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 $\varphi$ 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 $\varphi$ 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 $\text{NP} \neq \text{PSPACE}$. 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 $\gamma$ for achieving an outcome which satisfies $\varphi$. 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 $\mu$-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.