-corresponds to $\mathcal{M}_K$ will satisfy $\neg\varphi^\circ$.

As a result, there is a polynomial time translation from φ into a formula φ° such that $\varphi \in \mathtt{K}$ iff $\varphi^\circ \in \mathtt{Play}$, where the length of φ° is polynomial in the length of φ. Hence the satisfiability problem of $\mathtt{K}$ is polynomial time reducible to the satisfiability problem of Coalition Logic over Play. Since by Ladner's theorem [79], the satisfiability problem for $\mathtt{K}$ is PSPACE-hard, the satisfiability problem of Coalition Logic over weakly playable models is PSPACE-hard as well. ■

For the upper bound, as with general coalition models, we make use of a central lemma which allows us to tackle the satisfiability problem recursively. The only difference is that the conditions of the lemma are more complex since they need to capture playability rather than simply monotonicity. Also, subformulas are not enough anymore for the closure: Let

$$X_\varphi = sf(\varphi) \cup \{[N]\neg\delta | [\emptyset]\delta \in sf(\varphi)\} \cup \{[\emptyset]\neg\delta | [N]\delta \in sf(\varphi)\} \cup \{[C]\top, [C]\bot | C \subseteq N\}$$

and set $Cl(\varphi) = \{\bot, \top\} \cup X_\varphi \cup \{\neg\delta | \delta \in X_\varphi\}$. Note that $Cl(\varphi)$ is still finite and closed under subformulas and their negations. Since the following lemma is quite technical, we present it in some detail even though the structure of the proof follows that of lemma 3.25.

▶ LEMMA 3.28. A formula φ is satisfiable in a weakly playable coalition model iff there exists a semi-valuation v for φ such that the following five conditions hold:

1. $v([N]\bot) = 0$.
2. If $v([C]\bot) = 1$ and $C' \subseteq C$ then $v([C']\bot) = 1$.
3. If $[\emptyset]\psi_1, [C]\psi_2 \in Cl(\varphi)$ and $v([\emptyset]\psi_1) = v([C]\psi_2) = 0$, then $\psi_1 \vee \neg\psi_2$ is satisfiable.

4. If $[\emptyset]\psi_1, [N]\psi_2 \in Cl(\varphi)$ and $v([\emptyset]\psi_1) = v([N]\psi_2) = 0$, then $\neg\psi_1 \wedge \neg\psi_2$ is satisfiable.

5. If $[C]\psi, [C_1]\psi_1, \ldots, [C_k]\psi_k \in Cl(\varphi)$, $\forall i \neq j : C_i \cap C_j = \emptyset$, $C = \bigcup_i C_i$, $v([C]\psi) = 0$ and $\forall i : v([C_i]\psi_i) = 1$, then $\neg\psi \wedge \bigwedge_i \psi_i$ is satisfiable.

Proof. From left to right, suppose φ is satisfiable in a weakly playable coalition model $\mathcal{M} = ((S, E), V)$ at state $s \in S$. Then v defined by $v(\psi) = 1$ iff $\mathcal{M}, s \models \psi$ is a semi-valuation for φ, and it will satisfy the five conditions in virtue of $E(s)$ being weakly playable. Suppose, e.g., that $v([\emptyset]\psi_1) = v([N]\psi_2) = 0$, i.e., $\mathcal{M}, s \models \neg[\emptyset]\psi_1 \wedge \neg[N]\psi_2$. By N-maximality, $\mathcal{M}, s \models [N]\neg\psi_1$ and hence $\neg\psi_1^{\mathcal{M}} \not\subseteq \psi_2^{\mathcal{M}}$, i.e., $\neg\psi_1 \wedge \neg\psi_2$ must be satisfiable. Similarly for the other conditions.

From right to left, suppose we have a semi-valuation v satisfying the five conditions. This means that we have a sequence of models $\mathcal{M}_1, \ldots, \mathcal{M}_n$ and a sequence of states $s_1, \ldots, s_n$ which serve as witnesses to the five conditions. To simplify notation, we shall also use $V_i(\psi)$ for $\psi^{\mathcal{M}_i}$ when ψ is not atomic.

We will now construct a model $\mathcal{M} = ((S, E), V)$ satisfying φ which is roughly the union of the $\mathcal{M}_i$ models. For a new state s_0 which shall correspond to v, let $S = \{s_0\} \cup \bigcup_{i>0} S_i$. Let $V_0 : Cl(\varphi) \to \mathcal{P}(\{s_0\})$ be defined as $V_0(\psi) = \{s_0\}$ if $v(\psi) = 1$ and $\emptyset$ otherwise. Let $J : Cl(\varphi) \to \mathcal{P}(S)$ be defined by $J(\psi) = \bigcup_{i \geq 0} V_i(\psi)$.

To complete the definition of our newly constructed model, let $V(p) = J(p)$ for $p \in \Phi_0$. For $s_a \neq s_0$ and $C \neq N$, let $E(s_a)$ be defined as follows:

$$s_a E_C X \text{ iff } \begin{array}{l} \exists [C_1]\psi_1, \ldots, [C_k]\psi_k \in Cl(\varphi) : \\ (1)\ C = \bigcup_i C_i,\ (2)\ \forall i \neq j : C_i \cap C_j = \emptyset, \\ (3)\ \bigcap_i J(\psi_i) \subseteq X, \text{ and } (4)\ \mathcal{M}_a, s_a \models \bigwedge_i [C_i]\psi_i \end{array}$$

Note that in the above, k may be 1, in which case the right hand side reduces to $\exists [C]\psi \in Cl(\varphi)$ such that $J(\psi) \subseteq X$ and $\mathcal{M}_a, s_a \models [C]\psi$. For the set of all players, we define $s_a E_N X$ iff not $s_a E_\emptyset \overline{X}$.

For s_0, we use an analogous definition in terms of the semi-valuation v: For $C \neq N$, $E(s_0)$ is defined as above, except that we replace condition (4) by $\forall i : v([C_i]\psi_i) = 1$. Again, we define $s_0 E_N X$ iff not $s_0 E_\emptyset \overline{X}$.

$\mathcal{M}$ is weakly playable at every state s_a. For $s_a \neq s_0$, the playability conditions either hold by definition or they are essentially inherited from $\mathcal{M}_a$. For s_0, the first three conditions of the lemma are used.

Claim 1: If $C \neq N$, $[C]\psi \in Cl(\varphi)$, and $s_a E_C J(\psi)$ for $a > 0$, then $\mathcal{M}_a, s_a \models [C]\psi$. The proof uses superadditivity and monotonicity of $\mathcal{M}_a$.

Claim 2: If $C \neq N$, $[C]\psi \in Cl(\varphi)$, and $s_0 E_C J(\psi)$, then $v([C]\psi) = 1$. Suppose that $\exists [C_1]\psi_1, \ldots, [C_k]\psi_k \in Cl(\varphi)$ such that $\bigcap_i J(\psi_i) \subseteq J(\psi)$, and assume by reductio that $v([C]\psi) = 0$ while $\forall i : v([C_i]\psi_i) = 1$. Then using condition 5 of the lemma, $\neg\psi \wedge \bigwedge_i \psi_i$ is satisfiable, and so for some a we must have $\mathcal{M}_a, s_a \models \neg\psi \wedge \bigwedge_i \psi_i$, contradicting the fact that $\bigcap_i J(\psi_i) \subseteq J(\psi)$.

To show that $\mathcal{M}, s_0 \models \varphi$, we show that for all $\psi \in Cl(\varphi)$, $V(\psi) = J(\psi)$. For $C \neq N$, the argument is as before. For $C = N$, condition 4 of the lemma is used. Suppose $s_a \in V([N]\psi)$, i.e., $s_a \notin V([\emptyset]\neg\psi)$, and hence by the previous argument $s_a \notin J([\emptyset]\neg\psi)$. In the case that $s_a \neq s_0$, $\mathcal{M}_a, s_a \not\models [\emptyset]\neg\psi$, and N-maximality gives $s_a \in J([N]\psi)$. In case $s_a = s_0$, $v([\emptyset]\neg\psi) = 0$. Now if we assume by reductio that $v([N]\psi) = 0$, condition 4 of the lemma would make $\psi \wedge \neg\psi$ satisfiable, a contradiction, and so $v([N]\psi) = 1$, establishing $s_0 \in J([N]\psi)$. The other direction makes use of conditions 1 and 5. ■

▶ THEOREM 3.29. The satisfiability problem for Play is in PSPACE.

PROOF. Consider the following description of the satisfiability game/algorithm for formula φ:

Game Sat(φ)
1) construct $Cl(\varphi)$
2) $\exists$-player: choose an appropriate semi-valuation v for φ
3) if φ contains no modalities, $\exists$-player wins; otherwise:
4) $\forall$-player: choose a condition (3-5);
5) if condition = 3 then
    5.1) $\forall$-player: chooses $[\emptyset]\psi_1, [C]\psi_2 \in Cl(\varphi)$
        such that $v([\emptyset]\psi_1) = v([C]\psi_2) = 0$
    5.2) continue playing Sat($\psi_1 \vee \neg\psi_2$)
6) if condition = 4 then
    6.1) $\forall$-player: chooses $[\emptyset]\psi_1, [N]\psi_2 \in Cl(\varphi)$
        such that $v([\emptyset]\psi_1) = v([N]\psi_2) = 0$
    6.2) continue playing Sat($\neg\psi_1 \wedge \neg\psi_2$)
7) if condition = 5 then
    7.1) $\forall$-player: choose a subset $[C]\psi, [C_1]\psi_1, \ldots, [C_k]\psi_k \in Cl(\varphi)$
        such that the coalitions C_i are pairwise disjoint, $C = \bigcup_i C_i$,
        and for all i: $v([C_i]\psi_i) = 1$ and $v([C]\psi) = 0$
    7.2) continue playing Sat($\neg\psi \wedge \bigwedge_i \psi_i$)

Assuming that a player who cannot choose as instructed loses (e.g., $\exists$-player loses in step 2 if there is no semi-valuation for φ), we have defined a 2-player game. Note that step 2 involves the choice of an *appropriate* semi-valuation, i.e., a semi-valuation satisfying conditions (1) and (2) of lemma 3.28. By lemma 3.28, $\exists$-player has a winning strategy in this game iff φ is satisfiable. To analyze the time it takes to play the game, i.e., the maximal length of a play, let $n = |Cl(\varphi)|$. Note that the size of $Cl(\varphi)$ will be linear in $|sf(\varphi)|$, hence we can indeed analyze the complexity of the algorithm in terms of n. For the purposes of this algorithm, we allow for generalized conjunctions $\bigwedge_i \psi_i$ where $sf(\bigwedge_i \psi_i) = \{\bigwedge_i \psi_i\} \cup \bigcup_i sf(\psi_i)$.

The construction of $Cl(\varphi)$ and checking whether v is a semi-valuation takes time linear in n. At step 7.1, a maximum of n formulas $[C_i]\psi_i$ are chosen from $Cl(\varphi)$ so that checking all $v([C_i]\psi_i)$ takes time $O(n^2)$. Finally, the game is continued with $\neg\psi \wedge \bigwedge_i \psi_i$, where $|Cl(\neg\psi \wedge \bigwedge_i \psi_i)| \leq |Cl(\varphi)| = n$. Note that this recursive call reduces the modal depth of the formula by 1 until eventually φ contains no more modalities, hence the number of recursive calls is at most n.

Thus, at most n rounds of Sat are played, each round taking time polynomial in n. Since the size of each game configuration is also polynomial in n, doing backward induction on the game tree can be done in PSPACE by a depth-first search algorithm. In other words, since Sat(φ) contains a high-level description of an Alternating Turing Machine (see [31]) there is an alternating polynomial time algorithm for satisfiability checking, and given that APTIME = PSPACE, this means that there is a deterministic polynomial space algorithm for satisfiability. ■

Before turning to extensive games without simultaneous moves, we shall extend the previous complexity result to weakly playable models which are maximal.

▶ THEOREM 3.30. The satisfiability problem for MaxPlay is PSPACE-complete.

PROOF. For the lower bound, we can use the same argument used for Play, making use of the fact that the normal modal logic K is also a fragment of MaxPlay.

For the upper bound, we modify the argument for weakly playable models as follows: We let $X_\varphi = sf(\varphi) \cup \{[\overline{C}]\neg\delta | [C]\delta \in sf(\varphi)\} \cup \{[C]\top, [C]\bot | C \subseteq N\}$ in order to include the dual of every statement, not just for the extreme coalitions. Condition 4 of lemma 3.28 is then replaced by the following:

4. If $[C]\psi_1, [\overline{C}]\psi_2 \in Cl(\varphi)$ and $v([C]\psi_1) = v([\overline{C}]\psi_2) = 0$, then $\neg\psi_1 \wedge \neg\psi_2$ is satisfiable.

In the proof of the lemma, like in the completeness proof for MaxPlay, we partition N by taking a subset $P \subseteq \mathcal{P}(N)$ such that $C \in P$ iff $\overline{C} \notin P$ and P is closed under subsets. Arguments for $C \neq N$ become arguments for $C \in P$, similarly $C = N$ is turned into $C \notin P$. Finally the alternating algorithm will contain a slight modification due to the modified condition 4 of lemma 3.28. ■

3.6.3 Extensive Games without Simultaneous Moves

When dealing with extensive games without simultaneous moves, it proves advantageous to work directly with extensive games rather than with their associated coalition frames: Instead of constructing an individualistic coalition model which satisfies a formula, we will construct an extensive game without simultaneous moves.

Let $X_\varphi = sf(\varphi) \cup \{[\overline{C}]\neg\delta | [C]\delta \in sf(\varphi)\} \cup \{[\emptyset]\bot\}$ and set $Cl(\varphi) = \{\bot, \top\} \cup X_\varphi \cup \{\neg\delta | \delta \in X_\varphi\}$. Semi-valuations are defined a bit differently for the present case: In case $v([\emptyset]\bot) = 0$, the semi-valuation v also provides us with a player, namely the player who is to move at the given state. We shall denote this player by $v_p \in N$. In case $v([\emptyset]\bot) = 1$, we are dealing with a terminal state and the semi-valuation provides us with the set of players who win the game. This set shall be denoted by $v_W \subsetneq N$.

▶ LEMMA 3.31. A formula φ is satisfiable in **Ind** iff there exists a semi-valuation v for φ such that either (1) $v([\emptyset]\bot) = 1$ and $v([C]\psi) = 1$ iff $C \subseteq v_W$, or (2) $v([\emptyset]\bot) = 0$ and for every $[C]\psi \in E$, $\psi \wedge \bigwedge F$ is satisfiable, where $E = \{[C]\psi | v([C]\psi) = 1 \text{ and } v_p \in C\} \cup \{[\overline{C}]\neg\psi | v([C]\psi) = 0 \text{ and } v_p \notin C\}$ and $F = \{\psi | v([C]\psi) = 1 \text{ and } v_p \notin C\} \cup \{\neg\psi | v([C]\psi) = 0 \text{ and } v_p \in C\}$.

PROOF. If φ is satisfiable in an extensive game without simultaneous moves $\mathcal{M}$ at state s then the valuation at that state provides the appropriate semi-valuation.

For the converse, if $v([\emptyset]\bot) = 1$, we associate it with the empty strategic game where v_W is the set of winners. The extensive game will then consist of nothing else but this one empty strategic game, the valuation being given by v. If $v([\emptyset]\bot) = 0$, assume that the semi-valuation v satisfies the stated conditions, and for every $[C_i]\psi_i \in E$, assume that model $\mathcal{M}_i$ and state s_i are such that $\mathcal{M}_i, s_i \models \psi_i \wedge \bigwedge F$. Then the extensive game $\mathcal{M}$ which satisfies φ can be defined in the obvious way: It is the union of all the $\mathcal{M}_i$ plus an additional state s, the starting state of the game which will satisfy φ. The player who is to move at state s is v_p and the successors she can choose from are all the s_i. The valuation at s is defined according to v for the atomic propositions, and it can be shown by induction that the valuations of all $\psi \in Cl(\varphi)$ at s conform to v. ■

▶ THEOREM 3.32. The satisfiability problem for **Ind** is PSPACE-complete.

PROOF. For the upper bound, it remains to show that this lemma gives us a PSPACE-algorithm; again, as in the case of extensive games with simultaneous moves, we present a game which corresponds to an alternating Turing machine running in polynomial time. As before, let $n = |Cl(\varphi)|$. The $\exists$-player chooses a semi-valuation and provided $v([\emptyset]\bot) = 0$, the $\forall$-player chooses a $[C]\psi \in E$. The game then continues with the formula $\psi \wedge \bigwedge F$ which has smaller modal depth and size less than or equal to n. Hence, at most n rounds are played with each round taking time polynomial in n, thus we again have a PSPACE-algorithm.

As for PSPACE-hardness, observe that as in the previous subsection, `K` $=$ `1−Play` essentially forms a fragment of `Ind` via the translation procedure of theorem 3.27, and hence the theorem extends to extensive games without simultaneous moves as well. ■

3.7 The Individual Fragment of Coalition Logic

One of the general theses of this thesis is that reasoning about multi-agent systems is markedly different from reasoning about individual agents. In fact, we claim that this holds true already in the simple framework of coalition models which are intended to model nothing more than the agents' ability to change the world. One of the ways in which we shall give formal content to this claim is in terms of expressiveness.

The logical language just introduced contains a modality $[C]\varphi$ for every coalition $C \subseteq N$. In order to study the difference between reasoning about individuals and reasoning about groups, we shall distinguish the full language of Coalition Logic from its individual fragment:

▶ DEFINITION 3.33 (INDIVIDUAL FRAGMENT). The *individual fragment* of Coalition Logic is the set of formulas in which only singleton coalitions occur.

Showing differences between full Coalition Logic and its individual fragment is thus one way to learn how coalitional reasoning differs from individual reasoning.

3.7.1 Expressiveness

▶ DEFINITION 3.34 (EXPRESSIVENESS). Given two logical languages L_1 and L_2 whose semantics are defined with respect to the same class of models. L_1 is *at least as expressive as* L_2 over the class of models K iff for every formula $\varphi_2 \in L_2$ there is some K-equivalent formula $\varphi_1 \in L_1$, i.e., the truth values of φ_1 and φ_2 agree at every state of every model in K.

The notions of expressive equivalence and strictly greater expressive power are then defined in the obvious way. Using this terminology, we can show that Coalition Logic is strictly more expressive than its individual fragment on all the model classes discussed except the class of extensive games without simultaneous moves. This result is the logical analogue of theorem 2.35 of the previous chapter which implied that only in the case of extensive games without simultaneous moves is coalitional effectivity determined by individual effectivity.

▶ THEOREM 3.35. Over Mon, Play and $\mathsf{MaxPlay}$, Coalition Logic is more expressive than its individual fragment, provided that $|N| > 1$. Over Ind, the individual fragment is equally expressive.

PROOF. First we prove that full coalition logic is strictly more expressive than its individual fragment over maximal weakly playable coalition models. From this the other results for Mon and Play follow immediately. For $N = \{1, 2\}$, let $\mathcal{M} = ((\{s, t\}, E), V)$ and $\mathcal{M}' = ((\{s, t\}, E'), V)$ where $V(p) = \{t\}$. In $\mathcal{M}$, we associate the left strategic game G_1 of figure 2.2 to s and let $E(s) = E^{\alpha}_{G_1}$. In

$\mathcal{M}'$, we associate the right strategic game G_2 to s and let $E'(s) = E^{\alpha}_{G_2}$. Recall that both of these games are maximal. It is an easy proof by induction that for every formula φ of the individual fragment $\varphi^{\mathcal{M}} = \varphi^{\mathcal{M}'}$. This is as it should be, for at state s, the power of the two individuals is the same, they can force all supersets of $\{s\}$. Coalition $\{1, 2\}$ on the other hand can force more, in particular $\mathcal{M}, s \models [\{1, 2\}]p$ while $\mathcal{M}', s \models \neg[\{1, 2\}]p$.

As for extensive games without simultaneous moves, coalitional modalities do not add any expressive power to individual modalities. We can define a translation function $^\circ$ which maps formulas of Coalition Logic into the individual fragment, mapping boolean formulas simply componentwise (e.g., $(\varphi \vee \psi)^\circ = \varphi^\circ \vee \psi^\circ$) and $([C]\varphi)^\circ$ for $C \neq \emptyset$ into

$$([\bot] \to \bigwedge_{i \in C}[i]\bot) \wedge (\neg[\bot] \to \bigvee_{i \in C}[i]\varphi^\circ).$$

Intuitively, in case we are at a terminal state, all members of C must be winners, in case we are at a non-terminal state, there must be some member of C which can force φ. In case $C = \emptyset$, $([C]\varphi)^\circ$ is equivalent to $[\bot] \vee \neg \bigvee_{i \in N}[i]\neg\varphi^\circ$, stating that either we are at a terminal state or none of the players can force $\neg\varphi^\circ$. ∎

3.7.2 Complexity

Given the increased expressive power of coalitional formulas at least over Mon, one might guess that reasoning about coalitions will also be more complex than reasoning about individuals. We will consider the complexity of the satisfiability problem for each of the relevant model classes in turn.

Regarding Mon, it is clear that no complexity difference between Coalition Logic and its individual fragment will emerge, since individuals are not any different from coalitions due to the lack of extra assumptions on coalitional effectivity. Formally, the satisfiability problem cannot be less complex than NP since propositional logic is still part of the individual fragment. Hence, the complexity of the satisfiability problem for the individual fragment over Mon is NP-complete.

For Play, the situation is markedly different. The PSPACE lower bound was established by noticing that the normal modal logic K was a fragment of Play. Since the translations of the $\Box$ and $\Diamond$ modalities was in terms of $[\emptyset]$ and $[N]$, however, this argument cannot be utilized anymore to establish a PSPACE lower bound for the individual fragment. In fact, one can show that the satisfiability problem can be solved in NP:

▶ **Theorem 3.36.** The satisfiability problem for the individual fragment over Play is NP-complete in case $|N| > 1$.

Proof. Since NP-hardness is obvious, we only need to provide an algorithm which runs in nondeterministic polynomial time. As before, the algorithm relies on a recursive satisfiability lemma which is based on theorem 2.11 of the previous chapter and hence must be limited to cases where $|N| > 1$.

As before, let $sf(\varphi)$ be the set of subformulas of φ, and let

$$X_\varphi = sf(\varphi) \cup \{[i]\top, [i]\bot | i \in N\} \text{ and } CL(\varphi) = X_\varphi \cup \{\top, \bot\} \cup \{\neg\delta | \delta \in X_\varphi\}.$$

Again, a semi-valuation for φ is a function $v : Cl(\varphi) \to \{0, 1\}$ such that (1) $v(\psi) = 1$ iff $v(\neg\psi) = 0$, (2) $v(\psi_1 \vee \psi_2) = 1$ iff $v(\psi_1) = 1$ or $v(\psi_2) = 1$, (3) $v(\bot) = 0$, and (4) $v(\varphi) = 1$.

Now via the same construction that was used in previous arguments, we can show that a formula φ is satisfiable iff there exists a semi-valuation v for φ such that

1. If $[i]\psi_1, [i]\psi_2 \in Cl(\varphi)$, $v([i]\psi_1) = 1$ and $v([i]\psi_2) = 0$ then $\psi_1 \wedge \neg\psi_2$ is satisfiable.
2. If there is some $j \in N$ such that $v([j]\top) = 0$, then for all $i \in N$, if $[i]\psi_1, [i]\psi_2 \in Cl(\varphi)$, $v([i]\psi_1) = 1$ and $v([i]\psi_2) = 0$ then $\neg\psi_1 \wedge \psi_2$ is satisfiable.
3. If $N = \{1, \ldots, n\}$ and $[1]\psi_1, \ldots, [n]\psi_n \in CL(\varphi)$ such that for all $i \in N$, $v([i]\psi_i) = 1$, then $\bigwedge_i \psi_i$ is satisfiable.

The first condition expresses monotonicity and the other two conditions correspond to conditions (2) and (1) of definition 2.10 of the previous chapter. Again, we can utilize this lemma to write a nondeterministic algorithm which guesses a semi-valuation and checks for satisfiability recursively, as in the case of Mon. If the size of $|Cl(\varphi)|$ is n, we need to check at most n^2 formulas of the form $\psi_1 \wedge \neg\psi_2$ and at most $n^{|N|}$ formulas of the form $\bigwedge_{i \in N} \psi_i$ recursively, in order of increasing modal depth. Hence, we again, have an algorithm running in nondeterministic polynomial time, the only difference with the case of Mon being that the polynomial will be of a much higher degree if $|N|$ is big. ∎

We have considered MaxPlay for the case where $|N| = 2$ already when we considered modal logic in section 3.5 (the complexity was also NP-complete), and so we turn to the class of weakly individualistic models Ind. Note first that we know from the previous subsection that coalitions do not add any expressive power over individualistic models. This does not imply, however, that there will be no complexity difference between the full Coalition Logic and its individual fragment. Rather, as a consequence of this result, any such complexity difference would have to be attributed to a difference in notational succinctness, not to increased expressive power. Analyzing the translation from Coalition Logic into its individual fragment shows, however, that the improvement in conciseness is only marginal.

▶ THEOREM 3.37. The satisfiability problem for the individual fragment over Ind is PSPACE-complete.

PROOF. (We only need to show PSPACE-hardness.) Theorem 3.35 provided us with a translation function $^\circ$ from Coalition Logic into its individual fragment such that φ and φ° are equivalent over individualistic models. While the length of φ° in symbols may be exponentially longer than the length of φ, this is not the case when comparing $|sf(\varphi)|$ to $|sf(\varphi^\circ)|$. Consider, e.g., the coalitional formula $[\{1,2\}][\{1,2\}]p$. Its translation $([\{1,2\}][\{1,2\}]p)^\circ$ is identical to

$$([\bot] \to ([1]\bot \wedge [2]\bot)) \wedge (\neg[\bot] \to ([1]([\{1,2\}]p)^\circ \vee [2]([\{1,2\}]p)^\circ)).$$

Note that while $([\{1,2\}]p)^\circ$ is indeed duplicated in $([\{1,2\}][\{1,2\}]p)^\circ$, we are dealing with two occurrences of the same subformula which means that in terms of subformulas we do not obtain an exponential blow-up. More generally, looking at the structure of the formula $([C]\psi)^\circ$, we can see that $|sf(([C]\psi)^\circ)| = c_C + |sf(\psi^\circ)|$ where c_C depends only on $|C|$ and hence we know $|sf(([C]\psi)^\circ)| \leq c_N + |sf(\psi^\circ)|$. Hence, the size of $sf(\varphi^\circ)$ will be polynomial in the size of $sf(\varphi)$, and consequently the satisfiability problem for full Coalition Logic over Ind can be effectively reduced to the satisfiability problem of the individual fragment. ■

3.8 Discussion

Throughout this chapter we have focused our attention on 4 different classes of coalition models which are ordered by inclusion as follows:

$$\mathsf{Mon} \supset \mathsf{Play} \supset \mathsf{MaxPlay} \supset \mathsf{Ind}$$

By the characterization results obtained, these classes represent effectivity in respectively general monotonic coalition models, extensive games with simultaneous moves, determined extensive games with simultaneous moves and extensive games without simultaneous moves. Programs or 1-player games are represented by 1−Play = 1−Ind. This chapter has mainly investigated questions of complexity, expressiveness, and axiomatization. A metatheoretic question we did not pose concerns the *finite model property* (fmp). We conjecture that via filtration one can show that all the model classes mentioned have the fmp. A related question concerns the poly-size or *small model property* (smp), stating that every formula φ which is K-satisfiable is also K-satisfiable in a model of size polynomial in $|\varphi|$. Note that since `Ind`, `Play`, and `MaxPlay` all contain the normal modal logic `K` as a fragment, the associated model classes cannot have the small model property. For Mon on the other hand, the question is still open.

3.8.1 Modal Logic and Complexity

Comparing normal and non-normal modal logic yields a precise description of the difference between individual ability in programs and games. If we look at `K` and

M purely from the perspective of player 1, we formulate all axioms in terms of $\Diamond$, then the axioms

$$\Diamond(\varphi \vee \psi) \to (\Diamond\varphi \vee \Diamond\psi) \text{ and } \neg\Diamond\bot$$

provide a complete description of the difference between programs (where these axioms hold) and determined 2-player games (where the axioms do not hold). Note that the second axiom shows that it is not true that player 1 always has more power in a solitaire game than in a 2-player game (as suggested by the first axiom), since 2-player games allow for terminal positions where player 1 wins, whereas all terminal positions of solitaire games are losses for player 1.

To summarize the coalitional view of modal logic, normal and non-normal modal logic differ in two respects: First, regarding the underlying semantic structures which are described: normal modal logic describes one-player games whereas non-normal modal logic describes determined 2-player games. Second, regarding the expressive power of the language used: non-normal modal logic is non-coalitional in that it allows one to talk about the ability of individual agents (i.e., singleton coalitions) only. As a result, the fact that the complexity of the satisfiability problem for M is NP-complete fits perfectly well into the general picture summarized in figure 3.5. Normal modal logic on the other hand is a bit of a hybrid case, since it can express the ability of the empty coalition, yet the ability of the empty coalition is completely determined by the ability of the one player. Still, such simple coalitional content is sufficient to yield a satisfiability problem which is PSPACE-complete.

Taking another look at the proof of lemma 3.25, we can get some intuition for the complexity difference between K and M. Since Coalition Logic over all coalition models is nothing but multi-modal M, the analogue of lemma 3.25 for the modal logic M is as follows:

> A formula φ is satisfiable iff there exists a semi-valuation v for φ such that if $\Box\psi_1, \Box\psi_2 \in sf(\varphi)$, $v(\Box\psi_1) = 1$ and $v(\Box\psi_2) = 0$ then $\psi_1 \wedge \neg\psi_2$ is satisfiable.

As mentioned, this lemma gives us a decision procedure in NP: If $|sf(\varphi)| = n$, at most $O(n^2)$ formulas of the form $\psi_1 \wedge \neg\psi_2$ need to be checked, and this can be done according to modal depth so that we end up with a nondeterministic algorithm which runs in polynomial time. Let us compare this to the situation with K. The analogous recursive satisfiability lemma for K is the following:

> A formula φ is satisfiable iff there exists a semi-valuation v for φ such that if $\Box\psi_1, \ldots, \Box\psi_k, \Box\delta \in sf(\varphi)$, for all i: $v(\Box\psi_i) = 1$, and $v(\Box\delta) = 0$, then $\neg\delta \wedge \bigwedge_i \psi_i$ is satisfiable.

If we want to apply the same technique we used for M, we would have to check according to increasing modal depth all formulas of the form $\neg\delta \wedge \bigwedge_i \psi_i$, but since i can vary, there are essentially 2^n such formulas, since every subset of $sf(\varphi)$ gives rise to one formula. As a consequence, the nondeterministic algorithm does not run in polynomial but in exponential time.

For the coalition logic `Play`, the superadditivity axiom gives rise to a similar sequence of formulas $[C]\psi, [C_1]\psi_1, \ldots, [C_k]\psi_k \in Cl(\varphi)$ in lemma 3.28 which prevents an NP-algorithm. For its individual fragment on the other hand, in case $|N| > 1$, the sequence of modalities $[1]\psi_1, \ldots, [n]\psi_k \in CL(\varphi)$ is bounded by $k = |N|$, thus yielding at most $n^{|N|}$ formulas of the form $\bigwedge_i \psi_i$ which need to be checked.

3.8.2 Programs vs. Games

Questions of axiomatization and complexity were raised and answered for all the 4 classes mentioned. Has this investigation yielded any insights into differences between these various semantic classes and, as a possible consequence, between programs and games? In all cases, a complete axiomatization was obtained by a straightforward translation of the model-theoretic conditions into formulas of coalition logic. Note that the axioms of figure 3.2 provide a uniform axiomatization for programs and games with simultaneous moves, i.e., the only difference in axiomatization is the number of players for which these axiom schemes are instantiated. For extensive games without simultaneous moves (with more than 1 player), an additional axiom needs to be added, however.

In terms of complexity, note that model checking can be done in time linear in the size of the model and the size of the formula for both programs and games. Note, however, that for certain complex extensive games with simultaneous moves, the representation size will be much larger than for programs, and hence the results are not immediately comparable.

Regarding the complexity of the satisfiability problem, a number of interesting conclusions can be drawn. The relevant results are summarized in figure 3.4.

	Play	Ind
$\|N\| = 1$	PSPACE	PSPACE
$\|N\| > 1$	NP	PSPACE

Figure 3.4: Complexity results for the satisfiability problem: the individual fragment of Coalition Logic for different numbers of players and different classes of games. (Note that for $|N| = 1$, Play = Ind.)

Suppose that we are given some behavioral specification φ formulated in the individual fragment of Coalition Logic. If φ only contains the singleton coalition $\{i\}$, i.e., if φ only specifies the ability of the single player i, we can ask whether there is a program/process which satisfies this specification. This is the satisfiability problem for 1−Play which we have seen to be PSPACE-complete.

Similarly, finding out whether there is a 2-player extensive game without simultaneous moves satisfying φ will be PSPACE-complete. On the other hand, finding a satisfying extensive game *with* simultaneous moves is NP-complete. Thus, if NP $\neq$ PSPACE then game synthesis can actually be easier than program synthesis, depending on the type of game. Put differently, what is important is not so much the number of players but the presence or absence of simultaneous moves. Put differently, the issue is whether we are working in an environment of perfect information or not.

To get an intuitive understanding of these results, note that 1-player games are structurally almost identical to extensive games without simultaneous moves with more than 1 player. The only difference lies in the player assigned to each position. Extensive games with simultaneous moves on the other hand allow for more freedom in satisfying a given specification. At every state, we need to find a strategic game satisfying certain properties. This strategic game may be a dictatorship, but much more options are available. Put differently, a local effectivity function satisfying the specification demands needs to fulfill fewer requirements in order to be legal, i.e., α-corresponding to a strategic game.

3.8.3 Individuals vs. Coalitions

In the previous section we have introduced the individual fragment of Coalition Logic in order to compare reasoning about individuals to reasoning about coalitions. The expressiveness result shows that in extensive games without simultaneous moves, reasoning about coalitions is no different from reasoning about individuals except for allowing a more concise notation. In games with simultaneous moves and general coalition models, however, the full coalitional language has an increased expressive power. As to the complexity of the satisfiability problem, we have seen that Coalition Logic and its individual fragment differ only over the class of weakly playable models. The table of figure 3.5 summarizes these results:

	Mon	Play	Ind
Complexity of full Coalition Logic	NP	PSPACE	PSPACE
Complexity of individual fragment	NP	NP	PSPACE
Is individual fragment less expressive?	yes	yes	no

Figure 3.5: Comparison between Coalition Logic and its individual fragment: expressiveness and complexity of the satisfiability problem (we assume that $|N| > 1$).

One conclusion to draw from these results is that among the classes of models considered, Play is the essential class from a coalitional perspective, for it is the only class where the difference in expressivity which nontrivial coalitional formulas can make also manifests itself in terms of complexity. In other words, over the class of weakly playable models, Coalition Logic is really different from a logic which only talks about the ability of individual agents. Note, however, that Play is not the only class with this property, since, e.g., the results for 2−MaxPlay are in fact the same as those for Play.

3.8.4 Epistemic Logic vs. Coalition Logic

As a final remark, it is worth mentioning an epistemic analogue of coalition formation. When investigating various non-normal epistemic logics in [120], the author observes a complexity difference which hinges on the presence of the formula

$$K\varphi \wedge K\psi \rightarrow K(\varphi \wedge \psi) \tag{3.2}$$

where $K\varphi$ should be read as "the agent knows that φ". Among the various epistemic systems investigated, logics which do not contain this principles have their satisfiability problem in NP, whereas those containing (3.2) are in PSPACE. While in the latter case, no lower bound is proved, it is conjectured that this principle which formalizes an agent's ability to epistemically combine facts, i.e., to reason about the world, causes the (conjectured) complexity increase. As can be seen from the form that the superadditivity axiom takes in `1−Play`, superadditivity is the game-theoretic analogue of the epistemic principle (3.2): While the direct analogue of principle 3.2 would be $[C]\varphi \wedge [C]\psi \rightarrow [C](\varphi \wedge \psi)$ which is characteristically not valid in Coalition Logic, superadditivity $[C_1]\varphi \wedge [C_2]\psi \rightarrow [C_1 \cup C_2](\varphi \wedge \psi)$ is. In a multi-agent setting in which we want to express group knowledge, superadditivity would allow groups of agents to join their knowledge, formalized by the epistemic version of superadditivity $K_1\varphi \wedge K_2\psi \rightarrow K_{\{1,2\}}(\varphi \wedge \psi)$. Thus. $K_{\{1,2\}}\varphi$ expresses that agents 1 and 2 together have *implicit* knowledge of φ, the knowledge of φ is distributed among the agents and can be made explicit, e.g., through communication within the group [58].

Instead of expressing the ability of an agent or a group of agents to combine facts, superadditivity expresses the ability of agents to combine their strategies when forming a coalition. And in the case of Coalition Logic, it is possible to locate a complexity increase precisely in this ability to combine strategies, since one can show that for Coalition Logic without superadditivity, the satisfiability problem is NP-complete.

3.9 Bibliographic Notes

Parts of the material in this chapter has been published in [98, 104]. The classes of strongly playable and strongly individualistic coalition models were axiomatized already in [98], whereas the present chapter has focused on weak playability and weak individualism. For strongly playable coalition models, the complexity of the satisfiability problem was also first investigated in [98].

For a textbook on normal modal logic, see [20] which also contains results on the complexity of the satisfiability problem for various normal modal systems. A book which also treats non-normal modal systems is [32]. Coalition Logic in its most general form is the multi-modal equivalent of non-normal monotonic modal logic (terminology from [32]). For the class of general coalition models, axiomatic completeness essentially follows already from the results in [32], where it is proved that the non-normal modal logic $\mathtt{M}$ is complete for the class of monotonic neighborhood models.

The complexity results of section 3.6 follow the general approach of [120], where the complexity of various non-normal epistemic logics is studied. The result on the complexity of model checking in Coalition Logic (theorem 3.7) is an adaptation of the analogous result for normal modal logic in [121].

Chapter 4

Extended Coalition Logic

When Coalition Logic is used to reason about coalitional ability in extensive games, for instance, the formula $[C]\varphi$ holds in case coalition C can bring about φ in one move. Thus, Coalition Logic is a logic for reasoning about local coalitional ability in games. Naturally, many essential properties of such games cannot be expressed in terms of local effectivity alone, most notably the property of a coalition having a winning strategy in the extensive game as a whole. For this reason, the present chapter extends the language of Coalition Logic with an additional modality to talk about ability in the long run, i.e., about what coalitions can bring about eventually.

We will start by discussing various kinds of effectivity in the long run and their interrelationship. In most cases, Extended Coalition Logic, formally defined in section 4.2, will be able to express all of these different notions through one single modality. The complexity of the model checking problem and the expressive power of the richer language of Extended Coalition Logic will be investigated in section 4.3. Furthermore, we will discuss the relationship between Extended Coalition Logic and Alternating Temporal Logic (section 4.4) and work in distributed artificial intelligence (section 4.5).

4.1 Ability in the Long Run

Given a coalition frame $\mathcal{F} = (S, \{E_C | C \subseteq N\})$ which contains information about effectivity at every state, we can also investigate effectiveness in the long run. The two basic notions of long-term effectivity we shall use are goal maintenance and (eventual) goal achievement. A coalition C can eventually achieve a set of states X provided that it has a strategy which establishes X after some finite number of moves which does *not* need to be fixed before the game starts. Using a fixpoint construction (see appendix A for background material on fixpoints), we can formally define eventual goal achievement.

▶ **Definition 4.1 (Eventual Goal Achievement).** Given a coalition frame $\mathcal{F} = (S, \{E_C | C \subseteq N\})$, the *eventual goal achievement* effectivity function E^* is defined as

$$E^*_C(X) = \mu Y.X \cup (\overline{E_\emptyset(\emptyset)} \cap E_C(Y)).$$

Note that we give fixpoint operators lowest precedence, so that the scope of a fixpoint operator extends as far to the right as possible.

Intuitively, $sE^*_C X$ holds precisely when at state s coalition C can bring about X in the long run. The term $\overline{E_\emptyset(\emptyset)}$ guarantees that states will not be included simply because they are terminal states where all members of C win. As a consequence of this definition of E^*, $E_C(X) \not\subseteq E^*_C(X)$, for at a terminal state $s \notin X$ where C wins, $sE_C X$ but not $sE^*_C X$. What we do have is $E_C(X) \cap \overline{E_\emptyset(\emptyset)} \subseteq E^*_C(X)$ and $E_C(X) \subseteq E^*_C(X \cup E_C(\emptyset))$.

Turning now toward goal maintenance, coalition C can maintain a set of states X provided that it has a strategy which will guarantee that every future position of the game play will be in X.

▶ **Definition 4.2 (Goal Maintenance).** Given a coalition frame $\mathcal{F} = (S, \{E_C | C \subseteq N\})$, the *goal maintenance* effectivity function $E^\times$ is defined as

$$E^\times_C(X) = \nu Y.X \cap (E_\emptyset(\emptyset) \cup E_C(Y)).$$

Intuitively, $sE^\times_C X$ holds precisely when at state s coalition C can maintain X indefinitely. Now the additional union with $E_\emptyset(\emptyset)$ is necessary to guarantee that states will not be excluded simply because they are terminal states where some members of C lose.

The definitions of effectivity in the long run are general in that they do not presuppose the coalition frame to satisfy any additional properties such as weak playability. On the other hand, the example given to motivate certain aspects of the definition assumed an application to extensive games, and this is in fact the class of models we shall look at when dealing with some applications in chapter 5.

Recall from section 2.4.2 that coalition frames can be used to model extensive games as well as game forms. For extensive game forms, $sE^*_C X$ will hold in case coalition C can eventually achieve X, no matter how the empty games at terminal states are defined (thanks to the $\overline{E_\emptyset(\emptyset)}$ term in the definition of E^*). For extensive games (with significant payoff information), we can additionally consider whether coalition C has a winning strategy, expressed by $E^*_C(E_C(\emptyset))$, or whether it can achieve a win or X, expressed by $E^*_C(X \cup E_C(\emptyset))$. Note, however, that due to the possibility of infinite runs, having a winning strategy can be interpreted in two ways: In the (for C) best case, coalition C has a strategy which guarantees that game play will terminate in a state which is a win for all members of C. This is the interpretation corresponding to $E^*_C(E_C(\emptyset))$. If C is somewhat less fortunate, however, it may only have a strategy which guarantees that *if* game play terminates, *then* all members of C win. As the example of a game where all plays are infinite illustrates, the two kinds of strategies are not equivalent. This distinction between strong and weak winning strategies is not usually made in game theory, where it is assumed that also infinite runs generate payoffs to the players [93]. Note also that in order to distinguish strong from weak winning strategies, it is crucial that our semantic model allows for terminal states (compare this with Alternating Temporal Logic, discussed in section 4.4).

Total vs. Partial Terminal Effectivity

We shall formalize the distinction between strong and weak strategies as the difference between total and partial terminal effectivity. While the notion of eventual goal achievement formalizes what it means to bring about a goal at *some* point in the future, we are often more interested in what terminal outcomes or outcome states a coalition can achieve. If we are dealing with an extensive game, this question asks for whether a coalition has a strategy which will yield a win for all its members. For an extensive game form on the other hand, we want to know which sets of terminal states can be achieved.

Given a coalition frame $\mathcal{F} = (S, \{E_C | C \subseteq N\})$, we define *total terminal effectivity* as

$$E^t_C(X) = E^*_C(E_\emptyset(\emptyset) \cap X) = \mu Y.(E_\emptyset(\emptyset) \cap X) \cup (\overline{E_\emptyset(\emptyset)} \cap E_C(Y)).$$

The set $E^t_C(X)$ includes the set $E_\emptyset(\emptyset) \cap X$ which denotes the set of terminal states which are in X. Furthermore, it includes the nonterminal states from which C can achieve one of those terminal states in X, and so on. The weaker version of total terminal effectivity is *partial terminal effectivity* defined as

$$E^p_C(X) = E^\times_C(\overline{E_\emptyset(\emptyset)} \cup X) = \nu Y.(\overline{E_\emptyset(\emptyset)} \cup X) \cap (E_\emptyset(\emptyset) \cup E_C(Y)).$$

The goal to be maintained here is that "if the present state is terminal then it is in X". If we are dealing with an extensive game which has infinite runs, we can then distinguish a strong winning strategy $E^t_C(E_C(\emptyset))$ from a weak winning strategy $E^p_C(E_C(\emptyset))$.

For extensive games and game forms, the four possible instantiations of the scheme "some/all plays of the game are finite/infinite" are of particular interest. All of them can be defined in terms of the notions just introduced. For weakly playable frames $\mathcal{F} = (S, \{E_C | C \subseteq N\})$, we have

$E^t_\emptyset(S) = \mu Y.E_\emptyset(Y)$ holds iff all plays of the game are finite.

$E^t_N(S) = \mu Y.E_\emptyset(\emptyset) \cup E_N(Y)$ holds iff at least one play of the game is finite.

and their negations for the other two instantiations of the scheme.

▶ THEOREM 4.3. For every C-regular and C-maximal coalition frame, eventual goal achievement and goal maintenance are duals, as are total and partial terminal effectivity. Formally, $\overline{E^*_C(X)} = E^\times_{\overline{C}}(\overline{X})$ and $\overline{E^t_C(X)} = E^p_{\overline{C}}(\overline{X})$.

PROOF. For the first duality,

$$E^\times_{\overline{C}}(\overline{X}) = \nu Y.\overline{X} \cap (E_\emptyset(\emptyset) \cup E_{\overline{C}}(Y)) = \neg\mu Y.X \cup (\overline{E_\emptyset(\emptyset)} \cap \overline{E_{\overline{C}}(\overline{Y})}) = \overline{E^*_C(X)}$$

by C-maximality and C-regularity. The second claim is then an easy corollary. ■

The relationship between partial and total terminal effectivity is further described by the following result which should be seen as a generalization of Dijkstra's work, establishing links between partial and total program correctness.

▶ THEOREM 4.4. For every coalition frame, total implies partial terminal effectivity, i.e., $E^t_C(X) \subseteq E^p_C(X)$. For superadditive coalition frames, at states where no infinite play is possible, the converse also holds, i.e., $E^t_\emptyset(S) \cap E^p_C(X) \subseteq E^t_C(X)$.

PROOF. Let $\mathcal{F} = (S, \{E_C | C \subseteq N\})$ be any coalition frame.

(1) Given $Y \subseteq S$, let $F_t(Y) = (E_\emptyset(\emptyset) \cap X) \cup (\overline{E_\emptyset(\emptyset)} \cap E_C(Y))$ and $F_p(Y) = (\overline{E_\emptyset(\emptyset)} \cup X) \cap (E_\emptyset(\emptyset) \cup E_C(Y))$. We show by transfinite induction that for all ordinals κ, $F_t^{\uparrow\kappa} \subseteq F_p^{\downarrow\kappa}$. For the inductive step,

$$F_t^{\uparrow\kappa+1} = (E_\emptyset(\emptyset) \cap X) \cup (\overline{E_\emptyset(\emptyset)} \cap E_C(F_t^{\uparrow\kappa})) \subseteq (\overline{E_\emptyset(\emptyset)} \cup X) \cap (E_\emptyset(\emptyset) \cup E_C(F_p^{\downarrow\kappa})) = F_p^{\downarrow\kappa+1}$$

using the induction hypothesis and monotonicity. For limit ordinals, the inductive step follows from the fact that for all $\kappa_0 \leq \kappa$, $F^{\uparrow\kappa_0} \subseteq F^{\uparrow\kappa}$ and $F^{\downarrow\kappa} \subseteq F^{\downarrow\kappa_0}$. Now to prove the first claim, if $s \in E^t_C(X)$, there is some closure ordinal α such that for all $\beta \geq \alpha$, $s \in F_t^{\uparrow\beta}$ and hence $s \in F_p^{\downarrow\beta}$. Thus, for all ordinals γ, $s \in F_p^{\downarrow\gamma}$, in particular for the closure ordinal γ_0 for which $F_p^{\downarrow\gamma_0} = F_p(F_p^{\downarrow\gamma_0}) = E^p_C(X)$.

(2) Given $Y \subseteq S$, let $F_f(Y) = E_\emptyset(Y)$, and assume that $\mathcal{F}$ is superadditive. We show by induction that for all ordinals κ, $F_f^{\uparrow\kappa} \cap F_p^{\downarrow\kappa} \subseteq F_t^{\uparrow\kappa}$. For the inductive step $\kappa + 1$, we must show that

$$E_\emptyset(F_f^{\uparrow\kappa}) \cap (\overline{E_\emptyset(\emptyset)} \cup X) \cap (E_\emptyset(\emptyset) \cup E_C(F_p^{\downarrow\kappa})) \subseteq (E_\emptyset(\emptyset) \cap X) \cup (\overline{E_\emptyset(\emptyset)} \cap E_C(F_t^{\uparrow\kappa}))$$

which follows from superadditivity and the induction hypothesis. The rest of the proof is analogous to the proof of (1). ■

4.2 Syntax & Semantics

As the name suggests, Extended Coalition Logic (ECL) extends Coalition Logic with an extra modality for expressing effectivity in the long run. As we have seen in the previous section, all of the notions discussed can be expressed in terms of eventual goal achievement E^* and goal maintenance $E^\times$.

► Definition 4.5 (Extended Coalition Logic Syntax). Given a finite non-empty set of agents/players N, and a set of atomic propositions Φ_0, formulas φ of *Extended Coalition Logic* can have the following syntactic form:

$$\varphi := \ \perp \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid [C]\varphi \mid [C^*]\varphi \mid [C^\times]\varphi$$

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

As before, the other boolean connectives are defined in the standard way. Like in basic Coalition Logic, ECL formulas are interpreted over coalition models $\mathcal{M} = ((S, \{E_C | C \subseteq N\}), V)$ just as before, the only difference being the additional modalities:

► Definition 4.6 (Extended Coalition Logic Semantics). Given a coalition model $\mathcal{M} = ((S, \{E_C | C \subseteq N\}), V)$, the truth of a formula φ in a model $\mathcal{M}$ at a state s is defined as follows:

$$\begin{array}{ll}
\mathcal{M}, s \not\models \perp & \\
\mathcal{M}, s \models p & \text{iff } p \in \Phi_0 \text{ and } s \in V(p) \\
\mathcal{M}, s \models \neg\varphi & \text{iff } \mathcal{M}, s \not\models \varphi \\
\mathcal{M}, s \models \varphi \vee \psi & \text{iff } \mathcal{M}, s \models \varphi \text{ or } \mathcal{M}, s \models \psi \\
\mathcal{M}, s \models [C]\varphi & \text{iff } sE_C\varphi^{\mathcal{M}} \\
\mathcal{M}, s \models [C^*]\varphi & \text{iff } sE^*_C\varphi^{\mathcal{M}} \\
\mathcal{M}, s \models [C^\times]\varphi & \text{iff } sE^\times_C\varphi^{\mathcal{M}}
\end{array}$$

By theorem 4.3, we know that for all C-regular and C-maximal coalition models, $[C^*]\varphi \leftrightarrow \neg[\overline{C}^\times]\neg\varphi$ is valid. In fact, the proof of theorem 4.3 shows that it is sufficient that $\mathcal{M}$ is C-maximal and C-regular at all non-terminal states, i.e., at all states not in $E_\emptyset(\emptyset)$. Finally, to simplify notation even more, we introduce natural abbreviations for total and partial terminal effectivity, writing $[C^*]([\emptyset]\perp \wedge \varphi)$ as $[C^t]\varphi$ and $[C^\times]([\emptyset]\perp \rightarrow \varphi)$ as $[C^p]\varphi$.

Theorem 3.4 which showed that formulas of Coalition Logic are invariant for bisimulation can be generalized to Extended Coalition Logic.

► Theorem 4.7. Bisimilarity implies ECL-equivalence.

Proof. Two additional cases need to be added to the proof of theorem 3.4. Let $\mathcal{M} = ((S, \{E_C | C \subseteq N\}), V)$ and $\mathcal{M}' = ((S', \{F_C | C \subseteq N\}), V')$ be two coalition models such that $s \leftrightarrows s'$. We show that if $\mathcal{M}, s \models [C^*]\varphi$ then $\mathcal{M}', s' \models [C^*]\varphi$. The case for $[C^\times]\varphi$ is dealt with analogously.

Let $X = \varphi^{\mathcal{M}}$, $X' := \{x' \in S' | \exists x \in X : x \leftrightarrows x'\}$ and

$$Z := \{z \in S | \forall z' : z \leftrightarrows z' \Rightarrow z' F^*_C X')\}.$$

Now it is sufficient to show that $E^*_C(X) \subseteq Z$, and given the definition of $E^*_C(X)$ as a least fixpoint, it suffices to show that Z is a fixpoint, i.e. that

$$X \cup (\overline{E_\emptyset(\emptyset)} \cap E_C(Z)) \subseteq Z.$$

Supposing that $x \in X$ and for some x' we have $x \leftrightarrows x'$, we have $x' \in X' \subseteq F^*_C(X')$. On the other hand, suppose that $x \in (\overline{E_\emptyset(\emptyset)} \cap E_C(Z))$ and $x \leftrightarrows x'$. Then by bisimulation, there is some Z' such that $x' \in (\overline{F_\emptyset(\emptyset)} \cap F_C(Z'))$ and for all $z' \in Z'$ there is some $z \in Z$ such that $z \leftrightarrows z'$. But then $Z' \subseteq F^*_C(X')$, and so by monotonicity, $x' \in \overline{F_\emptyset(\emptyset)} \cap F_C(F^*_C(X')) \subseteq F^*_C(X')$. ■

Axiomatically, we can define extensions of the coalition logics defined in section 3.4. For each of the 2 new modalities, one axiom and one inference rule needs to be added.

▶ DEFINITION 4.8 (EXTENDED COALITION LOGIC AXIOMATICS). Given the set of players N, an *extended coalition logic for N* is a set of ECL-formulas Λ which is a coalition logic and which additionally is closed under the inference rules of figure 4.1 below.

Axioms:	$(\varphi \vee (\neg[\emptyset]\bot \wedge [C][C^*]\varphi)) \to [C^*]\varphi$
	$[C^\times]\varphi \to (\varphi \wedge ([\emptyset]\bot \vee [C][C^\times]\varphi))$
Inference Rules:	$\dfrac{(\varphi \vee (\neg[\emptyset]\bot \wedge [C]\psi)) \to \psi}{[C^*]\varphi \to \psi}$
	$\dfrac{\psi \to (\varphi \wedge ([\emptyset]\bot \vee [C]\psi))}{\psi \to [C^\times]\varphi}$

Figure 4.1: Inference rules for Extended Coalition Logic.

Intuitively, the axiom for * states that $[C^*]\varphi$ is a fixpoint of the operation $\varphi \vee (\neg[\emptyset]\bot \wedge [C]X)$ and the fixpoint rule states that $[C^*]\varphi$ is the least such fixpoint. Similarly for $^\times$ and the greatest fixpoint.

Let $\texttt{Play}^*$, $\texttt{MaxPlay}^*$, and $\texttt{Ind}^*$ be the extended coalition logics with the additional axioms for playability, maximal playability and individualism, respectively. We conjecture that these logics are complete with respect to Play, $\mathsf{MaxPlay}$, and Ind, respectively, but at present we only have the following:

▶ THEOREM 4.9. $\texttt{Play}^*$, $\texttt{MaxPlay}^*$, and $\texttt{Ind}^*$ are sound with respect to Play, $\mathsf{MaxPlay}$, and Ind, respectively.

4.3 Some Meta-Theory

The additional modalities of Extended Coalition Logic do not only allow for a range of applications, they also pose some interesting meta-theoretic questions, some of which we will not be able to answer in the present section. The most notable opportunities for future work concern axiomatization and the complexity of the satisfiability problem, even though chapter 6 will shed some light on the difficulties involved here since eventual goal achievement is in fact very similar to iteration in Game Logic.

We start by considering the relationship between local and global properties. Given that certain conditions such as playability are imposed on the local effectivity function E, will these conditions be preserved on the global level by E^t? For playability, we shall answer this question affirmatively, also pointing out the link to the game-theoretic concept of strategic normal form.

As might be expected, the additional modalities of Extended Coalition Logic will increase the complexity of the model checking problem, and below we will give a precise upper bound which should be compared to the upper bound obtained for basic Coalition Logic. Furthermore, we take up the issue of coalitional expressiveness: We saw in section 3.7.1 that in extensive games without simultaneous moves, coalitions did not add any expressive power beyond individuals. We will see in section 4.3.3 that when we also consider effectivity in the long run, coalitions do add expressive power even for these games.

4.3.1 Local vs. Global Properties of Ability

The different kinds of coalition frames associated, e.g., with extensive games have been defined in terms of local requirements, i.e., properties such as weak playability which the local effectivity functions have to satisfy. Some of these properties will be maintained globally or terminally. An important case is weak playability: One can show that for games without infinite plays, the total (= partial) terminal effectivity function is playable.

► THEOREM 4.10. If $\mathcal{F} = (S, \{E_C | C \subseteq N\})$ is a weakly playable coalition frame such that $sE^t_\emptyset S$, then $E^t(s)$ is strongly playable.

PROOF. The strong playability conditions can be checked one by one: For the first condition, one can check that $E^t_C(\emptyset) = \emptyset$. For the second condition, $E^t_\emptyset(S) \subseteq E^t_C(S)$ since $E(s)$ is coalition monotonic in case $s \notin E_\emptyset(\emptyset)$. For N-maximality, if $s \notin E^t_\emptyset(\overline{X})$, since $s \in E^t_\emptyset(S)$, we have $s \notin E^p_\emptyset(\overline{X})$ by theorem 4.4 and hence $sE^t_N X$ by theorem 4.3.

For superadditivity, we show generally that for all $C_1 \cap C_2 = \emptyset$ we have $E^t_{C_1}(X_1) \cap E^t_{C_2}(X_2) \subseteq E^t_{C_1 \cup C_2}(X_1 \cap X_2)$; we proceed again by transfinite induction.

as in the proof of theorem 4.4. Let

$$\begin{aligned}F_1(Y) = &\quad (E_\emptyset(\emptyset) \cap X_1) \cup (\overline{E_\emptyset(\emptyset)} \cap E_{C_1}(Y))\\ F_2(Y) = &\quad (E_\emptyset(\emptyset) \cap X_2) \cup (\overline{E_\emptyset(\emptyset)} \cap E_{C_2}(Y))\\ F_3(Y) = &\quad (E_\emptyset(\emptyset) \cap X_1 \cap X_2) \cup (\overline{E_\emptyset(\emptyset)} \cap E_{C_1 \cup C_2}(Y))\end{aligned}$$

We show that $F_1^{\uparrow\kappa} \cap F_2^{\uparrow\kappa} \subseteq F_3^{\uparrow\kappa}$. The heart of the proof is the inductive step for $\kappa + 1$, where one can check that

$$((E_\emptyset(\emptyset) \cap X_1) \cup (\overline{E_\emptyset(\emptyset)} \cap E_{C_1}(F_1^{\uparrow\kappa}))) \cap ((E_\emptyset(\emptyset) \cap X_2) \cup (\overline{E_\emptyset(\emptyset)} \cap E_{C_2}(F_2^{\uparrow\kappa})))$$

is a subset of

$$(E_\emptyset(\emptyset) \cap X_1 \cap X_2) \cup (\overline{E_\emptyset(\emptyset)} \cap E_{C_1 \cup C_2}(F_3^{\uparrow\kappa})).$$

The only two cases possible are $s \in E_\emptyset(\emptyset) \cap X_1 \cap X_2$ and $s \in \overline{E_\emptyset(\emptyset)} \cap E_{C_1}(F_1^{\uparrow\kappa}) \cap E_{C_2}(F_2^{\uparrow\kappa})$. The latter case makes use of the superadditivity of E. ■

Consequently, for every extensive game G without infinite plays, there is a nonempty strategic game G' such that the total terminal effectivity function of G is the α-effectivity function of G'. In fact, one such strategic game G' is simply the strategic normal form of G. As for the logical analogue of theorem 4.10, the preservation of local properties on the global level, observe that the four strong playability conditions for E' can be translated into the logical language:

(1) $\neg[C']\bot$
(2) $[C']\top$
(3) $\neg[\emptyset']\neg\varphi \rightarrow [N']\varphi$
(4) $([C_1']\varphi_1 \wedge [C_2']\varphi_2) \rightarrow [(C_1 \cup C_2)'](\varphi_1 \wedge \varphi_2)$ where $C_1 \cap C_2 = \emptyset$

Theorem 4.10 thus shows that all four axiom schemas are valid for extensive games without infinite plays.

As another example, consider majority voting which we will discuss in more detail in the next chapter. If the game linked to a particular state is a voting game where all of the players can choose between a number of alternatives, we might want to demand that every majority of players can completely determine the outcome. This property can be captured in terms of effectivity: Call an effectivity function $E : \mathcal{P}(N) \rightarrow \mathcal{P}(\mathcal{P}(S))$ *majorative* iff for every coalition C with $|C| > \frac{1}{2}|N|$ we have $E(N) \subseteq E(C)$.

The notion of majorativity is a local notion which guarantees that at every stage of a voting process, voting is democratic. As one would hope, this property holds for the voting procedure as a whole as well: If E is majorative at every state of an coalition frame, then so is E', so a democratic procedure will maintain democracy overall. That the converse is not true can be gathered from the extensive game in figure 4.2, where at the initial state, E' is majorative while E is not.

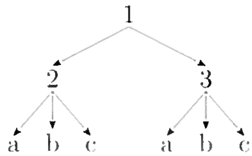


Figure 4.2: A game which is majorative overall without being locally majorative

4.3.2 Model checking

As may be expected, the presence of modalities for eventual goal achievement and goal maintenance leads to a more complex model checking problem. Since these modalities are defined by fixpoint constructions, a model checking algorithm and its complexity will resemble algorithms used for model checking of the modal μ-calculus. For the μ-calculus, the best known upper bound is NP $\cap$ co-NP, while for bounded alternation depth, the problem can be solved in deterministic polynomial time. Since Extended Coalition Logic does not allow for nestings of fixpoint operators which could yield formulas of the form $\mu X.\nu Y[C]X \wedge [D]Y$, one can expect model checking to be solvable in polynomial time where the polynomial is of very low degree. This should be compared to the analysis of the model checking problem for Game Logic in section 6.5.

▶ THEOREM 4.11. Given a formula φ of Extended Coalition Logic and a coalition model $\mathcal{M}$, there is an algorithm for calculating $\varphi^{\mathcal{M}}$ which runs in time $O(|\mathcal{M}|^2 \times |\varphi|)$.

PROOF. The proof extends the argument used for basic Coalition Logic by two additional cases, eventual goal achievement and goal maintenance: In case $\varphi_{k+1} = [C^*]\varphi_k$, after determining all the $\varphi_i^{\mathcal{M}}$ for $i \leq k$ in time $O(k \times |\mathcal{M}|^2)$, we now need to calculate

$$E_C^*(\varphi_k^{\mathcal{M}}) = \mu X.\varphi_k^{\mathcal{M}} \cup (\overline{E_\emptyset(\emptyset)} \cap E_C(X)) = \bigcup_{0 \leq i \leq |S|} F^{\uparrow i}$$

where $F(X) = \varphi_k^{\mathcal{M}} \cup (\overline{E_\emptyset(\emptyset)} \cap E_C(X))$. We can assume that at the very beginning we have marked all states s according to whether or not $sE_\emptyset\emptyset$ holds. Initially, we then label all states satisfying φ_k with $[C^*]\varphi_k$. Next, for every state $t \notin E_\emptyset(\emptyset)$ we check whether there is a set X with $tE_C^c X$ such that all states in X are labeled with $[C^*]\varphi_k$, and if so, we label t with $[C^*]\varphi_k$ as well. We repeat this step at most $|S|$ times, and since each step can be done in $O(|\mathcal{M}|)$ time, $E_C^*\varphi_k^{\mathcal{M}}$ can be calculated in $O(|\mathcal{M}|^2)$, yielding a $O((k+1) \times |\mathcal{M}|^2)$ bound for calculating φ_{k+1}.

In case $\varphi_{k+1} = [C^\times]\varphi_k$, after determining all the $\varphi_i^{\mathcal{M}}$ for $i \leq k$ in time $O(k \times$

$|\mathcal{M}|^2$), we need to calculate

$$E_C^{\times}(\varphi_k^{\mathcal{M}}) = \nu X.\varphi_k^{\mathcal{M}} \cap (E_\emptyset(\emptyset) \cup E_C(X)) = \bigcap_{0 \le i \le |S|} F^{1i}$$

where $F(X) = \varphi_k^{\mathcal{M}} \cap (E_\emptyset(\emptyset) \cup E_C(X))$.

Initially, we label all states satisfying φ_k with $[C^\times]\varphi_k$. Next, for every state $t \notin E_\emptyset(\emptyset)$ labeled with $[C^\times]\varphi_k$, we check whether there is a set X with $tE_C^c X$ such that all states in X are labeled with $[C^\times]\varphi_k$, and if not, we remove the $[C^\times]\varphi_k$ label from t. Again, this yields an $O((k+1) \times |\mathcal{M}|^2)$ bound for calculating φ_{k+1}. ∎

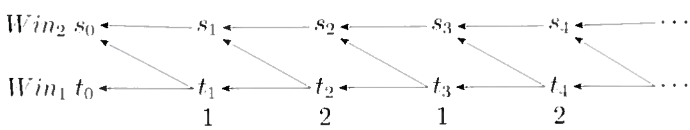
Roughly speaking, while model checking for basic Coalition Logic is linear time, model checking for Extended Coalition Logic is quadratic. Note again that as with basic Coalition Logic, this is a rather rough estimate: Inspecting the proof carefully, the real calculation time is actually $O(|S| \times |\mathcal{M}| \times |\varphi|)$ where S is the universe of $\mathcal{M}$. Since $|S|$ is usually much smaller than $|\mathcal{M}|$, we consider the result as an argument for the feasibility of doing model checking in practice, also for Extended Coalition Logic.

4.3.3 Expressiveness

In theorem 3.35 of the previous chapter, we have compared the expressiveness of full Coalition Logic to the expressiveness of its individual fragment. The result stated that while over Mon, Play and MaxPlay full Coalition Logic is more expressive than its individual fragment, this is not the case for Ind. In other words, for extensive games without simultaneous moves, group modalities do not add anything to individual modalities in terms of expressive power. For Extended Coalition Logic, it turns out that even for extensive games without simultaneous moves, coalitional modalities do add expressive power to individual modalities.

▶ THEOREM 4.12. Over Mon, Play, MaxPlay and Ind, Extended Coalition Logic is more expressive than its individual fragment, provided that $|N| > 1$.

PROOF. For Mon, Play and MaxPlay, the proof of theorem 3.35 for basic Coalition Logic easily extends to Extended Coalition Logic. For Ind, consider the weakly individualistic coalition model $\mathcal{M} = ((S, \{E_C | C \subseteq N\}), V)$ with $S = \{s_i, t_i | i \ge 0\}$, $\Phi_0 = \emptyset$ and $N = \{1, 2\}$, which α-corresponds to the extensive game below:



Every state s_i has only a single successor state so that it is irrelevant whose turn it is at s_i. At t_i for $i > 0$ on the other hand, player 1 moves if i is odd, player 2 moves if i is even. As the game indicates, player 1 wins at state t_0 whereas player 2 wins at state s_0, so the empty games associated with these states are $\{1\}$ and $\{2\}$, respectively. Given these empty games, note that $\mathcal{M}$ is C-maximal for all coalitions C at every state.

We will use this model to show that the formula $[N^*][1]\bot$ cannot be expressed in the individual fragment of Extended Coalition Logic. The formula states that the players together have a strategy to achieve a win for player 1. Note that $[N^*][1]\bot$ is true at all t_i and false at all s_i. This means that the denotation of $[N^*][1]\bot$ in $\mathcal{M}$ is neither finite nor co-finite since it is true at an infinite number of states and also false at an infinite number of states. To establish our result, we shall show by induction that the denotation of all formulas of the individual fragment of ECL are either finite or co-finite.

The base case and the boolean cases are easy to check, so we only consider the modalities for player 1. For $[1]\varphi$, if $\varphi^{\mathcal{M}}$ is finite, the denotation of $[1]\varphi$ must also be finite, simply because $E_C(X)$ is finite provided that X is, with the proviso that there is only a finite number of terminal states. If $(\neg\varphi)^{\mathcal{M}}$ is finite, then $(\neg[1]\varphi)^{\mathcal{M}} = ([2]\neg\varphi)^{\mathcal{M}}$ using maximality and the latter set must again be finite.

Consider now $[1^*]\varphi$ and assume that $\varphi^{\mathcal{M}}$ is finite. We can also assume that $\varphi^{\mathcal{M}} \neq \emptyset$, for otherwise $[1^*]\varphi$ will be false everywhere and hence of finite denotation. We consider two different cases: (1) Suppose first that there is some state s_c such that $\mathcal{M}, s_c \models \varphi$. As a result, $[1^*]\varphi$ will hold at all states to the right of s_{c+1}, i.e., for all $d > c + 1$ we have $\mathcal{M}, s_d \models [1^*]\varphi$ and $\mathcal{M}, t_d \models [1^*]\varphi$, the strategy for player 1 being to choose a state in the top row as soon as possible. Consequently, $[1^*]\varphi$ can fail to hold only to the left of s_{c+1} (top or bottom row), hence $(\neg[1^*]\varphi)^{\mathcal{M}}$ is finite. In the second case (2), there is no state s_c such that $\mathcal{M}, s_c \models \varphi$. Let t_c be the rightmost state at which $\mathcal{M}, t_c \models \varphi$ (since $\varphi^{\mathcal{M}}$ is finite such a largest c exists). Then for all $d > c + 1$ we have $\mathcal{M}, s_d \not\models [1^*]\varphi$ and $\mathcal{M}, t_d \not\models [1^*]\varphi$ since player 2 can simply choose a state in the top row as soon as possible. Consequently, $[1^*]\varphi$ can hold only to the left of t_{c+1} (top or bottom row), hence $([1^*]\varphi)^{\mathcal{M}}$ is finite.

The final situation to consider is when $\varphi^{\mathcal{M}}$ is co-finite. Then by theorem 4.3, $(\neg[1^*]\varphi)^{\mathcal{M}} = ([2^\times]\neg\varphi)^{\mathcal{M}}$, and since $E_C^\times(X)$ must be finite in case X is, $([1^*]\varphi)^{\mathcal{M}}$ is co-finite. ∎

The proof of this expressiveness result actually establishes a difference in expressive power in a very strong form: It would have been sufficient to show that there is an ECL-formula which is not equivalent to any formula of the individual fragment of ECL *over all coalition models*. The proof of theorem 4.12 on the other hand establishes something stronger, namely that there is an ECL-formula and a model such that no formula of the individual fragment is equivalent to that formula in that model. So in a very strong sense, long-term coalitional effectivity cannot be reduced to individual effectivity.

Note also that the model used in the proof had to be infinite, i.e., for every finite model $\mathcal{M}$, every coalitional ECL-formula φ is equivalent (on $\mathcal{M}$) to a formula of the individual fragment of ECL. The reason is that since $\mathcal{M}$ is finite, φ can be rewritten into a CL formula φ' which is true at exactly the same states, simply by expanding the fixpoint definition at most n times, where n is the number of states in the model. Formula φ' can in turn be rewritten into an equivalent formula φ'' of the individual fragment of CL. Note, however, that this argument does *not* show that over finite individualistic models, the individual fragment of ECL is equally expressive as full ECL, for φ'' will only be equivalent to φ in $\mathcal{M}$, not in all models. In fact, we conjecture that also over finite individualistic models, full ECL will be more expressive than its individual fragment.

4.4 Alternating Temporal Logic

As shown in [56] on which the following discussion is based, Extended Coalition Logic is closely related to Alternating Temporal Logic (ATL) [3], a generalization of temporal logic for reasoning about open systems. Traditional linear-time and branching-time temporal logics such as LTL and CTL describe *closed systems*, i.e., systems whose behavior is completely determined by the present state of the system. In *open systems* on the other hand, the system interacts with the environment and hence the behavior of the system depends on the present state and the behavior of the environment. The picture can be generalized to multi-agent systems where the different agents may represent different components of the system and the environment. An open system is modeled as an alternating transition system:

▶ DEFINITION 4.13 (ALTERNATING TRANSITION SYSTEM). For agents N and a set of atomic propositions Φ_0, an *alternating transition system* (ATS) is a triple $\mathcal{M} = (S, V, \delta)$ where S is a nonempty set of states, $V : \Phi_0 \to \mathcal{P}(S)$ is the valuation function, and $\delta : S \times N \to \mathcal{P}(\mathcal{P}(S))$ models the effectivity of a player at a state. The function δ must satisfy the following intersection property: Assume that $N = \{1, \ldots, n\}$. Then for every state $s \in S$ and all sets $X_1, \ldots, X_n$ such that $X_i \in \delta(s, i)$ for all $i \in N$, we have $|\bigcap_{i \in N} X_i| = 1$.

Thus, δ essentially associates an individual effectivity function with each state. The only requirement on δ is that every set of choices which the agents make determines precisely one resulting state, the new state of the system.

There is a tight connection between alternating transition systems and strongly playable coalition models. Recall that strongly playable coalition models associate a strategic game form with every state. Given an alternating transition system, one can easily define a corresponding strongly playable coalition model: For every state $s \in S$, associate a strategic game form with s such that the strategies Σ_i of player i are all the sets for which she is effective, i.e., $G(s) = (N, \{\Sigma_i | i \in N\}, o, S)$ where $\Sigma_i = \{X_i \subseteq S | X_i \in \delta(s, i)\}$ and $o(X_1, \ldots, X_n) = \bigcap_{i \in N} X_i$. The two models correspond in the sense that at every state, the coalitional effectivity is the same in both models: Extending the individual effectivity δ of ATSs to coalitions by defining $X \in \delta(s, C)$ iff $X \supseteq \bigcap_{i \in C} X_i$ and for all $i \in C$, $X_i \in \delta(s, i)$, one can show that $X \in \delta(s, C)$ holds in an ATS iff $sE_C X$ holds in its corresponding coalition model.

Conversely, it is not the case that every strongly playable coalition model corresponds to an alternating transition system. Consider the following 2-player strategic game form:

	m	e
m	*win*	*loss*
e	*loss*	*win*

The game form models, e.g., the coordinated attack problem where two generals have to decide independently when to attack a common enemy. If both attack in the morning (m) they will win, and similarly if both attack in the evening (e). If they attack at different times however (i.e., they fail to coordinate), they will lose. While this strategic game can be associated to a state in a strongly playable coalition model (where *win* and *loss* are states), there is no individual effectivity map δ which correctly captures the individual effectivity in this game and which satisfies the intersection property. For note that in this game player 1 is α-effective for $\{win, loss\}$ and so is player 2, and there are no smaller sets for which they are α-effective. Since the intersection of $\{win, loss\}$ with itself is not a singleton, there can be no map δ which correctly captures this game. Consequently, there are strongly playable coalition models which do not correspond to an ATS. (The class of game forms which can be correctly captured by an individual effectivity map δ satisfying the intersection property is the class of rectangular game forms.)

Various types of ATSs are discussed in [3]: In a *turn-based synchronous* ATS, there is a single agent at every state which determines the next state of the system. Since the agent may be different at every state, turn-based synchronous ATSs thus correspond to strongly individualistic coalition models where each state knows a local dictator. In a *lock-step synchronous* ATS, each state of the system is divided into local states for each agent, and at every state, each agent can determine its next local state, possibly dependent on the current local states of the other agents, but independent of the actions of the other agents. In a *turn-based asynchronous* ATS finally, there is a designated agent called the scheduler who chooses at every state an agent who gets to determine the next state. Usually the scheduling policy will be subject to various fairness constraints, prohibiting, e.g., that the scheduler assigns the same agent to every state.

The central feature of the language of ATL are its three modalities $\langle C \rangle X\varphi$ (next), $\langle C \rangle G\varphi$ (always) and $\langle C \rangle \varphi U \psi$ (until). The formula $\langle C \rangle X\varphi$ is true at a state in case coalition C is locally effective for φ, i.e., it corresponds to the Coalition Logic formula $\langle C\rangle\varphi$. The formula $\langle C\rangle G\varphi$ is true at a state in case coalition C has a joint long-term strategy which maintains φ in the future. Consequently, $\langle C\rangle G\varphi$ is the temporal analogue of the Extended Coalition Logic formula $[C^\times]\varphi$. Finally, $\langle C\rangle\varphi U\psi$ expresses that coalition C has a joint long-term strategy which will guarantee ψ at some point in the future and maintain φ until that point.

Coalition Logic and Extended Coalition Logic over strongly playable coalition models form a fragment of ATL. We have seen that the language of (E)CL forms a sublanguage of ATL. Furthermore, while not every strongly playable coalition model corresponds to an ATS, for every strongly playable coalition model there is an ATS satisfying the same formulas of (E)CL. The reason can easily be demonstrated using the earlier coordinated attack game: While that game cannot be modeled by an ATS, the following game can easily be modeled by an ATS:

	m	e
m	win_1	$loss_1$
e	$loss_2$	win_2

If we see to it that the states win_1 and win_2 on the one hand and $loss_1$ and $loss_2$ on the other hand are identical in terms of observable properties, the game cannot be distinguished from the original coordinated attack game by any formula of the language of CL or ATL. Hence, the axiomatization we gave in section 3.4 can be mapped into an axiomatization for a fragment of ATL. In [56], this axiomatization has been extended to full ATL as well. Note, however, that the coordinated attack example shows that a satisfying ATS may be much larger than a satisfying coalition model, so complexity results regarding the satisfiability problem do not immediately transfer from one logic to the other.

Coalition Logic has a number of advantages over ATL: First, since coalition models allow for terminal states, we can model endpoints of a process explicitly. In an ATS, such terminal states would have to be represented via terminal loops, where all transitions lead to the same state again. The problem with this solution is that it does not allow us to distinguish partial from terminal effectivity anymore, since all runs will be infinite. Second, we saw that coalition models can associate arbitrary strategic games to states while ATSs can only capture rectangular games. In fact, coalition models can even capture interactions which cannot be modeled by strategic games at all, for example situations where coalition monotonicity is violated. Third, while the expressive power of basic and Extended Coalition Logic is weaker than that of ATL, Coalition Logic will be computationally less complex. This means that satisfiability problems which may be intractable for ATL may still be feasible for Coalition Logic. This will certainly be the case for basic Coalition Logic, and as chapter 5 will show, many applications will not require more.

4.5 Distributed Artificial Intelligence

The research area of distributed artificial intelligence has developed various logics for reasoning about multi-agent systems, formalizing not only the ability of an agent but also its beliefs, desires, intentions, and so on. Hence, it is useful to point out some similarities and differences between Coalition Logic and the approach taken in the artificial intelligence literature.

The semantics used here to formalize multi-agent ability is based on minimal models with a neighborhood relation for each agent. For the single agent case, such models have been used in [28] to study the logic of ability. This logic of ability is a very weak modal logic since properties such as $\Diamond(A \vee B) \rightarrow (\Diamond A \vee \Diamond B)$ fail. The example given to illustrate the failure of this principle refers to a deck of cards turned face down. Since the colors (red or black) are concealed, the agent is not able to draw a red card nor a black card, while he is able to draw a card which is either black or red. From our perspective, we interpret the situation as a game against nature, i.e., as a 2-player game where Nature chooses which card to give to the agent. The advantage of this approach is that it makes the roles of the players explicit, and hence one can point out that if the situation were in fact a 1-player game, the modal distribution principle would hold after all.

The approach taken here to formalize the ability of groups of agents differs somewhat from the existing literature on multi-agent ability. Among the earliest works, Tennenholtz and Moses in [119] conceive of an agent as a set of finite state machines whose transitions model the agent's actions. As in an extensive game with simultaneous moves, the joint actions of the agents determine the new configuration of the system. Of central concern is the *cooperative goal achievement* (CGA) problem: Is there a run of the system in which all agents achieve their goal? It is argued that this problem is PSPACE-complete, which seems to correspond nicely to the complexity of the satisfiability problem of basic Coalition Logic, but the decision problems are quite different. Note also that our logical approach is more general in that it is not specifically tailored to the CGA problem alone, but also allows to ask, for example, whether an infinite run can be forced by some group of agents.

While the work of Werner in [125] is more directly related in its logical approach, his framework includes much more than just ability, covering also time, intentions, actions and knowledge. Given the more complex aims, his semantics is much more complicated than what is proposed here, and some of the fundamental issues which arise purely on the level of abilities are not investigated, e.g., the relationship between local and global ability, basic cooperative axioms such as superadditivity, and so on.

Related to Werner's work, Wooldridge and Fisher in [126] also take a logical approach to multi-agent interaction which includes communication between agents. Their notion of goal achievement essentially corresponds to α-effectivity. Their logic is very expressive (including "at least" first-order logic) and hence also much more complex then the rather simple system presented here. Some of their axioms, however, have direct analogues in Coalition Logic. As an example, one of their axioms states that bigger groups cannot achieve less (coalition-monotonicity) which they write as $\forall x \forall y((Can(x, \varphi) \wedge x \subseteq y) \rightarrow Can(y, \varphi))$ where x and y refer to groups of agents.

Note that our general approach is different from the works cited and from the approach taken in multi-agent systems more generally: Our aim is to provide a formal logical theory of ability in a multi-agent setting, without adding any other notions such as beliefs, intentions, etc. which would complicate the picture. What is more important, we want a general model of ability, and this is what effectivity functions allow us to do. Effectivity in game-like situations (which is taken as basic in the other approaches) is only a special case which can be characterized by certain axioms, precisely the properties of group ability which characterize strategic games. This approach still allows us to model situations which would be beyond the scope, e.g., of [126] because they violate the coalition-monotonicity mentioned. Even our relatively simple model, however, is sufficient to ask many of the questions raised in the literature such as the CGA problem.

4.6 Summary

Extended Coalition Logic is a sufficiently expressive yet computationally simple extension of basic Coalition Logic. In this sense, ECL is similar to temporal logics like CTL which are simple and expressive fragments of the highly complex modal μ-calculus. Theorem 4.11 has shown that the complexity of model checking is still relatively low, in particular when compared to Game Logic whose complexity we shall discuss in section 6.5. On the other hand, ECL can express the existence of a winning strategy in an extensive game, one of the central properties we will want to express in applications. Further evidence of the expressive power of ECL is given by theorem 4.12 which shows that even over extensive games of perfect information, the existence of a coalitional winning strategy cannot be reduced to individual strategies. Consequently, there can be no logic for reasoning about individual ability only which is as expressive as ECL.

Of the various long-term ability notions we have considered, we saw that at least for determined extensive games, all can be reduced to eventual goal achievement (theorem 4.3). For reasoning about undetermined games, we also need the other primitive modality $[C^{\times}]$ defined in terms of goal maintenance $E^{\times}$.

Finally, the generality of coalition models has allowed us to obtain a generalization of Dijkstra's relations between partial and total program correctness. These relations state that for programs, total correctness implies partial correctness and that for programs without infinite runs, the converse implication holds as well. Formulated in our framework, these relations amount to the following two claims:

$$E^t_\emptyset(X) \subseteq E^p_\emptyset(X) \text{ and } E^t_\emptyset(S) \cap E^p_\emptyset(X) \subseteq E^t_\emptyset(X)$$

holds for the class of models 1−Play. Theorem 4.4 has generalized this result from programs to games with/without simultaneous moves, formally, from 1−Play to Play and from the empty coalition $\emptyset$ to general coalitions C.

4.7 Bibliographic Notes

Parts of the material in this chapter has been published in [103, 102].

For books on Dijkstra's theory of partial and total program correctness, see [38, 39, 73, 7]. For an overview of temporal logic, see [43].

All the results concerning the relation between Coalition Logic and Alternating Temporal Logic are from [56]. ATL has been extended and modified in various ways (also in [3]), yielding ATL* (which mirrors the step from CTL to CTL*), the alternating-time μ-calculus, and ATL over ATSs with incomplete information, where agents have only partial knowledge of the current state of the system. Alternating refinement relations for ATSs and their computational complexity have been studied in [4] which also contains a notion of alternating bisimulation similar to what was proposed in section 2.5.

Chapter 5

Applications of Coalition Logic

Two things you never want to see made are sausages and laws.
Otto von Bismarck

Having focused on a theoretical analysis of Coalition Logic in the previous chapters, we now turn towards its applications in the analysis of social software. Conceptually, the applications can be divided into two categories, model-checking problems and satisfiability problems.

In the first case of model-checking problems, we have some social process such as a voting procedure whose properties we want to investigate. Is the procedure strategy-proof? Can every majority of voters determine the final outcome? Are certain groups of voters more powerful than others? Formally, we have some formulas of Coalition Logic (the properties to be verified) whose truth value is calculated at some state of a coalition model (the voting procedure). In the second case of satisfiability problems, we want to find a social process which satisfies a given specification. To give an example, we may want to find a voting agenda which is democratic and has certain additional structural properties. Formally, we check whether a given set of CL-axioms can be satisfied by a coalition model of a particular kind. If the answer is positive, the satisfying model will then give us a voting procedure which meets the requirements.

We shall discuss these applications using four examples, most of which are essentially voting problems. As an example of social software, voting processes have the advantage of being important and non-trivial, yet sufficiently well-defined to allow for formalization relatively easily. Basic Coalition Logic will be sufficient as a specification language in most cases. This basic logic can then be enriched as the application at hand requires, e.g., by extending the language with long-term modalities (section 5.3) or by adding an additional inference rule for Nash-consistent implementation (section 5.1). In any case, the applications will show that Coalition Logic can serve as the basis for a formal framework for the analysis and synthesis of social software.

5.1 Fashion Rights

When commenting on the interpretation of effectivity functions in sections 2.2 and 2.3, we said that $X \in E(C)$ was supposed to be interpreted as "the group of agents C can bring about (a state in) X". In other words, bringing about X constitutes an *alethic* possibility for group C. Similarly, effectivity functions have also been used as a model of *deontic* possibility, interpreting $X \in E(C)$ as "the group of agents C has the right to bring about (a state in) X". Under this deontic reading then, effectivity functions are viewed as *rights-systems* or *constitutions*.

In chapter 2, we have discussed a number of properties of effectivity functions and we have characterized, e.g., the α-effectivity functions of strategic games. While these characterization results are of course in no way dependent on the interpretation of the effectivity functions, the reasonableness of the properties selected was based on the alethic interpretation. When interpreted deontically, different properties may be considered. To start with, regularity may fail: If two people Alice and Bob enter a room which contains an empty chair, both of them may have the right to sit on that chair or to remain standing. Denoting the three different states by s_0 (both remain standing), s_A (Alice sits down) and s_B (Bob sits down), we have $\{s_A\} \in E(\{Alice\})$ and $\{s_B, s_0\} \in E(\{Bob\})$ which contradicts regularity. What this failure of regularity in fact shows is that not both individuals are able to simultaneously *exercise* their right to sit down. An example of a deontically acceptable principle is the following condition of minimal libertarianism [50]: For every $i \in N$ there is some $X \subsetneq S$ such that $X \in E(\{i\})$. This condition can be read as saying that all individuals have the right to determine certain decisions by themselves.

Given that certain conditions such as regularity which are implied by playability may fail to hold deontically, rights-systems cannot necessarily be represented by game forms (theorem 2.27). In spite of this, some of the literature has identified deontic with alethic possibility. The advantage of such an identification is that it allows one to consider the decentralization of a constitution as a simple game-theoretic problem: Given a constitution which specifies the rights which individuals and groups of individuals are supposed to have, is there a strategic game form which guarantees these rights? A further desirable criterion which such a game form should satisfy is Nash-consistency: No matter what preferences the players have over the possible outcomes of the game, the game should have a Nash equilibrium. Such a Nash-consistent decentralization of a constitution will guarantee some kind of stability in the strategies the players take, since once a/the Nash equilibrium is reached there is no incentive to change strategies.

As a simple illustration of the decentralization of a constitution, consider the following example: Abelard and Eloise each have a white and a blue shirt, and they have to decide which shirt to wear. Each person has the right to determine the color of his/her own shirt. Using basic Coalition Logic, we can model this situation by letting $N = \{a, e\}$ and using two atomic proposition p_a and p_e, where

p_a should be read as "Abelard wears white" and p_e as "Eloise wears white". The rights of Abelard and Eloise can then be captured by the following formula ρ^+ of Coalition Logic:

$$[a]p_a \wedge [a]\neg p_a \wedge [e]p_e \wedge [e]\neg p_e$$

Furthermore, we assume that these are the only rights which we want to give to them, formalized by formula ρ^-:

$$\bigwedge_{i\in\{a,e\}} \neg[i]((p_a \wedge p_e) \vee (\neg p_a \wedge \neg p_e)) \wedge \bigwedge_{i\in\{a,e\}} \neg[i]((p_a \wedge \neg p_e) \vee (\neg p_a \wedge p_e))$$

To ask whether this constitution can be decentralized by a strategic game means checking whether $\rho = \rho^+ \wedge \rho^-$ is satisfiable by a weakly playable coalition model. The following coalition model $\mathcal{M} = (S, \{E_C | C \subseteq N\}, V)$ with $S = \{s_0, (w, w), (w, b), (b, w), (b, b)\}$ satisfies ρ at state s_0: $V(p_a) = \{(w, w), (w, b)\}$, $V(p_e) = \{(w, w), (b, w)\}$, and $E(s_0)$ is the α-effectivity function associated with the following strategic game:

	w	b
w	(w, w)	(w, b)
b	(b, w)	(b, b)

For states other than s_0, E can be defined arbitrarily. At state s_0, each player can choose which shirt to wear, and the resulting state reflects Abelard's choice in the first component and Eloise's choice in the second.

As it turns out, however, this game form is not Nash-consistent. Consider the situation of the so-called Gibbard paradox where Abelard is primarily conformist, preferring to wear the same color as Eloise, and besides that he prefers white to blue. Eloise on the other hand has the same color preference, but she primarily is a non-conformist, wanting to avoid the same color. The following game captures these preferences:

	w	b
w	$(4, 2)$	$(2, 3)$
b	$(1, 4)$	$(3, 1)$

As can easily be checked, the game has no Nash equilibrium. Even worse, it can be shown that the constitution given has no Nash-consistent representation [106]. Peleg et al. provide an additional condition on effectivity functions which guarantees a Nash-consistent implementation. Their condition gives rise to the following inference rule which can be added to the coalition logic `Play`:

$$\frac{\bigvee_{i\in N} \varphi_i}{\bigvee_{i\in N} [N \setminus \{i\}]\varphi_i}$$

Based on the characterization by Peleg et al. we conjecture that the addition of this inference rule yields a complete axiomatization of weakly playable Nash-consistent coalition models. For the example at hand, one can show that in a system $\Lambda \supseteq$ `Play` which includes this inference rule, ρ is inconsistent: Since

$$((p_a \wedge p_e) \vee (\neg p_a \wedge \neg p_e)) \vee ((p_a \wedge \neg p_e) \vee (\neg p_a \wedge p_e))$$

is a propositional tautology, we can use the new inference rule to derive

$$[a]((p_a \wedge p_e) \vee (\neg p_a \wedge \neg p_e)) \vee [e]((p_a \wedge \neg p_e) \vee (\neg p_a \wedge p_e))$$

which contradicts ρ^-.

To sum up, we have seen a very simple example of how a system of rights, represented by an effectivity function, can be decentralized in a strategic game. Formalizing the rights as coalition logic formulas, this decentralization problem can be turned into a satisfiability problem, and this approach can be extended to cover Nash-consistent decentralization. The logical reformulation of constitutional decentralization has the advantage that it can easily handle partial specifications of rights. In the example of the two shirts, we have assumed that the players' rights were completely specified. This forced us to add a conjunct ρ^- which specified what the players were unable to bring about. In order to avoid such an additional conjunct, we would have to make use of some kind of non-monotonic reasoning mechanism which would allow us to conclude that the rights specified are *all* the rights the players are supposed to have. On the other hand, we may also treat ρ^+ as a partial specification, where we want to ensure that the players have the rights specified in ρ^+ but do not care about whether players also obtain additional rights. Checking satisfiability of ρ^+ thus corresponds to decentralizing a partially defined effectivity function, a problem which does not seem to have received much attention in the social choice theory literature.

5.2 Telephone Democracy

The example discussed in the previous section was simple because it involved very little dynamics: players rights were implemented by a single strategic game, so that the resulting social process consisted only of a single move by every player. The following example requires a more sophisticated implementation.

A father of three daughters has decided that time has come to broaden his mind a bit by reading a controversial book about the relationship between the sexes. He wants to ask his three daughters which of two books they recommend, the options being Simone de Beauvoir's "Le deuxième sexe" or Susan Faludi's "Backlash". Wanting to be impartial, he decides that the majority opinion among the daughters will determine the book he will buy. For $N = \{1, 2, 3\}$, if we let p refer to "Simone de Beauvoir's book is chosen" and q to "Susan Faludi's book is chosen", we can write down this system of rights as

$$\rho = \bigwedge_{\{C \subseteq N \,:\, |C|>1\}} ([C]p \wedge [C]q) \wedge [\emptyset]\neg(p \wedge q).$$

The last conjunct states that the father is not allowed to buy both books. It is easy to see that the following coalition model $\mathcal{M}_1 = ((S, \{E_C | C \subseteq N\}), V)$ with $S = \{s_0, b, f\}$ satisfies ρ at state s_0, where $V(p) = b$, $V(q) = f$ and $E(s_0)$ is the α-effectivity function associated with the following majority voting game

	B	F
B	b	b
F	b	f

	B	F
B	b	f
F	f	f

where daughter 3 decides between the left and the right table. Note that compared to the satisfiability problem considered in the preceding section, checking the satisfiability of ρ is more complex, for while we are still considering satisfiability over weakly playable coalition models, we are not dealing with a formula of the individual fragment anymore. We have thus moved from an NP-complete problem to a PSPACE-complete problem.

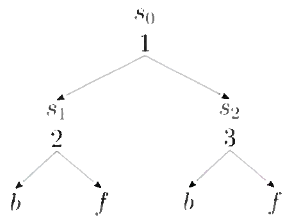
To consider a slightly different situation, suppose that the three daughters have already moved out from the parents' house and live abroad in different countries. The father then decides to call his daughters in order to solicit their opinions. Since he wants to spend as little money as possible, he wonders how many phone calls he will have to make, guaranteeing that any two of his daughters can determine the book he will buy. As a consequence, the implementation $\mathcal{M}_1$ is not practicable since he can only call each daughter individually to get her vote. We thus have to check for satisfiability in weakly individualistic coalition models which enforce sequential decision making.

Formally, consider the formula

$$\rho^k = \bigwedge_{\{C \subseteq N \,:\, |C|>1\}} ([C]^k p \wedge [C]^k q) \wedge \bigwedge_{1 \leq i \leq k} [\emptyset]^i \neg(p \wedge q)$$

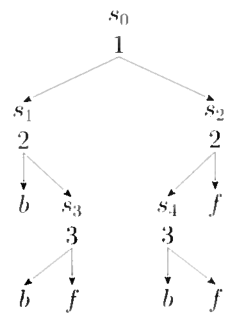
for $k \geq 1$. As will become clear, this formula expresses that all majorities are able to determine the outcome after at most k moves. While we have already seen that $\rho^1 = \rho$ is satisfiable in the class of weakly playable coalition models, it is not satisfiable in the class of weakly individualistic coalition models, for $\vdash_{\mathtt{Ind}} \rho \to \bot$: From $[\{1, 2\}]p$ and axiom $\mathtt{D}$ we can derive w.l.o.g. $[1]p$. But since we also have $[\{2, 3\}]q$, axiom $\mathtt{D}$ again gives us e.g. $[2]q$, and hence by superadditivity $[\{1, 2\}](p \wedge q)$. Together with $[\emptyset]\neg(p \wedge q)$, superadditivity again gives us $[\{1, 2\}]\bot$, a contradiction. Hence, ρ^1 is not satisfiable in weakly individualistic coalition models, showing that the father needs to make more than 1 phone call.

Maybe somewhat surprisingly, it turns out that ρ^2 is satisfiable already. The satisfying model is described by the following extensive game of perfect information $\mathcal{M}_2$:



As in $\mathcal{M}_1$, p is true at b only and q is true only at f. The effectivity at the terminal states does not matter. According to this model, the father first calls his daughter 1. He explains the situation to her, that he has to choose between two books, and asks her which of her two sisters he should call next. This second daughter chosen will then be able to determine by herself which book the father will buy. This procedure still gives any majority of daughters a strategy to determine the book the father will buy.

Thus, the father manages to solve his problem by calling only 2 of his 3 daughters. Hence, the procedure described by $\mathcal{M}_2$ is more efficient (i.e., less costly to the father) then the naive procedure $\mathcal{M}_3$ below of calling daughters in some order until the majority is clear.



At each position of the game, making the left move corresponds to voting for Beauvoir and making the right move to voting for Faludi. In the worst case, the first two daughters called will have different views and hence the father needs to call all of his daughters. Consequently, $\mathcal{M}_3$ does not satisfy ρ^2 at s_0, it only satisfies ρ^3, provided in $\mathcal{M}_3$ we let $V(p) = \{b\}$ and $V(q) = \{f\}$ as before. Furthermore, we need to assume that at states b and f we have terminal loops, i.e., there is a 1-player dictatorship associated with these states whose only possible successor is the state itself. It is because of these terminal loops that ρ^3 indeed holds at s_0, since it allows us to avoid complications with branches which are shorter than 3 moves.

While game $\mathcal{M}_2$ is more efficient, it puts some additional burden on the sisters. To start with, when called by the father, daughter 1 may not know her sisters' preferences, so before being able to suggest a sister to her father, she may have to call her sisters to find out their views. If one of her sisters agrees with her, daughter 1 can call her father back and suggest that sister, yielding a strategy for these two sisters to achieve the book of their choice. Two things should be noted here: First, while the father's telephone costs go down, his daughters telephone costs may increase. Second, even if 1 has found a sister (say 2) who agrees with her preference, sister 2 may, when called by the father, vote for the other book nonetheless. In other words, daughter 1 has no means to enforce an agreement she made with her sister.

Finally, note that if the father does not care about money, we can express the problem more naturally in Extended Coalition Logic. Instead of asking for each k whether ρ^k is satisfiable, we can simply check the satisfiability of

$$[\emptyset]^{\times}\neg(p \wedge q) \wedge \bigwedge_{\{C \subseteq N\,:\,|C|>1\}} ([C]^{t}p \wedge [C]^{t}q)$$

over weakly individualistic coalition models. The first conjunct ensures that p and q are never true simultaneously, and the second conjunct makes sure that each majority of sisters is able to achieve each alternative at the end. Hence both $\mathcal{M}_2$ and $\mathcal{M}_3$ satisfy this formula, and in fact any telephone procedure which is democratic overall will satisfy it, no matter how inefficient it is.

5.3 Eternal Voting

Because the examples given so far were finite games, the language of basic Coalition Logic sufficed as a description language. But once we turn to social procedures which may not reach any final terminal state, the increased expressive power of Extended Coalition Logic is needed, as the following example will illustrate. Using a voting agenda which may lead to infinite voting, we shall also demonstrate the difference between partial and total terminal effectivity which can serve as a basis to order different groups of agents according to their power.

Consider a political body $N = \{1, 2, 3, 4, 5, 6\}$ which has to decide on passing a new law. First, a subcommittee $D = \{2, 3, 4\}$ has to decide (by majority) which precise version of the law is to be presented to the full political body. Subsequently, the whole political body decides whether the law is passed or not. Again, the majority of the votes decides, and in case of a draw, the vote of the chairman 1 is decisive. If the law (as proposed by committee D) is not passed, the initiative is returned to committee D which has to make a new proposal for the law, and the process repeats itself.

We assume for simplicity that there are only two versions of the law which are under discussion, version 1 and version 2. If the body N rejects the proposal of committee D, the committee can either decide to propose the other version of the law, or it can resubmit its original proposal, possibly resulting in a stalemate which may turn into an infinite loop. (Some might claim that this model is sufficiently realistic to capture the essentials of the legislative process in some countries.) Figure 5.1 depicts the situation as a graph.

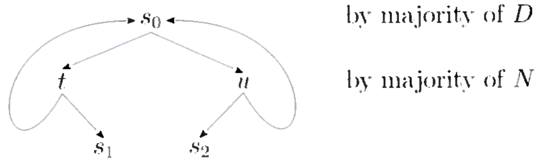


Figure 5.1: An example of binary majority voting with subcommittees.

One can think of the situation described in terms of coalitional effectivity: $s_0 E_C X$ holds iff at state s_0, coalition C can force the local voting outcome to lie in set X, i.e., iff one of the following two conditions is met: (1) $\{t, u\} \cap X \neq \emptyset$ and $|C \cap D| > 1$, or (2) $\{t, u\} \subseteq X$. Analogous definitions can be given for $tE_C X$ and $uE_C X$, incorporating the special role of the chairman. Since we are here not interested in winning and losing per se, the empty games associated with the terminal states s_1 and s_2 can be defined in an arbitrary way.

Let $\mathcal{M} = ((S, \{E_C | C \subseteq N\}), V)$ be the model which captures the procedure depicted in figure 5.1, where $\Phi_0 = \{law_1, law_2, dlock\}$ and $V(law_i) = \{s_i\}$, and $V(dlock) = \{s_0, t, u\}$. Note that $\mathcal{M}$ is maximal and regular for all coalitions (at least at nonterminal states), so that we can make use of the duality stated in theorem 4.3. Furthermore, $\mathcal{M}$ is also weakly playable since the democratic voting process at every state is a strategic game among the 6 players.

As the designer of a voting procedure such as figure 5.1, we may want to know whether it can be manipulated in ways we consider undesirable, i.e., whether an agent or a group of agents has a strategy to achieve an outcome which it considers desirable but which we as the designer of the procedure would consider undesirable in terms of the social welfare of all agents. For example, a certain group of agents may have an incentive to delay passing a new law as long as possible, preferably indefinitely. As the designer, we may want our voting procedure to prevent any group of agents from steering the voting process into an infinite loop. As is easily seen, the voting procedure of figure 5.1 is not strategy-proof in this respect: Both $\{1, 2, 3\}$ and $\{1, 4, 5\}$ can globally maintain $\{s_0, t, u\}$, i.e., they have the power to keep the process going forever, never reaching any decision. In the ECL formalization, checking for strategy-proofness becomes a question of model checking, verifying, e.g., that $\mathcal{M}, s_0 \models [\{1, 2, 3\}^\times] dlock \wedge [\{1, 4, 5\}^\times] dlock$.

More generally, we can also use this example to illustrate goal maintenance, partial and total terminal effectivity. Figure 5.2 displays some interesting examples which demonstrate the unequal powers of four 3-player coalitions at the initial state s_0: Total terminal effectivity (t) refers to cases where a coalition C is able to eventually achieve a set of terminal states X, formally $s_0 E^t_C X$. Goal maintenance (m) refers to the ability to maintain a property throughout the whole game, i.e., $s_0 E_C^{\times} X$. Finally partial terminal effectivity describes the ability of a coalition to guarantee a state in X or an infinite play of the game, $s_0 E_C^{p} X$.

$coalition \backslash states$	$\{s_1\}$	$\{s_1, s_2\}$	$\{s_0, t, u\}$
$\{1, 2, 3\}$	t	t	m
$\{2, 3, 4\}$	p	p	$-$
$\{1, 4, 5\}$	$-$	t	m
$\{4, 5, 6\}$	$-$	$-$	$-$

Figure 5.2: Goal maintenance (m), total (t) and partial (p) terminal effectivity in the voting example of figure 5.1 at state s_0.

At the initial state, $\mathcal{M}, s_0 \models [\{1, 2, 3\}^t]law_1 \wedge [\{1, 2, 3\}^t]law_2 \wedge [\{1, 2, 3\}^{\times}]dlock$, i.e., the coalition $\{1, 2, 3\}$ can achieve any possible outcome as well as a stalemate, whereas the coalition $\{1, 4, 5\}$ is weaker, $\mathcal{M}, s_0 \models [\{1, 4, 5\}^t](law_1 \vee law_2)$ but on the other hand $\mathcal{M}, s_0 \not\models [\{1, 4, 5\}^t]law_1 \vee [\{1, 4, 5\}^t]law_2$. Furthermore, $\mathcal{M}, s_0 \models [\{1, 4, 5\}^{\times}]dlock$, so this coalition can block any law from getting passed. Even weaker, coalition $\{4, 5, 6\}$ has virtually no power, since its counter-coalition $\{1, 2, 3\}$ is all-powerful. Thus, these facts about truth in a given model are the logical analogue of figure 5.2.

Based on the different global, partial and terminal abilities of the various coalitions, one can obtain an ordering of groups of agents with respect to their abilities. Inspecting figure 5.2, the following partial ability order of these four 3-player coalitions emerges:

$$\{4, 5, 6\} \prec \{2, 3, 4\}, \{1, 4, 5\} \prec \{1, 2, 3\}$$

Note that the coalitions $\{2, 3, 4\}$ and $\{1, 4, 5\}$ are incomparable: On the one hand, coalition $\{2, 3, 4\}$ is more powerful since it is able to pass any law provided that the procedure terminates eventually. On the other hand, coalition $\{1, 4, 5\}$ can force the procedure to terminate or keep it going forever, something which coalition $\{2, 3, 4\}$ cannot do.

5.4 Bonn vs. Berlin

After the "German Question" had been solved, on June 20th, 1991, the German parliament was faced with the Berlin question: Should the German parliament and the seat of government move to Berlin or stay in Bonn? In this historic debate, the parliament was very divided and this division ran through all parties. About 100 speeches were made while another 100 speeches were placed on record. We present this debate here as a real-life example of the importance of agenda choice, and to see how far Coalition Logic goes in helping in the analysis of such real-life examples.

In its debate, the German parliament considered 5 different motions, the 3 central ones being:

p_1 Parliament and government move to Berlin.

p_2 The parliament moves to Berlin but the seat of government remains in Bonn.

p_3 Both parliament and government remain in Bonn.

The two other motions did not play an essential role concerning the final decision, and we shall simplify our discussion by not considering them.

Since there were more than 2 motions up for vote, the parliamentary council of elders first had to decide on a voting procedure. Let us put ourselves into the shoes of the designers of the voting agenda, assuming that we have at our disposal an automatic agenda generator (i.e., a satisfiability checker for Coalition Logic). We can think of a voting procedure as an agenda tree which is an extensive game where each game position is associated to the set of alternatives which have not been eliminated yet.

Formally, we start at a position where $p_1 \wedge p_2 \wedge p_3$ holds. Each move in the voting game will then eliminate some alternative(s) until only a single alternative remains. The first requirement we impose is that alternatives which have been eliminated at some stage remain eliminated:

$$\neg p \rightarrow [\emptyset]\neg p \quad \text{for all atomic } p. \tag{5.1}$$

Second, bearing in mind the eternal voting example of the previous section, we want to prevent a voting procedure where no progress is made, demanding that at every stage at least one alternative has to be eliminated. We can formalize this requirement by stating that every move has to change the situation. Let $Sit = \{l_1 \wedge l_2 \wedge l_3 | l_i = p_i \text{ or } l_i = \neg p_i\}$ denote the set of possible situations, all possible combinations of atomic facts. Then we can write our second axiom as

$$\bigwedge_{\delta \in Sit} (\delta \rightarrow [\emptyset]\neg\delta). \tag{5.2}$$

Third, we want each vote of the voting procedure to be a democratic majority vote, formalized by

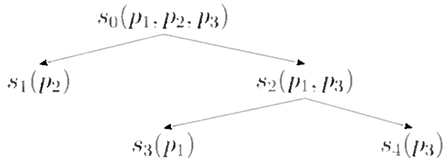
$$[N]\varphi \rightarrow [C]\varphi \quad \text{for every } C \text{ with } |C| > \tfrac{1}{2}|N| \tag{5.3}$$

where (to simplify the discussion) we assume that $|N|$ is odd. As a final fourth requirement, we want each vote to be between two alternatives only. This will allow us to exclude the problematic situation where a vote is taken between three alternatives none of which gets a majority. The axiom which ensures this binary decision making is

$$[N]\varphi \wedge [N](\neg\varphi \wedge \psi) \rightarrow [\emptyset](\varphi \vee \psi). \tag{5.4}$$

The axiom ensures that if a state has two successors which can be distinguished by φ such that one successor satisfies φ and the other ψ, then all successor states have to verify φ or ψ.

Being the agenda designers, we would like to know whether the formula $p_1 \wedge p_2 \wedge p_3$ is satisfiable in a weakly playable coalition model in which the four axioms (5.1) to (5.4) are valid. Giving these axioms to our automatic agenda generator, we receive a positive answer and the following satisfying model $\mathcal{M}_1$:



The propositional variables true at each state can be seen in the figure. At states s_0 and s_2, the decision between the left and the right move is made by majority. To the terminal states we can associate arbitrary empty games. It is easy to see that the four axioms are valid in $\mathcal{M}_1$. Intuitively, according to this procedure the parliament first votes whether its seat and the government offices should be in the same city. If so, the decision is made where the parliament and the government offices should be located.

Procedure $\mathcal{M}_1$ is the one actually adopted by the council of elders, and using this procedure the parliament ended up deciding for alternative p_1. It does, however, have a structural property which one might consider undesirable, namely that not all 3 alternatives are treated equally. In the first vote of $\mathcal{M}_1$, alternative p_2 has to compete against two other alternatives, p_1 and p_3 together. There may be a reason for making life difficult for alternative p_2, e.g., one might argue that separating the seat of parliament from the seat of government causes logistic problems which should only be accepted if a majority prefers this option to the other two together. If on the other hand we feel that all alternatives should be treated equally, we will not want to adopt $\mathcal{M}_1$, and instead are led to accept a fifth axiom which allows at most one alternative to be excluded in every vote (an agenda which has this property is called *complete*):

$$p \wedge q \rightarrow [\emptyset](p \vee q) \quad \text{for all atomic } p \neq q. \tag{5.5}$$

Note that this axiom is not valid in $\mathcal{M}_1$ since it fails at s_0 for $p_1 \wedge p_3$. Together with axiom (5.2), the new axiom ensures that at every vote exactly one alternative is excluded. Thus, if we happen to be unhappy with the first model presented to us by the agenda generator, we can add this additional axiom and again ask for a voting procedure satisfying the new specification, yielding, e.g., the following model $\mathcal{M}_2$:

$$
\begin{tikzcd}[column sep=small]
& & & s_0(p_1, p_2, p_3) \arrow[dll] \arrow[drr] & & & \\
& s_1(p_1, p_3) \arrow[dl] \arrow[dr] & & & & s_2(p_2, p_3) \arrow[dl] \arrow[dr] & \\
s_3(p_1) & & s_4(p_3) & & s_5(p_2) & & s_4(p_3)
\end{tikzcd}
$$

Agenda $\mathcal{M}_2$ satisfies all the axioms. First, a decision is made between p_1 and p_2, the winner of the first vote is then put up against p_3. Note also that an additional structural property distinguishes $\mathcal{M}_2$ from $\mathcal{M}_1$: While the number of votes taken in $\mathcal{M}_1$ depends on how earlier votes turn out, in $\mathcal{M}_2$ the number of votes is fixed, the agenda is *uniform*. Axiomatically, an agenda is uniform if it satisfies the following axiom

$$[N]^k[\bot] \to [\emptyset]^k[\bot] \tag{5.6}$$

for all $k > 0$.

To conclude, we hope to have shown that Coalition Logic can be a useful tool in the design of voting procedures. Even basic Coalition Logic is expressive enough to formalize structural properties of agendas which have been studied in the theory of voting. The importance of such structural properties can be seen in the Bonn vs. Berlin case: As discussed in [115], one can construct a preference profile for the members of parliament which yields the actual voting outcome p_1 under agenda $\mathcal{M}_1$, but outcome p_3 under agenda $\mathcal{M}_2$. While it is argued in [80] that both voting agendas would have led to the same result p_1, it suffices to illustrate that the outcome of political decision making can be highly sensitive to the agenda structure. It is disconcerting to think that possibly not the German parliament decided to move to Berlin but rather its council of elders, when it chose agenda $\mathcal{M}_1$ over agenda $\mathcal{M}_2$.

(As noted in [80], the actual voting agenda with its 5 motions exhibits a further peculiar logical feature. The fourth motion under consideration by the German parliament stated that in order to preserve the functioning of parliamentary democracy, the seat of parliament and the seat of government should not be geographically separated. After having voted whether or not to accept motion p_2, the parliament voted on this fourth motion p_4. In the actual course of events, motion p_2 was rejected (i.e., no separation of government and parliament) while motion p_4 was subsequently also rejected, seemingly adopting the proposition $\neg A \wedge \neg\neg A$. Thus, a closer look at the Bonn-Berlin debate shows that classical 2-valued logic seems insufficient to capture political discourse, at least in Germany, possibly due to its rich history in philosophical dialectics.)

5.5 Bibliographic Notes

The example and discussion of section 5.3 is based on [103, 102], but the formalization here is somewhat different.

As deontic models of constitutions, effectivity functions have been used in [50, 105, 106, 64]. The characterization of effectivity functions which correspond to Nash-consistent strategic games is from [106]. The example of section 5.1 concerning the colored shirts is from [52], where the colors refer to wallpaper.

Miller in [84] gives a survey of agenda issues in committee voting, covering agenda trees and their properties such as completeness and uniformity.

Chapter 6

Game Logic

As we have seen in section 2.4, effectivity functions can be used to model local as well as global ability of players in a game. For Coalition Logic, we have assumed that a (coalition) model represents a complex game whose local effectivity structure is described by the $[C]$-modality. We are now switching from this internal game view to an external view in which we only represent games by what players can achieve in them in the end. The game models underlying this external view associate multiple determined 2-player games to every state. The logic used to reason about these models is Game Logic, introduced in [97], which includes game expressions explicitly in its logical language. Where in Coalition Logic $[C]\varphi$ expressed that coalition C had a strategy for bringing about φ in one move in the game which is the underlying model, the Game Logic formula $\langle\gamma\rangle\varphi$ expresses that player 1 has a strategy to bring about φ in game γ.

We formally introduce the syntax and semantics of Game Logic in section 6.1. In section 6.2, we show that all the operations of Game Logic preserve bisimulation so that (as with Coalition Logic) all formulas of Game Logic are invariant for bisimulation. The rest of this chapter again concerns questions of axiomatization and complexity. Unfortunately, the results are somewhat incomplete and leave many open questions: A completeness result for the natural axiom system for full Game Logic is still wanting, as is a proof of the conjectured lower bound for the complexity of the satisfiability problem.

For studying the complexity of Game Logic, it is useful to relate Game Logic to another well-known calculus for reasoning about program behavior, the μ-calculus. Although formulas of the μ-calculus have traditionally been interpreted over Kripke models, this restriction is in no way necessary. Section 6.4 introduces the μ-calculus over general game models.

The chapter closes with some brief remarks on our second theme, programs vs. games. While most of this discussion is postponed until the end of the next chapter, we comment on some first differences and similarities between full Game Logic and its program fragment.

6.1 Syntax & Semantics

Game Logic (GL) is a logic to reason about individual ability in determined 2-player games; it extends the individual fragment of $2-$`MaxPlay` in two ways: First, multiple effectivity functions are associated with each state, one for every game in Γ_0, and second, game operations are added to talk not only about atomic games but also about complex ones. To add some metaphysical significance to these 2-player games and to adopt established terminology from the literature on the refinement calculus [7], player 1 will often be called Angel and player 2 Demon. The language of *GL* consists of two sorts, games and propositions.

▶ **Definition 6.1** (Game Logic Syntax). Given a set of atomic games Γ_0 and a set of atomic propositions Φ_0, games γ and propositions φ can have the following syntactic forms, yielding the set of GL-games Γ and the set of GL-propositions/formulas Φ:

$$\begin{aligned} \gamma :=\ & g \mid \varphi? \mid \gamma;\gamma \mid \gamma \cup \gamma \mid \gamma^* \mid \gamma^d \\ \varphi :=\ & \bot \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\gamma\rangle\varphi \end{aligned}$$

where $p \in \Phi_0$ and $g \in \Gamma_0$.

Furthermore, we define $[\gamma]\varphi := \neg\langle\gamma\rangle\neg\varphi$ and the demonic analogues of angelic choice and iteration: Demonic choice between γ_1 and γ_2 is denoted as $\gamma_1 \cap \gamma_2$ which abbreviates $(\gamma_1^d \cup \gamma_2^d)^d$. Demonic iteration of γ is denoted as $\gamma^\times$ which abbreviates $((\gamma^d)^*)^d$.

On certain occasions, it is useful to treat demonic choice and iteration as primitive, for it allows us to consider Game Logic formulas and games in dual normal form:

▶ **Definition 6.2** (Dual Normal Form). A GL-formula (GL-game) is in *dual normal form* iff the duality operator occurs only as demonic iteration, demonic choice, or in front of atomic games or tests.

Using the following game-theoretic versions of the de Morgan laws, every Game Logic formula can be rewritten into an equivalent dual normal form:

$$\begin{array}{lll} (\alpha \cup \beta)^d \rightsquigarrow \alpha^d \cap \beta^d & (\alpha;\beta)^d \rightsquigarrow \alpha^d;\beta^d & \alpha^{*d} \rightsquigarrow \alpha^{d\times} \\ (\alpha \cap \beta)^d \rightsquigarrow \alpha^d \cup \beta^d & & \alpha^{\times d} \rightsquigarrow \alpha^{d*} \end{array}$$

▶ **Definition 6.3** (Program Fragment). A game which does not contain the duality-operation at all (i.e., also not hidden in a demonic operation) is a *program*. The set of GL-formulas which only contain games which are programs is the *program fragment* of Game Logic.

The formula $\langle\gamma\rangle\varphi$ expresses that Angel has a φ-strategy in game γ, and $[\gamma]\varphi$ expresses that Angel does not have a $\neg\varphi$-strategy, which by determinacy is equivalent to saying that Demon has a φ-strategy. To provide some first intuition regarding the game operations, $\gamma_1 \cup \gamma_2$ denotes the game where Angel chooses which of the two subgames to continue playing, and the sequential composition $\gamma_1;\gamma_2$ of two games consists of first playing γ_1 and then γ_2. In the iterated game γ^*, Angel can choose how often to play γ (possibly not at all); each time she has played γ, she can decide whether to play it again or not. Playing the dual game γ^d is the same as playing γ with the players' roles reversed, i.e., any choice made by Angel in γ will be made by Demon in γ^d and vice versa. Hence, $\gamma_1 \cap \gamma_2$ will refer to the game where Demon chooses which subgame to play, leaving the roles of the players in γ_1 and γ_2 intact. The test game $\varphi?$ consists of checking whether a proposition φ holds at that position. This construction can be used to define conditional games such as $(p?;\gamma_1) \cup (\neg p?;\gamma_2)$: If p holds at the present state of the game, γ_1 is played, and otherwise γ_2.

Recall from chapter 2 that a *game model* $\mathcal{M} = ((S, \{E_g | g \in \Gamma_0\}), V)$, consists of a set of states S, a valuation $V : \Phi_0 \to \mathcal{P}(S)$ for the propositional letters and a collection of functions $E_g : S \to \mathcal{P}(\mathcal{P}(S))$ which are monotonic, i.e. $X \in E_g(s)$ and $X \subseteq X'$ imply $X' \in E_g(s)$. The idea is that sE_gX (i.e., $X \in E_g(s)$) holds whenever Angel has a strategy in game g to achieve X, i.e., E_g represents the effectivity of player 1 at every state. By the proof of theorem 2.12 and corollary 2.13 of chapter 2 we know that we can think of every state as being associated either with a number of determined strategic games (internal view), or with a number of extensive games of perfect information (external view).

By simultaneous induction, we define truth in a game model on the one hand and the effectivity functions for non-atomic games on the other hand.

▶ Definition 6.4 (Game Logic Semantics). The truth of a formula φ in a model $\mathcal{M}$ at a state s (denoted as $\mathcal{M}, s \models \varphi$) is defined as follows:

$$\begin{array}{ll}
\mathcal{M}, s \not\models \bot & \\
\mathcal{M}, s \models p & \text{iff } p \in \Phi_0 \text{ and } s \in V(p) \\
\mathcal{M}, s \models \neg\varphi & \text{iff } \mathcal{M}, s \not\models \varphi \\
\mathcal{M}, s \models \varphi \vee \psi & \text{iff } \mathcal{M}, s \models \varphi \text{ or } \mathcal{M}, s \models \psi \\
\mathcal{M}, s \models \langle\gamma\rangle\varphi & \text{iff } sE_\gamma\varphi^{\mathcal{M}}
\end{array}$$

The function $E_\gamma : S \to \mathcal{P}(\mathcal{P}(S))$ is defined inductively for non-atomic games γ. Let $E_\gamma(Y) := \{s \in S | sE_\gamma Y\}$. Then

$$\begin{array}{rl}
E_{\alpha;\beta}(Y) & := E_\alpha(E_\beta(Y)) \\
E_{\alpha\cup\beta}(Y) & := E_\alpha(Y) \cup E_\beta(Y) \\
E_{\varphi?}(Y) & := \varphi^{\mathcal{M}} \cap Y \\
E_{\alpha^d}(Y) & := \tilde{E}_\alpha(Y) = \overline{E_\alpha(\overline{Y})} \\
E_{\alpha^*}(Y) & := \mu X.Y \cup E_\alpha(X)
\end{array}$$

It can be shown that monotonicity of the E_g-functions is preserved under the game operations, so the fixpoint $\mu X.Y \cup E_\alpha(X)$ always exists. We also define the notions of validity and logical consequence in the standard way (see section 3.1).

6.2 Bisimulation Again

As an equivalence notion, bisimulation applies to game models as it does to coalition models. We have seen in a previous chapter that all formulas of Coalition Logic were bisimulation invariant. To show that also Game Logic formulas are invariant for bisimulation, one needs to show that clauses 2 and 3 of the definition of bisimulation can be generalized from atomic games to complex games.

▶ THEOREM 6.5. Let $\mathcal{M} = ((S, \{E_g | g \in \Gamma_0\}), V)$ and $\mathcal{M}' = ((S', \{E'_g | g \in \Gamma_0\}), V')$ be two game models such that $s \leftrightarrows s'$. Then

1. For all $\varphi \in \Phi$: $\mathcal{M}, s \models \varphi$ iff $\mathcal{M}', s' \models \varphi$
2. For all $\gamma \in \Gamma$: If $sE_\gamma X$ then $\exists X' \subseteq S'$ such that $s'E'_\gamma X'$ and $\forall x' \in X'\ \exists x \in X : x \leftrightarrows x'$.
3. For all $\gamma \in \Gamma$: If $s'E'_\gamma X'$ then $\exists X \subseteq S$ such that $sE_\gamma X$ and $\forall x \in X\ \exists x' \in X' : x \leftrightarrows x'$.

PROOF. For atomic games and formulas, the claims hold by bisimilarity. For non-atomic formulas, the boolean cases are immediate and we shall only show one direction of (1.) for $\langle\gamma\rangle\varphi$. If $\mathcal{M}, s \models \langle\gamma\rangle\varphi$, $sE_\gamma\varphi^{\mathcal{M}}$ and so (by induction hypothesis (2.) for γ) there is some X' such that $s'E'_\gamma X'$ and for all $x' \in X'$ there is some $x \in \varphi^{\mathcal{M}}$ such that $x \leftrightarrows x'$. By induction hypothesis (1.) for φ, this means that $X' \subseteq \varphi^{\mathcal{M}'}$, and so by monotonicity, $s'E'_\gamma\varphi^{\mathcal{M}'}$, which establishes that $\mathcal{M}', s' \models \langle\gamma\rangle\varphi$.

As for proving that the game constructions of *GL* are safe for bisimulation, we shall prove (2.) for non-atomic games. Consider first the case of test $\varphi?$: If $sE_{\varphi?}X = \varphi^{\mathcal{M}} \cap X$, let $X' := \{x' | \exists x \in X : x \leftrightarrows x'\}$. Then $s'E'_{\varphi?}X'$ by induction hypothesis (1.) for φ, and for all $x' \in X'$ there is some $x \in X$ such that $x \leftrightarrows x'$, simply by definition of X'.

For union, if $sE_{\alpha\cup\beta}X$ we can assume w.l.o.g. that $sE_\alpha X$ and apply the induction hypothesis, i.e. for some X', we have $s'E'_\alpha X'$ and hence also $s'E'_{\alpha\cup\beta}X'$.

For composition, suppose that $s \in E_\alpha(E_\beta(X))$. Using the induction hypothesis for α, there is some Y' such that $s'E'_\alpha Y'$ and for all $y' \in Y'$ there is a $u \in E_\beta(X)$ such that $u \leftrightarrows y'$. Now let $X' := \{x' | \exists x \in X : x \leftrightarrows x'\}$. We must show that $s'E'_\alpha(E'_\beta(X'))$. For this, it suffices by monotonicity to show that $Y' \subseteq E'_\beta(X')$. So suppose that $y' \in Y'$, i.e., for some $u \in E_\beta(X)$ we have $u \leftrightarrows y'$. Using the induction hypothesis for β, there is some V' such that $y'E'_\beta V'$ and for all $v' \in V'$

there is some $x \in X$ such that $x \leftrightarrows v'$. Hence $V' \subseteq X'$ and so by monotonicity, $y'E'_\beta X'$.

Dual: Suppose $sE_{\alpha^d}X$, i.e. not $sE_\alpha\overline{X}$ and let X' be as before. It is sufficient to show that $s'E'_\alpha\overline{X'}$ does not hold. Suppose by reductio the contrary. Then there is some Z with $sE_\alpha Z$ and for all $z \in Z$ there is some $x' \notin X'$ such that $z \leftrightarrows x'$. From this it follows that $Z \subseteq \overline{X}$, so by monotonicity $sE_\alpha\overline{X}$, a contradiction.

Iteration: Let X' be as before and let Z be

$$\{z|\forall z' : z \leftrightarrows z' \Rightarrow z'E'_{\alpha^*}X')\}.$$

Now it is sufficient to show that $E_{\alpha^*}(X) \subseteq Z$, and given the definition of $E_{\alpha^*}(X)$ as a least fixpoint, it suffices to show that Z is a fixpoint, i.e. that

$$X \cup E_\alpha(Z) \subseteq Z.$$

Supposing that $x \in X$ and for some x' we have $x \leftrightarrows x'$, we have $x' \in X' \subseteq E'_{\alpha^*}(X')$. On the other hand, suppose that $x \in E_\alpha(Z)$ and $x \leftrightarrows x'$. Then by induction hypothesis, there is some Z' such that $x' \in E'_\alpha(Z')$ and for all $z' \in Z'$ there is some $z \in Z$ such that $z \leftrightarrows z'$. But then $Z' \subseteq E'_{\alpha^*}(X')$, and so by monotonicity $x' \in E'_\alpha(E'_{\alpha^*}(X')) \subseteq E'_{\alpha^*}(X')$ which completes the proof. ■

6.3 Axiomatization

▶ DEFINITION 6.6 (GAME LOGIC AXIOMATICS). A *game logic* is a set of formulas Λ which contains all propositional tautologies together with all instances of the axiom schemas of figure 6.1, and which is closed under the rules of Modus Ponens, Monotonicity and a new Fixpoint Rule. Let $\mathtt{GL}$ denote the smallest game logic.

Intuitively, the axiom for iteration states that $\langle\gamma^*\rangle\varphi$ is a fixpoint of the operation $\varphi \vee \langle\gamma\rangle X$ and the fixpoint rule states that $\langle\gamma^*\rangle\varphi$ is the least such fixpoint.

▶ THEOREM 6.7. $\mathtt{GL}$ is sound with respect to the class of all game models.

Together with Parikh, we conjecture that $\mathtt{GL}$ is complete with respect to the class of all game models, but proving this conjecture is one of the main open technical problems in Game Logic. Some weaker results exist, however. If sq is a sequence of operators of Game Logic such as d or $d, *$, let $\mathtt{GL}^{-sq}$ denote Game Logic without the operators of sq, i.e., restricted to formulas without these operators and without the axioms involving them.

▶ THEOREM 6.8 (PARIKH [97]). Dual-free Game Logic $\mathtt{GL}^{-d}$ is sound and complete with respect to the class of all game models.

Axioms:

$$\langle\alpha \cup \beta\rangle\varphi \leftrightarrow \langle\alpha\rangle\varphi \vee \langle\beta\rangle\varphi$$
$$\langle\alpha;\beta\rangle\varphi \leftrightarrow \langle\alpha\rangle\langle\beta\rangle\varphi$$
$$\langle\delta?\rangle\varphi \leftrightarrow (\delta \wedge \varphi)$$
$$\langle\gamma^d\rangle\varphi \leftrightarrow \neg\langle\gamma\rangle\neg\varphi$$
$$(\varphi \vee \langle\gamma\rangle\langle\gamma^*\rangle\varphi) \rightarrow \langle\gamma^*\rangle\varphi$$

Inference Rules:

$$\frac{\varphi \quad \varphi \rightarrow \psi}{\psi} \qquad \frac{\varphi \rightarrow \psi}{\langle C\rangle\varphi \rightarrow \langle C\rangle\psi} \qquad \frac{(\varphi \vee \langle\gamma\rangle\psi) \rightarrow \psi}{\langle\gamma^*\rangle\varphi \rightarrow \psi}$$

Figure 6.1: The axioms and inference rules (Modus Ponens, Monotonicity and the Fixpoint Rule) of Game Logic.

The rest of this section is devoted to showing that $\mathtt{GL}^{-*}$, i.e., Game Logic with dual but without iteration (and hence without the Mix axiom $(\varphi \vee \langle\gamma\rangle\langle\gamma^*\rangle\varphi) \rightarrow \langle\gamma^*\rangle\varphi$ and the fixpoint rule) is complete with respect to the class of all game models. As before, the proof is via a canonical model construction.

Let Λ be any game logic and let S be the set of all maximally Λ-consistent sets of formulas. Define the canonical Λ-model $\mathcal{C} = ((S, \{E_g | g \in \Gamma_0\}), V)$ as before in the case of Coalition Logic:

$$\begin{aligned} s \in V(p) \quad & \text{iff } p \in s \\ sE_gX \quad & \text{iff } \exists\widehat{\varphi} \subseteq X : \langle g\rangle\varphi \in s \end{aligned}$$

► Lemma 6.9. For any maximally Λ-consistent set $s \in S$ and any formula φ: $\mathcal{C}, s \models \varphi$ iff $\varphi \in s$. Equivalently, $\varphi^{\mathcal{C}} = \widehat{\varphi}$.

Proof. We shall prove the following two claims by simultaneous induction on φ and γ:

$$(1)\ \varphi^{\mathcal{C}} = \widehat{\varphi} \quad \text{and} \quad (2)\ \forall\psi : E_\gamma(\widehat{\psi}) = \widehat{\langle\gamma\rangle\psi}$$

The base case of both claims holds by definition. For the boolean inductive steps of (1), the argument is standard. For $\langle\gamma\rangle\varphi$, suppose $s \in (\langle\gamma\rangle\varphi)^{\mathcal{C}}$ where $\varphi^{\mathcal{C}} = \widehat{\varphi}$ by induction hypothesis. Then the claim follows from (2). What remains is to show (2) for complex games γ making use of the axioms. Some sample cases:

If $sE_{\varphi?}\widehat{\psi}$ then $s \in \varphi^{\mathcal{C}} \cap \widehat{\psi}$. By induction hypothesis claim (1), $s \in \widehat{\varphi}$ and so $\varphi \wedge \psi \in s$, hence by the test axiom, $\langle\varphi?\rangle\psi \in s$. Similarly for the converse.

If $sE_\alpha E_\beta(\widehat{\psi})$, then by induction hypothesis, $E_\beta(\widehat{\psi}) = \widehat{\langle\beta\rangle\psi}$, and again by induction hypothesis, $\langle\alpha\rangle\langle\beta\rangle\psi \in s$. Analogously for the converse and the other cases. ■

The previous truth lemma then allows us to prove the canonical model theorem, from which completeness follows as a corollary. To conclude, we have axiomatic completeness for $\mathtt{GL}^{-d}$ as well as $\mathtt{GL}^{-*}$, but iteration together with duality remains a problem.

▶ THEOREM 6.10. $\mathtt{GL}^{-*}$ is complete with respect to the class of all game models.

6.4 Alternations

6.4.1 The Generalized μ-Calculus

The modal μ-calculus is a very expressive logic which subsumes most program logics (e.g., PDL) and temporal logics (e.g., CTL, CTL^*). In its original formulation, the μ-calculus consists of a modal language with special operations denoting least and greatest fixpoints, interpreted over Kripke models. In this section we shall introduce the language of the μ-calculus and generalize its semantics from Kripke models to game models. We can then translate formulas of Game Logic into formulas of the μ-calculus, and this translation will subsequently be appealed to when discussing expressiveness and complexity in this chapter and the next.

The language of the propositional modal μ-calculus consists of the language of modal logic together with least and greatest fixpoint operations which make use of variables $X, Y, \ldots \in Var$. Note that in contrast to GL, modalities are always atomic in the μ-calculus.

▶ DEFINITION 6.11 (μ-CALCULUS SYNTAX). The set of μ-calculus formulas is defined inductively as

$$\varphi := \bot \mid p \mid X \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle g \rangle \varphi \mid \mu X.\varphi$$

where $p \in \Phi_0$, $g \in \Gamma_0$, $X \in Var$ and in $\mu X.\varphi$, X occurs *strictly positively* in φ, i.e., every free occurrence of X in φ occurs under an even number of negations.

The language is interpreted over game models $\mathcal{M} = ((S, \{E_g | g \in \Gamma\}), V)$, but a variable assignment $v : Var \to \mathcal{P}(S)$ is needed to interpret the variables.

▶ DEFINITION 6.12 (μ-CALCULUS SEMANTICS). The truth of a formula φ in a model $\mathcal{M}$ at a state s (denoted as $\mathcal{M}, s \models \varphi$) is defined as follows:

$$\begin{array}{lll}
\mathcal{M}, v, s \not\models \bot & & \\
\mathcal{M}, v, s \models p & \text{iff} & p \in \Phi_0 \text{ and } s \in V(p) \\
\mathcal{M}, v, s \models X & \text{iff} & X \in Var \text{ and } s \in v(X) \\
\mathcal{M}, v, s \models \neg\varphi & \text{iff} & \mathcal{M}, v, s \not\models \varphi \\
\mathcal{M}, v, s \models \varphi \vee \psi & \text{iff} & \mathcal{M}, v, s \models \varphi \text{ or } \mathcal{M}, v, s \models \psi \\
\mathcal{M}, v, s \models \langle g \rangle \varphi & \text{iff} & sE_g\{t \in S | \mathcal{M}, v, t \models \varphi\} \\
\mathcal{M}, v, s \models \mu X.\varphi & \text{iff} & s \in \bigcap\{T \subseteq S | \{t \in S | \mathcal{M}, v_{X:=T}, t \models \varphi\} \subseteq T\}
\end{array}$$

where $v_{X:=T}(Y) = v(Y)$ for all $Y \neq X$ and $v_{X:=T}(X) = T$.

Since φ was assumed to be strictly positive in X, monotonicity is guaranteed and hence by theorem A.1, $\mu X.\varphi$ indeed denotes the least fixpoint of the operation associated with $\varphi(X)$. Note that the scope of the fixpoint operator extends as far as possible to the right. We define the greatest fixpoint $\nu X.\varphi(X)$ as the abbreviation of $\neg\mu X.\neg\varphi(\neg X)$.

▶ DEFINITION 6.13 (POSITIVE NORMAL FORM). Formula φ is said to be in *positive normal form* if no variable is quantified (i.e., bound by μ or ν) twice and all negations occurring in φ apply to atomic propositions only.

Using the de Morgan laws for the boolean connectives, the box-diamond duality and the greatest fixpoint, every μ-calculus formula can be rewritten into positive normal form.

When we will discuss model-checking algorithms in the next chapter, fixpoint nestings will play a crucial role. The simplest kind of nesting is exhibited by the following formula

$$\mu X.p \vee (X \wedge \mu Y.q \vee \langle g\rangle Y).$$

Since the variable X does not occur inside the μY fixpoint, the latter can be calculated first, independently of the μX fixpoint. We shall consider nestings of this kind to be *vacuous nestings*. A non-vacuous nesting is exhibited by the formula

$$\mu X.p \vee (q \wedge \mu Y.X \vee \langle g\rangle Y),$$

where the calculation of the inner fixpoint depends on the current value of X. We define the *fixpoint depth* $d(\varphi)$ of a μ-calculus formula φ as the maximal number of such nestings, ignoring vacuous nestings (we shall give a formal definition shortly). Finally, it will turn out that even nestings of the kind we just saw are not that bad after all, since the two fixpoints which are nested are both least fixpoints. The formula

$$\mu X.p \vee (q \wedge \nu Y.X \wedge \langle g\rangle Y)$$

on the other hand nests a greatest fixpoint inside a smallest fixpoint. This kind of nesting is formally captured by the notion of *alternation depth* [44].

▶ DEFINITION 6.14 (ALTERNATION DEPTH). If a μ-calculus formula φ is in positive normal form, we define its alternation depth $ad(\varphi)$ by induction on φ as follows:

$$\begin{array}{rcl}
ad(X) = ad(p) & = & 0 \text{ for } p \in \Phi_0,\ X \in Var \\
ad(\varphi \vee \psi) = ad(\varphi \wedge \psi) & = & max(ad(\varphi), ad(\psi)) \\
ad(\neg\varphi) & = & ad(\varphi) \\
ad(\langle g\rangle\varphi) = ad([g]\varphi) & = & ad(\varphi) \\
ad(\mu X.\varphi) & = & max(1, ad(\varphi), 1 + ad(\nu Y_1.\psi_1), \ldots, 1 + ad(\nu Y_n.\psi_n)) \\
 & & \quad \text{where } \nu Y_i.\psi_i \in sf(\varphi) \text{ and } X \text{ occurs free in } \psi_i \\
ad(\nu X.\varphi) & = & max(1, ad(\varphi), 1 + ad(\mu Y_1.\psi_1), \ldots, 1 + ad(\mu Y_n.\psi_n)) \\
 & & \quad \text{where } \mu Y_i.\psi_i \in sf(\varphi) \text{ and } X \text{ occurs free in } \psi_i
\end{array}$$

Thus for any μ-calculus formula φ, we can define its alternation depth by first rewriting it into positive normal form. The difference between fixpoint depth $d(\varphi)$ and alternation depth $ad(\varphi)$ lies in the last two clauses of the preceding definition: For $d(\mu X.\varphi)$ as well as $d(\nu X.\varphi)$, we add 1 to the maximal fixpoint depth of both μ- and ν-subformulas of φ which contain X free. Denote the set of μ-calculus formulas as L_μ, and let $L_{\mu_k} \subseteq L_\mu$ denote the set of those formulas which (when put into positive normal form) are of alternation depth at most k. Let $L^k_\mu \subseteq L_\mu$ denote the set of formulas which contain at most k distinct set variables.

6.4.2 Embedding Game Logic into the μ-Calculus

The following function $^\circ : GL \to L^2_\mu$ maps every Game Logic formula φ to a μ-calculus formula φ°, using two auxiliary translation functions on games, ^{x} and ^{y}.

$$
\begin{aligned}
g^x &= \langle g\rangle X & g^y &= \langle g\rangle Y\\
(\alpha\cup\beta)^x &= \alpha^x \vee \beta^x & (\alpha\cup\beta)^y &= \alpha^y \vee \beta^y\\
(\alpha;\beta)^x &= \alpha^x[X := \beta^x] & (\alpha;\beta)^y &= \alpha^y[Y := \beta^y]\\
(\varphi?)^x &= \varphi^\circ \wedge X & (\varphi?)^y &= \varphi^\circ \wedge Y\\
(\alpha^d)^x &= \neg\alpha^x[X := \neg X] & (\alpha^d)^y &= \neg\alpha^y[Y := \neg Y]\\
(\alpha^*)^x &= \mu Y.X \vee \alpha^y & (\alpha^*)^y &= \mu X.Y \vee \alpha^x
\end{aligned}
$$

$$
\begin{aligned}
p^\circ &= p\\
(\neg\varphi)^\circ &= \neg\varphi^\circ\\
(\varphi\vee\psi)^\circ &= \varphi^\circ \vee \psi^\circ\\
(\langle\alpha\rangle\varphi)^\circ &= \alpha^x[X := \varphi^\circ]
\end{aligned}
$$

The expression $\varphi[X := \psi]$ refers to the result of substituting ψ for every occurrence of X. Note that φ° will have no free variables so that we can simply write $\mathcal{M}, s \models \varphi^\circ$ instead of $\mathcal{M}, v, s \models \varphi^\circ$. The correctness of the translation can easily be proved by induction.

▶ Theorem 6.15. There is a translation function $^\circ : GL \to L^2_\mu$ such that for all game models $\mathcal{M}$ we have $\mathcal{M}, s \models \varphi$ iff $\mathcal{M}, s \models \varphi^\circ$.

While this translation from Game Logic into the μ-calculus allows us to talk about the alternation depth of a Game Logic formula, it is useful to define the notion of alternation depth for Game Logic directly.

▶ Definition 6.16 (Alternation depth). The *alternation depth* of a Game Logic formula φ which is in dual normal form is inductively defined as follows:

$$
\begin{aligned}
ad(p) &= 0 \text{ for } p \in \Phi_0 \\
ad(\varphi \vee \psi) &= max(ad(\varphi), ad(\psi)) \\
ad(\neg\varphi) &= ad(\varphi) \\
ad(\langle\gamma\rangle\varphi) &= max(ad(\gamma), ad(\varphi)) \\
ad(g) &= 0 \text{ for } g \in \Gamma_0 \\
ad(\varphi?) &= ad(\varphi) \\
ad(\gamma^d) &= ad(\gamma) \\
ad(\alpha \cup \beta) = ad(\alpha \cap \beta) &= max(ad(\alpha), ad(\beta)) \\
ad(\alpha; \beta) &= max(ad(\alpha), ad(\beta)) \\
ad(\gamma^*) &= max(1, ad(\gamma), 1 + ad(\alpha_1^\times), \ldots, 1 + ad(\alpha_n^\times)) \\
&\qquad \text{where } \alpha_i^\times \text{ is a subgame of } \gamma \text{ not in the scope of } ? \\
ad(\gamma^\times) &= max(1, ad(\gamma), 1 + ad(\alpha_1^*), \ldots, 1 + ad(\alpha_n^*)) \\
&\qquad \text{where } \alpha_i^* \text{ is a subgame of } \gamma \text{ not in the scope of } ?
\end{aligned}
$$

In this definition, maximization in $ad(\gamma^*)$ needs to be restricted to subgames of γ not in the scope of a test operator. This restriction is the Game Logic equivalent to the μ-calculus restriction to subformulas which contain the fixpoint variable free. As an example, consider the Game Logic formula $\langle(\langle g^\times\rangle q?)^*\rangle p$ which corresponds to the μ-calculus formula $\mu X.p \vee (X \wedge \nu Y.q \wedge \langle g\rangle Y)$ and has alternation depth 1 rather than 2. In general, while $(\varphi?)^x = \varphi^\circ \wedge X$ has a free variable, that variable will never be in the scope of any fixpoint operator resulting from translating φ.

The following theorem shows that using the above definition of alternation depth for Game Logic, the translation function preserves alternation depth.

▶ **Theorem 6.17.** For every Game Logic formula φ, $ad(\varphi) = ad(\varphi^\circ)$.

Proof. We show by simultaneous induction that for all Game Logic formulas φ and games γ, $ad(\varphi) = ad(\varphi^\circ)$ and $ad(\gamma) = ad(\gamma^x)$. We will assume here for simplicity that φ and γ are in dual normal form, and that the translation function ^{x} also contains the following clauses for the demonic game operations:

$$(\alpha \cap \beta)^x = \alpha^x \wedge \beta^x \text{ and } (\alpha^\times)^x = \nu Y.X \wedge \alpha^y$$

We shall briefly consider the most difficult inductive steps for sequential composition and iteration. Since $(\alpha;\beta)^x = \alpha^x[X := \beta^x]$, we need to show that $max(ad(\alpha^x), ad(\beta^x)) = ad(\alpha^x[X := \beta^x])$. To see that this is indeed the case, note that the substitution of β^x for X can only lead to an increase in alternation depth in case a free variable will become bound by an outer fixpoint operator. But this cannot happen since X is the only free variable in β^x which will still be free in $\alpha^x[X := \beta^x]$. An analogous argument is used to show that $max(ad(\gamma^x), ad(\varphi^\circ)) = ad(\gamma^x[X := \varphi^\circ])$.

For iteration, we need to show that $ad(\gamma^*) = ad(\mu Y.X \vee \gamma^y)$, where γ^y contains Y free. Inspecting the definitions of ad, it suffices to show two claims:

> (1) For every subformula $\nu Z.\psi$ of γ^y in which Y occurs free, there is a subformula $\alpha^\times$ of γ which is not in the scope of a test whose translation is $\nu Z.\psi$.

On the one hand, every subformula $\nu Z.\psi$ of γ^y must be the translation of a subformula $\alpha^\times$ of γ. On the other hand, if Y occurs free in ψ then $\alpha^\times$ cannot be in the scope of a test.

> (2) For every subformula $\alpha^\times$ of γ which does not occur in the scope of a test and for which $ad(\alpha^\times) = ad(\gamma)$, its translation $\nu Z.\psi$ will be a subformula of γ^y which contains Y free.

Note that we can also assume that $\alpha^\times$ does not occur in the scope of another $^\times$-iteration, i.e., there is no subformula $\beta^\times$ of γ such that $\alpha^\times$ is a subformula of β. For if that were the case, $ad(\beta^\times) \geq ad(\alpha^\times)$ and it suffices to consider $\beta^\times$ instead of $\alpha^\times$.

To see that $\nu Z.\psi$ must indeed contain Y free, note that the free variable Y in γ^y is passed on through the game operations with the exception of test and iteration. In other words, the only way in which Y could not be free in $\nu Z.\psi$ is if this formula occurs within the scope of another (translated) iteration, for we assumed it is not within the scope of a test. As mentioned, we can assume that $\nu Z.\psi$ occurs within the scope of a least fixpoint, i.e., it is a subformula of $\mu Z'.\delta$. But then we would have

$$ad(\gamma) \geq ad(\mu Z'.\delta) > ad(\nu Z.\psi) = ad(\alpha^\times)$$

which contradicts our assumption that $ad(\alpha^\times) = ad(\gamma)$. ■

Thus, while two variables suffice to translate Game Logic into the μ-calculus, iteration and duality allow one to create formulas of arbitrary alternation depth: there is no k such that for all Game Logic formulas φ, $ad(\varphi) \leq k$. If we define $g_1 = g$, and for $n > 0$ we let $g_{2n} = g^*_{2n-1}$ and $g_{2n+1} = g^\times_{2n}$, then $\langle g_n \rangle \bot$ will be a formula of alternation depth $n-1$. Note, however, that this result is only syntactic: It may very well be that there is some k such that for all GL formulas φ, φ° is equivalent to a formula of alternation depth at most k. We will have more to say on this matter in the next chapter (section 7.2). All that we know so far is that such a k would have to be greater than 1:

► THEOREM 6.18. There is a GL formula φ which is not equivalent to any μ-calculus formula of alternation depth less than 2.

PROOF. Consider the μ-calculus formula

$$\delta := \quad \nu X.\mu Y.\langle g \rangle((p \wedge X) \vee Y).$$

On Kripke models with accessibility relation g, it expresses that "on some g-path p occurs infinitely often" ($EGFp$ in CTL^* notation). This formula has alternation depth 2 and it has been shown that there is no μ-calculus formula δ' of alternation depth 1 or less such that δ is equivalent to δ' over Kripke models (see, e.g., [44]). Consequently, the same holds when considering equivalence over all game models. On the other hand, δ is equivalent to the translation of

$$[((g^*; g; p?)^d)^*]\top.$$

thus showing that it cannot be the case that all GL formulas are equivalent to μ-calculus formulas of alternation depth less than 2. ■

While the translation function does not yield μ-calculus formulas of bounded alternation depth, and while we know certain properties expressible with Game Logic require alternation depth at least 2, it is easy to see that for Game Logic formulas within the program fragment, alternation depth 1 is all we need.

▶ THEOREM 6.19. If φ lies in the program fragment of Game Logic, $ad(\varphi) = 1$. Consequently, Game Logic is more expressive than its program fragment.

6.5 Complexity of Model Checking

As may be expected, the presence of iteration makes model checking for Game Logic more complex than model checking for Coalition Logic. Instead of providing a model-checking algorithm for Game Logic directly, we shall consider an algorithm for the generalized μ-calculus. For the μ-calculus over Kripke models, the best known upper bound on complexity is NP $\cap$ co-NP, while for bounded alternation depth, the problem can be solved in deterministic polynomial time. As it turns out, the situation is similar for game models.

As for the representation of a game model and the definition of its size, the same considerations apply as for coalition models. Consequently, we can easily transfer the definition of section 3.3 to game models:

▶ DEFINITION 6.20 (MODEL SIZE). Given a game model $\mathcal{M} = ((S, \{E_g | g \in \Gamma_0\}), V)$, we define its size $|\mathcal{M}|$ as

$$|\mathcal{M}| = |S| + \sum_{\{s|s\in S\}} \sum_{\{g|g\in\Gamma_0\}} \sum_{\{X|sE_g^c X\}} |X|,$$

where E_g^c is the non-monotonic core of E_g.

Recall also that for Coalition Logic we defined the length of a formula φ as the number of its subformulas. For Game Logic, this approach will not work since we also have to account for the complexity of the games a formula contains. For

PDL, one gets around this problem by using the Fischer-Ladner closure $FL(\varphi)$ of the formula rather than its subformula closure. This Fischer-Ladner closure can easily be extended to include the duality operator, so that we could define $|\varphi| = |FL(\varphi)|$. Usually, it is sufficient however to think of $|\varphi|$ as the syntactic length of φ. We shall draw the reader's attention to the precise definition of $|\varphi|$ on the few occasions when this will be necessary.

Given a μ-calculus formula φ, a game model $\mathcal{M}$, and a variable valuation v, there is an algorithm for calculating $\{s|\mathcal{M}, v, s \models \varphi\}$ which runs in time $O(|\mathcal{M}|^{r+1} \times |\varphi|)$, where $r = d(\varphi)$, the depth of (non-vacuous) fixpoint nestings. To see this, suppose that $\varphi = \mu X.\psi$, where $d(\psi) < d(\varphi)$. Then we need to calculate $\bigcup_{0 \leq i \leq |S|} F^{|i}$ where S is the universe of $\mathcal{M}$ and $F(T) = \{t \in S|\mathcal{M}, v_{X:=T}, t \models \psi\}$. We thus need to calculate the extension of ψ under at most $|S|$ different valuations $v_{X:=T}$, so calculating the extension of φ requires time at most $|S|$ times the time it takes to determine the extension of ψ.

There is one respect, however, in which the previous bound can be improved dramatically, and this improvement will also turn out to be relevant conceptually when discussing differences between programs and games. Consider, e.g., the formula $\mu X.\mu Y.\varphi(X, Y)$, where φ contains no additional fixpoints. For sufficiently complex φ, the result obtained would suggest the model checking problem to require time roughly quadratic in the size of the state space of the model under consideration. As it turns out, however, the formula can be evaluated in linear time, due to lack of alternation and the generalized Knaster-Tarski fixpoint theorem.

Consider again how we can evaluate $\mu X.\mu Y.\varphi(X, Y)$. Given model $\mathcal{M} = ((S, \{E_g|g \in \Gamma_0\}), V)$, valuation v and sets $A, B \subseteq S$, let $\varphi(A, B) = \{s \in S|\mathcal{M}, s, v_{X:=A,Y:=B} \models \varphi\}$. Then the algorithm suggested in theorem A.1 proceeds by calculating

$$F_0 = \emptyset,\ F_1 = \mu Y.\varphi(F_0, Y),\ F_2 = \mu Y.\varphi(F_1, Y), \ldots$$

until for some $m \leq |S|$, $F_m = F_{m+1}$. To calculate F_i for $i > 0$, we need to calculate

$$f_0 = \emptyset,\ f_1 = \varphi(F_{i-1}, f_0),\ f_2 = \varphi(F_{i-1}, f_1), \ldots$$

until for some $n \leq |S|$, $f_n = f_{n+1}$. As a result, we require at most $|S| \cdot |S|$ calculations of $\varphi(X, Y)$.

Fortunately, we can do better than that. Suppose we have calculated $F_i = \mu Y.\varphi(F_{i-1}, Y)$. Since $F_{i-1} \subseteq F_i$ and φ is monotonic in both arguments,

$$F_i = \mu Y.\varphi(F_{i-1}, Y) \subseteq \mu Y.\varphi(F_i, Y) = F_{i+1} \text{ and } F_i = \varphi(F_{i-1}, F_i) \subseteq \varphi(F_i, F_i)$$

and hence by the generalized Knaster-Tarski theorem A.2,

$$F_{i+1} = \mu Y.\varphi(F_i, Y) = \varphi(F_i, F_i) \cup \varphi(F_i, \varphi(F_i, F_i)) \cup \cdots,$$

i.e., we can start the fixpoint approximation at F_i instead of $\emptyset$. As a consequence, the state space has to be traversed only once, i.e., we require at most $|S|$ calculations of $\varphi(X, Y)$.

The previous argument can be generalized to arbitrary finite sequences of nested μ-operators as long as no ν-operator intervenes, and an analogous argument applies to sequences of nested ν-operators without intervening μ-operators. As a result, we obtain a complexity bound formulated not in terms of fixpoint depth but in terms of alternation depth.

▶ THEOREM 6.21. Given a μ-calculus formula φ, a game model $\mathcal{M}$ and a variable valuation v, there is an algorithm for calculating $\{s|\mathcal{M}, v, s \models \varphi\}$ which runs in time $O(|\mathcal{M}|^{ad(\varphi)+1} \times |\varphi|)$.

PROOF. Consider a game model $\mathcal{M} = ((S, \{E_g | g \in \Gamma_0\}), V)$ and a valuation v in which we want to determine the extension of a μ-calculus formula φ. First, we rewrite φ to positive normal form. Next, we initialize all μ-variables to $\emptyset$ and all ν-variables to S. We have a boolean array *done* that keeps track of which subsentences have been evaluated already. Initially, $done[\psi] = false$ for all subsentences of φ. Lastly, the array $value[\psi]$ will store the set of states where ψ is true. The work is done by the following recursive function which initially is called with argument φ.

Evaluate(φ):

If φ is a sentence and $done[\varphi] = true$ then return $value[\varphi]$;
Case φ of the form

X (variable):	$R := v(X)$
p (atom):	$R := V(p)$
$\neg\psi$:	$R := S \setminus \text{Evaluate}(\psi)$
$\alpha \vee \beta$:	$R := \text{Evaluate}(\alpha) \cup \text{Evaluate}(\beta)$
$\alpha \wedge \beta$:	$R := \text{Evaluate}(\alpha) \cap \text{Evaluate}(\beta)$
$\langle g \rangle \psi$:	$R := \{s \in S \mid \exists X \subseteq S : sE^c_g X \text{ and } X \subseteq \text{Evaluate}(\psi)\}$
$[g]\psi$:	$R := \{s \in S \mid \forall X \subseteq S : sE^c_g X \Rightarrow X \cap \text{Evaluate}(\psi) = \emptyset\}$
$\mu X.\psi$:	If the surrounding fixpoint formula is a greatest fixpoint, then $v(X) := \emptyset$; For each open ψ-subformula $\mu Y.\delta$ such that there is no ψ-subformula $\nu Z.\chi$ containing it, set $v(Y) := \emptyset$. Repeat: $R := v(X)$; $v(X) := \text{Evaluate}(\psi)$ Until $R = v(X)$.
$\nu X.\psi$:	If the surrounding fixpoint formula is a least fixpoint, then $v(X) := S$; For each open ψ-subformula $\nu Y.\delta$ such that there is no ψ-subformula $\mu Z.\chi$ containing it, set $v(Y) := S$. Repeat: $R := v(X)$; $v(X) := \text{Evaluate}(\psi)$ Until $R = v(X)$.

If φ is a sentence then

$done[\varphi] := true;$
$value[\varphi] := R;$
Return R

Note that we assume that the test in the conditional for the μ-formulas succeeds only if there is a surrounding fixpoint formula. To verify that the running time of the algorithm is indeed in $O(|\mathcal{M}|^{ad(\varphi)+1} \times |\varphi|)$, one shows by induction on φ that Evaluate(φ) runs in time $O(|\varphi| \times |\mathcal{M}| \times |S|^{ad(\varphi)})$.

If φ is a variable X or some atomic proposition p, the extension of φ is already part of the description of the model/valuation and the function call terminates immediately. Skipping the boolean cases, suppose that $\varphi = \langle g \rangle \psi$. After determining Evaluate(ψ) in $O(|\psi| \times |\mathcal{M}| \times |S|^{ad(\psi)})$, we check for every $sE_g^c X$ whether $X \subseteq$ Evaluate(ψ) which can be done in $O(|\mathcal{M}|)$. Hence, we end up with a time bound of $O(|\varphi| \times |\mathcal{M}| \times |S|^{ad(\psi)})$.

In case $\varphi = \mu X.\psi$ consider first the case where $ad(\varphi) = ad(\psi)+1$. In the worst case, the surrounding fixpoint formula is a greatest fixpoint, so that X is reset to $\emptyset$ before we calculate $\bigcup_{0 \leq i \leq |S|} F^{\uparrow i}$ where $F(T) = \{t \in S | \mathcal{M}, v_{X:=T}, t \models \psi\}$. As a result, we calculate Evaluate(ψ) at most $|S|$ times for different valuations, each calculation requiring $O(|\psi| \times |\mathcal{M}| \times |S|^{ad(\psi)})$ time, yielding a total of $O(|\psi| \times |\mathcal{M}| \times |S|^{ad(\varphi)})$.

For the case where $ad(\varphi) = ad(\psi)$, the approximations of μX will be absorbed by Evaluate(ψ): If (1) there is another least-fixpoint subformula $\mu Y.\delta$ of the same alternation depth, the algorithm will refrain from resetting Y to $\emptyset$ for new assignments to X. If (2) there is some greatest-fixpoint subformula $\nu Y.\delta$ of the same alternation depth, then X cannot occur in δ, and so $\nu Y.\delta$ only needs to be evaluated once. ∎

Note that the time bound provided is a rather rough estimate: As the proof shows, the real calculation time is more accurately described by $O(|S|^{ad(\varphi)} \times |\mathcal{M}| \times |\varphi|)$ where S is the universe of $\mathcal{M}$. Since $|S|$ is usually much smaller than $|\mathcal{M}|$, model checking is often somewhat more feasible than suggested by the bound given in the theorem. Since we are, however, not interested in producing an efficient implementation but only in obtaining a rough idea of the complexity of the model checking problem, the bound stated in the theorem will do.

While theorem 6.21 provides a polynomial time model checking procedure for bounded alternation depth, for unbounded alternation depth, we conjecture that, similar to the case of Kripke models, the model-checking problem is in NP $\cap$ co-NP.

What are the consequences of theorem 6.21 for Game Logic? Using the translation from Game Logic into the μ-calculus, we can get an upper bound for the complexity of Game Logic model checking as well.

► Corollary 6.22. Given a Game Logic formula φ and a finite game model $\mathcal{M}$, model checking can be done in time $O(|\mathcal{M}|^{ad(\varphi)+1} \times |\varphi|)$. Consequently, if φ lies in the program fragment, model-checking can be done in time $O(|\mathcal{M}|^2 \times |\varphi|)$.

PROOF. Since by theorem 6.17 $ad(\varphi) = ad(\varphi^\circ)$, we only need to check that the translation from Game Logic into the μ-calculus is efficient, for then the result follows by theorems 6.21 and 6.19.

Inspecting the definition of $^\circ$, the only problematic clause is $(\alpha \cup \beta)^x$, for $\langle g_1 \cup g_2 \rangle \varphi$ will be translated as $\langle g_1 \rangle \varphi^\circ \vee \langle g_2 \rangle \varphi^\circ$, duplicating φ° and hence resulting in an exponential increase in length. As pointed out in [46], however, this does not really create a problem, since a clever representation of subformulas can be chosen which consolidates common subformulas. ■

6.6 Complexity of the Satisfiability Problem

The appendix of [97] contains an argument which demonstrates that the satisfiability problem for Game Logic is decidable. The argument uses a translation of Game Logic formulas into modal μ-calculus formulas, reducing Game Logic satisfiability to μ-calculus satisfiability. The translation used, however, is not the one presented in section 6.4.2, for the aim in this case is a translation into the μ-calculus interpreted over Kripke models rather than game models. Hence, since Game Logic and the standard modal μ-calculus are interpreted over different models, the models have to be translated as well. As for the formula translation, however, it turns out that the length of a formula can grow exponentially in the translation process, thereby seemingly making the reduction inefficient. A closer look, however, reveals that this exponential blow-up can easily be circumvented, as in the case of corollary 6.22. We now present the argument which establishes an exponential-time upper bound for the satisfiability problem.

Formulas as well as models of Game Logic can be translated into formulas and models of the μ-calculus as follows: Given a game model $\mathcal{M} = ((S, \{E_g | g \in \Gamma_0\}), V)$, we construct its Kripke-version by introducing new states for every subset of S, i.e., we let $\mathcal{M}_K = ((S', R_e, \{R_g | g \in \Gamma_0\}), V')$ where $S' = S \cup \{s_X | X \subseteq S\}$, $s_x R_e t$ iff $t \in X$ and $t R_g s_X$ iff $t E_g X$. In other words, the new set includes all the old states (individual-states) plus all subsets of old states (set-states), and we have two sorts of accessibility relations. Relation R_g relates individual-states to set-states just as E_g did in the game model, and R_e is nothing but the converse of the element-of relation, relating set-states to individual-states. Finally, we also introduce another propositional atom p_e which holds at a state iff it is an individual-state, so we let $V'(p) = V(p)$ for $p \neq p_e$, and $V'(p_e) = S$. We can then translate Game Logic formulas over atomic games Γ_0 into μ-calculus formulas over $\Gamma_0 \cup \{e\}$ (where $e \notin \Gamma_0$) as follows:

$$
\begin{array}{llll}
g^x & = \langle g\rangle[e](p_e \wedge X) & (\varphi?)^x & = \varphi^\circ \wedge p_e \wedge X \\
(\alpha \cup \beta)^x & = \alpha^x \vee \beta^x & (\alpha;\beta)^x & = \alpha^x[X := \beta^x] \\
(\alpha^d)^x & = p_e \wedge \neg\alpha^x[X := p_e \wedge \neg X] & (\alpha^*)^x & = \mu Y.p_e \wedge (X \vee \alpha^x[X := Y])
\end{array}
$$

$$
\begin{array}{ll}
p^\circ & = p \\
(\neg\varphi)^\circ & = p_e \wedge \neg\varphi^\circ \\
(\varphi \vee \psi)^\circ & = \varphi^\circ \vee \psi^\circ \\
(\langle\alpha\rangle\varphi)^\circ & = \alpha^x[X := \varphi^\circ]
\end{array}
$$

Note that this translation function is different from the one previously used in section 6.4 to translate from Game Logic into the generalized μ-calculus, the reason being that now we need a translation which links Kripke models to game models.

▶ LEMMA 6.23. For all Game Logic formulas φ, φ is satisfiable in a game model iff $\varphi^\circ \wedge p_e$ is satisfiable in a Kripke model.

PROOF. It is easy to check that if φ is satisfiable in $\mathcal{M}$, $\varphi^\circ \wedge p_e$ is satisfiable in $\mathcal{M}_K$ as constructed before.

For the converse, assume $\mathcal{M}_K = ((S, \{R_e, R_g|g \in \Gamma_0\}), V)$ satisfies $\varphi^\circ \wedge p_e$ at state $s \in S$. Let $\mathcal{M} = ((S', \{E_g|g \in \Gamma_0\}), V')$ be such that $S' = V(p_e)$, $V'(p) = V(p)|S'$ for all atoms $p \neq p_e$, and finally

$$sE_gU \text{ iff } \exists t \in S : sR_gt \text{ and } \forall u \in S : tR_eu \Rightarrow u \in U$$

where $s \in S'$ and $U \subseteq S'$. We claim that for all Game Logic formulas χ and games γ, (1) For all $s \in S'$: $\mathcal{M}_K, s \models \chi^\circ$ iff $\mathcal{M}, s \models \chi$, and (2) For all $s \in S', T \subseteq S$: $\mathcal{M}_K, s, T \models \gamma^x$ iff $sE_\gamma(T \cap S')$. As should be evident, the notation $\mathcal{M}_K, s, T \models \gamma^x$ refers to γ^x being true at s in $\mathcal{M}_K$ when the free variable X has denotation T. The proof is by simultaneous induction on φ and γ, and the different inductive steps involve no major difficulties. We shall only show the case of iteration:

To show that $\mathcal{M}_K, s, T \models (\gamma^*)^x$ iff $sE_{\gamma^*}(T \cap S')$ for all $s \in S', T \subseteq S$, it suffices to show that for all $T \subseteq S$,

$$\mu U.S' \cap (T \cup \{s \in S|\mathcal{M}_K, s, U \models \gamma^x\}) \quad = \quad \mu U.(T \cap S') \cup E_\gamma(U).$$

That these two fixpoints coincide can most easily be seen if we consider the approximation stages for calculating them: If we let $F(U) = S' \cap (T \cup \{s \in S|\mathcal{M}_K, s, U \models \gamma^x\})$ and $G(U) = (T \cap S') \cup E_\gamma(U)$, it is easily seen that for every ordinal κ we have $F^{\uparrow\kappa} = G^{\uparrow\kappa} \subseteq S'$. For the inductive step for $\kappa + 1$, one must show that

$$S' \cap (T \cup \{s \in S|\mathcal{M}_K, s, F^{\uparrow\kappa} \models \gamma^x\}) = (T \cap S') \cup E_\gamma(G^{\uparrow\kappa}) \subseteq S'$$

which follows by the induction hypotheses for κ and γ. ∎

▶ THEOREM 6.24. The satisfiability problem for Game Logic is in EXPTIME.

PROOF. By lemma 6.23, we have reduced the satisfiability problem for Game Logic to the satisfiability problem for the modal μ-calculus over Kripke models. As a consequence of [45, 117], the satisfiability problem for the μ-calculus is EXPTIME-complete, so all we need to check is that the translation from φ to $\varphi^\circ \wedge p_e$ is effective.

As in the case of theorem 6.22, one can see that the problematic case is program/game union which causes the translation not to be efficient. Again, this problem can be avoided if common subformulas are represented only once. Inspecting the proof of the complexity result for the modal μ-calculus in [117] reveals that what is important is not the syntactic length of a μ-calculus formula φ but rather the size of its Fischer-Ladner closure $FL(\varphi)$ (an extension of the subformula closure). Since $|FL(\varphi^\circ)|$ is $O(|\varphi|)$, the translation function $^\circ$ does indeed provide an efficient reduction. ■

It is reasonable to conjecture that the satisfiability problem is EXPTIME-complete, though we have no proof of this conjecture yet.

6.7 Discussion

6.7.1 Simulating Game Models by Kripke Models

The proof of theorem 6.24 concerning the complexity of the satisfiability problem relied on translating formulas of Game Logic into the μ-calculus and simulating game models by Kripke models. In fact, this technique of treating game models as Kripke models also could have been used to analyze the complexity of model checking. More generally, the simulation technique is not only applicable to the specific case of Game Logic, but to non-normal modal logics more generally. As shown in [51], a modal formula φ can be translated into a multi-modal formula φ° such that φ is satisfiable in neighborhood models iff φ° is satisfiable in Kripke models. Similar translations are given which link, for example, satisfiability in monotonic neighborhood models to satisfiability in Kripke models. Consequently, theorem proving in non-normal modal logics can be reduced to theorem proving in normal modal logics. This line of investigation is carried further in [78] where it is shown that one can obtain translations into uni-modal formulas which preserve satisfiability and also a number of other properties.

In spite of this reduction to normal modal logic, we think that in general, working with Game Logic or Coalition Logic is easier than working with their normal modal simulations. While we have seen that simulating neighborhood models (game models, coalition models) by Kripke models can be useful to obtain certain results, there is no free lunch here. These simulations do not yield immediate results, for example, concerning the completeness and complexity of the coalition logics studied in chapter 3 (with the exception of Mon). The reason is that the neighborhood relations E_C are not simply required to be monotonic, they have to be interrelated in a certain way. The normal modal logic simulating, for example, Play will be complex enough to prevent any quick conclusions about its complexity. Even for Game Logic we needed a significant extension of the results in [51, 78] to deal with the program/game operations.

6.7.2 Programs vs. Games

In order to compare game operations to program operations, this chapter has focused on game models which are extremely simple compared to the structures we have investigated for Coalition Logic. However, even for these simple structures which only describe determined 2-player games, we have seen some interesting differences between Game Logic and its program fragment, summarized in figure 6.2.

	Game Logic GL	program fragment $\mathtt{GL}^{-d}$
complete axiomatization	not yet	yes
maximal alternation depth	none	1
expressive power	$>$	
complexity model checking	$O(\lvert\mathcal{M}\rvert^{ad(\varphi)+1} \times \lvert\varphi\rvert)$	$O(\lvert\mathcal{M}\rvert^2 \times \lvert\varphi\rvert)$
complexity satisfiability	in EXPTIME	in EXPTIME

Figure 6.2: Differences between Game Logic and its program fragment over general game models.

In terms of expressive power, we have seen that the program fragment is less expressive than full Game Logic. By theorems 6.18 and 6.19, there is a Game Logic formula which is not equivalent to any formula of the program fragment. In fact, there is even a formula of alternation depth 1 which cannot be expressed within the program fragment (see theorem 7.2 of the following chapter). Put differently, while the program fragment can be translated into L_{μ_1}, there are also formulas in L_{μ_1} which are not equivalent to any formula of the program fragment.

Syntactically, we have seen that Game Logic formulas can have arbitrary alternation depth whereas programs only have alternation depth 1. As shown in [18], this alternation hierarchy for Game Logic does not collapse semantically, i.e., there are Game Logic formulas of arbitrary alternation depth which are not equivalent to formulas of lower alternation depth.

Due to the difference in alternation depth, we have seen that model checking for programs seems easier than model checking for games. Naturally, this difference is rather tentative, for it might be that better algorithms will be found for the modal μ-calculus which run in polynomial time independent of the alternation depth. At the current state of knowledge, however, the maximal number of subsequent role-changes which are linked to iteration determines the complexity of model checking. If it is indeed the case that also model checking for the generalized μ-calculus is in NP $\cap$ co-NP, the question whether model checking for programs is simpler than for games is linked to one of the basic open issues of complexity theory, namely, the relationship between NP $\cap$ co-NP and P.

The existence of a complete axiomatization constitutes a final difference between Game Logic and its program fragment. Again, this difference is only a rather weak difference due to our insufficient knowledge. Still, it also should not be dismissed too easily. Take, e.g., the modal μ-calculus (interpreted over Kripke models): The decade it took from the creation of the modal μ-calculus and its proposed axiomatization in [76] to a completeness proof in [124] bears witness to the fact that the conceptual complexity of this logic far exceeds that of Propositional Dynamic Logic, even though, e.g., the complexity of the satisfiability problem is EXPTIME-complete in both cases.

6.8 Bibliographic Notes

Game Logic was first introduced in [96, 97], a further introductory reference is [101]. The completeness result for Game Logic without dual is from [97], as is the decidability of full Game Logic. Our argument in section 6.6 essentially follows [97], except that we used a different translation function which eliminates the need for an additional conjunct which forces the Kripke model to be of a particular shape. As a result, the proof of lemma 6.23 is simplified.

Concurrent Propositional Dynamic Logic (CPDL, [108]) is a system closely related to Game Logic. Where Game Logic talks about games, CPDL talks about concurrent programs. Due to this different interpretation, CPDL is not exactly a sublogic of Game Logic since the E_γ functions are not assumed to be monotonic. CPDL contains two disjunctions, corresponding to demonic and angelic choice, but no dual operator. Axiomatic completeness is established in [54].

The standard modal μ-calculus interpreted over Kripke models was originally proposed in [76]. The notion of alternation depth as well as the model checking algorithm which we extended from Kripke models to game models in section 6.5 go back to [46]. The axiomatization proposed in [76] was proved complete in [124]. See also [6] for a recent book on the μ-calculus.

The generalized μ-calculus as defined in section 6.4 is closely related to the Alternating μ-calculus (AMC) proposed in [3] which is an extension of Alternating Temporal Logic (ATL) discussed in section 4.4. Like ATL, AMC is interpreted over alternating transition systems which are essentially a subclass of coalition models. The model-checking complexity for AMC is also dependent on the alternation depth of the formula, i.e., there is an analogue of theorem 6.21 for AMC.

Chapter 7

Game Logic over Kripke Models

Having considered Game Logic over general game models in the previous chapter, this chapter focuses on Game Logic over Kripke models. In order to illuminate further the difference between programs and games, considering Kripke models allows one to investigate the difference between program operations and game operations when applied to programs as the basic building blocks. In other words, if we start with 1-player games at the atomic level, how do complex structured programs compare to complex structured games? As before, the comparison will focus on a number of technical questions.

Having recalled the semantics of Game Logic over Kripke models, we discuss how the expressive power of Game Logic compares to PDL on the one hand and to the modal μ-calculus on the other hand. Turning then toward axiomatization, we can observe that even over Kripke models the induction axiom is not valid for games whereas it is valid for programs. As for general Game Logic, we provide an axiomatization which we conjecture to be complete. Regarding the complexity of the satisfiability problem, we are able to give a precise characterization due to results for Propositional Dynamic Logic and the modal μ-calculus.

Sections 7.4 and 7.6 take a closer look at the operations of Game Logic. While iteration seems to be a conceptually unambiguous notion for programs, it turns out that for games at least two different candidate interpretations suggest themselves. In section 7.6, we consider the question of whether there might be operations other than the ones provided by Game Logic by which to construct new games. Using the notion of bisimulation-safety, we are able to partially answer this question in the negative.

7.1 Semantics: GL, PDL, and the μ-Calculus

When Game Logic is interpreted over Kripke models, the idea is that the basic atomic games are assumed to be 1-player games, i.e., programs. All choices within an atomic game are made by the first player, and all interaction is introduced only by the game constructions, more specifically by the dual operator. Thus, even over Kripke models, the games which can be constructed by the operations of Game Logic allow one to construct "real" games, but all interaction can be located at the non-atomic and hence syntactic level.

Formally, the previous chapter was concerned with arbitrary game models $\mathcal{M} = ((S, \{E_g | g \in \Gamma_0\}), V)$ which by corollary 2.13 we can think of as a collection of states which are linked to determined 2-player games. In the present chapter, we restrict ourselves to game models where for every $g \in \Gamma_0$, E_g is disjunctive. As discussed in section 2.4.3, corollary 2.16 allows us to think of $\mathcal{M}$ as a collection of states linked to a 1-player game. As a consequence, we mentioned that $\mathcal{M}$ α-corresponds to a Kripke model $\mathcal{M}_K = ((S, \{R_g | g \in \Gamma_0\}), V)$ such that sE_gX holds iff there is some $t \in X$ such that sR_gt, equivalence (2.2).

Given a Kripke model $\mathcal{M}_K = ((S, \{R_g | g \in \Gamma_0\}), V)$, we can thus restate the semantics of Game Logic as follows: We define truth of a formula at a state $\mathcal{M}_K, s \models \varphi$ as before, also leaving the inductive definition of E_γ unchanged. The only difference lies in the definition of E_g for atomic games $g \in \Gamma_0$. For game models, E_g was provided by the model, whereas for Kripke models, we naturally define

$$E_g(X) = \{s \in S | \exists t \in X : sR_gt\}. \tag{7.1}$$

Note that this gives us a semantic definition of $\langle g \rangle \varphi$ standard in modal logic: $\mathcal{M}_K, s \models \langle g \rangle \varphi$ holds iff there is some $t \in S$ such that sR_gt and $\mathcal{M}_K, t \models \varphi$. Such a definition, however, is only possible for atomic games since non-atomic games will generally not be 1-player games anymore, unless we restrict ourselves to dual-free games. An easy induction on γ shows that disjunctivity is preserved by the program operations.

▶ THEOREM 7.1. For all dual-free games γ, E_γ is disjunctive.

As a consequence, by corollary 2.16, all dual-free non-atomic games are essentially 1-player games as well, i.e., for every dual-free game γ there is an accessibility relation R_γ such that $E_\gamma(X) = \{s \in S | \exists t \in X : sR_\gamma t\}$. In fact, R_γ can be constructed inductively as well:

$$\begin{array}{rl}
sR_{\alpha;\beta}t & \text{iff } \exists u : sR_\alpha u \text{ and } uR_\beta t \\
sR_{\alpha\cup\beta}t & \text{iff } sR_\alpha t \text{ or } sR_\beta t \\
sR_{\varphi?}t & \text{iff } s = t \text{ and } \mathcal{M}_K, s \models \varphi \\
sR_{\alpha^*}t & \text{iff } \exists n \geq 0 : \exists s_0 \dots s_n \forall i : s_i R_\alpha s_{i+1} \text{ and } s = s_0 \text{ and } t = s_n
\end{array}$$

One can show that using these definitions and defining

$$\mathcal{M}_K, s \models \langle \gamma \rangle \varphi \text{ iff } \exists t \in S : sR_\gamma t \text{ and } \mathcal{M}_K, t \models \varphi, \tag{7.2}$$

equation (7.1) holds for all dual-free games γ.

Dual-free Game Logic over Kripke models is nothing but Propositional Dynamic Logic (PDL). PDL is usually defined by way of accessibility relations R_γ rather than effectivity functions E_γ, but we shall switch back and forth between these two perspectives.

As discussed in the previous chapter, GL forms a fragment of the generalized μ-calculus. Over Kripke models, we are back on well-known terrain since the generalized μ-calculus over Kripke models is just the standard modal μ-calculus. Hence, using the translation function of theorem 6.15, Game Logic over Kripke models can be embedded in the modal μ-calculus, and one can again ask how expressive the GL-fragment of the μ-calculus actually is.

7.2 Expressiveness

As mentioned, Game Logic is very closely related to PDL on the one hand and to the μ-calculus on the other hand. Since all three logics can be interpreted over Kripke models, we can compare them in expressive power.

7.2.1 GL vs. PDL

As a consequence of theorems 6.18 and 6.19 of the previous chapter, GL is more expressive than PDL, as the following theorem states. The proof given here, however, makes use of a formula of alternation depth 1 which cannot be expressed by PDL.

► THEOREM 7.2. GL (over Kripke models) is more expressive than PDL. Without iteration, they are equally expressive.

PROOF. It has been shown in [116, 76] that the μ-calculus formula $\mu X.[g]X$ which expresses the absence of an infinite g-branch is not equivalent to any PDL formula. Since the formula is equivalent to the GL-formula $\langle (g^d)^* \rangle \bot$, GL is more expressive than PDL.

Without iteration, every GL-formula is equivalent to a purely modal formula (i.e., a formula where all modalities are indexed by atomic games/programs only), using the axioms of figure 6.1 as rewrite rules from left to right. ■

Note that expressiveness here is measured in terms of the *propositions* which a given language can express. Taking into account also the *games* which a language can express, it is clear that GL is more expressive than PDL even without iteration: the game g^d cannot be expressed without dual, simply because all dual-free games are disjunctive whereas one can easily construct a model where g^d is not. Hence, there cannot be any dual-free game γ such that $E_\gamma = E_{g^d}$ for all models.

As it turns out, even if the formula used in the proof of theorem 7.2 is added to PDL, GL remains more expressive. Consider RPDL, PDL with an added predicate *repeat*(γ) which holds at states where γ can be executed infinitely often. In other words, *repeat*(γ) expresses that γ is not conversely well-founded. Formally, given a Kripke model $\mathcal{M}_K = ((S, \{R_g | g \in \Gamma_0\}), V)$, we define

$$\mathcal{M}_K, s_0 \models repeat(\gamma) \text{ iff } \exists s_1, s_2, \ldots \forall i \geq 0 : s_i R_\gamma s_{i+1}.$$

In μ-calculus terms, *repeat*(γ) is equivalent to $\neg\mu X.[\gamma]X$, since $\mu X.[\gamma]X$ expresses the well-foundedness of γ. The previous result has thus shown that RPDL is more expressive than PDL. GL on the other hand is even more expressive than RPDL.

▶ THEOREM 7.3. GL (over Kripke models) is more expressive than RPDL.

PROOF. To see that GL is at least as expressive as RPDL, note that *repeat*(γ) is equivalent to $[(\gamma^d)^*]\top$. For a GL formula which cannot be expressed in RPDL, Niwinski showed in [92] that $\nu X.\langle a\rangle X \wedge \langle b\rangle X = [(a^d \cup b^d)^*]\top$ is not expressible in *RPDL*. ■

7.2.2 GL vs. μ-Calculus

As shown in theorem 6.15 of the last chapter, GL can be translated into the 2-variable fragment of the μ-calculus, and hence we know that GL is a fragment of L^2_μ also over Kripke models. While we conjecture that GL is a proper fragment of L_μ, this question is open so far. Two ways to answer this question suggest themselves.

First, one may try to prove that the alternation hierarchy for either L^2_μ or GL collapses. More precisely: Bradfield [25] and Lenzi [81] have shown that the alternation hierarchy of the μ-calculus is strict, i.e., for any natural number k there are μ-calculus formulas of alternation depth k which are not equivalent to formulas of smaller alternation depth. If one could show that either for L^2_μ or for GL there is some k such that all formulas are equivalent to a formula of alternation depth less than k, Game Logic (both over general game models and over Kripke models) would be less expressive than the μ-calculus. Furthermore, it would also mean that model-checking for Game Logic is simpler than for the full μ-calculus, given that the efficiency of model-checking seems to depend on the alternation depth (as we saw in the previous chapter). Work of [18], however, suggests that the alternation hierarchy does *not* collapse, neither for L^2_μ nor for GL, so this approach does not seem to succeed.

A second approach could focus on the finite variable fragments of L_μ. What is needed is a μ-calculus formula φ which makes use of 3 variables and which is not equivalent to any μ-calculus formula with less than three variables. Examples could be complex fairness properties such as the following one from [26]

$$\nu X.\mu Y.\nu Z.[a]X \wedge (\langle a\rangle\top \rightarrow [b]Y) \wedge [b]Z$$

which expresses that there is no path on which action a is enabled infinitely often but occurs only finitely often (we assume here that the only two available actions are a and b). In general, it would seem that via pebble games for the μ-calculus one could prove that a formula like the one given cannot be expressed with fewer variables, but it seems that finite variable fragments of the μ-calculus have not been investigated yet.

7.3 Axiomatization and Induction

7.3.1 Axiomatization

Like for general Game Logic, duality together with iteration presents a problem for axiomatization. Hence we will mainly focus our attention on $\mathsf{GL_K}^{-*}$ and $\mathsf{GL_K}^{-d}$.

▶ DEFINITION 7.4 (KRIPKE GAME LOGIC AXIOMATICS). A *Kripke game logic* is any game logic which contains the additional axioms of figure 7.1. Let $\mathsf{GL_K}$ denote the smallest Kripke game logic.

$$\boxed{\langle g\rangle(\varphi \vee \psi) \to \langle g\rangle\varphi \vee \langle g\rangle\psi \qquad\qquad \neg\langle g\rangle\bot}$$

Figure 7.1: The additional axiom schemas of $\mathsf{GL_K}$, where $g \in \Gamma_0$ is atomic.

▶ THEOREM 7.5. $\mathsf{GL_K}$ is sound with respect to the class of all Kripke models.

One can show by induction that $\langle\gamma\rangle(\varphi \vee \psi) \to \langle\gamma\rangle\varphi \vee \langle\gamma\rangle\psi$ and $\neg\langle\gamma\rangle\bot$ hold for all dual-free games γ. As a consequence, $\mathsf{GL_K}^{-d} = \mathsf{PDL}$, the standard Segerberg axiomatization of PDL. The main difference between the axiomatic systems is that where $\mathsf{GL_K}^{-d}$ uses the Fixpoint Rule, PDL uses the induction axiom (to be discussed later). The interderivability of these principles is shown, e.g., in [77]. Hence, the completeness result for $\mathsf{GL_K}^{-d}$ follows from the completeness result for PDL.

▶ THEOREM 7.6. $\mathsf{GL_K}^{-d}$ is complete with respect to the class of all Kripke models.

As an analogue to the previous chapter, we shall also show that $\mathsf{GL_K}^{-*}$ is complete w.r.t. the class of all Kripke models. As in the previous chapter, the proof is via a canonical model construction, the difference being that now the canonical model we construct is a Kripke model.

Let Λ be any Kripke game logic and let S be the set of all maximally Λ-consistent sets of formulas. Define the canonical Λ-model $\mathcal{C} = ((S, \{R_g | g \in \Gamma_0\}), V)$ as follows:

$$\begin{array}{rl} s \in V(p) & \text{iff } p \in s \\ sR_g t & \text{iff } \{\langle g\rangle\varphi | \varphi \in t\} \subseteq s \end{array}$$

Before proceeding to prove completeness, we need an auxiliary existence lemma which was already appealed to in section 3.4.3 when proving the completeness of Coalition Logic over individualistic coalition models.

► LEMMA 7.7. For any maximally Λ-consistent set s and any formula $\langle g\rangle\varphi \in s$, there is some maximally Λ-consistent set t such that $\varphi \in t$ and for all $\psi \in t$, $\langle g\rangle\psi \in s$.

PROOF. Let $t_0 = \{\varphi\} \cup \{\delta | [g]\delta \in s\}$. Assume by reductio that t_0 is inconsistent, i.e., $\vdash_\Lambda \bigwedge\Delta \to \bot$ where w.l.o.g. $\Delta = \{\varphi, \delta_1, \ldots, \delta_n\}$. This would mean that $\langle g\rangle\varphi, [g]\delta_1, \ldots, [g]\delta_n \in s$ and hence using the distribution axiom of figure 7.1 $\langle g\rangle(\varphi \wedge \delta_1 \wedge \ldots \wedge \delta_n) \in s$. Since $\bigwedge\Delta$ implies $\bot$ this means that $\langle g\rangle\bot \in s$ which contradicts the other axiom of figure 7.1. This shows that t_0 is indeed consistent.

Consequently, t_0 can be extended to a maximally Λ-consistent set $t \supseteq t_0$ which satisfies the condition: take any $\delta \in t$ and assume by reductio that $\langle g\rangle\delta \notin s$. Then $[g]\neg\delta \in s$ and hence $\neg\delta \in t_0 \subseteq t$, contradicting the consistency of t. ■

► THEOREM 7.8. $\texttt{GL}_{\texttt{K}}{}^{-*}$ is complete with respect to the class of all Kripke models.

PROOF. As in the proof of theorem 6.10 for Game Logic over general game models, we first establish the following truth lemma: For any maximally Λ-consistent set $s \in S$ and any formula φ: $\mathcal{C}, s \models \varphi$ iff $\varphi \in s$. The proof is the same as before, except that the atomic case showing that $sE_g\varphi^{\mathcal{C}}$ holds iff $\langle g\rangle\varphi \in s$ does not immediately follow from the definition of the canonical model. Still, it can be proved easily using the previous existence lemma.

Using this truth lemma, the canonical model theorem can be established: Every Kripke game logic Λ is complete with respect to its canonical model $\mathcal{C}$. Completeness w.r.t. the class of all Kripke models then follows as a corollary. ■

7.3.2 The Induction Axiom

As mentioned, in the standard axiomatization of PDL the following induction axiom `IndAx`

$$\langle\alpha^*\rangle\varphi \to (\varphi \vee \langle\alpha^*\rangle(\neg\varphi \wedge \langle\alpha\rangle\varphi))$$

replaces the Fixpoint Rule. The induction axiom formalizes what might be called a *sudden miracle principle*. If φ does not hold at present but Angel has a strategy for achieving it after playing α^*, then she must be able to do this by means of a sudden miracle: She must have a strategy for playing α^* which lets her achieve a state where φ is false but from where she can achieve φ through one play of α. The induction axiom is usually better known in its dualized version

$$(\varphi \wedge [\alpha^*](\varphi \to [\alpha]\varphi)) \to [\alpha^*]\varphi.$$

It is an easy exercise to check that the induction axiom is a sound principle provided dual is not present. In a world without demons, all miracles are sudden.

▶ THEOREM 7.9. $\mathtt{GL_K}^{-d} \vdash \mathtt{IndAx}$.

PROOF. It is shown in [77] that the Fixpoint Rule and the induction axiom are interderivable in PDL. Below we shall give a semantic argument showing that the induction axiom is valid in dual-free Game Logic over Kripke models. By theorem 7.6, this argument is sufficient.

Semantically, we need to show that for every dual-free game α,

$$E_{\alpha^*}(Z) \subseteq Z \cup E_{\alpha^*}(\overline{Z} \cap E_\alpha(Z)).$$

Without dual, theorem 7.1 guarantees all effectivity functions to be disjunctive. By theorem A.3, this means that fixpoints can be finitely approximated and hence

$$E_{\alpha^*}(Z) = \mu X.Z \cup E_\alpha(X) = \bigcup_{i<\omega} F^{\uparrow i}$$

where $F(X) = Z \cup E_\alpha(X)$. Thus it suffices to show by induction that for all i we have $F^{\uparrow i} \subseteq Z \cup G^{\uparrow i}$, where $G(X) = (\overline{Z} \cap E_\alpha(Z)) \cup E_\alpha(X)$. The inductive step consists of proving the following inclusion

$$Z \cup E_\alpha(F^{\uparrow i}) \subseteq Z \cup (\overline{Z} \cap E_\alpha(Z)) \cup E_\alpha(G^{\uparrow i})$$

which follows from the induction hypothesis, monotonicity and disjunctivity of E_α. ∎

Axiomatically, finite atomic disjunctivity is guaranteed by the two axioms of figure 7.1. Once dual is present, however, disjunctivity and consequently also the induction axiom can fail to hold for non-atomic games.

▶ THEOREM 7.10. $\mathtt{GL_K} \nvdash \mathtt{IndAx}$.

PROOF. The Kripke model of figure 7.2 falsifies the induction axiom for the game $(a \cap b)^*$.

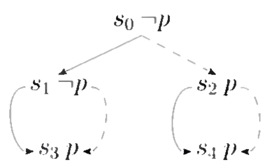


Figure 7.2: A Kripke model with two accessibility relations R_a (solid arrows) and R_b (dashed arrows) which falsifies the induction axiom.

The Kripke model $\mathcal{M}_K$ consists of 5 states where $\mathcal{M}_K, s_0 \models \langle (a \cap b)^* \rangle p$, for Angel has a strategy for achieving p after playing two rounds of $(a \cap b)$. On the other hand, $\mathcal{M}_K, s_0 \not\models p$ and similarly $\mathcal{M}_K, s_0 \not\models \langle(a \cap b)^*\rangle(\neg p \wedge \langle a \cap b\rangle p)$ because at state s_2 the formula p holds. Formally,

$$\begin{aligned} E_{(a\cap b)^*}(\{s_2, s_3, s_4\}) &= \{s_2, s_3, s_4\} \cup E_{a\cap b}(\{s_2, s_3, s_4\}) \cup \ldots \\ &= \{s_1, s_2, s_3, s_4\} \cup E_{a\cap b}(\{s_1, s_2, s_3, s_4\}) \cup \ldots \\ &= \{s_0, s_1, s_2, s_3, s_4\} \cup E_{a\cap b}(\{s_0, s_1, s_2, s_3, s_4\}) \cup \ldots \\ &= \{s_0, s_1, s_2, s_3, s_4\} \end{aligned}$$

whereas $(\neg p \wedge \langle a \cap b\rangle p)^{\mathcal{M}_K} = \{s_1\}$ and $E_{(a\cap b)^*}(\{s_1\}) = \{s_1\} \cup E_{a\cap b}(\{s_1\}) \cup \ldots = \{s_1\}$. ■

7.4 Varieties of Iteration

When explaining the semantics of iteration in the previous chapter, we said that for γ^*, Angel is allowed to choose how often to play game γ, with the possibility not to play γ at all. More precisely, game γ is played repeatedly, and after each play of γ, Angel can decide whether or not to continue. Alternatively, we can think of a second kind of iteration where Angel immediately has to decide on how often she wants to play γ. The second kind of iteration is formally defined as follows:

▶ DEFINITION 7.11 (ALTERNATIVE ITERATION). The alternative iteration of a game γ is denoted as $\gamma^{\circledast}$. Given game model $\mathcal{M} = ((S, \{E_g | g \in \Gamma_0\}), V)$, define $E_{\gamma^{\circledast}}(X) = \bigcup_{i<\omega} E^i_\gamma(X)$.

From the informal semantics we have given it should be clear that a winning strategy for Angel in $\gamma^{\circledast}$ will also provide her with a winning strategy in γ^* whereas the converse does not hold, since the number of times Angel has to play γ may depend on the strategy chosen by Demon during the play of the game.

▶ THEOREM 7.12. $\langle\gamma^{\circledast}\rangle\varphi \rightarrow \langle\gamma^*\rangle\varphi$ is valid whereas $\langle\gamma^*\rangle\varphi \rightarrow \langle\gamma^{\circledast}\rangle\varphi$ is not.

PROOF. Using the fixpoint results of the previous chapter, induction on i shows that

$$E^i_\gamma(X) \subseteq \mu Y.X \cup E_\gamma(X)$$

which establishes the validity. For a counterexample to the validity of the converse implication, consider the model of figure 7.2 modified by making p false at state s_4. In the resulting model, $\langle(a \cap b)^*\rangle p$ will be true at s_0 whereas $\langle(a \cap b)^{\circledast}\rangle p$ will be false. ■

With respect to the induction axiom, the two types of iteration behave the same: $(\varphi \wedge [\alpha^{\circledast}](\varphi \rightarrow [\alpha]\varphi)) \rightarrow [\alpha^{\circledast}]\varphi$ is not valid, either. For a counterexample, one can use the model of figure 7.2 which was used to show that induction for * fails. Analogous to the case of *, induction does hold for $^{\circledast}$ in dual-free game logic, the reason being that in that case both kinds of iteration coincide.

▶ THEOREM 7.13. For disjunctive E_γ, $E_{\gamma^*} = E_{\gamma^{\circledast}}$, and hence for dual-free γ, $\langle\gamma^*\rangle\varphi \leftrightarrow \langle\gamma^{\circledast}\rangle\varphi$ is valid.

PROOF. An easy corollary of theorems A.1 and A.3. ■

In other words, for programs, there is no distinction between the two kinds of iteration. Intuitively, if Angel makes all the choices in game γ, there can be no way in which the number of times she needs to play γ can depend on the choices Demon makes.

The difference between γ^* and $\gamma^{\circledast}$ might remind programmers of the difference between while-loops `while` φ `do` γ and for-loops `for i:=1 to n do` γ, where we assume that the value of the variable i is not modified in the body of the for-loop. Under this assumption, for-loops seem weaker than while-loops since the number of iterations is fixed at the beginning. In case the body of the loop γ involves interaction with the user (e.g., letting the user specify the value of a particular variable), the number of γ-iterations cannot be determined in advance, and hence the two loops do not have the same expressive power. If on the other hand γ contains no user-interaction, at run time, the number of iterations of the while-loop can be calculated before the loop is entered, and hence n can be chosen accordingly.

7.5 Complexity

As for model checking, recall that since Kripke models are game models, the model-checking algorithm of the previous chapter can be used for Kripke models as well. As mentioned when discussing the complexity of model checking for Coalition Logic, special purpose model checkers for Kripke models may very well perform better in practice, still the complexity should not differ essentially from the complexity of model checking for general game models. In particular, a crucial feature which determines the complexity will be the alternation depth of the formula.

For satisfiability, we have a precise characterization of the complexity of the decision problem.

▶ THEOREM 7.14. The satisfiability problem for Game Logic over Kripke models is EXPTIME-complete.

PROOF. For the lower bound, since $\mathtt{GL_K}$ includes all of PDL which is known to be EXPTIME-complete (see, e.g., [77, 60]), $\mathtt{GL_K}$ is EXPTIME-hard. For the upper bound, $\mathtt{GL_K}$ is a fragment of the modal μ-calculus (theorem 6.15) whose satisfiability problem is decidable in EXPTIME (see, e.g., [26]). We have already remarked in the proof of theorem 6.22 that the embedding is efficient. ■

Finally, it would be interesting to take a closer look at a more direct proof of EXPTIME-hardness. In particular, one way to prove the EXPTIME-hardness of PDL is via a reduction of the 2-person corridor tiling game to PDL-satisfiability [33, 42]. It would seem that GL would be a much more natural logic to encode these games since it has the direct means for expressing 2-player games.

7.6 Characterizing Game Operations

The set of game operations of Game Logic, sequential composition, union, iteration, test and dual, has intuitive appeal since it seems to be a natural minimal extension of the set of programming constructs of PDL. On the other hand, we have seen in section 7.4 that an alternative form of iteration can be defined which has a perfectly intuitive interpretation as well. Thus the question arises how the game operations of Game Logic can be characterized. What distinguishes the set of game operations of GL? Is there still a sense in which the game operations of GL are complete or maximal? For program operations, a partial answer to this question has been obtained, and in this section we show how this answer can be extended from programs to games, the crucial notions needed being bisimulation invariance and bisimulation safety.

Recall that the general notion of bisimulation defined in section 2.5 corresponds to standard bisimulation over Kripke models. We have come across the notion of bisimulation invariance before.

▶ DEFINITION 7.15 (INVARIANCE & SAFETY). A GL-formula φ is *invariant for bisimulation* if for all game models $\mathcal{M}$ and $\mathcal{M}'$, $s \leftrightarroweq s'$ implies $\mathcal{M}, s \models \varphi \Leftrightarrow \mathcal{M}', s' \models \varphi$. A GL-game γ is *safe for bisimulation* if for all models $\mathcal{M}$ and $\mathcal{M}'$, $s \leftrightarroweq s'$ implies (1) if $sE_\gamma X$ then $\exists X' \subseteq S'$ such that $s'E'_\gamma X'$ and $\forall x' \in X'\ \exists x \in X : x \leftrightarroweq x'$, and (2) if $s'E'_\gamma X'$ then $\exists X \subseteq S$ such that $sE_\gamma X$ and $\forall x \in X\ \exists x' \in X' : x \leftrightarroweq x'$.

While bisimulation invariance should be seen as a generalization of the first bisimulation condition to non-atomic propositions, bisimulation safety generalizes the other two conditions to non-atomic games. Bisimulation safety requires that if Angel can guarantee φ in game γ in one model, she must be able to guarantee something at least as strong in the other model. If this were not the case, the two models could be distinguished by playing γ, since Angel can achieve more in one model than in the other. Theorem 6.5 of the previous chapter thus states that all GL-formulas are invariant for bisimulation and all GL-games are safe for bisimulation. More informally, GL is sound for bisimulation equivalence, i.e., not too expressive: Bisimilar states cannot be distinguished by formulas of the language (invariance), and the game constructions provided do not produce games which can distinguish bisimilar states either (safety).

From the perspective of modal logic, there is a tight correspondence between bisimilar states of a process (Kripke model) and states which make the same modal formulas true: Bisimilar states satisfy the same modal formulas, and for certain classes of Kripke models (e.g. finite models), the converse holds as well. We saw in chapter 3 that such results can be extended to more general models and to more general modal languages such as Coalition Logic (theorems 3.4 and 3.5). Bisimulation-invariance results of this kind make bisimulation an attractive notion of equivalence between models, since it matches the expressive power of the modal language rather well. Furthermore, bisimulation has provided a characterization of the modal fragment of first-order logic (FOL). Modal formulas can be translated into formulas of FOL, and it turns out (see theorem 7.17) that the *modal fragment* of FOL is precisely its bisimulation-invariant fragment.

This line of investigation can be extended from modal logic to Propositional Dynamic Logic: As a corollary to theorem 6.5, PDL-formulas are bisimulation-invariant. Secondly, iteration-free PDL-programs can be translated into FOL as well, raising the question how to characterize the FOL-fragment which (translations of) PDL-programs define. In [15], such a result has been obtained: The *program fragment* of FOL can be characterized as its bisimulation-safe fragment. This result shows that if we take bisimulation as our notion of process equivalence and FOL as our language, the program operations provided by PDL are complete, i.e., no additional program operations will allow us to construct new programs. In this section, we present results which carry the investigation one step further, moving from nondeterministic programs (i.e., 1-player games) to 2-player games, more specifically from PDL to GL.

7.6.1 First-Order Games

It is well-known that modal logic and PDL without iteration can be translated into FOL. In spite of the second-order appearance of Game Logic, a translation into FOL is possible here as well: The signature contains a unary relation symbol C_p for every propositional letter $p \in \Phi_0$, and a binary relation symbol C_g for every atomic game $g \in \Gamma_0$. Furthermore, we allow for second-order variables $X, Y, \ldots$ as well. Thus, the unary relation symbols now comprise constants as well as variables. As will become clear later, we will not quantify over these variables but only use them as a matter of convenience to serve as placeholders for substitution; hence, we can still consider the language to be first-order. We define the translation function $^\circ$ which maps a GL-formula φ to a FOL-formula with one free variable x, and an iteration-free GL-game γ to a FOL-formula with two free variables x and Y.

$$
\begin{aligned}
p^\circ &= C_p x \quad \text{for } p \in \Phi_0 & g^\circ &= \exists z(xC_g z \wedge Yz) \quad \text{for } g \in \Gamma_0\\
(\neg\varphi)^\circ &= \neg\varphi^\circ & (\varphi?)^\circ &= \varphi^\circ \wedge Yx\\
(\varphi \vee \psi)^\circ &= \varphi^\circ \vee \psi^\circ & (\alpha \cup \beta)^\circ &= \alpha^\circ \vee \beta^\circ\\
(\langle\gamma\rangle\varphi)^\circ &= \gamma^\circ[Y := \varphi^\circ] & (\alpha;\beta)^\circ &= \alpha^\circ[Y := \beta^\circ]\\
& & (\alpha^d)^\circ &= \neg\alpha^\circ[Y := \neg Yx]
\end{aligned}
$$

In this definition, substitution for second-order variables is used as follows: Given two FOL-formulas δ and ξ where ξ contains exactly one free first-order variable, say x, $\delta[Y := \xi]$ denotes the result of replacing every occurrence Yt in δ by $\xi[x := t]$. As an example, $\exists z(xC_g z \wedge Yz)[Y := \neg Yx]$ yields $\exists z(xC_g z \wedge \neg Yz)$. Some more remarks on notation: $\varphi(x_1, \dots, x_m, X_1, \dots, X_n)$ refers to a formula φ whose free first-order variables are among $x_1, \dots, x_m$ and whose second-order variables are among $X_1, \dots, X_n$. When a formula has been introduced in this way, $\varphi(t_1, \dots, t_m, T_1, \dots, T_n)$ denotes $\varphi[x_1 := t_1, \dots, x_m := t_m, X_1 := T_1, \dots, X_n := T_n]$, i.e. the simultaneous substitution of t_i for x_i and T_j for X_j in φ.

Regarding the semantics, we can interpret a Kripke model $\mathcal{M} = ((S, \{R_g | g \in \Gamma_0\}), V)$ as a first-order model in the obvious way, taking R_g as the interpretation of C_g, and interpreting C_p as $V(p)$. For a unary predicate symbol C_p and $T \subseteq S$, let $\mathcal{M}_{p:=T}$ be the model which is the same as $\mathcal{M}$ except that $V(p) = T$. Given a model $\mathcal{M}$, states $s_1, \dots, s_m \in S$, sets of states $S_1, \dots, S_n \subseteq S$ and a FOL-formula $\varphi(x_1, \dots, x_m, X_1, \dots, X_n)$, we write $\mathcal{M} \models \varphi[s_1, \dots, s_m, S_1, \dots, S_n]$ to denote that φ is true in $\mathcal{M}$ according to the standard FOL semantics when x_i is assigned the value s_i and X_i the value S_i.

The following result states the semantic correctness of the translation function.

▶ Lemma 7.16. For all GL-formulas φ, games γ and Kripke models $\mathcal{M}$: $\mathcal{M}, s \models \varphi$ iff $\mathcal{M} \models \varphi^\circ[s]$ and $sE_\gamma T$ iff $\mathcal{M} \models \gamma^\circ[s, T]$.

As with the safety result for program constructions, the safety result for game constructions makes use of the characterization of the modal fragment of FOL as its bisimulation-invariant fragment. The definition of invariance and safety which was phrased for GL has its natural first-order analogue: A FOL-formula $\varphi(x)$ is *invariant for bisimulation* if for all models $\mathcal{M}$ and $\mathcal{M}'$, $s \leftrightarrows s'$ implies that $\mathcal{M} \models \varphi[s]$ iff $\mathcal{M}' \models \varphi[s']$. A first-order formula $\varphi(x, Y)$ is *safe for bisimulation* if for all models $\mathcal{M}$ and $\mathcal{M}'$, $s \leftrightarrows s'$ implies (1) if $\mathcal{M} \models \varphi[s, T]$ then there is some T' such that $\mathcal{M}' \models \varphi[s', T']$ and for all $t' \in T'$ there is some $t \in T$ such that $t \leftrightarrows t'$, and (2) if $\mathcal{M}' \models \varphi[s', T']$ then there is some T such that $\mathcal{M} \models \varphi[s, T]$ and for all $t \in T$ there is some $t' \in T'$ such that $t \leftrightarrows t'$.

By a *modal formula* we mean a GL-formula which only contains atomic games (i.e., also no tests). The classic result can now be stated as follows:

▶ Theorem 7.17 (van Benthem [14]). A FOL-formula $\varphi(x)$ is invariant for bisimulation iff it is equivalent to the translation of a modal formula.

For the rest of this section, we will assume that games are iteration-free. In a given model $\mathcal{M}$ with domain S, a FOL-formula $\varphi(x, Y)$ gives rise to a function $E_\varphi^\mathcal{M} : \mathcal{P}(S) \to \mathcal{P}(S)$ by defining $E_\varphi^\mathcal{M}(T) := \{s \in S | \mathcal{M} \models \varphi[s, T]\}$. Call φ *monotonic (disjunctive)* iff for any $\mathcal{M}$, $E_\varphi^\mathcal{M}$ is monotonic (disjunctive). Similarly, a modal formula φ and a proposition letter $p \in \Phi_0$ give rise to such a function $E_{\varphi,p}^\mathcal{M}$ by defining $E_{\varphi,p}^\mathcal{M}(T) := \{s \in S | \mathcal{M}_{p:=T}, s \models \varphi\}$, and we say that φ is *monotonic (disjunctive) in* p iff $E_{\varphi,p}^\mathcal{M}$ is for any $\mathcal{M}$.

Lastly, let $Pos(\varphi)$ ($Neg(\varphi)$) be the set of atomic propositions which occur positively (negatively) in φ, i.e., under an even (odd) number of negations. Thus, formula φ is strictly positive (negative) in p iff $p \notin Neg(\varphi)$ ($p \notin Pos(\varphi)$).

The following lemma relates the syntactic notion of positivity to the semantic notion of monotonicity. It makes use of a Lyndon-interpolation theorem for modal logic and the global deduction theorem, where $\models_g$ refers to *global* consequence over Kripke models.

▶ THEOREM 7.18 (FITTING [47]). If $\models \alpha \to \beta$ for modal formulas α, β, then there exists a modal formula δ such that (1) $\models \alpha \to \delta$, (2) $\models \delta \to \beta$, (3) $Pos(\delta) \subseteq Pos(\alpha) \cap Pos(\beta)$, and (4) $Neg(\delta) \subseteq Neg(\alpha) \cap Neg(\beta)$.

▶ THEOREM 7.19 (FITTING [48]). For modal formulas α and β, $\alpha \models_g \beta$ iff there is some $n > 0$ such that $\models (\Box^1\alpha \wedge \ldots \wedge \Box^n\alpha) \to \beta$, where each $\Box^i$ represents a possibly empty sequence of universal modalities labeled by (possibly different) atomic games.

▶ LEMMA 7.20. A modal formula φ is monotonic in p iff it is equivalent to a modal formula strictly positive in p.

PROOF. One can easily check by induction that strictly positive modal formulas are monotonic, so we shall only prove the other direction. If $\varphi(p)$ is monotonic in p, then taking a proposition letter q not occurring in φ, we have $p \to q \models_g \varphi(p) \to \varphi(q)$. By theorem 7.19, we know that

$$(\Box^1(p \to q) \wedge \ldots \wedge \Box^n(p \to q)) \to (\varphi(p) \to \varphi(q))$$

is valid, and as a consequence,

$$\varphi(p) \to ((\Box^1(p \to q) \wedge \ldots \wedge \Box^n(p \to q)) \to \varphi(q))$$

is also valid. By theorem 7.18, this implies that

$$\varphi(p) \to \delta \quad \text{and} \quad \delta \to ((\Box^1(p \to q) \wedge \ldots \wedge \Box^n(p \to q)) \to \varphi(q))$$

are valid, for some modal formula δ which does not contain q and which is strictly positive in p. The second conjunct implies that $\delta \to \varphi(p)$ is valid: For suppose $\mathcal{M}, s \models \delta$ and $T = \{t | \mathcal{M}, t \models p\}$. Then since δ does not contain q, $\mathcal{M}_{q:=T}, s \models \delta$. From this it follows that $\mathcal{M}_{q:=T}, s \models \varphi(q)$ and hence $\mathcal{M}, s \models \varphi(p)$. Thus, φ is equivalent to δ, a modal formula strictly positive in p. ■

The main lemma we need for our safety result relates monotonic modal formulas to GL-formulas of a special kind.

▶ LEMMA 7.21. Every modal formula φ which is monotonic (disjunctive) in p is equivalent to a GL-formula $\langle\gamma\rangle p$, where γ is a game (program) which does not contain p.

PROOF. The claim about programs and disjunctivity is proved in [68]. As for games and monotonicity, we prove by induction that every modal formula φ which is strictly positive (negative) in p is equivalent to a *GL*-formula $\langle\gamma\rangle p$ $(\neg\langle\gamma\rangle p)$, where γ does not contain p. Then the result follows by lemma 7.20. The following table provides the equivalent *GL*-formulas for every modal formula φ depending on whether φ is strictly positive or strictly negative in p.

modal formula	str. pos/neg	*GL*-formula	ind. hyp.
p	*pos*	$\langle\top?\rangle p$	–
$q \neq p$	*pos*	$\langle q?;\bot?^d\rangle p$	–
$q \neq p$	*neg*	$\neg\langle q?^d;\bot?\rangle p$	–
$\neg\varphi$	*pos*	$\langle\gamma\rangle p$	$\models \varphi \leftrightarrow \neg\langle\gamma\rangle p$
$\neg\varphi$	*neg*	$\neg\langle\gamma\rangle p$	$\models \varphi \leftrightarrow \langle\gamma\rangle p$
$\varphi_1 \vee \varphi_2$	*pos*	$\langle\gamma_1 \cup \gamma_2\rangle p$	$\models \varphi_i \leftrightarrow \langle\gamma_i\rangle p$
$\varphi_1 \vee \varphi_2$	*neg*	$\neg\langle\gamma_1 \cap \gamma_2\rangle p$	$\models \varphi_i \leftrightarrow \neg\langle\gamma_i\rangle p$
$\langle g\rangle\varphi$	*pos*	$\langle g;\gamma\rangle p$	$\models \varphi \leftrightarrow \langle\gamma\rangle p$
$\langle g\rangle\varphi$	*neg*	$\neg\langle g^d;\gamma\rangle p$	$\models \varphi \leftrightarrow \neg\langle\gamma\rangle p$

■

▶ THEOREM 7.22. A FOL-formula $\varphi(x, Y)$ is equivalent to the translation of an iteration-free GL-game iff it is safe for bisimulation and monotonic in Y. A FOL-formula $\varphi(x, Y)$ is equivalent to the translation of an iteration-free GL-program iff it is safe for bisimulation and disjunctive in Y.

PROOF. If $\varphi(x, Y)$ is equivalent to the translation of a GL-game γ, then using lemma 7.16, φ will be monotonic in Y (because E_γ is monotonic) and safe for bisimulation (by theorem 6.5).

For the converse, assume that $\varphi(x, Y)$ is monotonic and safe for bisimulation. Taking a new predicate symbol C_p which does not occur in φ, $\varphi(x, C_p)$ will be invariant for bisimulation. By theorem 7.17, $\varphi(x, C_p)$ is equivalent to the translation of a modal formula δ, i.e. $\models \varphi(x, C_p) \leftrightarrow \delta^\circ$. Since $\varphi(x, Y)$ was monotonic, δ will be monotonic in p and by lemma 7.21, $\models \delta \leftrightarrow \langle\gamma\rangle p$ where γ is a GL-game which does not contain p, and so $\models \varphi(x, C_p) \leftrightarrow (\langle\gamma\rangle p)^\circ$. It can now be checked that $\models \varphi(x, Y) \leftrightarrow \gamma^\circ$: If $\mathcal{M} \models \varphi[s, T]$ then given that C_p does not occur in φ, $\mathcal{M}_{p:=T} \models \varphi(x, C_p)[s]$ and so $\mathcal{M}_{p:=T} \models (\langle\gamma\rangle p)^\circ[s]$. Since p does not occur in γ, this implies that $\mathcal{M} \models \gamma^\circ[s, T]$. The converse is proved along the same lines, and the proof for programs is completely analogous. ■

On the one hand, proposition 7.22 provides a characterization result for the iteration-free games which can be constructed in Game Logic: GL-games are the monotonic bisimulation-safe formulas $\varphi(x, C_p)$ of first-order logic (we can simply replace the variable Y by a designated unary predicate constant C_p). In other words, the game-fragment of FOL is precisely the monotonic bisimulation-safe fragment. On the other hand, looking at the set of operations on games which GL provides, one may ask whether one could not add other natural operations to create new games (e.g., playing games in parallel), thus increasing the expressive power of the language. Proposition 7.22 demonstrates that if the new game operation is (1) first-order definable, (2) monotonic and (3) safe for bisimulation, then it is expressible in GL already. As argued before, requirements (2) and (3) are natural desiderata for games, i.e., they are minimal requirements for any alleged game operation, and so the operations of test, union, composition and dual are sufficient to construct all first-order definable games.

The above result concerning bisimulation-safe programs is different from the result presented in [15, 68]. As discussed in section 7.1, disjunctivity allows one to model programs as relations on states. The program operations of PDL then create complex relations $R_\pi \subseteq S \times S$ which can be translated into FOL-formulas $\varphi(x, y)$. The second part of theorem 7.22 will then read as follows:

> A FOL-formula $\varphi(x, y)$ is safe for bisimulation iff it is equivalent to (the translation of) an iteration-free PDL-program.

While this formulation can do without additional conditions such as monotonicity or disjunctivity, it does not allow for a comparison between programs and games since games cannot be modeled as relations on states. Theorem 7.22 above on the other hand shows that the dual operator makes all the difference between programs and games; without dual, we obtain all first-order definable programs, with dual, all first-order definable games.

7.6.2 Beyond First-Order Games

The last theorem was concerned with Kripke models rather than game models in general. The reason for this restriction is that game models are rather unorthodox structures. We do not know of any logical languages besides non-normal modal logics and Game Logic which have been proposed for these structures. Consequently, this prevents an easy extension of the definability result of proposition 7.22 to GL over general game models.

Even for Kripke models, the translation into FOL carried out in the previous section relied on the restriction to iteration-free games. For programs, a stronger definability result covering iteration has been obtained in [68] which characterizes the class of monadic-second-order definable programs which are safe for bisimulation. The proof makes use of the fact that the bisimulation-invariant fragment of monadic second-order logic is the μ-calculus [71]. An extension of proposition

7.22 along these lines, however, would require a better understanding of how exactly GL relates to the μ-calculus. As for the μ-calculus itself, many fundamental properties were established only recently, such as completeness [124], the non-collapse of the alternation-hierarchy [25, 81] and uniform interpolation [36], and others such as Lyndon interpolation are still open.

To summarize, the restriction of the scope of proposition 7.22 to FOL is due to the fact that FOL is one of the logics we know most about and is able to express the most fundamental game operations. When moving to stronger languages, different options are available, always depending on the game constructions one is interested in.

7.7 Discussion

To sum up our investigation of Game Logic, chapters 6 and 7 have considered four logics which we shall identify by their (conjectured) axiomatic versions: Game Logic over general game models $\mathtt{GL}$, Game Logic over Kripke models $\mathtt{GL}_\mathtt{K}$, and their respective program fragments $\mathtt{GL}^{-\mathtt{d}}$ and $\mathtt{PDL}$. The previous chapter focused on a comparison between $\mathtt{GL}$ and $\mathtt{GL}^{-\mathtt{d}}$, making use of the generalized modal μ-calculus to obtain complexity and expressiveness results.

The present chapter has yielded a number of differences between $\mathtt{PDL}$ and $\mathtt{GL}_\mathtt{K}$. We can interpret these as differences between program operations and game operations in the case where all (if any) interaction is introduced by these operations, i.e., the atomic level only consists of 1-player games. The differences and similarities are summarized in figure 7.3:

	Game Logic $\mathtt{GL}_\mathtt{K}$	program fragment $\mathtt{PDL}$
complete axiomatization	not yet	yes
induction axiom valid	no	yes
expressive power		>
complexity satisfiability	EXPTIME-complete	EXPTIME-complete
FOL-definable operations	$\cup$; ? ^{d}	$\cup$; ?

Figure 7.3: Differences between Game Logic and its program fragment over Kripke models.

The last entry of the table shows that games are programs plus role switch. Under bisimulation equivalence, the operations of choice, sequential composition and test suffice to obtain all first-order definable programs; choice, sequential composition, test and duality suffice to obtain all first-order definable games. A further result not covered by the table is that games allow for two kinds of iteration which coincide for programs.

As was to be expected, a number of open questions remain, most notably a proof of axiomatic completeness and a generalization of the bisimulation-safety result to include iteration. What is more interesting, the translation of Game Logic into the modal μ-calculus has brought up questions which so far have not been considered in the literature on the subject: What are examples of μ-calculus formulas which can be expressed with n variables but not with less than n? How expressive are these finite variable fragments? An answer to these questions will also provide a better characterization of the precise expressive power of Game Logic.

In spite of theorem 7.22, the investigation of other game operations, in particular parallelism, also constitutes a promising line for future research. Semantically, a strategic game can be viewed as the parallel composition of the actions/strategies of the players, so in a sense strategic games already involve parallelism. Still, Game Logic in its current form has no operation for playing two games in parallel. In general, such an operation should be an extension of parallelism as defined for programs, as is done, e.g., in Concurrent Dynamic Logic or in Process Algebra. Alternatively, inspirations can be taken from Linear Logic, where infinite concurrent plays have been used in the semantics for the multiplicative operators. As an example, $A \otimes B$ denotes the 2-player game where A and B are played concurrently and player 2 (Opponent) is allowed to switch back and forth between the component games. Some first attempts to define parallelism for Game Logic have been made in [89, 90].

7.8 Bibliographic Notes

The results of section 7.6 were first published in [100]. Theorem 7.14 has been stated in [45], but one should be careful to note that the "known lower bound due to PDL" only holds for Game Logic over Kripke models.

For Propositional Dynamic Logic, see [59, 77] for comprehensive survey articles and [60] for a recent textbook. A further reference which treats also other logics used in computer science is [54]. RPDL (which is also known as ΔPDL) and a closely related system LPDL (PDL + *loop*) are studied in [61]. Concurrent Propositional Dynamic Logic was introduced in [108] and followed up by [107].

For textbooks on Process Algebra, see [49, 8]. Game semantics for Linear Logic is discussed in [21, 2].

Chapter 8

Game Over

In this final chapter, we take another look at the relationship between Game Logic and Coalition Logic, showing that they embody two different approaches to reasoning about multi-agent systems. As shall be explained, the difference between these two approaches is in fact well-known in computer science. In section 8.2, we then take stock of the theoretical results obtained in this thesis, on the one hand regarding the relationship between programs and games and on the other hand regarding the differences between reasoning about individuals and reasoning about coalitions. Finally, section 8.3 addresses the question how far Game Logic and Coalition Logic go on the way to a logic of social software.

8.1 Bringing it All Together

As shown in chapter 2, both Coalition Logic and Game Logic make use of essentially the same underlying semantics, interpreted either as an internal model of a single game or as an external model of multiple games. In fact, the difference between the two logics and their uses is analogous to the difference between PDL and temporal logic (TL), as summarized by the following equation:

$$\frac{\text{CL}}{\text{GL}} = \frac{\text{TL}}{\text{PDL}}$$

As logics for reasoning about software, temporal logics such as CTL, CTL*, etc. differ in a number of ways from program logics such as PDL and its extensions (see e.g. [70] for a more detailed comparison): In the terminology of [77], PDL is an *exogenous* logic since programs are an explicit part of the logical language. In contrast, temporal logic is *endogenous*: The model itself is the fixed program over which expressions are interpreted. The difference between endogenous and exogenous logics thus corresponds to the difference between the internal and the external view of games, explained in section 2.4. This central difference has far reaching technical consequences. First, program verification takes a different form in the two approaches: In the endogenous approach, program verification takes the form of model checking, in the exogenous approach it takes the form of theorem proving or satisfiability checking. Second, the endogenous approach can be used only to reason about systems with a finite number of states. This restriction comes from the use of model checking for verification: If there are infinitely many states, the denotation of a formula may not be computable. The exogenous approach on the other hand is not limited to such finite-state systems. Third and finally, the class of programs one can reason about in the exogenous approach is limited to compositional systems, where complex programs are constructed using a limited number of program constructions for which inductive proof rules exist. The endogenous approach on the other hand can also deal with non-compositional systems and is thus more general in this respect.

The differences between TL and PDL on the right software side of the equation equally apply on the left social software side between GL and CL. Naturally, there are some more differences on the social software side: In contrast to GL, CL can describe games with more than two players and it can express the effectivity of non-singleton coalitions, differences which have no analogue on the right software side of the equation. Still, it is easy to see how GL could be extended to games with more than 2 players, in fact [97] already contains such a proposal.

In terms of applications to social software, chapter 5 has illustrated some applications of Coalition Logic. The examples have illustrated not only verification (via model checking) but also synthesis (via satisfiability tests) of social software. As for applications of Game Logic in the verification of social software, Parikh in [97] provides an example of how Game Logic can be used to verify the fairness of a cake-cutting algorithm, suggesting that even propositional Game Logic can be useful to verify properties of simple multi-agent algorithms. Not being completely convinced by this application, however, we think that stronger logical frameworks such as the refinement calculus [7] (see also section 8.3) are needed to treat interesting examples. There is, however, another way in which Game Logic could indirectly turn out to be very useful for social software verification, namely as game algebra.

Game Algebra

The operations of Game Logic have also be studied from an algebraic perspective [55, 122]. Recall that a complex game expression γ of Game Logic denotes a predicate transformer $E_\gamma : \mathcal{P}(S) \to \mathcal{P}(S)$. Hence it is natural to call two game expressions γ_1 and γ_2 *equivalent* provided that $E_{\gamma_1} = E_{\gamma_2}$ holds for all game models. Put differently, γ_1 and γ_2 are equivalent if $\langle\gamma_1\rangle p \leftrightarrow \langle\gamma_2\rangle p$ is valid for a p which occurs neither in γ_1 nor in γ_2. When γ_1 and γ_2 are equivalent, we say that $\gamma_1 = \gamma_2$ is a *valid game identity*.

Basic game algebra studies the game operations of sequential composition, choice (demonic and angelic) and duality. The test-operator is excluded since it would take us out of the purely algebraic framework; iteration on the other hand could be added but has not been investigated so far. The central problem of basic game algebra is to axiomatize the set of valid game identities. The conjectured axiomatization of [12] has been proved complete in [55], and an alternative algebraic proof has been given in [122]. Consider the following game identities (taken from [122]), where we write duality as $-$, angelic choice as $\vee$ and demonic choice as $\wedge$:

$$
\begin{array}{ccr}
x \vee x = x & x \wedge x = x & \text{(G1)} \\
x \vee y = y \vee x & x \wedge y = y \wedge x & \text{(G2)} \\
x \vee (y \vee z) = (x \vee y) \vee z & x \wedge (y \wedge z) = (x \wedge y) \wedge z & \text{(G3)} \\
x \vee (x \wedge y) = x & x \wedge (x \vee y) = x & \text{(G4)} \\
x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z) & x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z) & \text{(G5)} \\
- - x = x & & \text{(G6)} \\
-(x \vee y) = -x \wedge -y & -(x \wedge y) = -x \vee -y & \text{(G7)} \\
x; (y; z) = (x; y); z & & \text{(G8)} \\
(x \vee y); z = x; z \vee y; z & (x \wedge y); z = x; z \wedge y; z & \text{(G9)} \\
-(x; y) = -x; -y & & \text{(G10)} \\
x; y \vee x; (y \vee z) = x; (y \vee z) & & \text{(G11)}
\end{array}
$$

Axioms G1-7 are well known axioms of boolean algebra, where angelic choice corresponds to disjunction or join, demonic choice to conjunction or meet, and dual to negation. Axiom G9 is a principle which is already present in process algebra [49, 8]: If a choice of player i between x and y is followed by game z in any case, then player i might as well choose between $x; z$ and $y; z$ directly. Note that the right-distributive law $x; (y \vee z) = x; y \vee x; z$ on the other hand is not valid. In the first game, player 1 can postpone her choice until after game x has been played. She may have a winning strategy which depends on how x is played, and hence such a strategy will not necessarily be winning in the second game, where she has to choose before x is played. Axiom G11 may be easier to read as $x; y \leq x; (y \vee z)$ or as the quasi-equation

$$y \leq z \to x; y \leq x; z,$$

where $a \leq b$ abbreviates $a \vee b = b$. The axiom states the right-monotonicity of sequential composition, based on the monotonicity of the predicate transformers E_γ which interpret the game terms.

Soundness of these axioms can easily be verified. Furthermore, one can show that any valid game identity can be derived from these axioms using equational logic. So far, this result has not been extended to a version of game algebra which includes iteration. As shown in [11], game algebra also provides an interesting perspective on the semantic evaluation games of first-order logic.

Moving from theory to practice, how can Game Algebra be useful for the analysis of social software? It can serve as the basis for an algebraic description language for multi-agent systems. In the same way in which Process Algebra can be used as a language for describing concurrent systems, the language of Game Algebra can be used to describe 2-player games. The laws of Game Algebra can then be used to simplify or more generally transform the games described preserving semantic equivalence. Finally, Coalition Logic (or Alternating Temporal Logic) can then serve as a specification language, i.e., formulas of Coalition Logic can be used to specify properties which are verified by model checking in the model generated from the Game Algebra expression. Succinctly, we obtain the following equation relating Coalition Logic and Game Algebra (GA) to temporal logic and Process Algebra (PA):

$$\frac{CL}{GA} = \frac{TL}{PA}$$

Needless to say, Game Algebra in its current form can only be the first step in the direction sketched, comparable to basic Process Algebra (BPA) which serves as a basis for a wealth of different extensions including concurrency and communication. Furthermore, the equation suggests that the semantic notion of equivalence employed in Game Algebra may actually not be the appropriate one. If a Game Algebra expression denotes a complex game whose properties are to be verified using Coalition Logic, equivalence defined in terms of simple overall effectivity will be too crude. Instead, generalized bisimulation as introduced in section 2.5 should be much more suitable. Investigating the axiomatic differences between Game Algebra as defined above and its relative which is based on bisimulation equivalence is an interesting open question for further research.

8.2 Summary of Theoretical Results

As we have seen, reasoning about programs can be compared to reasoning about games using endogenous as well as exogenous logics. In each case, the technical results obtained for games can be compared to the results we have for programs. Below we put these theoretical insights into a number of slogans and show how standard meta-theoretic results about axiomatization, complexity, bisimulation and expressiveness can yield insights not only about differences between programs and games but also about differences between reasoning about individuals vs. coalitions. Furthermore, we remark on the role which iteration plays in these results, and we point out the relevance of the results for the practice of social software design and analysis.

The External View: Game Logic vs. Program Logic

> *Semantically, disjunctivity and induction distinguish programs from games.*

Semantically, any individual effectivity function E models ability in a determined 2-player game (corollary 2.13). If E is disjunctive, it models ability in a 1-player game (theorem 2.16), so disjunctivity makes all the difference between programs and determined 2-player games. As a consequence of disjunctivity, the induction axiom is valid for programs but not for games (theorems 7.9 and 7.10).

Syntactically, program operations + duality = game operations.

Comparing the program operations of Propositional Dynamic Logic to the game operations of Game Logic, duality is the only difference. Theorem 7.22 suggests that this is no coincidence: The operations of test, sequential composition and choice suffice to obtain all first-order definable programs. Adding duality to these operations yields all first-order definable games.

In the end, games have more expressive power than programs.

Using duality, Game Logic can express properties of models which cannot be expressed in Propositional Dynamic Logic (theorem 7.2). The expressive difference relies on the presence of iteration which allows one to express the existence of a strategy to achieve something in the long run. Without iteration, duality does not increase expressive power, and hence programs are as expressive as games.

Verifying properties of programs is easier than verifying properties of games.

At the time of writing, model checking is more complex for games than for programs (theorem 6.22). This claim is only tentative since better polynomial-time algorithms may be found for μ-calculus model checking. Hence one should say more accurately that verifying properties of games is equally complex as verifying properties of programs if (and only if [18]) model checking for the μ-calculus can be done in polynomial time.

The Internal View: Coalition Logic vs. Temporal Logic

The comparison between Temporal Logic and Coalition Logic is unfortunately more difficult than the comparison between PDL and GL. The reason is that there is a bunch of systems to choose from: On the one hand we have basic Coalition Logic and Extended Coalition Logic, on the other hand we have simple modal/temporal logic, CTL, CTL*, and so on. Furthermore, the models over which these logics are interpreted differ, and interpreting Coalition Logic over Kripke models for purposes of comparison will rob it of all its characteristic features (in contrast to Game Logic).

On the most basic level, we would want to compare basic Coalition Logic with normal modal logic. The complexity of model checking is linear time for both modal logic and Coalition Logic, although the model size will be different. We saw that the complexity of the satisfiability problem for Coalition Logic is usually PSPACE-complete, as for modal logic. Similarly, there are cases where the complexity of the satisfiability problem turns out to be NP-complete, for strong normal modal logics such as S5 as well as for formulas of the individual fragment of Coalition Logic, when interpreted over extensive games with simultaneous moves. As remarked in section 3.8.2, these complexity results do give rise to an interesting difference between programs and games. If it turns out that NP $\neq$ PSPACE (which we shall assume at least for these slogans):

> *Game synthesis is easier than program synthesis, provided we allow for simultaneous moves.*

That is, given an individual specification for a single player, formulated in basic Coalition Logic, finding a satisfying extensive game with simultaneous moves is an NP-complete problem whereas finding a satisfying program/process or a satisfying extensive game without simultaneous moves is PSPACE-complete.

Given the close relationship between Extended Coalition Logic and Alternating Temporal Logic (ATL), one can also compare ATL with its standard temporal counterpart CTL to get a better idea of how closed systems (programs) differ from open systems (games). For the complexity of model checking, such a comparison has been carried out in [3].

Instead of comparing Coalition Logic to temporal or modal logic, below we will compare Coalition Logic to its individual fragment. Putting our results in this light will yield some interesting differences between coalitional and individual reasoning.

> *Game synthesis is easier for individual than for coalitional specifications, provided we allow for simultaneous moves.*

For extensive games with simultaneous moves, the satisfiability problem for Coalition Logic is PSPACE-complete (theorems 3.27 and 3.29) while the satisfiability problem of its individual fragment is NP-complete (theorem 3.36). Generating such a game from a specification formulated in the individual fragment of basic Coalition Logic is thus simpler than generating it from a coalitional specification, provided that NP $\subsetneq$ PSPACE. In other words, NP = PSPACE if and only if multi-agent synthesis for individual specifications is equally complex as for coalitional specifications.

> *Game synthesis is easier for extensive games with than for extensive games without simultaneous moves, given individual specifications.*

For individual specifications, the satisfiability problem is NP-complete (theorem 3.36) over extensive games with simultaneous moves but PSPACE-complete (theorem 3.37) over extensive games without simultaneous moves. So again, NP

= PSPACE if and only if multi-agent synthesis for individual specifications is equally complex in extensive games with and without simultaneous moves. Assuming NP $\neq$ PSPACE, we can also read this result as demonstrating a difference between environments of perfect and imperfect information: It is simpler to generate multi-agent environments satisfying certain specifications in case we can hide information from the agents. Developing an environment which gives agents certain powers is easier if we have the means to prevent agents from being perfectly informed about the others' actions.

In the end, coalitions have more expressive power than individuals.

For extensive games with simultaneous moves, a gain in expressiveness can already be observed on the local level (theorem 3.35): a language which can express local coalitional ability is more expressive than a language which can only express local individual ability. For extensive games without simultaneous moves, coalitions only add expressiveness in the end, i.e., when the language is enriched to express what can be achieved at some point in the future (theorem 4.12).

Modal logics are game logics.

Normal modal logics describe 1-player games (i.e., Kripke models), in particular the basic normal modal logic `K` coincides with basic 1-player Coalition Logic over weakly playable coalition models (theorem 3.22). Non-normal monotonic modal logics describe determined 2-player games (i.e., neighborhood models), the basic monotonic modal logic `M` coincides with the individual fragment of basic 2-player Coalition Logic over weakly playable maximal coalition models (theorem 3.24). While normal modal logic is coalitional (it can express the ability of the empty coalition), non-normal modal logic is not (it can only express the ability of the two players individually), thus providing one explanation of the complexity difference in the satisfiability problem (see above).

The Role of Iteration

Both on the internal and on the external view, increased expressive power can be seen to depend on the presence of some form of iteration. In the case of extensive games without simultaneous moves, coalitions only add expressive power when long-term ability can be expressed. On the other hand, game operations only lead to more expressiveness in case iteration is present.

Iteration is also responsible for the observed complexity differences. But while on the internal view a difference between individuals and coalitions can only be observed with the satisfiability problem, on the external view the difference between programs and games emerges in model checking only: In case the atomic games are actually programs, the satisfiability problems for Game Logic and for Propositional Dynamic Logic are both EXPTIME-complete (theorem 7.14);

in case no restriction is placed on the atomic games, we only know that the satisfiability problem for Game Logic is in EXPTIME (theorem 6.24), but we conjecture that EXPTIME-hardness also holds, for games as well as programs.

Implications for Social Software

Besides being interesting on a theoretical level, the technical results obtained are all of practical importance for the development and analysis of social software. This is most easily seen for the axiomatization and complexity results: Based on a complete axiomatization, (semi-)automatic theorem provers can verify properties of social procedures or deduce that a particular specification of a voting procedure is inconsistent. In order to verify properties of social software or synthesize social procedures which meet certain specifications, we also need algorithms for model checking and satisfiability testing. Furthermore, it is important to know how complex these algorithms are. Similarly, studying the expressiveness of the logics involved will allow us to pick the right logical language for the task at hand. If a specification can be formulated in the individual fragment of basic Coalition Logic, we can generate an implementation much more efficiently than if a specification expresses something about the long-term ability of groups of agents, something for which full Extended Coalition Logic is needed. Bisimulation-invariance and -safety results become important once we have generated, e.g., a voting procedure and want to simplify it in various ways. As long as this simplification results in a bisimilar voting procedure, we are guaranteed that its properties will not change, provided these properties are expressible in Coalition Logic.

Note that the kinds of results established in this thesis are certainly not the only ones which could be of relevance for the formal study of social software. To give one example, analogous to the case of temporal logic, it would be useful to have preservation theorems which link a class of model transformations (such as adding/deleting states, etc.) to the class of CL-formulas whose truth values remain unchanged by these transformations.

8.3 The Future of Social Software

As discussed in the previous sections, Game Logic and Coalition Logic represent two very different approaches to reasoning about social software, one being exogenous and the other endogenous. In giving examples of applications, we have focused on Coalition Logic because we think that in contrast to Game Logic, Coalition Logic can be applied to the analysis of social software *as it is*, analogous to how Temporal Logic is applied to the analysis of, e.g., concurrent systems. The insufficiency of Game Logic does not mean, however, that it cannot be extended to a logical framework for reasoning about multi-agent algorithms. In fact, the refinement calculus can be viewed as one such extension.

The aim of the refinement calculus [7] is to analyze "real" programs like the *gcd*-program discussed in the introduction. For this aim, a simple propositional logic like Game Logic is not sufficient for formalization; instead, higher-order logic is used in [7]. What makes the refinement calculus relevant to the study of programs and games is that it also provides a general framework which can be used to reason about programs as well as games. This is not so surprising after all since the basic semantic notion of the refinement calculus is the predicate transformer, and we have seen that predicate transformers can model programs as well as determined 2-player games. The extended programming language of the refinement calculus is very similar to the language of Game Logic. While the operation of duality is also discussed, its role is less central in the refinement calculus. Preference is given to using demonic and angelic versions of the basic programming operations such as nondeterministic choice. In this enriched programming language, programs are viewed as *contracts* which define the rights and obligations of the parties involved. Contracts are essentially what we have been calling games all along. The advantage of the contract metaphor is that it allows for a natural interpretation of empty games: under the contract interpretation, this is simply the case where one of the agents has breached the contract. 2-player zero-sum games such as Nim can be programmed using the language of the refinement calculus. The result is a game expression γ such that (in Game Logic terms) $\langle\gamma\rangle\bot$ holds precisely when Angel has a winning strategy in the game. Besides formally proving the existence of such a winning strategy, the refinement calculus can also be used to extract a concrete winning strategy: Starting with the original game γ, the choices of angel are restricted step by step until all the choices left are demonic in the final program γ'. If we ensure that each step in this refinement is semantically sound, we can guarantee that γ' is indeed a winning strategy for γ.

Returning to the topic of social software, how far do Coalition Logic, Game Algebra and (extensions of) Game Logic go in providing formal tools for the analysis and synthesis of social software? First, which kinds of examples of social software can we handle? As argued in chapter 5, the main requirement is that the social process to be analyzed is itself well-defined, allowing us e.g. to identify the set of agents involved in the process and the relevant properties of the states of the process.

Many of the social software examples presented were essentially voting problems since these problems meet the requirement of well-definedness. More generally, there is a practical argument for focusing on social software from the domain of social choice theory (broadly conceived), namely that a lot of research has been done in social choice theory which can be usefully applied, so that social choice theory and logic can meet half way. An example is the rich temporal logic of rights developed in [63, 64] which distinguishes alethic from deontic possibility. The language of this *Deontic Logic of Action* contains formulas of the form $CanDo_i(t_k, \varphi)$ (agent i can act at time t_k so as to bring about φ), $MayDo_i(t_k, \varphi)$ (agent i has the right at time t_k to act in a way which brings about φ) and others. Preferences are also added to the model so that various paradoxes involving liberalism and constitutional decision making can be discussed and formalized.

Having suggested problems of social choice theory as the kind of social software to which the logics presented can usefully be applied, what kinds of questions do these logics allow us to address? Chapter 5 has given examples of verification and synthesis of social procedures. Furthermore, the example of telephone democracy even suggests that the efficiency of such procedures can be analyzed using basic Coalition Logic. Note, however, that the logics discussed cannot express anything about the preferences of the agents involved, nor about how their actions will be influenced by their preferences. For this reason, we cannot capture any strategic considerations, e.g., in the telephone democracy example. In spite of this limitation, the examples have shown that an interesting analysis of social software can be done even without considering the agents' preferences. Before asking what people want to do in a social process, we should make sure that the process gives them the rights and duties they should have.

Comparing social software to computer software in terms of its complexity, one might be tempted to think that social processes must be far more complex than computational processes. On the other hand, we conjecture that even the most complex voting system used in any human society will be much less complex than the operating system used on most computers. Consequently, problems which may be intractable for computer software may well turn out to be tractable for social software. As shown in chapter 5, in the case of voting procedures the synthesis of social software reduces to satisfiability testing in basic Coalition Logic. The results of chapter 3 show that this problem is certainly feasible, since it is no harder than theorem proving in standard modal logic.

To conclude, we hope to have convinced the reader that one can treat social processes as social software by developing logics as analytical tools as is done in computer science. Exogenous program logics and in particular endogenous temporal logics can both be extended to yield logics for reasoning about social processes. We do not think that we will arrive at *one* general framework which is adequate to analyze all or even most social processes. Rather, we expect to see a variety of logics developed for different purposes, varying in complexity and expressive power like the different logics used in computer science. And if the theoretical results obtained and the examples provided still do not manage to convince the reader, we will have to close with a quote by Vince Lombardi, an American football coach:

We didn't lose the game; we just ran out of time.

Appendix A

Fixpoint Facts

Chapters 4 and 6 make use of fixpoint constructions to define long-term ability and iteration, respectively. This appendix recalls some standard results about fixpoints, namely, the Knaster-Tarski fixpoint theorem and the upward and downward hierarchies for fixpoint approximation. The material is standard, with the possible exception of theorem A.2, a generalization of the Knaster-Tarski fixpoint theorem.

Consider any monotonic operation on the nonempty set of states S, i.e., any function $F : \mathcal{P}(S) \to \mathcal{P}(S)$ such that $X \subseteq Y$ implies $F(X) \subseteq F(Y)$. We say that a set $Z \subseteq S$ is a *fixpoint* of F iff $F(Z) = Z$. Z is a *least (greatest) fixpoint* of F iff (1) Z is a fixpoint and (2) Z is a subset (superset) of every fixpoint of F. Note that least and greatest fixpoints are unique. We denote the least fixpoint of F as $\mu X.F(X)$ (the smallest set X such that $F(X) = X$) and the greatest fixpoint of F as $\nu X.F(X)$ (the greatest set X such that $F(X) = X$).

For repeated application of the operation F, we define the following downward and upward hierarchies by ordinal induction:

$$
\begin{array}{ll}
F^{\uparrow 0}(X) = X & F^{\downarrow 0}(X) = X \\
F^{\uparrow \kappa+1}(X) = F(F^{\uparrow \kappa}(X)) & F^{\downarrow \kappa+1}(X) = F(F^{\downarrow \kappa}(X)) \\
F^{\uparrow \lambda}(X) = \bigcup_{\kappa<\lambda} F^{\uparrow \kappa}(X) & F^{\downarrow \lambda}(X) = \bigcap_{\kappa<\lambda} F^{\downarrow \kappa}(X)
\end{array}
$$

where κ and λ are ordinals and λ is a limit ordinal. In most cases, the upward hierarchy will be used for $X = \emptyset$ and the downward hierarchy for $X = S$, and for ease of notation, we use $F^{\uparrow \kappa}$ for $F^{\uparrow \kappa}(\emptyset)$ and $F^{\downarrow \kappa}$ for $F^{\downarrow \kappa}(S)$. A central result on fixpoints is the well-known Knaster-Tarski fixpoint theorem:

► THEOREM A.1 (TARSKI [118]). If $F : \mathcal{P}(S) \to \mathcal{P}(S)$ is any monotonic operation, then

1. $\mu X.F(X) = \bigcap\{Y \subseteq S | F(Y) = Y\} = \bigcap\{Y \subseteq S | F(Y) \subseteq Y\} = \bigcup_{\kappa} F^{\uparrow \kappa}$, where κ ranges over all ordinals of cardinality at most $|S|$ and $F^{\uparrow 0} \subseteq F^{\uparrow 1} \subseteq F^{\uparrow 2} \ldots$.

2. $\nu X.F(X) = \bigcup\{Y \subseteq S | F(Y) = Y\} = \bigcup\{Y \subseteq S | F(Y) \supseteq Y\} = \bigcap_\kappa F^{\downarrow\kappa}$, where κ ranges over all ordinals of cardinality at most $|S|$ and $F^{\downarrow 0} \supseteq F^{\downarrow 1} \supseteq F^{\downarrow 2} \ldots$

In section 6.5, a less well-known generalization of this theorem will allow us to reduce the complexity of a model-checking algorithm substantially. It is an easy consequence of the previous result.

▶ COROLLARY A.2 (EMERSON & LEI [46]). If $F : \mathcal{P}(S) \to \mathcal{P}(S)$ is any monotonic operation, then

1. $\mu X.F(X) = \bigcup_\kappa F^{\uparrow\kappa}(X_0)$ for any $X_0 \subseteq F(X_0) \cap \mu X.F(X)$, where κ ranges over all ordinals of cardinality at most $|S|$ and $F^{\uparrow 0}(X_0) \subseteq F^{\uparrow 1}(X_0) \subseteq F^{\uparrow 2}(X_0) \ldots$

2. $\nu X.F(X) = \bigcap_\kappa F^{\downarrow\kappa}(X_0)$ for any $X_0 \supseteq F(X_0) \cup \nu X.F(X)$, where κ ranges over all ordinals of cardinality at most $|S|$ and $F^{\downarrow 0}(X_0) \supseteq F^{\downarrow 1}(X_0) \supseteq F^{\downarrow 2}(X_0) \ldots$

Finally, there are cases in which the fixpoint approximation provided by the upward and downward hierarchies is guaranteed to reach the fixpoint after at most ω stages. A well-known sufficient condition for such a closure at ω is disjunctivity. As defined in section 2.4.3, an operation $F : \mathcal{P}(S) \to \mathcal{P}(S)$ is disjunctive iff for all $V \subseteq \mathcal{P}(S)$ we have $F(\bigcup_{X \in V} X) = \bigcup_{X \in V} F(X)$. Recall that disjunctivity implies monotonicity and that $F(\emptyset) = \emptyset$. As an analogue to disjunctivity, call F *conjunctive* iff for all $V \subseteq \mathcal{P}(S)$ we have $F(\bigcap_{X \in V} X) = \bigcap_{X \in V} F(X)$. Also conjunctivity implies monotonicity and furthermore that $F(S) = S$.

The following result shows that indeed disjunctivity (conjunctivity) is a sufficient condition for approximating the fixpoint after at most ω steps. Note that there are weaker conditions such as continuity which are also sufficient (see, e.g., [37, 91]), but for our purposes the following result is exactly what we need.

▶ THEOREM A.3. If F is disjunctive then $\mu X.F(X) = F^{\uparrow\omega}$, and if F is conjunctive then $\nu X.F(X) = F^{\downarrow\omega}$.

PROOF. Disjunctivity immediately implies that $\bigcup_{i<\omega} F^{\uparrow i}$ is a fixpoint of F, and given any fixpoint Z of F one can show by induction on i that $F^{\uparrow i} \subseteq Z$ and consequently $\bigcup_{i<\omega} F^{\uparrow i} \subseteq Z$. ■

Bibliography

[1] J. Abdou and H. Keiding. *Effectivity Functions in Social Choice.* Kluwer, 1991.

[2] S. Abramsky and R. Jagadeesan. Games and full completeness for multiplicative linear logic. *Journal of Symbolic Logic*, 59:543–574, 1994.

[3] R. Alur, T. Henzinger, and O. Kupferman. Alternating-time temporal logic. In *Compositionality: The Significant Difference*, LNCS 1536, pages 23–60. Springer, 1998.

[4] R. Alur, T. Henzinger, O. Kupferman, and M. Vardi. Alternating refinement relations. In *Proceedings of CONCUR '98*, LNCS 1466, pages 163–178. Springer, 1998.

[5] K.R. Apt and G.D. Plotkin. Countable nondeterminism and random assignment. *Journal of the ACM*, 33:724–767, 1986.

[6] A. Arnold and D. Niwinski. *Rudiments of μ-calculus.* Elsevier, 2001.

[7] R.-J. Back and J. von Wright. *Refinement Calculus - A systematic introduction.* Springer, 1998.

[8] J. Baeten and W. Weijland. *Process Algebra.* Cambridge University Press, 1990.

[9] A. Baltag. A logic for suspicious players: epistemic actions and belief-updates in games. Technical Report SEN-R0044, Centrum voor Wiskunde en Informatica (CWI), 2000.

[10] P. Battigalli and G. Bonanno. Recent results on belief, knowledge and the epistemic foundations of game theory. *Research in Economics*, 53:149–225, 1999.

[11] J. van Benthem. Basic game algebra inside predicate logic. Unpublished manuscript.

[12] J. van Benthem. *Logic in Games.* Lecture notes, unpublished.

[13] J. van Benthem. When are two games the same? Unpublished manuscript.

[14] J. van Benthem. *Modal Correspondence Theory.* PhD thesis, University of Amsterdam, 1976.

[15] J. van Benthem. Program constructions that are safe for bisimulation. *Studia Logica*, 60(2):311–330, 1998.

[16] J. van Benthem and J. Bergstra. Logic of transition systems. ILLC Prepublication Series CT-93-03, University of Amsterdam, 1993.

[17] J. van Benthem, J. van Eijck, and V. Stebletsova. Modal logic, transition systems and processes. *Journal of Logic and Computation*, 4(5):811–855, 1994.

[18] D. Berwanger. Game logic is strong enough for parity games. In M. Pauly and G. Sandu, editors, *Proceedings of the ESSLLI Workshop on Logic and Games*, 2001.

[19] K. Binmore. *Fun and Games – A Text on Game Theory.* Heath, 1992.

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

[21] A. Blass. A game semantics for linear logic. *Annals of Pure and Applied Logic*, 56:183–220, 1992.

[22] A. Bleeker and J. van Eijck. Epistemic action and change. In G. Bonanno, E. Colombatto, and W. van der Hoek, editors, *Proceedings of the Fourth Conference on Logic and the Foundations of Game and Decision Theory (LOFT4)*, 2000.

[23] G. Bonanno. Branching time logic, perfect information games and backward induction. Unpublished manuscript.

[24] G. Bonanno. The logic of rational play in games of perfect information. *Economics and Philosophy*, 7:37–65, 1991.

[25] J. Bradfield. The modal mu-calculus alternation hierarchy is strict. In U. Montanari and V. Sassone, editors, *Proceedings of CONCUR '96*, LNCS 1119, pages 233–246. Springer, 1996.

[26] J. Bradfield and C. Stirling. Modal logics and mu-calculi: an introduction. In J. Bergstra, A. Ponse, and S. Smolka, editors, *Handbook of Process Algebra*. Elsevier, 2001.

[27] S. Brams and A. Taylor. *Fair Division: From Cake-cutting to Dispute Resolution*. Cambridge University Press, 1996.

[28] A. Brown, S. Mantha, and T. Wakayama. Exploiting the normative aspect of preference: a deontic logic without actions. *Annals of Mathematics and Artificial Intelligence*, 9:167–203, 1993.

[29] B. de Bruin. Game transformations and game equivalence. Technical Report X-1999-01, University of Amsterdam, ILLC, 1999.

[30] B. de Bruin. Modeling knowledge in games - topology and logic. Master's thesis, University of Amsterdam, 2000.

[31] A. Chandra, D. Kozen, and L. Stockmeyer. Alternation. *Journal of the ACM*, 28(1):114–133, 1981.

[32] B. Chellas. *Modal Logic - An Introduction*. Cambridge University Press, 1980.

[33] B. Chlebus. Domino-tiling games. *Journal of Computer and System Sciences*, 32:374–392, 1986.

[34] T. Clausing. Behavioural vs. habitual rationality and backward induction. In H. de Swart, editor, *Logic, Game Theory and Social Choice*, pages 46–56. Tilburg University Press, 1999.

[35] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. MIT Press, 1990.

[36] G. D'Agostino. *Modal Logic and non-well-founded Set Theory: translation, bisimulation, interpolation*. PhD thesis, University of Amsterdam, 1998.

[37] B.A. Davey and H.A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, 1992.

[38] E. Dijkstra. *A Discipline of Programming*. Prentice-Hall, 1976.

[39] E. Dijkstra and C. Scholten. *Predicate Calculus and Program Semantics*. Springer, 1990.

[40] H. van Ditmarsch. *Knowledge Games*. PhD thesis, University of Groningen, 2000.

[41] K. Doets. *Basic Model Theory*. CSLI, 1996.

[42] P. van Emde Boas. The convenience of tilings. In A. Sorbi, editor, *Complexity, Logic and Recursion Theory*, Lecture Notes in Pure and Applied Mathematics 187, pages 331–363. Dekker, 1997.

[43] E. A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 995–1072. MIT Press, 1990.

[44] E. A. Emerson. Model checking and the mu-calculus. In N. Immerman and P. Kolaitis, editors, *Descriptive Complexity and Finite Models*. DIMACS 31. American Mathematical Society, 1997.

[45] E. A. Emerson and C. S. Jutla. On simultaneously determinizing and complementing ω-automata. In *Proceedings of the 4th IEEE Symposium on Logic in Computer Science (LICS)*, pages 333–342, 1989.

[46] E. A. Emerson and C.-L. Lei. Efficient model checking in fragments of the propositional mu-calculus. In *Proceedings of the 1st IEEE Symposium on Logic in Computer Science (LICS)*, pages 267–278, 1986.

[47] M. Fitting. *Proof Methods for Modal and Intuitionistic Logics*. Reidel, 1983.

[48] M. Fitting. Basic modal logic. In D. Gabbay, C. Hogger, and J. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 1, pages 365–448. Oxford University Press, 1993.

[49] W. Fokkink. *Introduction to Process Algebra*. Springer, 2000.

[50] P. Gärdenfors. Rights, games and social choice. *Noûs*, 15:341–356, 1981.

[51] O. Gasquet and A. Herzig. From classical to normal modal logics. In H. Wansing, editor, *Proof Theory of Modal Logic*. Kluwer, 1996.

[52] A. Gibbard. A pareto-consistent libertarian claim. *Journal of Economic Theory*, 7(4):388–410, 1974.

[53] R. van Glabbeek. *Comparative Concurrency Semantics and Refinement of Actions*. PhD thesis, Free University of Amsterdam, 1990.

[54] R. Goldblatt. *Logics of Time and Computation*. CSLI, second edition, 1992.

[55] V. Goranko. The basic algebra of game equivalences. In M. Pauly and G. Sandu, editors, *Proceedings of the ESSLLI Workshop on Logic and Games*, 2001.

[56] V. Goranko. Coalition games and alternating temporal logics. In J. van Benthem, editor, *Proceedings of the 8th conference on Theoretical Aspects of Rationality and Knowledge (TARK VIII)*, pages 259–272. Morgan Kaufmann, 2001.

[57] B. Greene. *The Elegant Universe: Superstrings, Hidden Dimensions, and the Quest for the Ultimate Theory.* Vintage Books, 2000.

[58] J. Halpern and Y. Moses. A guide to the modal logics of knowledge and belief. In A. Joshi, editor, *Proceedings of the Ninth International Joint Conference on Artificial Intelligence (IJCAI-85)*, pages 480–490. Morgan Kaufmann, 1985.

[59] D. Harel. Dynamic logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume II. Kluwer, 1984.

[60] D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic.* MIT Press, 2000.

[61] D. Harel and R. Sherman. Looping vs. repeating in dynamic logic. *Information and Control*, 55:175–192, 1982.

[62] P. Harrenstein, W. van der Hoek, and J.-J. Meyer. A modal interpretation of Nash-equilibria and some related concepts. In G. Bonanno, E. Colombatto, and W. van der Hoek, editors, *Proceedings of the Fourth Conference on Logic and the Foundations of Game and Decision Theory (LOFT4)*, 2000.

[63] M. van Hees. *Rights, Liberalism and Social Choice.* PhD thesis, University of Nijmegen, 1994.

[64] M. van Hees. *Rights and Decisions. Formal Models of Law and Liberalism.* Kluwer, 1995.

[65] J. Hintikka and G. Sandu. Game-theoretical semantics. In J. van Benthem and A. ter Meulen, editors, *Handbook of Logic and Language*, pages 361–410. Elsevier, 1993.

[66] W. Hodges. Elementary predicate logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume I. Kluwer, 1984.

[67] W. Hodges. *Model Theory.* Cambridge University Press, 1993.

[68] M. Hollenberg. *Logic and Bisimulation.* PhD thesis, University of Utrecht, 1998.

[69] J. Huizinga. *Homo ludens: proeve eener bepaling van het spel-element der cultuur.* Tjeenk Willink, 1938.

[70] M. Huth and M. Ryan. *Logic in Computer Science: Modelling and Reasoning about Systems.* Cambridge University Press, 2000.

[71] D. Janin and I. Walukiewicz. On the expressive completeness of the propositional mu-calculus with respect to monadic second order logic. In *Proceedings of CONCUR '96*, LNCS 1119, pages 263–277. Springer, 1996.

[72] N. D. Jones. *Computability and Complexity - From a Programming Perspective.* MIT Press, 1997.

[73] A. Kaldewaij. *Programming: the derivation of algorithms.* Prentice Hall, 1990.

[74] M. Kaneko and T. Nagashima. Game logic and its applications I. *Studia Logica*, 57:325–354, 1996.

[75] J. Kelly. *Social Choice Theory: An Introduction.* Springer, 1988.

[76] D. Kozen. Results on the propositional μ-calculus. *Theoretical Computer Science*, 27:333–354, 1983.

[77] D. Kozen and J. Tiuryn. Logics of programs. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B. MIT Press, 1990.

[78] M. Kracht and F. Wolter. Normal monomodal logics can simulate all others. *Journal of Symbolic Logic*, 64(1):99–138, 1999.

[79] R. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM Journal of Computing*, 6(3):467–480, 1977.

[80] W. Leininger. The fatal vote: Berlin versus Bonn. *Finanzarchiv*, 50:1–20, 1993.

[81] G. Lenzi. *The MU-calculus and the Hierarchy Problem.* PhD thesis, University of Pisa, 1997.

[82] J. Loeckx and K. Sieber. *The Foundations of Program Verification.* Wiley-Teubner, second edition, 1987.

[83] P. Lorenzen and K. Lorenz. *Dialogische Logik.* Wissenschaftliche Buchgesellschaft, 1978.

[84] N. Miller. *Committees, Agendas, and Voting.* Harwood Academic Publishers, 1995.

[85] M. Mizutani, Y. Hiraide, and H. Nishino. Computational complexity to verify the unstability of effectivity function. *International Journal of Game Theory*, 22:225–239, 1993.

[86] H. Moulin. *The Strategy of Social Choice.* North-Holland, 1983.

[87] H. Moulin and B. Peleg. Cores of effectivity functions and implementation theory. *Journal of Mathematical Economics*, 10(1):115–145, 1982.

[88] F. Nack. *AUTEUR: The Application of Video Semantics and Theme Representation for Automated Film Editing.* PhD thesis, Lancaster University, 1996.

[89] I. Netchitailov. An extension of game logic with parallel operators. Master's thesis, Institute for Logic, Language and Computation, Amsterdam, 2000.

[90] I. Netchitailov. Principles of alternating game logic. Master's thesis, St. Petersburg State University, St. Petersburg, 2001.

[91] H. R. Nielson and F. Nielson. *Semantics with Applications.* Wiley, 1992.

[92] D. Niwinski. The propositional μ-calculus is more expressive than the propositional dynamic logic of looping. Unpublished manuscript, 1984.

[93] M. Osborne and A. Rubinstein. *A Course in Game Theory.* MIT Press, 1994.

[94] G.-J. Otten, P. Borm, T. Storcken, and S. Tijs. Effectivity functions and associated claim game correspondences. *Games and Economic Behavior*, 9:172–190, 1995.

[95] R. Parikh. Social software. *Synthese.* To appear.

[96] R. Parikh. Propositional logics of programs: New directions. In M. Karpinski, editor, *Foundations of Computation Theory*, LNCS 158, pages 347–359. Springer, 1983.

[97] R. Parikh. The logic of games and its applications. In M. Karpinski and J. van Leeuwen, editors, *Topics in the Theory of Computation*, Annals of Discrete Mathematics 24. Elsevier, 1985.

[98] M. Pauly. A modal logic for coalitional power in games. *Journal of Logic and Computation.* To appear.

[99] M. Pauly. Transforming predicates or updating states - total correctness in dynamic logic and structured programming. Master's thesis, Institute for Logic, Language and Computation, Amsterdam, 1997.

[100] M. Pauly. From programs to games: Invariance and safety for bisimulation. In P. Clote and H. Schwichtenberg, editors, *Computer Science Logic*, LNCS 1862. Springer, 2000.

[101] M. Pauly. An introduction to game logic. In M. Faller, S. Kaufmann, and M. Pauly, editors, *Formalizing the Dynamics of Information*. CSLI, 2000.

[102] M. Pauly. Coalitional ability in multi-agent systems: A logical approach. Technical Report SS-01-03, American Association for Artificial Intelligence, 2001. Papers from the 2001 AAAI Spring Symposium.

[103] M. Pauly. A logical framework for coalitional effectivity in dynamic procedures. *Bulletin of Economic Research*, 53(4):305–324, 2001.

[104] M. Pauly. On the complexity of coalitional reasoning. In E. Yanovskaya, editor, *Proceedings of the 2nd International Conference on Logic, Game Theory and Social Choice*. St. Petersburg State University, 2001.

[105] B. Peleg. Effectivity functions, game forms, games and rights. *Social Choice and Welfare*, 15:67–80, 1998.

[106] B. Peleg, H. Peters, and T. Storcken. Nash consistent representation of constitutions: A reaction to the Gibbard paradox. Technical Report METEOR Research Memorandum RM/00/023, University of Maastricht, 2000.

[107] D. Peleg. Communication in concurrent dynamic logic. *Journal of Computer and System Sciences*, 35:23–58, 1987.

[108] D. Peleg. Concurrent dynamic logic. *Journal of the ACM*, 34(2):450–479, 1987.

[109] A. Pietarinen. *Games Logic Plays: Informational Independence in Game-Theoretic Semantics*. PhD thesis, University of Sussex, 2000.

[110] S. Rahman and H. Rückert. Dialogische Modallogik (für T,B,S4 und S5). *Logique et Analyse*. To appear.

[111] S. Rahman and H. Rückert. Eine neue dialogische Semantik für lineare Logik. Unpublished manuscript.

[112] A. Rubinstein. *Economics and Language*. Cambridge University Press, 2000.

[113] M. Shubik. *Game Theory in the Social Sciences*. MIT Press, 1982.

[114] E. Sperner. Ein Satz über Untermengen einer endlichen Menge. *Mathematische Zeitschrift*, 27:544–548, 1928.

[115] A. Storcken and H. de Swart. *Verkiezingen, Agenda's en Manipulatie*. Epsilon Uitgaven, 1992.

[116] R. Streett. Propositional temporal logic of looping and converse is elementarily decidable. *Information and Control*, 54:121–141, 1982.

[117] R. Streett and E. A. Emerson. An automata theoretic decision procedure for the propositional mu-calculus. *Information and Computation*, 81:249–264, 1989.

[118] A. Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5:285–309, 1955.

[119] M. Tennenholtz and Y. Moses. On cooperation in a multi-entity model (preliminary report). In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI-89)*, pages 918–923, 1989.

[120] M. Vardi. On the complexity of epistemic reasoning. In *Proceedings of the 4th annual Symposium on Logic in Computer Science (LICS)*, pages 243–252, 1989.

[121] M. Vardi. Why is modal logic so robustly decidable? In N. Immerman and P. Kolaitis, editors, *Descriptive Complexity and Finite Models*, DIMACS 31. American Mathematical Society, 1997.

[122] Y. Venema. Representing game algebras. In M. Pauly and G. Sandu, editors, *Proceedings of the ESSLLI Workshop on Logic and Games*, 2001.

[123] A. Vilks. A logic for changing beliefs with applications to reasoning about choice and games. In H. de Swart, editor, *Logic, Game Theory and Social Choice*, pages 68–76. Tilburg University Press, 1999.

[124] I. Walukiewicz. A note on the completeness of Kozen's axiomatisation of the propositional μ-calculus. *Bulletin of Symbolic Logic*, 2(3):349–366, 1996.

[125] E. Werner. What can agents do together? In *Proceedings of the European Conference on Artificial Intelligence (ECAI-90)*, pages 694–701, 1990.

[126] M. Wooldridge and M. Fisher. A first-order branching time logic of multi-agent systems. In B. Neumann, editor, *Proceedings of the 10th European Conference on Artificial Intelligence (ECAI-92)*, pages 234–238, 1992.

[127] E. Zermelo. Über eine Anwendung der Mengenlehre auf die Theorie des Schachspiels. In E. Hobson and A. Love, editors, *Proceedings of the 5th Intl. Congress of Mathematicians*, volume II, pages 501–504. Cambridge University Press, 1913.

Index

List of symbols

Titles in the ILLC Dissertation Series:

ILLC DS-1996-01: **Lex Hendriks**
Computations in Propositional Logic

ILLC DS-1996-02: **Angelo Montanari**
Metric and Layered Temporal Logic for Time Granularity

ILLC DS-1996-03: **Martin H. van den Berg**
Some Aspects of the Internal Structure of Discourse: the Dynamics of Nominal Anaphora

ILLC DS-1996-04: **Jeroen Bruggeman**
Formalizing Organizational Ecology

ILLC DS-1997-01: **Ronald Cramer**
Modular Design of Secure yet Practical Cryptographic Protocols

ILLC DS-1997-02: **Nataša Rakić**
Common Sense Time and Special Relativity

ILLC DS-1997-03: **Arthur Nieuwendijk**
On Logic. Inquiries into the Justification of Deduction

ILLC DS-1997-04: **Atocha Aliseda-LLera**
Seeking Explanations: Abduction in Logic, Philosophy of Science and Artificial Intelligence

ILLC DS-1997-05: **Harry Stein**
The Fiber and the Fabric: An Inquiry into Wittgenstein's Views on Rule-Following and Linguistic Normativity

ILLC DS-1997-06: **Leonie Bosveld - de Smet**
On Mass and Plural Quantification. The Case of French 'des'/'du'-NP's

ILLC DS-1998-01: **Sebastiaan A. Terwijn**
Computability and Measure

ILLC DS-1998-02: **Sjoerd D. Zwart**
Approach to the Truth: Verisimilitude and Truthlikeness

ILLC DS-1998-03: **Peter Grunwald**
The Minimum Description Length Principle and Reasoning under Uncertainty

ILLC DS-1998-04: **Giovanna d'Agostino**
Modal Logic and Non-Well-Founded Set Theory: Translation, Bisimulation, Interpolation

ILLC DS-1998-05: **Mehdi Dastani**
Languages of Perception

ILLC DS-1999-01: **Jelle Gerbrandy**
Bisimulations on Planet Kripke

ILLC DS-1999-02: **Khalil Sima'an**
Learning efficient disambiguation

ILLC DS-1999-03: **Jaap Maat**
Philosophical Languages in the Seventeenth Century: Dalgarno, Wilkins, Leibniz

ILLC DS-1999-04: **Barbara Terhal**
Quantum Algorithms and Quantum Entanglement

ILLC DS-2000-01: **Renata Wassermann**
Resource Bounded Belief Revision

ILLC DS-2000-02: **Jaap Kamps**
A Logical Approach to Computational Theory Building (with applications to sociology)

ILLC DS-2000-03: **Marco Vervoort**
Games, Walks and Grammars: Problems I've Worked On

ILLC DS-2000-04: **Paul van Ulsen**
E.W. Beth als logicus

ILLC DS-2000-05: **Carlos Areces**
Logic Engineering. The Case of Description and Hybrid Logics

ILLC DS-2000-06: **Hans van Ditmarsch**
Knowledge Games

ILLC DS-2000-07: **Egbert L.J. Fortuin**
Polysemy or monosemy: Interpretation of the imperative and the dative-infinitive construction in Russian

ILLC DS-2001-01: **Maria Aloni**
Quantification under Conceptual Covers

ILLC DS-2001-02: **Alexander van den Bosch**
Rationality in Discovery - a study of Logic, Cognition, Computation and Neuropharmacology

ILLC DS-2001-03: **Erik de Haas**
Logics For OO Information Systems: a Semantic Study of Object Orientation from a Categorial Substructural Perspective

ILLC DS-2001-04: **Rosalie Iemhoff**
Provability Logic and Admissible Rules

ILLC DS-2001-05: **Eva Hoogland**
Definability and Interpolation: Model-theoretic investigations

ILLC DS-2001-06: **Ronald de Wolf**
Quantum Computing and Communication Complexity

ILLC DS-2001-07: **Katsumi Sasaki**
Logics and Provability

ILLC DS-2001-08: **Allard Tamminga**
Belief Dynamics. (Epistemo)logical Investigations

ILLC DS-2001-09: **Gwen Kerdiles**
Saying It with Pictures: a Logical Landscape of Conceptual Graphs

ILLC DS-2001-10: **Marc Pauly**
Logic for Social Software