Marc Pauly  
Rohit Parikh

**Game Logic — An Overview**

**Abstract.** Game Logic is a modal logic which extends Propositional Dynamic Logic by generalising its semantics and adding a new operator to the language. The logic can be used to reason about determined 2-player games. We present an overview of meta-theoretic results regarding this logic, also covering the algebraic version of the logic known as Game Algebra.

**Keywords:** modal logic, propositional dynamic logic, game theory

**1. Introduction**

When two people are faced with dividing a cake among themselves, *you cut, I choose* is a well-known algorithm which guarantees a fair division. This algorithm can be generalised to yield a fair division of the cake among more than 2 people, the so-called Banach-Knaster (BK) last diminisher procedure [10].

The setting is that there are assumed to be $n$ people who want to share a cake fairly. However they have no scales or other measuring devices. How do they achieve the goal of fair sharing? What happens in the BK procedure is as follows. The first person takes a slice from the cake which she asserts is her fair share. The slice is then looked at by the other people from 2, 3 up to $n$. Anyone who finds the current slice too big may diminish it and put some cake back into the main part. When everyone has looked at the cake, one of two things must have happened. Either no one reduced the slice, in which case player 1 takes the slice. The other possibility is that someone did reduce the slice in which case the last reducer or *diminisher* takes the reduced slice. In any case we now have a smaller cake and $n - 1$ people to share it. We repeat the procedure.

In what sense is this procedure fair? Let $\mu$ be a measure which has value 1 for the whole cake and is finitely additive. Then if player $i$ receives the piece $P_i$ she has a winning strategy to make sure that $\mu(P_i) \geq \frac{1}{n}$.

---

1 Clearly, the players have no time to take advantage of countable additivity if it should obtain.

Special Issue on Game Logic and Game Algebra

*Edited by Marc Pauly and Rohit Parikh*

The procedure need not be fair in another sense. Suppose player $i$ is on a diet and she would like to make sure that $\mu(P_i) \leq \frac{1}{n}$. Can she always achieve this with the BK-procedure? No, she cannot, for suppose she is player 1 among 3. Suppose now that player 1 picks out a slice which player 2 may or may not reduce, but after her player 3 reduces the slice to a very small value, say 0.1, and being the last diminisher, takes it. Now player 1 who is still in the game cuts a slice somehow, player 2 reduces it again to a small value and takes it, and now player 1 is stuck with a large piece. So the BK-procedure which is fair for greedy players is not fair for players on diet. However its mirror image, the last increaser procedure, does achieve this.

What distinguishes even the 2-person algorithm from the kind of algorithms usually treated in computer science is that the former involves more than one agent and also the goals of the agents. A procedure like the BK-procedure allows all agents to achieve their goals. But of course for a zero sum game, where the players' goals conflict, this would not be possible. For the one-agent case, we are essentially dealing with a (possibly nondeterministic) program which can be constructed using operations such as sequential composition, iteration, etc. Logics such as Propositional Dynamic Logic (PDL) [14, 15] have been developed for reasoning about such programs, and the meta-theoretic study of these logics has given us valuable insights, e.g., into the complexity of reasoning about programs and the expressive power of various programming constructs.

When moving from 1-agent algorithms to many-agent algorithms like the BK procedure, these program logics are not sufficient anymore. A natural way to conceive of algorithms for more than one agent is in terms of games. Game Logic, introduced in [17, 18], is a generalisation of PDL for reasoning about determined 2-player games, allowing us to describe algorithms like the cake-cutting algorithm and to reason about their correctness.

On a less applied note, one reason why Game Logic is an interesting object of study is the following: By comparing Game Logic to program logics like PDL, we can get an idea of how reasoning about games differs from reasoning about programs. This comparison can be carried out along various technical dimensions. First, there are the standard comparisons in terms of axiomatisation, complexity and expressive power. Second, one can compare program operations to game operations, trying to isolate the operations which essentially distinguish programs from games. Third, Game Logic also implicitly formulates a notion of game equivalence. Based on this notion of equivalence, one can investigate the algebra of game operations and compare it to its algebraic counterpart for programs, process algebra [12, 2].